University Of California
Santa Cruz

Play Design

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor In Philosophy

in

Computer Science

by

Chaim Gingold

June 2016

The Dissertation of Chaim Gingold
is approved:

Professor Michael Mateas, Chair

Professor Nancy Chen

Henry Lowood, Ph.D.

Professor Janet Murray

Professor Noah Wardrip-Fruin

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Contents

List of Figures

Abstract

Play Design

Chaim Gingold

This thesis argues that it is productive to consider playthings, playmates, playgrounds, and play practices as constituting a set with shared design characteristics. I make this argument by analyzing a series of examples, teasing out commonalities, and articulating these as play design principles. Since play is central to human experience, play design techniques allow us to design for the whole human, whether we are crafting games, tools, learning experiences, or playthings. *SimCity,* a software plaything that confounds game-centric approaches (e.g. game studies and game design), is the keystone in an arch of case studies that takes us from some of the earliest examples of computer simulation all the way to model cities enacted with children, cardboard, and costumes, and unusual playgrounds made of junk.

Before turning to the case studies that lead to the principles of play design, we must first address two foundational methodological points:

First, in order to analyze something as play, we must be able to speak constructively about play itself, which is a bewildering subject. In chapter 1, *Play*, we review the literature on play, reconciling multiple perspectives and definitions, and distill seven play characteristics that underpin the thesis.

Second, in order to analyze software, we must have methods for doing so. Chapter 2, *Software*, advances an analytical framework for this purpose. This is a methodological contribution to the nascent field of software studies, which seeks to interpret the semi-visible infrastructure of computing that mediates modern life, from our bodies and our most intimate relationships to our public and political lives. To link software to play, I introduce an additional analytical

framework for considering software as a resource for play.

Will Wright created *SimCity* to amuse himself and learn about cities. To build it, he appropriated from multiple traditions in which computers are used as tools for modeling and thinking about the world as a complex system, most notably system dynamics and cellular automata. Wright's make believe play was scaffolded by these software practices, which offered inspiration and guidance, as well as abstract computational primitives for world building. Chapters 3–5 trace the historical contexts and origins of *SimCity's* many design influences, from system dynamics (chapter 3) and cellular automata (chapter 4)—two very different ways of seeing, thinking about, and computationally representing the world—to *Pinball Construction Set* and *Raid on Bungling Bay* (chapter 5).

Taking up the evolution of software in this way allows us to see how it is formed, what it is made of, and how ideas are embedded within and perpetuated by it. Deconstruction also helps us to understand software as a medium of dynamic representation, a scaffold for thought, an aesthetic experience, and its appeal as a resource for play.

In Chapter 5, *SimBusiness*, I give a historical account of *SimCity's* creation and the social circumstances that shaped its design, and sketch the history of Maxis, the company that marshaled and published *SimCity*. The trajectory of Maxis offers a parable about play and creativity. We see in Maxis's formation and unraveling the inescapable tension between play and capitalism, and between intrinsic and extrinsic play—the private autotelic play that innovates and creates, and the public play of player-consumers that pays the bills.

Chapter 6, *SimCity,* completes the *SimCity* case study by considering it as play artifact and experience. Using extensive diagrams that translate and map its code, I perform a close reading of *SimCity,* explaining how it conjures the illusion of a miniature living city, and how this living world scaffolds play.

Two non-digital examples round out the play design case studies. In chapter 7, *City Building Education*, we look at Doreen Nelson's practice of building and role playing model cities with children in classrooms. Nelson's simulation is an excellent counterpoint to Wright's, and their comparison elucidates many play design principles. Chapter 8, *Adventure Playground*, looks at an unusual playground in which children build with junk, and play with risks and materials, like wood, paint, and nails, that are typically withheld from them. In addition to illuminating principles of play design, the adventure playground tradition reveals play's "refructifying" (Sutton-Smith 1999) capacity to sweep up everything, even the detritus of civilization, and creatively reimagine it. Conceived amidst the darkness of World War II, adventure playgrounds illustrate how life transcends ruin through play—an important lesson for the 21st century's unfolding challenges.

In chapter 9, *Play Design*, I articulate play design principles drawn from the case studies. The principles are analytical, enabling us to see how play is scaffolded, as well as generative, prescribing design strategies for scaffolding play. This analytical-generative pairing enables us to deconstruct the design of a plaything, and transfer these design techniques to a new project—a technique that should be of interest to the educators, marketers, and designers of all stripes who have often envied the deep focus, enthusiasm, and pleasure afforded by make believe caves, dungeons, cities, and computationally animated living worlds. Play design is also deeply relevant to new embodiments of computation on the horizon, such as augmented reality and tangible dynamic media. Play is profoundly appropriative, and good play designs teach us how to robustly accommodate unpredictable environments and activities—a key design consideration for builders of such systems, which must gracefully take in the human world in all its glorious messiness.

Acknowledgments

I am indebted to my advisor, Michael Mateas, and my committee members, Nancy Chen, Henry Lowood, Janet Murray, and Noah Wardrip-Fruin, for scaffolding my research. Henry not only offered energizing ideas and conversations, but helped steer me towards Santa Cruz, where Michael and Noah entertained my somewhat divergent interests, and gave me just the right amount of structure to bring them to fruition. Michael's earnest critiques and encouragement helped me to undertake such an expansive project, as well as focus and strengthen my arguments. I am lucky that Nancy agreed to host me, a wayward game designer and computer scientist, in her graduate anthropology seminar, which was a thrilling and transformative experience. Without her guidance and encouragement, my adventure playground research would never have developed past the daydream stage. I am profoundly grateful to Janet for what has became a lifelong friendship and career-long mentorship. Without her encouragement I never would have gone back to do a Ph.D., which has now become the latest *I told you so* that she and I can laugh about.

This project was made possible by the generosity of many interview subjects and collaborators. *SimCity* is, by now, probably one of Will Wright's least favorite subjects of conversation, and I am grateful to him not only for making *SimCity* and mentoring me early in my career, but in agreeing to speak with me, yet again, about *SimCity's* creation. For a man with so many projects and interests, Will was gracious in indulging me in a conversation about what was, in retrospect, only a fleeting dalliance. Jeff Braun was magnanimous in sharing his time, entertaining stories, and photographs. His openness and insightful recollections were crucial in helping me piece together the overarching narrative of the "SimBusiness" chapter, and guided me towards new interview subjects and ideas. I am grateful to Don Bayless, Michael Bremer,

Brian Conrad, Claire Curtin, Mick Foley, Don Hopkins, and Bruce Joffe for being good sports in reminiscing events decades past. Don Hopkins should be celebrated for his efforts to port and preserve *SimCity,* which led directly to its open sourcing, and his generosity in sharing some archived and publicly unreleased code with me. My research builds directly upon his efforts.

I am fortunate that Doreen Nelson, whose teaching methodology I discuss in the chapter on City Building Education, engaged me in hours of spirited and far-ranging conversation. She invited me to visit and observe a teaching training workshop at Pasadena Art Center, which was an invaluable experience. She is one of the many fascinating people this project has led me to, and it has been a pleasure to get to know her.

The Berkeley Adventure Playground, and my chapter about it, exist thanks to the guidance, generosity, and expertise of Patty Donald, who has been running the playground for well over thirty years. Patty is a treasure trove for adventure playground enthusiasts around the world. I am fortunate that she was willing to invite me onto her playground, thoughtfully balancing my needs with those of the wider community she is responsible to, and was so giving of her time, experience, stories, and photographs. As should be clear from the resulting adventure playground chapter, it would not have been possible without her participation. The players and staff of the playground also deserve my thanks, as does Claire Latané, the landscape architect who worked on redesigning the adventure playground in Irvine, California. While co-chairing the IndieCade 2013 conference, I extended a speaking invitation to Claire, who graciously accepted and produced a talk whose research I draw from.

I am grateful to Jio Javier for transcribing most of my research interviews, as well as the efforts of a few other transcribers whose names I do not know.

This project flourished through the support of multiple institutions. The University of Califor-

nia Santa Cruz, and Michael Mateas and Noah Wardrip-Fruin's Expressive Intelligence Studio research lab in the Computer Science department, provided the initial sponsorship for my research. Bret Victor and Alan Kay, by bringing me into the orbit of the Communications Design Group, enabled this research to achieve a scope and depth which reflects, I hope, the space and resources that were put at my disposal. I am grateful to SAP for funding me while at CDG, and to the many individuals who sustain it. CDG is a wonderfully supportive and inspiring community, and I am honored to have been part of it. Y Combinator Research, to which I am also grateful, provided support during the final month of the project.

Software often goes unseen, a semi-visible infrastructure that sustains our public and private lives. This research is no different, and while it would be futile to catalog the vast menagerie of software that has enabled my work, I believe it is important to acknowledge and celebrate some of these tools. I found *Zotero*, a free bibliography manager, to be transformative—especially when used in conjunction with Google Scholar and Google Books. Of course bibliography management would have been of little use without the fabulous University of California Santa Cruz library, which quickly became one of my favorite parts of going back to school. Emulators, complex programs that bring back and animate long dead computer hardware and software, were vital research tools. I used *OpenEmu, Power64, SheepShaver, VICE, Virtual][,* and *Virtual64*. The emulators, like any tool I might acknowledge, are simply the surface phenomena of deep and diffuse social enterprise. Emulator programmers, along with those who archive old programs, computer hardware, documentation, and ephemera, are uncelebrated preservationists of computing cultures that might vanish as quickly as they appeared, were it not for their efforts.

By reading over incomplete drafts and talking through partially formed ideas, many people, too numerous to fully enumerate, have encouraged me and helped to shape this project:

Göetz Bachmann, Nancy Chen, Sherol Chen, Gonzalo Frasca, Mark Guzdial, Emily Harris, Eric Kaltman, Marko Röder, May-Li Khoe, Henry Lowood, Michael Mateas, Janet Murray, Robert Ochshorn, Lea Redmond, Alex Rosmarin, Toby Schachman, Fred Turner, Marc ten Bosch, Bret Victor, and Noah Wardrip-Fruin. Janet Murray coached and guided at critical moments, read heroic quantities of very rough text, and helped me to conceptualize the project's diverse parts as a whole, and think through its many details. Her ideas and influence can be felt throughout.

Thank you to my friends and family, especially my parents, for scaffolding a life full of play, creativity, adventure, warmth, and learning.

Chapter 1

Play

Figure 1.1: “Now you just throw it back, apparently.” *Cartoon by Liam Walsh. From* The New Yorker*, August 24, 2015.*

Why look at play?

In this thesis, play moves to center stage. In the sister disciplines of game studies and game design, it enacts a supporting role, and toys such as *SimCity* are seen as degenerate games, lacking a win or lose condition. But play is more than this. It is more than a liminal game, the nimbus of games, or an elemental ingredient. Play is a category in its own right, and demands its own design framework and analytic criteria. By turning the table on games, playthings such as *SimCity* move to the center, opening up new horizons of scholarship and design. By analyzing *SimCity* alongside the family of playthings and play practices to which it rightly belongs, we can consider these on their own terms. In this framing, as we will see, games draw from the deep well of play, but are simply an ossified sub-genre of it.[1]

---

[1] I aim to account for play, a grey zone in Jesper Juul's definition of game: activities and objects with only some characteristics of games (Juul 2005). Rather than see these activities and objects as liminal or degenerate games (as Juul's scheme does), or allow the definition of "game" to slip and apply to playful activities which are not actually games (as is common usage), I wish to consider them on their own terms, as *play*. Play constitutes a category of its own.

By using the word "game" to describe activities which are not (as is commonly done, but not in Juul's paper) we unwittingly import a whole set of formulations (a contest, fixed rules, etc…), expectations, analytic criteria, and design tools into a foreign domain, awkwardly blinding ourselves to the unique qualities and themes of play. If *SimCity* is a toy, and not a game, as both players and Will Wright maintain, then surely we can say more about its shared features and qualities with other forms of play than simply that they are games lacking valorized outcomes. Juul's definitional grey zone is not robust enough to account for play and its unique themes and variegated forms. Play must be dealt with on its own terms.

Definitions of "game" project normative sensibilities (e.g. Juul 2005) at odds with the rich variety of design issues play and game designers actually engage with. Nonetheless, the discourse of game design has expanded to encompass art games, interactive personal expressions, and liminal playthings like *SimCity* under the banner of "game." Part of this is linguistic: play is (mostly) a verb, and game is (mostly) a noun.

(Witness the awkwardness of my not using "game" for playthings, play experiences, and play practices. This is language dependent. While game is usable as noun ("Stop the game?") or verb ("We'll game the system."), its noun usage predominates. English is peculiar in its employment of two words to make this distinction (Parlett 1999). Many European (e.g. French and German) languages do not differentiate between play and game, a limitation that Gonzalo Frasca observes Roger Caillois attempted to overcome by employing the latinate terms *paida* and *ludus* (free-form vs. rule governed) to work (Frasca 2007, p. 38). Translations of Huizinga and Caillois, influential scholars who wrote in German and French, use both "game" and "play" freely, hiding this terminological issue from those of us reading their English translations.)

This distinction, however, is crucial. Discourses of games talk about non-games like *SimCity* as games, yet we can learn more about *SimCity's* design—I claim—by comparing it to toys and non-game play experiences than we can by comparing it to games. While one could design computer games as "software" (rather than "games")

The shift in perspective from game to play is the crux of my argument: to properly understand things like *SimCity* we need a framework for conceptualizing them as play, not games. While it is productive to think of *SimCity* as a game, or as software (which I will do later), a play design perspective is called for.[2] Will Wright, the creator of *SimCity,* corroborates this claim:

> Toys can be used to build games. You can play games with toys. But you can also engage in more freeform play with toys. It doesn't have to be a goal-directed activity. I think of toys as being more open-ended than games. We can use a ball to play a game such as basketball, or we can just toss the ball back and forth, or I can experiment with the ball, bouncing it off of different things. So, I would think of toys as a broader category. Also, toys can be combined. I can strap Barbie to my R.C. car and drive her around, thus making up a new activity by combining toys. Games tend to be isolated universes where there's a rule set, and once you leave that universe the rule set is meaningless. …
>
> Our games [Wright's software toys] are more like a hobby, which you approach in a different way. Like with a model train set, some people get totally into the scenery and the details on the cliffs and the hills. Other people get into the little village in the middle. Other people get into the switching on the tracks. And sometimes these will play off of each other when a community builds around a hobby. You'll have certain people in the community who are very into certain aspects of the hobby and they have expertise which they can teach to other people. And you have subspecializations within the community. People can create things and trade them, or they can just share ideas. I tend to think of hobbies as being a bit more community based than the cinematic model (Rouse 2001).

Rather than stretch the boundaries of games, and ways of thinking about them, play designers perceive their work as part of a broad continuum of practices that includes hobbies, toys, and a universe of playful practices and objects. What if *SimCity* were a prototypical rather than a liminal example? Moving play from the periphery to the center allows us to ask new questions. What shared characteristics unify play experiences? What would it mean to

---

this arrangement projects a different set of priorities and design sensibilities than "game." It is the same with play.

[2] I first used the phrase "play design" in (Gingold 2009).

think and design in terms of such shared play characteristics? And if games, in the end, are no longer central, then where do they fit?

In order to talk and think in terms of play, we turn now to the problem of defining and conceptualizing it. The remainder of this chapter touches upon the problem of defining play, and then conducts a thorough review of many influential conceptions of play. Readers uninterested in a detailed survey of the play literature may wish to hop to the chapter's end, to the definitional matrix, and get straight to the discussion of the seven characteristics of play that I take from this review. This comparative analysis and reconciliation of play is done in the spirit of Jesper Juul's distillation of game definitions, a search "for a Heart of Gameness" (Juul 2003).

The Ambiguity of Play

Rhetorics of Play

Scholars of play are highly attuned to the challenge of defining it (Sutton-Smith 1997). While play has been studied from a variety of disciplines, from neuroscience (Panksepp 2004), ethology (Fagen 1981; Burghardt 2005), developmental psychology (Garvey 1977/1990; Vygotsky 1978), and folklore (Sutton-Smith 1959), to anthropology (Geertz 1972), history (Huizinga 1955; Cross 1990), performance studies (Schechner 1988), philosophy (Wittgenstein 1953; Bateson 1955; Caillois 1961; Suits 1978; Spariosu 1989), and game design (Crawford 1984; Salen and Zimmerman 2003)—to name just a few—a comprehensive and clear image of what play is and what can be said about it remains elusive. The renowned play scholar Brian Sutton-Smith wrote towards the end of his long and productive career that "[a] skeptic… might easily come to the conclusion that the future of play theory is that play theory has no future. There seems to be so many difficulties in locating the appropriate phenomena as well as in locating a useful methodology to capture such phenomena" (Sutton-Smith 1995).

Sutton-Smith proposed a framework of play rhetorics to organize the seemingly incompatible methodologies and epistemologies of play scholarship, a theme he fully took up in *The Ambiguity of Play*, which cataloged seven rhetorics of play (Sutton-Smith 1997). Sutton-Smith classifies these rhetorics as either ancient or modern. The ancient rhetorics are *fate* (chance and gambling), *power* (contests), *identity* (communal celebrations and festivals), and the *frivolous* (carnivals and tricksters). The modern rhetorics of play are more familiar: *progress* (play as learning), *imaginary* (creativity, art), and *self* (personal experience such as relaxation, satisfaction, and escape). Sutton-Smith offers counterbalance to what he describes as the modern idealization of play, observing that there is little scientific evidence to support the claim that play helps young children and animals to learn, imagination often yields the phantasmagoric and irrational (not just positive innovation and creativity), and that play often takes anti-social forms (play that is dark or cruel).

Design Rhetoric of Play

One piece of practical advice I take from Sutton-Smith's analysis is that all play scholarship is based on a particular interpretation or rhetoric of what play is; and to offer a new definition of play is to engage in a rhetorical practice. To Sutton-Smith's set of rhetorics I propose a design rhetoric of play. Contemporary game design textbooks, practitioners of computer game development, and game studies can all be seen as participating in a design rhetoric of play, one which engages play from generative and reflective stances. But these discourses traditionally focus on games, not play.

What are the desired qualities of a play design rhetoric? Play design should afford:

- **Broad applicability.** It should apply to toys, computer software, and playgrounds. Its design principles also be useful for non-play applications like interface, industrial, and architectural design.

- **Functional explanation.** It should be possible to take a play thing or play practice, and use play design to understand how it engenders play. One should be able to explain why a play design principle works.

- **Design transfer.** Abstraction of these functional explanations enables the transfer of play design ideas from one artifact or practice to another.

- **Prescription.** Like game design, play design can be articulated as a set of principles and best practices—like those that appear in game design textbooks and manuals. It should be useful for designing play experiences.

- **Distinction between games and play.** Play design will harbor a different set of core values and commitments than game design, and articulate those differences. The relation between play and games should be easy to articulate in play design terminology. Positive definitions of both play and games should result. Play and game, in other words, should not be constructed negatively, in terms of what they lack. Ultimately, we will come to see games as a genre of play. Games, as we will see most clearly in the final chapter, are built out of the substrate of play.

To formulate this play design rhetoric, I will opportunistically adopt ideas from from across various play discourses. Theory as well as empirical examples, from history, science, and case studies, should inform this undertaking. But to get to a play-centric design discourse, we first need to come to terms with play itself. What is play? How should we talk about it?

Play Definitions

Play is ripe with ambiguity and paradox. This leaves us with the very practical question of how to employ such a slippery concept as play. A review of the play literature, however, reveals some clear patterns. Despite its ambiguity and multidisciplinarity, play, it seems, harbors consistent characteristics. We turn now to a summary of key definitions of play from across its multitudinous literature.[3] These definitions come from seminal play theorists (e.g. Huizinga, Caillois, and Sutton-Smith) as well as diverse fields: ethology (animal behavior), developmental psychology (child play), philosophy, performance studies, and game design. As will become evident, a coherent set of motifs resonate throughout. These motifs inform the seven characteristics of play that function as this chapter's payoff.

Johan Huizinga

Huizinga's seminal book on play, *Homo Ludens,* offers the following definition:[4]

> Summing up the formal characteristics of play we might call it a free activity standing quite consciously outside "ordinary" life as being "not serious", but at the same time absorbing the player intensely and utterly. It is an activity connected with no material interest, and no profit can be gained by it. It proceeds within its own proper boundaries of time and space according to fixed rules and in an orderly manner. It promotes the formation of social groupings which tend to surround themselves with secrecy and to stress their difference from the common world by disguise or other means (Huizinga 1955, p.13).

[3]Some of the play definitions below continue on to define games, which I've incorporated to help us better compare and contrast the two.

[4]Huizinga often reiterates his definition of play in *Homo Ludens*, sometimes with different inflections.

This requires some unpacking, which can begin in Huizinga's own words. "Play creates order, is order. Into an imperfect world and into the confusion of life it brings a temporary, a limited perfection. Play demands order absolute and supreme." "[I]t stands outside the immediate satisfaction of wants and appetites, indeed it interrupts the appetitive process," an "*interlude* in our daily lives" that "contains its own course and meaning." For Huizinga, play is a transient and self-contained alternate order.

Another important quality he calls attention to is "tension," which "means uncertainty, chanciness; a striving to decide the issue and so end it." Tension, the emotional product, is a figure that stands for uncertainty, agency, luck, and emotional investment.

Huizinga dwells on how play is often thought of as "non-seriousness," and is characterized by a *"disinterestedness,"* but "being 'only a pretend' does not by any means prevent it from proceeding with the utmost seriousness, with an absorption, a devotion that passes into rapture, and temporarily at least, completely abolishes that troublesome 'only' feeling." This dialectic between non-serious and serious can be destabilizing: "A game can at any time wholly run away with the players. The contrast between play and seriousness is always fluid." This calls attention to the fact that the play frame is socially sustained, fragile: "The play-mood is *labile* in its very nature. At any moment 'ordinary life' may reassert its rights either by an impact from without, which interrupts the game, or by an offence against the rules, or else from within, by a collapse of the play spirit, a sobering, a disenchantment."

That play as a joint, collaborative activity promotes social formation seems eminently logical to me—we tend to play with our friends, and befriend those we play with. And as play becomes conventionalized and institutionalized the order, rules, and norms of play become embedded within a specific community of practice—think of the "secret" society of *Pokémon* players, speaking a language and knowing a world alien to their parents. Huizinga explains further:

> The "differentness" and secrecy of play are most vividly expressed in "dressing up." Here the "extra-ordinary" nature of play reaches perfection. The disguised or masked individual "plays" another part, another being. He is another being. The terrors of childhood, open-hearted gaiety, mystic fantasy and sacred awe are all inextricably entangled in this strange business of masks and disguises.

This is play's quality as make believe, as earnest transformation, as acceptance of that which is not. Play as *pretense*. If one is not "in on" the specific transformations at work—what rules are in effect, what stands for what, and who has been transformed into whom—then one is in the dark, outside of the secret. Players accept an alternate "order absolute and supreme" in play: certain rules, restrictions, and make believe prevail. This transformation of ordinary life into something else can be marked by masks and disguises, and the earnestness with which this pretense is accepted mirrors the dialectic of non-serious/serious mentioned earlier. Turning to "savage" and Western ritual, Huizinga argues that in play there is a "unity and indivisibility of belief and unbelief." That is, while play commands our earnest absorption in its alternate order, we are simultaneously aware that it is "only a pretend." Play is simultaneously accepted as real and unreal.

And action within the real/unreal sphere of play has a special enactive status. Speaking of rites as forms of play, "The rite produces the effect which is then not so much *shown figuratively* as *actually reproduced* in the action. The function of the rite, therefore, is far from being merely imitative; it causes the worshippers to participate in the sacred happening itself." Play is not representation, but enactment. Soccer does not represent a contest; it is a contest.

TABLE OF CONTENTS

Figure 1.2: Table of contents for Huizinga's Homo Ludens *(Huizinga 1955).*

It is important to note that the focus of Huizinga's scholarship is not on play itself, but on play as the substrate of human civilization. While his book begins with a discussion of the play of animals and children, he is not interested in play as fun and games. We can see this clearly in the subtitle of *Homo Ludens,* which is *A Study of the Play Element in Culture.* The object of his study is the serious business of language, ritual, law, art, philosophy, and war as complex social play—expressions of an enigmatic, universal, and deep well of play.[5]

Roger Caillois

Huizinga attends to social play as an expansive phenomena underpinning the serious business of civilization, but Roger Caillois, responding to Huizinga's work[6], hones in on a definition of play that could be used to study what we would commonly recognize as play and games—kite

[5]This deep well of play, Huizinga argues, is shared by animals and humans and lies beyond comprehension. Much scholarship on play can be seen to descend into this mysterious well.

[6]He is clearly in conversation with Huizinga in (Caillois 1957).

flying, make believe games, board games, sports, and so on—rather than its broader cultural manifestations, as was Huizinga's aim. Caillois defines play as:

> 1. Free: in which playing is not obligatory; if it were, it would at once lose its attractive and joyous quality as diversion;
> 2. Separate: circumscribed within limits of space and time, defined and fixed in advance;
> 3. Uncertain: the course of which cannot be determined, nor the result attained beforehand, and some latitude for innovations being left to the player's initiative;
> 4. Unproductive: creating neither goods, nor wealth, nor new elements of any kind; and, except for the exchange of property among the players, ending in a situation identical to that prevailing at the beginning of the game;
> 5. Governed by rules: under conventions that suspend ordinary laws, and for the moment establish new legislation, which alone counts;
> 6. Make-believe: accompanied by a special awareness of a second reality or of a free unreality, as against real life.
>
> (Caillois 1961, pp. 9–10)

Caillois introduces a few other useful concepts. He classifies games and play[7] as contests (agon), chance (alea), fantasy (mimicry; make believe play), or vertigo (ilinx; physical play and thrills). This isn't a terribly robust taxonomy, but it does provide a nice vocabulary for considering qualities of play. Caillois also posits a continuum of play, ranging from the rule dominated (ludus) to the free form (paidia).

Gregory Bateson

Gregory Bateson's observation that play is framed through metacommunicative paradox has been massively influential (e.g. Bekoff 1974a; Schechner 1988; Sutton-Smith 1997; Salen and Zimmerman 2003). He uses play to meditate on communication, and argues that the ability to interpret and generate non-literal gestures underpins both play and communication.

[7] Writing in French, he just has one root word to work with for both game and play (jeu/joue).

> What I encountered at the zoo was a phenomenon well known to everybody: I saw two young monkeys playing. i.e., engaged in an interactive sequence of which the unit actions or signals were similar to but not the same as those of combat. It was evident, even to the human observer, that the sequence as a whole was not combat, and evident to the human observer that to the participant monkeys this was "not combat" (Bateson 1955).

Echoing Huizinga, Bateson observes that the enacted phenomena is simultaneously real and unreal (combat and not combat). To sustain this effect, players, Bateson reasoned, "must often be reminded that "This is play." ":

> Now, this phenomenon, play, could only occur if the participant organisms were capable of some degree of metacommunication. i.e., of exchanging signals which would carry the message "This is play" (Bateson 1955).

Bateson reasoned that play unfolds "within a delimited psychological frame, a spatial and temporal bounding of a set of interactive messages." The "labile" "frame" of play is maintained by "metalinguistic and metacommunicative messages" that signal " 'This is play.' " Paradox serves a critical role in this metacommunicative messaging. The play frame signifies that

> 'These actions, in which we now engage, do not denote what would be denoted by those actions which these actions denote.' The playful nip denotes the bite, but it does not denote what would be denoted by the bite (Bateson 1955).

With the contradiction itself serving as a signal of play, action and meaning is transformed, giving rise to make believe irreality: "play, fantasy, and art."

> Paradox is doubly present in the signals which are exchanged within the context of play, fantasy, threat, etc. Not only does the playful nip not denote what would be denoted by the bite for which it stands, but, in addition, the bite itself is fictional. Not only do the playing animals not quite mean what they are saying but, also, they are usually communicating about something which does not exist (Bateson 1955).

Bateson is not after a definition of play, nor does he produce one—his interests lie instead in play as a peculiar mode of communication. Nonetheless, Bateson's analysis of play as metacommunicative paradox pervades play scholarship—from performance studies to ethology, and so merits our consideration.

Catherine Garvey

A researcher of child play, Garvey defines play as follows:

> (1) Play is pleasurable, enjoyable. Even when not actually accompanied by signs of mirth, it is still positively valued by the player.
>
> (2) Play has no extrinsic goals. Its motivations are intrinsic and serve no other objectives. In fact, it is more an enjoyment of means than an effort devoted to some particular end. In utilitarian terms, it is inherently unproductive.
>
> (3) Play is spontaneous and voluntary. It is not obligatory but is freely chosen by the player.
>
> (4) Play involves some active engagement on the part of the player.
>
> (5) Play has certain systematic relations to what is not play.

The last property, according to Catherine Garvey, is the most intriguing as the same behavior "can be performed as play but can also, of course, be performed in a nonplay fashion and with nonplayful intent. … All play requires the players to understand that what is done is not what it appears to be. It is this nonliteral attitude that allows play to be buffered from its consequences," which has been called "behavior in the simulative mode" by Peter Reynolds (Garvey 1977/1990, pp. 4–7).

Her summary definition of play is "a special subjective orientation to resources" (Garvey 1990, p. 103). One can play with—have a special subjective orientation to—a wide variety of resources. In her book *Play,* Garvey discusses the following resources: motion and interaction, objects, language, social materials (e.g., people, roles, plans), and rules (following or subverting). Play often takes "ritualized" forms Garvey characterizes as "controlled repetition," replete with rhythm, role-reversal, call and response, and iterative improvisation.

Pretend play is "a voluntary transformation of the Here and Now, the You and Me, and the This or That, along with any potential for action that these components of a situation may have" accompanied by redundant markings that "signal the transformation." Pretend play is often scaffolded by parents and siblings. Play scaffolding can take other forms, as when caretakers attribute intentionality to the unintentional acts of infants.

Play is inextricable from communication and intersubjectivity. Garvey writes that "[b]oth smiling and playing are implicated in the child's first experiences of mutually shared attention and awareness with his parents and become more and more systematically linked with communicative exchanges." Play is marked and facilitated by metacommunication, from insincerity in verbal play to the interleaving into pretend play of out-of-role signals for setting up, prompting, correcting, and negotiating such play. Players, in other words, are fluent in the boundary between reality and fantasy.

Safety and well being are preconditions for play. She writes that "laughter as a sign of joy and play behavior are both likely to appear under conditions of well-being. The sick, bewildered, frightened child does not smile or play."

Garvey defines games in terms of play. Games are "play activities that have been institutionalized … structured by explicit rules that can be precisely communicated; games can be taught and learned." They "tend to have traditional names" and the "quality of 'social objects'—that is, a game has a clear beginning and end, and its structure can be specified in terms of moves in a fixed sequence with a limited set of procedures for certain contingencies" (Garvey 1977/1990).

Lev Vygotsky

The psychologist Vygotsky approaches play from a developmental perspective, seeing it not just as a means to abstract thought, but "a major source of development."[8] Play arises in young children in order to satisfy "desires that cannot be immediately gratified or forgotten … the preschool child enters an imaginary, illusory world in which the unrealizable desires can be realized, and this world is what we call play." For Vygotsky, imagination is central to play: "in play a child creates an imaginary situation."

He sees imagination as underpinning every type of play, including play that is overtly rule based, such as chess. Simultaneously, he sees rules as underpinning not just rule based play, but also fantasy play:

> One could go even further and propose that there is no such thing as play without rules. The imaginary situation of any form of play already contains rules of behavior, although it may not be a game with formulated rules laid down in advance. The child imagines himself to be the mother and the doll to be the child, so he must obey the rules of maternal behavior. … Just as the imaginary situation has to contain rules of behavior, so every game with rules contains an imaginary situation. For example, playing chess creates an imaginary situation. Why? Because the knight, king, queen, and so forth can only move in specified ways; because covering and taking pieces are purely chess concepts. … Just as we were able to show at the beginning that every imaginary situation contains rules in a concealed form, we have also demonstrated the reverse—that every game with rules contains an imaginary situation in a concealed form (Vygotsky 1978).

In play, the meaning of things is separated from the things themselves, "liberating the child from constraints." This extends to motivation, as "things lose their determining force," and the "*[t]he child sees one thing but acts differently in relation to what he sees."* Things take on new meanings: "In play thought is separated from objects and action arises from ideas rather than

[8]He sees play as creating a zone of proximal development (ZPD), an idea we will examine more closely in chapter 2.

from things: a piece of wood begins to be a doll and a stick becomes a horse." Play is a "transitional stage" for the "severing" and "[t]ransfer of meanings," a process aided by the use of what Vygotsky refers to as *pivots*.

When a player appropriates something, they invest it with a new meaning and purpose. Pivots are things, like a stick, banana, or tiny pewter hat, that receive the meaning of other things—like a horse, telephone, or Monopoly player. Pivots are instruments for transferring the meaning of objects and actions—"severing the meaning of horse from a real horse"—as well as vessels that hold severed meanings: " 'the-horse-in-the-stick.' " To use Vygotsky's example, a stick is a pivot that allows the meaning of horse to detach from a real horse, and attach to the stick. Pivots are not signs, but embodied props one enacts with, and whose embodied properties constrain and correspond to their possible meanings.[9] They are objects that stand for objects. A pivot's embodied and enactive properties—that the stick can be placed between the legs and ridden like a horse—are essential. They influence the kinds of meanings players can transfer into the pivot. A stick, as Vygotsky points out, is more amenable than a postcard to becoming a make believe horse (Vygotsky 1978).[10]

It is not just things and actions whose meanings transform in play. "[P]lay gives a child a new form of desires." Players themselves are transformed into "a fictitious 'I'," subordinating existent desires to new ones taken on through play (Vygotsky 1978).

---

[9]The choice of object cannot be arbitrary, as it would be in full blown symbolism: "This is not to say that properties of things as such have no meaning. Any stick can be a horse but, for example, a postcard cannot be a horse for a child. … A symbol is a sign, the stick does not function as the sign of a horse for the child, who retains the properties of things but changes their meaning. Their meaning, in play, becomes the central point and objects are moved from a dominant to a subordinate position" (Vygotsky 1978). Vygotsky conceives of play and pivots developmentally; these are intermediate stages in the journey to fully abstract and symbolic thinking and activity.

[10]Another example of a thing standing for a thing is in how the materials which comprise Japanese gardens, like rocks, plants, and water, are used to evoke naturalistic phenomena such as mountains, waterfalls, habitats, oceans, and lakes (Slawson 1991, p. 61).

Richard Schechner

Richard Schechner, from the discipline of performance studies, offers a gritty and iconoclastic "theory-to-be" of play:

> A coherent theory of play would assert that play and ritual are complementary, ethologically based behaviors which in humans continue undiminished throughout life; that play creates its own (permeable) boundaries and realms: multiple realities that are slippery, porous, and full of creative lying and deceit; that play is dangerous and, because it is, players need to feel secure in order to begin playing; that the perils of playing are often masked or disguised by saying that play is fun, voluntary, a leisure activity, or ephemeral—when in fact the fun of playing, when there is fun, is in playing with fire, going in over one's head, inverting accepted procedures and hierarchies; that play is performative involving players, directors, spectators, and commentators in a quadralogical exchange that, because each kind of participant often has her or his own passionately pursued goals, is frequently at cross-purposes. …
>
> Security is needed at the outset of play more than later on. Once play is under way, risk, danger, and insecurity are part of playing's thrill. Usually there is a safety net, or a chance to call 'time out,' or appeal to an umpire or other nonplaying authority who take care of the rules (Schechner 1988).

Schechner seeks to confound our normative sense of play, which quarantines play in firmly framed boundaries. Schechner argues that "[i]n the West, play is a rotten category tainted by unreality, inauthenticity, duplicity, make-believe, looseness, fooling around, and inconsequentiality. Play's reputation has been a little uplifted by being associated with ritual and game theory. … [In the West,] the multiple realities of playing are situated inside a pyramidical hierarchy of increasing reality leading from unreal make-believe to 'just the facts, Ma'am.' " Using the Sanskrit notions of maya and lila (illusion and play, roughly) Schechner contrasts this with an alternate cosmology of "multiple realities" in which both "reality and experience are networks of flexible constructions," and play, performance, and the phenomena of reality are not hermetically framed as much as porously "netted."

For Schechner our everyday phenomenological, social, and make believe realities are porous constructions netted out of the raw materials of experience. These realities are disrupted through play, which "is a mood, an attitude, a force. It erupts or one falls into it. It may persist for a fairly long time—as specific games, rites, and artistic performances do—or it comes and goes suddenly—a wisecrack, an ironic glimpse of things, a bend or crack in behavior."

"From this perspective" of porous realms, Schechner writes, "the Batsonian [sic] play frame is a rationalist attempt to stabilize and localize playing, to contain it safely within definable borders." Schechner quotes the influential anthropologist Victor Turner, finding support for his argument that play is not necessarily separate (or superfluous):

> Playfulness is a volatile, sometimes dangerously explosive essence, which cultural institutions seek to bottle or contain in the vials of games of competition, chance, and strength, in modes of simulation such as theatre, and in controlled disorientation, from roller coasters to dervish dancing—Callois's "ilinx"or vertigo (Turner 1986).

Schechner illustrates the porousness of play with the concept of "dark play," in which "all the players don't know they are playing," "subvert[ing] the metacommunicational aspect of the play frame," "as in *Candid Camera*, a scam or sting, or as the butt of a practical joke." But "dark play," inspected closely, isn't problematic. To be an unwitting participant in someone else's "dark play"—and thus "in the dark" about it—is to be shut out of the play frame in a very specific way. The unwitting participant is a *play-thing*, not a *play-mate* with whom one shares an intersubjective play frame. A kitten may play with an injured mouse, but the mouse cannot be said to be playing with the cat. "Dark play," with its appropriately antisocial overtones, is a tidy concept for describing this type of scenario.

Salen & Zimmerman

Focusing on a *mechanical* sense of play, Katie Salen and Eric Zimmerman's game design textbook defines play as "free movement within a more rigid structure". They find this definition applies to playing games ("Game play … is the experience of a game set into motion through the participation of players."), "ludic activity" (e.g. playing with a ball), as well as "being playful"—unusual behavior within social and mechanical constraints (Salen and Zimmerman 2003).

Salen and Zimmerman are focused on game design through authorship of rules, so it makes sense that they would see play primarily in mechanical terms, as the animating force that puts the rules of a game through its paces, activating the space of possible action and meaning. With this focus, they highlight three senses of play: game play, ludic activity, and attitude. For Salen and Zimmerman, games are "a subset of play."

Gordon Burghardt

Animal play researchers are faced with a very pragmatic epistemic and ontological problem. How do you know if an animal is playing? In a recent comprehensive review of the literature on animal play, Gordon Burghardt tells of play in mammals, yes, but also birds, lizards, and fish. Burghardt visits many existing definitions of animal play, finds them lacking rigor as tools for identifying play, and offers a "usable operational method" in the form of five criteria that "must [all] be met in at least one respect before the play label can be attached":

1. Limited immediate function … includes elements, or is directed toward stimuli, that do not contribute to current survival
2. Endogenous component … spontaneous, voluntary, intentional, pleasurable, rewarding, reinforcing, or autotelic

3. Structural or temporal difference … differs from ‘serious’ performance… it is incomplete… exaggerated, awkward, or precocious; …modified form, sequencing, or targeting.

4. Repeated performance … performed repeatedly in a similar, but not rigidly stereotyped, form

5. Relaxed field … psychological safety … the behavior is initiated when … adequately fed, healthy, and free from stress

(Burghardt 2005)

Brian Sutton-Smith (1972)

Earlier in his career Sutton-Smith characterized play as follows:

> in play…each individual differentiates himself from his own entrapment and in so doing becomes, momentarily at least, a free spirit. … The player substitutes his own conventions and his own urgencies for those of society and nature. … This mixture of lowered tension in external relations and induced arousal within the novel constraints is probably the euphoric state we call fun (Sutton-Smith 1972).

“Cognitively speaking, being in control means a reversal of life structures,” which play performances themselves are not immune to, often resulting in play that is unstable, hilarious, or nonsensical. “Since voluntary controls are hard to achieve, play takes place most easily in a setting established for that purpose and with scaled down objects; i.e., dolls” (Sutton-Smith 1972).

Continuing on to games, he writes that “[i]n *games*, where the rules are based on mutual agreement, there is no reversal of the orders of convention and nature. They are mirrored rather than put aside.” Games promise “to reverse a man’s fate within these orders, though only in terms of the testing of powers over fate, nature, and others. There are many types of play, such as imitation, exploration, and construction, which are only moderately assimilated into the game format; they find more adequate representation in drama, magic, and museums;

while games give primary place to the play of contest. As well as rules, games also involve an opposition of forces and a disequilibrial outcome. … "Ring a Ring of Roses," are in fact the earliest form of games in which a contest takes place between the players and the forces of anarchy." As contests between players, "games are a microcosm of cultural fate, but one in which the customary order of superiority is, if not reversed, at least temporarily open for rectification" (Sutton-Smith 1972).

Brian Sutton-Smith (1986)

Sutton-Smith's *Toys as Culture* is written for a less academic audience than his other books, but is nonetheless steeped in scholarship. In it, he gives one of his richest and most accessible definitions of play. He defines play in terms of five propositions:

First, *"play is a primitive form of communication."* Animals and humans do it, and no creature needs to be taught how. It is as natural as "eating or sex" and "apparently satisfying to the participants." At the same time, practice produces more proficient players. In both animals and humans, play "requires "signals" " such as wagging tails, smiling faces, and exaggerations and distortions. "They make funny noises, they make absurd mimicry, their eyes gleam, they nudge and tickle, they laugh, they jump up and down, they gambol and they "galumph." "

Second, *"play is a primitive form of expression."*[11] As a form of communication, it has its own style which Sutton-Smith describes as "schematic … [p]lay is more akin to caricature or pantomime than it is no ordinary theatre." From sports to playing house, "play schematizes life." The "exaggerations, iconic sounds, mimicry, gamboling and galumphing of play" do more than signify play; they are intrinsic to "play's own structure."[12]

Third, play is a "*paradoxical* form of communication and expression." Sutton-Smith explicitly builds upon Bateson, writing that "a playful nip connotes a bite, but not what a bite connotes." Play acts simultaneously are and are not what they appear to be. This paradox allows psychologically troubling matters to safely manifest within play, allowing us to express what we cannot "in the light of day."[13]

---

[11] Propositions one and two reiterate points 4 and 6 of a Sutton-Smith play definition:

> 1. that play's definition must be broad rather than narrow, including passive or vicarious forms as well as the active participant forms, including daydreams as well as sports and festivals.
> 2. that it should apply to animals as well as humans, and children as well as adults.
> 3. that it should not be defined only in terms of the restricted modern Western values that say it is nonproductive, rational, voluntary, and fun. These are not concepts that can prevail as universals, given the larger historical and anthropological evidence to the contrary.
> 4. that play is not just an attitude or an experience; it is always characterized by its own distinct performances and stylizations.
> 5. that it can be as momentary as a piece of wit, or can endure as long as the one-year cycles of festivals or the four-year cycles of the Olympics. That is can be spatially either as diffuse as a daydream or as articulate as a sports stadium.
> 6. that play is like language: a system of communication and expression, not in itself either good or bad.

(Sutton-Smith 1997)

[12] Because of this, Sutton-Smith argues, play "retreats increasingly from the original objects of reference," becoming increasingly "ludicrous or schematic." He offers a neat elaboration of Huizinga's emphasis on secrecy: "The normal course of events for a playing group is to become increasingly arcane in their play signals and increasingly out of touch with others who are not in the game" (Sutton-Smith 1986).

[13] Bateson makes a similar point, observing that Hollywood is free to explore homosexual themes in a Puritanical culture, and connecting play to psychotherapy (Bateson 1955).

Fourth, play is inherently unstable and oscillatory. "Plays own induction of schematic rules is itself a violation of everyday rules, and *this state of affairs appears to generate within play a constant succession of bipolar and disequilibrating structures which are recurrently equilibrated."* Sutton-Smith gives examples of the resulting alternating repetition: "up and down, to and fro, in and out, back and forth," which are eventually elaborated into "chase and escape, attack and defend, score and outscore, win and lose."

"Fifth, play is a *primitive form of symbolization of the underlying motivations."* Play enables "similarly afflicted" individuals to "share meaning and receive companionship." In both animals and humans, play provides a kind of psychological release, which Sutton-Smith argues is part of why play seems "to mean so much to those who indulge in it, and so little to the others who do not happen to share those particular secrets."

> In sum, we metaphorize play as bathos, and we define it as primitive and paradoxical communication, schematic expression, and a succession of disequilibrial bipolar states, with their own rules, sequences and climaxes Brian Sutton-Smith (1986).

Play's onset is facilitated by special signals (paradoxical framing; a special form of expression and communication), a "relaxed state," and "ludic modelling." Sutton-Smith observes that play harbors two dialectics. First, within play, an internal dialectic between the forces of chaos and order. Second, an external dialectic between the stringencies of the external world, and their violations within play.

Following this logic, he argues that toys not only miniaturize "worldly phenomena to our own scale and our own terms," but are schematic and abstracted, which marks them as separate from ordinary life, and also affords clarity and familiarity, allowing them "to be subsumed to the players' own causes, to be "assimilated"[14] to the players' own fantasies"—a material

[14] Assimilation here probably refers to Piaget. Garvey describes play's assimilative quality—citing Piaget—that "rather than accommodating itself to perceived (or absolute) reality it transforms and absorbs its object to previously held perceptions" (Garvey 1974, p.170).

for the grist of playful attitudes and transformations (Sutton-Smith 1986).

Brian Sutton-Smith (1999)

Sutton-Smith's prolific career spanned four decades, so it should come as no surprise that he has articulated many definitions of play. This publication, which is one of his last, is the only I know of where his main aim is to define play. He writes:

> Play, as a unique form of adaptive variability, instigates an imagined but equilibrial reality within which disequilibrial exigencies can be paradoxically simulated and give rise to the pleasurable effects of excitement and optimism. The genres of such play are humor, skill, pretence, fantasy, risk, contest, and celebrations, all of which are selective simulations of paradoxical variability.

The twinned terms equilibrial and disequilibrial are freighted by Sutton-Smith with multiple meanings. One meaning captures the "binary relationship of safety and risk within play." Equilibrial safety is joined to disequilibrial excitement. Play is "safe yet exciting." Play also transforms that which is not play. This is the second sense in which Sutton-Smith uses equilibrial/disequilibrial, as a mirroring and mocking of "the rest of society," a transformation which "relieves the boredom of our everyday succession of habitual events." The "variable existential contraries of everyday life" are transmuted and sublimated into new and "quirky" disequilibrial "syntheses, known culturally as simply playing, passing the time, playing games, fantasizing, partaking of contests, being festive and making celebrations, or being humorous."

While some celebrate play as freedom, as indeed Sutton-Smith has in his earlier writings, he recognizes that the matter is more complex. "Clearly, play is itself a compound of voluntary and involuntary elements, not totally one or the other." Play can be socially "obligatory," and in order to play one must submit to it, as one does when obeying the rules of a game.

He concludes by emphasizing the positive affective value of play:

> Now perhaps one finds in play not just a variability exercise, but instead a parody on the vicissitudes of our mortal existence. In such a case, play would now bring a redemption from earthly inadequacy. It would function more like a religion in offering alternatives to the everyday life, even though in the case of play these would be relatively temporary. Still, the opposite of play—if redefined in terms which stress its reinforcing optimism and excitement—is not work, it is depression. Players come out of their ludic paradoxes, whether as participants or fans, with renewed belief in the worthwhileness of merely living. The cry of "We're number one," or that I climbed Mount Everest, or that I have 10 Barbie dolls, or that I have another successful novel, or that I have a Motion Picture Academy Award, or that I am a master at chess, or that I won the lottery, or that I am a beauty queen, are all exultant cries, and they convince most of us, for a while at least, that the world is not such a bad place to be living in. Perhaps play, while it may continue biologically to be about energizing and making us more flexible, now psychologically is more like a branch of existential utopian philosophy, carrying us absurdly forward with optimism and confidence in the life we are leading. Perhaps as Baktin said so mysteriously about the Rabelaisian carnivals that he studied, play works because it is refructifying.

More prosaically, Sutton-Smith has been quoted as saying that

> We study play because *life* is *crap*. Life is crap, and it's full of pain and suffering, and the *only* thing that makes it worth living the *only* thing that makes it possible to get up in the morning and go on living is play. Art, and play (McGonigal 2012).[15]

Characteristics of Play

Certain themes clearly repeat across these diverse definitions. Putting the definitions into a matrix, as I have done below, not only teases out the components of each definition, but elicits comparisons. Through-lines and disagreements are foregrounded.

[15] I have transformed McGonigal's use of uppercase to italics.

	Intrinsic	**Safe**	**Transformational**	**Separate**
Huizinga	absorbing, free	not serious	pretend	outside “ordinary” life
Caillois	free		make-believe	unproductive
Sutton-Smith (1972)	arousal, euphoric, voluntary	lowered tensions in external relations	reversal	differentiates
Sutton-Smith (1986)	primitive, satisfying	relaxed state	schematizes life, ludicrous, bathos	paradoxical
Sutton-Smith (1999)	pleasurable, voluntary	safety	imagined, quirky, parody	simulated
Garvey	positively valued, spontaneous	well-being	transformation	buffered
Schechner	passionately pursued goals	security	multiple realities, inverting	boundaries, realms
Burghardt	endogenous, autotelic	relaxed field	difference, exaggerated, awkward	limited immediate function
Bateson			fantasy, paradox	spatial and temporal bounding
Salen and Zimmerman				
Vygotsky	fulfills … needs, freedom		imaginary situation, pivot	liberating the child from constraints

Figure 1.3: Matrix of play characteristics, part one. Cells contain quoted text from authors.

	Structured Variability			Attitude	Shared
	Structure	Variable	Agency		
Huizinga	order	uncertainty	striving	play-mood	promotes the formation of social groupings
Caillois	governed by rules	uncertain	*(implied)*		
Sutton-Smith (1972)	novel constraints		*(implied)*		
Sutton-Smith (1986)	bipolar and disequilibrating structures which are recurrently equilibrated				share meaning and receive companionship
Sutton-Smith (1999)	imagined but equilibrial reality	variability	*(implied)*		obligatory
Garvey	ritualized		active engagement	special subjective orientation	mutually shared attention
Schechner	rules	risk	performative	mood, attitude	dark play
Burghardt	performed repeatedly in a similar, but not rigidly stereotyped, form				
Bateson	delimeted		interactive sequence	psychological frame	metacommunication
Salen and Zimmerman	free movement within a more rigid structure			being playful	
Vygotsky	rules		action	new form of desires	

Figure 1.4: Matrix of play characteristics, part two. Cells contain quoted text from authors.

Each definition occupies one row of the matrix, and columns correspond to themes. For example, if we read across one row—say Caillois—we see the components of his definition:

free, make believe, unproductive, governed by rules, and uncertain. (Because of space constraints, the matrix is broken down into two halves, so Caillois's definition, along with every other row, spans both matrixes.) If we read up and down from his definition, we can make comparisons. Compare Caillois to Huizinga, who sits one row above, and you can clearly see how closely related these two definitions are.

The definitions we reviewed constitute the rows of this matrix. What happens if we read the columns, across definitions? What are the patterns?

Intrinsic

There is broad agreement that play is autotelic—performed for its own sake. It is described as absorbing (Huizinga), arousing and euphoric (Sutton-Smith), and pleasurable (Sutton-Smith and Garvey). Schechner voices the minority opinion, that play is not always fun. Fun, in any case, is an inexact and vexed term, which explains why it is so studiously avoided, and why more specific terms such as autotelic, absorbing, and arousal are used in its stead. Garvey's phrasing, that play is "positively valued by the player" is helpfully agnostic. Gamblers, for example, may not describe their experiences as pleasurable or fun, but they are nonetheless deeply absorbed (Malaby 2007; Schüll 2012).

Play is frequently characterized as voluntary or freely chosen. Observers of animal behavior identify behaviors as emerging from an internal biological drive. Play stands, alongside the drive for food and sex, as an endogenous behavior (Burghardt).[16] Play as an internal drive con-

[16]One of the basic arguments of Huizinga's *Homo Ludens* is that animals (including humans) share a basic underlying play capability, "the primaeval soil of play" (Huizinga 1955, p. 5). Jaak Panksepp, a neuroscientist specializing in emotion and play, identifies a handful of primary "emotional operating systems" in mammals, four that are always in effect: seeking (appetitive, stimulus seeking), fear, panic (expressing social dependence), and rage, plus three that are selectively activated: lust, care, and play. The drive to play manifests mostly during youth, but returns, to a degree, in parents caring for children. Panksepp makes a mirror image version of Huizinga's claim, arguing that parsimony demands we ascribe play's variegated and complex expressions to a single "ludic

cords with the evidence that play is freely chosen (Huizinga, Caillois, Sutton-Smith), voluntary (Garvey, Suits), and intrinsically motivated (Garvey). But question marks hang over this broad consensus. While Sutton-Smith describes play as voluntary, he also notes that it is sometimes a social obligation. And Schechner classifies pranks, whose targets don't choose to participate, as a form of asymmetric play called "dark play." But the butt of a prank is not actually playing, and is more plaything than playmate, and thus the object of someone else's play, not a player in their own right. These edge cases call attention to two important characteristics we will return to shortly: play as an *attitude,* and play as a *shared* activity.

Safe

Observers of play note that it has an important affective precondition: a feeling of psychological safety. To play, one must be comfortable. Animals and children play only when they are well fed and free of stress. Burghardt refers to this as a "relaxed field," and while it isn't in Garvey's enumerated definition, she does note that safety is a precondition to play. Sutton-Smith refers to this as a state of lowered tensions in external relations, a construction that leaves room for heightened tension internal to the play. Schechner, despite challenging many other play orthodoxies, agrees. While noting that play can be "deep" (e.g. Bentham 1871) and risky, he is attuned to the fact that play requires a feeling of safety in order to begin and continue. While play might feel exciting and risky, it is predicated upon safety. While not every definition

---

circuit." Evidence bolstering Panksepp's claim that our brains are wired for play is that the rats he studied clearly had an internally generated play drive. "The basic desire to play is not dependent on sensory inputs. It is an endogenous urge of the brain." Play deprivation produced rats with a higher desire to play, and rats would perform instrumental tasks (run through mazes) in order to play. Rats with various disabled senses solicited play less successfully, as the usual play signaling mechanisms had been harmed, but persisted in seeking play. Rats deprived of socialization during development still desired play, but were clumsier, and less desirable playmates (Panksepp 2004). Both rats and children prefer playmates with more play experience, and tend to play only when comfortable, and not hungry or tired, or feeling threatened (Garvey 1990). Play qualities that resurface and interoperate across species supports the hypothesis of shared ludic circuitry.

ticks off safety as a precondition, it's notable that there are no objections. The champions of this quality, furthermore, are the scholars who empirically study and engage with players of all kinds: children (Garvey and Sutton-Smith), animals (Burghardt), and performers (Schechner).

Transformation

Garvey writes that play's "most intriguing" characteristic is that it "has certain systematic relations to what is not play." Play's transformational quality is, perhaps, its most confounding one. Players build new consensual meanings, engage in pretense and make believe, and transform familiar materials and performances into new stylized forms. Actions that typically serve some instrumental purpose are appropriated and transmuted into stylized and unproductive gestures, performed for intrinsic purposes. With these transformations, players construct separate spheres of meaning and activity.

Play refracts the world into new meanings, pretenses, and configurations. Play is thought of as separate in large part because of how play and players transform the world via pretend, make believe, inversion, and paradox. Play is pretend and make believe (Huizinga, Caillois), a reversal (Sutton-Smith), an equilibrial-disequilibrial transformation of ordinary life (Sutton-Smith), "a voluntary transformation of the Here and Now, the You and Me, and the This or That" (Garvey), an inversion (Schechner), a modified and un-serious performance (Burghardt), and, in the case of games, a voluntarily accepted transformation of rules and means (Suits 1978), "an imaginary situation" (Vygotsky 1978).

Separate

We have seen that play is autotelic and voluntary—done for intrinsic motivations. That play is performed for its own sake means that it is not done for some other sake. Intrinsic and extrinsic motivations are opposed. Players play because they want to, not because they want to get something done. This unproductiveness, or separateness, is another banner theme of play, hoisted by a variety of descriptors: unprofitable (Huizinga), unproductive (Caillois, Garvey), lacking extrinsic goals (Garvey), means—rather than ends—oriented (Suits 1978), and limited function (Burghardt). Shades of play's unproductiveness can be seen in Suits's definition of games as inefficient (running around the track is not the quickest way to the finish line), and in hobbyists' voluntary employment of outdated and economically inefficient means: hiking and camping instead of driving; knitting clothing instead of buying it (Gelber 1999; Maines 2009).[17]

Play is often seen in developmental terms, confounding its separateness. While Sutton-Smith has written that a player "substitutes his own conventions and his own urgencies for those of society and nature," (Sutton-Smith 1972) he is self-consciously conflicted by this intrinsic/extrinsic binary. Despite having cataloged one of play's modern rhetorics as progress, as educational or improving—and highlighting the fact that there is little scientific evidence to underwrite this claim—Sutton-Smith can't help but contribute to the rhetoric of play as progress. He describes play's profit as "adaptive," both psychologically, and as an "adaptive potentiation" that contributes to survival by nourishing behavioral variability (Sutton-Smith 1997; Sutton-Smith 1999). Nonetheless, much of this hand wringing can be ascribed to play's recapitulation of what is not play, albeit in a stylized form that is "incompletely functional"

[17] I have not included Suits in my earlier review for a number of reasons, but he offers some useful ideas I will begin to draw upon.

(Burghardt 2010). Play's transformation of the instrumental into the non-instrumental kicks dust into our eyes, making it hard to see what, if any, gain it offers.

If play is unproductive, and done for its own sake, then it is somehow separate from the realm of the ordinary and everyday. Separateness is a note sung, in one way or another, by most of these definitions. Play is separate (Huizinga, Caillois), a simulation (Sutton-Smith), and nonliteral (Garvey). Play is framed via metacommunication (Bateson, Garvey) that signals what is play, separating it from the serious (Bekoff 1974b; Bekoff 1977).

Schechner identifies play as existing in its own realm, but challenges the Western idea that play is a nested reality within that of the everyday, arguing instead that play is one of many porous realms we inhabit. This porosity and lack of a fixed reference realm, however, fits comfortably with transformation. Play transforms what is not play, a process that does not leave behind a clear separation, but porous boundaries, and a confounding set of relationships.

Structured Variability

The definitions align on three other points: that play is variable, players exercise agency, and that play unfolds according to certain stereotyped patterns. In Garvey's account, it is when play becomes fixed into certain culturally repeatable, identifiable, and transmittable forms that it acquires the quality of a social object: a game. But play is patterned before it achieves such a level of fixity.

Structured. Salen and Zimmerman cast play as movement taking place within fixed constraints. Play projects novel constraints (Sutton-Smith 1972) and is an equilibrial structure (Sutton-Smith 1999). Huizinga and Caillois—focused on institutionalized forms of play such as games, ritual, and cultural practices—also note order and fixed rules. (Huizinga, remember, was interested in play as underwriting law, war, art, and philosophy.) Garvey's research shows that even not-games—non-institutionalized play—exhibits a high degree of patterning she refers to as "ritualized." Her research, which focuses a microscope on the verbal and non-verbal interactions of child play, reveals complex iterated patterns of repetition, rhythms, reversals, and grammatical transformations. Burghardt observes an analogous phenomena in his review of animal play literature. When animals play they repeat actions in a non-rigidly stereotyped form. Take, for example, the stereotyped play bow of canids, which also serves the metacommunicative function of inviting and delimiting play (Bekoff 1974a; Bekoff 1974b).

Figure 1.5: Canid play bow (Bekoff 1974b).

Variable. Play's variability is touched upon by many of these definitions. Play is characterized by tension—not knowing how things will turn out (Huizinga), uncertainty (Caillois), variability (Sutton-Smith), and risk (Schechner). Suits notes that the goals of play can be closed, in which an end or closure is sought, as in a competitive game, or open, in which the goal is to keep play going, as in an improvisational make believe act or infinite ping pong volley (Suits 1978). In either case, there is striving and an unknown outcome: variability. Huizinga's term “tension” does the best job of projecting the affective qualities of this indeterminacy, capturing the feeling of involvement with an unknown outcome. Variability is also seen in animal play. Burghardt writes that play actions are “performed repeatedly in a similar, but not rigidly stereotyped, form,” a characterization that evokes variability, agency, and structure.

Agency. Variability is closely linked to agency, for it is variation that permits active engagement. Agency is characterized as active engagement (Garvey), performance (Schechner), free movement (Salen and Zimmerman), action (Vygotsky), and striving (Huizinga). Even if not explicitly called out, agency is nonetheless implicit in many of these works (e.g. Sutton-Smith 1972).

Attitude

Play has a vital attitudinal quality: *playfulness* (Stenros 2015). Huizinga describes the “play-mood” and “play spirit.” Schechner describes play as a mood, which is consonant with Garvey’s summary definition of play as a “a special subjective orientation to resources.” Garvey identifies one quality of this orientation as a “non-literal attitude.” Suits describes the orientation that constitutes play as the “lusory attitude,” and his degenerate player modes—the spoilsport, trifle, and cheat—all point to breakdowns in shared attitude (Suits 1978).

One can play (perform) a game, but without a playful attitude. This is a strange idea until you realize that we are linguistically limited to *playing*—using the verb *play* for games—but this says nothing about the attitude of those who perform games. One can play un-playfully, as a spoilsport, trifle, or cheat, as one who has been compelled to participate, or simply be in the dark—an unwitting participant in someone else’s play, the victim of a prank. The attitude of play is labile, as Huizinga points out, and always at risk of falling apart, especially in instrumentalized play, such as gamification and serious (e.g. educational) games, or when players take the objectives of play too seriously. These are play without play—play the act without play the attitude. If play cannot be located exclusively in particular activities or materials, then we need to speak about attitude, the orientation towards activities and materials. And if play is a mood, then it can assert itself and appropriate anything, as when Tom Sawyer and Mary Poppins transform chores into games.

Shared

Play has profound social qualities. Huizinga notes that play, often accompanied by secrecy and disguises, "promotes the formation of social groupings." Players are set apart from the wider world by virtue of shared intent, shared pretense, and shared attention (see also Onishi et al. 2007). All of this is supported through metacommunication that synchronizes the intersubjective frames of play.

Shared Intent. In order to play together, players must collectively adopt playful attitudes. The intent to play must be shared, even if individual players harbor divergent playful goals (as in a contest).[18] Schechner's example of "dark play," in which someone is the butt of a joke or prank, calls attention to the intersubjective attitude. If someone is in the dark about play—the butt of a prank—then they are outside of a shared play frame; they are a plaything, not a playmate.

Shared Pretense. The transformations of play are underwritten by a collective as if attitude, a voluntary acceptance of what is not for what is.[19] It is through pretense that players come to

[18]Suits's spoilsport, trifle, and cheat point to breakdowns in the collective intent of play.

[19]Following Huizinga, Suits, and Vygotsky, I identify pretense as underwriting the transformations of both make believe and rule based play. Neither rules nor make believe have priority over one another: players abide by the imaginary rules of a game through make believe, and players enact the imaginary by consistently following

believe and participate in the shared make believe of both rules and fantasy. Shared pretense allows competitors to compete with one another within the shared pretense of a race. Pretense allows us to collectively invest the senseless with meaning, and fashion determinedly held purposes; pretense can also destroy meaning, demote value, and obliterate purpose.

Shared Attention. Play is a developmentally early and prototypical example of “mutually shared attention” and understanding (Garvey 1990). Attention is focused upon a shared scene of activity: players, actions, physical materials, make believe transformations, meanings, and so on.

As Bateson has first pointed out, play proceeds within a “delimited psychological frame” supported by “metalinguistic and metacommunicative messages.” Play’s stylized transformations, such as paradox and exaggeration, function as “play signals” (Bekoff and Allen 1998) that help to frame play as such (Bateson 1955; Bekoff 1977; Sutton-Smith 1986). Metacommunication helps to build the intersubjective play frame that supports shared intent, shared pretense, and shared attention.

Conclusion

Despite its ambiguity, play exhibits consistent characteristics. Summarized below, I use these characteristics to talk and think about play in the chapters that follow. My aim is not to propose a set of membership criteria for defining play, but rather to establish an analytical framework that underpins my discussion of play.

Intrinsic. Play is positively valued by players, who find it absorbing or pleasurable. Play is primarily voluntary and pursued for intrinsic, as opposed to extrinsic (for profit or gain),

---

behavioral rules (Huizinga 1955; Suits 1978; Vygotsky 1978; Onishi et. al. 2007).

purposes.

Safe. A precondition for play is a feeling of safety. Play emerges within a relaxed field, a state of lowered tension.

Transformational. Play has, following Garvey, special relations to what is not play. Play kaleidoscopically transforms, inverts, exaggerates, stylizes, varies, destabilizes, and reconfigures that which is not play.

Separate. Play transformations and actions are buffered from their ordinary consequences. Play acts are limited, unproductive, or simulative. Transformations are typically fleeting, transient, or otherwise bounded. Since players are encompassed by a social play frame, not everyone is privy to a play's pretense, separating players from non-players.

Following Schechner, this separation isn't simple and clean, but porous, netted, and unstable. Even if play is not, in actuality, separate, it is important to remember that it is, at the very least, perceived or experienced as separate from that which is not play.

Structured Variability. Play joins an ordered structure, such as ritual and rules and constraints, to variability. Huizinga identifies an affective quality of play's variability as tension—becoming emotionally engaged in play's unknown outcome. Variability also makes agency possible, the feeling of having a transformational effect.

Attitude. Play is a disposition, mood, or orientation. Following Garvey, I summarize play as a special orientation towards a broad array of materials—e.g. cultural, social, linguistic, and material.

Shared. Play involves shared intention, pretense, and attention. Shared pretense allows a collective understanding of happenings within an imaginary situation—whether fantasy or rule driven—to prevail. A shared attentional scene focuses activity around a jointly coherent set of people, things, and activity. Shared intent joins players into a collective playful undertaking.

The intersubjective play frame is coordinated through metacommunication. Many of the metacommunicative signals which coordinate play are symptoms of play's intrinsic transformations. For example: stylization, inversion, paradox, and exaggeration (e.g. Sutton-Smith 1986).

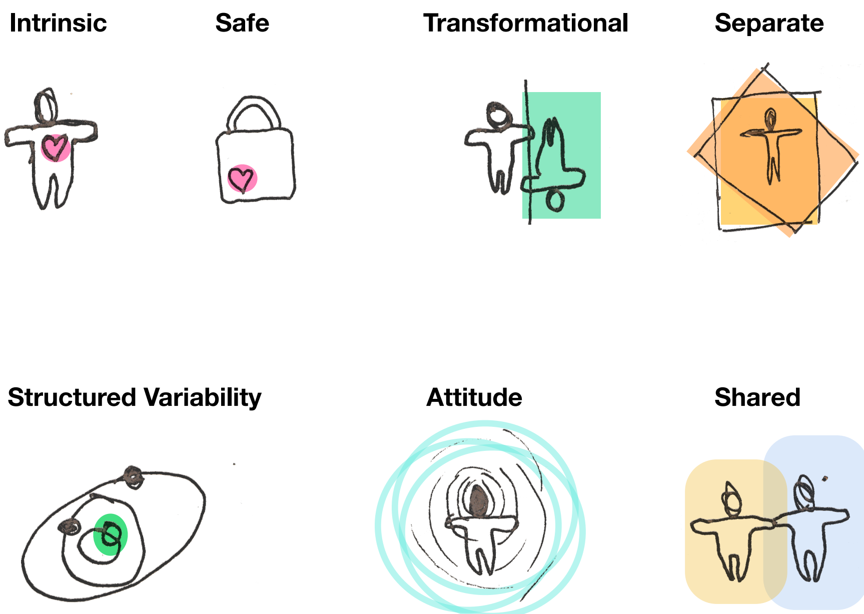


Figure 1.6: Characteristics of play.

Chapter 2

Software

> I am proposing a genre of writing one could call "computer criticism" by analogy with such disciplines as literary criticism and social criticism. … The computer is a medium of human expression and if it has not yet had its Shakespeares, its Michelangelos or its Einsteins, it will. … We have scarcely begun to grasp its human and social implications (Papert 1987).

> It is software that gives form and purpose to a programmable machine, much as a sculptor shapes clay (Kay 1984).

This thesis is composed of multiple case studies, many of which are software artifacts. But what does it mean to study software? Why should one do it? And how does one do such a thing? Software studies is a nascent field to which this thesis contributes methodology as well as case studies.

This chapter offers two analytical frameworks. The first is for studying software as software. I offer a rationale and set of techniques for studying and interpreting software programs, their source code, and the broader cultural, social, and aesthetic worlds in which they are enmeshed. I review extant theories, methods, and case studies of software, and regularize these sprawling and not entirely consonant approaches into a single framework. The framework maps the existing scholarship on software studies to my software case studies. Seen as ethnography, my close analysis of the evolution of system dynamics (chapter 3) provides the raw data from which I generalize the categories used in the framework.

The second framework is for studying software as play. Drawing upon the characteristics of play articulated in chapter one, I propose a way to think about computers as a resource for play. To help link software to play, I draw upon the notion of scaffolding.

Why Study Software?

The fact that *SimCity's* source code is available to study is a remarkable opportunity. *SimCity* has been open-sourced, which means that unlike most commercial software programs, its code is publicly available. This opens up new horizons for its interpretation. Furthermore, many of the preceding software and simulation works that informed *SimCity*, such as Jay Forrester's system dynamics, have many publicly available sources.

But why read code when I could interview Will Wright, the author of *SimCity*? Why not just play it? What can we hope to learn from reading code that isn't possible otherwise? The code, I claim, reveals influences and emergent code structures (e.g. bugs) that Will Wright might not be aware of or remember. For example, it is one thing to quote Wright describing how *SimCity's* design was informed by Forrester's work, but the final code is incontrovertible proof, exact in ways that English is not, and supportive of alternate possible interpretations. Such as: there is some discomfiture between Forrester's system dynamics and the rest of *SimCity* that Wright may have struggled with, but might not articulate. This mode of analysis is familiar to literary critics, cultural analysts, and historians, who would never rely purely on someone's comments and give up alternate interpretations of a source artifact or text. Neither is playing a substitute for reading code: the fact that *SimCity* is illusionistic means that outwardly perceptible behavior is not a reliable indicator of what is going on inside; we must look under the hood to understand how the simulation actually works if we are to study and learn from it. Code, in other words, provides essential and complementary material to work with.

What is Software?

To fully make the case that code should be studied, as well as articulate techniques for doing so, requires us to answer a deceptively simple sounding question: What is software? It might seem that software is just an app you download and install on your smart phone—a colorful icon, a word processor, a game, or a web browser. We typically view software as a tool or toy, usually some kind of product. Some see software as something designed, or something to design. Others might see it as intellectual property. Those with a technical disposition might see it as a collection of algorithms and data structures, "objects," source code, or even as an extended sequence of data encoded and transmitted through layers of material encodings such as magnetic marks, electrical impulses, and optical pulses (e.g. Kirschenbaum 2004).

But such conceptions are quite limited, effacing the experiential consequences of software as well as the social, cultural, and material circumstances that construct and constitute it. Software is made by particular people to advance particular agendas. These people wrestle with and work within the material limitations and affordances of computer hardware, software algorithms, human beings, and social organizations (Ensmenger 2010). The software they build is laced with the resulting compromises as well as their own, often unquestioned, world views.

I aim to convince you not only that software is worth studying, but of a particular conception of it that accounts for its technical and human dimensions. I conceive of software as an artifact fashioned by social forces in order to shape abstract computing machines to their particular ends. The resulting programs, functioning as media, then go on to reshape lived human experience along multiple avenues, from the psychological to the social. To motivate this formulation, we turn to the work of the historian Michael Mahoney. Mahoney's conception of software will help us to frame and organize the multitudinous ways in which software can be studied.

The Protean Machine

Mahoney observed that while the history of computing machines is a relatively developed field, the history of computer software is a nascent endeavor. If you travel to the The Computer History Museum in San Jose California, for example, you will find an incredible collection of computing devices bristling with historical context, narrative, and interpretative displays. The museum reflects the mature scholarly form of computer history, which focuses on the governmental and industrial contexts that drove the evolution of computing machinery, with their attendant emphases on data processing, calculation, and military command and control. The history of software, by contrast, is underdeveloped, as can be seen in both this museum and scholarly histories. In a corner of the museum's sprawling hardware collection, which ranges from slide rules and abaci to a hulking Cray supercomputer and an unassuming Apple I, you will find a single room dedicated to software—making apparent the relative maturity of these two endeavors. In short, historians understand how to do the history of hardware, but software is another matter. "Though hard to interpret, the hardware is at least tangible. Software by contrast is elusively intangible," writes Mahoney (Mahoney 1988).[1]

[1] Mahoney differentiates mature history from that of insider history, which is "full of facts and firsts," "pioneers," and the vivid yet uncritical contemporaneous journalistic accounts. Even the mature examples of computer

But why, Mahoney asks, should this be? What makes the history of computing hard? The answer is in the nature of electronic digital computers: labile, "protean," "a schema … which could assume many forms and could develop in many directions." This is theoretically true, as the Church-Turing thesis[2] makes clear, as well as materially evident. Consider the many forms computers have taken: watches, tablets, phones, personal computers, laptops, Tamagotchi, handheld calculators, databases, search engines, cash registers, calculating machines, ATMs, smart pens, toothbrushes, game systems, and so on. The computer, at heart, is an abstract symbol manipulating machine, and its multitudinous incarnations reflect not the essential "nature of the computer"—an automated symbol manipulator, labile and amorphous—as much as "the purposes and aspirations of the communities who guided those designs and wrote those programs" (Mahoney 2005).

The computer, in other words, is a protean machine. And "[i]t is software that gives form and purpose to a programmable machine, much as a sculptor shapes clay" (Kay 1984). But who are these sculptors? What can we say about their "purposes and aspirations" (Mahoney 2005)?

---

history end

> at the point where computing becomes a significant presence in science, technology, and society. There historians stand before the daunting complexity of a subject that has grown exponentially in size and variety, which looks not so much like an uncharted ocean as like a trackless jungle. We pace on the edge, pondering where to cut in (Mahoney 1988).

[2]The Church-Turing thesis, authored by Alan Turing and his Ph.D. advisor Alonzo Church, proved the equivalence of Turing machines, the Lambda calculus, and pretty much any symbol manipulation a human computer could do with pencil and paper. (Remember that "computer" once referred to a person—it was a job description.) Turing machines and the Lambda calculus are two alternate mathematical models of computation. Any computer which is "Turing complete" can compute anything that any other Turing machine, the Lambda calculus, or human computer can do. In short, once a symbol manipulating machine is general purpose enough, it can do anything any other kind of symbol manipulating machine can do.

Communities of Computing

These sculptors are members of communities who shape the protean machine, fashioning it to fulfill their aims. The plastic nature of computers make them into a kind of Rorschach test upon which practitioners—communities of developers, funders, and users—project their own histories, desires, and designs.[3] Mahoney argues that there is no single history of computing, but multiple histories "of what people wanted computers to do and how people designed computers to do it." There are histories of particular "communities of computing," of data processing, management, military command and control, calculation, intelligence augmentation, artificial intelligence, etc… (Mahoney 2005).

Figure 2.1: Nintendo manufactured Hanafuda cards before creating electronic games such as the Game & Watch series. Images from Wikipedia, by Francis Bijl (left) and Japanexperterna.se (right).

Mahoney argues that computing communities usually predate the computer itself, simply extending by computational means their own practices and history. Consider a Japanese hanafuda playing card manufacturer founded in 1889: Nintendo, whose name means "leave luck to heaven." Postwar, Nintendo tinkered with toys and novelties, taking up electronics and then computers, extending its own playful history into the realm of the digital, first with the electronic Love Tester novelty, then with the Game & Watch series of handheld games

[3] Both Mahoney and Sherry Turkle make versions of this argument (Turkle 1980; Mahoney 2005).

(inspired by, and using the same microprocessor as, a pocket calculator), and eventually the Famicom, released in the United States as the Nintendo Entertainment System (NES) (Sheff 1993; Izushi et al. 2010).[4]

That different communities transform and remake the protean computer in their own image makes the job of the historian difficult. There as many histories of computing as communities of computing.

Mahoney observes that the presentation software running on his laptop is enabled by

> some two dozen processes, each in itself a program of some complexity. All these programs reflect the histories of the communities from which they come. The operating system, for example, embodies the history of corporate organisations designed to distribute responsibilities and authority in a hierarchical structure. The graphical user interface, known for its main features as 'WIMP' (windows, icons, mouse, pull-down menus), emerged from the human augmentation community, with its roots in behaviourist perceptual psychology and military command and control systems. Microsoft PowerPoint reflects the adaptation of computer assisted design to the needs of management systems and the corporate boardroom. The communications community provides the networking. And so on (Mahoney 2005).

Software, intangible and elusive, plays a pivotal role in this framing of computation; it mediates between the universal schema of computation on the one hand, and the specific desires and designs of a community on the other. It is through programming that the universal symbol manipulating machine is transformed into a specific kind of machine such as a smart phone, financial database, or *Super Mario Bros.*[5] Mahoney's claim is that in order to unravel

[4]Like the Western playing cards brought to Japan in the 16th c. that hanafuda cards are derived from, video games are largely considered to be an American invention, but Nintendo's idiosyncratic history as a game and toy maker does a lot to explain its particular synthesis of computers and commercial entertainment.

[5]Of course, the accommodation of computers to human needs proceeds through many avenues. The difference between a Tamagotchi and an iPad reflect differences in hardware design, industrial design, fabrication, marketing, etc…, and not just software. Mahoney's emphasis on software is an attempt to decenter the material computer, which up to this point has received the lion's share of historical attention, and to help us see ineffable software for what it is: essential to the transmutation of an abstract symbol manipulating machine into a specific kind of machine manipulating certain symbols that stand for specific things. Software convolves a community of computational practice with the abstract hardware of computation.

the social forces at play in computer history, we most attend closely to software. He writes that "[t]he histories of computing will involve many aspects, but primarily they will be histories of software" (Mahoney 2005).

The shaping of computers by communities of use is seldom straightforward. Software is the result of negotiations with the computer.[6] "Programming is where aspiration meets reality. The enduring experience of the communities of computing has been the huge gap between what we can imagine computers doing and what we can actually make them do." Software design is about how "we have put the world into computers … translating a portion of the world into terms a computer can 'understand' ". This translational work into what Mahoney calls an "operative representation" is, of course, highly dependent upon the community which performs it. The word *design* neatly captures both the constrained, intentional, and constructed qualities of this translation. With *operative*, Mahoney refers to the fact that programs have behavior as well as structure (Mahoney 2005).

As should be very clear by now, software design is highly reflective of the communities who make it. Furthermore, "the computer as tool and medium is not neutral," writes Mahoney. Different communities make different kinds of computing machines by putting the world into these machines in specific ways. Software reflects the commitments of its makers. "Tools embody history," writes Mahoney. When using software we "work with other people's histories,"

[6] Mahoney sees software design as the translation of the world into computers, and notes that this translation is a negotiation, for example between the desires of a community and technical feasibility. Errors in design manifest as bugs. Bruno Latour, in his study of the failed personal transit system Aramis, offers a vision of technological design as reconciling powerful social and technical forces—an application of Actor Network Theory (ANT). Rather than see material and social concerns as separate, Latour sees them as intimately bound up. Designers are responsible for negotiating between the demands of technical and social forces, and the artifacts they design bear the mark of the powerful energies exerted upon them. Technological systems fail to come into existence when competing demands are not reconciled (Latour 1996). Steve Woolgar, in a case study of usability trials, reveals computer designers trying to reconcile their design with the users they hope will use the machine. The designers, in a pragmatic endeavor that is both comedic and paradoxical, undertake to learn about their hypothetical end users while simultaneously struggling to identify who exactly these users are; the machine and its users must be simultaneously defined with respect to one another (Woolgar 1991).

a state of affairs that "calls for critical awareness" (Mahoney 2005).

Software is a Medium

Not only does software enact and reflect traces of the invisible histories, agendas, and world views of its makers, but it transforms practice. *PowerPoint* entrains us into particular ways of doing, seeing, and thinking. As a medium, software transforms how we think, work, play, make art, and communicate (Kay 1972; Kay and Goldberg 1977; Kay 1984; Murray 2011). "We do not read about them [technology, e.g. software], we act with, in and through them. … They are artefacts of our culture, embodying both its explicit and its tacit knowledge" (Mahoney 2005).

Mediums shape us. Facebook, the operating system of a smartphone, or an inventory management system all affect the way people work and think—very much in line with the intent of the software makers. Software, whether made for practical or playful purpose, has consequences. "[W]e have put the world into computers" and the consequences are that software is "model" and "experience" and "medium of thought" and "environment within which people work and live." The creators of programs don't escape the transformative consequences of their designs. Articulating, reifying, and using these "worlds of software" has "(re)shaped" the practice of the very communities which designed them. Consider the number crunchers and military operators who commissioned computer systems to suit their needs, in turn transforming their own practices and histories (Mahoney 2005). As a medium and dynamic process, software transforms how we think, act, and feel.

Computers do not *understand* anything. The computer is simply a mechanical manipulator of symbols—it has no idea what these symbols mean. The operative representations made via computer programs rest upon the surface between computer and outside world, between symbols in the machine and referents outside of it. This mapping is highly elastic. The symbols which pass into and out of the machine might refer to numbers, the actuators of an ATM or self-driving car, the location of a mouse on our desk, lattices of colors we interpret as photographs, or strings of text we might interpret as a story, email, or nonsense. "It is a matter," as Mahoney writes, "of representations in and representations out" (Mahoney 2005). In short, these symbols passing into, through, and out of the machine stand for nothing aside from what we map them to. It is a highly flexible mapping. Signification might be interpreted by people, as in our interpretations of digital imagery, or take place mechanically, as when the computer is coupled to actuators or inputs.

Reading Software as an Artifact

To gain "critical understanding of how various communities of computing have put their portion of the world into software, we must uncover the operative representations they have designed and constructed." But how is one to uncover these representations? Where are they to be found? How does one actually interpret and study software representation? Mahoney proposes that "[l]ike the historians of computing, digital scholars must learn to read software to elicit the history and practice that it embodies." But how does one read software? Mahoney offers an analogy from computer hardware, quoting from Tracy Kidder's *The Soul of a New Machine*, in which computer designer Tom West interprets the design of a competitor's machine:

> Looking into the VAX, West had imagined he saw a diagram of DEC's corporate organization. He felt that the VAX was too complicated. He did not like, for instance, the system by which various parts of the machine communicated with each other; for his taste, there was too much protocol involved. He decided that VAX embodied flaws in DEC's corporate organization. The machine expressed that phenomenally successful

> company's cautious, bureaucratic style. Was this true? West said it didn't matter, it was a useful theory (Kidder 1982).

But while the VAX hardware can be physically taken apart, software is trickier—"we can't 'pull the boards' ". Software has no material form—there are only static programs and their dynamic behaviors. Complicating matters, Mahoney points out that the "dynamic process" is "the primary source" for analysis. Not only should we somehow study software—intangible and cryptic—in its static form, but we must attend to it as a dynamic process, as a medium that transforms experience, thought, and action (Mahoney 2005).

Summary

Even if Mahoney does not offer specific methods for studying software, he has framed the problem. Software, as we have seen, manifests the negotiations by which communities of practice turn the universal symbol processing computer into a machine tailored for their specific purposes. Through software design, abstract computers are adapted towards specific needs, and a world is translated into an operative software representation. Examining program designs reveals the compromises made in search of desired operational representations, reflecting the aspirations, practices, and beliefs of their creators. Software, therefore, calls for critical interpretation. We should examine it as a static artifact (e.g. source code) as well as a dynamic medium that reshapes our lived experience.

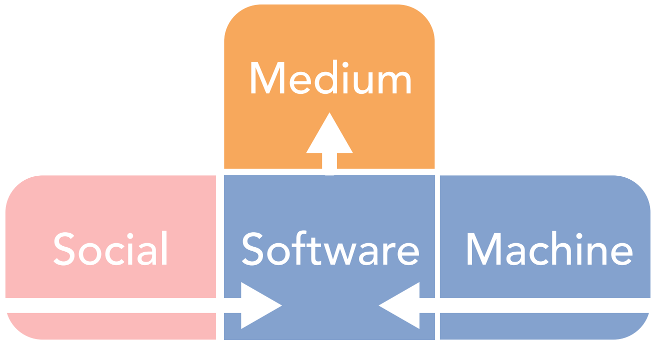


Figure 2.2: Software.

From Mahoney, we gain not just a motivation for studying software, but an orientation for doing so. This orientation has four key parts:

- **Machine**. Computers are protean machines.
- **Social**. Software is how the protean machine is given specific form by communities of practice.
- **Software**. Software is the mediating artifact through which social forces shape the machine.
- **Medium**. Software reshapes lived experience.

We are left with a very practical problem. Mahoney doesn't tell us how to analyze software, just that we should, and which facets we should be sensitive to. So what methods should we use to analyze the interlocking facets of software design, social forces, and media? If we can't "pull the boards" of software, as one might with a physical computer, then what can we look at? How can we read the artifacts? How should we interpret social forces? And how to make sense of software's qualities as a medium?

I'll address these questions in turn, but I first want to focus on just one facet, software as a medium. To understand software as a dynamic medium—as reshaping lived experience—we turn to the literature on situated cognition and scaffolding, which will not only ground our analysis of software as a medium, but will help us connect software to play.

Scaffolds

There are many ways to talk about something as a medium. One can discuss media abstractly and evocatively, like Marshall McLuhan (McLuhan 1964). Or one can take a more meticulous and cognitively grounded route. To do the latter, I want to introduce some very concrete language and ideas for thinking about media grounded in the tradition of scaffolding and distributed cognition.

It is now commonly accepted that cognition is situated and distributed in bodies, culture, social organizations, and artifacts (e.g. Latour 1986; Suchman 1987; Agre 1997; Clark 1998; Johnson 2007). When cooking breakfast you do not have to remember all the details—whether the eggs have been fried or the table has been set, as the world is its own best model. You simply look at the world to understand where in the process you are (Agre 1997). When solving jigsaw puzzles or playing Tetris we rotate the pieces so that we see, in the world, possible solutions (Kirsh and Maglio 1992).[7] Long division relies upon socially transmitted cultural knowledge (an algorithm), physical materials (paper, pencil), and our bodies (hands and eyes as well as the brain). The cognitive anthropologist Edwin Hutchins documents the navigational system of a navy ship in great detail, showing that the cognitive processes of navigation emerge from activities distributed across multiple artifacts (e.g. maps, rulers, view-

[7] These rotations are examples of *epistemic actions*, knowledge operations we perform with our hands, eyes, and the world (Kirsh and Maglio 1992).

ing instruments), cultural practices (procedures and scripts followed by people), and people performing specialized roles (Hutchins 1995).[8] We lean heavily on the environment; thinking is distributed in the eye, hand, brain, and world. Cognition does not take place primarily in the mind—through manipulation of mental representations of the world, as was once commonly believed to be the case—but through our artifacts, bodily practices, and social interaction.[9]

The term *scaffolding* is used to describe how new capabilities bootstrap through cognition distributed in the environment. A scaffold is an external structure that elevates our abilities, expanding and transforming our cognitive perspective and abilities. The idea descends from Vygotsky's notion of the Zone of Proximal Development (ZPD), a sphere of activity beyond what a person is ordinarily capable, but made possible when "interacting with people in his

[8]Hutchins, a cognitive anthropologist by training, examines how navigational work unfolds on a Navy vessel. This is cognition "in the wild" as opposed to *in the lab,* where cognition is typically studied. Laboratory experiments contrived by cognitive scientists a priori eliminate the external cognitive apparatuses of culture, people, and artifacts. Hutchins attributes the erroneous location of cognition as centered inside our skulls to our externalized cognitive apparatuses; an abacus or pencil and paper, by offering observable manifestations of thought, offers a working prototype of cognition as mechanized symbol manipulation. Cognitive scientists, Hutchins argues, misinterpreted our external instruments of thought, misattributing their symbolic machinations to our own minds, thus effacing the externalized and distributed mechanics of cognition. But brains are not abaci, and arithmetic is done not simply with our brains, but with brains and bodies in conjunction with external tools and culturally prescribed procedures (Hutchins 1995).

[9]Building on foundational critiques by Hubert Dreyfus, Edwin Hutchins, and Lucy Suchman, Phil Agre gives a cogent critique of the preeminent symbolic mode of artificial intelligence (sometimes referred to as "Good Old Fashioned Artificial Intelligence," or GOFAI) that employed systems such as theorem provers and planning systems in an effort to build artificially intelligent agents. This symbolic processing formulation of intelligence rests upon a notion of thought as happening inside the brain, where symbols inside the mind represent things outside of the mind. One can caricature this model of human intelligence, which extends Cartesian dualism into computer artifacts, as the brain in a jar model. Thinking inside this framework, AI researchers have spent decades building systems such as robots which perceived the world, constructed internal symbolic mental models of it, and then acted in accordance with their internal symbolic models. Unable to keep pace with their projected delivery dates for artificially intelligent agents, funding for AI research (which was primarily doing GOFAI) eventually dried up. The symbolic AI project had hit a technical dead end which Agre argues stemmed directly from its stubborn and unquestioned enthusiasm for a limited model of intelligence as symbolic reasoning unfolding inside brains. Agre makes a strong case that technical hangups in AI agent designs are directly traceable to this symbolic paradigm. Agre argues for an alternate *interactionist* conception of intelligence, in which thinking is seen not as symbol manipulation inside of the brain, but rather as something that emerges from the interaction between agents, their bodies, and the environment. Insects, for instance, seem to get along just fine without symbolic reasoning (Dreyfus 1972; Suchman 1987; Brooks 1991; Hutchins 1995; Agre 1997).

environment and in cooperation with his peers."[10] Our environment reshapes us as we assimilate the terms of cognition set forth by the external scaffolding we interact with. Although Vygotsky saw the ZPD primarily as a social mechanism, the idea of scaffolding has since been generalized to include our material, cultural, and social environment. Through "subordination to rules," he argued that play was a potent ZPD, boosting a child's developmental level.[11] Not only is play a powerful scaffold (a spur to development), but play itself can be scaffolded through things like games and playmates—a point I will elaborate on shortly (Vygotsky 1978; see also Bruner 1986, and Clark 1998).

Scaffolds boost and transform thought, perception, and action. External instruments of cognition offer various kinds of affordances, rendering certain operations easy or hard to do. What such tools frequently do is reformulate difficult problems into straightforward perceptual ones. A map and compass, for instance, shift computational work to the hand and eye, rendering the result of certain cognitive operations visually obvious—transforming, for instance, the process of determining a ship's location into simply seeing where three lines meet on a map (Latour 1986; Hutchins 1995).[12] As a scaffold, software functions similarly. Spreadsheets and

---

[10]Vygotsky himself never used the term "scaffold." He defines the ZPD as

> *the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers* (Vygotsky 1978). [italics in original]

Development occurs as a child internalizes these new capabilities exercised in the ZPD (Vygotsky 1978; Bruner 1986).

[11]Vygotsky saw rules as underpinning both fantasy and overtly rule based play. (For a succinct description of what Vygotsky means by play see chapter 1.) Vygotsky wrote that

> … play creates a zone of proximal development of the child. In play a child always behaves beyond his average age, above his daily behavior; in play it is as though he were a head taller than himself. As in the focus of a magnifying glass, play contains all developmental tendencies in a condensed form and is itself a major source of development (Vygotsky 1978).

[12]In this argument Hutchins builds, in part, on the work of Bruno Latour, who meditates on the power of paper representations as affording a bilateral translation between marks on paper and the world, the reliable

databases are externalized apparatuses that assist in thinking and doing—in particular ways, for particular ends (Mahoney 2005).[13] Reshaping thought, action, and perception in this way also has experiential and aesthetic repercussions. Games such as Poker and Go, when considered as distributed cognitive apparatuses, are scaffolds that foster feelings of excitement, fear, greed, regret, delight, and so on (Lantz 2011).[14] My use of the term *scaffold* is somewhat idiosyncratic, and follows in the footsteps of Andy Clark (Clark 1998). I use the term to refer to anything (noun) in the environment that boosts and transforms activity, as well as the process (verb) of boosting and transformation, whether or not the scaffold is temporary or permanent.[15]

Scaffolds have vital social effects. The externalization of cognition facilitates the thought and action of not just individuals, but groups. One effect of externalization is the formation of a common ground for joint activity. People can huddle around a map, point at it, draw on it together, and learn from one another (Hutchins 1995). Direct manipulation interfaces are particularly good at grounding interaction between people and software (Brennan 1991).[16]

---

transmission and reproduction of complex ideas, and the recombination of these flattened forms (Latour 1986).

[13]As Mahoney puts it, when communities of computing go about "translating a portion of the world into terms a computer can ''understand''", they put their model of the world into the computer. Software reifies a way of thinking about the world. Like an astrolabe, map, abacus, or pocket calculator, software is very often an externalized cognitive apparatus built to extend the practices of a particular community into a computerized mode. The Cold War era SAGE defense system was a computerized apparatus to aid in the interception of hostile intrusions into North American airspace. SABRE, built by IBM for American Airlines in the 1950's and 1960's, facilitated the organization and management of airline bookings.

[14]Frank Lantz argues that part of the aesthetic effect of games (think of Go and Poker) is precisely in reshaping our mental processes and enabling us to observe ourselves in, for example, the throes of greed or a fight over life and death. Lantz compares games to machines we inject into our brains that simultaneously reconfigure our operation and enable, like an injected dye, us to observe ourselves in this altered state (Lantz 2011). Lantz's analysis suggests an aesthetics of cognitive scaffolding, and identifies games—certain kinds anyway—as instruments of externalization and reflection. According to Lantz, if humans are the animals of instrumental action, then games are the art form of instrumentality (Lantz 2014b).

[15]The term scaffolding comes from the education literature, which emphasizes this boost as temporary—a spur to development (Wood et al. 1976; Brown et al. 1989). Scaffolds are temporary structures that are supposed to fade, which is not how I use the term.

[16]Summarizing the work of Susan Brennan, who applies the conversationally derived notion of common ground to interface design, Laurel writes that "common ground is a jointly inhabited "space" where meaning takes shape through the collaboration and successive approximations of the participants" (Laurel 1993, p.4). Laurel quotes

Common ground is both material (maps, gaze, hands, icons, mouse cursor) and immaterial (rules, attention, language), as well as explicit and implicit (Brennan 1991; Clark and Brennan 1991; Laurel 1993). The tokens, spaces, and rules of a board game scaffold the shared understanding of the game's performance. Games and rituals are scaffolds of shared grounding and coordinated activity (Murray 2006).[17] Language is a scaffold that extends our basic

extensively from the work of Clark and Brennan (Clark and Brennan 1991), which I have reproduced, in part, below:

> It takes two people working together to play a duet, shake hands, play chess, waltz, teach, or make love. To succeed, the two of them have to coordinate both the content and process of what they are doing. … They cannot even begin to coordinate on content without assuming a vast amount of shared information or common ground—that is, mutual knowledge, mutual beliefs, and mutual assumptions [Clark and Carlson, 1982; Clark and Marshall, 1981; Lewis, 1969; Schelling, 1960]. And to coordinate on process, they need to update, or revise, their common ground moment by moment. All collective actions are built on common ground and its accumulation.

Human computer interaction is predicated upon common ground shared by a running program and its user, which is precisely what things like icons, menus, pointing devices, progress bars, and so on do: they provide a shared stage of referents for coordinating activity. Brennan's analysis of direct manipulation interfaces complements Lucy Suchman's critique of systems which frustrate our expectations of multi-agent interaction (Suchman 1987). Brennan writes:

> Implicitly or explicitly, people seek and provide evidence of their understanding, step by step; this is the process of *grounding* (Clark and Brennan, 1990). Grounding includes back-channels, monitoring and displaying attentiveness (as indicated by eye contact), and all of the other things people do to check on one another's understanding. Whether two people are having an argument, moving a piano, or dancing a tango, each monitors what the other is doing. They use all the evidence they can get, including linguistic evidence, visual evidence, and tactile feedback. Grounding is an important part of any coordinated activity between two partners, however different they may be from one another.

Direct manipulation interfaces handle the grounding problem well, at least for simple tasks that can be concretely represented; a user can get back continual evidence about how the system has "understood" an action (Brennan 1991).

[17] Janet Murray observes that games, which by all accounts constitute an ancient part of culture, are spheres of activity that afford and heighten the fundamental human faculties of sharing attention, coordinating intention, and making meaning with symbols. The ideas of shared intentionality and attention offers a profound set of tools for understanding social play, games, and interactive design, which Janet Murray has demonstrated by applying the ideas of Tomasello and Merlin Donald to play, games, and narrative (Murray 2006; Murray 2011).

Murray makes the claim that games might have functioned as evolutionary ratchets that scaffolded participation in shared attentional scenes, culturally defined symbolic realities, and ritualized action (Murray 2006), an argument whose outline mirrors that of Johan Huizinga, who argues that play and competition underpin human social reality and the structures of civilization (Huizinga 1955). In this argument, Murray builds upon the research of Michael Tomasello, who studies how it is that infants learn to communicate through pointing, perceiving and directing the attention of others, and undertaking joint activities, as well as Merlin Donald, who argues that such attentional management underpins the structures of our communication, thought, and media (e.g. Tomasello et al. 2005; Tomasello 2010; Donald 2001).

intersubjective capabilities, elevating not only individual thought, but group cognition and communication (Clark 1998; Donald 2001). Language, social norms, and social institutions are all underwritten by humans' distinguishing capacities for focusing shared attention (e.g. language and pointing) and synchronizing intentionality (Tomasello 2005; Tomasello 2010).[18] Scaffolds help us to express, see, and synchronize our thoughts, intentions, actions, and perceptions.

Play is often scaffolded. Puzzles and games are prototypical[19] examples of distributed cognition, externalized stages upon which one or more players see, remember, synchronize, enact, and transform clear and stylized dramas of thought and feeling.[20] The cognitive, physical, and social stages of games are scaffolded by embodied tokens and prescribed rules of play (Murray 2006). Seen from the perspective of scaffolds, our playthings—from symbols and rules to materials and people—are powerful instruments that guide, shape, and focus play, whether such play is solitary or social.[21] And this flow is bidirectional: people fashion

[18] According to Michael Tomasello, as humans evolved from apes, we acquired increasing intersubjective capacities. This enabled us to synchronize both intentionality and frames of reference, giving rise to intersubjective coordinative capacity (Tomasello 2010). Tomasello writes that "the crucial difference between human cognition and that of other species is the ability to participate with others in collaborative activities with shared goals and intentions: shared intentionality," resulting in "species-unique forms of cultural cognition and evolution, enabling everything from the creation and use of linguistic symbols to the construction of social norms and individual beliefs to the establishment of social institutions" (Tomasello 2005).

[19] External cognitive instruments shed light on the internal workings of our minds (Hutchins 1995), a perceptual quality that takes on aesthetic dimensions in the case of games, which enable us to reflect upon and wallow in our thoughts, desires, and feelings (Lantz 2011). It is perhaps because of this quality that cognitive psychologists have focused attention on games such as Tetris, Scrabble, and Towers of Hanoi (e.g. Zhang 1991; Zhang and Norman 1994; Kirsh and Maglio 1992; Kirsh and Maglio 1994; Kirsh 1995; Maglio and Kirsh 1996; Maglio et al. 1999). This also explains, in large part, LeBlanc's equation of games with computers—both are externalized cognitive scaffolds (LeBlanc 2004), and Murray's identification of games, alongside stories, as ancient cultural scaffolds of social cognition (Murray 2006).

[20] These externalized cognitive stages are not just about synchronization—they are stages upon which dramas of dissembling and interpretation of intent and knowledge might unfold. Poker, for example, provides a shared embodied world of information—cards, chips, etc…—but the tokens of this world are distributed asymmetrically, setting the stage for us to synchronize, lie, and read the intentionality and information of others. Poker is a practice of externalized, distributed cognition whose stylized scaffolds facilitate not just synchronization, but atop this synchronization, an intersubjective drama of competition, calculation, deceit, and interpretation.

[21] Scaffolds, such as the metacommunicative signals that delimit play, and the materials and procedures of games, help to synchronize our playful intentions, attitudes, attentions, and activities. We can think of toys, playgrounds, people, rituals, language, cultural materials, scripts, and materials as scaffolds for play. They invite

scaffolds to suit and support their playful purposes. Scaffolds are the things that we, as play designers, can design.

Make believe play is scaffolded by older playmates, such as siblings or parents, who inspire and guide. Play is *inspired* by scaffolding that validates make believe as worthwhile. And scaffolding can sustain the interest, enthusiasm, and involvement of players who might otherwise be disoriented or lose interest. Play is *guided* by scaffolding that provides a role, sets the scene or plan of action, narrates and comments upon, directs the action, asks questions, and offers a menu of options to choose from. When scaffolded in these ways, make believe play is boosted. Players are able to participate in make believe play at a higher level of fluency, eventually becoming more sophisticated players (Garvey 1970/1990).[22]

---

and structure play, directing, guiding, and yielding to its transformations. Scaffolds, such as the rules and tokens of a game, coordinate the goings on and common grounding of play. And certain scaffolds, like a teddy bear, can foster a feeling of safety. Some materials, such as building blocks and masks, are more amenable to playful transformations than others.

[22]I summarize the effects Garvey describes as follows:

1. *Arousal*. Scaffolds maintain the interest, enthusiasm, and involvement of players who might otherwise be disoriented or lose interest.
2. *Boosted fluency*. While scaffolded, players engage in make believe at a higher level of fluency.
3. *Learning*. Make believe skills are cultivated, leading to skill improvement that persist beyond the scaffolding. It has a developmental effect on pretend play.
4. *Validation*. Players learn that make believe play is a worthwhile activity.

Garvey describes a variety of techniques that have the effect of guiding and inspiring make believe. These can happen both in character (in pretend) and out of character (about pretend). Guidance occurs through declarations about:

- *Role*. Provide a role. Address the player in their newly assigned pretend role.
- *Scene*. Announce or provide a theme or scene.
- *Plan*. Announce a plan of action.
- *Object Transformation*. Declare that an object is now something else.

By directing player actions:

- *Direct*. Tell them what to do next.
- *Question*. Ask leading questions that prompt the player for a response, e.g. “Is the dolly tired?” (Garvey 1970/1990).

Scaffolding is a powerful concept for thinking about the effects of software as a medium. As a form of distributed cognition, software is a scaffold that mediates thought, action, feeling, and social activity. When we convene with software, and incorporate it into our extended cognitive apparatus, the way we think and feel transforms, individually and socially.[23] Software can scaffold socially distributed cognition, facilitating the formation of common ground, focusing shared attention, and synchronization of intent. And since play can be scaffolded, we can conceive of software itself as a scaffold, or medium, for play.

Framework for Studying Software

Mahoney's framing of suggests three primary perspectives from which to think about software: as a social process, a medium, and as an artifact. In what follows, we will adopt each of these perspectives, review the literature from that vantage, and discuss particular analytic techniques.

- *Choices*. Describe actions they might take.

Or describing what is happening:

- *Comment*. e.g. "Oh, the bunny is scared" (Garvey 1970/1990).
- *Narrate*. Describe in words what the player's actions depict.

Also, Murray's notion of *scripting the interactor* can also be seen in this light (Murray 1997).

[23] Just as using a pocket calculator transforms practices that involve calculation, the SABRE flight booking system mutated processes involving flight reservation at American Airlines, and the digital spreadsheet ushered in new practices for accountants and others that used spreadsheets. These transformations ripple beyond the immediate cognitive domain of an individual, permeating the diffuse cognitive activity of organizations, often reorganizing the division of labor, and transforming social relations. Instantly updating electronic spreadsheet, for instance, meant that trying out *what if* scenarios became easy, and accounting as a practice required less human computing labor. Software reorganizes cognition, both individual and social, yielding new ways of doing and thinking.

Software as Social Construction

Figure 2.3: Software is shaped by social forces.

Software is made by people who shape and are in turn shaped by the digital medium they make. Software design reflects and reifies the history and ideas of particular computing communities, who are then reshaped by their own software tools (Mahoney 2005). This is, after all, the point of software. It is made to transform practices: spreadsheets transformed accounting, SABRE changed how flight reservations were made and tracked, and SAGE was intended to augment Cold War defense.

To study software as arising from particular social forces, I attend to six infrastructures that constitute and perpetuate it. These categories are drawn from my history of system dynamics. Each is grounded in the existing literature.

- **Actors.** One can focus directly upon participants, recovering their motivations, influences, and struggles. Developers can be interviewed, residual documentation read (e.g. emails and reports), and ethnographic observations conducted in an attempt to uncover actors' agendas, beliefs, histories, and negotiations (e.g. Turkle 1984; Latour 1996; Edwards 1997; Hayles 1999; Edwards 2010; Schüll 2012).

- **Discursive Practices.** Examine the manual, box, marketing materials, books, design documents, papers, presentations, technical documentation, source code, user interface, packaging, etc… that developers use to explain, describe, frame, and perpetuate their software projects (Latour 1987; Hayles 1999; Zhu 2009). These can be seen as a "hermeneutic network" that circumscribes and shapes our reception and experience of software (Zhu 2009).[24] Discursive strategies tether software to the world outside the ma-

[24]Jichen Zhu offers the notion of a "hermeneutic network" that circumscribes the narrative and interpretive strategies surrounding artificial intelligence systems, but which we can see at work in software more broadly. Her idea explores how "system intentionality is narrated and interpreted by its human creators and users in their socio-cultural settings." Zhu draws attention to the "discursive strategies" employed by system authors, whether "embedded in their source code" or "technical literature" (Zhu 2009, p. xii). Diverging from the argument "that intentional vocabulary is an epidemic whose usage should be minimized" (Zhu 2009, p. 90), she argues instead that these discursive strategies are "a constitutive component of AI" (Zhu 2009, p. xii). This conception closely mirrors that of Michael Mateas, who sees discursive strategies as "inextricably part" of the AI enterprise (Mateas 2002, p. 132).

Zhu's recasting of these narrative tactics as "constitutive" rather than "wishful" hinges upon philosopher Daniel Dennett's idea of the intentional stance, which formulates intentionality as a stance one takes towards something, rather than as an intrinsic quality of that thing. Agent-hood, in other words, is in the eye of the beholder—a formulation with echoes of the Turing Test (Turing 1950). Zhu observes that we can, and often do, take the intentional stance towards things such as Tamagotchi toys, Roomba vacuum cleaners, and Joseph Weizenbaum's Rogerian therapist program ELIZA. Zhu calls attention to the hermeneutics of software, specifically AI systems. This hermeneutic framework, and its focus on the rhetorical strategies employed by systems and their authors, can also be used to think about the discursive practices surrounding simulations.

Katherine Hayles, in a chapter entitled "Narratives of Artificial Life," observes that the narratives attached to AL systems by their authors play a crucial role in linking these systems to biological phenomena such as reproduction, mutation, cells, death, etc…. For example, while a simulation might output only bar graphs, the "life force of the "creatures" " acquires meaning through the work of "human interpreters" that "scrutinize the binary codes that constitute the "creatures" ". Hayes pinpoints the narratives attached to these systems as establishing correspondences between software worlds and that the worlds of biology, and contrasts this with the desires of AL practitioners for aliveness to be found an intrinsic property of their simulations (Hayles 1999, pp. 230–231). This line of reasoning echoes Bruno Latour's description of the constitutive function of scientific texts (Latour 1987).

The hermeneutic network includes the running program—graphics and feedback—all the ways in which a software system describes its own machinations. Discursive strategies can be seen as a technique of artifice, and attending to them reveals how software is co-constituted by the hermeneutic network that floats around it. Considered as make believe, discursive practices function very much like the play scaffolding of more experienced playmates, inspiring and guiding the pretend belief and activity of less experienced players (Garvey 1970/1990).

While this kind of thinking has traditionally been focused on intensionality and mnemonics in AI and AL, considering software systems as embedded in hermeneutic networks is a mode of analysis applicable beyond these domains. I claim, and will later demonstrate, that discursive strategies are constitutive of simulations in general. For example, Ian Bogost's analysis of "exergaming"—exercise games—analyzes how the rhetoric of exercise has been deployed to recruit players into computer mediated physical activity (Bogost 2005). But this rhetoric does more than recruit players, it also frames their experience and interpretation of play. In addition to persuading players to get up and dance or run in place, the exercise rhetoric functions as a software narrative that co-constitutes chine, anchoring it in the world of human meaning and activity (McDermott 1976; Agre 1997; Hayles 1999; Mateas 2002). For example, the originator of *SimCity*, Will Wright, learned about Jay Forrester's city simulation model and techniques through the book *Urban Dynamics,* which brings these systems to life in readers' imaginations (chapter 5).

- **Social Practices.** Examine social organizations and activities, for example: journals, conferences, contests, and schools (Edwards 1997; Hayles 1999; Edwards 2010). Forrester's simulation methodology, for example, was taught and cultivated within the Sloan School of Business to masters students (chapter 3), and Conway's *Game of Life* gained momentum from a contest publicized in *Scientific American* which gave rise, in turn, to the hobbyist driven *Lifeline* newsletter (chapter 4).

- **Craft.** Look at how the software artifact is used. Forrester, for example, taught a particular style of simulation building, teaching his students to rely upon intuition and art rather than data. He succeeded in spreading system dynamics as a way of seeing the world, along with a particular style of model building (chapter 3). Game playing also illustrates the specificity of craft. Some console games originally published in the 1990's are now the focus of vibrant communities who modify and reprogram them to new ends, exemplifying a shift in the craft of engagement (Moss 2015).

- **Schema.** Attend to schemas. As we saw in our discussion of scaffolds, software often introduces particular modes of abstraction and representation. System dynamics, for instance, offers a schema for thinking about and seeing the world in terms of levels and rates. Forrester even offers a visual notation for this schema, reinforcing this particular

the make believe realities of running and dancing.

mode of schematic abstraction (chapter 3).

- **Software Artifact.** Look at the artifact itself. Software functions as a reified representation of a schema, an "evocative" (Turkle 1984) prototype that spreads ideas, and as a practice transforming artifact. GitHub and Facebook support and enforce certain kinds of social interactions. Compilers, software that converts program descriptions from one form to another, are perhaps the ultimate example of software tools reifying schemas. Compilers take input in a certain form. To simulate a city with system dynamics, you feed the DYNAMO compiler a description of a city in a particular format; deviations result in errors. DYNAMO enforces and enacts a schema (chapter 3). Conway's *Game of Life* is an evocative simulation design that operated like a vector, spreading the ideas and practice of cellular automata far and wide (chapter 4). The beliefs of people working in fields like artificial intelligence[25] and interface design[26] are reflected in, and can

[25]Phil Agre, an artificial intelligence researcher, formulated a method called Critical Technical Practice in which practitioners identify the unarticulated ideas embedded in their technical practice, critique these commitments, and formulate new technical designs. Agre's work is an excellent case study in unearthing the intellectual commitments embedded in a particular technical practice. Closely examining GOFAI software designs revealed the unquestioned commitments of their makers. Their software designs could be used to reflect upon the computing community of symbolic AI, critique assumptions, and articulate new design directions (Agre 1997).

Agre, not surprisingly, is not alone in his critique of GOFAI. The critique of symbolic AI's underlying philosophy, an extension of an outdated conception of mind, was preceded by Hubert Dreyfus's *What Computers Can't Do: A Critique of Artificial Reason* (Dreyfus 1972). Non-representational AI systems have been proposed and used by Rodney Brooks (Brooks 1986; Brooks 1991). Agre's approach, however, is unique in joining a technical interpretation of software, a discussion of the intellectual commitments that underly and sustain a computing community, and practical design considerations.

[26]Lucy Suchman carries the critique of intelligence as symbolic planning into the domain of interface design, and by different means (Suchman 1987). Embedded in the Xerox PARC research lab in the mid-1980's, Suchman conducted ethnographic research on the design of a photocopy machine with an automatic help system. The machines used what we might think of as a "wizard" style interface; the machine would walk users through tasks such as fixing machine malfunctions. The design of this software, however, enacted a conception of users as agents working through mental plans, and a very limited sense of the richness of multi-agent interaction—in this case between the user and machine-as-agent. The machine would deduce which step of what plan the user was on, and then try and guide the user through the remaining steps. Suchman probes this machine's design failures using ethnographic methods. Taking issue with the model of intelligence as symbolic planning, Suchman argues that people do not formulate and then execute hierarchical plans. Intelligent action, especially in interaction between agents, is far more fluid. Responses unfold in response to shifting circumstances, which themselves are never neatly delineated. Take, for example, a conversation. It is never clear a priori whether a

be recovered from, the software they craft (Dreyfus 1972; Suchman 1987; Agre 1997; Wardrip-Fruin 2009).

As should be clear, these infrastructures mutually reinforce one another. Within a class (social practice) a teacher might instruct students in the proper use (craft) of a simulation tool (software artifact) he has made (agent), encouraging them to see the world in a particular way (schema). The class might use lectures, books, tests, and sample programs (discursive practices). They are also enmeshed in one another. Schemas are a particular way of seeing the world, and while they can exist independently of software artifacts and discursive practices, they are often expressed in these materialities.

---

gap in speech signals a pause in speech or the end of someone's speaking turn. Rather, the meaning of the gap is jointly determined by the participants of the conversation. Both Agre and Suchman unearth and critique symbolic planning models of human intelligence, but while Agre takes up the design of AI programs, Suchman engages user interface design.

Although Suchman doesn't address the graphical user interfaces that began to disseminate widely at the time of her work, such interfaces reflect a contrasting view of human intelligence and activity. The design of the Macintosh computer, through which our contemporary models of graphical user interfaces gained a popular audience, can be seen as enacting a model of human intelligence as situated—or interactionist, to use Agre's preferred term. Macintosh GUI design conventions favor amodal activity, where the user is free to take actions whenever they wish, rather than progressing through preordained activity sequences ("Apple Human Interface Guidelines" 1987). This design paradigm does not conceive of users as working through hierarchical plans. Furthermore, interaction between user and machine is grounded in a shared set of visually represented referents—icons, windows, buttons, etc… (Brennan 1991). Rather than construing thought as happening inside one's head, GUIs model and afford thinking as actively externalized in the world, situated, operating with and through the environment, and drawing upon our embodied visual and kinesthetic capacities (Kay 1987).

Software as Medium

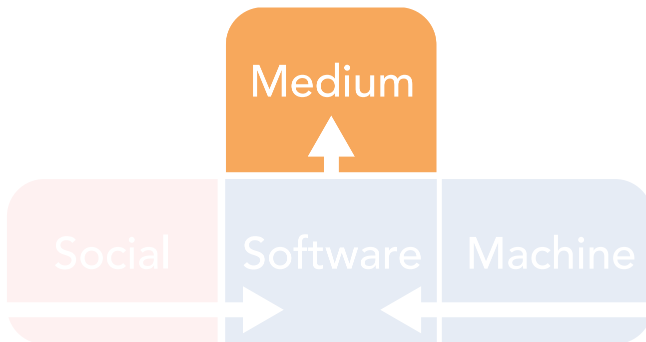


Figure 2.4: Software as a medium.

Examine Experience

Software shapes our lived and aesthetic experience.[27] There are two broad strategies for studying software's effects on experience: empirical and reflective. Empiricists do market analysis, ethnographic research, formal and informal user studies, and so on (e.g. Turkle 1984; Bartle 1996; Lazzaro 2004; Schüll 2012).[28] Reflection is a more humanistic mode of analysis that relies upon a sensitivity to one's own experience (e.g. Sudnow 1983; Gingold 2003;

[27] Just as video games call attention to the aesthetics of games, they also draw our attention to the aesthetics of software, and how formal rules—whether game rules or software code—engender particular kinds of aesthetic experience. Frank Lantz argues that our understanding of games has blossomed in the 20th century, a transformation made possible by video games, a new manifestation of games played with computers and constructed with software. Video games call our attention to games as objects of design rather than folk culture, and have ushered in an awareness of games as an aesthetic form. They call our attention to the aesthetic qualities of games that have been there all along (Lantz 2014a). Whether considering the game Go's elegant minimalist design, or the experience of grappling with its florid complexity, video games call attention to the aesthetic qualities of formal procedures (e.g. Sudnow 1983). Computer games, and by extension software, is simultaneously designed, aesthetic, and emerging from rules and code. Computer games constitute an opening, or a lens, for perceiving the larger phenomena of software as aesthetic form and experience.

[28] Observation can be formal or informal. Lazzaro's coded observations of player facial expressions illustrates a more formal approach (Lazzaro 2004), while informally noting player responses typifies the play testing conducted by practicing designers. Ethnographic approaches emphasize observation of and interviews with people about their experiences, an approach exemplified by Sherry Turkle's masterful study of the interpretation and experience of computing and Natasha Schüll's study of video poker players (Turkle 1984; Schüll 2012).

Hunicke, LeBlanc, and Zubek 2004; LeBlanc 2004; Swink 2009; Lantz 2011; Lantz 2014b).[29] Most of this work, perhaps unsurprisingly, has been focused on games. Analysts connect game design choices to aesthetic effects. Seeking a more fine grained way of thinking about game experience, the term *fun*—a clumsy term often applied to games—is replaced with more finely grained taxonomies of aesthetic experience.[30] In this thesis, my primary strategy is to use reflection, and to employ language that is attentive to the particulars of experience. At times, however, I do rely upon ethnographic study and accounts of others' experiences.

One idea I invoke frequently is that of the "evocative object." I use the term *evocative* in the

[29]In my past work I have, following the elliptical and explicit suggestions of certain designers, used Japanese gardens, postmodern literary theory, and other materials to interpret the aesthetic qualities of certain games (Gingold 2003). Frank Lantz uses reflection to evocatively consider the aesthetic experiences of Go and Poker (Lantz 2011), an approach reminiscent of *Pilgrim in the Microworld*, a deeply embodied first person account of playing Breakout written by David Sudnow, a pianist and phenomenologist (Sudnow 1983). Steve Swink's *Game Feel* is a study of how the phantom tactile touch and feel of games is created (Swink 2009).

[30]An excellent example of how video games foreground the aesthetics of games and software is the MDA framework. Mechanics, Dynamics, and Aesthetics (MDA) is a framework for thinking about the articulation between the fixed rules of a game, the dynamics processes they give rise to, and the resulting experiential aesthetics (Hunicke, LeBlanc, and Zubek 2004; LeBlanc 2004). MDA's primary utility is as a lens for identifying and disentangling three distinct but causally related sets of phenomena. With MDA, one can speak of the fixed rules of a game (mechanics), how those rules establish particular dynamic systems and behaviors (dynamics), and how those dynamic systems in turn relate to the feeling of playing particular games (aesthetics). This framework neatly applies to software of any kind, which isn't surprising given that deep within MDA is a conception of games and computers as isomorphic. Both the rules of a game and software code define fixed mechanics; tumbling forth from mechanics are the dynamics of game sessions or computational processes; aesthetics are the instrumental function of the system, which in the case of games is "fun". LeBlanc stakes the maximalist claim that games are computers, and computers are games, arguing that a chess game is a kind of computer, and an abacus can be a game (LeBlanc 2004). Whether one subscribes to this maximalist view or not, the MDA framework is useful for deconstructing the aesthetic operation of both software and games.

MDA enumerates eight types of aesthetics: sensation ("sense-pleasure"), fantasy ("make-believe"), narrative ("drama"), challenge ("obstacle course"), fellowship ("social framework"), discovery ("uncharted territory"), expression ("self-discovery"), and submission ("pastime"). While MDA's taxonomy does not claim to be exhaustive, a more fine grained consideration of aesthetics compels us to think in more careful terms than simply "fun." Unpacking the generic term "fun" is an avocation of video game developers, who have struggled with finding suitable languages and ideas for practicing game development as an intentional design craft (e.g. Costikyan 1994; Church 1999; Koster 2004). Having labored to articulate a vocabulary for understanding the aesthetic potentialities of games, many frameworks have been proposed and used. Nicole Lazzaro, in her observationally grounded research identifies four coordinates of fun: hard, easy, serious, and people (Lazzaro 2004). Designers at Maxis sometimes use an in house typology of players, a technique famously used by Richard Bartle in writing about online MUD game players (Bartle 1996). Csikszentmihalyi's idea of flow as a balance between challenge and skills has often been made use of in considering game experience, even though the idea is not specific to games or even play (Csikszentmihalyi 1975). And so on; you get the idea.

tradition of Sherry Turkle's "evocative object, an object that fascinates, disturbs equanimity, and precipitates thought." Such objects might provoke reflection on ourselves, our culture, or our politics (Turkle 1984). A more limited framing of such an idea sees software and games as persuasive or rhetorical.[31] Not everyone, however, is comfortable with making and interpreting games as vessels filled with messages, seeing them instead as social practices ripe with indeterminate meanings shaped by their players—a characterization more in line with Turkle's use of the term *evocative*.[32]

[31] If software changes the way we think and feel, transforms reality through artifice, and emerges from particular cultural standpoints, then it is natural to conceive of software as persuasive—recruiting people to a particular way of thinking. In this humanistic conception of software, in which software is a vessel filled with meaning, it harbors import on par with the other the components of our cultural landscape, such as books, films, and advertisements. Work in this field has been concerned with interpreting the meaning of games, as well as making works infused with meaning and thus demonstrating the power of this perspective by works that are political, critical, or journalistic (Frasca 2001a; Frasca 2001b; Frasca 2001c; Bogost 2005; Bogost 2007; Bogost 2008; Frasca 2007; Raz 2011; Treanor et al. 2012; Treanor et al. 2011; Treanor and Mateas 2011). Alan Kay, for example, critiques *SimCity* as opaquely reproducing an erroneous model of civic life (Kay 2007), and science fiction author Jerry Pournelle warned against in the authority given to simulations and their assumption (Pournelle 1990, quoted in Friedman 1999). My discussion of software as rhetoric relies upon a kind of conflation between games and software—a move made earlier in the discussion of aesthetics, and typically by practitioners of this approach.

Proponents of this perspective are particularly attuned to the qualities that make games different from other forms of media: that games are simulations, invite and require action on the part of players, and are constituted by rules and software. Scholars such as Frasca and Bogost argue that simulations are always subjective, and explore the rhetorical dimensions of digital media's procedural and participatory qualities (Frasca 2001a; Frasca 2001b; Frasca 2001c; Bogost 2007; Frasca 2007; Bogost 2009).

[32] Under the slogan "games are not media" Frank Lantz critiques the application of the "message model of meaning" to games, arguing that it betrays a naive understanding of games as consumable content and conduits of messages—a conception that is at odds with the ancient history of games and play, the complex social practices they participate in, and the many curious ways in which they are meaningful. Lantz attributes this mischaracterization of games to a conflation of games with their commodity manifestations, such as commercial board games and video game cartridges— cultural artifacts bought and consumed like magazines, books, advertisements, and music. Lantz compares games, instead, to "habits that we weave into our daily life, languages that we learn, clubs we join, practices that we become proficient at, a part of us not for 5 or 50 hours, but for our entire life." Lantz compares games instead to "meaning networks, little non-linear meaning-generating machines" (Lantz and Samyn 2009). Contemporary game studies contains an entire school of thought that rebels against the idea of games carrying determined meanings (e.g. Sicart 2011; Wilson 2011). Certainly Clifford Geertz would agree with such a broader conception of the meaningfulness of games (Geertz 1972).

Examine Representation

Mahoney writes that "various communities of computing have translated large portions of our world – our experience of it and our interaction with it – into computational models to be enacted on computers" (Mahoney 2005). Through software, computers create *enactive representations*.[33] Windows, icons, databases, spreadsheets, file systems, and city simulations are examples of dynamic enactive representations. Brenda Laurel describes these representations as constituting "imaginary worlds" within which multiple agents—human and computer—come together and act (Laurel 1991).[34] Our ability to participate in enactive representations is

[33]Ted Nelson introduces the term *virtuality* to describe the synthetic reality projected by software, from a word processor's scrolling text to the imaginary worlds of games (Nelson 1977). A related idea, *operational logics*, is introduced by Noah Wardrip-Fruin to describe patterns of such enactive representations that are apparent to end users, such as game players (Wardrip-Fruin 2006; Mateas and Wardrip-Fruin 2009).

[34]Brenda Laurel, a software designer and writer with a background in theatre, argues that interfaces and stage plays are both mimetic. "A mimesis is a particular kind of representation," it is "a made thing," a "closed system" ("it has finite potential and is limited in some way."), and is "internally consistent". One of Laurel's most intriguing arguments is about the desirability of a certain type of interactive mimesis which creates a feeling of immersion, or "first-personness." Enactment, in which users operate directly within and upon a "mimetic context," is preferable to a second or third person narrative mode, in which users indirectly converse with the machine about the objects of interest. Laurel argues these design principles are applicable to both fantastic (e.g. games) and serious (e.g. spreadsheets) interface designs (Laurel 1986).

(A similar sentiment is voiced by Ben Shneiderman, who identifies an overlap in approach between well designed user interfaces and computer games he refers to as "direct manipulation," and enumerates some shared design characteristics (Shneiderman 1983). Hutchins et al. elaborate this theme and offer a cognitive explanation for the desirability of direct manipulation (Hutchins et al. 1985). "Direct manipulation" has terminologically superseded "first-personness," although Hutchins, et al. identify first-personness as a characteristic of direct manipulation.)

Laurel elaborates the connections between theatre and computation in her book *Computers as Theatre* (Laurel 1991/ 1993). Finding the conventional notions of "interface" to be inherently murky and problematic, Laurel argues that

> an interface is not simply the means whereby a person and a computer represent themselves to one another; rather it is a shared context for action in which both are agents. … interface design should concern itself with representing whole actions with multiple agents. This is, by the way, precisely the definition of theatre (Laurel 1991).

Representation and agency are thus central concerns to both theatre and interface designers, who create "representations of worlds that are like reality only different." Enactment is central to both:

> Drama, unlike novels or other forms of literature, incorporates the notion of performance; that is, plays are meant to be acted out. A parallel can be seen in interface design (Laurel 1991).

grounded in make believe and play (Laurel 1991), employing what Janet Murray describes as the *active creation of belief* (Murray 1997).[35] Digital enactive representations—for example a file, icon, data structure, or component of a simulation schema—can function as make believe pivots, enactive vessels in which new meanings can be invested.[36] In the hands of a writer, simulation maker, or artist, a file becomes a letter to one's mother, a simulation variable becomes the population of a virtual city, and an icon turns into a postage stamp that represents, in turn, a mail program.

As scaffolds, enactive representations transform how we perceive, think, act, and feel. Spreadsheets, for example, encourage the organization of data into columns and rows linked by formula; contemporary email software organizes sequences of messages into threaded conversations, and draw our attention to unread messages with special markings; search systems encourage us to treat document collections as giant unstructured pools of information we fish into with carefully placed textual queries; my new Macintosh operating system by default shows me items from the *iTunes* store when searching my hard drive, encouraging me to buy goods from Apple.

To examine software representation in this way is to ask the following: What are the en-

---

She identifies the production of a feeling of participation in the ongoing action as a central concern of software design, which enables one "to *act within a representation."* Laurel asks "How can people participate as agents within representational contexts?", identifying the answer with the deep seated knowledge of actors and the primordial act of "make-believe" (Laurel 1991). This, of course, returns us to play.

[35]Laurel writes:

> Buried within us in our deepest playful instincts, and surrounding us in the cultural conventions of theatre, film, and narrative, are the most profound and intimate sources of knowledge about interactive representations. A central task is to bring those resources to the fore and to begin to use them in the design of interactive systems (Laurel 1991, p. 21).

These themes are later taken up later by Janet Murray, who calls the feeling of being an agent "agency," and the design practice of situating people as agents as "scripting the interactor." Murray contrasts the *active creation of belief* to Coleridge's idea of *suspension of disbelief.* We don't suspend belief so much as actively create it, a view congruent to the literature on pretend play reviewed in chapter 1 (Murray 1997).

[36]For an explanation of Vygotsky's pivot in play, see chapter 1 (Vygotsky 1978).

active representations that a particular software creates? How representationally flexible are they? In other words, can they function as a pivot, becoming a representation for something else?[37] As a scaffold, how does the enactive representation transform perception, thought, action, and feeling?

Software as Code

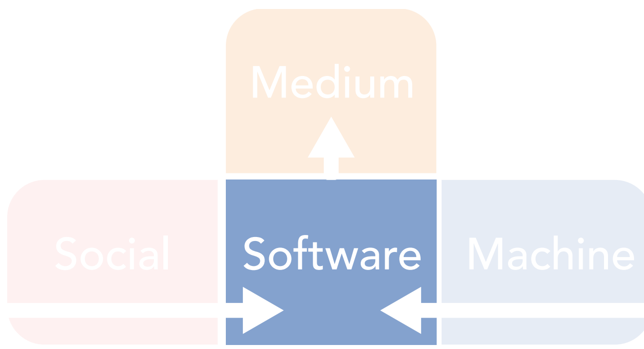


Figure 2.5: Software at the nexus.

Software is the nexus. It is how the protean computing machine is sculpted and transformed, reconciled to the agendas of communities of computing. Software gives rise to the digital medium in which we find ourselves immersed. Closely examining software artifacts in order to deconstruct artifice, recover hidden schemas, and surface evidence of developer struggles is a fine idea, but it raises the very practical problem of how we are to discuss code, the formal descriptions that define computer programs. How can we make coherent the many fragments of code, distributed within many source files, that constitute a software system? And how to do this within my prose—the text you are now reading? This is further problematized, first

[37]Mateas sees an "expressive architecture" as offering a "surplus" of meanings that authors (system builders) take advantage of. Such an "architectural surplus" might enable a new "way of thinking about the world." By considering discursive practices as an "explicitly manipulated resource," and manipulating those resources, new meanings can be created (Mateas 2002).

by the fact that I don't assume that you, my reader, are a programmer, and second by the diffuse and interlinked nature of *SimCity's* code in particular and software code in general. This complexity is such that, even for an experienced programmer, a discussion incorporating raw code would not be an ideal representation to think or talk with.

Reverse Diagrams

My solution to this problem is to borrow from the methods of game designers. Designers often make visual and textual design descriptions of systems that are then programmed, or translated into code. These design documents can be thought of as blueprints. In the best cases, these design documents are synoptic overviews that facilitate reflection, reasoning, and social discourse about a proposed design. I reverse the usual flow of translation from design into code, and produce retroactive design documentation that follows rather than precedes programming. If a game design document is a blueprint, then reverse diagrams are an archaeologist's field notes. Employing the synoptic techniques of a designer enables me to produce representations of code that allow reflection, reasoning, and discourse.[38]

[38]This approach can be seen to reconcile divergent approaches towards the study of software. Some scholars attend to the architecture and processes of software, while others make the source code itself the object of analysis (Marino 2006). Wardrip-Fruin, who favors the study of architecture and process, notes that these two approaches complement one another (Wardrip-Fruin 2009). Reverse diagrams allow us to smoothly move between and hybridize these analytic approaches, flattening code and architecture into unified representations to think with.

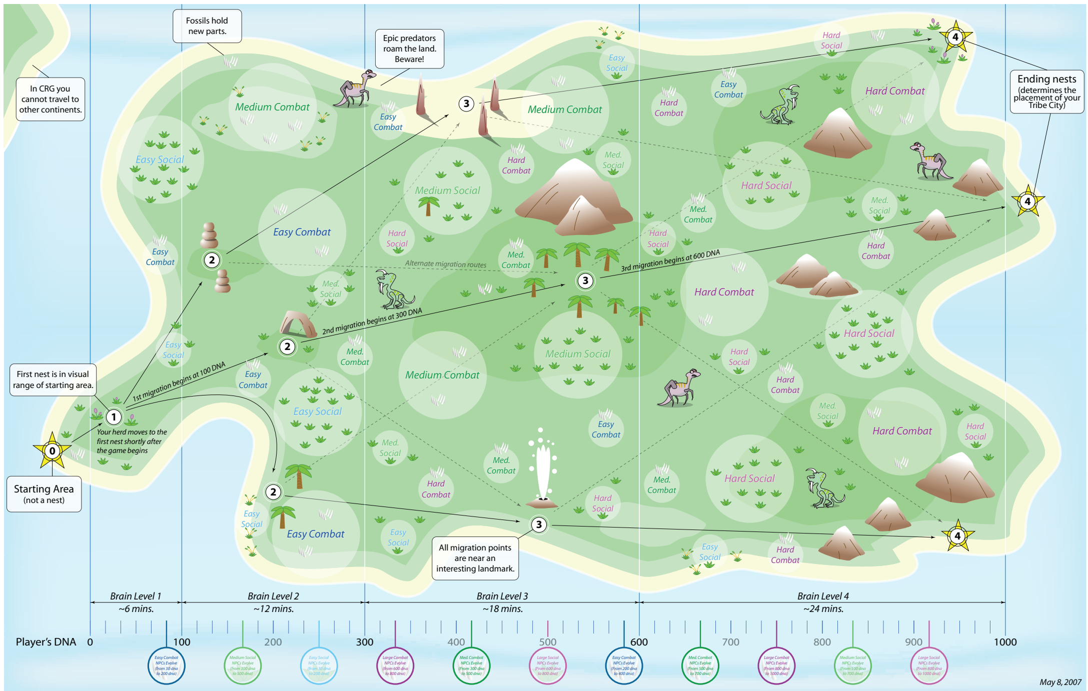


Figure 2.6: A one page design document for the game Spore *(2008) created by Stone Librande to describe the unfolding of its Creature Game (Librande 2010). Vector illustration courtesy of Librande.*

I opt for a highly visual and diagrammatic style inspired by Stone Librande's one page design document method. Librande's highly visual approach, in turn, is inspired by architectural blueprints, Lego construction manuals, cutaway illustrations, and the kinds of infographics one finds in an Edward Tufte volume (Librande 2010). One of Librande's insights is to limit the description of a system (or sub-system) to a single page, a constraint which enforces a beneficial economy of representation and thought. Resulting works provide holistic and systemic overviews that nonetheless elicit and capture fine detail. One page designs enforce conciseness and a panoptic understanding of the complete integrated whole, which in turn affords reasoning about and problem solving within a design, as well as social exchange.[39]

[39]Game design documents are typically long textual affairs that are integral to the design and production of games, yet often go unread on account of their unwieldiness. Observing that one of the most important functions of design is efficient communication—to convey a clear sense of what is to be built—Librande sought to remedy this problem. Drawing inspiration from architectural blueprints and renderings, step by step Lego instructions,

Drawing upon Librande's approach, I created informationally dense and visually succinct representations of distributed and interlinked sections of code. My retroactive design documentation maps the underlying design and logic of *SimCity*, making available its otherwise invisible software functions to analysis and discussion. Ultimately, the diagrams allow me to embed *SimCity's* complex and diffuse code into a flat and linear prose document (Latour 1986).[40]

Reverse diagrams accomplish two key functions: translation and mapping. As translations, they function as conversions of software source code into diagrammatic form.

---

cutaways, and infographics, Librande cultivated a practice of visually working through and explaining game designs in single pages. While there is nothing new about using appealing visual explanations to articulate a design, as evidenced by Librande's inspirational source material, his contribution is in methodically applying this strategy to game design. Although building these one page design documents is labor intensive, Librande argues that the designer's responsibility is efficient communication, so the labor of distillation and visual translation is part of their job description. Librande notes his approach does successfully get designs in front of more people, as their visual appeal and informational density exerts a magnetic appeal, causing people to hang them on walls and use them in meetings, and so on. Such designs can be annotated with pen and pencil, and easily discussed in a collaborative setting.

[40]By approaching the code artifact visually, through visual representation, I aim to accomplish what Latour describes in *Visualization and Cognition* (Latour 1986), and that is to flatten large, complex, and ungainly things in the world—in my case software code—into two-dimensional figures that capture the whole, are easily recombined, and embedded within a text that engages its aesthetic, cognitive, technical, and cultural dimensions.

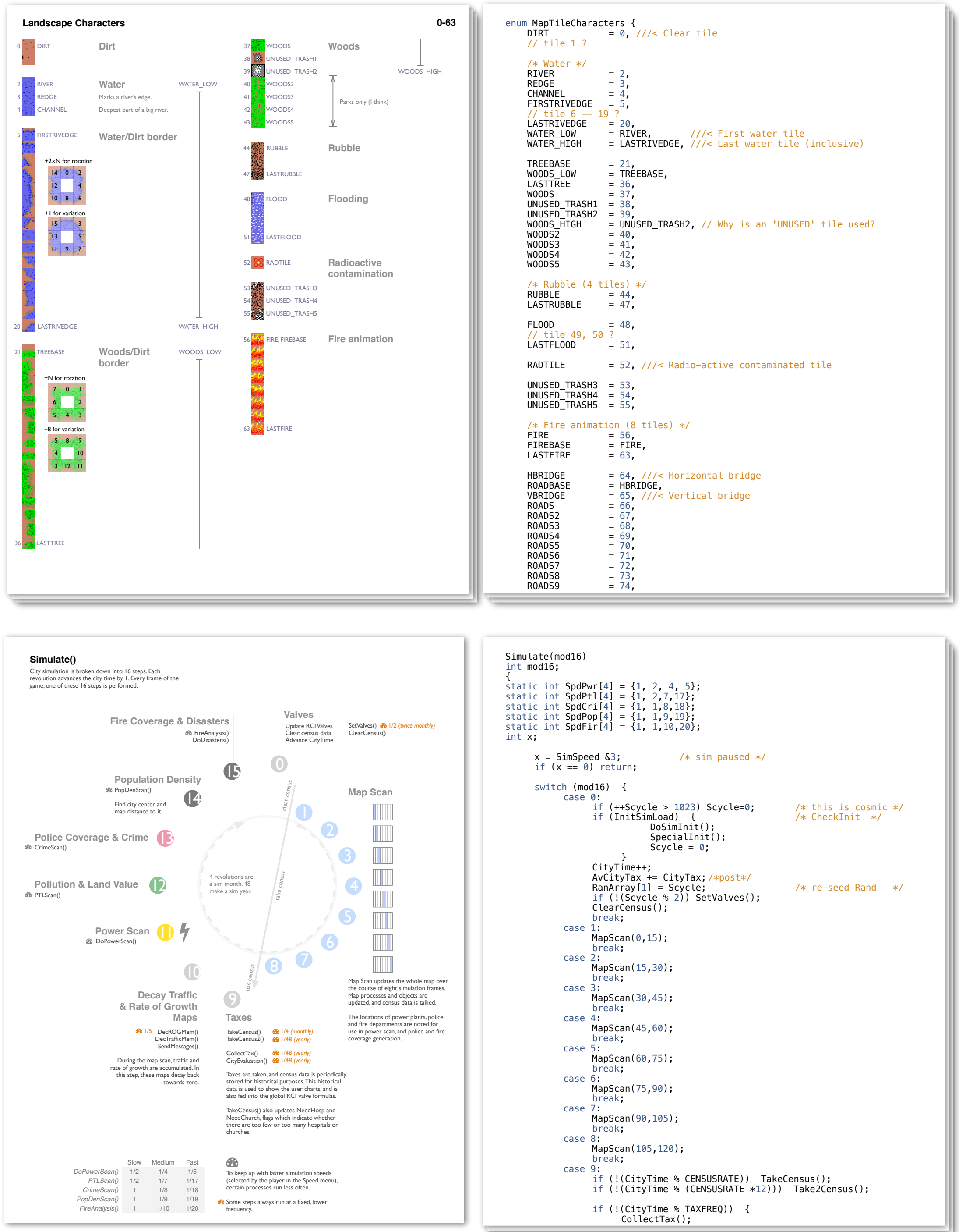


Figure 2.7: Two reverse diagrams illustrating translation. Diagrams on the left are translations of the code at their right. Top two figures describe map characters. Map character code includes organizational work and notes by Don Hopkins (see chapter 5). Bottom two figures are for the main simulation loop. Simulation code is from the original Macintosh version of SimCity.

As maps, the diagrams synthesize a sprawling network of software code. The diagrams grew out of my own note taking while surveying the code, and are the trace output of my attempt to understand how fragments of dispersed code interlink into coherent wholes. I took notes while exploring this archaeological city, and the diagrams both manifest and enabled my incremental building up of a synoptic understanding of *SimCity's* programming.

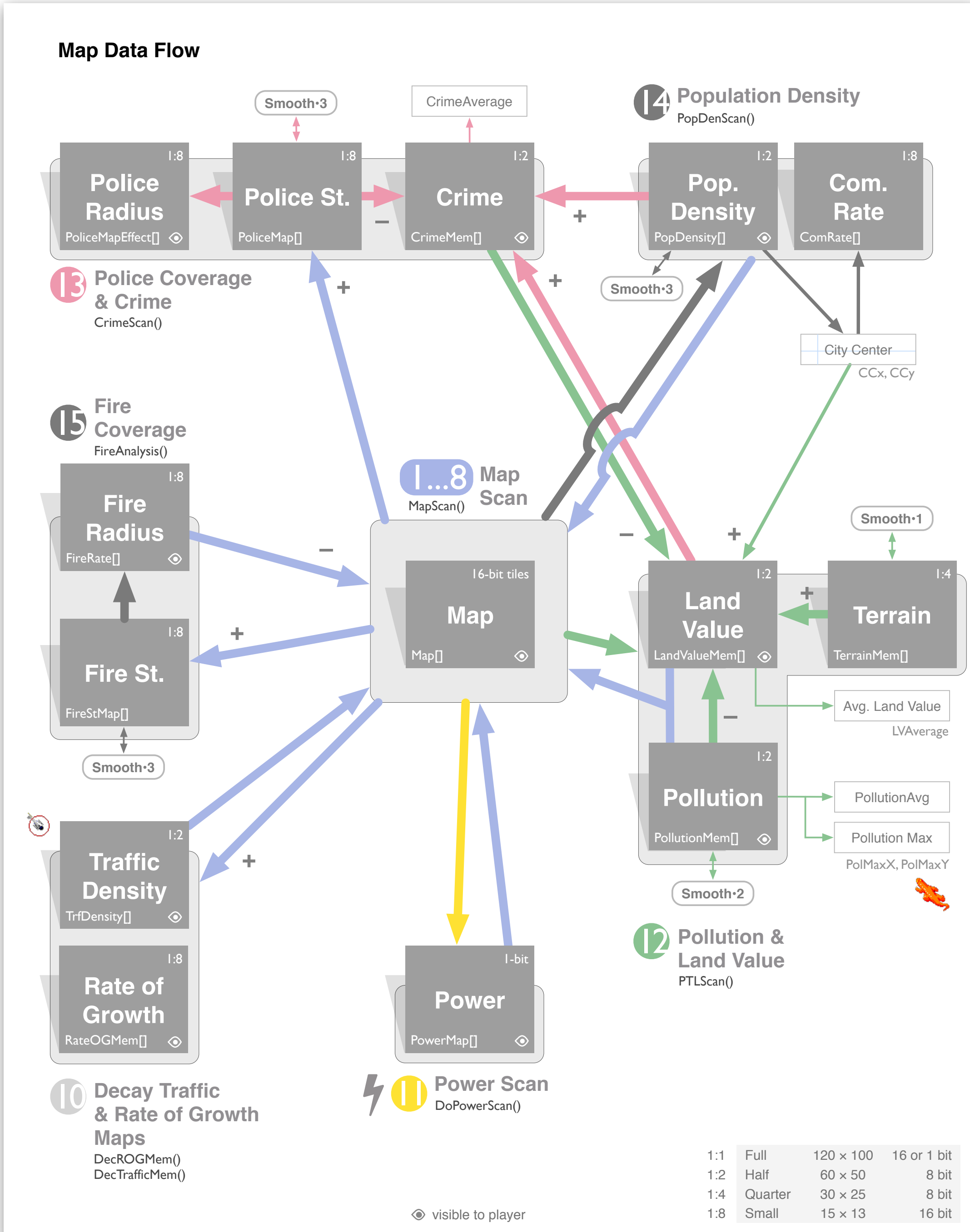


Figure 2.8: Reverse diagram illustrating mapping. I pored over many sections of code dispersed across many files (not shown) to build this single data flow map.

The diagrams I created are more than travel guides that shepherd the reader along the trail of my argument; they are maps of a territory. Because these diagrams are synoptic representations of code, they contain detail well beyond what is needed to follow my argument. In other words, the diagrams also function as territorial maps that others might use to chart alternate interpretations and journeys. This is why they sometimes contain cryptic seeming names, which are in fact symbols referencing code entities. In short, the diagrams should help you understand the text, but don't worry if they seem at times to contain extraneous or nonsensical information. (The appendix contains additional diagrams which didn't fit into the main text.)

Notes on Reverse Diagrams

Since they are a novel technique that others might want to use, I want to add a note about creating such diagrams. I believe that it would be disingenuous to offer prescriptive instructions on how to build such diagrams, since I see them as highly idiosyncratic and deeply grounded in craft. My diagrams express a particular interpretation of how *SimCity's* code works, attend to aspects that interest me, and describe it as my mastery of the craft of graphical and information design allows. I believe the most helpful guidance I can offer to aspiring diagram makers are the diagrams themselves (to be read in conjunction with the code), and some notes describing my process (as best as I can remember).

Reverse diagramming, as a craft, builds upon particular faculties: the ability to handle and read code and other assets that describe computer programs, interpret the meaning of that code and relate it to other phenomena of interest (dynamic behavior, software representation, history, interaction design, etc…), and design appealing and legible informationally dense graphics that conveys that interpretation. Diagrams, interpretation, and code feed one another. For example, I built many diagrams simply to scaffold my own understanding of *SimCity*. As my understanding improved, I was able to refine and build more complex diagrams. It's important to point out that I already knew a great deal about *SimCity's* design in a general sort of way from talking with Will Wright and reading interviews with him, which bootstrapped this entire process.

For *SimCity,* my diagramming process probably began with the character set that composes the map. In retrospect, it was an obvious place to start. First, the characters are inherently visual, while the symbolic and numeric constants in the code, along with their associated arithmetic operations, are not. I found myself struggling to understand what manipulations on characters as integers meant. A key was obviously needed, making it easy to see the correspondences. Second, Don Hopkins had more or less systematically collated all the character constants into one place, so I faced a linear sequence of graphics and linear sequence of symbols and integers that neatly, if not completely, mapped to one another. I had only to zip those sequences—character art, integers, and code symbols—together in order to build a handy reference diagram. As I built these diagrams of the characters, certain indexical patterns popped out, first with the organization of forest and water edges, and then with the buildings and network characters. Organizing the key in a way that manifested the intrinsic structure of the character ordering allowed me to visually see, on the page, why arithmetic operations on certain map characters worked as they did. Also, I wrote some scripts that stitched characters together to make this process less tedious. These scripts only became possible to write after I understood the organization of the characters.

Once I had the map character key, it became much easier for me to interpret more of the program. This is because just as a spreadsheet primarily traffics in numbers, and a paint program traffics in bitmaps, *SimCity* largely traffics in these characters. I would copy and paste bits of code into my diagrams to help me puzzle over the meaning of the code and the character set. Together, they formed a kind of puzzle whose solution I strove to make clear in diagram form. By putting bits of code into the diagram and working through the code and visuals together, the diagram functioned as scratch paper that evolved alongside my understanding of the code, which could later be deleted, transformed, or referenced as needed. Makeshift understandings gradually coalesce and solidify, are verified against code, and settle into the diagrams you see here.

The Simulate diagram is inspired by the layout of a clock, whose hands move clockwise. In code, a modulus and switch statement select the appropriate action in Simulate, and since this time slicing technique was already familiar to me, it was straightforward for me to conceive of it as a clock, a cyclical schedule of actions. From there, the circular layout became a scaffold around which I collected, elaborated, and refined notes on what was happening, when, and how often. I tried to capture the sense in which information flows around the clock face with the annotations about taking, using, and clearing the census. The color coordination between Simulate and Map Data Flow was a happy accident. Initially, I made step 11 yellow and added a lightning bolt to signify Power Scan, mostly for fun. Only later did I discover that this could be pushed farther. I color coded the edges in Map Data Flow to delineate the different phases, and to better establish the correspondences with the steps of Simulate.

Two of the most difficult diagrams to make were Map Data Flow and Map Scan, on account of how much information I condense into a single diagram. I believe this density helps the viewer to attend to both details and overview, and relate information at these two scales (Tufte 1990). Both of these involved a great deal of detective work across many different functions dispersed in multiple source files, as well as a great deal of visual design work. Map Data Flow was motivated by my desire to know where map data was being created and consumed. The answers were hard to find, and involved a lot of searching for symbols across source files. As the answers were not collected anywhere in particular, I started building a reference diagram. As it came together, I had some extra space that I use to hint at connections between maps and other parts of the simulation: between pollution and the monster, and between the maps and other global variables. The temporary maps used for smoothing would have made this diagram too large to fit on a page, and would have distracted from the content at hand, so they were moved to an ancillary diagram, leaving behind an indicator (Smooth•1•2•3). In any event, they function very much like subroutines, and the ancillary diagram functions like a diagram of those subroutines.

Finally, it is worth commenting upon the visual rhetoric of the Set Valves diagram. Despite the fact that Forrester would have represented this kind of formula as an algebraic formula, I have intentionally designed it in a way that evokes elements of Forrester's system dynamics diagrams. This is another highly dense diagram that took a great deal of visual design labor. Once I had all the elements accounted for, the diagram was a bit of jumble, so I worked hard to translate functional relationships into visual ones, for example by aligning elements so the diagram as a whole could be easily taken in. Page columns correspond to valves, and rows to the stages of processing. A shortcoming of this diagram is that the flow of information from bottom to top could be clearer; perhaps giant arrows in the background, gently indicated as watermarks, could help with this. The overview diagrams of all the simulation systems were made after Janet Murray pointed out that she didn't understand how all the subsystem diagrams fit together. Of course, I was too close to all of this to realize that, which highlights the importance of testing the diagrams in the use cases you have in mind—as part of a presentation, a discussion, or in prose—to see how they fare.

For all of these diagrams, I would estimate that more time was spent doing graphic design work than poring over code. I say this not to claim that reading the code was easy, but rather to underline how much effort has gone into clear presentation. Such work isn't merely aesthetic; lucid visualization is a means to and practice of lucid cognition.

Deconstruct Artifice

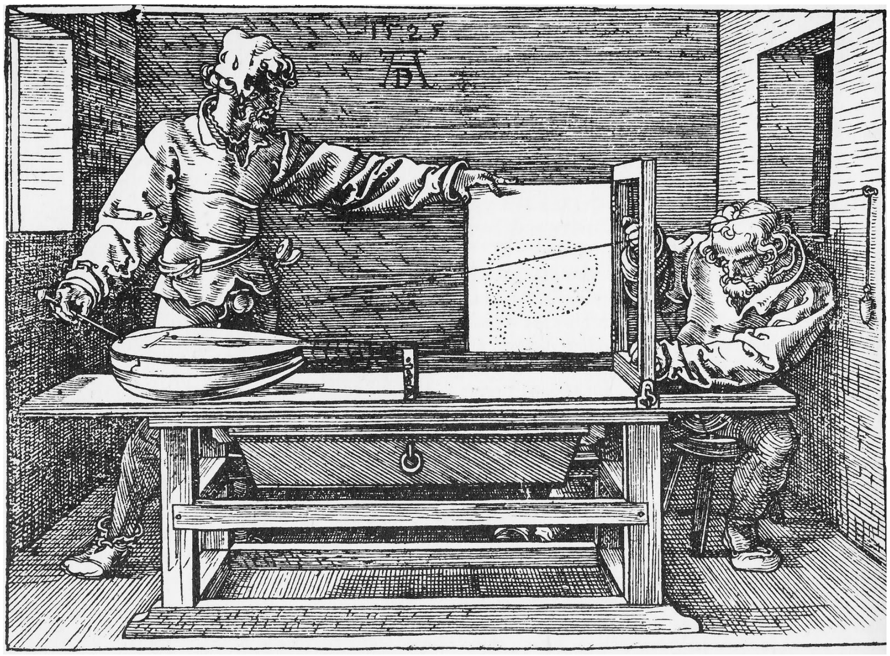


Figure 2.9: Albrecht Durer, Underweysung der Messung.

The enactive representations we encounter on our computers, such as user interfaces[41] and illusionistic three dimensional graphics[42], are created through artifice. Software artifice is an analogue of the craftwork of the novelist, painter, filmmaker, or actor—those who conjure the illusion of alternate worlds, characters, and stories. Look below the surface of our computerized enactive representations—windows, emails, buttons, photographs, simulated cities—and you will find not just a single layer of artifice, but many dizzying levels of abstraction, a telescoping series of enactive representations crafted out of other, more simple ones. Disassembling an enactive representation in order to explain its artifice is akin to demystifying the trickery of

[41] User interface designers also see themselves as producers of digital artifice. When we make contact with computer windows, menus, icons, text, mail, buttons, scrolling displays, and so on we perceive and manipulate a painstakingly crafted world. Researchers at Xerox PARC, where elements of the modern GUI were developed, conceived of the interface as the "user illusion." Illusions crafted in the service of interfaces are so important that while they were "once the last part of a system to be designed," they now come first (Kay 1984). The phrase *user illusion* suggests magic as well as a two tiered reality: the fictional world of the user, and the careful craftwork of the illusion maker who labors to produce this fictional world.

Bruce Tognazzini, an influential user interface designer, compares human interface design to stage magic, arguing that working designers should look to stage magic for lessons on producing effective illusions as well as the ethics of such illusion making. Tognazzini argues that many principles are held in common between magicians and interface designers: consistency, unity, simplicity, familiarity, user testing, character (as in: appeal and individuality), smoothness, and brevity. Magicians are well aware of the delicacy of their illusions, and manage two convolved realities: that of the magician, and that of the spectator. Drawing upon primary magic sources, Tognazzini goes on to discuss simulation and dissimulation, two techniques used in the service of misdirection—which is the craftwork of deception and artifice. Simulation is the synthesis of a new reality, while dissimulation conceals from the spectator something from the magician's underlying reality. Magicians and software also manipulate time, dissembling about when things happen, either doing something ahead of time for later presentation, or stretching out the performance of something for dramatic effect, e.g. making something seem harder than it really is. Houdini, for example, would often sit and wait out of sight while his audience thought they were witnessing his struggle to escape some contraption. Tognazzini contrasts Houdini's theatricality to the tic-tac-toe playing game on the Fairchild F, which would respond instantly to player moves, thus betraying no effort on the part of the machine. (Claude Shannon was well aware of this in the design of a machine that played the game Hex. Due to the game's design the machine would always win, and simply mirrored its opponents moves. It dissembled this fact by waiting before playing a move, producing the illusion of careful consideration.) Finally, Tognazzini stresses the importance of ethics and honesty in magic, extending the notion to software design: "the magician is not supernatural; the character he plays is. The computer is not capable of human intelligence and warmth; the character we create is" (Tognazzini 1993).

[42] Computer graphics, for example, extends the techniques of visual illusionistic representation into computational form. Especially since the Renaissance, Western visual artists have cultivated a craft of producing naturalistic illusion beyond the picture plane, a practice now taken up by practitioners of computer graphics, who produce computable equations for synthesizing naturalistic imagery. The SIGGRAPH research community (Special Interest Group GRAphics), now decades old, cultivates techniques for digitally synthesizing imagery and animation. The fruits of this labor can be seen in modern film special effects and computer games.

the illusionist. Not only do see how the trick is performed, but we manage a glimpse into the otherwise invisible structure of software.

Such analysis requires not just the technical skill to open the hood and take apart the machinery within, but the facility to interpret and explain this machinery[43], and understand how it engenders believability.[44] The reverse diagrams play a critical role in this process, and help me to take apart, understand, and explain software artifice.

The protean symbol processing machine is a flexible and powerful representational device. It is a mimetic machine (Laurel 1986) whose illusionistic power[45] has been both feared and cel-

[43]Interpretation and design are two ways on the same street. Cognitive scientist Donald Norman distinguishes users' *mental models*, formed through interaction with an object, from the object itself and the designer's mental model. Designers create a system that in turn shapes the user's mental model. "The designer must develop a conceptual model that is appropriate for the user … [and] must ensure that everything about the product is consistent with and exemplifies the operation of the proper conceptual model." Norman's advice, addressed to designers of physical objects and computer interfaces alike, is to consider carefully the conceptual models projected by their craftwork, and ensure that the designed systems are coherent and consistent with that model. This approach embodies a notion of human centered design as the artful crafting of operative representations (Norman 1988/1990). Norman implicates designers in the production of user mental models, whether they have done so intentionally or not.

[44]Joseph Bates argues that for a conception of AI rooted in believability and the arts, drawing explicitly on "a notion in the Arts of "believable character" ". Bates distinguishes the AI systems made by scientists, who have "effectively recreated scientists" in their system building, from a new practice of AI inspired by "artistic inquiry, especially the insights into character animation such as those expressed in *The Illusion of Life,"* the classic manual of Disney animation by Thomas and Johnstone, and other animation texts. Just as animators strive to produce through their craft an illusion of believable thinking and feeling characters, Bates's Oz project strove towards "believable interactive characters" he called "believable agents" (Bates 1994). Bates's approach marks a turn from *AI as cognitive science*, in which cognitive processes thought to take place in humans, such as planning, are reproduced in machines, to *AI as art*, where the objective is to create believable representations of agents. This research agenda was furthered by Bates's students. Michael Mateas, one of Bates's students at Carnegie Mellon University, calls the mutual informing of art and AI "expressive AI" (Mateas 2002). The behavioral patterns of *Pac-Man's* ghosts, for example, are optimized for producing a satisfying aesthetic experience; the ghosts do not crush the player with a brutal tactical efficiency borne from a purely technical desire to defeat the player. *Pac-Man's* ghost AI work to produce dramatic tension, not merely competition (Mateas 2003). Conceiving of an AI practice which extends, and is in conversation with, the representational traditions of the arts, is to consciously adopt a view of software as artifice.

[45]The *Eliza* effect emerges from the concealment of information, inviting projection (Weizenbaum 1966; Murray 1997). Wardrip-Fruin names the *Tale-Spin* effect for the eponymous software storytelling system. In his analysis, the Eliza effect "creates a surface illusion of system complexity," while the *Tale-Spin* effect "creates a surface illusion of system simplicity." Both illusions, because of how they engender misleading mental models of software processes, frustrate players. A happy alternative is the *SimCity* effect, which ostensibly reveals to the player a good approximation of what is happening inside the simulation, thus producing a more playable artifice (Wardrip-Fruin 2009). Wardrip-Fruin doesn't look at the *SimCity* code, so it is unclear how he arrives at this conclusion,

ebrated.[46] I am particularly interested in deconstructing illusions, pulling back the curtain, and

other than it appears to be the case. It is a simulation, after all, so things may not be what they seem. And while the careful sharing and concealment of information is a crucial factor in the production of artifice, as Zhu also points out (Zhu 2009), *SimCity's* deployment of these strategies is more nuanced than Wardrip-Fruin's analysis suggests. The coherence of mental models to system structure, and its implications for both interaction and design has also been taken up elsewhere (e.g. Norman 1988; Murray 1997).

[46]One of the earliest, if not the first, example of a believable AI agent was Joseph Weizenbaum's ELIZA program, which simulated a Rogerian psychotherapist, and gave rise to the phrase the "Eliza effect." Although Weizenbaum's actual research aim was in the domain of natural language processing, he was somewhat surprised and unsettled by the way people interacted with his creation (Weizenbaum 1966; Murray 1997). Weizenbaum sought to produce a plausible conversation partner as a means to study natural language. Having accomplished this plausible interlocutor, he was taken aback by its effectiveness.

"Like the Eliza of Pygmalion fame, it can be made to appear even more civilized, the relation of appearance to reality, however, remaining in the domain of the playwright." Weizenbaum, in the vein of a Houdini seeking to demystify someone who claimed to possess actual magic powers, sought to unmask the "illusion" and "rob ELIZA of the aura of magic" via an explanation of its inner workings. ELIZA is, after all, "merely a translating processor in the technical programming sense." Akin to a concerned magician who seeks to explain the power of a remarkable illusion he has, without fully intending to, created, Weizenbaum explores the characteristics that engendered ELIZA's "credibility." He appraises the "elegance" of ELIZA's illusion making craft, observing that it "maintains the illusion of understanding with so little machinery."

The Rogerian psychiatric interview was deliberately chosen by Weizenbaum because of a subject's expectations about such an interview.

> If, for example, one were to tell a psychiatrist "I went for a long boat ride" and he responded "Tell me about boats", one would not assume that he knew nothing about boats, but that he had some purpose in so directing the subsequent conversation. It is important to note that this assumption is made by the speaker. Whether it is realistic or not is an altogether separate question. In any case, it has a crucial psychological utility in that it serves the speaker to maintain his sense of being heard and understood (Weizenbaum 1966).

A therapeutic interview provides a cover story for concealing a "lack of understanding," and the speaker perceives and bestows believability upon the conversation and machine.

> The human speaker will, as has been said, contribute much to clothe ELIZA'S responses in vestments of plausibility. … ELIZA shows, if nothing else, how easy it is to create and maintain the illusion of understanding … A certain danger lurks here (Weizenbaum 1966).

Janet Murray offers a rich reading of Weizenbaum's work, addressing it from a literary, rather than artificial intelligence, standpoint. Rather than seeing ELIZA "in the coldest possible light," as does Weizenbaum the traditional AI researcher, and finding her to be troubling, Murray warms to ELIZA's exciting potential as a harbinger of computational literary artifice.

In *Hamlet on the Holodeck* Murray reframes the active contribution of "vestments of plausibility" (Weizenbaum 1966) on the part of the human as the *active creation of belief* rather than Coleridge's notion of *willing suspension of belief.* Reframing belief as actively created is particularly well suited to the *participatory* medium of the computer, and Murray suggests that we carefully consider how software designs cue the *interactor* to appropriate action as *scripting the interactor.* (I've italicized Murray's terminological contributions.) *Agency* is the term Murray uses to describe the feeling of being an agent—a crucial aesthetic quality of the digital medium. Digital representations make promises to the interactor, and when those promises are fulfilled, meeting the interactor's expectations, then the interactor experiences both agency and the active creation of belief is bolstered. Murray's text does the useful work of building upon prior suggestions in Laurel, thinking through, refining, and giving useful understanding how it is that a master painting, so to speak, constructs vivid and illusionistic senses of space, light, and feeling; how the craftwork of software illusion produces smoothness, continuity, and solidity; how it dissimulates and simulates; constitutes internal layers of representation; produces a feeling of directness, agency, and enactment, and invites the active creation of belief. Doing so allows us to closely examine the intangible stuff that software is made of, and answer a question at the heart of computation: given the abstract symbolic quality of the protean machine, and its resulting ability to flexibility represent anything, how is it that software comes to represent something in particular?[47] Examining the machinery of artifice calls attention to the schemas reified in software artifacts, reveals the agendas and struggles of its creators, and helps us to understand how it reshapes our lived experience.

---

names to important concepts identified, but not concretely pinned down by Laurel and others. Murray identifies key concepts and offers helpful terms for considering how, through interaction between human and computer, operational representations come to life (Murray 1997).

[47]Computers, recall, are automated manipulators of symbols. Whether these automated symbol manipulations refer to automatic language translation, arithmetic, airline bookings, surveilled communication metadata, or the logic of a game matter not at all to the computer. The traffic of representations into and out of computers, and by extension the operations of software models, gain representational force only through couplings to the outside world.

A symbol in a computer might come to represent something in the world via actuators and sensors. If an internal symbol in the computer produces the mark “A” on paper or screen then the symbol in memory, the number 65—whose numeric meaning, by the way, is itself still contingent upon numerous internal layers of encoding and convention—can come to stand for the letter “A.” And while the above is a rather trivial example, it’s easy to imagine how the mapping between symbols in the machine and things in the outside world becomes a far more complex affair in software models of cities, games, companion creatures, intelligences, and so on.

Critics pay close attention to the ways in which AI systems are described. Some see the use of intensional vocabulary, words like “planning” and “goal,” as a major stumbling block in artificial intelligence research. McDermott’s essay, “Artificial intelligence meets natural stupidity,” harshly criticizes what he calls an “epidemic” of “wishful mnemonics” (McDermott 1976; Agre 1997).

This question of representation and intensionality has vexed practitioners and critics of Artificial Intelligence and Artificial Life (e.g. Turing 1950; McDermott 1976; Searle 1980; Harnad 1990; Agre 1997; Hayles 1999; Zhu 2009). What does it mean to ascribe intelligence or aliveness to a machine or simulation? Can a machine ever be said to understand something? Can a simulation ever produce the qualities of aliveness? Are these qualities ever something other than narratives applied to computer software? If that were true, does that pose a fundamental challenge to the fields of AI and AL? At the root of such concerns, of course, are deep and longstanding philosophical questions (e.g. Turing 1950; Wittgenstein 1953).

Mateas turns these issues on their head, and sees the discursive practices around AI systems as expressive affordances (Mateas 2002).

Leverage Non-Academic Efforts

Despite their non-academic flavor, amateur efforts to preserve, document, and study old computer software and hardware are sophisticated and labor intensive enterprises that lay the groundwork for academic research.[48] For example, Montfort and Bogost's *Racing the Beam* is indebted to AtariAge (Montfort and Bogost 2009), an online community of Atari enthusiasts who collaboratively disassemble games, exchange game materials, and share programming and emulation information. Another example is the numerous open source emulation projects that allow old games and computers to be simulated on modern computers. Researching and building these emulators is complex and labor intensive work, without which the study of old software would be very difficult. Amateur efforts feed and sustain academic research, constituting an ecology of software study.

My analysis of *SimCity* is predicated upon prior work by Don Hopkins, who ported *SimCity* to different flavors of UNIX, including the One Laptop Per Child (OLPC), and also advocated for open sourcing *SimCity*. Hopkins, in releasing many iterations of the original *SimCity* over many

[48] In the game enthusiast and development community there are many sophisticated yet amateur efforts to interpret old software code: amateur code archaeology. *Pac-Man* has supplied much material for such efforts, as players look to understand its inner workings to effect better play (Pittman 2011), sometimes identifying and fixing bugs (Hodges 2015). Fixing such bugs indicates that the original design intent has been understood from the code and game behavior, found to have been inconsistent with the implementation, and has been rectified. Bugs reveal insight into the software developer's negotiation with the computer. All of these efforts are even more remarkable given that they rely upon disassembling and interpreting *Pac-Man's* machine code. Games such as *Super Mario Bros.* have been disassembled (1wErt3r 2012), and various "hacks" or modified games released, but *Pac-Man* has been subjected to the most thoroughgoing interpretation. Fabien Sanglard, working with the code to many classic games (*Another World, Prince of Persia, Doom, Quake*, etc…) has produced explanations of how parts of these titles work, often elaborating on the efforts of others, and relying upon diagrams and rewriting for exposition. Sanglard's audience seems to be mostly other programmers who are curious to know how the inner logic of these games works, and doesn't offer any comprehensive analysis—only partial explanations of subsystems. This is understandable because of the massive scope of these titles, and Sanglard's programming background (http://fabiensanglard.net, accessed May 2016). While these amateur code archeologists have undertaken technically impressive feats, only the *Pac-Man* projects elucidates and offers insights into the overall experience design, and none substantively go after questions of culture, aesthetics, cognition, design, and so on.

years, has produced a refined version whose code is often easier to read than the original. For example, *SimCity* cities are comprised of 956 graphical characters, and in the original code these characters are often referred to by number. Hopkins has introduced more symbolic constants into the code, and thus replaced these cryptic numbers with meaningful mnemonics. Kinked up and hard to parse assembly routines have been translated into human readable C code. At other times his code revisions are less clear, as they incorporate optimizations that render the resulting code faster, but harder to decipher than the original. My further efforts to explain the *SimCity* code in visual form elaborates upon Hopkins's work, and would not be possible without his preservation, interpretations, and advocacy.

Software as Play

I now introduce an analytic framework for studying software as play. Building upon the notion of scaffolding introduced earlier—computers scaffold, and play can be scaffolded—I explore the special relationship between play and computation. This framework explores ways to think about computers and software as a resource for play.

Games, Play, and Computation

Nicholas Negroponte, writing in 1979, states that it is through sensuality, emotional appeal, delight, and ease of use that computation will become widespread:

> Such startling advances and cost reductions are occurring in microelectronics that we believe future systems will not be characterized by their memory size or processing speed. Instead, the human interface will become the major measure, calibrated in very subjective units, so sensory and personalized that it will be evaluated by feelings and perceptions. Is it easy to use? Does it feel good? Is it pleasurable?
>
> It is this interface that will bring computers directly to generals, presidents of companies, and six-year-old children (Bolt 1979).

Negroponte could be writing about our precious smart phones, which through artful and sensual design, ease of use, and clear appeals to pleasure and delight, now pervade modern life. Are these qualities playful? Do computing and play exist in a special relationship to one another?

Many scholars have pointed out that a special relationship exists between computation and games. Essentializing games as rule driven, and computers as procedural, the argument is that games and computers are especially well suited to one another.[49] Even Miguel Sicart, despite launching a pro-play polemic against what he sees as the overly formalist and rule focused "proceduralist" bent of game studies and design, makes the case that computers are well suited to play because of their procedural and rule following prowess (Sicart 2011; Sicart 2014).

I offer another account of the affinity between computation and play. Yes, rules and procedures are important, but this is only part of the story. Something more profound than rules

[49]How was this conclusion arrived at? In articulating a field of game studies, formalists have identified all games—from soccer to chess to Tetris—as belonging to a coherent set, and thus subjects of a coherent field of study. Games as a phenomena are not specific to tennis balls, board games, language, social practices, or microelectronics, but rather to shared abstract properties. Formalists abstract away material details, and in doing so construct a set of things, and a field of study, encompassing things as different as the ancient game of Go, a two player strategy game that unfolds on a gridded wooden board with bits of rock and shell, and Nintendo's *Wind Waker*, a visually and aurally rich immersive story driven single player experience played with specialized computing hardware and software. What, they ask, enables such different phenomena to retain their essential game-ness despite being cast in different materialities? Constructing this set is a kind of alchemy in which specific games are transmuted, their material embodiments dissolving away, leaving behind a set of formal properties. What is left behind when games are abstracted from the specific social, cultural, and material circumstances they are embodied by and embedded within? The answer, not surprisingly, is rather abstract, emphasizing their rule based qualities. And it is these rule based qualities, the argument goes, that make games well suited to computers. Computers, as machines that follow programs, excel at rules, making them natural partners in game playing. Variations of this argument can be found in a number of places (e.g. Turkle 1984; Juul 2003). LeBlanc goes as far as to argue that games are computers, and computers are games (LeBlanc 2004). Essentializing games, and the casting of such a definitional net, in order to construct a unified field of game study has a long tradition. In *The Study of Games*, Avedon and Sutton-Smith note that particular taxonomic systems are far from inevitable, and have always "served the purposes of the categorizer" (Avedon and Sutton-Smith 1971). They then proceed to offer a definition of games, netting the object of study for a nascent field of game scholarship. (This stance anticipates Sutton-Smith's approach to play studies in terms of rhetorics.) Of course, the more recent desire to include computer games in the wider cultural field of human gaming undoubtedly motivates the emphasis on rules as a shared properties between games and computers.

links computation to play.

Computers as a Resource for Play

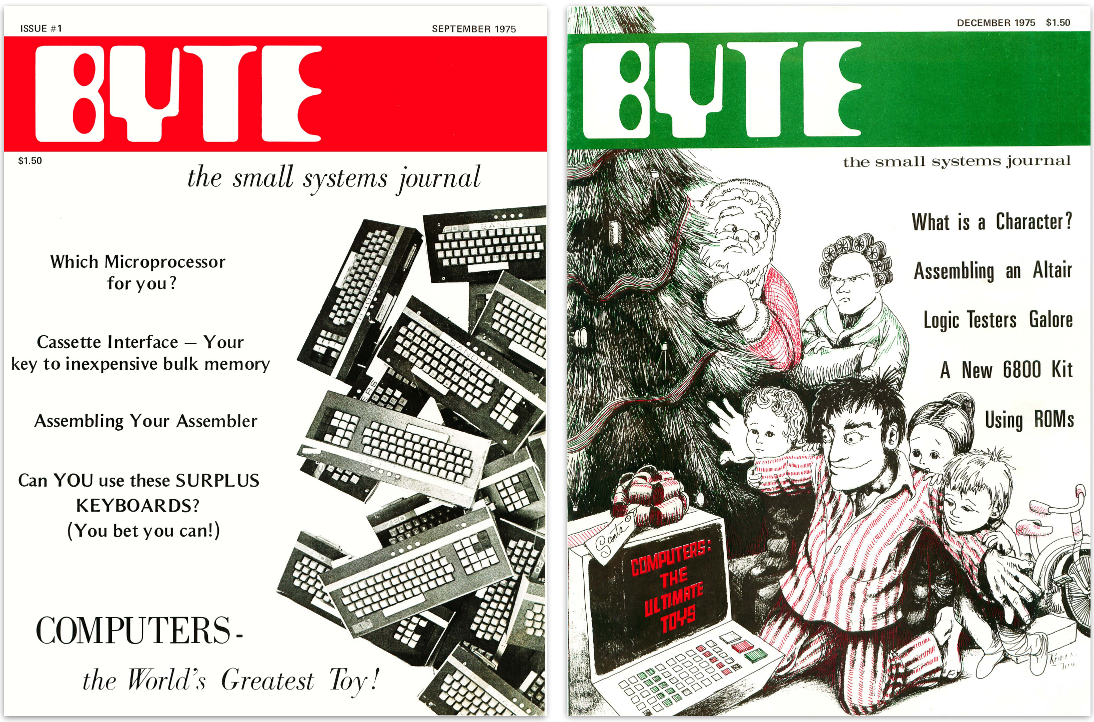


Figure 2.10: "COMPUTERS– the World's Greatest Toy!" and "Computers: The Ultimate Toys." Byte magazine covers from 1975, issues #1 and #4.

Computers are materials—like cardboard, paper, balls, string, and language—that are appropriated and fashioned into playthings. They are used to fulfill the playful desires of those who appropriate them. A software developer, when crafting a plaything, appropriates the computer for play, as does the player who uses a preexisting computer software artifact. But how do computers speak to the characteristics of play enumerated in chapter 1? What makes them good playthings and playmates? What kind of resource for play is a computer?

Intrinsic

How does computation lend itself to intrinsically motivated, autotelic activity? What makes it susceptible to the desires, autonomous activity, and transformations of players? Do computers harbor a special ability to produce delight and pleasure?

Agency is a central affordance and pleasure of computers (Murray 1997). From simple handheld electronic calculators to richly immersive game worlds, computers are responsive to the control and whim of players. Automatic computation gives computers a sense of aliveness, which is also a source of delight.[50] While agency is the pleasure of being a cause, autonomy means the ability to determine one's own course, and the power to appropriate resources to one's own ends. ATMs—Automatic Teller Machines—are computers that afford agency but little autonomy. To afford intrinsic activity, then, the computer must be available for player appropriations.

The history of computing suggests that computers become susceptible to appropriation when they are readily available, easy to operate and program, and open to receiving new meanings. When computers were expensive and tightly controlled machines they were available for playful appropriation only to a vanishingly small elite. As they became cheaper, diffusing through society, their susceptibility to appropriation increased.[51] But availability isn't

[50]Aliveness heightens the pleasure of agency. Computers, animated by the breath of automatic symbol processing, afford rule driven make believe (Vygotsky 1978; Turkle 1984), mystery and surprise, and delightful transformation. The automatic machinations of a screensaver or poem generator, the buzz of aliveness emanating from *SimCity*, and the effervescence of Conway's *Game of Life* are all akin to a bouncing ball—unpredictable, quirky, alive, responsive, and transformative. This is a source of deep delight, contributing to what Turkle refers to as the *holding power* of computational artifacts (Turkle 1984). Procedurality, another defining quality of computation (Murray 1997), gives rise to aliveness, as well as the player's voluntary and rewarding submission to rules (Vygotsky 1978).

[51]Consider the story of *SpaceWar!'s* development, which reveals the critical role of availability. Play proceeded from rarified machines entering more ordinary realms. One can see in such stories of playful appropriation a Promethean quality. Computation is delivered from the gods to the common man—a connection Graetz makes, but is perhaps too modest to call out directly, when he compares the inaccessible and tightly controlled room sized computers to "Gods," "approachable only through the intercession of The Operator," which he compares to

enough, computational affordances must also be easy to operate and program. BASIC, for example, is a language that made programming—and by extension control over computer resources—very easy, and was instrumental in giving rise to an avalanche of playful appropriations (e.g. Albrecht 1972; Ahl 1973; "What to Do After You Hit Return Or P.C.C.'s First Book of Computer Games" 1975; Wozniak 1986; Wright 2011; Maher 2013; Albrecht 2014).[52] Assimilability is how amenable a computational resource is to taking on new meanings. The enactive representations of computation, such as a programming language, the icons on a desktop, file data structures, or the levels and rates of system dynamics, function like Vygotsky's pivots, and can easily take on new make believe meanings invested in them by players, programmers, and users. This is assisted by stylization—miniaturization and abstraction, for example—which enables a thing "to be subsumed to the players' own causes, to be "assimilated" to the players' own fantasies" (Sutton-Smith 1986).

---

an "oracular shrine." Graetz goes on to write that "ordinary mortals" did not "fool around with" these computers. He describes the arrival of minicomputers as "A Stone's Throw from Olympus" (Graetz 1981). As soon as they become more available, they were pressed to playful ends.

There are many examples of computers—serious materials, after all, intended for serious ends, like military simulation of ballistics, weather forecasts, and nuclear detonations—to frivolous, unproductive, and pleasurable ends. For example, researchers appropriated computers intended for nuclear simulation to play with cellular automata (chapter 3). It is commonplace in the history of computing to find researchers and technicians turning expensive institutional computing equipment to unproductive ends, for their own pleasurable and intrinsic purposes.

Examples of such transformations, in which computers, at the time expensive machines destined for serious work, were pressed into pleasurable activities of limited function, resound throughout the history of computing. These examples aren't limited to MIT's particular hacker culture, either. Tennis for Two, one of the first video games, was created in 1958 by Willy Higinbotham and Dave Potter at Brookhaven National Laboratory, out of an oscilloscope and an analog computer. The game was made to engage laboratory visitors. Potter recalls that "The high schoolers liked it best. You couldn't pull them away from it" (Anderson 1983).

[52] See chapter 5 for a discussion of specific examples.

Figure 2.11: KidPix. *Source: https://www.youtube.com/watch?v=NJDOhPXhmnM, accessed Feb. 2016*

Software can be designed to support the intrinsic activities of players. Intrinsic activity is not directed at satisfying an external aim; it is of limited function. Intrinsic and extrinsic are opposed, an opposition that is easy to see if we compare Adobe Illustrator to Craig Hickman's *KidPix*, a child-like reinterpretation of the more serious *MacPaint*. *KidPix* was intended to be more robust and less delicate than *MacPaint*, eliminating the possibility of incorrect use. *KidPix* produces random and hard to control effects: it delights and surprises with wacky painting tools that leave streams of quirky graphics and sounds; an open door tool reveals new and surprising paintings beneath your own; to delete a picture you dynamite it (Hickman 2015). *KidPix*, in other words, is a grotesque transformation and mockery of a rational productivity tool. It is an inefficient plaything. It subverts, mocks, and recombines paint program conventions into something new.[53] So while *Illustrator*, *KidPix*, and *SimCity* all afford the manipulation of

[53] It mocks, in fact, much more than that. *KidPix* tools transform and mock the visual form of math problems,

symbolic worlds, only *Illustrator* is a serious tool, marketed as an efficient graphics program for professionals, whose output is destined for extrinsically oriented work activity. *KidPix* and *SimCity* are deliberately wacky, inefficient, grotesque, and resistant. They mock not only the genre conventions of productivity tools, but also the possibility of serious output. They are inwardly focused, intrinsically oriented materials.[54]

Safe

Play has been frequently used to make computers comprehensible, non-threatening, and enjoyable, redeeming them from alien threat and turning them into familiar, comfortable objects.[55] Although computers offer a possibility of safety through the buffered quality of symbol manipulation, a sense of safety is contingent on design, framing, and reception.

This is clearly seen in Episode 1 of the Canadian educational television show *Bits and Bytes*. The anxious computer neophyte, once a game is underway, remarks "They're really

---

dice, typewriters, games, and more.

[54] For the purposes of my argument, I draw a sharp distinction between software that is playful and software that is not, but in doing so risk running roughshod over some finer points. First, that play is characterized by a special orientation, and one could of course use *Illustrator* for play and *KidPix* or *SimCity* for serious activity. Second, work and play are not actually opposed (Cross 1990; Sutton-Smith 1999). Third, it is possible for serious work tools to be playful and draw upon whimsy and surprise. Think of *Photoshop* filters, and their expressive range from the subtle to the surprising and grotesque. A key difference is in how such tools promise or enable productivity: are they designed for efficiency? Suits's notion of inefficient means comes in handy (Suits 1978). *KidPix* and *Noby Noby Boy* for iOS are grotesque and inefficient caricatures that delight in subverting the efficient tool genres of graphics software and smart phones. But this doesn't prevent *MacPaint* or the core iOS functionality from being playful and delightful, or being used for play.

[55] When the computer in the film *WarGames* (Badham 1983) asks "Shall we play a game?" play is evoked as both a threat (anxiety over war simulation games; potential confusion between reality from the pretense of play), means to enlightenment (repeated self-playing leads the computer to learn that Global Thermonuclear War is a "A strange game. The only winning move is not to play."), and ultimately safety ("How about a nice game of chess?"). This film neatly captures attitudinal complexities towards both play and (for the time) computers. The computer's cultural redemption from that of a threat (Edwards 1997), which Fred Turner has charted as an interchange between mainstream computing culture and the 1960's counterculture (Turner 2006), is furthered and reflected through play. The computer hobbyists that first took up personal computers used them primarily as playthings (Campbell-Kelly and Aspray 2009), a the covers of the first two issues of *Byte* magazine make clear. In all this we can see the reiteration of Huizinga's basic argument that play is constitutive of culture, in this case play has worked to domesticate and deliver computing into the hands of the masses.

More examples of this are given in the upcoming section on *Attitude*.

friendly, aren't they?" (Beecroft 1983). Play has figured prominently in the marketing of computers, for example through corporate support of playful organizations such as the People's Computer Company (Albrecht 2014), IBM's use of Charlie Chaplin's Tramp character to market their personal computer (Campbell-Kelly and Aspray 2009), and specialized entertainment computers (e.g. arcade games and home game consoles). Play and games are familiar, safe, and pleasurable, appealing to and assuaging anxious users.

Transformational

Computers, at heart, are symbol jugglers. Through symbolic representation, computers ingest and recreate the world, producing caricatured symbolic realities. Symbolic representations—whether abstract or naturalistic—function as scaled down objects for play. Galaxies, cities, wars, financial markets, and sporting arenas unfold in miniature, on screens, within spreadsheets, and in the palms of our hands.[56]

Computers automate and assist in the recombination, transformation, and representation of symbols—words, tweets, three dimensional imagery, simulated households, sounds—that can come to stand for anything. Janet Murray describes this recombinant quality as *kaleidoscopic,* characterizing it as one of computation's essential characteristics (Murray 1997).

---

[56]Sutton-Smith remarks that "Since [play's] voluntary controls are hard to achieve, play takes place most easily in a setting established for that purpose and with scaled down objects; i.e., dolls" (Sutton-Smith 1972). He writes that video games miniaturize "the heavens, global battles, star wars, tennis games, Pacman, Donkey Kong, Asteroids, Space Invaders, and the like," reproducing the world in miniature form. Video games "reduce these worldly phenomena to our own scale and our own terms" (Sutton-Smith 1986).

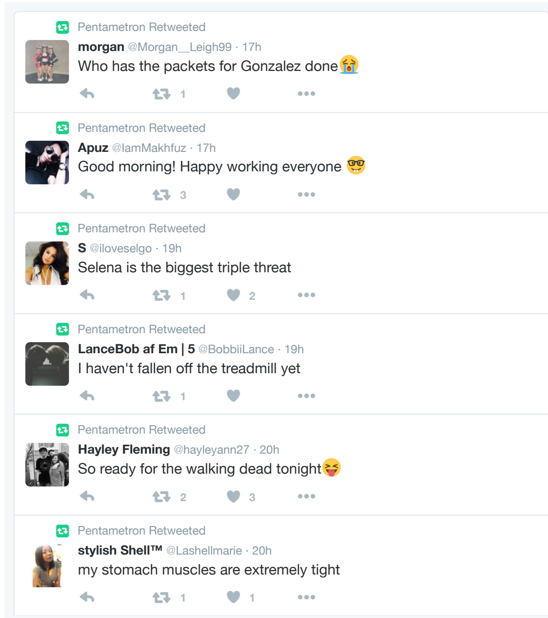


Figure 2.12: Screenshot from the Twitter feed of Ranjit Bhatnagar's sonnet-bot @Pentametron.

Transformation is automatic—computers are automated symbol manipulators. They are animated and lively, mutating, generating, and manipulating these symbolic landscapes and forms, as in the living worlds of *The Sims*, the randomly generated worlds of *Rogue*-likes and *Minecraft*, or the sonnet-bot @pentametron's automatic retweeting of iambic tweets into couplets (Bhatnagar 2012). While computers lack the commonsense knowledge that enables you or me to perform semantic operations—operations with an awareness of the *meaning* of symbols—they are players without peer when it comes to syntactic transformations of these symbols.[57]

[57] Programming and design enables them to recombine these symbols in complex ways, according to carefully crafted algorithms. Computers have special powers and limits when it comes to play's transformation, exaggeration, stylization, variation, and inversion of the familiar. Computers can generate and operate upon complexly modeled worlds, from cities to households and alien worlds, and provide carefully structured grounds for conviviality, like Twitter, but they do not know what we tweet about, or what transpires in these worlds—if they can be said to know anything at all. @Pentametron can identify and pair iambic tweets into couplets, but it cannot appreciate or craft the delightful semantic juxtapositions and poetry that result.

Separate

While computers can control things like printers, cash dispensers, microwave ovens, and nuclear reactors, they are, at heart, symbolic manipulators, which means that they function quite naturally in a simulative mode. The transformation and simulation of symbolic realities enables the enactment of actions buffered from their ordinary consequences. Just as toy guns scaffold, script, and enable the enactment of shooting, computation can enable and scaffold a huge variety of stylized and simulated actions. When, on your computer, you shoot space invaders, paint, or unleash Godzilla on your city, nothing dies and no paint is consumed. Computerized representations, bounded within the realm of the machine, are by nature fleeting and transient; they are ineffably small electrical charges percolating through circuitry. Consequences are limited.

But consequences in play are never totally limited. Tweets go around the world, and computers can be used to steer drones. Examples of non-separateness includes electronic health records, surveillance, website cookies and tracking data, or when computerized symbolic realities otherwise interconnect with those of the non-play world. This separating boundary between play and not-play, as we have seen, is porous and unstable.

Structured Variability

The computer excels at play's structured variability. Structure manifests in the procedures that govern the computer's automatic transformations, and the rules that govern its symbolic encodings. Variability emerges from rewritable computer memory. Agency, as we have seen, is inherent to the computational medium (Murray 1997). The computer is a reification of structured symbolic variability. But it is structured variability of a particular flavor. Unlike Lego or cardboard, the variations lack physicality—they are immaterial electrical impulses. And unlike the structured variability that animals and humans are capable of, it is a purely symbolic and autistic world the computer creates; its transformations are blind to semantics, feelings, and intersubjectivity.

Attitude

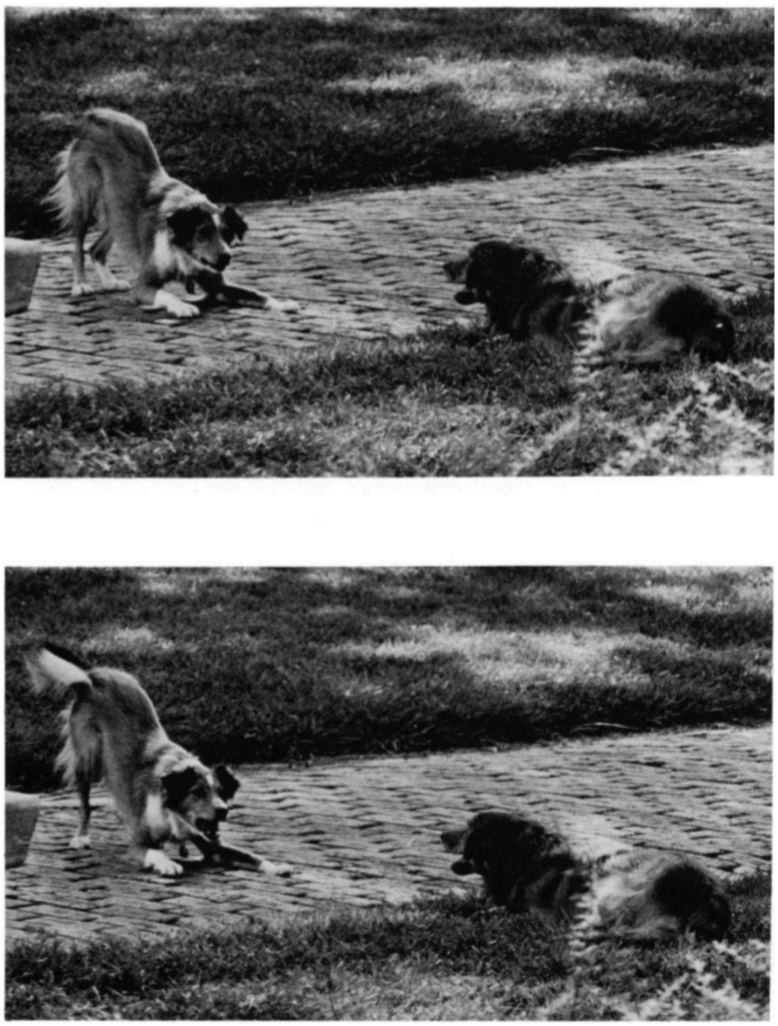


Figure 2.13: Play bark and bow. Image and its original caption from Marc Bekoff: "The dog on the left performs a "bow" in front of her desired playmate (upper), and when this was unsuccessful in initiating social play, she began barking and wagging her tail (lower). Barking may function as an attention-getting device" (Bekoff 1974a).

Software cannot, of course, inject a playful orientation into the hearts and minds of players. It can, however, appeal to the player, tirelessly signaling and inviting them to enjoin a play frame. Like the playful bow and bark of a dog soliciting play (Bekoff 1972), the culturally specific primary colors and simple shapes of a child's toys, the mischievous mien of a joke teller, a rising theatre curtain, or the guidance and inspiration of a more experienced playmate (Garvey 1970/1990), software such as *SimCity* can enjoin people to become players, and enter its playful frame. Computer playthings can strive to coordinate and configure people into players.

Computer programs don't harbor playful intents, but they can convey the playful intentions of others. Enthusiasts fashioned computers into playthings so that they could play with others, in particular visitors who, on demo day, couldn't understand the charm of their whirring hulking computing machines.[58] Playful demos welcomed visitors to research labs, functioning as gifts, toys, and playful offerings.[59] Whirlwind had *Bouncing Ball*—a tiny variation, really, on the

[58]Why should imbuing their gadgets with playful intentions render them more approachable and understandable? How is it that nerdy engineers, scientists, and technicians, not to mention marketers, could use play to warm up the world to them and their computers? Sutton-Smith offers a psychological answer to this question. Writing in 1986, when games such as *Dig Dug* and *Galaga* populated arcades, he points out that these games enact a contest between man and computer. Sutton-Smith writes that

> Video games are, among other things, a human response to the fear of the great machine. And it is the player's control over the machine that is most emphasized by those who play these games (Sutton-Smith 1986).

Play with computer games enable people to work through their anxieties about computers (Sutton-Smith 1986), just as third graders' love of riddles, "a game of rhetoric or arbitrary power in which victory is achieved by prior access to arcane knowledge," enables them to confront and acculturate to a world in which adults hold capricious seeming answers to questions. Riddles are "both a model and a mockery" of "adult-child process of communication," allowing children to practice and grow comfortable with this communication (Sutton-Smith 1973; see also Roberts and Forman 1971). Riddles, and games more broadly, are "expressive models" in which "conflicts are assuaged and … buffered learning occurs which has enculturative value" for a particular culture (Sutton-Smith et al. 1963). Sutton-Smith's psychological explanation argues that computer games enable a kid of conflict-enculturation to computer anxieties. Roberts and Sutton-Smith have identified "conflict enculturation" as the process by which children rehearse and assimilate psychosocial conflict via play (Roberts and Sutton-Smith 1962), and argued that games correlate to the cultures in which they are found (Roberts et al. 1959), players' psychological profiles (Sutton-Smith and Roberts 1971), and the problem solving strategies embedded within a culture's folktales (Roberts et al. 1963).

[59]Graetz, in his account of the invention of the seminal game *Spacewar!* (1962), describes this phenomena:

animated rocket trajectories it produced, ostensibly for serious military purposes, and which made for excellent material on CBS's *See It Now* television program, along with an animated welcome marquee, and music (Murrow 1951). The TX-0 had *Mouse in the Maze*, in which "a stylized mouse searched through a rectangular maze until it found a piece of cheese which it then ate, leaving a few crumbs" (Graetz 1981). It was written by Douglass T. Ross and John E. Ward with "the idea of having a good demonstration program for visitors" (Ward 1959).[60] The TX-0 also had the interactive proto-screensaver HAX, and a Tic-Tac-Toe game. *SpaceWar!* became the most famous example of such a playful offering (e.g. Kuhfeld 1971; Brand 1972; Kay 1972; Nelson 1974b; Graetz 1981; SAIL Timesharing System 1991).[61]

Shared

How do computers participate in shared pretense? While computers do not hold the *as if* attitudes of pretense, they are capable of supporting whatever pretenses we assign to these symbols. As we have seen, this leaves space for discursive practices about software and

> When computers were still marvels, people would flock to watch them at work whenever the opportunity arose. They were usually disappointed. Whirring tapes and clattering card readers can hold one's interest for only so long. They just did the same dull thing over and over; besides, they were obviously mechanical — at best, overgrown record changers — and thus not mysterious. The main frame, which did all the marvelous work, just sat there. There was nothing to *see*.
>
> On the other hand, something is always happening on a TV screen, which is why people stare at them for hours. On MIT's annual Open House day, for example, people came to stare for hours at Whirlwind's CRT screen. What did they stare at? Bouncing Ball (Graetz 1981).

[60] For more information on Mouse in the Maze see *https://www.youtube.com/watch?v=ki95Z8Tx8go* and (McKenzie 1974).

[61] *SpaceWar!* was consciously fashioned by Graetz and Russell in order to entertain and engage visitors with the new PDP-1 computer and, in the process, show off the machine's capabilities. In order to "involve the onlooker in a pleasurable and active way" they consciously chose the format of a game. Why a game? Because while a visitor might not know how, or feel comfortable, engaging a computer, a visitor does know how to participate in a game. They play it. Furthermore, you know why you should play a game: because it's pleasurable—it is, after all, an end in itself. Not only that, but games are approached with expectations for engagement: there are players, moves, winners, losers, and fun. The *SpaceWar!* program transformed the PDP-1 computer into a plaything, a medium through which Graetz and Russell transmitted their playful intent, and entertained visitors (Graetz 1981).

computation to co-construct phenomena such as artificial intelligence, artificial life, and simulation. Through mimesis—images, sounds, animations, behaviors—the computer as media beckons us to join make believe realities, where particular symbols come to stand for particular things. And as we have seen (e.g. Vygotsky), both make believe play and rule based play are predicated upon consistent behavior: rules. And computers are excellent, if inflexible, rule followers.

How is a shared intent to play fostered? Computers can tirelessly solicit play, and like a teddy bear, take on comforting forms that signal and bring about a relaxed field. Computers, when imbued with playful intents, issue signals that function very much like the play bow or stylized play bark of a dog (Bekoff 1972), cultivating a sense of safety, non-threat, and playful orientation. Computers can send the metacommunicative signals that signal *this is play*.

And what about shared attention? As we have already seen, designers—from game developers to the creators of graphical user interfaces—have refined how computers and humans jointly act in shared attentional scenes.

As active agents, computers can take on some responsibilities of scaffolding the intersubjective play frame. Like an older sibling, parent, or playmate, they can guide and inspire, coaching make believe transformations, rule based play, and pretense. Computation can guide us into play's shared frame of pretense, intent, and attention.

Chapter 3

System Dynamics

Introduction: SimCity and System Dynamics

The broad plan of this thesis is to arrive at a set of play design principles by undertaking a series of case studies. We began, in the first two chapters, by defining methods for analyzing play and software. The last chapter will conclude with play design principles, patterns drawn from the case studies that make up the middle of the thesis. *SimCity* is the focal case study, but before taking it up in full we turn, in the next few chapters, to projects that informed its creation. Originating as a plaything Will Wright created for himself, *SimCity* is a hybrid composed of the raw materials of two simulation practices: system dynamics and cellular automata (Rouse 2001). Approaching these simulation techniques as resources for play, Wright appropriated them, turning them to his own playful ends (chapter 5). To fully appreciate this appropriation, as well as to flesh out additional play design case studies, we look now at these precursor works through the lenses of software and play.

SimCity is unusual in that it simulates a social system. Computer simulations and games typically model simple physical phenomena. Take, for example, the bouncing balls simulations of *Pinball Construction Set, Pong*, and *Breakout,* the fictional ballistics modeled in *Raid on Bungeling Bay,* or the real world ballistic tables generated by ENIAC. *SimCity*, alongside Wright's other best known work, *The Sims*, are unusual in that they model *social systems,* pushing beyond the balls, bullets, and budgets of most games. Modeling ballistics on a computer, however, is a straightforward task involving expressing Newtonian mechanics in computer code. But how does one model a social system on a computer? What does this even mean?

To accomplish this feat Will Wright borrowed heavily from the work of Jay Forrester, a computer simulation pioneer who elaborated a practice for abstracting and representing social systems as networks of feedback loops. Forrester began by modeling economic entities (1956, 1958, 1961) and then explicitly turned to social systems (1960, 1964) before attempting to model cities (1969) and ultimately the "demographic, industrial, and agricultural subsystems" of the entire world (Forrester 1971/1973). The simulations encode the behavior of cities, companies, and the world as dynamic numerical systems. After running, they leave behind trace output in the form of non-interactive numerical output and graphs. They are highly abstract mathematical affairs. In published form, the trace output gains dynamic representational force in the minds of readers via Forrester's descriptive text, graphs, and diagrams. These clockwork worlds spring to life in one's imagination through these supplementary descriptive devices, a practice shared by many computer simulation traditions (chapter 2).

While working on the *Raid on Bungeling Bay* map editor which would evolve into *SimCity*, Wright discovered and drew upon Forrester's prior urban dynamics simulation. In examining Will Wright's work alongside that of Jay Forrester, the specific influence of urban dynamics upon *SimCity* can be seen, in addition to the broader effect of Forrester's simulation practice upon Wright's oeuvre. Forrester's overall approach, which he called system dynamics, applies the cybernetic paradigm of feedback and control to social systems, laying the groundwork for both Forrester and Wright's social simulations.

While Forrester intended system dynamics to function as a serious simulation practice, Wright combined it with a highly visual, vivid, and responsive aesthetic informed by computer games, toys, and the long history of human computer interaction design. System dynamics, in other words, became one of many materials swept up into Wright's computer play. Wright initially made software toys for himself, but some of these toys would eventually become public, elaborated, translated, and refined in collaboration with others into commercial products such as *SimCity*.

Wright appropriated preexisting simulation practices for play. But where do simulation practices come from? What does it mean to invent one? And what is a simulation practice, anyway? What, exactly, constitutes one? In this chapter as well as the next, we will answer these questions by tracing the formation of two simulation practices Wright made heavy use of not just in *SimCity*, but in his later simulation works. The software lens of chapter 2 helps us by focusing our attention on the actors, discursive practices, social practices, craft, schemas, artifacts, and enactive representations that collectively comprise them. (My software analytical framework, in fact, is informed in large part from my examination of the history of system dynamics and cellular automata.) Tools, publications, practices, and enactive representations perpetuate a schema, a particular way of thinking and model making. As a schema, system dynamics is abstract enough to accommodate new and creative applications.

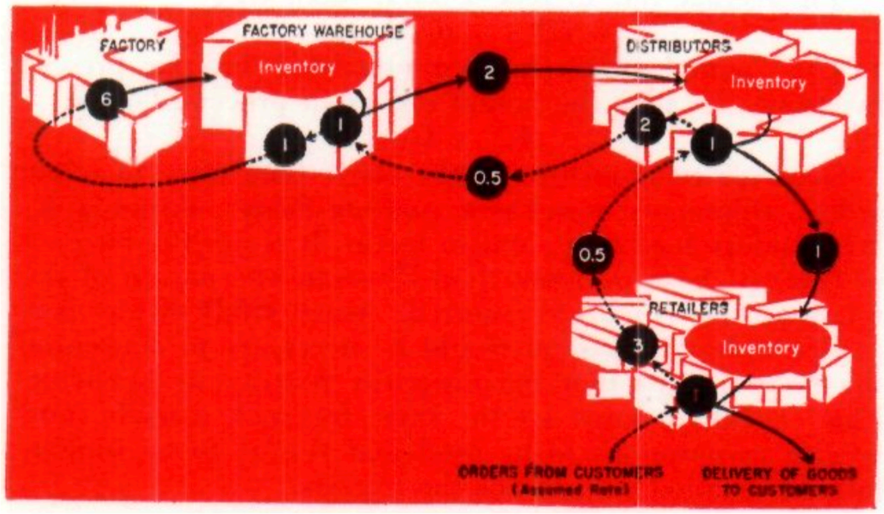


Figure 3.1: Levels and rates in a supply chain (Forrester 1958).

Forrester and Wright's simulation of diverse and complex subjects is predicated on the aggressive and artful abstraction of a domain into systems composed of variables interwoven into positive and negative feedback loops. There are two key components in this schema: levels (the variables) and rates (the feedback loops). Levels represent a quantity such as population, pollution, hunger, or money. Rates transform one level into another, functioning like regulatory valves that react to the system state. But where does this schema come from? How did it gain popularity? The approach is indebted to the cybernetic paradigm's abstraction of psychological, mechanical, social, and neurological phenomena into a unified notion of feedback and control.[1] These strategies have made their way from World War II era fire control technologies developed at MIT, to Forrester's simulation work, and into *SimCity* as well as

[1] Cybernetics provided a totalizing perspective for assimilating an incredible diversity of domains to computer simulation, of which Artificial Intelligence and Artificial Life are the best known examples (e.g. Hayles 1999). Forrester's system dynamics work can be seen as pioneering the application of the cybernetic paradigm towards social systems, even if Forrester himself resisted classifying his work as cybernetics.

Richardson attempts to tease apart cybernetics and servomechanisms, arguing that Forrester's work belongs to a different intellectual history than that of cybernetics (Richardson 1983). Even if distinctions can be drawn, and many can, for my purposes it is enough to group them together as emerging from the same broad intellectual moment, focused on feedback systems, and inspired by the same World War II era prototypes.

Wright's other projects.

Forrester intentionally cultivated and taught this representational practice. He used it himself and taught it to others, disseminating specific simulation models as well as an overarching simulation craft and philosophy. The growth and spread of system dynamics was accelerated by tools such as the DYNAMO compiler, classes at the Sloan School of Management, a diagrammatic visual language, and publications such as Forrester's books and specialized journals (*Dynamica* and the *System Dynamics Review*). System dynamics was intended as a schema which others could use to build their own models. It was built to be appropriated.

One of the strange ironies of system dynamics is that while Forrester took it seriously, seeing it as a hybrid of social science, management, and engineering, he eschewed the use of real world data. He encouraged practitioners to draw upon their intuition, and see model making as an art. It was a curious disciplinary fusion, and while the models might not be very good science or engineering, and often had little predictive success—*Limits to Growth* is the famous example (Meadows 1972; Edwards 2010)—they made for excellent make believe. In the final analysis, we see the simulation practice itself as an excellent resource for play. The levels, rates, and feedback loops that constitute system dynamics become, in the hands of Forrester, his students, and Wright, abstract primitives for world building. Abstract enough to constitute a stylized representation of just about anything, system dynamics—its craft, schema, enactive representations, and publications—were an ideal scaffold for Wright's own make believe computer play.

Genesis: From Servos to Supply Chains

> Our social systems are a great deal more complex than the information-feedback systems that have already been mastered in engineering. Are we ready to tackle them?

(Forrester 1960/1975)

Science and Divination

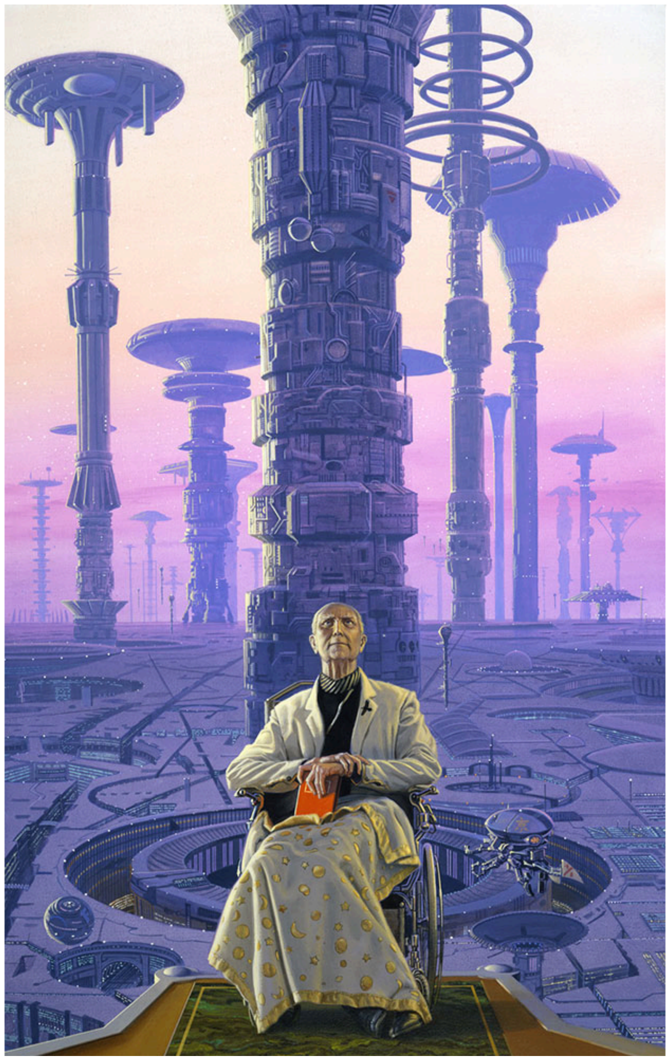


Figure 3.2: "Hari" by Michael Whelan. Cover illustration of Hari Seldon for Asimov's Foundation. Source: michaelwhelan.com

In Isaac Asimov's *Foundation* trilogy—written while Forrester's career with computers and simulation was getting started—the fictional mathematician Hari Seldon elaborates a futuristic form of social divination based upon statistics called psychohistory (Asimov 1951; Asimov 1952; Asimov 1953). While foretelling the future of an individual person would be as impossible as predicting the motion of a single molecule, populations in aggregate—such as the molecules constituting a gas, or the multitudinous citizens that comprised the Galactic Empire—could be modeled statistically. A set of equations, stored in a device called the Prime Radiant, is used to predict the future development of humanity.

Soon after Asimov finished the original *Foundation* trilogy, digital computer pioneer Jay Forrester began what would become a lengthy exploration of computerized social simulations. Like Hari Seldon, Forrester used equations to model economic and social systems, quantifying population aggregates, material resources, and the collective social attitudes and phenomena that a sociologist might call social facts. Both the predictions of Seldon and Forrester had political consequences. While Seldon strove to hasten the Galactic Empire's rebirth following a predicted collapse into a dark ages, and was exiled for these treasonous projections, Jay Forrester's world simulation work, prompted and funded by a concerned global elite, also predicted a civilization crash, stirring widespread popular and political attention (Forrester 1971/1973; Edwards 2010; Wikipedia, Foundation series, 2015).

Psychohistory, albeit fictional, is one of many analytical overlays one can use to model and study social systems. Game theory, which originates in the study of parlor games, is an analytical lens that has been used in the study of social sciences. It considers social systems as *games* enacted by competing and cooperating *players,* and provides mathematical tools for studying the strategies of such players (Leonard 1995). Forrester's system dynamics applies a different metaphorical overlay, transmuting social systems into interwoven *levels* of people, resources, and immaterial social facts that *flow* from one to another in a network of feedback loops. While game theory originated in the study of actual games, system dynamics emerged from the study of feedback and control structures cultivated during WWII.

Jay Forrester

Jay Forrester was born in 1918 to college educated cattle ranchers. An electronics tinkerer from a young age, he obtained, in 1939, an undergraduate degree in electrical engineering from the University of Nebraska, and then moved to the East Coast to take a research assistantship at MIT.

At MIT, Forrester worked under the auspices of electrical engineering professor Gordon Brown, a pioneer in the feedback control systems and servomechanisms which formed an integral part of WWII technology research. In 1940 Forrester become associate director of and co-founded, with Brown, MIT's Servomechanism Laboratory (Lane 2007). During World War II Forrester, alongside Bob Everett, who would later become the associate director of the Whirlwind computer project, developed "hydraulic servomechanisms for the control of radar antennas and gun mounts" (Forrester and Everett 1990), at one point personally installing an experimental system in a Navy vessel in the Pacific theatre, which gave him a keen appreciation for the link between research and real world problems (Lane 2007).

The basic idea of levels, rates, and feedback that informs and animates system dynamics can be traced back to these servo systems. But what is a servomechanism? We can best understand them by thinking about the thermostat that controls the furnace in your home. This thermostat is a classic example of a servomechanism.[2] It turns your furnace on and off by comparing the air temperature to the desired temperature you have programmed. The servomechanism closes a feedback loop between the current air temperature, a desired air temperature, and the furnace's activation. By doing so, the thermostat regulates the temperature of your home. Servomechanisms automatically provide feedback, guiding a system into

[2]This also happens to be the classic example from cybernetics (e.g. Wiener 1961).

a particular state of affairs.

Within Brown's Servomechanism Laboratory, Forrester would come to oversee the development of the seminal Whirlwind real-time computer project. Whirlwind began life as ASCA (Aircraft Stability and Control Analyzer), an analog flight simulator for training pilots and studying plane designs, and was funded by the ONR (Office of Naval Research). The ASCA project began in 1944, and in a process Paul Edwards describes as "mutual orientation,"[3] morphed into the Whirlwind digital computer project, transforming again under the sponsorship of the US Air Force, and the emerging political climate of the Cold War, into SAGE (Edwards 1997). SAGE, Semi-Automatic Ground Environment, was a massively ambitious semi-automated command and control system for tracking and intercepting Soviet airplanes.

By 1956 Forrester "felt the pioneering days of digital computers were over," and joined the MIT Sloan School of Management, which was founded in 1952 with a $10 million grant from Alfred Sloan, the former CEO of GM. This endowment was predicated on the notion that a technical institution would produce a unique type of management school, different, and perhaps better than those at liberal arts institutions (Forrester 2007). Forrester found the shift from computing to management natural, as he had already been involved "in running a several billion dollar operation [Project Whirlwind] in which we had complete control of everything: writing contracts, designing computers, deciding what went into production, and managing a vast enterprise that involved the Air Defense Command, the Air Material Command, the Air Research and Development Command, Western Electric, AT&T, and IBM. So going into management was not really a change" (Forrester 1995).

---

[3]Researchers attuned to the interests of funders, and funders adapted their agenda to the capabilities and interests of the researchers. Whirlwind historians Redmond and Smith refer to this reorientation as "mutual accommodation" (Redmond and Smith 1980).

Regenerative Loops

Embedded within the Sloan School of Management, Forrester laid the intellectual foundation for his future simulation technique. Although Forrester would later characterize system dynamics as the study of social systems, his early aims were not so grand. Initially, Forrester conceived only of applying computer simulation to the study of economic systems, and argued that the servomotor techniques successfully applied to military ends held promise for economic problems. When Forrester

> came to the [Sloan] School from his background in feedback control systems, computers, and practicing management, it was for the planned purpose of searching for and developing the linkages which might exist between engineering and management education. It was the expectation that these lay in the areas of operations research and the application of computers to processing management information (Forrester 1968).

Operations research, which sought to apply mathematics and science to industrial decision making, failed to consider the effect of feedback loops—a simplification Forrester found overly reductive (Forrester 1968). Of the possible applications of computers to management, Forrester found processing management information to already be a crowded field, and operations research to be interesting but lacking the “compelling practical importance” he valued (Forrester 1992).

Instead, Forrester proceeded with a different vision of the synthesis between computers, control systems, and management. In a 1956 faculty research seminar note, Forrester articulated the philosophy that would propel his economic and social models. He noted the failure of “most economic models”

> to reflect adequately the structural form of the regenerative loops that make up our economic system. The flows of money, materials, and information feed one another

> around closed re-entering paths. In one of these paths a disturbance causes a cascade of events that return to affect the initial disturbance. For example, such a system often has characteristics which can convert a single isolated impulse of disturbance into a series of oscillations (Forrester 1956/2003).

Modern economics and operations research were not up to the task of capturing the behavior that emerged from the "structural form" of "the firm and the economy," and

> [t]he behavior of such loops (their tendency to amplify or dampen disturbances, their natural frequency of oscillation, their ability to shift the phase or timing of events which feed into them) is determined by characteristics which are usually omitted from the models in the literature (ibid.)[4].

In accounting for the structures and behaviors economics missed, Forrester metaphorically mapped the structure (regenerative loops) and behavior (oscillation, delay) of electrical and servomechanism systems onto the structure and behavior of economic systems.[5] A "better analogy," he would soon write, was to be found in the "engineering and military models of telephone systems, of aircraft, of military systems, and of missile controls" (Forrester 1960/1975).

Economic systems were like the servomechanisms he was familiar with, and should be studied as feedback systems. In this seminar note, he proposed new strategies for studying economic systems, further drawing upon his engineering background.

First, he would import techniques for understanding the dynamic behavior of complex systems stemming from the study and use of servomechanisms in "complex military weapons systems." This included the use of differential equations for describing time-dependent, closed, sampled data systems. (The specific schema of levels and rates would come later.)

[4]By ignoring feedback, economic models omitted important characteristics such as a "resistance to change," accumulation of materials, temporal irregularities ("time lag" and "[q]uantizing"), and policy's "first-order effect on amplification characteristics of the system" (Forrester 1956/2003).

[5]Language for describing the structure and "dynamic behavior of a complex system" (e.g. electrical systems and servomechanisms) is applied to "our economic system." "[C]apacitance or reservoir effects," for example, is equated with "fluctuating inventories" (Forrester 1956/2003).

Second, he would harness advances in simulation. He saw that electronic digital computation made numerical processing extraordinarily cheap, as compared to manual computation—i.e. human computers. But advances were about more than technology, and included advances in "[t]he art of simulation, wherein an analog of a real system is set up and operated at an accelerated time scale." The craft of simulation making had been cultivated largely for the study and design of military equipment—mostly with analog computing tools—and was now ripe for combination with digital computers (Forrester 1956).[6] Forrester would use digital computers to represent economic models as feedback loops.

A First Model

He was soon confronted with the impetus to build a first model. At the Sloan School of Management, Forrester found himself in a chance conversation with managers from General Electric, who were puzzled by what appeared to be fluctuating demand for their household appliances. After interviewing them about their hiring and inventory practices, Forrester sat down and simulated what would now be called their supply chain with pencil and paper, and realized that the system structure itself produced instability—much like the servomechanisms Forrester had worked with earlier. He discovered that the oscillations were not produced by fluctuating consumer demand, but by the "entirely internally determined" decision making of the system itself. What appeared to be an oscillation in demand was in fact an oscillation inherent to the supply chain—the time delayed feedback loops coupling factory production with warehouse, distribution, and retail inventories (Forrester 2007; Lane and Sterman 2011). To return to our

[6]For example, the Link Aviation Company had produced many pilot trainer flight simulators during WWII, and Vannevar Bush employed "[s]tructural analysis by electric circuit analogies" to build scale models of urban power networks which were then used by electrical utilities (Bush 1934; Preston 2003). Of course, makers of analog computers had long used electrical circuits as *analogs* for other phenomena (Small 2013). Note that "analog" is used to mean *an analogous system*.

thermostat example, it was as if the room temperature was oscillating wildly not because of someone capriciously setting the temperature knob, but because the thermostat did a bad job governing the system. It overshot its mark, the air got too hot, and the temperature oscillated uncomfortably.

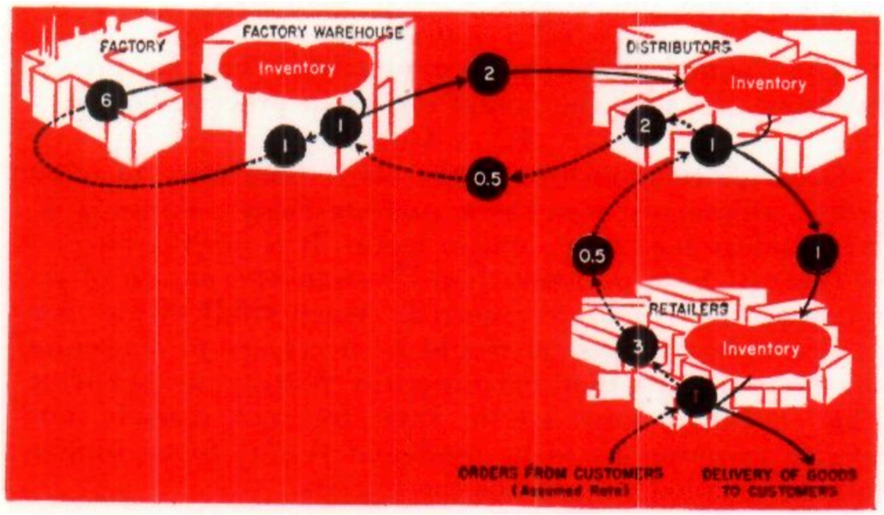


Figure 3.3: Supply chain diagram (Forrester 1958).

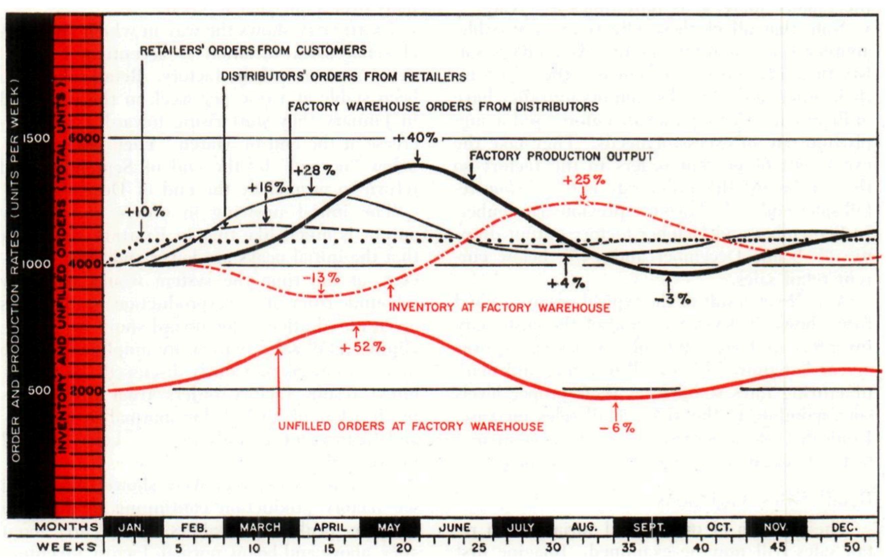


Figure 3.4: Oscillations in a supply chain model (Forrester 1958).

Two years later, in 1958, Forrester published his immodestly titled “Industrial Dynamics:

a major breakthrough for decision makers" in the *Harvard Business Review*, which marked Forrester's first use of electronic digital computation to build an industrial simulation (Forrester 1958).

This publication marks an incremental stage of abstraction in the journey from military servomechanisms to the craft, schemas, and software tools that would constitute system dynamics.[7] The 1958 article documents the first model Forrester built with his nascent simulation approach. He argues that management is transforming from "much more than an art" into an "exciting, dynamic, and intellectually demanding profession." Management's "goals are rooted as deeply in the public interest as the broad objectives of the legal, medical, and engineering professions." This rhetoric gives us a glimpse of Forrester the earnest engineer, in the process of becoming the humanistically engaged scientist who will later focus on the problems of cities and the world. (The earnest rhetoric also makes one wonder if Forrester is out to convince industrial managers or himself.)

Forrester foresaw that computer simulation would eventually transform the role of management. He argued that recently developed tools such as cheap electronic computers, simulation, and an understanding of feedback control systems would transform the project of management, whose task "is to interrelate the flows of information, materials, manpower, money, and capital equipment." (The computer spreadsheet would ultimately fulfill a variation of this vision.) With the aid of visually polished illustrations, Forrester carefully explains how oscillations, instabilities, and amplifications are inherent to the feedback control loops of industrial systems. Simulation is a useful tool, Forrester argues, for understanding these systems, and

[7] It would not be until later that the vocabulary for the concepts of levels and rates, diagrammatic conventions for notating such systems, and the DYNAMO compiler for executing such systems would develop. It wouldn't be until 1961, three years later, with the publication of the book *Industrial Dynamics*, that an abstracted, codified, and reusable set of techniques would be established.

then modifying them to dampen unwanted instabilities. The paper aims to instill in managers a sense for the dynamic properties of industrial systems as a whole, and argues that simulation is a useful tool for transforming management from an art into a profession. Forrester assures us that while "simulation studies will not require undue mathematical ability … details of setting up a model will need to be handled by experts because there are special skills required and pitfalls to be avoided." He envisioned managers working in tandem with simulation experts who would take care of the details. In this vein, the article buries the math. The only mathematical equations are in a side exhibit, and then only a sample of the system internals are exposed.

While working on this paper, Forrester lacked all of the conceptual and representational infrastructure that would come later: a compiler for easily making models, a diagrammatic notation, and the fully abstracted schema of levels and rates. Forrester asked Richard Bennett, who was working for him, to simply program a set of equations Forrester had worked out. Bennett refused, and instead wrote a compiler whimsically entitled SIMPLE, Simulation of Industrial Management Problems with Lots of Equations—the title of which suggests why Bennett opted to build a compiler rather than code all of Forrester's equations by hand. SIMPLE was later extended by Jack Pugh into DYNAMO (Forrester 1989). A domain specific language is much more than a software tool—it simultaneously abstracts a mode of thought, defines a language of expression, and constrains its usage. Forrester credits Bennett's compiler as a turning point, as "it accelerated later modeling that rapidly expanded system dynamics" (ibid.).

Using digital computers to simulate differential equations, a commonplace chore for analog computers, was in itself nothing new (Selfridge 1955; Clancy and Fineberg 1965; Small 2013). What distinguished the DYNAMO compiler from the many other digital languages that simulated continuous parallel processes was an enduring commitment to modeling economic and social processes, rather than purely physical phenomena. This endeavor was sustained by the philosophy, craft, and tools Forrester was marshaling.

The abstraction of Forrester's approach into a domain specific language and reusable software tool (the DYNAMO compiler) accelerated its use by new practitioners and application to new domains. The seed of an infrastructure supporting a particular schema of thought and representation had been planted.

Crystallization of a Technique

Writing about the origin of industrial dynamics, Forrester recalled that while "[t]he elementary idea of feedback as a circular cause-effect phenomena could be traced back through centuries of economic literature," and despite cyberneticists exploration of feedback processes, "the implications, the importance, and the principles of feedback processes were only beginning to be understood" (Forrester 1968).[8]

Forrester was advancing the creation of more than a set of enactive software representations or the DYNAMO compiler. Forrester was establishing a craft of model building, a way of seeing, thinking about, and representing the world. The stage was set for new models to be built, and by an increasing number of people.

In a 1960 lecture, Forrester drew a parallel between the transformation of military decision making by computers, and the transformation of management he now proposed. Just

[8]George Richardson, a future member of the System Dynamics Group at the Sloan School, contextualizes Forrester's approach within the broader stream of feedback based perspectives (including cybernetics), arguing that feedback based thinking has spanned hundreds of years in engineering, mathematics, biology, and the social sciences. This is an effort, it seems, to put system dynamics on a parallel footing to cybernetics. Richardson distinguishes the cybernetic thread from what he calls the "servomechanisms thread," in which Forrester developed a method of computer simulation that could easily handle multiple equations and nonlinear behavior. Richardson argues that Forrester's central contribution lies in his view that one can (and should) derive the dynamics of a system entirely from its internal, endogenous structure. For a more complete intellectual history of feedback thinking, and Forrester's place in it, see (Richardson 1983; Richardson 1991; Richardson 2011).

as computers had augmented the decision making of military men, it could augment the decision making of civilian managers. Drawing inspiration from the technological marvels and advances of wartime, Forrester argued that while the military was initially reluctant to allow machines to substitute for "military training and command experience," the "automatic execution of front-line military judgment" is now commonplace. This was done by "interpret[ing] the 'tactical judgment and experience' of military decision making into formal rules and procedures" (Forrester 1975).

By 1960, Forrester and his collaborators had established a "fundamental concept of systems structure, which realistically characterizes information-feedback systems, be they technical, industrial, or economic systems." This fundamental concept was two entities: *levels* (a.k.a. *state* or *stocks*) and *rates* (a.k.a. *flows* or *valves*).[9] Levels represent an accumulation of something, like inventory or population, that changes over time only as a result of inflows and outflows. Rate equations, in turn, are based upon a system's levels. Each rate is governed by an algebraic equation, and can be independently and efficiently evaluated for any given moment in time. Into this "straightforward model structure," writes Forrester, "we can cast any combination of flows, time delays, amplification, structure, and nonlinear decisions" (Forrester 1975).[10]

---

[9]Word count analysis reveals that Forrester prefers the terms *level* and *rate.* However, *state* and *flow* are also common. *Stock* and *valve* are rare, and *valve* is used exclusively to refer to the visual symbols used to depict rates. Ratios of use are consistent across the articles surveyed (e.g. Forrester 1958; Forrester 1961; Forrester 1969; Forrester 1971/1973).

[10]Forrester would give a more vivid explanation of these concepts in his 1961 book *Industrial Dynamics*, which introduced a visual notation for these systems, a diagram of the equations in motion, and explains the formal language of DYNAMO. We'll return to this book shortly.

How To Do It

In 1961, Forrester published *Industrial Dynamics,* which made public, for the first time, the fundamental concepts of industrial dynamics, a visual notation for diagramming them, the DYNAMO language, and the precise mechanisms of simulation employed (Forrester 1961). *Industrial Dynamics* included the 1958 *Harvard Business Review* "breakthrough" article, but went far beyond it. Rather than sweep the details under the rug, the book was deeply technical, articulating both a philosophy and technical methodology. While the 1958 article addressed itself to managers, and attempted to persuade them of the utility of this new style of analysis—best "handled by experts," *Industrial Dynamics* explained in detail how to be such an expert. The book emerged, in part, from Forrester's practice of creating experts by teaching industrial dynamics at the Sloan school (Forrester 1992). Now people with the inclination and means, situated beyond the Sloan School of Management, could adopt his techniques. *Industrial Dynamics* was a how-to manual.

I will undertake a rather detailed technical exposition here of industrial dynamics. I do this because *SimCity* borrows and reinterprets Forrester's simulation techniques, and it is only by attending to the details that one can see *what* Wright borrowed from Forrester as well as *how.* The simulation techniques Wright borrowed both contribute to and work at cross-purposes to *SimCity's* design, but to appreciate this we must look under the hood, get our hands dirty, and deconstruct the artifice that produces the enactive representations of levels and rates. This close analysis also helps us to explicate the evolution and commitments of Forrester's overall simulation approach. We are fortunate that Forrester's technical exposition is lavishly illustrated, and his language strikingly evocative, so a specialized technical background isn't necessary to understand it. Less technically inclined readers might want to skip the remainder of this section, although I have made an effort to make it broadly comprehensible.

Computer simulation meant that analytic mathematical approaches were no longer necessary, a sentiment later shared by Wright (Forrester 1968, p. 399).[11] But Forrester didn't do away with mathematics, but instead put it to a different type of use. In algebraic equations, say x = 2 + y, if one knows y, then one can solve for the unknown variable x. You solve for y by subtracting 2 from both sides. Given a system of equations—a list of equations sharing variables—it is sometimes possible to find closed form solutions for all the variables (and straightforward equations for computing the unknowns, as in the above example). More complex systems with feedback typically lack such closed form solutions, and cannot be solved in this way. In the 1956 seminar note we looked at earlier, Forrester critiqued traditional economics' reliance upon such analytical techniques as unrealistically eliminating feedback from their models. Forrester wanted to simulate feedback loops, not ignore them. To do this, he proposed a different approach based upon numerical simulation.

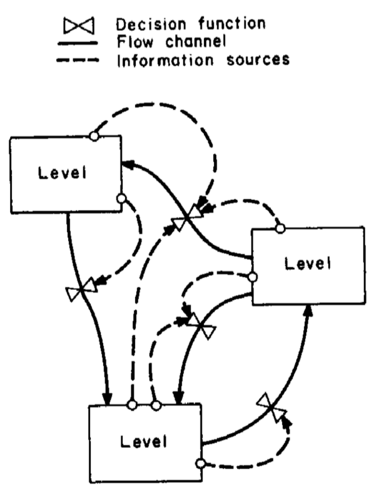


Figure 6-1 Basic model structure.

Figure 3.5: Diagram of a simple model. Boxes are levels. Solid black lines are flows between them. These flows are governed by decision functions indicated by the "valve" symbols (Forrester 1961).

[11]Consider the subtitle of Wright's 2003 talk: "Dynamics for Designers: Why I Hate Calculus… and why I Love Compression" (Wright 2003).

Industrial Dynamics offers a rich and concise visual language for describing the architecture of industrial models. The diagram above is a simple industrial dynamics model illustrating the key model components:[12]

- **Levels.** Boxes indicate buckets full of a certain quantity of stuff (e.g. inventory, money, people).
- **Flows.** Solid arrows indicate a flow of materials between the levels.
- **Rates.** Valve symbols (looks like twinned triangles) regulate the flow of materials across these channels. There is one rate per flow, so rate and flow are practically synonymous. They are labeled *decision functions* in the diagram's key, activating their analogical relation to industrial policy.
- **Information.** Dashed arrows indicate the flow of information from levels to decision functions.[13] This is how the system self-regulates, just as your thermostat does. The

[12]There are additional primitives for conveniently making delays and pipelines, but these are simply elaborations on the primitives above.

[13]Dashed vs. solid arrows indicate information vs. material flows—a critical distinction. Material flows are the movement of things between levels, but the rate of movement is regulated by information flows. *Industrial Dynamics* has a visual notation for four specific material types: orders, money, people, and equipment. Future iterations of Forrester's simulation work would retain the fundamental distinction between material and information, but leave behind this notion of visually differentiating material types.

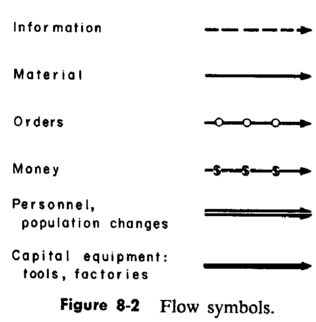


Figure 3.6: Industrial Dynamics offered six visual notations for flows. The most important distinction is between information (top) and material (the rest) (Forrester 1961).

thermostat is an informational arrow that goes from the air temperature *level* to your furnace's on and off switch, or *decision function*. When the furnace is on, it is a *source* of heat that flows into the current temperature *level*.

Such diagrams are structurally complete accounts of a feedback system. The details of how rates respond to their information sources requires a different representation, equations, which we will come to soon. Let's look at a detail of the *World Dynamics* model to get a sense for what a real model might look like.

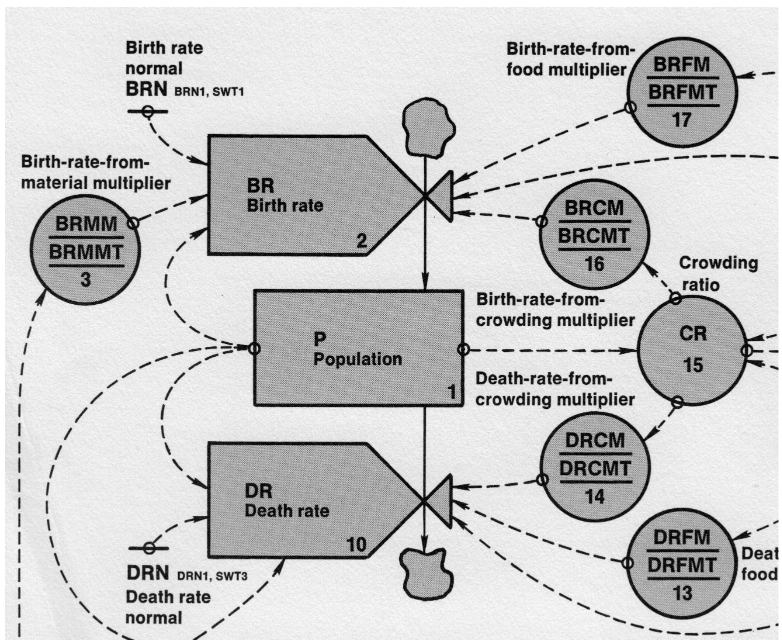


Figure 3.7: Detail of World Dynamics *model diagram (Forrester 1971/1973).*

This figure contains one level in it, population (P), which can be identified by its box shape. Since there is only one level for population, the system does not distinguish between people by race, age, or any other attribute. The apparatus surrounding it determines the inflow (births) and outflow (deaths) of people from the system. The people who move into and out of population (P) do not go to another level. Instead, they go to and come from outside the model—indicated by the source and sink clouds.

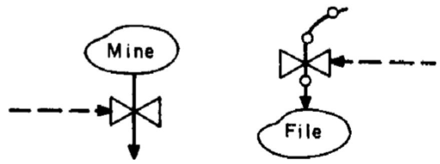


Figure 8-4 Sources and sinks.

Figure 3.8: Sources and sinks (Forrester 1961).

Birth rate (BR) and death rate (DR) are rates which dictate how many people enter and leave the model. The dashed lines pointing into these valves are *information flows* that influence the equations for birth and death rate. For example, factors such as crowding and the current population influence these rates.[14] Rates are visually represented in one of two ways, as the diagram below illustrates.

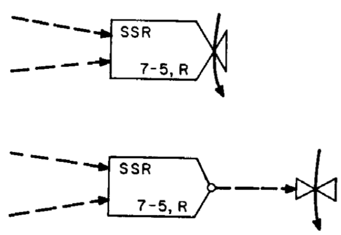


Figure 8-3 Decision functions (rate equations).

Figure 3.9: Two ways to diagram a rate equation (Forrester 1961).

Circles represent auxiliary variables derived from levels. For example, birth-rate-from-food multiplier is a model attribute computed from food and population (among other levels), which then influences the birth rate. It is not a level, but a reusable formula fragment.

[14]The equations for BR and DR detail exactly how their influences mathematically combine, and are described in a formula elsewhere.

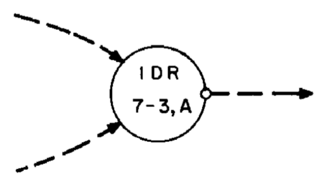


Figure 8-6 Auxiliary variable.

Figure 3.10: Auxiliary variable (Forrester 1961).

Finally, we see constants like birth rate normal (BRN), which are numbers that influence the model, but do not change over time. BRN represents an assumption about what the "normal" birth rate is.

The decision functions and auxiliary variables are defined by straightforward algebraic formulas, an example of which is shown below.

IAR.K = IAR.J + (DT)(SRR.JK − SSR.JK) 7-1, L

The symbols represent variables as follows, given with the dimensions of measurement:

IAR Inventory Actual at Retail (units), "actual" being used to distinguish from "desired" and other inventory concepts
DT Delta Time (weeks), the solution-time interval between evaluations of the set of equations
SRR Shipments Received at Retail (units/week)
SSR Shipments Sent from Retail (units/week)

Figure 3.11: Equation for inventory (Forrester 1961).

Such algebraic declarations constitute a practically complete model description that can run on the computer in conjunction with the DYNAMO compiler.

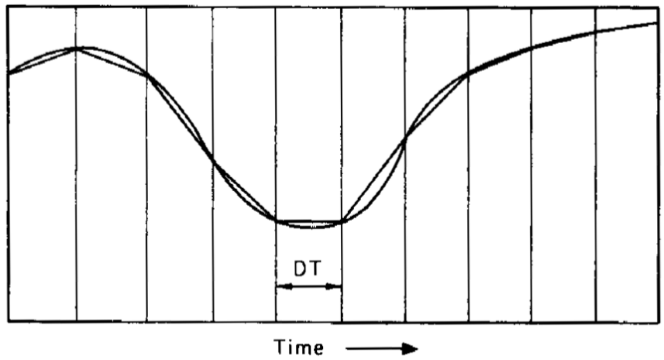


Figure 7-2 Straight-line approximation to a variable level.

Figure 3.12: Piecewise linear approximation of a continuous variable (Forrester 1961).

We will now step down a level of abstraction, and go over precisely how these enactive representations are simulated.

Simulation definitions, when run through DYNAMO, produce a series of still snapshots of the model levels. The graph of oscillating orders and inventory from Forrester's 1958 paper shown shows smooth curves for values changing over time, but those curves are actually generated by straight line approximations from one moment in time to the next.[15] Akin to how a movie produces an illusion of continuity out of a series of still photographs, a series of level values is connected into a continuous line and smoothed out by the hand of a draftsman. The illusion of continuity is furthered by writerly prose. Text and graphics work in conjunction to bring the models to life.

To begin simulation, levels are seeded with a starting state. The simulation then repeatedly steps forward in time, calculating values for the next time step using values from the current time step. Current levels are plugged into the rate equations, producing the rates of change. Adding the rates of change to the levels moves the model to the next time step. The simulator

[15]The method of simulation via discrete time steps is common.

can be thought of as walking forward in time, plotting its next step based upon its current location, or levels. The step size represents how much time has elapsed between these discrete moments in time, and is notated as DT (Δt), delta time.

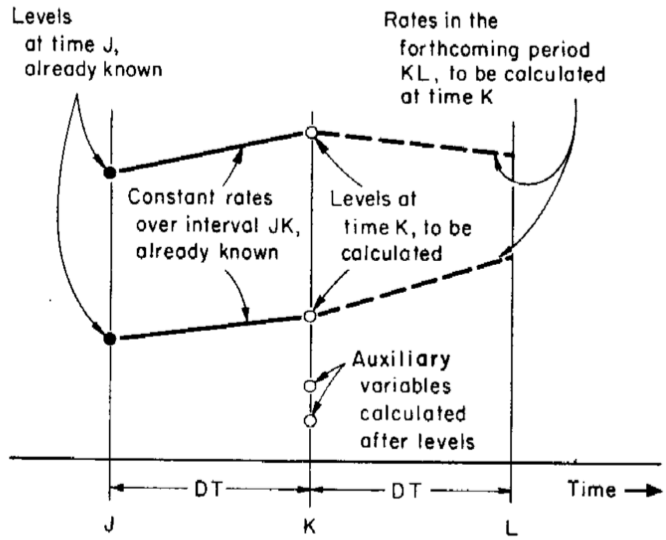


Figure 7-1 Calculations at time K.

Figure 3.13: How a level is simulated forward in time (Forrester 1961).

Discrete time steps means that the simulator is always located at a particular moment in time. J, K, and L refer to the immediate past, present, and future. If we are stepping from J to K, then J is the known past, K is the present state we are entering, and L is even farther in our future. As we step into time K, the levels for time J are all known and can be used to compute values for time K. Rates refer to change over time, as in miles per hour (miles/hours), sales per week, or births per year. Let's look more closely at the formula for the Inventory Actual at Retail (IAR) level shown earlier:

```
IAR.K = IAR.J + (DT)(SRR.JK - SSR.JK)
```

What this formula says, despite the cryptic names, is quite straightforward. To compute inventory at time K (IAR.K), take the known inventory one step in the past at time J (IAR.J), and add the rate of change formula (SRR.JK - SSR.JK) multiplied by the elapsed time (DT) to it. Analogously, if you were standing at mile marker 5 (.K), and are moving at 10 miles per hour (.JK), after 2 hours have elapsed (DT) you will be at mile marker

```
25 = 5 + (2)(10)

Location.Now = Location.Past + (TimeElapsed)(MPH)
```

Alternatively in DYNAMO's notation:

```
Location.K = Location.J + (DT)(MPH.JK)
```

Returning to the inventory equation (IAR), its rate of change isn't miles per hour, but a count of the number of things entering and leaving the inventory. This is the inflow SRR (Shipments Received at Retail) minus the outflow SSR (Shipments Sent from Retail).

The suffix .JK refers to the rate of change between times J and K. Seeing ".JK" helps us to read it as the change which takes us from time J to time K. These suffixes also help to distinguish between levels, whose suffixes indicate a specific moment in time (J, K), and rates, whose suffixes indicate a change between two moments in time (.JK, .KL). This syntax also allows the DYNAMO compiler to check for logical errors in a simulation description (Forrester 1961, p. 369).

Abstracting time steps as DT allows the same simulation to be run at various speeds, so one could explore the evolution of a system at any granularity (e.g. change per week or year). (In practice, simulations designed this way will exhibit strange behavior if time steps become very big.) Also, despite the fact that multiple interwoven feedback loops might be present, the simulator can calculate the levels for the next moment in time very efficiently. This is because the inputs for computing levels at time K only depend on the levels and rates from time J, the previous time step. (At time K, .J and .JK are consumed, and .K and .KL are produced.) In this way, complex feedback loops can be present in a system, but determining each level or rate is a straightforward calculation based upon known data. In theory, this could even afford a degree of parallelized execution. In practice this would never be necessary, as the simulation is a "straightforward mechanization" where "the amount of digital computer time required is not an important consideration in conducting a systems study project" (Forrester 1960/1975, p. 56). The computational efficiency of this technique, even for 1960 era computers, made it ripe for adoption over two decades later by Will Wright. There, Forrester's simulation mechanisms would run on a relatively underpowered microcomputer, in parallel with *SimCity's* many other simulation, graphics, and interface chores.

The Art of Industrial Dynamics

Forrester also taught a particular craft of model making, a way of converting messy social reality into discrete, formal models. Simulation makers, he wrote, practice a kind of "intuitive" " 'art'."[16] They begin by defining "goals and the questions to be answered."

> [T]he factors that bear on the answers must be visualized, interrelated, and described. This should not be a formal step of mere statistical procedure but rather the point where intuition and insight have their greatest opportunity. This is a step for the philosophical, sensitive, perceptive observer.
>
> Here we need experience, alertness, and a strong intuitive feel for the nature of information-feedback systems. We need to look in the proper places for the policies, delays, and information sources that determine dynamic system behavior (Forrester 1961, p. 44).

The simulation designer then comes to an unambiguous verbal description of the model which can then be converted into a formalized, mathematical one which can run on a com-

[16] Forrester puts the word *art* in quotation marks.

puter “that will allow us to experiment with the implications of the statements already made” (Forrester 1961, p. 44).

Deciding what to include and exclude from a model is central to simulation design, and is a highly contingent, almost subjective, procedure. While Forrester doesn’t quite come out and say it, his comments describe a conception of simulation building as highly subjective: an “ ‘art’.”

> In practice there will be no such thing as *the* model of a social system, any more than there is *the* model of an aircraft. An airplane is represented by *several* aerodynamic wind-tunnel models for various purposes, plus cockpit arrangement mock-ups, models for maximum stress loading, etc. In designing a dynamic simulation model of a company or economy, the factors that must be included arise directly from the questions that are to be answered.
>
> …model building cannot be limited to any one narrow classification of intellectual discipline. We must feel free to include technical, legal, managerial, economic, psychological, organizational, monetary, and historical factors.
>
> …Factors that are omitted, both in static analysis and in ordinary descriptive debates about a problem, may prove to be crucial (Forrester 1961, p. 60).

Ultimately, “ ‘art’ is a guide to proper use of the tools of science. For the present we can do no better than to discuss general principles and later supply specific examples that will help the beginner to start developing his own skills” (Forrester 1961, p. 61).

This art and craft relied upon expert knowledge and intuition more than data. Unencumbered by a rigorous methodological adherence to data, and aided by the aggressive abstraction afforded by the generic concepts of levels and rates—easily mapped to quantities from any “intellectual discipline”—Forrester’s simulations could easily be repurposed to represent non-industrial social feedback systems. Just as the abstraction of computation meant that a Viking rocket simulation on Whirlwind could easily become a bouncing ball, Forrester’s servomechanism inspired industrial dynamics could easily be remapped to represent any number of social systems. Industrial dynamics was poised to expand into non-industrial applications and become a generalized schema for modeling any type of feedback system.

The Infrastructure of Industrial Dynamics

With the 1961 publication of *Industrial Dynamics*, Forrester's approach to and philosophy of simulation had crystallized. The simulation practice was refined, embedded within, and perpetuated through a range of infrastructures:[17]

Actors. Echoing the institutional agenda of incubating a new type of business school within an engineering university, Forrester has transferred his servomechanism laboratory background into a management school. He personal background and expertise—feedback systems, digital computers, simulation, and servomechanisms—serve as lenses for understanding economic systems.

Schema. Forrester has settled upon a particular schema for seeing economic systems as interwoven feedback loops. Inspired by the servomechanisms with which he was familiar, a schema of levels and rates is used. Interwoven with feedback loops and time delays, these systems yield instabilities, oscillations, and harbor internal steady states (or resonant frequencies). Model descriptions are closed and formal. The model is closed in that no outside influences are involved, and system behavior results entirely from the simulated elements. Formal description allows for automated computer simulation.

Software Artifact. Special software programs simulate the formal model descriptions. These software tools—the SIMPLE and DYNAMO compilers, with their associated formal model languages—accelerate the implementation of new models, and adoption of the con-

[17] Here I am applying the terms and techniques from chapter 2's framework for analyzing software.

ceptual schema. Simulation involves updating a small number of scalar variables via discrete time steps. This updating is deterministic—randomness, in other words, is not involved. Deconstructing the simulation artifice, we see two layers of enactive representations. First, digital computers are used to simulate servomechanism like structures: levels and rates. Second, the level and rate primitives are assembled, by modelers, into enactive representations of industrial systems. The servo functions as a kind of metaphorical glue that binds electronic computation to social systems, allowing mature tools from engineering—used to study the endogenously produced dynamics of servomechanisms—to be applied towards social processes.

Craft. Forrester advances a particular style of simulation making. He believes that there is an "art of simulation," whereby a simulation designer crafts an "analog of a real system" (Forrester 1956). Background research is done via informal interview, and citations and real world data are spare, if used at all. Parameters are much less important than structure in governing behavior.[18] To achieve a closed and formal analogy of a social system's structure and behavior, aggressive abstraction is necessary. One consequence of this abstraction is cleanly dividing the inside from the outside of the model; what is inside the model must produce the dynamic behavior of interest, not external factors such as fluctuating consumer demand.

Discursive Practices. Simultaneously, the abstract model must be conjoined to the phenomena it represents. To function as an analog, correspondences must be established via discursive practices—annotated graphs, prose descriptions, level and rate diagrams—that bridge the gap between formal model and represented domain.[19] The conventions of a diagrammatic language also help people to think and communicate in terms of the simulation

[18] This will later become clear to others (Edwards 2010), but is an acknowledged premise of Forrester's project.

[19] In scientific publications, descriptions and narrative about experiments are crucial (Latour 1987). There is a tension between narrating system behavior in terms of what it represents, in graphs and text, and exhibiting the mathematical equations which govern its behavior. In this regard, the 1958 paper, by hiding much of the technical detail, is a departure from the tradition of engineering and mathematics.

schema.

Social Practices*.* The Sloan School of Management recruited more participants into Forrester's simulation practice, and spurred the refinement of the technique (Forrester 1992). Equipped with the feedback concepts of levels and rates (schema), knowledge of the SIMPLE/DYNAMO language (craft), and access to a compiler (software artifacts) and computer, non-programmers could now build models. Projects were done by staff (sponsored by the Ford Foundation), in collaboration with industrial organizations (who sponsored them), and students. Companies like Texas Instruments and Kennecott Copper sent emerging managers to year long Master Degree programs at the MIT School of Industrial Management. Some Sloan Fellows, who often had some technical background, used Industrial Dynamics to craft simulations of some aspects of their respective industries (Forrester 1960/1975).

System Dynamics

From Supply Chains to Cities

A key aid to the growth of industrial dynamics was the Sloan school itself, which pushed Forrester to expand the simulation practice to new domains and participants. The Sloan school brought Forrester into contact with Sloan fellows, emerging managers from U.S. industry, some of whom would develop simulation models under Forrester's guidance. The result of all this teaching and model making was that the simulation practice was expanded to new applications, and refined even further.

Another fundamental benefit of the Sloan school was the placement of an engineer, Forrester, within a management school where interdisciplinary connections between engineering and business would be made. The founding objective of the Sloan school, according to Forrester, was to cultivate a new type of management school steeped in a technical environment. In Forrester's case, at least, this arrangement admirably met its goals. Forrester's placement in the Sloan school initiated his formal concern for management as a science, and made possible the chance meeting with managers from General Electric which sparked his first pencil and paper industrial simulation. It also gave him access to the experts he relied upon to craft new models. The genesis and growth of industrial dynamics was a direct consequence of this institutional arrangement. Forrester's next simulation, a model of cities, would similarly be inspired by another fortuitous interdisciplinary connection catalyzed by the Sloan school. The leap from the study of industrial to urban systems was made possible by the increasingly refined software tools, schema, and craft of industrial dynamics, as well as the catalytic social context of the Sloan school.

In January 1968, John F. Collins completed his second four year term as mayor of Boston, chose not to run again, and began what would become a thirteen year visiting and consulting professorship at MIT. Collins was regarded as a progressive mayor who oversaw "a massive urban redevelopment program, paralleling a rejuvenation in business and city government" ("John F. Collins, former mayor and MIT professor, dies at 76" 1995). Collins was initially given a year long appointment in Urban Affairs as a visiting professor. A victim of the 1950's polio epidemic, Collins needed an easily accessible office. Forrester's building qualified, and Collins was given an office adjacent to Forrester, as its usual occupant was away on sabbatical (Forrester 1989).

Forrester's attention was drawn to what Collins called "the crisis of the cities, the greatest domestic crisis to challenge America in a century" (Forrester 1969, p. vii). In talking to Collins, Forrester "developed the same feeling" as when he spoke to corporate executives, "an uneasy sense that something was wrong or incomplete." Forrester suggested they pool their respective backgrounds in cities and in modeling, to which Collins agreed.

> I told him we would need advisers who knew a great deal about cities from personal experience, not those whose knowledge came only from study and reading. We needed people who had struggled with cities, worked in them, and knew what really happens. … Collins listened and said, "They'll be here on Wednesday afternoon." Collins' position in Boston at that time was such that he could call up almost anybody in politics or business, ask for their Wednesday afternoons for a year, and get them. He delivered the people and it was out of the following discussions that *Urban Dynamics* developed (Forrester 1989, p. 7).

Forrester came to these conversations "knowing the conceptual nature of the structure being sought" (the schema of industrial dynamics), into which the "specific details of the structure" gleaned from expert interviews would be "fitted" (Forrester 1969, p. ix). The conceptual structure into which this expert knowledge of cities would be fitted, of course, was the servo inspired schema of levels and rates. Cities would be artfully and aggressively abstracted from life in such a way that captured their essential feedback structure. The models would be unencumbered by a rigorous adherence to quantitative data. Reliance upon expert interview, qualitative research, and industrial dynamics explains why *Urban Dynamics* includes a mere six citations, five of which were of Forrester's own work, and one to psychologist Kurt Lewin.[20] Forrester's comment about "knowing the conceptual structure" also reveals the degree to which industrial dynamics constituted a specific modeling schema into which many domains could be "fitted." This process transmuted an "urban area" into "a system of interacting industries, housing, and people" (Forrester 1969, p. 1)—the "components of the urban system" (ibid., p. ix).

This shift from the purely material to the social was marked by a slight simplification of

[20]The single non-Forrester citation was to *Field Theory in Social Science* (Lewin 1951), by the psychologist Kurt Lewin, who attended the Macy conferences on cybernetics (Richardson 1983, p. 16).

the modeling structure, and a more concise explanation of the simulation approach.[21] Psychological quantities such as "perception" entered the models. Part of what made Forrester receptive to a leap from the physical (inventories and sales) to the social came from his experience on the board of the Digital Equipment Corporation (DEC), the outfit which manufactured the popular PDP minicomputers which descended from Whirlwind and the TX-0, and whose founders he had worked with on Whirlwind. As a board member of DEC, Forrester grappled with his limited understanding of "high-technology growth companies" which led him to build models of them that

> moved system dynamics out of physical variables like inventory into much more subtle considerations: the top management influence structure, leadership qualities, the characters of the founders, how goals are set the interactions between capacity, price, quality, and delivery delay, and how an organization's traditions determine its decision making and its future (Forrester 1995).

Modeling DEC's non-physical variables inspired the tentative first step into social systems. This step was enabled by the high altitude of abstraction inherent to industrial dynamics. *Urban Dynamics,* for the first time, clearly staked out applications beyond economics, a possibility latent in the philosophy, modeling ontology, tools, and practice all along.[22] The idea of a fully generalized *system dynamics* blossomed out of this departure (Forrester 1969), eventually becoming the focus of a robust community of practice: conferences, the Systems Dynamics Society, more software tools (e.g. STELLA), and a journal, *Dynamica* (1975), which later turned into the *System Dynamics Review* (1985).

---

[21] Diagrammatic conventions for indicating the type of material flows (orders, money, personnel, capital equipment) and their pipelining via "boxcar trains" became deemphasized.

[22] "We must feel free to include technical, legal, managerial, economic, psychological, organizational, monetary, and historical factors. … Economic and industrial activities are closed-loop, information-feedback systems. … This is a broad definition covering most human, social, and technical activities" (Forrester 1961, p. 61)

Urban Dynamics

Forrester was keenly aware that there is no single model of a system, and that an overarching set of questions must guide the development of a particular model. In the case of cities the question was: what are the underlying causes of urban growth and decay? The simulation model, as Forrester reiterates throughout the book, constitutes a particular theory of urban behavior. The model *is* a theory. Forrester conducted experiments with the model, trying out different urban policies (taxation policies, urban renewal, subsidized low income housing, etc…), in an attempt to determine their effects. He then interpreted the model behavior, narrating the hard to see causality within. For example:

> Figure 4-8a shows the effect of introducing such a low-cost-housing program at year 0. The housing available for the unemployed begins to rise immediately. Because of the increased housing, more underemployed are attracted to the city and the underemployed population rises for the first 10 years. But the low-cost-housing program exerts continuous pressure on the available unfilled land, making the area less favorable for other types of construction (Forrester 1969, p. 67).

In parallel with the repeated warnings that the model is a theory, a slippage, perhaps inescapable, occurs throughout the book. Forrester speaks from a position of immersion within his own make believe model. As the preceding excerpt demonstrates, the graphs and the textual narration read as if they are describing a real city, confidently describing its causal flows. The contingent and representational nature of the modeling act is elided.

The social world, Forrester argued, was hard to understand and control because of its complexity. Complex systems, according to Forrester, are structures with "high-order, multiple-loop, nonlinear feedback," to which "[a]ll social systems belong."[23] Such systems are coun-

[23] Forrester doesn't clearly articulate the relation between system dynamic models and complex systems, though one can easily imagine examples of complex system models, such as cellular automata, which are not system dynamic models.

terintuitive, and our understanding of them naive (Forrester 1969).[24] According to Forrester, "the intuitive solution to the problems of complex social systems will be wrong most of the time" (Stevens 1969). System dynamics was an antidote. Model building and analysis could attune decision makers to the critical points of intervention, heightening their ability to address the underlying structural causes of the urban crisis specifically, and social ills more broadly (Forrester 1969).

We turn, now, to the assumptions and mechanisms of Forrester's urban simulation. Unlike *Industrial Dynamics*, which described a methodology for simulating industrial systems, *Urban Dynamics* contains a particular simulation model. I give special attention to the ideas that will later be appropriated or challenged by Will Wright, to help us see the direct connections between urban dynamics and *SimCity*. In order to do this, the next section goes into some technical detail. Such a close reading will bear fruit when we later take up *SimCity*.

[24]Cause and effect are no longer "closely related in time and space," as they are in our everyday lived experience. Our habituated practices of perceiving causal relationships is frustrated. Causality is hard to determine, often leading to a treatment of symptoms rather than causes. There is, for example, a "conflict between short-term and long-term system responses." Matters often get worse before they get better (or the reverse), which often leads to a pursuit of policies with short term benefits that hide long term costs. Complex systems are also insensitive to parameter changes (numerical values), as their behavior is determined by their underlying structure. "In fact, social systems are dominated by natural and psychological factors that change very little." Policy changes that result in intended consequences are difficult to find, as such systems often push back against change via homeostatic tendencies. Complex systems, however, are sensitive to the right changes in policy and parameters, so the key is in discovering exactly how to intervene (Forrester 1969).

Structure of an Urban Area

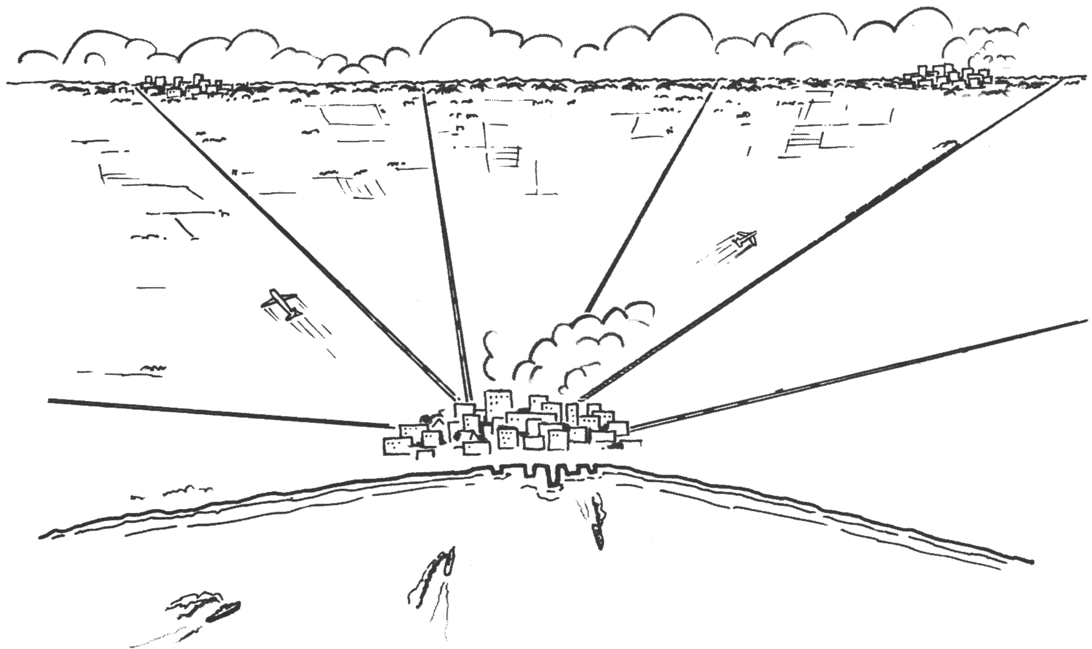


Figure 3.14: "The urban area in its limitless environment." In this illustration as well as his many diagrams, Forrester reveals himself to be a highly visual and spatial thinker (Forrester 1969).

Forrester includes a rare figurative illustration in *Urban Dynamics.* The only other example I know of is in his original article on industrial dynamics (Forrester 1958). These vivid illustrations help Forrester cast the spell of make believe upon himself and his readers. Their appearance, after all, mark the first time Forrester published models in new domains: first supply chains, and then cities. And their stylization, a kind of diagrammatic abstraction, reflects the system dynamics perspective. The city has been reduced to a small self-contained microworld that transacts with its environment via clearly defined avenues and vehicles. The external world continues beyond the horizon, outside the frame of the illustration. The simulation is about the microworld in the middle. The minutia of individual citizens, as well as the outside environment, have been abstracted away (Richardson 2011).

Urban Dynamics undertakes the representation of an entire class of systems. The model is not of Boston, San Francisco, the Lower East Side, or of any city or neighborhood in particular, but urban areas in general. Like all of Forrester's simulations, the model is closed; the only feedback loops which influence its evolution are inside the city.[25] But the closed urban area is part of a "limitless environment." Materials and people flow in and out as needed, but this is an open loop—someone who has left cannot affect the urban area's evolution. They have disappeared into one of the diagrammatic cloud symbols indicating an externality. Forrester's drawing of a city evokes the idea of a closed system interacting with its environment.

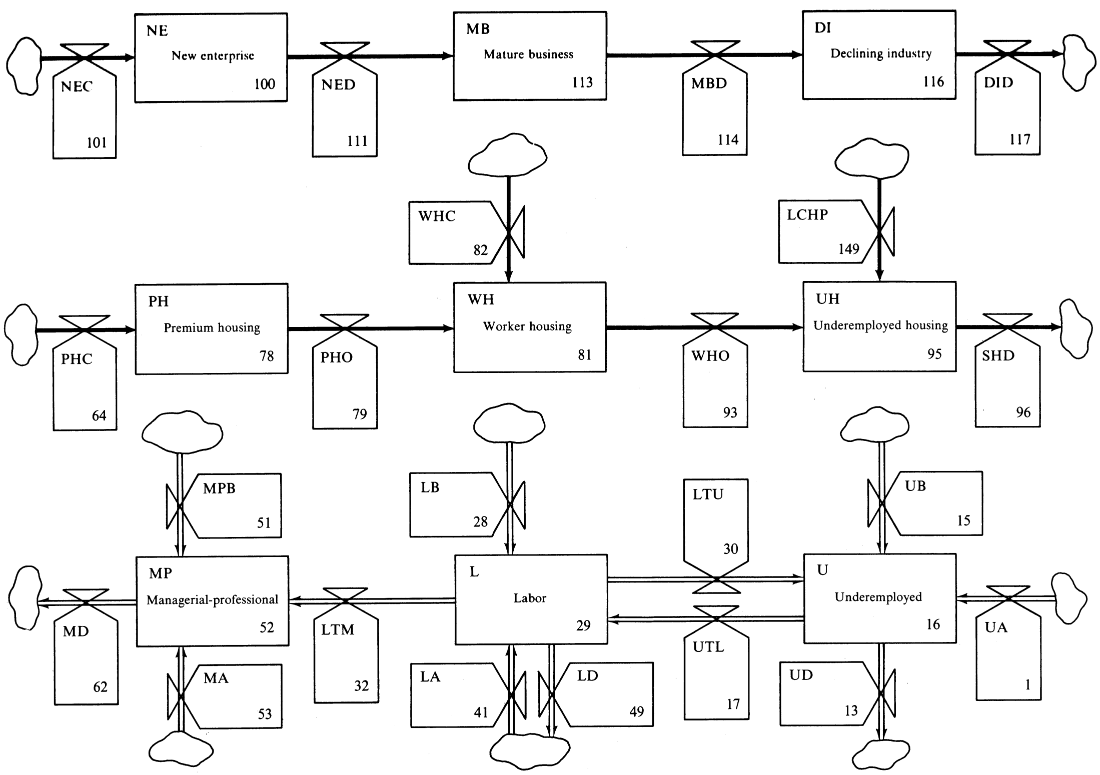


Figure 2-4 The major levels (rectangles) and rates (valve symbols) for the model of an urban area.

Figure 3.15: Overview of the Urban Dynamics *simulation model (Forrester 1969).*

[25]As in all his models, a closed and clearly defined boundary separates the inside of the system from the outside. Forrester begins, in fact, by defining the extent of the model. A tenet of system dynamics—going back to his model of the apparent fluctuating demand for General Electric appliances—is that the model structure is sufficient to generate the dynamics of interest.

Since the model was motivated by the question of growth and decay, it can simulate the development of a mature urban area from an empty landscape. Development emerges from the model shown in the diagram above. The diagram captures the system's major levels ("rectangles") and rates ("valve symbols"). The cloud shapes represent externalities (often the limitless environment) and signify, as in industrial dynamics, the flow of materials into and out of the system. There are three types of levels the system is concerned with: business, housing, and population.[26] These chunks of interconnected material flows are arranged in rows.

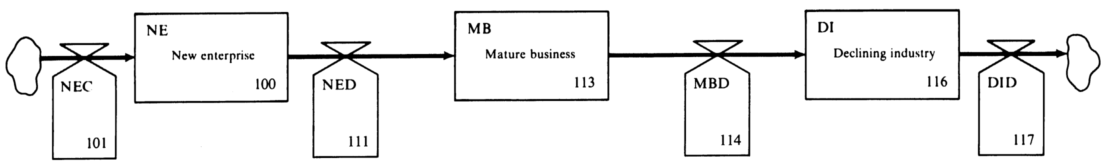


Figure 3.16: Business sector. Detail of Figure 2-4 from (Forrester 1969).

The top row models the lifecycle of businesses, which can be in one of three states: new, mature, or declining. Each of these levels represents the number of businesses in that state. Business flow from left to right in the model. They are created, mature, decline, and finally die—always in that order, moving from left to right across the diagram.

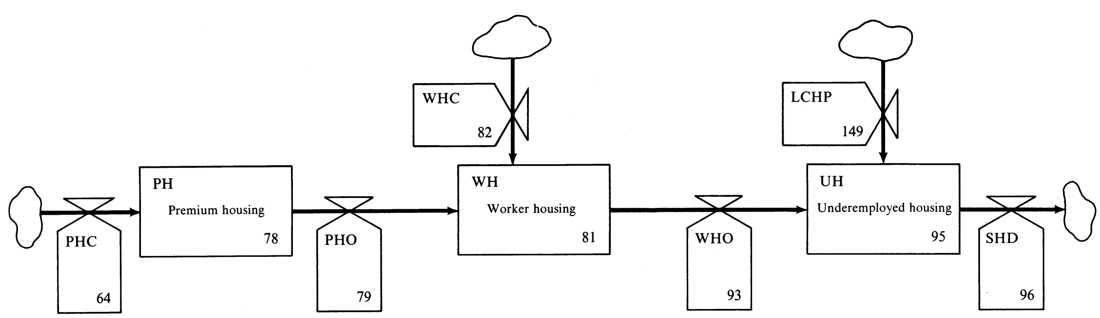


Figure 3.17: Housing sector. Detail of Figure 2-4 from (Forrester 1969).

Housing, the middle row, is a variant of this basic structure. Housing is represented by

[26]*SimCity* also models its populations as three types: residential, commercial, and industrial. But in *SimCity* population is directly tied to housing, and there are two types of business: commercial and industrial.

three levels: premium, worker, and underemployed. (These correspond to the labor types we will come to next.) The life cycle of housing is much like that of businesses. It flows from left to right over time, degrading from premium to worker to underemployed. Unlike businesses, each of which must journey across a complete lifecycle (creation, maturation, and decline), worker and underemployed housing can be built directly. This is what the inflow valves WHC (Worker Housing Construction) and LCHP (Low Cost Housing Program) represent. Housing leaves the model when it is destroyed by SHD (Slum Housing Demolition).

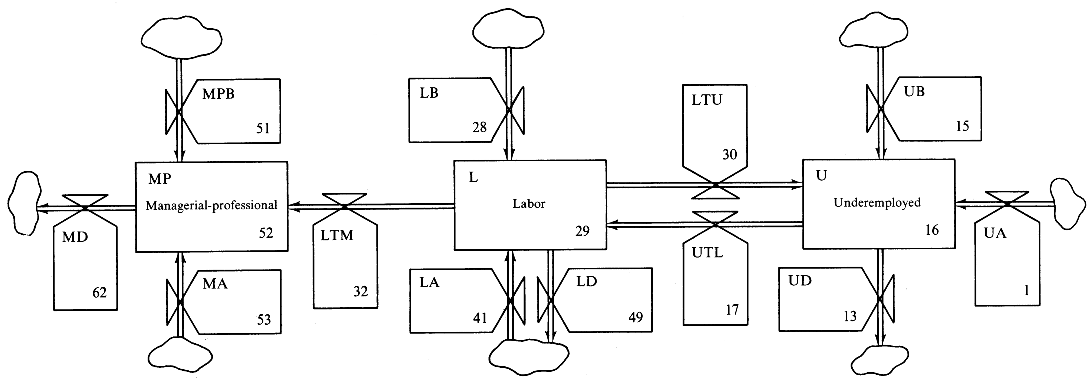


Figure 3.18: Labor sector. Detail of Figure 2-4 from (Forrester 1969).

The bottom row is the most complex, and represents three population classes: managerial-professional, labor, and underemployed. Every unit of population represents an entire family (implying single breadwinner families only), and different population classes have different sized families (ibid., p. 204). These levels change as a result of three factors: net births (includes deaths), migration, and class mobility. Each population level has a birth valve on top of it (MPB, LB, UB), representing the birth rate of that population class. A certain amount of class mobility is possible. Depending on the economic circumstances, labor and the underemployed can convert from one to the other. Labor is sometimes promoted to managerial-professional, but managerial-professional never becomes labor. The remaining inflows and outflows represent migration into and out of the city.

Migration is governed by the idea of relative attractiveness, a mechanism that is very important to *SimCity*. If the urban area is more attractive than the "limitless environment" to a particular type of population, then they move in; otherwise they leave.

This is not a complete diagram of the city; a hidden labyrinthian network of information flows connects these levels and rates to one another, guided by a host of intermediate variables and constants. All of these are precisely explained in an appendix to *Urban Dynamics*, which is a remarkably well organized combination of cross-referenced diagrams, formulas, and textual explanations (ibid., pp. 133-217). The overall system is divided into sectors (roughly corresponding to each level in the main diagram), many of which reuse intermediate variables and constants. I will now treat a few of these sectors in more detail, to give a better sense of how the simulation actually works. I choose to focus on simulation machinery that will reappear, in one form or another, in *SimCity*.

Managerial-professional population sector

Let's zoom into a detail of the above diagram: the managerial-professional sector. While each population sector is unique, they share many structural similarities, so this one is representative of the kinds of dynamics at work in each.

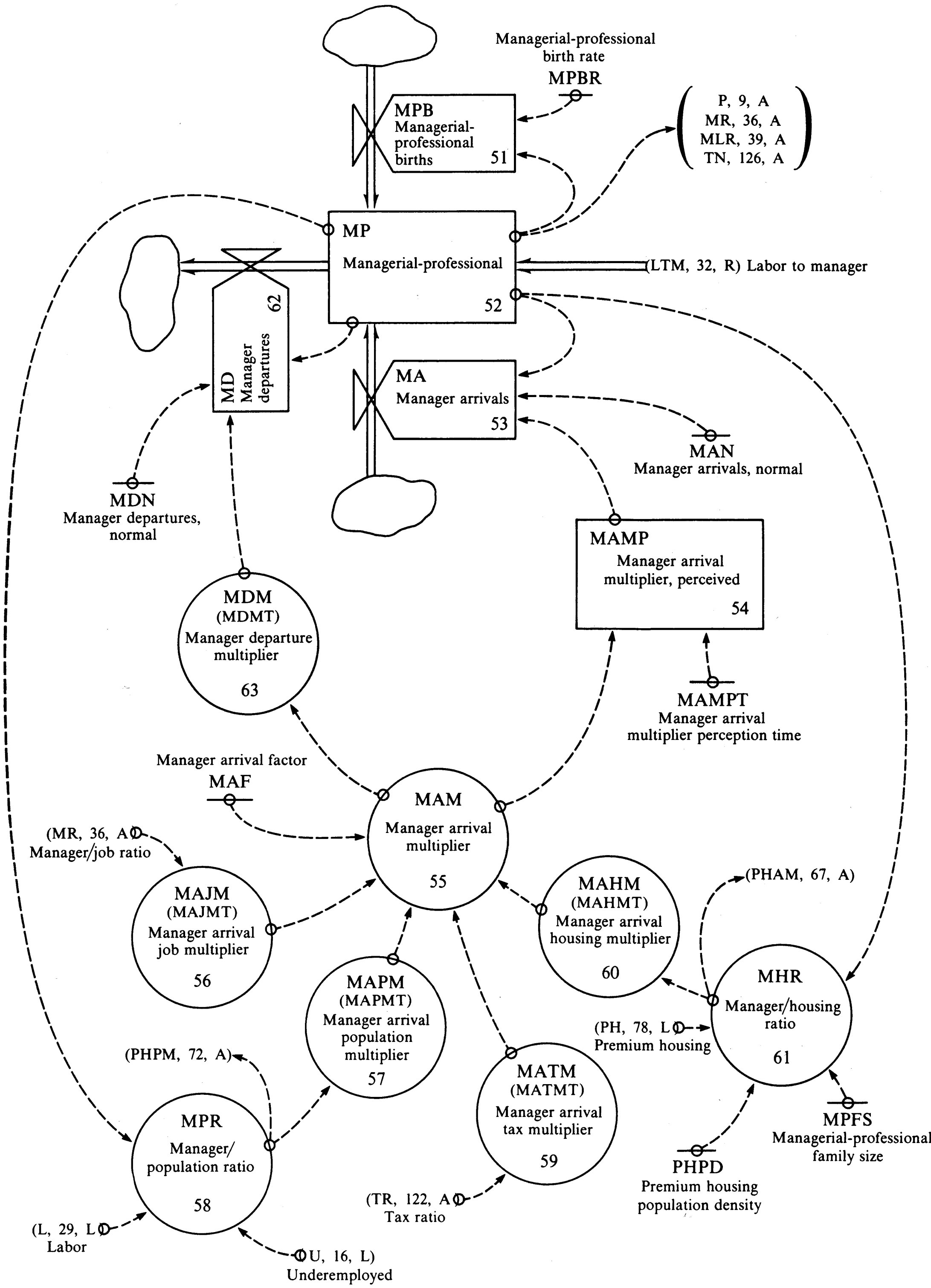


Figure 3.19: Managerial-professional sector (Forrester 1969).

Each population sector has its own birth rate which encapsulates the net of births and deaths. Managerial-professional birth rate (MPB) is a function of both a constant birth rate (MPBR) and the current population (MP). The constant managerial-professional birth rate is .75% per year, so births per year are calculated by multiplying .0075 times the current population.[27]

In addition to the birth rate, managerial-professional population changes in response to migration, and the promotion of labor to management. Migration is directed by the manager arrival multiplier (MAM), which effects the idea of relative attractiveness by aggregating a host of influences from the tangle of values at bottom. While departures are immediately affected by the area's relative attractiveness to the outside world, arrivals are subject to a perceptual time delay (MAMP). A ten year lag time delays the city's attractiveness to managers (it is even longer for other population groups). This means that if a manager doesn't like the urban area, then he or she will leave right away. But moving in is different; managers not living in the area respond to its attractiveness ten years in the past.

Many factors influence the manager arrival multiplier (MAM). One influence is the tax ratio (TR). As taxes go up, the urban area becomes less attractive. Tax rates generally have such negative effects throughout the system. In general, as taxes go up, construction and positive migration decrease (ibid., pp. 176, 168, and 171). Positive effects resulting from taxes are not common in Forrester's urban model, and include an increased migration of the underemployed into the city (taking advantage of greater per capita government expenditures), and greater labor to manager mobility (ibid., pp. 155 and 141).

Jobs and housing also contribute to the area's attractiveness. Job attractiveness (MAJM) results from the ratio of managers to managerial jobs (MAJM), and housing attractiveness

[27] Underemployed birth rates are twice this amount, 1.5%, and labor birth rates are 1%.

(MAHM) results from the ratio of managers to premium housing (MHR). These effects are non-linear, and exaggerate negative and positive attractiveness.

Finally, the population mix of the area affects its desirability to managerial-professionals. If the population ratio of managers goes up, the area becomes slightly more attractive to this population class. This constitutes a positive feedback loop: more managerial-professionals means the urban area is more attractive to them, while a dearth of managerial-professionals makes it less attractive to them.

Managerial-professional housing sector

Next, let's look at housing. Again, we'll take the premium housing sector as a representative example of how housing works for all population types. (I don't include the diagram this time around (ibid., p. 171), but it has a similar structure to the last one.)

Recall that premium housing is built at a certain rate, and obsolesces (turns into worker housing) at a certain rate. These rates are governed by a single premium housing multiplier which represents, in a sense, the area's *desire* for premium housing.[28] Such multiplier factors guide goal seeking behavior across the simulation, driving housing construction/obsolescence, population arrivals/departures, and businesses growth/decay. Like a thermostat servomechanism, they drive system variables towards desired states. If the premium housing multiplier goes down, indicating less demand for managerial housing, then less housing is built, and the excess premium housing is obsolesced into worker housing more rapidly.

The premium housing multiplier collates influences from throughout the simulation. One powerful influence upon the premium housing multiplier is the ratio of managers to managerial housing, which tells of an over- or under- supply of premium housing. Real estate developer

[28]These variables are labelled *multipliers* because they shrink or grow, through multiplication, baseline rates.

foresight is represented through a sensitivity to the same factors that influence managerial-professional migration. If it looks like more managerial-professionals will migrate in, then more premium housing is built.[29] Construction is also subject to speculation that carries forward in time the momentum of recent construction. Time delay and averaging are used to produce a positive feedback loop effect that represents speculation.[30] Taxes negatively influence premium housing construction.

Urban development is responsive to the area's overall economic vitality, which is dictated by the area's maturity. Maturity, in turn, is indicated by the land fraction occupied (LFO). Since system dynamics is disinterested in spatial quantities, space is modeled indirectly, as the percentage of land occupied. Urban land use is a zero sum game in which all housing and business types compete for available land. Land fraction occupied (LFO) is the ratio of occupied land to total land.[31] As the city fills up, there is less room for new construction (Forrester 1969, p. 174). In this way, cities can be somewhere between empty and full, with consequences that reverberate throughout the system.

[29] To accomplish this, supply is influenced by the area's job and housing desirability to managerial-professionals, the ratio of managers to the overall population (a self-reinforcing trend), and the growth of new enterprises that yield more managerial jobs (ibid., p. 157).

[30] This effect is produced by tracking a historical premium housing level, computed by averaging the premium housing level over time. By subtracting this moving historical average from the current premium housing level, a momentum is calculated, which then accelerates the construction of new premium housing (ibid., p. 177).

[31] Occupied space is the sum of housing and businesses, where each unit of housing is half the size of a unit of business.

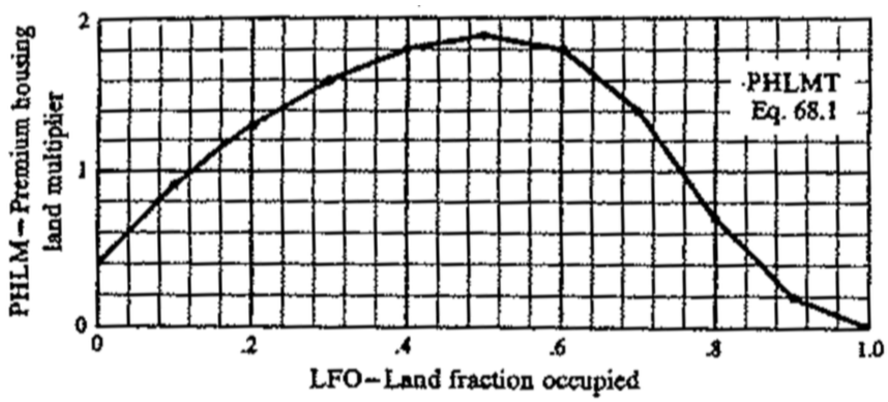


Figure 3.20: Urban Dynamics *makes extensive use of such graphs—encoded as a sequence of numbers—to represent theories about how variables influence one another. Here we see how Land fraction occupied (LFO) affects Premium housing land multiplier (PHLM) (Forrester 1969).*

As time goes on, the urban area tends to fill up with houses and businesses, so lack of space (LFO) is a proxy for the area's maturity. The premium housing multiplier responds to the city's maturity. Young cities grow quickly, and when a city's land area is 50% full, its growth rate peaks. Nascent (very empty cities) and old (relatively full cities) grow slowly. Completely full cities don't grow at all. The graph above encodes this behavior, along with the corresponding hypothesis that economic growth is linked to density (ibid., pp. 173-174, 182, and 193). This graph also conveniently enforces the constraint of limited space. As LFO approaches 100%, no empty space remains for construction, so construction multipliers are driven to zero. Simultaneously, rates of destruction increase to make room for new growth. As cities move towards 95% fullness, destruction of businesses and slums (underemployed housing) skyrockets (ibid., pp. 201 and 189).

Business sectors

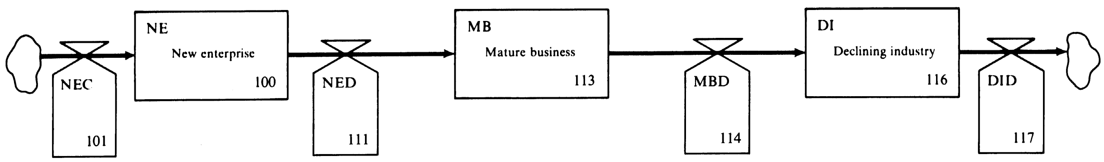


Figure 3.21: Business sector (Forrester 1969).

The business sector, like housing, is a pipeline. New enterprises are created, mature, decline, and are then destroyed. In the diagram above, businesses flow from left to right. A single factor, the enterprise multiplier (EM), is the underlying catalyst for movement through this pipeline. EM encourages the growth, maturation, decline, and destruction of businesses.[32] This multiplier is influenced by a variety of factors. Empty space encourages growth; half full cities grow very fast, and full cities have no room to grow. Growth is also stimulated by underemployed labor and managerial-professionals. High taxes, on the other hand, squelch business growth. All of these factors are multiplied together to form EM, and subjected to a speculative momentum.

Businesses create jobs. But not all businesses and jobs are the same; new enterprises, mature business, and declining industry each offer a different mix of managerial and labor jobs.[33] Jobs influence the area's attractiveness. If more jobs are available for a particular population class, then the urban area becomes more appealing to them. New construction creates jobs for the labor class, and a surplus of construction labor makes development cheap,

[32]New enterprise creation (NEC) is also influenced by a handful of factors in addition to the enterprise multiplier (EM). First, a desired level of new enterprise construction is computed from the enterprise multiplier, a positive feedback from existing industry (new, mature, and declining industries stimulate new enterprise growth), and any new enterprise programs in place (ibid., p. 190). This desired new enterprise multiplier is then scaled by the available labor for construction (underemployed labor means more labor for construction) to determine the new enterprise creation rate (NEC).

[33]As businesses mature, they require fewer managers and more laborers. Job availability also shifts according to demographics. Underemployed jobs become available as labor jobs go unfilled by a lack of labor, as well as underemployed job programs.

stimulating new construction.[34]

Taxes are automatically set by a simulated social-political model. Taxes try to keep up with expenditures, and high taxes retard growth.[35] A larger mix of the underemployed exaggerates taxes needed, reflecting the theory that the underemployed exert political influence to spend more taxes on public services for them (ibid., 202). Forrester's model posits that poverty leads to higher taxes, which then slows economic activity.[36]

Summing Up

With *Urban Dynamics,* Forrester built a software representation of the urban crisis. He did this by translating the processes of urban growth and decay into the enactive representations of system dynamics. Levels and rates were used to construct complexly interlinked feedback loops representing the interactions of business, housing, and population. Goal seeking behaviors react to these levels, slowing and accelerating processes such as new businesses creation, migration, construction, destruction, and taxation. Many of the goal seeking behaviors respond to ratios between quantities, such as taxes needed/assessed, housing

[34]Construction labor requirements reflect what is being built. New enterprises require 20 laborers per year, premium housing 2, worker housing 1, and low cost housing .6 (ibid, p. 206).

The availability of labor for construction generally follows the same shape as the labor employment rate, and influences the rate of new construction projects (ibid., pp. 157 and 207-210).

[35]The urban area's tax ratio (TR), which has mostly negative effects on growth in the model, evolves in response to the city's total assessed value (AV) and taxes needed (TN). If the area's tax needs surpass the amount of tax collected, then the tax ratio goes up; the inverse is also true. An urban area's assessed value (AV) reflects the value of housing and business, which have a fixed tax value per housing and business type. The urban area has certain costs, encapsulated as taxes needed (TN), which reflect the costs incurred by family members of all population classes and the relative voting power of the underemployed. This is subject to a lag time (a *perceived* tax ratio needed is produced by averaging in the tax ratio needed), so the final tax ratio will chase after the relation between AV and TN.

[36]The underemployed, because they are modeled as having larger families and more voting power, exert strong upward pressure on the tax rate. Recall that every population unit represents a breadwinner for a family, and that different population classes have different family sizes. In addition, a family member of each population class entails a certain tax cost (poorer population classes require more taxes). A given population class's taxes needed are computed by multiplying its population level, family size, and cost per person.

needed/available, and jobs desired/available.

The modeling paradigm reflects a commitment to feedback loops simulated on digital computers, as opposed to the closed form analytic solutions commonly used in economic models. The modeling mechanisms and terminology used, such as feedback loops, time delays, and goal seeking behavior, reflect a historical connection to and inspiration from servomechanisms, and their attendant dynamics, such as amplification, speculation, and smoothing.

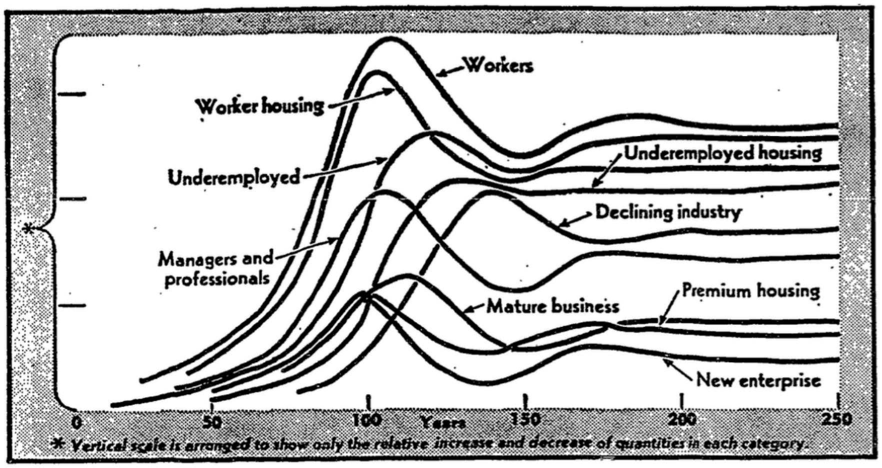
System state is encoded in a relatively small number of levels representing both concrete (e.g. population and housing) and abstract (e.g. "perceived" desirability) social quantities. These levels are statistical aggregates only, as the model does not care about the location, gender, or color of individual people, houses, or businesses. The small number of quantities (for a computer), in conjunction with the DYNAMO simulation architecture, make for a highly efficient computer simulation.

Population movement into and out of the city is central to the model, and is driven by the notion of relative attractiveness—how much more attractive the urban area is compared to the surrounding environment. Populations come in three varieties of wealth/skill: managerial-professional, labor, and underemployed. This tripartite classification is complemented by three corresponding housing types. These population classes also loosely correspond to businesses types—new, mature, and declining—which offer different quantities of jobs to the population types. Housing and businesses constitute a three stage wealth pipeline, as they are created by and for the wealthiest population. As they decay over time, they employ and house the labor class. Housing eventually decays into slums for the underemployed, and is eventually scrapped, alongside declining industry.

Space is modeled indirectly, as the percent of space occupied by housing and businesses (LFO). The model doesn't keep track of where these buildings lie, only how much space they take up, and how much space is available in the limited area of the urban area. Growth and decay is influenced by the amount of space available. Mostly empty cities grow slowly, and as space fills in, grow more quickly. (In fact, most growth rates in the model work like compounding interest.) As the area gets full, new development slows, and pressure is put upon available space, and the decay and scrapping of older housing and businesses accelerates. In short, space is modeled indirectly, as a statistically aggregate quantity that limits and shapes the growth, decay, and regeneration of the urban area. Space is a fixed numerical resource that is allocated towards an ever changing mix of housing or business. Housing and business, in turn, shape population levels by offering jobs and homes.

Forrester's urban model is an endogenous system. It is a closed and formally described model that accounts for the growth, decay, and policy relations which interested him. The system is closed, both as a closed loop simulation, and as body of work, as it has few references to outside texts (or references to external data sets). *Urban Dynamics* offers a complete account of Forrester's simulation philosophy, his theory of urban dynamics, and the detailed mechanics of the simulation algorithm, equations, and constants used. One could reproduce the model entirely based upon the description in *Urban Dynamics*. Forrester explains how his enactive representations work, how to use them, and immerses us into his make believe city simulation. These qualities made the system ripe for adoption, and are the culmination of the ongoing expansion and refinement of Forrester's system dynamics simulation craft, software tools, classes, and publications.

The model depends on both computation and specialized interpretive strategies to bring the simulated city to life. The model is highly abstract, and relies upon a set of discursive strategies employed by Forrester for its representational legibility. Forrester guides and inspires make believe participation in his models. Abstract levels and rates are endowed with social and material meanings through descriptive labels (e.g. “premium housing”), and the text and figures of *Urban Dynamics* that narrate the model’s assumptions, design, and behavior.



The New York Times Oct. 31, 1969
The Forrester urban model involves mathematical equations stored in a computer. He inserted in the equations data representing the condition of a hypothetical city at its birth. The computer then produced the above chart showing the city’s growth and stagnation over a 250-year period. Mr. Forrester says the general long-term relationships hold despite short-term fluctuations caused by technological innovations and world events.

Figure 3.22: Graph and caption from a New York Times *article about Urban Dynamics (Stevens 1969).*

A tension exists between the model’s superficially readable behavior and the opaque causal flows that generate it. The reader is dependent upon Forrester’s careful explanations to perceive the system’s complex causal flows, and to see the complex web of interrelationships that constitute its design.

A Tale of Two Cities: Space and Race

Notably absent from Forrester’s model and its associated text is much discussion of race and space. Implicit in the discussion of an urban crisis is the question of race and politics in the United States, but *Urban Dynamics* makes few explicit references to race or politics. By excluding race from the model (populations are distinguished only by economic status), Forrester makes the optimistic assumption that economics is color blind. Forrester argues that economic systems should not be segregated by race, and that his model reveal that a healthy mix of wealth levels leads to economic vitality (ibid., p. 115).

Comparing Forrester's system dynamics city model to Thomas Schelling's contemporaneous model of segregation, which does explicitly model both space and race (or gender, or other preference types), will throw into relief the underlying ideas and enactive representations that underpin them. Furthermore, it will help us appreciate *SimCity's* simulation design, which can be seen as a hybrid of these two simulation techniques, combining the distributed spatial mechanics employed by Schelling with the aggregate population mechanics of system dynamics.

Schelling, an influential Cold War game theorist, published his paper "Dynamic models of segregation" in 1971, which was based upon research performed while at the RAND institute (Schelling 1971).[37] The scientifically minded urban projects of Schelling and Forrester can be seen as part of what Jennifer Light describes as a broad orientation of intellectuals and defense contractors, such as RAND, towards new markets during the Cold War, and the subsequent adaptation of expertise and techniques developed for defense purposes towards cities. The "urban crisis" and "failures of urban renewal" of the 1960's stimulated demand for a "more scientific approach to managing cities." To address the urban riots of the late 1960's, Lyndon Johnson enjoined "military strategists, systems analysts and social scientists" rather than military troops. The Ford Foundation sponsored the "RAND Workshop on Urban Problems," which took place at the end of 1967 (Light 2002), and provided seed money for the MIT Urban Systems Laboratory (founded in 1968), through which it paid for the computer time Forrester used to develop *Urban Dynamics* (Forrester 1969, p. x; "MIT History | Department of

[37] For a playable essay version of Schelling's simulation see (Hart and Case 2014).

Urban Studies and Planning" 2004).[38]

Schelling, whose background was in game theory, proposed both spatial and aggregate numerical models to account for segregation by race. These models demonstrate how macro scale segregation can emerge from the accumulated decisions of individuals that reflect only slight preferences for not being in the minority.

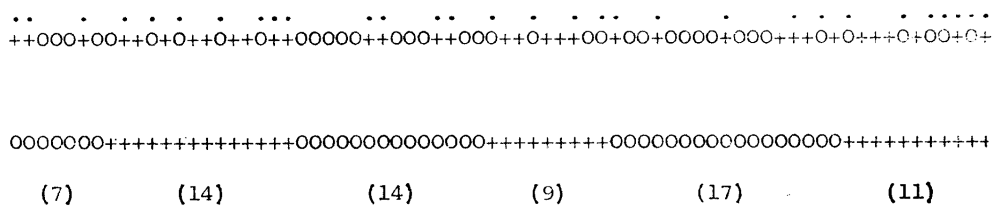


Figure 3.23: Schelling's one dimensional model of segregation. Top row is starting random configuration, and bottom are the results following preferential movements (Schelling 1971).

The one dimensional spatial model of segregation is shown above. Preferential movement rules are applied to the randomized before state (top) to yield the segregated after state (bottom). The O and + symbols represent agents of different color. Each agent wishes to not be in the minority of its neighborhood, which is the two symbols on its left, and the two symbols on its right. Unsatisfied symbols (marked with dots overhead) move to the nearest satisfiable space, and out of these preferential movements emerge radical changes.

[38]This focus of academic energies upon urban problems was further reflected in the renaming of the MIT Department of City and Regional Planning (home of the Urban Systems Laboratory), in 1969, to the Department of Urban Studies and Planning, which marked a "broader concern with issues of urban and regional development, such as needs of minorities, environmental problems, and social issues" ("MIT History | Department of Urban Studies and Planning" 2004).

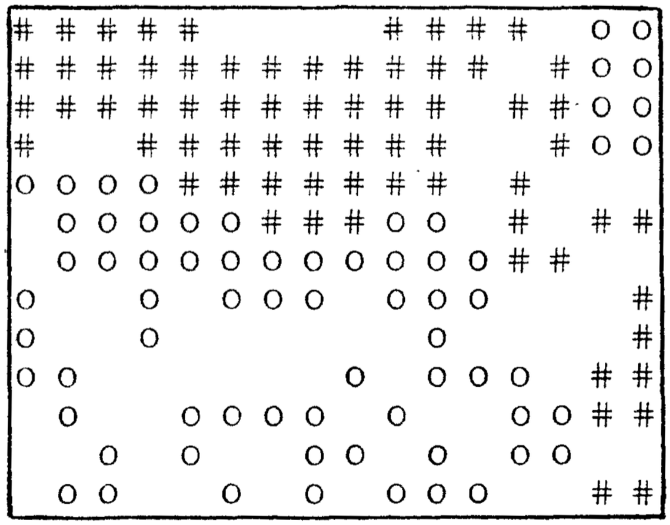


Figure 3.24: Segregation resulting from Schelling's two dimensional model (Schelling 1971).

Schelling then goes on to offer a two dimensional model, which shows equally striking results. Following these spatial models, a numerical model is offered of "neighborhood tipping," which considers how a population of blacks and whites will evolve over time in response to the tolerance preferences (for minority/majority status) of a population.

Both Schelling and Forrester employ the art of simulation and aggressively abstract from the real world, yet produce very different virtual cities, and employ divergent modes of simulation. This divergence is due in part to the research questions to which they address themselves. While Forrester was interested in the effect of policy on urban growth and decay, Schelling concerned himself with segregation. This leads Forrester to produce an economic model, and Schelling to produce a model about space and race. Their different backgrounds also influenced the approach taken and materials used, and, most relevant to the upcoming discussion of *SimCity*, the placement of goal seeking behaviors within the model. Schelling was steeped in game theory, which focuses on the interactions between multiple actors harboring diverse agendas. Thus, he produced models which explain macro-behavior in terms of individual choice, used manual simulation (physical tokens such as coins), and closed form mathematical analysis. Forrester's servomechanism inspired digital computer approach represents populations only in aggregate, and expresses goal seeking behavior in aggregate only. Both model feedback, but in Schelling goal seeking behavior emanates bottom up, from the desires of individuals in specific places, while in Forrester goal seeking behavior emanates from the top down, as the desires of an aggregate population seeking a certain state of affairs.

There are other differences between the approaches of Schelling and Forrester. Many more variables are employed by the spatial model (one per location in space) than by the system dynamic one (which has nine major levels, plus auxiliary rates and levels). Schelling's variables encode discrete (O or +) rather than scalar (numbers) state, like Forrester's simulation. The system dynamic model can take variable sized time steps, while time in the spatial model is locked to fixed size time steps. Rules for the spatial model are quite straightforward, and can be explained in a few paragraphs, while the system dynamic model contains a huge number of equations requiring an appendix to explain fully. This complexity, or opacity, is also reflected in the lengths of the narratives about the running models; the behavior in *Urban Dynamics* requires a lot of explication, while Schelling's segregation behavior is easily understood. Schelling foregrounds the materiality of his representation, underlining the aspect of make believe pretense. He repeatedly calls attention to his materials—the pennies and nickels that stand for families—vigilantly reminding us that we are looking at "stars," "zeros," and "squares" which collectively constitute a metaphorical " 'neighborhood'," and that these are only make believe " 'segregated patterns'." Perhaps he feels a need to counterbalance the striking impressionistic similarity between his segregational patterns and the real world. Acknowledgement of this *as if* attitude, which simultaneously connects and distances the make believe prop from what it represents, collapses once Schelling moves on to his quantitative model, a style which is familiar to Forrester as well.

This comparison foregrounds the specific commitments of Forrester's approach. Schelling's segregation acknowledges racial bias (or other preference types), while Forrester's urban dynamics model is colorblind, emphasizing socioeconomic differences only. But more fundamentally, Forrester and Schelling offer radically different schemas for modeling the world as a complex system. System dynamics doesn't consider individual agents or space; there are only statistical aggregates of populations, attitudes, and resources. The world takes shape through top down fluxes of population aggregates. Schelling the game theorist, on the other hand, sees the world as bottom up. For him, the world emerges from the microscopic interactions of many autonomous agents, each pursuing their own agendas.

Perceiving the distinction between these alternative modes of simulation is crucial to understanding the simulation hybridization found in *SimCity*. *SimCity*, as I will show, reinterprets Forrester's methods, reconciling a top down simulation of aggregate social quantities with representations that are bottom up, spatial and more agent based. For now, it is enough to recognize that an inherent tension exists between these approaches, which we will later see reconciled in *SimCity's* simulation architecture.

Conclusion: Modeling the World System

With his urban model, Forrester decided that *industrial dynamics* was no longer an appropriate title for his simulation practice—*system dynamics* was. Early on in *Urban Dynamics*, Forrester writes that "the term 'industrial dynamics' has become too restrictive, because the methods are applicable in many fields." The essential ideas of a closed system, feedback, levels, and flows apply to "all systems that change through time." These include "engineer systems, biology, social systems, psychology, ecology, and all those where positive- and negative-feedback processes manifest themselves in growth and regulatory action" (Forrester 1969).[39]

At this level of abstraction and maturity, system dynamics was ripe for adoption by others, including game designers. In order for Will Wright to adopt the techniques and outlook of system dynamics, a bridge was needed to connect Wright's interests to Forrester's methods, which is exactly what *Urban Dynamics* would have neatly provided (chapter 5). *Urban Dynamics* gives a streamlined account of the schema and craft of system dynamics, which is concerned with "structure as found in all dynamic systems." Forrester describes "four hierarchies of structure":

> Closed boundary around the system
>
> > Feedback loops as the basic structural elements within the boundary
> >
> > > Level (state) variables representing accumulations within the feedback loops
> > >
> > > Rate (flow) variables representing activity within the feedback loops
> > >
> > > > Goal
> > > >
> > > > Observed Condition
> > > >
> > > > Detection of discrepancy
> > > >
> > > > Action based on discrepancy
>
> (Forrester 1969, p. 12)

Exploring the vast new horizons *Urban Dynamics* had awoken him to, Forrester embarked on an even grander simulation project: modeling the world. Published in 1971, *World Dynamics* used the system dynamics paradigm to simulate worldwide population, resources,

[39]Forrester gives the example of Kurt Lewin's psychological field theory, equating its "life space" to level variables (Forrester 1969, p. 14). The broad applications of system dynamics is also reflected in the metaphorical equivalence Forrester draws between cities and living things (e.g., "a living, self-regulating system," ibid., 129).

pollution, population, technology, and capital (Forrester 1971/1973). Again, Forrester's expanded subject was prompted through his association with the Sloan School, this time via his colleague Carrol Wilson, who invited Forrester to attend a meeting of the Club of Rome, "a small international group of prominent businessmen, scientists, and politicians" which had envisioned a model for the "'world *problématique"'* (Edwards 2010, p. 366). *World Dynamics* functioned as a Malthusian argument updated for the computer age, modeling human development as an autocatalytic process, which left unchecked would lead to catastrophic collapse. Together with *The Limits to Growth* (Meadows et al. 1972), which followed in the footsteps of *World Dynamics,* this work provoked widespread debate and discussion (e.g. *Wall Street Journal, Science, Playboy)* (Forrester 1971/1973), for the first time bringing a simulation model to a popular and political audience (Edwards 2010, p. 366). Forrester writes in the preface to *World Dynamics* that

> Only by discovering how the ethical, political, physical, technical, economic and social forces of society interact with one another, can we understand the alternative patterns of future development. … System dynamics could be the unifying framework and vehicle for interdisciplinary communication. Not only is system dynamics capable of accepting the descriptive knowledge from diverse fields, but it also shows how present policies lead to future consequences. … Time is short. We must move quickly if we are to keep future options open (Forrester 1971/1973).

Like Asimov's science fictional mathematician and psychohistorian Hari Seldon, Forrester constructed a model for perceiving the future of human civilization, and averting potential systemic disasters. While Seldon forecast the fall of the Galactic Empire, Forrester saw the collapse of world civilization.

But this prediction, while wildly fascinating, turned out to be wildly wrong. It wasn't really science or engineering. It was, as Forrester had said all along, predicated on intuition and art. But this was exactly what Will Wright needed to make *SimCity*: enactive representations for modeling social systems on computers, accompanied by a craft which enabled, guided, and inspired imaginative appropriation. System dynamics lends itself to appropriation because it does not require a great depth of computer knowledge, computer power, or hard data. You don't need a supercomputer, a Ph.D. in social science, specialized software, or a deep programming background to build a system dynamic model. What you do need is a lot of imagination.

I was curious how Forrester the earnest engineer approached the imaginative dimensions of his work. I wondered whether he ever saw his model making in terms of make believe or play; if he and his colleagues took pleasure in designing, tinkering with, and running their toy models. To my delight, a 97 year old Forrester replied to my email, but reported that he seldom gave interviews anymore. Despite his poor health, he did agree to answer a handful of questions over email. Imaginative system dynamic models may harbor "interesting possibilities," he thought, but was not something that he had considered. No, it wasn't play or pleasure, but "[w]ork, aimed at a useful result." He still saw the world in terms of levels and rates. System dynamics, he thought, was like the profession of engineering in 1870, poised to one day explode and remake the world. Already, he noted, it was "spread thinly worldwide, and can enter every human activity." Forrester hadn't ever played *SimCity* or other popular simulation games, but he did know about them. A couple days later, I asked Will Wright if he continued to use system dynamics in his simulations after *SimCity*, and he regarded me curiously. What kind of question was that? "It's really just a way of thinking," he said, as "fundamental" and basic as "procedural thinking." System dynamics was a schema for the world; an imaginative modeling strategy; a way to represent with feedback loops (Forrester 2016; Wright 2016).

Figure 3: Simplified Flow Diagram Model of the Dark Night of the Soul.

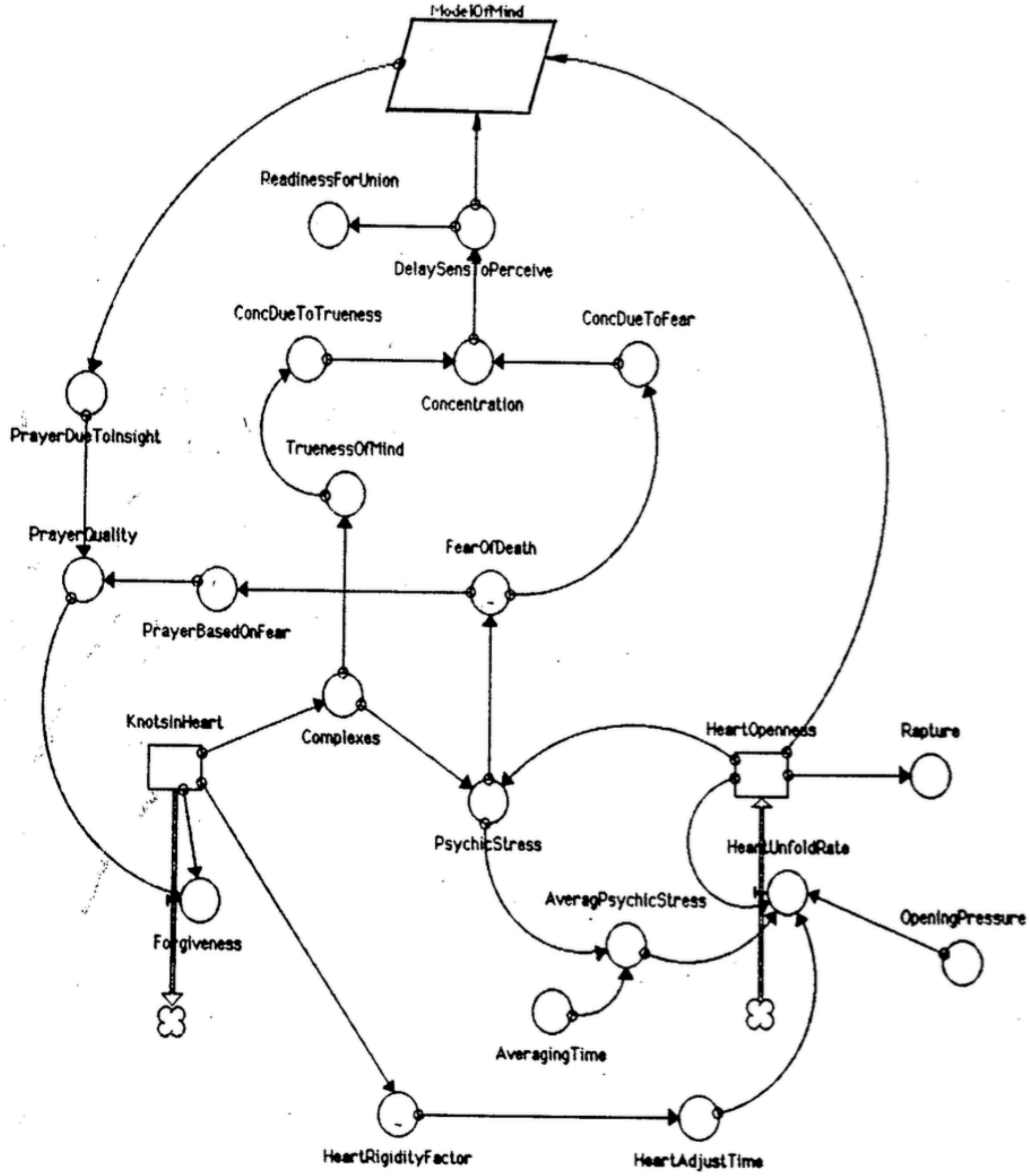


Figure 3.25: Figure from Arlen Wolpert's "Application of System Dynamics to the Study of a Religious Experience" (Wolpert 1992). This figure illustrates the flexibility of system dynamics in the hands of imaginative model builders, and the critical role of description in tethering a model to a phenomena.

Like Lego building blocks, Forrester's levels and rates are enactive representations abstract enough to be applied to anything. The craft he cultivated, its way of seeing the world, its technical implementation, and the associated publications and social practices, all worked to spread a tradition of computational make believe. Forrester's publications meet the willing reader halfway. They scaffold make believe participation, inspiring and guiding readers in the craft of believing in and then making system dynamic models. Forrester had produced a set of computer building blocks—enactive representations—for make believe that Wright could easily adopt. This is exactly what Wright did, first in *SimCity*, and then later in projects such as *The Sims* (Hopkins 2008).

But in order to make *SimCity*, Wright needed more than system dynamics. He needed something like Schelling's model, which was bottom up and spatial. He found this in cellular automata, the other major computational simulation tradition he appropriated, which we turn to next.

Chapter 4

Cellular Automata

> In Greek mythology, the machinery of the universe was the gods themselves. They personally tugged the sun across the sky, delivered rain and thunder, and fed appropriate thoughts into human minds. In more recent conceptions, the universe is created complete with its operating mechanism: once set in motion, it runs itself. God sits outside of it and can take delight in watching it.
>
> Cellular automata are stylized, synthetic universes defined by simple rules much like those of a board game. They have their own kind of matter which whirls around in a space and a time of their own. One can think of an astounding variety of them. One can actually construct them, and watch them evolve. … once we've been shown a cellular-automaton universe we'll want to make one ourselves; once we've made one, we will want to try another one. …
>
> A cellular automata machine is a universe synthesizer. Like an organ, it has keys and stops by which the resources of the instrument can be called into action, combined, and reconfigured. Its color screen is a window through which one can watch the universe that is being "played."
>
> This book, then, is an introductory harmony and orchestration manual for "composers" of cellular-automaton universes (Toffoli and Margolus 1987b).

Introduction

Cellular automata have been philosophical playthings for multiple generations of influential computer scientists and mathematicians: Alan Turing, John von Neumann, Konrad Zuse, Edward Fredkin, and Stephen Wolfram—to name a few.[1] Such systems are practically world building kits for modeling phenomena as extraordinarily diverse as morphogenesis, evolution, self-reproducing machines, physics, fantasy dungeons, circuits, insect colonies, and geomorphology. Unlike system dynamics, which models populations in terms of numerical aggregates, these cellular models are inherently spatial and bottom up: global dynamics emerge from local interactions.

Cellular automata are the other major simulation tradition Wright appropriated into his computer play.[2] Like system dynamics, the tradition of cellular automata was propelled by its own schema, craft, social practices, and evocative artifacts. Like many before him, Wright became fascinated with one cellular automaton system in particular, John Conway's *Game of Life,* losing almost a year of his life to it. While under its spell, he became versed in the dazzling possibilities of cellular automata as a resource for representation and play. And it was through creating his own versions of *Life* that Wright became a proficient Apple II programmer, able to program the machine in Basic, Pascal, and assembly language (Wright 2011; chapter 5).

System dynamics and cellular automata are different kinds of play materials. They offer vastly divergent ways of thinking about and representing the world as a complex dynamic system. To build *SimCity,* Wright would reconcile them into a hybrid simulation. Whereas the schema of system dynamics models the world from the high altitude vantage of levels and rates—quantities representing things like populations, attitudes, and inventories—cellular automata model the world from the ground up, as a landscape of interacting elements, very much like the Schelling neighborhood segregation model we looked at in chapter 3. The

[1] And systems that resemble them, such as partial differential equations (PDEs).

[2] Cellular automata style systems manifest in a wide variety of computer simulation playthings. Perhaps the most well known example is *Minecraft. Minecraft's* makers and players, in fact, recapitulate the history and origin of cellular automata, and create working computers with its 3-dimensional cellular world.

enactive representations of system dynamics lend themselves to sociological representation, and would be used in *SimCity's* economic and social model for things like unemployment, migration, and housing demand. But this wouldn't be enough for *SimCity,* which also models things like floods, fires, and the plan of the city itself. To make *SimCity,* Wright also needed the enactive representation of cellular automata, which, as we will see, is ideally suited to modeling the natural world as a dynamic spatial process. Deconstructing the artifice of cellular automata will also help us to understand both their aesthetics and appeal as a resource for play.

Owing to their distinct schemas and histories of practice, system dynamics and cellular automata scaffold play differently. While both traditions trade on imaginative representation, simulation makers approach their craft differently. System dynamics was self-serious and earnest, concerned with industrial optimization and social crises. In appropriating it, Wright turned it to a more playful and autotelic end. Cellular automata, on the other hand, have a long history of lighthearted use, making them more natural play scaffolds. Conway, after all, called his cellular automata system the *Game of Life.*

But the genesis of cellular automata lie not in a game, but rather in an earnest search for a computational model of organic life. Aided by the increasingly widespread availability of computers, it blossomed into a practice for modeling the universe at large. To understand the commitments and aesthetics of cellular automata—the modeling practice that Wright and others would appropriate for their own serious and playful ends—we must look towards their beginnings.[3] When John von Neumann, a brilliant mathematician and scientist bitten by the

[3]Looking at the history of cellular automata, which extends back to the late 1940's, helps us to explain not just how the approach came about, but the ideas—the peculiar way of looking at the world—embodied in it. History helps us to understand their appeal and generality.

My aim is not to give a complete history of cellular automata, but rather to elucidate the underlying ideas inherent to the approach, and show how those ideas came about, and to trace some of their consequences. Primary sources are sometimes sketchy for reasons that will soon become clear, and although numerous good surveys of cellular automata research exist (e.g. Mitchell 1996; Sarkar 2000; Ganguly et al. 2003), the closest bug of computation, begins to wonder how the dynamics of computation might be used to unravel one of the universe's deepest mysteries: life.

Life

John von Neumann

Von Neumann's efforts to synthesize a "combinatorial theory of automata and organisms" is where the story of cellular automata begins. When he began this project, his celebrated career had already encompassed topics as diverse as pure mathematics, nuclear weapons, computer design, and game theory. A synthetic thinker, he imaginatively mixed ideas across disciplines.[4] For example, he co-created the highly influential field of game theory, synthesizing a fusion of

---

thing to a thorough history is to be found in Stephen Levy's *Artificial Life* (1992), and to a lesser degree Stephen Wolfram's *A New Kind of Science* (2002), both of which have proven useful in building my account. While I lean heavily on Levy's account to map the narrative and bibliographic territory, my emphasis is quite different. Levy's concern is with the development of Artificial Life as a field and illustrating its wonders, a story which coincides with cellular automata from time to time. The partial histories found in practitioner publications, for example Mitchell's survey, emphasize technical history and detail. My study is more humanistic than journalistic or technical. While I am also interested in how and why cellular automata developed, my angle is to show how cellular automata crystallized into a particular kind of simulation practice, were underwritten by particular commitments, furthered the agendas and interests of its practitioners, and exhibit certain qualities I argue are intrinsic to the enterprise.

[4]If John von Neumann had not, in 1957, died of cancer at the young age of 53, having already made important contributions to a variety of fields—from mathematics, physics, and meteorology to computation and economics—how much more might he have accomplished? His friend, collaborator, and fellow European (Jewish) emigre Stanisław Ulam wrote in 1958 that "[t]o follow chronologically von Neumann's interests and accomplishments is to review a large part of the whole scientific development of the last three decades" (Ulam 1958). In addition to his mathematical contributions, von Neumann worked on the atomic bomb, co-invented game theory, and contributed key ideas to the design of the first programmable electronic computers. Ulam divides von Neumann's career into an early period of contributions "to the edifice of existing work" from his later efforts

> to blaze new trails and to create new syntheses. … This was to be a combinatorial theory of automata and organisms. His illness and premature death permitted him to make only a beginning (Ulam 1958).

In addition to his untimely death, his governmental consulting work—he was a member of the Atomic Energy Committee, a U.S. Atomic Energy Program commissioner, and headed the secret U.S. ICBM committee—explains the incompleteness of his work, and also the scattered nature of primary sources on the genesis of cellular automata (Burks 1970, p. xiii; Wikipedia, John von Neumann, 2016).

mathematics and the study of games that transformed social science, economics, and political strategy.[5] His contributions to the design of the first programmable digital computers (EDVAC) are infused with ideas from the then emerging study of neural mechanisms. He compares the vacuum tube elements of a computer to neurons; concepts and terminology from mathematics and biology are intermingled in the service of computer design.[6]

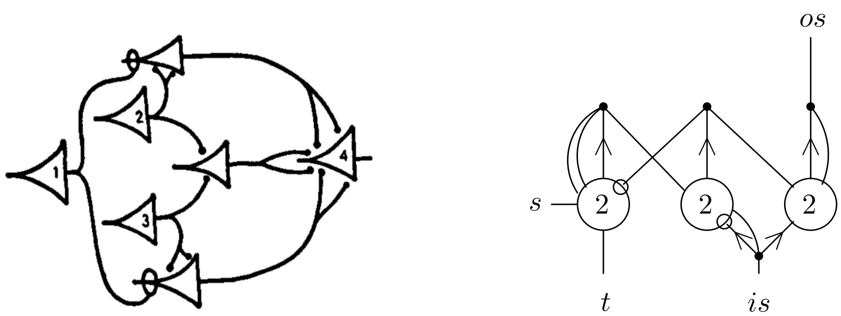


Figure 4.1: At left, a figure from a seminal mathematical model of neural activity (McCulloch and Pitts 1943). At right, a figure from von Neumann's EDVAC report (von Neumann 1945/1993).

His synthetic interest in computation, mathematics, and biology also gave rise to a project that would eventually lead to cellular automata: a mathematical model of biological reproduction.[7]

[5]Von Neumann and Morgenstern's work with game theory produced a mathematical models of social interactions inspired by games, injecting mathematics into the complex ambiguities of social life. Game theory has become a key tool in social sciences (Leonard 1995). In the work that led to cellular automata von Neumann made a similar move, imaginatively plumbing the relationship between math/computation and the systemic complexities of organic life. Once again, von Neumann established a highly generative relationship that linked the mathematical to a complex domain that had never received such formal analysis.

[6]The EDVAC report, which established the blueprint for programmable electronic computers, is suffused with a "neuron analogy" inspired by the simplified mathematical neural model of McCulloch and Pitts (von Neumann 1945).

[7]Biology and computation had been a source of mutual inspiration for von Neumann, and he had participated in the Macy conferences, where cybernetic ideas for exploring isomorphic relations between machines, organisms, thought, self-regulation, and society were hashed out by an interdisciplinary coalition of leading intellectuals (Hayles 1999).

He continued the mathematical synthesis of the organic and computational in a 1948 Hixon Symposium lecture, in which he explored the idea of self-replicating machines (von Neumann 1951), a project that would eventually lead to what are now called cellular automata. The talk establishes the conceptual foundation upon which later

In Search of the Universal Constructor

Inspired by recent scientific advances that offered mechanistic models of neurons, von Neumann sought a mechanistic explanation of organic life's reproductive capacity. In a 1948 lecture, he quested for a "general logical theory of automata" that could be applied equally well to "natural organisms" as well as "artificial automata" (von Neumann 1951).[8]

To accomplish this, von Neumann employed the abstraction, decomposition, and axiomatization of his mathematical craft. To do this, he employed two key abstractions.

First, he proposed to divide complex living organisms into assemblages of "well-defined" elements. Observing that "organisms can be viewed as made up of parts which to a certain extent are independent, elementary units" von Neumann separates the "the structure and functioning of such elementary units individually" from "understanding how these elements are organized into a whole[.]"

Second, he black boxed each elementary unit, transforming each organic cell into an identical and predictable automaton. Each living cell, now a black box, is divided into two domains: the inside of the box, governed by the laws of organic and physical chemistry, and its exterior behaviors which dictate how those black boxes behave and compose, like building blocks,

---

work would build. In the printed version of the talk von Neumann noted that

> In the present write-up it seemed appropriate to follow the dispositions of the talk; therefore this paper, too, is in many places more sketchy than desirable. It is to be taken only as a general outline of ideas and of tendencies. A detailed account will be published on another occasion (von Neumann 1951).

While the ideas would evolve under von Neumann's watch, and circulate within certain scientific communities, further published accounts would only emerge through the efforts of others.

[8]Admitting that "natural systems are of enormous complexity" and that any effort to mathematically model them would, by necessity, involve abstractions and problem subdivision that was limited, von Neumann nonetheless proceeded to articulate his strategy for tackling "natural organisms." He frequently calls computers "artificial automata," and the title of the paper is "The general and logical theory of automata," which alongside the content of the paper, hints at a conception of natural organisms as a class of *natural automata*.

into functional assemblages. The outward behavior of the box becomes totally predictable, amenable to the interests and tools of the mathematician or logician.

This axiomatic operation, of separating the material from the mathematical, provided an aperture through which mathematicians and simulation makers such as von Neumann might slip into the world of the biological and study living organisms.[9] Having framed organic life in this way, with the messy business of organic chemistry behind him, von Neumann proceeded to explore how complex living wholes are constituted by these elementary and mechanistic parts.[10] And he could now do so using the tools of mathematics.

In particular, he was interested in the question of self-reproduction:

> The problem of self-reproduction can then be stated like this: Can one build an aggregate out of such elements in such a manner that if it is put into a reservoir, in which there float all these elements in large numbers, it will then begin to construct other aggregates, each of which will at the end turn out to be another automaton exactly like the original one? This is feasible, and the principle on which it can be based is closely related to Turing's principle outlined earlier (von Neumann 1951).

Von Neumann outlines a "logico-mathematical" (Langton 1984) proof that these automata can asexually reproduce. Turing machines, abstract computers with a chameleon like abil-

---

[9] A rift is interposed between flesh, matter, and material on the one hand and structure, function, and behavior on the other. By dematerializing living cells into mathematical automata von Neumann makes a move congruent with Hayles's observation that the broader cybernetics project has enacted a divorce between material forms and information patterns, privileging the latter in a "condition of virtuality" (Hayles 1999, p. 19). Von Neumann, of course, was well aware of the obvious artificiality of this abstraction and wrote that he "need not emphasize the limitations of this [axiomatic] procedure."

By teasing apart the physiological, itself "closely connected with the most difficult chapters of organic chemistry and of physical chemistry," from the organizational, von Neumann was free to explore how "the functioning of the whole is expressed in terms of these elements," an endeavor "which is likely to attract those of us who have the background and the tastes of a mathematician or a logician."

[10] There are two problems, the physiological and the organizational, and as von Neumann is interested only in the latter, he "remove[s] the first part of the problem by the process of axiomatization, and concentrate[s] on the second one."

> investigate the larger organisms that can be built up from these elements, their structure, their functioning, the connections between the elements, and the general theoretical regularities that may be detectable in the complex syntheses of the organisms in question (von Neumann 1951).

ity to do anything any other automaton can do, become swept up in a design for asexually reproducing machines. Christopher Langton summarizes the proof:

> If self-reproduction is being carried out by a (highly complex) biochemical machine, then that machine's behavior is describable as a logical sequence of steps, i.e. as an algorithm. Now, if an algorithm can be performed by any machine at all, then there is a Turing machine which can perform the same algorithm. For this reason von Neumann set out to demonstrate the existence of a Turing machine which could effect its own reproduction (Langton 1984).

Von Neumann imagined a large number of "elements" floating in a "reservoir." A properly constructed machine made of such elements could, when introduced to the reservoir, seek out the necessary elements to reproduce itself. It would paddle about, collect parts, and then assemble those parts into copies of itself. The self-reproducing assemblage has four subsystems corresponding to steps in von Neumann's proof:

- *Description.* Akin to DNA, this is the blueprint for a new self-reproducing automaton.[11]
- *Constructor.* Fabricates machines from the descriptive blueprint.[12] The new automaton is made out of parts floating on the reservoir, and can be more complex than the constructor itself.
- *Description replicator.* A photocopy machine for the descriptive blueprint. Given a description, a copy of that description is produced.

[11] These informational instructions, when fed to the Constructor, cause it to build the machine specified. It has "all the notational properties of a tape with fields that can be marked," but rather than being made of literal tape it is made of the same "structural elements" that float in the reservoir and constitute all of the automata. Of interest in this proof is his modification of the Turing machine tape, upon which the machine Description is inscribed, into one with a different materiality, one made of machine parts. This "tape" encodes the algorithmic instructions for building a copy of the machine, and can itself can be replicated (von Neumann 1951).

[12] "[W]hen furnished the description of any other automaton in terms of appropriate functions, will construct that entity."

- *Control mechanism.* Orchestrates the other elements. It feeds the descriptive blueprint into the constructor and description replicator, yielding an offspring machine harboring all the same subsystems: description, constructor, description replicator, and control mechanism. Most importantly, the offspring also possesses the property of self-reproduction.

This proof, inspired by Turing machines and proceeding from von Neumann's logical argument, reiterates the function of and also anticipates the discovery of DNA's role in reproduction (Vichniac 1984). In effect, both the informational tape and the machines themselves must be ontologically flattened into the same substrate—parts on a lake—if the machines are to fully self-replicate.

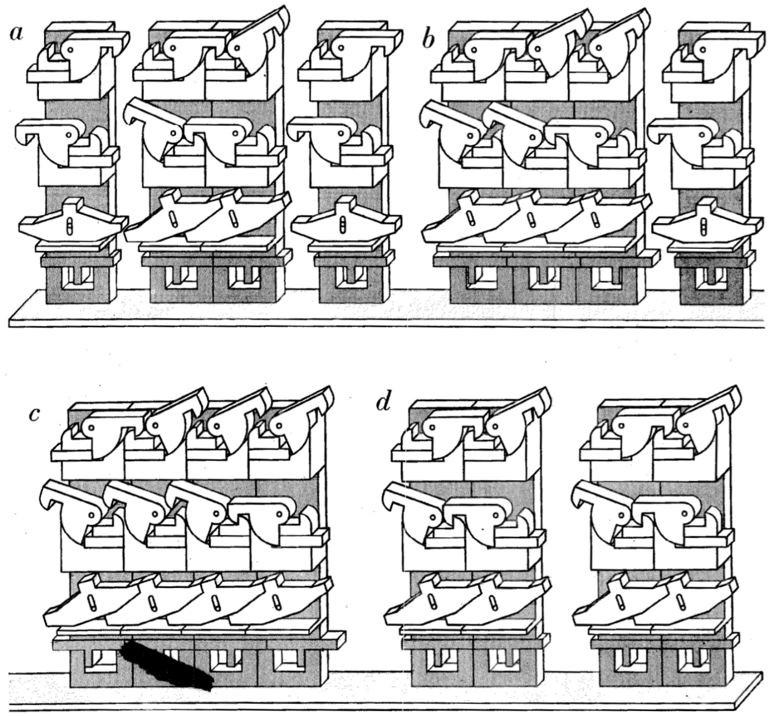


Figure 4.2: Self-reproducing machine diagram (Penrose 1959).

This wasn't the first design for a self-reproducing machine that had been entertained or even built (e.g. Penrose and Penrose 1957). Although an aggregate of simple identical machines can be seen to reproduce themselves (e.g. Penrose 1959), the machines don't do anything else. What distinguishes von Neumann's design is an ability to bootstrap up into greater complexity and eventually take on neural properties. By incorporating the informational tape and universal computing power of Turing machines into his design, von Neumann opened the door to evolution and intelligence.[13]

Three stories are apparent in von Neumann's make believe transformation of mechanistic primitives into organic life. The first is to see the logical possibility, however fantastic, of self-reproducing machines. This avenue has had ample follow on research, most memorably in the proposal of space exploration undertaken by self-reproducing robots and lunar factories (Freitas and Gilbreath 1982; Levy 1992).

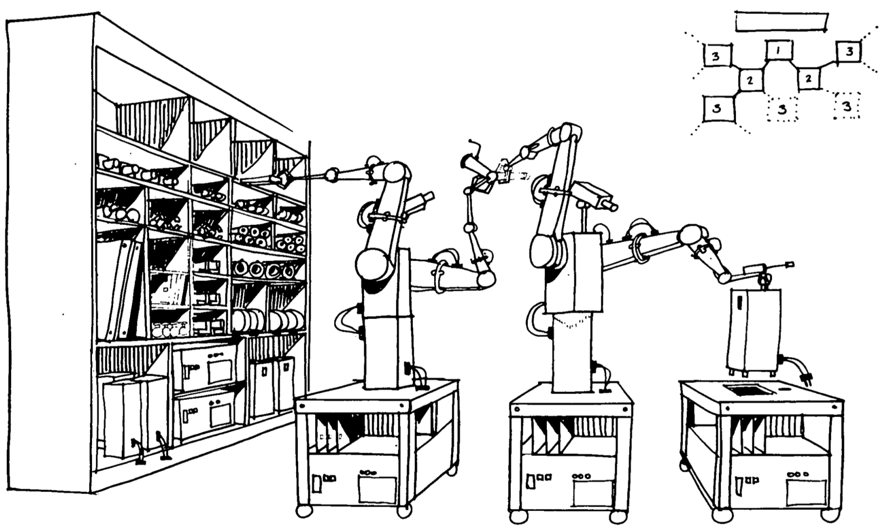


Figure 4.3: Artist rendering of "robot self-replication" from the proceedings on a NASA summer study on space exploration via self-replicating robots and lunar factories. Science fiction and science are interwoven, as speculative computer science springs to life in the imaginative and evocative imagery of NASA illustrators (Freitas and Gilbreath 1982, fig. 5.29, p. 257).

The second story is that we ourselves are self-reproducing machines. The mysterious

[13]Intelligence, of course, relied upon equating of computing primitives with neural primitives—a comparison which von Neumann did not shy away from (e.g. von Neumann 1945; von Neumann 1951; von Neumann 1958).

complexity of biological life is reduced to mechanistic processes. In this sense, the argument continues the scientific tradition of pushing the frontier of mechanistic explanations deeper into the mysteries of the natural world. Comparing automata and brains is a similarly spirited project, and in conjunction with the "universal automata" of the Turing machine, raises the possibility of self-reproducing machines with brains, which furthermore might mutate and evolve.

The third story is that organic life and mechanistic computation are isomorphic. The schema for self-reproduction is material agnostic: robots, computers, and carbon based life can all enact a single self-reproductive design.

The great mystery of life, felled by the expanding horizons of mechanistic explanation, might not be so mysterious after all. Von Neumann projected that we might one day produce life in the form of computers and robots that evolve, self-reproduce, and exhibit intelligence.

Von Neumann's imaginative vision would guide and inspire much follow on work, but the abstractions he introduced would have the most far reaching effects. Long after the vision of self-replicating machines would be set aside, the abstractions would persist, scaffolding the simulation of complex natural phenomena on computers. Two foundational abstractions underpinned this emerging enterprise:

1. *Decomposition.* Decomposition of complex natural phenomena, in this case living organisms, into identical functional units.[14] The broader strategy of decomposition would be familiar to his mathematical, scientific, and simulation practices.[15]

2. *Virtualization.* Black boxing the elementary units, carefully sealing up the messy details

[14]Biological cells provide a clue as to this decomposition, although von Neumann does not explicitly make reference to this isomorphism aside from comparing neurons to vacuum tubes.

[15]For example, the gridded lattices used in computer simulations of nuclear detonation and weather.

of biochemistry. Each box's outward behavior becomes a predictable automaton. This affords a virtualization, a divorce between materiality and information.

These abstractions free up the modeler, in this case von Neumann, to focus on how idealized elementary units combine and operate as a group to produce new wholes. The decomposition of artificial organisms into identical elements makes possible their subsequent reassembly; a self-reproducing machine would build copies of itself out of such elements.

Von Neumann has not, of course, built an actual self-reproducing machine. Nor has he even built a fully specified mathematical model. What he has done is draft a logical argument, in outline, of what such machines might look like, a proof of mathematical-computational possibility, and an outline of their form. In order to take this argument to the next level of mathematical and computational precision, and craft an enactive representation that others might reuse, he would need an additional abstraction, one suggested by a friend and collaborator of his.

The Infinite Lattice

Machines paddling around on a reservoir are, at best, a thought experiment. Although the biochemical complexity of life has been banished, a messiness foreign to mathematics remains in the kinetic physicality of floating machines. To banish this additional complexity, von Neumann would introduce a new abstraction: an infinite lattice of tessellating cells. The switch from a *kinematic model* to a *cellular model* was suggested by Stanisław Ulam, a longtime associate of von Neumann.[16] Ulam's cellular idea further purified von Neumann's abstraction,

[16]Von Neumann turned "from this mechanical model to a more abstract … two-dimensional array of elementary cells" (Shannon 1953b). A.W. Burks, who edited von Neumann's unfinished and unpublished work on self-reproducing automata, distinguishes the earlier *"kinetic model"* of self-reproduction with its attendant complexity

transforming the kinetic automata into mathematically formal elements and collating them into a spatially regular and infinite matrix.

This abstraction echoes the simulation lattices that Ulam and von Neumann would have already been familiar from weather and atomic bomb simulations.[17]

---

of "fuel and energy … geometrical-kinematic problems of movement, contact, positioning, fusing, and cutting" from the later, and more abstracted, *"cellular model"* (von Neumann 1966, p. 82).

The idea of introducing a regular lattice or grid for the enterprise's substrate was suggested by Stanisław Ulam, a longtime associate of von Neumann, and enabled an additional level of mathematical precision to enter the picture. This cellular model, according to Burks's editorial notes in a posthumously published monograph by von Neumann

> was stimulated by S. M Ulam, who suggested during a discussion of the kinematic model that a cellular framework would be more amenable to logical and mathematical treatment than the framework of the kinematic model. In the cellular model, self-reproduction takes place in an indefinitely large space which is divided into cells, each cell containing the same finite automaton. Von Neumann spoke of this space as a "crystalline regularity," a "crystalline medium," a "granular structure," and as a cellular structure.
>
> There are many possible forms of cellular structure which may be used for self-reproduction. Von Neumann chose, for detailed development, an infinite array of square cells (von Neumann 1966, p. 94).

Levy, Burks, and Wolfram all credit Ulam for this suggestion but there is no source known to me where Ulam or von Neumann explicitly state this attribution. The most authoritative claim is found in Burks's editorial notes for the posthumous von Neumann monograph, which ultimately relies upon Burks himself and no other more primary sources. E.g. "Von Neumann was going to refer to S. Ulam here" (von Neumann 1966, pp. 94 and 102). Burks again makes this claim in (Burks 1970, p. 8). There is no reason not to believe Burks, but the the trail's incompleteness is frustrating to me. This demonstrates, also, and as we will see again in Kemeny's article, how knowledge of this study circulated informally without a clear publication trail.

According to letter from von Neumann's widow to Burks, work on the cellular model began in 1952 (von Neumann 1966, p. 94). In his writing, Ulam describes the lattice formulation "considered by von Neumann and the author" (Ulam 1952).

[17] By dividing a space into a regular lattice one can numerically solve such partial differential equations, producing results that evolve in time and space. Such calculations are laborious and suited to automatic computation. ENIAC's first assignment was to simulate the explosion of a hydrogen bomb, an application suggested by von Neumann (Edwards 2010).

It was in the late 1930's that von Neumann became interested in fluid dynamics, a topic that would become of great military interest in the wake of the atomic bomb. The computer could numerically calculate what had been analytically impossible through the use of such lattices to solve partial differential equations. His encounter with ENIAC instilled in him a deep fascination with computing that would last "until his last conscious hours" according to his widow Klara von Neumann (von Neumann 1958). Alan Turing also employed spatial models of natural phenomena in his model of morphogenesis, a project, observes Burks, whose overall aim is of a kindred spirit to that of von Neumann's. Turing sought to show how globally coherent spatial biological patterning, such as zebra stripes, could emerge out of local chemical interactions. Turing's chemical basis of morphogenesis project is related in that it aims to mathematicize the organization of organisms using a computational and spatial structure (Turing 1952; Burks 1966, p. 99).

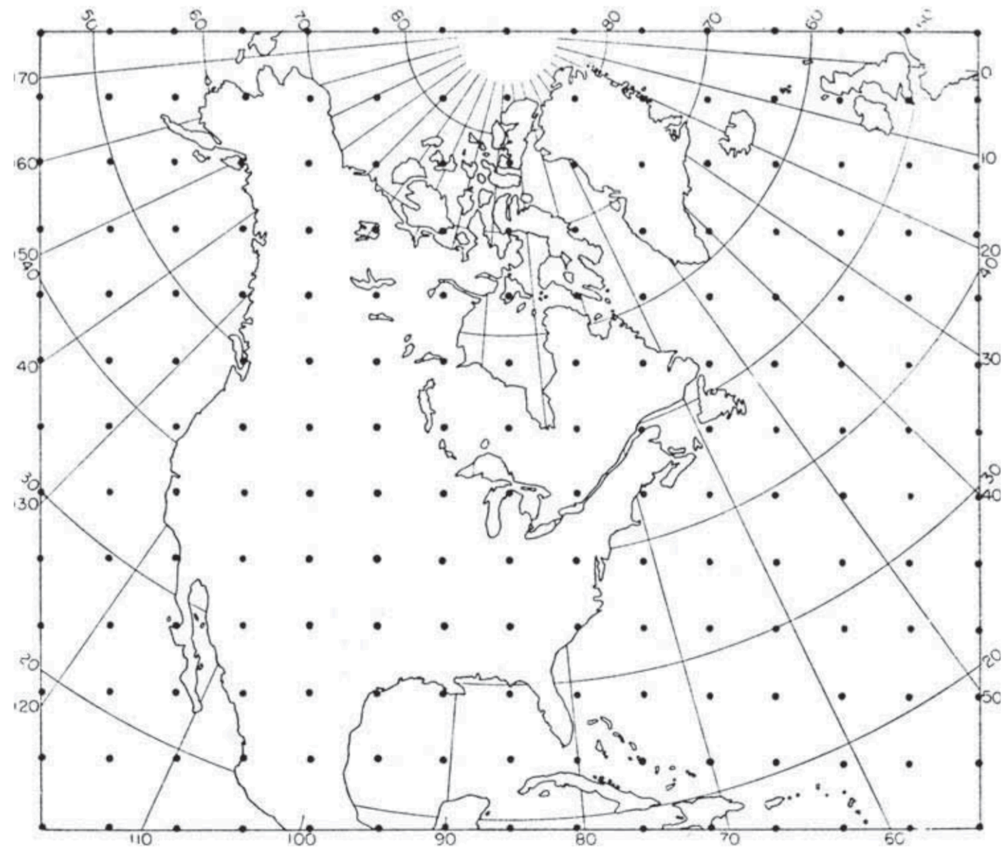


Figure 4.4: Lattice used in early ENIAC weather simulation (Edwards 2010; originally from Charney and Phillips 1953).

By adapting a simulation lattice into the self-reproducing machine model, the residual naturalistic messiness of the kinetic model—machines on a lake—could be left behind. A fully mathematical substrate offered a means for rigorously specifying the details of a self-reproducing machine. Such rigorous specificity meant that instead of a thought experiment, one could potentially even run a computer simulation. Ulam describes the enactive representation afforded by the lattice abstraction:

> Given is an infinite lattice or graph of points, each with a finite number of connections to certain of its "neighbors." Each point is capable of a finite number of "states." The states of neighbors at time t_n induce, in a specified manner, the state of the point at time t_{n+1}. This rule of transition is fixed deterministically or, more generally, may involve partly "random" decisions.

> One can define now closed finite subsystems to be called *automata* or *organisms. …* One aim of the theory is to establish the existence of subsystems which are able to multiply, i.e., create in time other systems identical ("congruent") to themselves (Ulam 1952).[18]

This formulation diverges from the tradition of lattice simulations that inspired it. Elements are visualized as squares rather than points, giving rise to a fully discrete universe: in space (cells), time (steps), state (enumerated), and action (transition functions). This makes them relatively easy to simulate with and without computers, and gives them something of the flavor of a board game like Chess rather than an approximation of a continuous phenomena like the weather, a nuclear detonation, or a physical sport.[19]

---

[18]Ulam's description captures the essential definition of what would become known by the late 1960's as *cellular automata* with one subtle difference. Ulam's formulation resembles the game of Go, where it is the points—the interstices of the lattice—that are conceptualized as existing in a state, whereas modern representations are more like Chess, where it is the squares of the lattice that are conceived of as holding state. According to Burks, "Von Neumann chose, for detailed development, an infinite array of square cells." Although a mathematician will tell you that in this case points and cells are equivalent, the cellular conception entails particular representational and descriptive strategies that much mileage has been derived from, as we will see.

[19]Traditional spatial simulations, as one would use in a fluid dynamic model of a nuclear explosion or the weather, are more or less fine grained approximations of an underlying world that is continuous in space, time, state, and activity. This is not the case with Ulam's cellular model, which models an intrinsically discrete universe. In a fluid dynamic simulation both the spatial grid and the temporal steps can in theory be made as high resolution as is computationally feasible, but this is not the case with cellular automata—one cannot "zoom in" without redefining the rules. States, also, are not continuous as in a fluid dynamic simulation. State in such a continuous simulation might represent temperature or pressure, which are inherently continuous phenomena, but state in a cellular automaton is discrete and in one of a finite number set of states.

Traditional lattices used for numerical solutions to PDEs form the backdrop to cellular automata, yet are qualitatively more continuous than discrete. The family resemblance, however, is such that Burks refers to PDEs as cellular automata (Burks 1970).

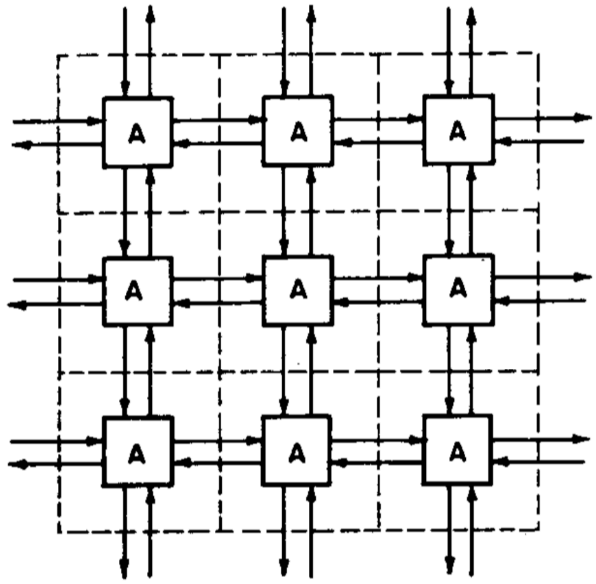


Figure 4.5: Each cell is an automaton that interacts only with its neighbors. Computational elements are interwoven into a "cellular space" (Burks 1970).

Von Neumann's self-reproducing machine design neatly fit into this enactive representation. The lattice's cells easily assumed the role of his identical elementary automata. Having black boxed the messy particulars of biochemistry, von Neumann was free to focus on how the elementary automata interacted to constitute a complex natural phenomena: a self-reproducing organism.

A Self Reproducing Automata

Cellular structure in hand, von Neumann proceeded to elaborate his design for a self-reproducing machine.[20] The cells in his system existed in 1 of 29 states, and constituted the building blocks for a universe of self-replicating Turing machines. States represent things such as empty inac-

[20]Von Neumann never completed or published his cellular model. Burks edited von Neumann's notes, which were posthumously published in 1966. By that time, Burks and his students at his University of Michigan were already pushing further into the domain first charted by von Neumann and Ulam, and which would eventually be called *cellular automata*.

tive space (a special “quiescent” state), wires, pulses, and logical operators.

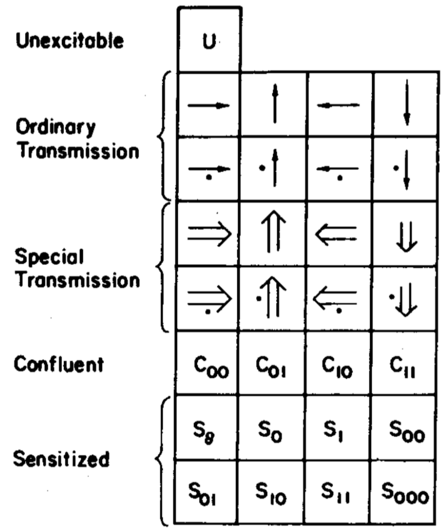


Figure 4.6: Catalog of the 29 cellular states in von Neumann’s cellular automaton design (Burks 1970, p.9).

From these primitives, von Neumann proved that one could build a Turing machine, and then showed how this machine could reproduce itself. Its design roughly reiterates the one von Neumann described in his kinetic model: a construction arm, control systems, and information tape.

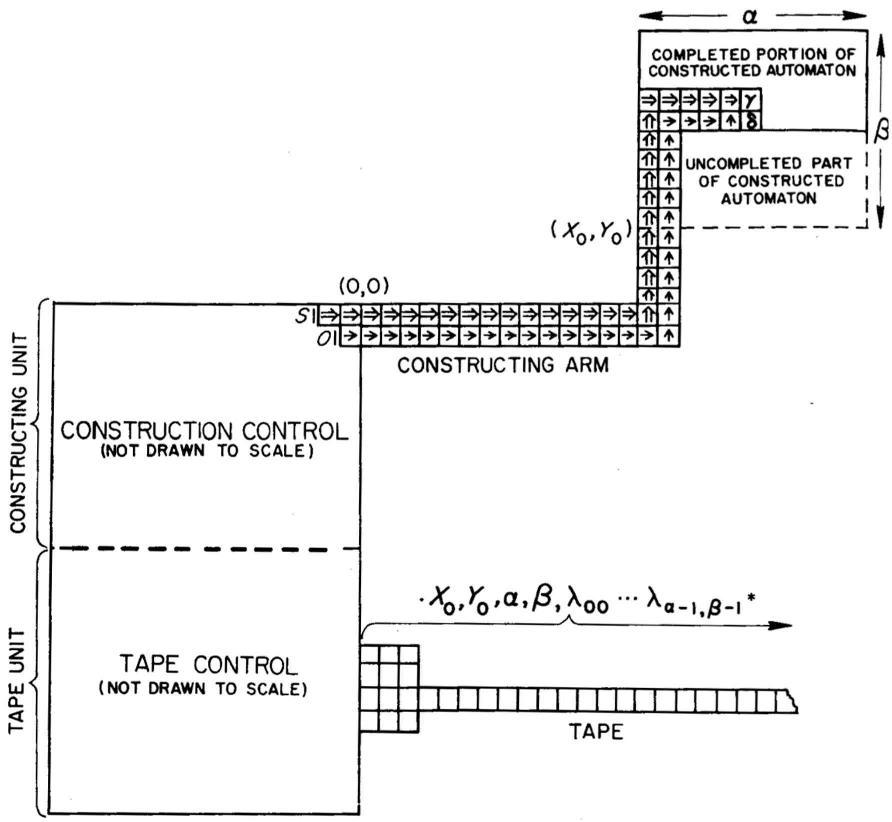


Figure 4.7: An illustration of von Neumann's universal constructor, a self-reproducing automaton embedded within a cellular automaton (Burks 1970, p. 44).

Modeling the Universe

In the illustration of von Neumann's universal constructor, one sees a self-reproducing machine embedded within a cellular world.[21] Two tiers of enactive representation are evident. Zoom out from the individual cells of the lattice, and you see multi-celled automata. The universe has two tiers: there is the computational fabric of the universe, and then there are the

[21] Note that not only is the diagram not to scale, as indicated, but it is still a schematic design. He has proved it is possible to design such a machine. It won't be until later, in the hands of future researchers, that a von Neumann cellular automaton will be specified in more detail.

automata *embedded*[22] within and built out this computational fabric.[23] The automaton follows the program encoded in the tape, and uses its construction arm to build a replica of itself.

This design is based upon that of EDVAC, the first stored program computer.[24] Before EDVAC there was ENIAC, but ENIAC, the first general purpose programmable electronic computer, was not programmed like modern computers. Programming it required rewiring cables and other physical modifications. Von Neumann codeveloped the successor design, called EDVAC.[25] EDVAC programs are stored in memory, alongside data, and so programming became a simpler affair: simply change the program data. In ENIAC there is an outer world of cables and an inner world of data. EDVAC, on the other hand, stores programs in memory and flattens these two worlds into one. Programs move from the machine's exterior to its interior, becoming just another kind of data stored inside the machine. An ontological flattening is effected, giving rise to the possibility of programs that build programs: assemblers, compilers, and interpreters.

---

[22] John Holland introduces the term "embedded" to capture this distinction (Holland 1962).

[23] Rather than model organisms directly as first order entities, the substrate of the universe itself is modeled. This is analogous to the difference between toy cars and toy cars made of Lego bricks. The two tiered model allows organisms to construct one another out of the underlying building blocks of the world. This is one reason that cellular automata are so popular—they suggest modeling the universe itself, a consideration inherent to von Neumann's original project. The "subsystem" comment also points to another facet of the appeal of such systems: their narrative flexibility; point to any collection of cells and call it a this or a that.

[24] Burks observes that the rules and states of von Neumann's 29 state cellular automaton "were based on those he used in the logical design of the EDVAC, the first stored program electronic computer," which were also "essentially the same primitives … used in his kinematic automata." Burks describes von Neumann's adaption of EDVAC's "logical design primitives" into a 29 state system as "ingenious" (Burks 1970, pp. 58-59). Von Neumann served as a consultant to the ENIAC project (where Burks himself also worked), and contributed to the design of the successor EDVAC project. Whereas ENIAC was reprogrammable via physical patch cables EDVAC was a stored program computer—the data and programs both resided in memory. In this, von Neumann was inspired by the "idealized neural elements of McCulloch and Pitts" according to Burks's editorial notes, a fact which is evident in the EDVAC report itself (von Neumann 1945; von Neumann 1966, p. 9). So, the 29 state von Neumann cellular automaton system was inspired by both the logical design of the programmable EDVAC computer and his kinetic automata system, all of which were informed by his thinking of neurons and electronics as isomorphic. While EDVAC was a realized programmable computer design—an electronic brain, the von Neumann cellular automaton constituted a mathematically realized self-reproducing thinking machine—an electronic organism.

[25] The EDVAC design would first be realized in the British EDSAC machine.

But what of creatures which build creatures—self-reproduction? Von Neumann's universal constructor reiterates this ontological flattening, and to similar effect. In order to have a Turing machine build a new Turing machine, the program and data upon which it operates would need to be flattened into the same world. The Turing machine control unit (program) would need to be embedded within, and made out of, the same universal substrate as the data (tape). A machine could then, by modifying elements of the combined tape/world, make new machines. Achieving this ontological flattening required conceiving of primitive building blocks from which Turing machines could be composed. In other words, von Neumann moved down a level of abstraction, breaking his self-reproducing machine into more primitive parts, and modeling a universe made of those parts. The von Neumann cellular automaton reproduces the hardware of computation in idealized form, as a 2d fabric of interacting computing elements. The possibility of self-reproducing machines emerges out of an appropriately designed universe.

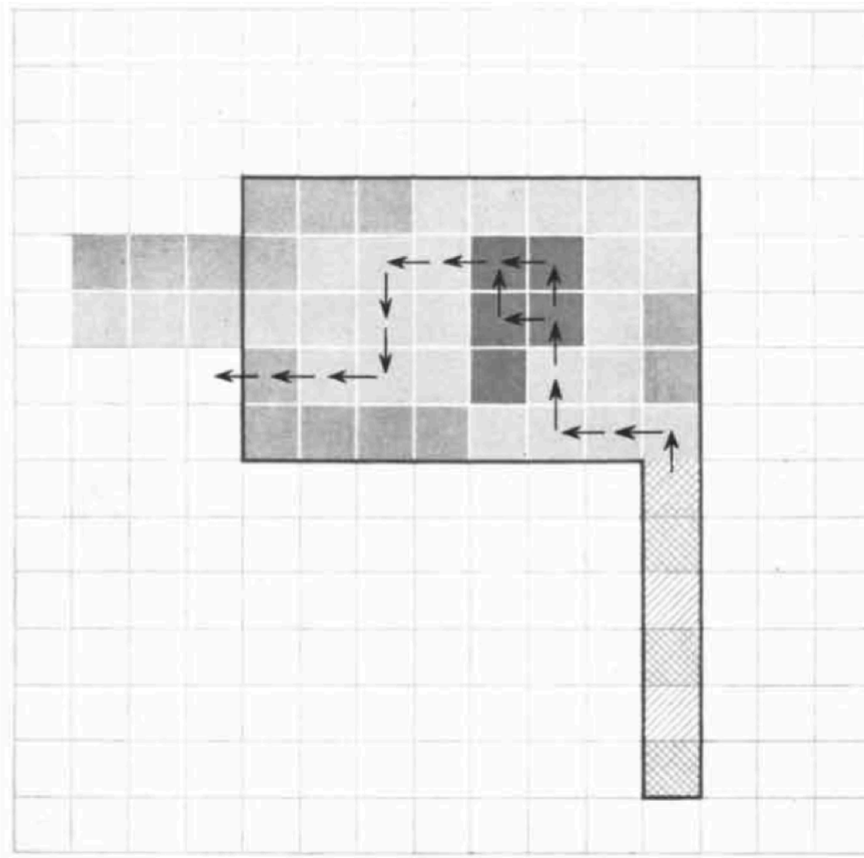
Discursive Practices

The figure of von Neumann's universal constructor also highlights the crucial role of discursive practices. Multicellular aggregates are outlined and annotated as a tape, construction arm, control system, and so on. An abstract lattice becomes a self-reproducing machine through the magic of make believe. The active creation of belief is scaffolded by illustrations that guide and inspire our imaginative participation. These discursive practices become a regular part of the cellular automaton simulation tradition, and will be apparent, in adapted form, in *SimCity* itself. They are also vividly on display in the first published synthesis of von Neumann's reproductive cellular model, written by John Kemeny and published in a 1955 issue of *Scientific*

American (Kemeny 1955).[26]

[26]John Kemeny, another Jewish-Hungarian emigre who worked on the Manhattan Project, is probably best known for his educational work co-developing the BASIC programming language. According to Burks, von Neumann had already promised a manuscript to the University of Illinois Press, so it was arranged for Kemeny to write up these ideas (Burks 1966, p. 95). Kemeny's article, "Man Viewed as a Machine," appeared in a 1955 issue of *Scientific American*, and is a accessible but breezy account of the thrust of von Neumann's thought experiment: that neural thinking processes are theoretically doable by computer, that it is possible for such a machine to reproduce itself and evolve, and the cellular model of a self-reproducing Turing machine (Kemeny 1955). If the specter of organic life as *natural automata* (in contrast to *artificial automata*) lingers unstated in von Neumann's work, Kemeny pulls no punches:

> This article attempted to show that there is no conclusive evidence for an essential gap between man and a machine. For every human activity we can conceive of a mechanical counterpart. … we still have not answered the question whether man is more than a machine. The reader will have to answer that question for himself (Kemeny 1955).



VON NEUMANN MACHINE is theoretically capable of reproducing itself. This is a highly simplified diagram of its conceptual units. The darkest squares are the "nerve cells" of the "brain." The next lightest squares are "muscle cells." The next lightest are transmission cells. The crosshatched squares are the "tail" which bears the instructions of the machine. The double hatching represents an "on" signal; the single hatching, an "off" signal. The empty squares are units of the environment which the machine manipulates. The arrows indicate that instructions are coming from the tail, on the basis of which the brain instructs a muscle cell to act on its surroundings. The machine has sent out a "feeler" to the left.

Figure 4.8: Annotated cellular figure from Kemeny's 1955 Scientific American *article (Kemeny 1955).*

In the figure above we see a heavily annotated cellular diagram. Cell shading, arrows, outlines, and textual descriptions—for example "nerve cells," "brain," "muscle cells," "tail," "feeler"[27]—are used to scaffold the viewer's perception of this abstract lattice as a mechanistic organism. The outline, like the labels, is not a part of the mathematical system itself, but

[27] Note that Kemeny has bracketed these labels in quotes, just as Schelling bracketed his labels (Schelling 1971; chapter 3).

is used to distinguish the parent system from its environment. Out of the primitive cellular soup of 29 states a simple creature is created, one constituted by mathematics and discursive gloss. Visual diagrams of the cellular space form an essential part of the representation. The figurative advantage of cells (squares, as in Chess) over points (lattice interstices, as in Go) is also evident.

The vital role of discursive practice in constituting simulation is made plain. Descriptions enable the interpretation of these cellular models as representations of organic phenomena. Purification, the teasing apart of the abstract processes of life from their physiological materiality, in turn demands descriptions that reintegrate immaterial mathematics with the organic processes they stand for. Why does Kemeny describe the cellular model in such vivid organic terms? It does more than explain and make accessible the project to a popular audience. The descriptions are constitutive of the entire modeling endeavor, linking vacuum tubes, automata, and cellular states to neurons, organisms, and the mystery of biological reproduction.

Kemeny's descriptive figures and text also reveal how amenable cellular automata are to such annotation. The *discreteness* of cellular automata offers a list of distinct parts (states), places (cells), and actions (transition functions) to name and describe. The *abstractness* of these materials invite storytelling on the part of their authors and observers, reflecting an underlying representational flexibility. The *spatiality* of two dimensional cellular universes further invites interpretation and narrative. Like Schelling's segregated neighborhood models, which function as evocative visual analogues of segregated urban geography, the forms of these cellular systems are both visually rich and abstract, and function in effect like ambiguous Rorschach forms—you can see in them what you want, and they suggest and lend themselves to diverse interpretations. I see in Kemeny's illustration the figure of a brain. The authors and players of cellular universes would come to make and see phenomena such as crystal formations, urban neighborhoods, dungeons[28], Turing machines, geological processes, and more. Abstract and evocative visual materials combined with visual and textual glosses constitute a powerful representational medium. And their flat two dimensionality means that stories and snapshots of these worlds can easily be embedded in magazines, articles, and computer screens (Latour 1986).

Foundational Abstractions

To summarize, von Neumann's interest in exploring the isomorphisms of computation and life led him to seek a mechanistic model of self-reproduction. He began with two foundational abstractions:

1. *Decomposition.* Decompose complex natural phenomena into identical functional units.

2. *Virtualization.* Black box these elementary units, sealing up the messy details of the natural world.

These abstractions focused attention on how identical elementary units combine to compose complex natural phenomena. The atomic elements were conceived of as machine parts floating on a lake. But this *kinematic* model was then replaced with a *cellular* model proposed by von Neumann's friend and colleague Stanisław Ulam. Ulam introduced a third foundational abstraction:

3. *Cellular lattice.* A cellular space composed of locally interacting automata. The automata are discrete in space, time, state, and transition functions (action). This fully

[28] *Dandy* (1983).

discrete universe was motivated by a desire to clean up the kinetic model, and make it more amenable to mathematical and logical analysis.

This gave rise to an enactive representation that von Neumann used to prove the possibility of mechanically self-reproducing creatures. Using 29 cellular states, he modeled an artificial universe into whose substrate higher order automata/creatures could be embedded. The pretense of this model, that groups of cells were multicellular organisms, was sustained by discursive practices such as illustrations and descriptions. The simulation practice emphasizes, by design, how simple atomic elements combine to form new and complex wholes.

In theory, the 29 state von Neumann automaton could even be run on a computer. The computational lattice was ideally suited to such automated simulation. And that is exactly what happened next. Catalyzed by increasingly widespread access to computers, newcomers expanded the practice, transforming cellular automata into a generalized medium for simulation. The abstractions which underpinned the cellular automata enterprise meant that a huge range of phenomena could be represented. Computer aided exploration fundamentally changed the nature of these explorations. Conceived of at the dawn of modern computation as a thought experiment, cellular automata blossomed when brought to life on high speed digital computers connected to visual displays, bringing players for the first time in direct contact with their synthetic worlds. A multitude of cellular automata universes came to life, transfixing a younger generation with their dynamics and representational possibilities. These newfound miniature worlds harbored their own laws of nature, and beckoned researchers and players to explore and create them.

The Universe and Everything

The next stage of the development of cellular automata would be catalyzed by two forces, Arthur Burks, who had edited and brought to publication von Neumann's unfinished manuscript on the subject, and the increasingly widespread availability of computers for simulating them.[29] Not only would the kaleidoscopic diversity of the natural world be represented with cellular automata—a straightforward generalization of von Neumann's work—but cellular automata would in themselves become worlds worthy of study, and some would ask whether the real world might in fact be some kind of cellular automaton.

The foundational abstractions constituted a powerful schema for seeing and thinking about the world. Cellular automata revealed themselves to be enactive representations well suited to modeling the natural world and to automatic computer simulation. Brought to life on computers, they delighted, excited, and entertained. These qualities are evident in the applications

---

[29] In 1949 Burks founded a research lab at the University of Michigan that would eventually come to be known as The Logic of Computers Group. Funding came from the Burroughs Company, and Burks steered the group towards research topics that bridged the biological and computational (Levy 1992, p. 59).

Under Burks's aegis, University of Michigan became a hot spot of cellular automaton research. Burks was the doctoral advisor to John Holland, a pioneer in the field of genetic algorithms—the seeds of which are found in Holland's work with cellular automata. Holland, in turn, was the doctoral advisor to another Univ. of Michigan graduate: E.F. Codd, best known for inventing relational databases, but whose dissertation project—funded by IBM, his employer—was a refinement of von Neumann's cellular automaton (Codd 1965; Codd 1968). Tommaso Toffoli's 1977 dissertation from the University of Michigan on cellular automata engendered further work on cellular automata as models for the basic laws of physics—the universal fundament as a computer (Toffoli 1977; Levy 1992, p. 61). Burks suggested to Christopher Langton, who was interested in and would eventually become known as a pioneer in the field of artificial life, that he study Codd's refined version of the von Neumann automaton, and in 1982 Langton joined the Logic of Computers Group at University of Michigan, where he completed a PhD in 1991.

Research was catalyzed by the use of computers, which were for the first time used to simulate cellular automata systems at Univ. of Michigan (Codd 1968; Burks 1970; Levy 1992, p. 60). The use of computers as a catalyst is best illustrated by E.F. Codd's work, who refined von Neumann's automaton through the use of computer aided design and simulation tools.

The difficulty of the posthumous von Neumann 1966 text is attributable in part to the fact that Burks intended the manuscript as a historical document reflecting von Neumann's creative process. "But while this contributes to the historical value of the manuscript, it also makes it exceedingly involved and difficult to follow" (Burks 1970, p. iiv).

they would be put to, which belong to three traditions of cellular automata craft:

1. *Representational*. Many early cellular automata practitioners extended and simplified von Neumann's self-replicating machine proof, but they eventually turned to new domains, and used cellular automata to represent both natural and man-made phenomena, from sand dunes to cities.

2. *Microcosms.* Computer simulation called attention to the surprising ways in which cellular automata systems autonomously unfolded. Researchers such as Stanisław Ulam, Stephen Wolfram, and Christopher Langton focused on cellular automata as microcosms, experimental models whose dynamics could be used as a proxy for understanding the complex and emergent nature of the natural world.

3. *Cosmogenic.* Some researchers, most notably Ed Fredkin, argued that the natural world might be, in fact, a cellular automaton. Physics and chemistry, in this worldview, are simply epiphenomena of a cellular automata substrate.

Representational

Early representational uses simply extended von Neumann's original automaton, simplifying and streamlining it. Using the computer as a design aid, Edgar Codd simplified von Neumann's automaton, reducing the number of states from 29 to 8 (Codd 1965).[30] This simplification

[30]A "sheath" layer of cells enveloped and directed signal transmission along a conductive core of cells. Through the combination of conductive and sheath cells, signals could be directed, split, and logically combined; tasks which had been performed by specialized states in the von Neumann construction.

Edwin Banks, in his dissertation, performs a 3-state simplification (Banks 1971). His advisor is Edward Fredkin, a proponent of digital physics, who I will discuss more fully later. He committee also includes Marvin Minsky, and Banks is well aware of Konrad Zuse's cosmogenic work (Zuse 1969/1970).

stems from his modeling the world at an even lower level of abstraction.[31]

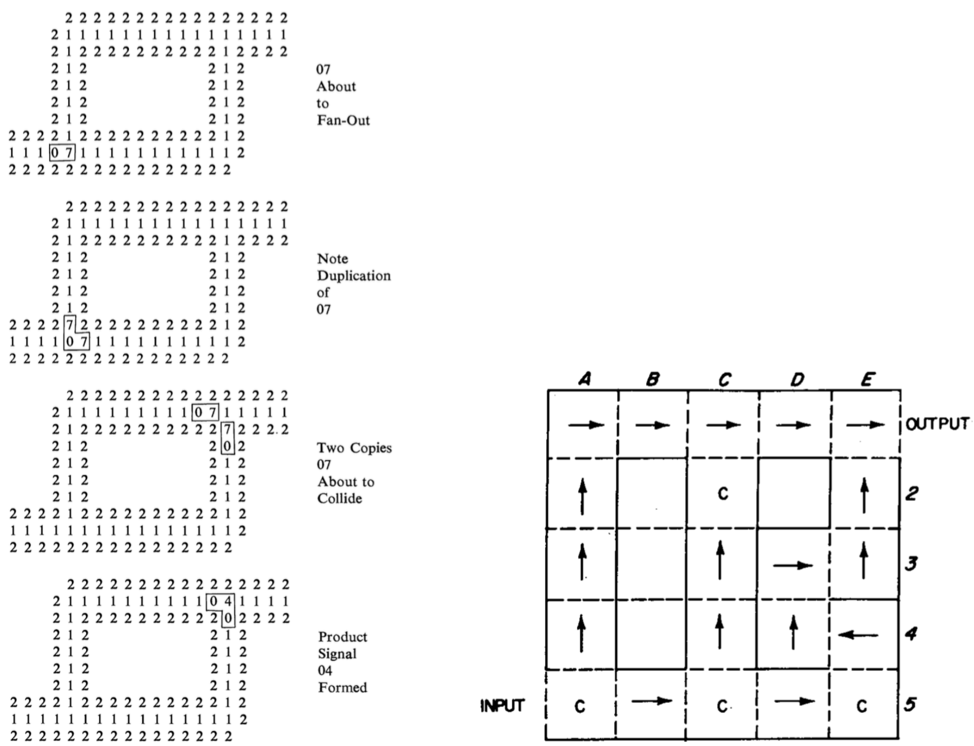


Figure 4.9: Note how in the figure at left, a sheath structure (state 2) directs signals down a homogenous conductive cellular core (state 1) (Codd 1968). Contrast to figure at right, an example from von Neumann's 29-state machine, in which distinct cell states (←→↑↓) indicate the direction of transmission (Burks 1970, p.13).

Interactive computer simulation enabled Codd to rapidly explore possibilities and make surprising discoveries. He credits the computer with making his research more creative, exciting, efficient, and pleasurable.[32] Moving beyond the purely proof based tradition charted by

[31] "[W]hereas von Neumann took disjunction and conjunction as primitives and synthesized negation, Codd was able to synthesize all switching functions from operations which, in a cellular framework, are more fundamental" (Burks 1970, p. 62).

[32] Burks ascribes Codd's success in building a more elegant model to the use of a computer, which assisted in his work. "The most striking case of this assistance," according to Burks, was Codd's accidentally discovery of "the echo phenomena," which sensed the state of a cell by bouncing a signal off of it (Burks 1970, p. 63). Codd describes a man-machine collaboration in which the computer performed the tedious calculations and bookkeeping of simulation, freeing him to focus on the creative work, formulating rules and adjusting tactics and strategy (Codd 1968, p. 107). "Rapid interaction between man and machine," wrote Codd, "tends to develop the intuition and perceptivity of the experimenter. The immediacy of the machine's response has the effect of

von Neumann, cellular automata become an experimental and *empirical* endeavor.[33]

Cellular automata offered a schema into which the world could be mapped. As a representational technique, cellular automata were flexible, suggestive, and powerful enough to model more than asexually reproducing machines. They could, Burks believed, be used to model a huge variety of "natural systems." Burks and his colleagues used them to study vibrating membranes, heat flows, diffusion processes, and neural nets, and he speculated that they might one day be used to study evolution. Cellular automata, it seemed, conformed to the information processing structures of the natural world; through the lens of cellular automata,

---

maintaining a high level of cerebral activity in the experimenter." Working by hand "would be agonizingly slow and the experimenter would very likely decide to turn his attention to other problems" (Codd 1968, p. 110). Computer simulated cellular automata, in other words, was not tedious and boring, as it would be by hand, but was an exciting and creative endeavor peppered with fortuitous discoveries. At first these online experiments were done via keyboards and printers (Codd 1968, p. 107), and later with displays and light pens (Burks 1970, p. 64; Levy 1992, p. 60). These man-machine combinations didn't just reduce manual simulation labor, but offered new ways to play with cellular automata. Time could be rewound, the rules changed, and then played forward again (Burks 1970, p. 64). Also, the computer was used to simulate cellular automata universes without fully specified rules. The simulator would stop if it encountered a state without a transition rule, notify the experimenter, and continue once it had received an updated rule set (Codd 1968, p. 107). In this way, the rule set for a cellular automaton universe could be improvised as a particular world evolved.

[33]Although Codd had used a computer to simulate his cellular automata, like von Neumann he never actually specified in full a self-reproducing machine—he designed the various logical units and then proved that they could be combined for the intended goal. And whereas the computer formed the intellectual backdrop of von Neumann's thought experiment, it was now an essential tool in furthering the research project. Codd's work bridges the experimental—simulation based explorations of cellular automata—with the proof based tradition charted by von Neumann.

the natural world appears to be a spatial information processing system (Burks 1970).[34]

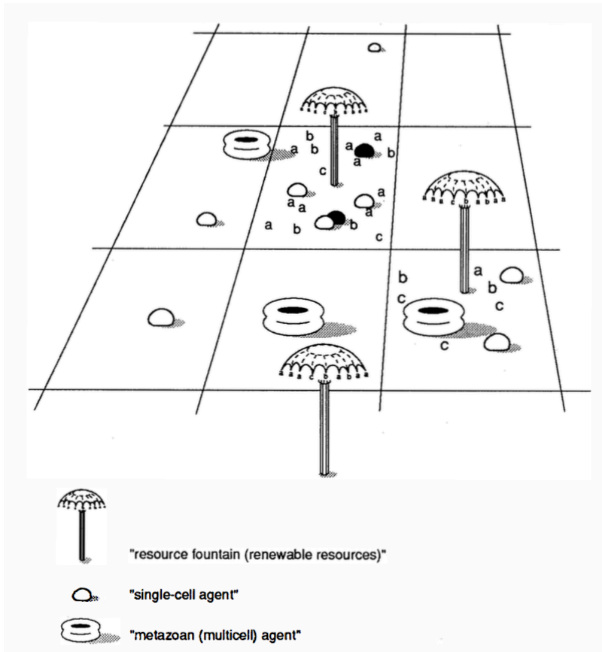


Figure 4.10: Illustration from Holland's Adaptation in Natural and Artificial Systems. *Schematic representation of a landscape of evolving creatures (Holland 1975/1992).*

Inspired and scaffolded by their mimetic qualities as models for life, Burks's former Ph.D. student John Holland established the field of *genetic algorithms*. While the enactive representa-

[34]Burks was attuned to the broader representational possibilities of cellular automata, beyond the initial problem of self-replication von Neumann had focused upon.

> Many natural systems can be fruitfully approximated and simulated in a cellular framework. The chief theoretical restrictions are two in number. First, the natural system to be studied must be governed by a local law … This precludes the use of cellular automata for the simulation of Newtonian gravitational systems … [i.e.] action-at-a-distance … Second, the behavior of the system to be simulated cannot depend on essential discontinuities … However, there may be no such natural systems (Burks 1970, p. 53).

For Burks, the partial differential equations used to study a "vibrating membrane or heat flow or diffusion processes"—or, we can add, the fluid dynamic processes that engaged von Neumann's attention, such as weather forecasting or nuclear detonation—"by handling them in a discrete grid" were all in the family of cellular automata systems. Researchers in Burks's group used cellular automata to study "neural nets" and "information processing by heart tissue," and he speculated that they "will be very useful for simulating evolutionary systems" (Burks 1970, p. 53). A key motivation behind von Neumann's study of cellular automata, according to Burks, is that they "are closer in structure to actual informational processing systems" while retaining the handy abstract and formal qualities exhibited by Turing machines (Burks 1970, xvi).

tions of cellular automata would eventually be left behind, it was their representational force, biological associations, and von Neumann's original agenda which scaffolded the idea of computationally modeling an evolving population of organisms in a space, leading, in turn, to the mature computational abstraction of genetic algorithms. Cellular automata provided the metaphorical and representational light which illuminated the path from von Neumann's computational biology to Holland's genetic algorithms (von Neumann 1966; Burks 1970; Holland 1959; Holland 1960; Holland 1962; Holland 1968).[35]

---

[35]Holland would eventually leave behind the computationally represented petri dish, as the spatiality of cellular automata was a complex encumbrance in the overall scheme of genetic algorithms—systems in which population of solutions were scored, selected, mutated, and bred (Holland 1975/1992).

This idea had always been in von Neumann's mind:

> [C]an the construction of automata by automata progress from simpler types to increasingly complicated types? Also, assuming some suitable definition of "efficiency," can this evolution go from less efficient to more efficient automata? (von Neumann 1966, p. 92; quoted in Burks 1970, p. xxiv).

This work was carried on by John Holland, a pioneer in the field of evolutionary computing—the use of "genetic algorithms" to find optimal solutions to problems. Holland earned his Ph.D. with Burks at the University of Michigan, becoming a professor there and serving as Codd's dissertation advisor. Holland envisioned a souped up cellular automaton called an "Iterative Circuit Computer" (ICC), where the steps of a program were inscribed as a path in two dimensional cellular space. A population of programs could thus be embedded in space, and leveraging the parallelism inherent to the cellular automata paradigm, run simultaneously and interact with one another. These program populations, in Holland's speculative computer science writings, represented a population of competing organisms or parts thereof, and were subjected to evolutionary pressures and recombinations. Trading deeply on the schematic yet deeply suggestive representation of a space teeming with reproducing organisms—a representational quality inherent to the original enterprise of cellular automata—Holland extended the mimetic quality of cellular automata further, adding a billiard ball physics model in which organisms could move about and come into contact with one another (Holland 1959; Holland 1960; Holland 1962; Holland 1968).

Like the cellular automata tradition from where it came, genetic algorithms sat at the fringe of computer science. It was a late bloomer, eventually becoming a mainstream area of study (Holland 1975/1992).

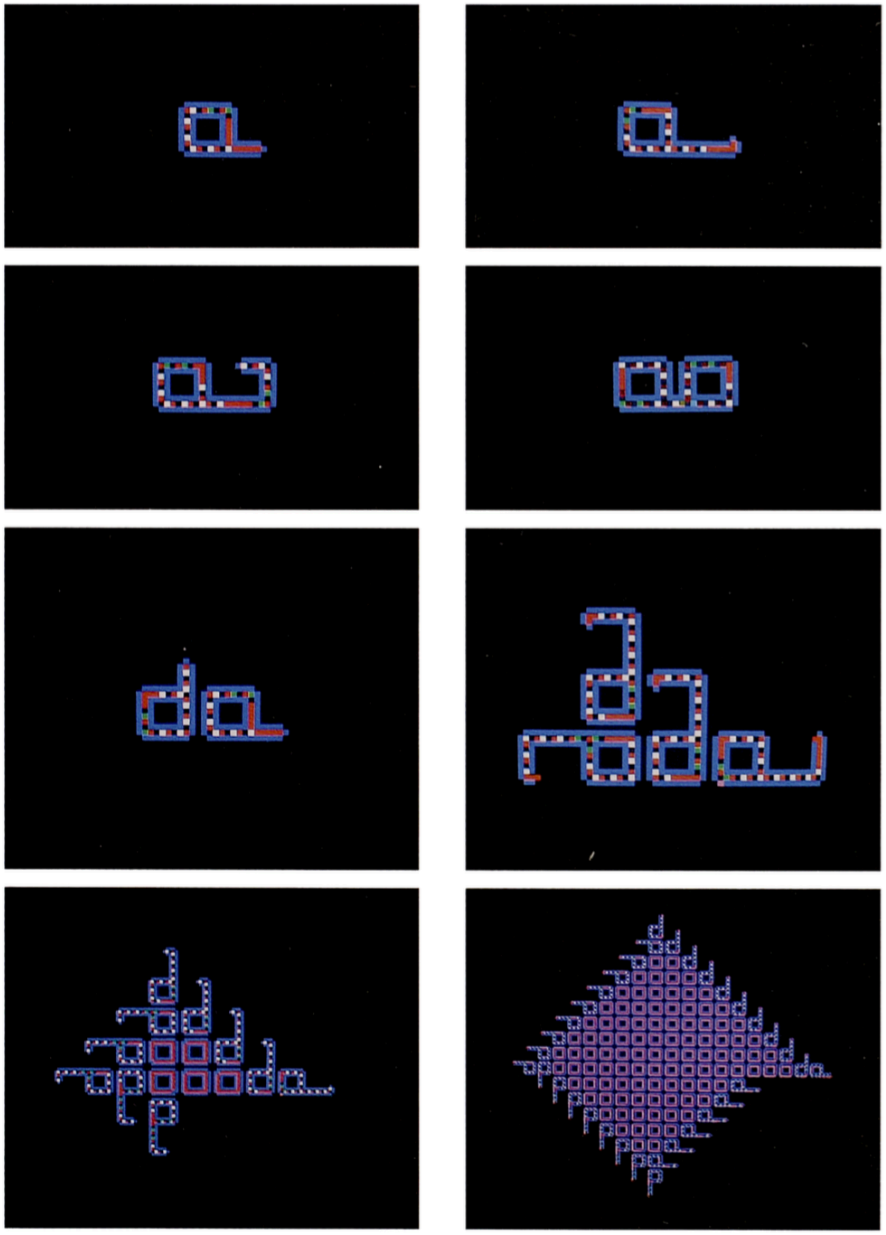


Figure 4.11: Langton loops replicating. One loop is a sheath (the blue outlines) encasing the information that directs the activity of the loop. The debt to Codd's sheath idea is visually apparent (Langton 1986, p.140).

The field of *artificial life,* whose prime exponent was Christopher Langton, was also inspired and grounded in cellular automata. Like Wright and many others, it was John Conway's *Game of Life* that brought Langton to cellular automata and the world of Burks (Langton 1984; Levy

1992).[36] Captivated by the goal of designing self-reproducing structures, he designed "Langton loops," simple structures that reproduce themselves.[37] Their visual forms are evocative of life, suggesting basic living beings such as cells, worms, and coral. His virtual ant model, or "vant," was similarly evocative.

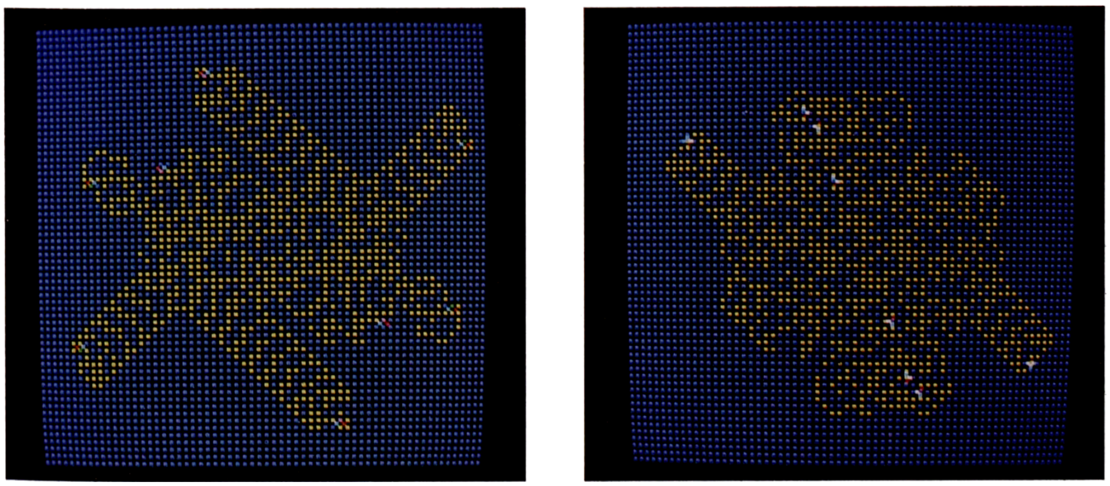


Figure 4.12: Langton's virtual ant, or "vant." One or more very simple ant agents move about a cellular world, resulting in complex and surprising results (Langton 1986, p.139).

Cellular automata were also used to simulate the physical world more broadly. Tommaso Toffoli (another University of Michigan Ph.D.), wrote that

> the importance of cellular automata lies in their connection with the *physical world.* … Von Neumann himself devised cellular automata to make a reductionistic point about the plausibility of life being possible in a world with very simple primitives … But even von Neumann, who was a quantum physicist, neglected completely the connec-

[36] Langton's overarching interest in modeling life with computers led him to the work of von Neumann and Burks. Langton entered the University of Michigan as a Ph.D. student in 1982, earning the last doctorate from Burks's Logic of Computers Group, with Holland as his advisor, and working as Burks's teaching assistant (Levy 1992, pp. 97–). Burks suggested to Langton that he look at Codd's simplified eight state model, which Langton further simplified by reframing the problem. Langton refocused the problem onto the designing self-replicating structures only, dropping the requirement that the structures be Turing complete (Langton 1984). In addition to doing foundational work with cellular automata, Langton organized the first "Workshop on the Synthesis and Simulation of Living Systems" at the Los Alamos National Laboratory in 1987, which became known as "Artificial Life I." The conference spawned a series of successors, and continues to this day.

[37] Whereas von Neumann and Codd's models hadn't in full been realized in computational form, and were in large part theoretical proofs, Langton's designs were small and simple enough to be fully simulated.

tions with physics—that a cellular automata could be a model of fundamental physics (quoted in Levy 1992, p.61-62).

With a newfound orientation towards cellular automata as objects of *empirical* rather than *formal* study, the stage was set for new types of models to be made.[38] With computer simulation, cellular automata were used to explore the evolution of crystal growth, spiral galaxies, snowflakes, sand dunes, tumors, infection, ecosystems, migration, chemotaxis, bacterial colonies, atmospheric pollution, gases, alloy formation, material phase transitions, cooperation, artificial societies, economics, and urban development. These are just a handful of applications gathered from various surveys (Wolfram 1983; Levy 1992; Ball 2001; Ganguly et al. 2003).

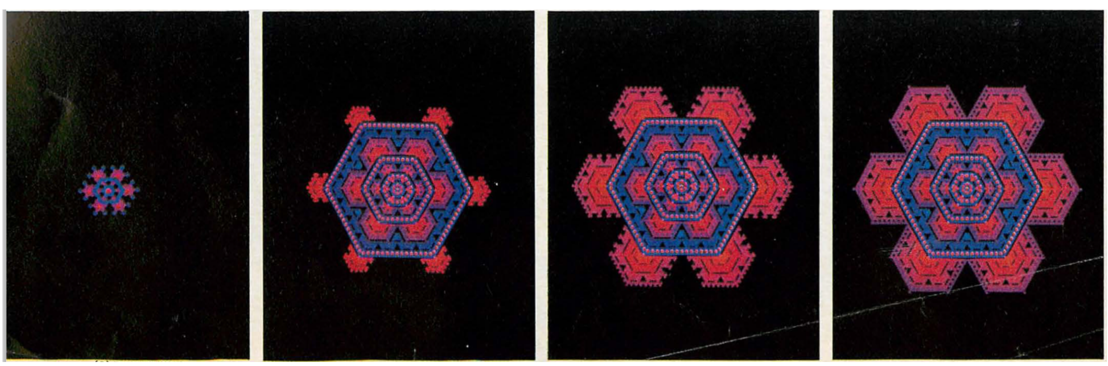


Figure 4.13: Norman Packard's snowflake model. Figure adapted from (Reiter 1984); photographs by John Madere.

Microcosms

All of these blossoming worlds—biological, physical, social—were trackless miniature universes, computationally animate microcosms open to scientific inquiry and exploration. Just as laboratory rats and mice can function as models of humans, cellular automata became

[38]Unlike the original approach taken by von Neumann, which proceeded as a logical proof, this work is grounded in computer simulation. Burks describes simulation based cellular automata research as "heuristic," connecting it to "the traditional hypothetical-deductive-experimental method of science" (Burks 1970, p.xviii–).

models of an emergent cosmos. Emergence was rooted in von Neumann's foundational abstractions which, after all, framed complex behavior as emanating from the interaction of elementary units. In the new craft of cellular automata as microcosm, naturalistic representation doesn't disappear entirely, but hovers in the periphery, linking the abstract dynamics of cellular automata to the abstract dynamics of the natural world.

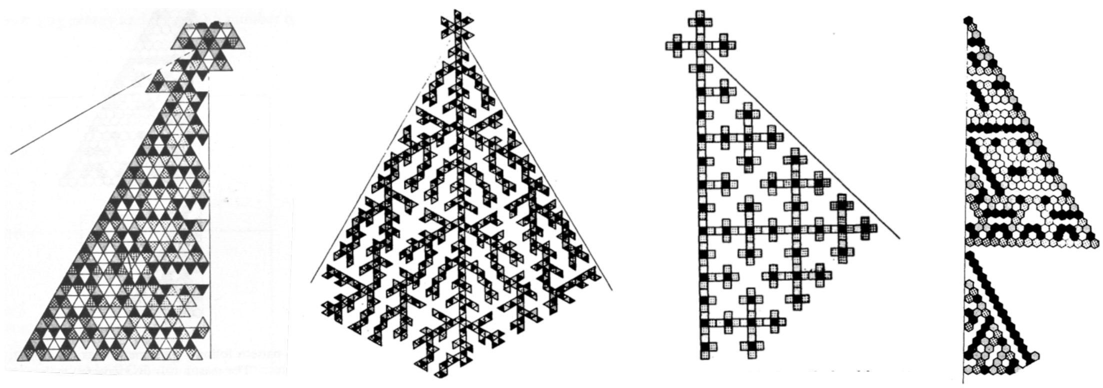


Figure 4.14: Pattern growth with triangular, square, and hexagonal cellular automata (Ulam 1962; in Burks 1970).

A precocious example of studying the abstract mechanics of cellular automata universes was the simulations made by Stanisław Ulam and his collaborators.[39] They were surprised

[39]Another notable result of exploring cellular automata as intrinsically interesting mathematical systems is E.F. Moore's characterization of "Garden-of-Eden" configurations. This term, suggested by John W. Tukey (who, incidentally, is also credited with inventing the term "bit"), describes a state of a cellular automaton that "cannot occur except at time *T = 0*. … there is no configuration at time *T - 1* which will give rise to the given configuration" (Moore 1970, revision of Moore 1962, collected in Burks 1970, p. 195; Gardner 1971). But this is still in the older formal, rather than the new empirical mode of inquiry made possible by the computer. Moore also offers a nine cell neighborhood (diagonals are included) as a generalization of the five cell neighborhood defined by von Neumann. These 5 and 9 cell neighborhoods are sometimes referred to as von Neumann and Moore types (e.g. Toffoli and Margolus 1987b).

and delighted by the complexity and richness which emerged from very simple rules.[40]

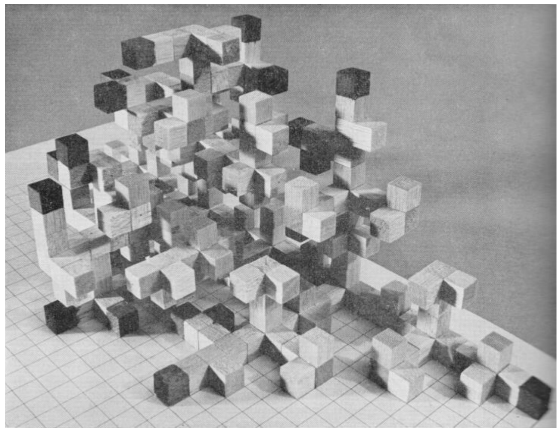


Figure 4.16: Photograph of a three dimensional model generated by a 3d cellular automata. The pattern grew from a single cell over 30 generations (Schrandt and Ulam 1967; in Burks 1970).

These experiments are also a precocious example of cellular automata as plaything. The simulations had the scent of the more rigorous studies to come, but lacked a clear method and produced few theoretical insights that a mathematician or computer scientist would recognize

Figure 4.15: Five and nine cell neighborhoods (Langton 1990). A 1d CA is effectively a three cell neighborhood.

[40]Stanisław Ulam, a collaborator of von Neumann's and originator of the cellular automata formalism, did this work in conjunction with J. Holladay and Robert Schrandt. These experiments studied how complex figures grew in space via the iterated application of simple rules. Simulations were done in two- and three- dimensions, and with tessellating spaces composed of square, cubic, triangular, and hexagonal cells (Holladay and Ulam 1960; Schrandt and Ulam 1960; Ulam 1962).

Many remarkable forms emerge which the authors compare to crystals, and whose complexity they observe to hover between the inorganic and organic. The emergence of complex and aperiodic form out of simple rules is clearly what holds the authors' attention:

> Our examples show both the complexity and the richness of forms obtained from starting with a simple geometrical element (a square or a cube!) and application of a simple recursive rule. The amount of "information" contained in these objects is therefore quite small, despite their apparent complexity and unpredictability (Schrandt and Ulam 1967; in Burks 1970, p. 233).

as a meaningful scholarly contribution. Their trial and error simulation experiments were akin to recreational mathematics, a simulation adventure pursued for its own joyful and autotelic purposes. Not only was the work of limited use, it appropriated and subverted valuable computer time on the MANIAC computer at Los Alamos intended for serious endeavors like nuclear weapons research.

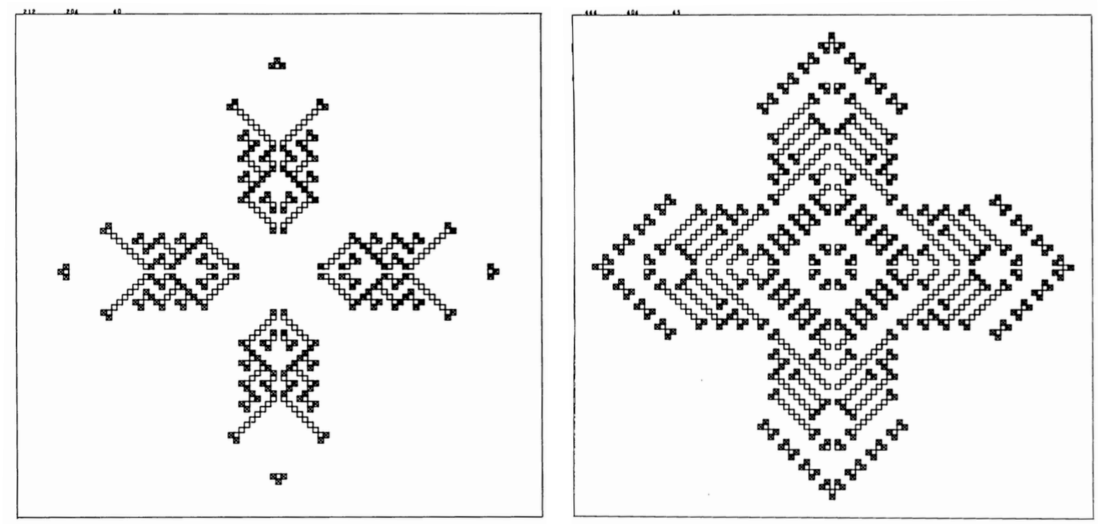


Figure 4.17: "Patterns grown with an erasure rule" (Schrandt and Ulam 1967).

They played games, and discovered a phenomena that anticipated Conway's *Game of Life.* By allowing cells to switch off, or die, they found that they could make multicellular figures appear to move, split apart, and reproduce. A startling sense of aliveness appeared.[41]

---

[41] After experimenting with growing these delightfully complex forms—figures which grew into intricate space filling patterns, but otherwise remained static—a new rule was introduced that infused the figures with dynamic activity: previously grown cells could be switched off, or die.

> This makes the figure move in the plane. … We show some cases of such motion, with occasional splitting of the figures into separate connected pieces. In some cases these figures are similar to the original ones and so we have phenomena both of motion and self-reproduction (Schrandt and Ulam 1967; in Burks 1970, pp. 232-233).

In a 1967 paper glossed with the language of living organisms, Schrandt and Ulam describe discrete time steps as "generations," cells as potentially being in an "alive" state (versus empty or dead), and the "death" of cells. This language reflects a debt to the biological aspirations and commitments inherent to the cellular automata tradition, but also aptly describes the vivacious character of the phenomena they were witnessing. While von Neumann's

Their playful attitudes, combined with the lively activity they discovered, led them to describe "another amusement we tried out on the computers [as] the following game" where two figures embedded in a cellular automaton develop independently and compete for space "[which] gives rise to a game for survival or a "fight" between two such systems" (Schrandt and Ulam 1967; in Burks 1970, p. 233).

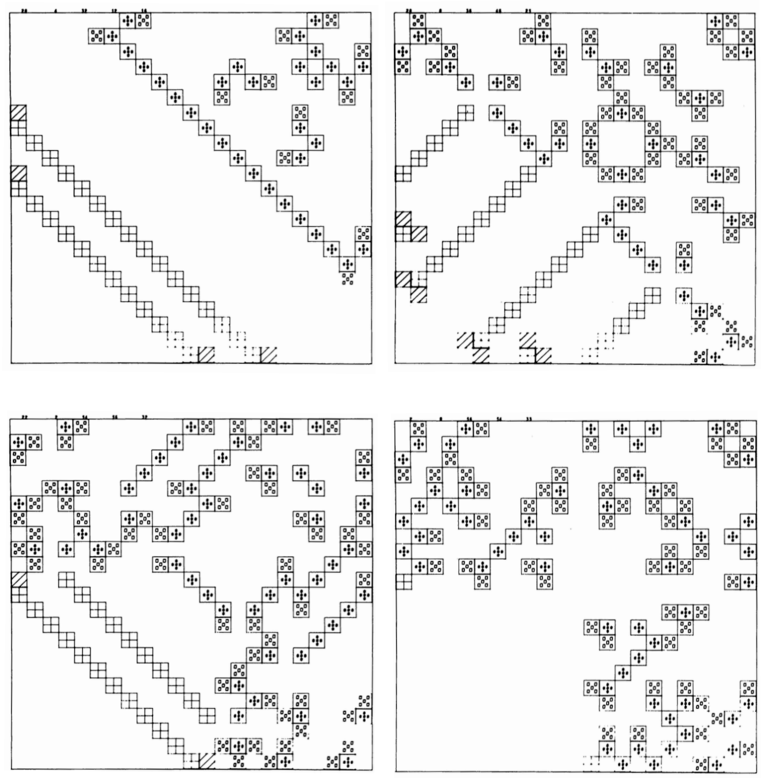


Figure 4.18: Multiple snapshots of a "game" in which two patterns "fight" for survival (Schrandt and Ulam 1967).

Cellular automata, so far, had existed in a kind of intellectual backwater. Sure, mathematicians such as von Neumann, Conway, and Ulam had played around with them, but they didn't yet have the respectability of serious mathematics or science. This changed with Stephen Wol-

notion of aliveness is grounded in the functional and logical self-reproduction of a complex automaton in space, the notion of aliveness here is very different, and constituted by a computer program whose dynamical activity evokes life.

fram's paper "Statistical mechanics of cellular automata" (Wolfram 1983), which brought a new kind of rigor and mainstream respectability to the study of cellular automata (Levy 1992; Wolfram 2002). Rather than a provocative thought experiment, plaything, or impressionistic and at times fanciful medium of representation, Wolfram linked cellular automata to the emerging study of complexity, bringing the endeavor closer to mainstream science.

Wolfram saw, in the swirling eddies and complex forms of these miniature universes, a model of the natural world's dynamics. If cellular automata could be used to represent so much of our universe, then perhaps their underlying statistical properties could tell us something about our universe's fundamental properties. This project wasn't about using cellular automata to represent the natural world—an organism, sand dune, or city—but rather using cellular automata as controlled laboratory models of complexity in general. "What is now needed," he wrote, "is a general mathematical theory to describe the nature and generation of complexity" (Wolfram 1983). Leveraging the core idea built into von Neumann's schema, that complex phenomena be studied as a result of the organization of simpler interacting units, Wolfram undertook a rigorous mathematical study of cellular automata.[42]

[42]Wolfram's study reveals both the debt to von Neumann's original work and the tradition of cellular automata it gave rise to, as well as an important shift in emphasis. What had begun as a stripping down of sophisticated naturalistic systems into elementary black boxes, and a study of their operational synthesis, transformed into something else. Elaborating on the kinds of computer experiments undertaken by Ulam and his collaborators, Wolfram focused on the question of how simple elements combine and give rise to complex forms characteristic of the natural world in general.

> It is common in nature to find systems whose overall behaviour is extremely complex, yet whose fundamental component parts are each very simple. The complexity is generated by the cooperative effect or many simple identical components. Much has been discovered about the nature or the components in physical and biological systems; little is known about the mechanisms by which these components act together to give the overall complexity observed. What is now needed is a general mathematical theory to describe the nature and generation of complexity (Wolfram 1983).

Wolfram's paper begins with a literature review of the diverse representations that use cellular automata. The representational generality of cellular automata is enlisted to help justify the claim that cellular automata are worthy of study, specifically in that they constitute a kind of general mathematical-computational substrate for modeling the world. This then motivates the study of cellular automata dynamics abstractly, which is the tack taken by Wolfram: cellular automata statistics are interesting because cellular automata can represent anything,

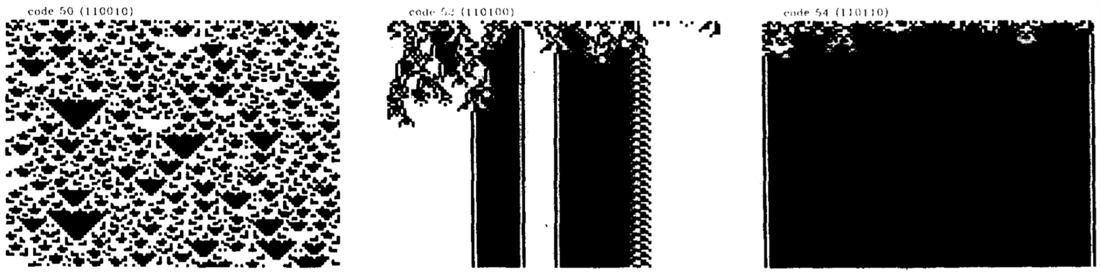


Figure 4.19: Examples of of 1d CA (Wolfram 1984). Each successive row is a different moment in time.

He abstracted even farther, looking at the evolution of 1d cellular automata. Their evolution could be shown in a single two dimensional image by plotting successive moments in time below one another.[43] And by limiting cells to one of two states (on/off), all possible transition rules could be encoded with 8-bit binary strings. Each of the 256 (2^8) possible rule sets could be referred to with a number, and one could speak, for example, of rule 90 or 28.

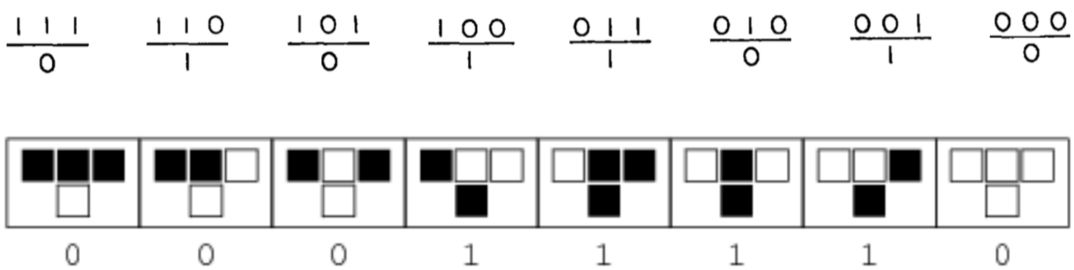


Figure 4.20: Depictions of 1d CA rule sets. Above image is from (Wolfram 1983). Lower image is rule 30 (0001110_2) from Wolfram's MathWorld web site, and is in the graphical style characteristic of his later work (Wolfram 2002). The idea, however, is the same.[44]

---

and therefore in speaking of their mechanics we speak, at some level, of mechanics which underpin the natural world. Here we clearly see at work the claim I made earlier, that representation in this second mode of analysis hovers at the periphery, motivating statistical work without engaging directly in representation.

[43]Wolfram undertakes a systematic exploration of a simplified variant of the 2d cellular automata discussed so far. Down shifting to 1d cellular automata, where cells only have neighbors at their left and right (a 3 cell neighborhood), afforded a static representation of a system's evolution, as snapshots of successive moments in time can be stacked vertically. The top row represents the initial conditions, with each of the rows beneath depicting successive iterations. Thus, each two dimensional image represents a time-series snapshot of an evolving cellular automaton.

[44]In a two state cellular automaton system there are eight possible combinations of a cell and its two neigh-

By distilling the cellular automata formalism to its simplest form, Wolfram was well situated to undertake empirical inquiries into the dynamical behaviors of *collections* of cellular automata systems, and systemically explore every rule—from 1 to 255. Furthermore, with 1d cellular automata one could produce images of a system's evolution and compare these images, something which was hard to do with 2d system.[45] Wolfram opened up new horizons in the study of cellular automata as microcosms of our world, but he had also made an important representational innovation. Time, space, and the laws of physics of these miniature worlds could now all be represented in visually schematic form.

Artificial life pioneer Christopher Langton built upon Wolfram's approach, asking what kinds of universes might be capable of supporting life. Arguing that life is predicated upon a material substrate which affords computation—"the transmission, storage, and modification of information"—Langton asked what kinds of cellular automata support computation and thus, in turn, life, finding that the answer lay in systems at the "edge of chaos," which he compares to the phase transition between "the *solid* and *fluid* phases of matter."[46]

---

bors, enumerated by each column in the figures above. In a column, the three values at top label the column's preconditions—a cell and its neighbors (it doesn't change across rule sets). At bottom, how the middle cell will transform. Thus, one can specify an entire rule set with an eight bit string which describes the new state for each set of preconditions.

[45] Evolutionary trajectories were of central significance to Wolfram's study, which explored the statistical properties of these systems, linking them to research into chaos theory. One result of Wolfram's research was the classification of cellular automata into four categories, and a correlation of these categories to types of dynamical systems used in the chaos theory literature (Wolfram 1984; for a concise summary see Langton 1990).

[46] Through a series of empirical experiments whose method mirrors Wolfram's, Langton randomly generated various 2d cellular automata systems based upon a parameter called λ. λ represents the frequency with which cellular automata transition rules go to the quiescent, or off, state. Higher λ values means cells have a higher likelihood of turning/staying on; lower λ values means cells are more likely to turn off. It is important to note that any particular cellular automaton system remains deterministic; λ is used to determine the makeup of randomly generated cellular automata systems. Langton found that when λ is low systems calm down into uninterestingly ordered blank steady states. When λ is too high the system boils over with chaotic activity. At the phase transition between these two states exists a kind of goldilocks zone in which "large spatial and temporal correlations" make possible the transmission, storage, and transformation of information. Computations are "special instances of the kinds of processes that occur in a physical system in the vicinity of a solid/liquid or a liquid/vapor transition."

> Perhaps the most exciting implication is the possibility that life had its origin in the vicinity of a phase tran-

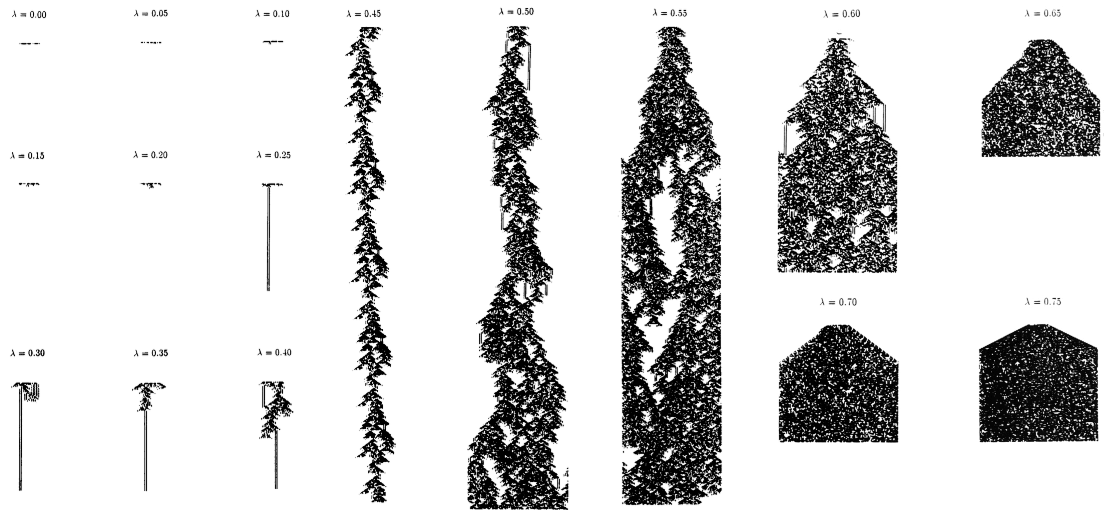


Figure 4.21: One-dimensional cellular automata with λ varying from 0 to .75. Note how when λ is close to zero activity dies out (at left). At high λ levels it boils over into dense noise (at right). Interesting activity arises at the phase transition, middle (Langton 1990).

Synthetic microcosms become laboratory models for studying the natural world. Wolfram and Langton find in the generality and dynamics of cellular automata evidence that the world is computational. The computational conclusion, however, is prefigured by the basic premise of cellular automata: the use of a computational abstraction to represent the natural world.[47] In all this there is an identification of basic physical laws with model cellular worlds. In the cosmogenic tradition, the third and last craft of cellular automata we will look at, the hypothesis that cellular automata might underly physical reality is given serious consideration.

---

sition, and that evolution reflects the process by which life has gained local control over a successively greater number of environmental parameters affecting its ability to maintain itself at a critical balance point between order and chaos (Langton 1990).

Langton, in exploring the relation between computation, complexity, and life, returns us back to the original von Neumann themes of organic life, albeit via a detour through complex systems theory.

[47] I don't think this apparent circularity dooms the enterprise. I believe that the application of a computational lens to organic life and the natural world at large is a fruitful mode of inquiry. It does seem to me, however, that that there is an element of proving the premises in this overall project that is worth noting. Von Neumann emphasized that the original computational abstraction he introduced was axiomatic. Wolfram and Langton probe this quality, but its axiomatic quality is inescapable.

Cosmogenic

The chief exponent of the cosmogenic tradition of cellular automata is Edward Fredkin. There are weak and strong versions of this argument.[48] The weak version argues that cellular automata can replace traditional mathematical representations of physics, such as differential equations, and is a straightforward elaboration of the microcosm tradition. In it, cellular automata are a new and improved mathematics and physics for representing the world. The strong version claims that the universe we inhabit is in fact a cellular automaton. In this vision, a cellular automaton lattice is the "engine of an informational process" that "runs a computation" in which our spacetime is embedded. This computation produces the "spacetime of physics," and stable structures embedded in the lattice correspond to elementary particles (Fredkin 2003).[49]

[48]Fredkin refers to these as digital mechanics and digital philosophy. Fredkin's Digital Mechanics (DM) project sought to replace the traditional mathematical models of physics with digital computer versions. The project is an instance of Fredkin's wider Digital Philosophy (DP) project, which takes space and time as discrete, and represents these with a cellular automaton.

[49]Fredkin even compares gliders in Conway's *Game of Life* to such elementary particles. Fredkin has suggested that gliders may be a model for subatomic particles as an epiphenomena of a cellular substrate (Wainwright 1974; Wainwright 2010; Berlekamp et al. 2004).

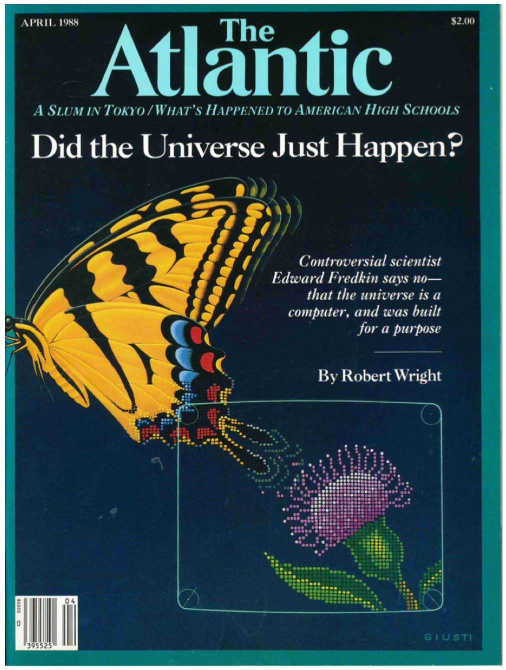


Figure 4.22: Cover of The Atlantic *April 1988 cover story on Edward Fredkin. Illustration depicts the natural world as a cellular automaton.*

Fredkin also espoused an even more radical cosmogenic idea, that our universe is itself an artifact, an "informational process" running on an "Ultimate Computer" made by unknowable and strange intelligences. This idea echoes the science fiction novel *A Hitchhiker's Guide to the Galaxy*[50], which projected a similar idea—that the Earth was itself a computer made by powerful beings to answer a question (Adams 1979; Wright 1988; Fredkin 1992; Chandler 2003).[51] If von Neumann's original work opened a passageway between computation and

[50]Originally a radio play.

[51]Why would beings powerful enough to build such a computer need a computer in the first place? Why couldn't they simply work out the answers themselves? Fredkin's answer is simple: the halting problem. Turing had proved that the results of some computations can only be predicted by actually performing those calculations (Fredkin 1992).

Fredkin writes:

> We should not be afraid to consider intellectual activity as the driving force behind the creation of the universe. By a close and quantitative examination of the possible parameters of Digital Mechanics, we can arrive at reasonable guesses as to what might be the purpose behind the creation of a universe like ours. That, in turn, can lead us towards intelligent speculations about Other, the space that contains the engine of our world.

reality, allowing them to be equated, here it has opened to its maximal size in the radical version of the cosmogenic argument: the cosmogenic fabric of our universe is produced by a cosmic supercomputer made by aliens.

Proponents of the more conservative, weak cosmogenic tradition—whose works appear in journals and books (e.g. Toffoli 1977; Margolus 1984; Toffoli 1984; Vichniac 1984; Wolfram 2002)—argue that cellular automata are not only viable models of physical reality, but potentially superior to the traditional differential equations used in its study.[52]

> If we assume that the Ultimate Computer was purposefully constructed in order to find an answer, we still have the possibilities that:
>
> 1. What we see as our Universe might be working towards that answer.
> 2. The Universe we know, in its entirety, might be an artifact.
>
> In either case, our existence here on Earth might or might not be completely incidental to the purpose (Fredkin 1992).

Adams's novel:

> "I speak of none other than the computer that is to come after me," intoned Deep Thought, his voice regaining its accustomed declamatory tones. "A computer whose merest operational parameters I am not worthy to calculate – and yet I will design it for you. A computer which can calculate the Question to the Ultimate Answer, a computer of such infinite and subtle complexity that organic life itself shall form part of its operational matrix. And you yourselves shall take on new forms and go down into the computer to navigate its ten-million-year program! Yes! I shall design this computer for you. And I shall name it also unto you. And it shall be called …The Earth" (Adams 1979).

[52]Proponents of the more conservative vision of digital physics argued that cellular automata afford more natural representation of physical reality than the symbolics of differential-equations (Toffoli 1984), better capture the stochastic atomistic nature of reality than differential-equations, whose dimensions were themselves already aggregate abstractions of nature's inherent atomicity (Vichniac 1984), and more efficient parallel computer designs. Wolfram's *A New Kind of Science* is prototypical of this line of thought (Wolfram 2002). Fredkin compares Newtonian motion to the historically unquestioned "vitalistic theories of life," arguing that conceiving of physics as an informational processes allows one to hide "nothing under the rug," and instead conceive of informational representations which give rise to basic physical phenomena such as motion and mass (Fredkin 2003, p. 245).

Fredkin, for a variety of reasons—some of which should already be self-evident, didn't fit neatly into the academic world in which he spent much of his career (Wright 1988; Levy 1992).

During Fredkin's tenure as the head of Project MAC in the early 1970's he became aware of the German computer pioneer Konrad Zuse's 1967 *Rechnender Raum*, which postulated similar ideas to those he had been entertaining about cellular automata computers simulating the universe. Zuse was invited to speak at MIT, where "he found the ideas in his book appreciated for the first and only time during his life" (Fredkin 2003). Project MAC commissioned an English translation of Zuse's text, entitled *Calculating Space* (Zenil 2012).

At MIT, Fredkin's "students and colleagues, including Roger Banks, Tommaso Toffoli, and Norman Margolus expanded and elaborated" upon the field of digital physics (Fredkin 2003). Toffoli and Margolus would design

The journey from von Neumann's vision of organic life as automata to an ever widening series of representations of the world reveals an extraordinarily flexible and evocative representational technique. A growing body of published examples and craft expertise guided new practitioners in applying the schema of cellular automata to an ever wider body of phenomena. The evolution of system dynamics revealed a similar story. The enactive representations, software, schema, and craft of cellular automata became powerful tools for seeing, thinking about, and modeling the world. The abstract cellular lattice, when combined with the imagination, descriptions, and diagrams of practitioners, came to stand for just about anything. And, as we have seen, the representation holds a deep appeal: models are exciting, lively, evocative, and dynamic. The story so far has revealed a glimpse of cellular automata as play, but it is Conway's *Game of Life* that will make this connection most clear.

The Game of Life

Conway's *Game of Life* is the most famous cellular automaton system. Not only did it popularize cellular automata, but it is the prototypical example of cellular automata and emergence in both popular and scholarly circles. Aside from inspiring and influencing Will Wright's computer play, it is the cellular automaton par excellence, manifesting within discussions about and catalyzing work in digital physics, emergence, complexity, artificial life, play, and cellular automata (e.g. Levy 1992; Eno 1996; Pearce 2001; Wolfram 2002; Fredkin 2003; Eno and Wright 2006; Adamatzky 2010).

Why is *Life* such an evocative object? It has been described as offering an "unparalleled

---

and market a relatively inexpensive and specialized piece of hardware called the Cellular Automata Machine (CAM) for personal computers that rapidly simulated cellular automata universes. This machine is one of many examples—practices, hardware, software, publications—which accelerated the propagation of cellular automata as a modeling practice.

combination of functional simplicity with behavioural complexity" (Adamatzky 2010). *Life* is simple, easy to grasp, yet exhibits all the seductive and surprising aesthetic characteristics of cellular automata. It is simple enough to be simulated by hand, with coins and a chess board, but its behaviors are surprising, hard to envision, and out of our predictive reach. Computer simulation amplifies the delightful distance between comprehensible rules and surprising outcomes. These qualities make *Life* an ideal vector for the ideas of cellular automata and emergence—a perfect prototype for Will Wright and others to pick up, catch the bug, and spread to others.

The patient zero of the *Game of Life* epidemic is the recreational mathematician Martin Gardner, who spread word of Conway's automaton in his "Mathematical Games" *Scientific American* column (Gardner 1970). Conway's mathematical accomplishments gave him the cover to spend extraordinary amounts of time on recreational mathematics. Enchanted by von Neumann's 19 state automaton, Conway spent two years trying to create a system that exhibited a sense of aliveness.[53]

[53]A British mathematician, Conway practiced both serious and recreational mathematics, although he rarely published his recreational discoveries (Gardner 1970). (The distinction between serious and recreational mathematics enterprises is ambiguous and itself a philosophically interesting question.) Conway's contributions to group number theory gave him the professional cover and flexibility he needed to engage in more recreational mathematical pursuits. Conway says that he has "spent a fantastic amount of my time playing childish games" (Levy 1992, p. 50).

In 1968 Conway became obsessed with cellular automata, and wondered if there existed a cellular automaton rule set simpler than von Neumann's original 29-state rule set, with its attendant colossal embedded Turing machine, which would still exhibit complex results. Conway also shared von Neumann's interest in aliveness, perpetuating the founding motivation of the cellular automata enterprise. "I wanted to see some self-reproducing animal … Displaying some interesting behavior. In a weak form, living." Finding the right rules took two years of tinkering, according to Conway, "experimentation at coffee time to get it right. And coffee time lasts all day here" (Levy 1992, pp. 50-52). Conway sought a system whose dynamic behavior would be unpredictable; patterns should sometimes fade away, settle into stable forms, oscillate, or appear to grow forever (Gardner 1970). In short, he wanted a dynamic system exhibiting a *sense of aliveness.* Conway successfully fulfilled von Neumann's vision of a formally modeled system with a quality of aliveness, albeit with a different sense of "alive". This search can also be thought of in terms of Langton's λ phase transition, a system which tends neither towards quiescence nor boils over into noisy chaos, but is balanced precariously between the two.

Conway goes for a sense of aliveness in the sense of lively dynamic activity, a sense also evident in the terminology chosen, but drops the idea of Turing machine—at least in the short term.

"Our younger readers won't have learned much about Life, so we'd better tell you some of the facts." So begins a description of the rules, with characteristic insouciance, in Berlekamp, Conway, and Guy's recreational mathematics compendium, *Winning Ways for Your Mathematical Plays, Volume 4*:

> Life is a "game" played on an infinite squared board. At any time some of the cells will be **live** and others **dead**. Which cells are live at time 0 is up to you! But then you've nothing else to do, because the state at any later time follows inexorably from the previous one by the rules of the game:
>
> **BIRTH.** A cell that's dead at time *t* becomes *live* at *t* + 1 only if *exactly three* of its eight neighbors were live at *t.*
>
> **DEATH by overcrowding.** A cell that's live at *t* and has four or more of its eight neighbors live at *t* will be dead by time *t* + 1.
>
> **DEATH by exposure.** A live cell that has only one neighbor, or none at all, at time *t*, will also be dead at *t* + 1.
>
> These are the only causes of death, so we can take a more positive viewpoint and describe instead the rule for
>
> **SURVIVAL.** A cell that was live at time *t* will remain live at *t* + 1 if and only if it had just 2 or 3 live neighbors at time *t* (Berlekamp et al. 2004).

They also give a shorthand version of the rules:

> Just 3 for BIRTH
>
> 2 or 3 for SURVIVAL

Counters, sheets of paper, and shells in hand, Conway and his colleagues explored the fantastic dynamics of this system. They discovered stable configurations they named after their representational resemblances: "block, ship, longboat, beehive, load, canoe, pond," as well as oscillating patterns they called "toads, blinkers, clocks, and traffic lights" (Levy 1992, p. 52).

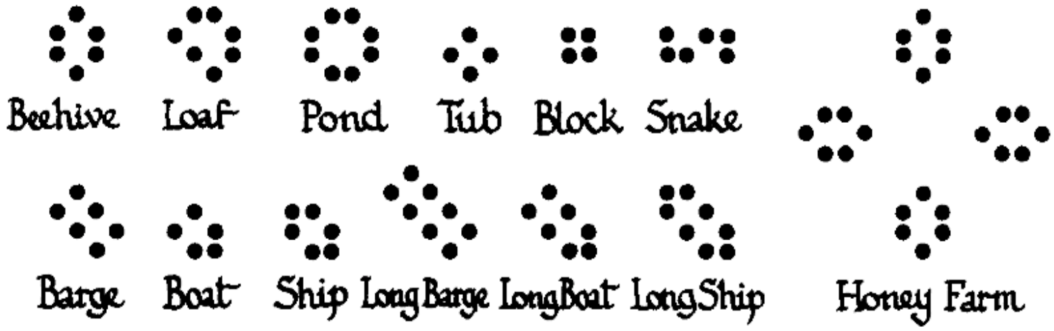


Figure 4.23: "Some of the Commoner Forms of Still Life." Stable configurations of patterns whose component cells will not switch to alive or dead (Berlekamp et al. 2004).

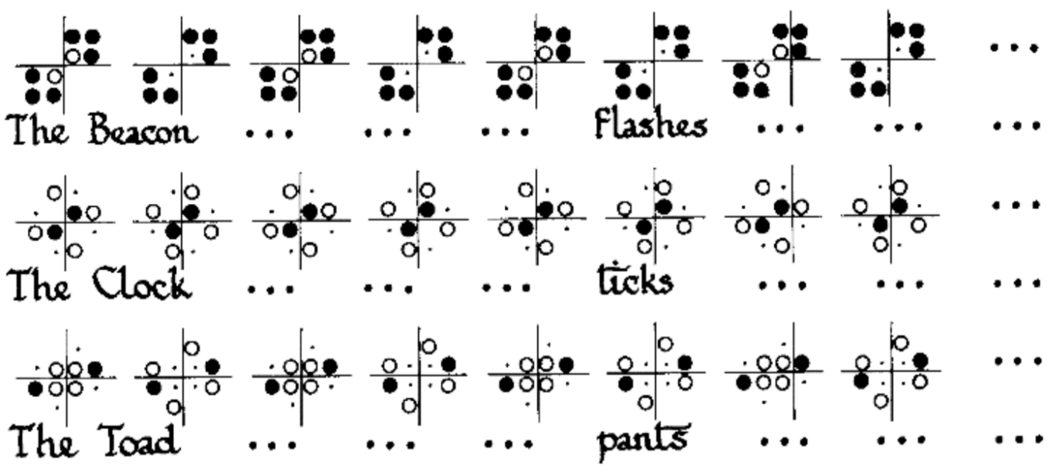


Figure 4.24: "Three Life Cycles with Period Two." Oscillating patterns (Berlekamp et al. 2004).

Richard Guy, another Cambridge mathematician, discovered the "glider," a configuration made of five living cells whose dynamics caused it to appear to amble through space (Levy 1992, p. 53).

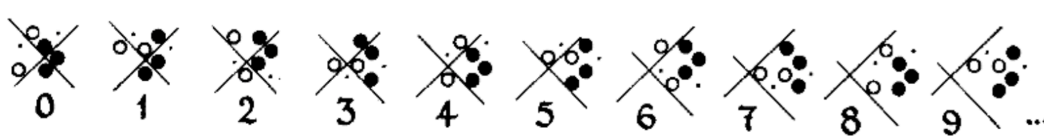


Figure 4.25: "The Glider Moves One Square Diagonally Each Four Generations" (Berlekamp et al. 2004).

The *Game of Life* universe, they knew, harbored many more secrets to be discovered. It was for this reason that Conway shared his "game" with Gardner and his broad readership, offering a $50 prize to anyone who could, Gardner wrote, "prove or disprove" the conjecture that "no pattern can grow without limit … before the end of the year." Such a pattern was needed to prove that a Turing machine could be embedded within the *Game of Life.*[54] With the prize offering, Conway recruited Gardner's readership into his project, a brilliant marketing maneuver that turned many people onto Conway's "game." Some of these people were equipped with computers, amplifying the *Game of Life's* pleasures, as well as their odds of winning the contest.[55]

[54]In order to embed a Turing machine within the *Game of Life*, the population of living cells would need to "grow without limit." While Conway and his collaborators had found patterns that vanished without a trace or oscillated at steady states, they had yet to find patterns that grew indefinitely. Gardner offered that to disprove the conjecture one could discover a pattern which acted as a "glider" "gun", a pattern which periodically emitting gliders, or a "puffer train," a moving pattern which left behind a trail (Gardner 1970; Wainwright 1974).

[55]Wainwright recalls visiting the Project MAC lab in the end of 1971 and seeing, for the first time, *Life* patterns rapidly unfolding on a screen (Wainwright 2010). This, of course, was a competitive edge held by the MIT team, and illustrates the power and appeal of cellular automata simulated by computer, in real time, and presented on a visual display. Wainwright, by contrast, submitted his *Life* simulations to an IBM mainframe, and received the output of his jobs the next day. Noting that his *Lifeline* newsletter "readership is definitely divided into two separate groups with large differences in exploratory capability," Wainwright asked his readers how much they would be willing to pay for a device that would connect to their television and play *Life*—as he had seen it done at MIT (Wainwright 1972).

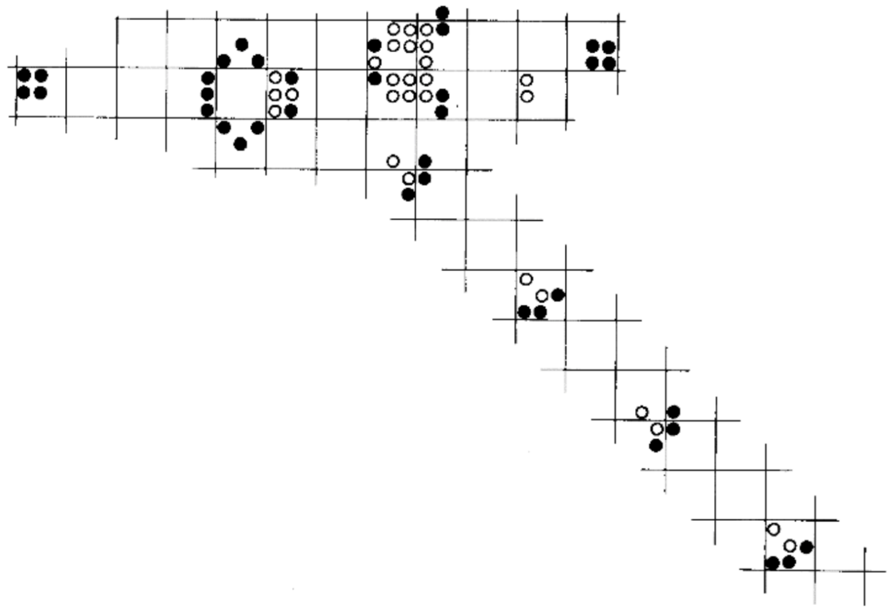


Figure 4.26: A glider gun occupies the top part of this figure. Moving diagonally towards the bottom right, a stream of gliders emitted from the gun (Berlekamp et al. 2004).

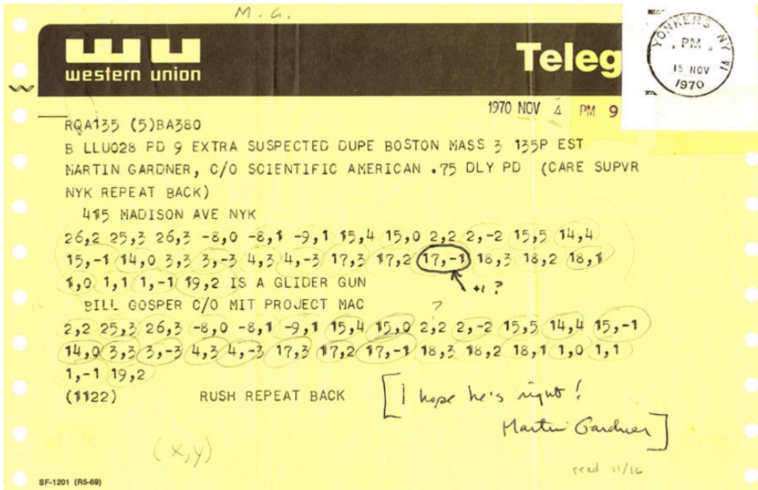
Gardner's readership responded enthusiastically, prompting a followup column, and a spinoff quarterly newsletter called *Lifeline* "for enthusiasts of John Conway's Game of Life."[56] The prize was claimed by Project MAC researchers at MIT, who discovered the hoped for glider gun (Gardner 1971).[57] The discovery connected the *Game of Life* enterprise back to von Neumann's original project, proving that a Turing machine, and thus a self-reproducing

[56]The account of Robert Wainwright, who self-published *Lifeline* (March 1971–September 1973), is indicative of the effect *Life* had on readers—especially those with access to computers. Wainwright found *Life* "mildly interesting" until he became "curious about the outcome of random patterns" and "located an old checkerboard and a small jarful of pennies to investigate this new game." Realizing that coins were an unwieldy way to explore the *Life*, Wainwright used an IBM mainframe to which he had access through work (Wainwright 2010).

The initial Gardner column "stirred such interest among computer scientists" (Gardner 1971) to warrant a followup column. Gardner reported to Wainwright that the *Life* column precipitated an unparalleled response from his readership, and that he had to convince the magazine editors to permit the followup article. With the blessings of Conway and Gardner, and equipped with the addresses of Gardner's interested correspondents, Wainwright started a quarterly newsletter. The subscription fee was one dollar a year, and the newsletter served as a hub for *Life* enthusiast activity, collecting and publishing their discoveries (Wainwright 2010). Enthusiast efforts continue to this day, see for example the web site *conwaylife.com*, which includes a repository of thousands of game of life patterns.

[57]Bill Gosper (given name Ralph William Gosper), using a program written by Michael Speciner, discovered a pattern that emitted gliders every 30 time steps (Gardner 1971).

machine, could be embedded within the minimalist *Game of Life* universe.


M.G.

western union

Teleg

YONKERS NY
PM
15 NOV
1970

1970 NOV 4 PM 9

RQA135 (5)BA380
B LLU028 PD 9 EXTRA SUSPECTED DUPE BOSTON MASS 3 135P EST
MARTIN GARDNER, C/O SCIENTIFIC AMERICAN .75 DLY PD (CARE SUPVR
NYK REPEAT BACK)
415 MADISON AVE NYK
26,2 25,3 26,3 -8,0 -8,1 -9,1 15,4 15,0 2,2 2,-2 15,5 14,4
15,-1 14,0 3,3 3,-3 4,3 4,-3 17,3 17,2 17,-1 18,3 18,2 18,1
1,0 1,1 1,-1 19,2 IS A GLIDER GUN
BILL GOSPER C/O MIT PROJECT MAC
2,2 25,3 26,3 -8,0 -8,1 -9,1 15,4 15,0 2,2 2,-2 15,5 14,4 15,-1
14,0 3,3 3,-3 4,3 4,-3 17,3 17,2 17,-1 18,3 18,2 18,1 1,0 1,1
1,-1 19,2
(1122) RUSH REPEAT BACK

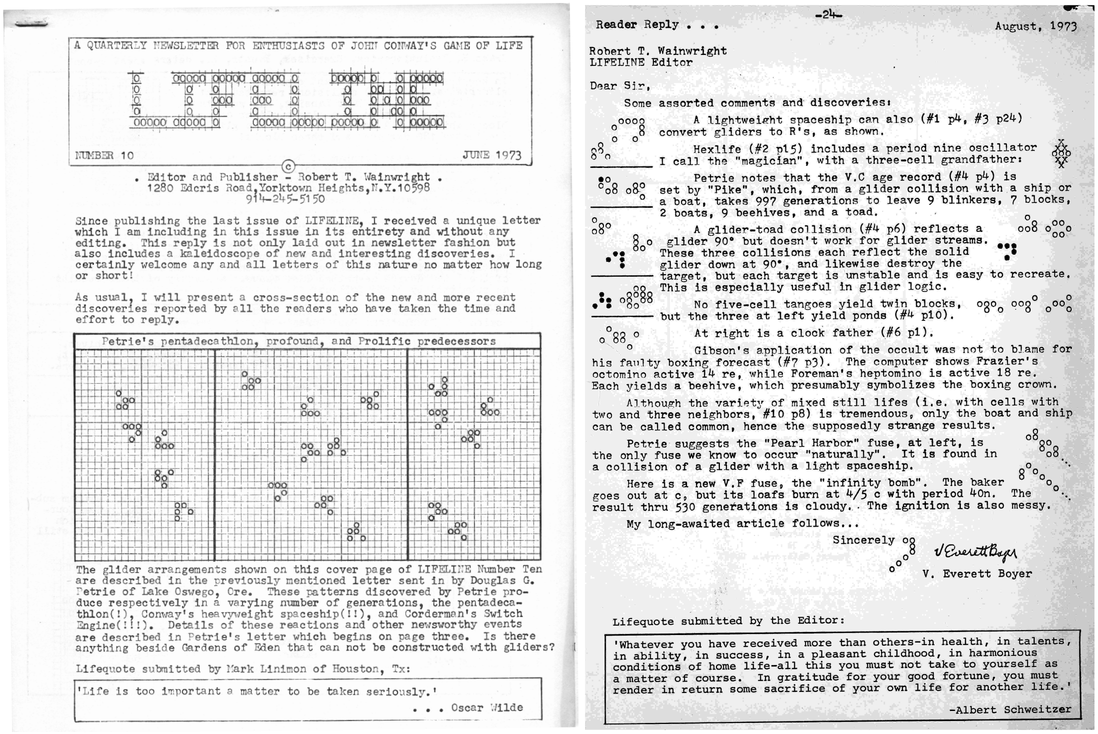
[I hope he's right!
Martin Gardner]

(x,y)

SF-1201 (R5-69)

Figure 4.27: Telegram from Gosper to Gardner containing the pattern for a glider gun, sent in response to Conway's challenge and prize. Wainwright, who had access to a computer, was asked by Gardner to verify the pattern, as it would have been very difficult to do so without access to one (Wainwright 2010). The October 1970 issue of Scientific American *issued the challenge. Gosper's telegram is dated November 15, 1970.*

Life demonstrates the evocative power of a simulation artifact. It inspired people to play not just the *Game of Life*, but to play with cellular automata more broadly. In doing so, it spread the underlying commitments of cellular automata: cellular decomposition, the themes of organic life, emergence, and computer simulation.



A QUARTERLY NEWSLETTER FOR ENTHUSIASTS OF JOHN CONWAY'S GAME OF LIFE

LIFELINE

NUMBER 10 JUNE 1973

©

. Editor and Publisher - Robert T. Wainwright .
1280 Edcris Road, Yorktown Heights, N.Y. 10598
914-245-5150

Since publishing the last issue of LIFELINE, I received a unique letter which I am including in this issue in its entirety and without any editing. This reply is not only laid out in newsletter fashion but also includes a kaleidoscope of new and interesting discoveries. I certainly welcome any and all letters of this nature no matter how long or short!

As usual, I will present a cross-section of the new and more recent discoveries reported by all the readers who have taken the time and effort to reply.

Petrie's pentadecathlon, profound, and Prolific predecessors

The glider arrangements shown on this cover page of LIFELINE Number Ten are described in the previously mentioned letter sent in by Douglas G. Petrie of Lake Oswego, Ore. These patterns discovered by Petrie produce respectively in a varying number of generations, the pentadecathlon(!), Conway's heavyweight spaceship(!!), and Corderman's Switch Engine(!!!). Details of these reactions and other newsworthy events are described in Petrie's letter which begins on page three. Is there anything beside Gardens of Eden that can not be constructed with gliders?

Lifequote submitted by Mark Linimon of Houston, Tx:

'Life is too important a matter to be taken seriously.'
. . . Oscar Wilde

-24-

Reader Reply . . . August, 1973

Robert T. Wainwright
LIFELINE Editor

Dear Sir,

Some assorted comments and discoveries:

A lightweight spaceship can also (#1 p4, #3 p24) convert gliders to R's, as shown.

Hexlife (#2 p15) includes a period nine oscillator I call the "magician", with a three-cell grandfather:

Petrie notes that the V.C age record (#4 p4) is set by "Pike", which, from a glider collision with a ship or a boat, takes 997 generations to leave 9 blinkers, 7 blocks, 2 boats, 9 beehives, and a toad.

A glider-toad collision (#4 p6) reflects a glider 90° but doesn't work for glider streams. These three collisions each reflect the solid glider down at 90°, and likewise destroy the target, but each target is unstable and is easy to recreate. This is especially useful in glider logic.

No five-cell tangoes yield twin blocks, but the three at left yield ponds (#4 p10).

At right is a clock father (#6 p1).

Gibson's application of the occult was not to blame for his faulty boxing forecast (#7 p3). The computer shows Frazier's octomino active 14 re, while Foreman's heptomino is active 18 re. Each yields a beehive, which presumably symbolizes the boxing crown.

Although the variety of mixed still lifes (i.e. with cells with two and three neighbors, #10 p8) is tremendous, only the boat and ship can be called common, hence the supposedly strange results.

Petrie suggests the "Pearl Harbor" fuse, at left, is the only fuse we know to occur "naturally". It is found in a collision of a glider with a light spaceship.

Here is a new V.F fuse, the "infinity bomb". The baker goes out at c, but its loafs burn at 4/5 c with period 40n. The result thru 530 generations is cloudy. The ignition is also messy.

My long-awaited article follows...

Sincerely

V. Everett Boyer

Lifequote submitted by the Editor:

'Whatever you have received more than others-in health, in talents, in ability, in success, in a pleasant childhood, in harmonious conditions of home life-all this you must not take to yourself as a matter of course. In gratitude for your good fortune, you must render in return some sacrifice of your own life for another life.'
-Albert Schweitzer

*Figure 4.28: Lifeline ran for 11 issues, until September 1973. Shown here, the first page of #10 and the last page of #11 (Source: http://*conwaylife.com*).*

The cellular automata simulation tradition is perpetuated by many infrastructures: software artifacts (e.g. *Life*[58], computer simulations), social practices (a contest, the community supported *Lifeline* newsletter), actors advancing particular agendas (e.g. recreational mathematicians such as Conway and Gardner; the biological aspirations of von Neumann and Langton), discursive practices (named patterns, Gardner's column), a craft of participation (the three craft traditions: representational, microcosm, and cosmogenic; a shift enabled by automatic computer simulation from *formal* to *empirical* engagement), and a schema (seeing and decomposing the world into a lattice of interacting computing elements; the cosmogenic view of the universe as a cellular automaton).

[58]*Life,* of course, exists without software and computers. But as an algorithm it can be, and often is, reified in software. It is a hybrid simulation algorithm/software artifact

The players of cellular automata romp in toy universes. The world in all its swirling complexity has been reduced to a dynamic and controllable miniature. The enactive representations can come to stand for anything, making them powerful props for make believe. Witness the imaginative names given to *Life* patterns, and the broad representational uses to which cellular automata are put. Their lively activity sparks enthusiasm and delight, recruiting many, including scientists, to turn their time and equipment to autotelic ends, pursuits of limited usefulness. This playful orientation is mirrored in the language employed. Ulam and his collaborators describe a *game* in which two cellular automata patterns fight for survival; Conway chose to call his automaton a *game*; Berlekamp et al. describe *Life* as a "no-player game".[59]

Poetics of Cellular Automata

What makes cellular automata so versatile and appealing to model makers? With what traits have cellular automata seduced multiple generations of practitioners? What gives them their representational grace, their ability to model phenomena as diverse as microbes, cities, sand

[59] Playfulness manifests in the naming of the diverse flora and fauna of patterns discovered in the *Game of Life* (e.g. Berlekamp et al. 2004; Wainwright 1972; Levy 1992). This pattern of imaginative naming manifests more broadly in the diverse representational practices of cellular automata.

Consider also the tone directed towards cellular automata. In their introduction to Conway's *Game of Life*, Berlekamp et al. include an epigraph by Oscar Wilde: "Life's too important a matter to be taken seriously." Conway and his collaborators used shells to mark cells that would "die," exclaiming "Shell the living!" (Levy 1992). This irreverent tone of engagement happened within the larger frame of Conway's recreational mathematics.

The *Game of Life* was published in Gardner's recreational mathematics column, in large part to offer a contest (a type of game) to a readership interested in mathematics for the pleasure of doing it. Ulam and his collaborators borrowed precious time from a machine intended for nuclear weapons research, subverting this serious machine to their un-respectable cellular automata experiments. Just as it was natural for Conway, steeped in the mathematical tradition of games and puzzles, to refer to his creation as a "game," Schrandt and Ulam refer to some of their explorations as an "amusement," a "game" (Schrandt and Ulam 1967; in Burks 1970, p.233). It took decades before cellular automata achieved a measure of scientific respectability, becoming more than a plaything (Levy 1992).

Is it an accident that these lively cellular simulations gave rise to playful language, and emerged from playful contexts? What does "game" have to do with "life"? In addition to the recreational nature of their activities, Schrandt, Ulam, and Conway were taken with the lively, vivacious, dynamic, and surprising activity that they discovered. This, in turn, led them to think of what they were witnessing as play: a "game."

dunes, and the spiral arms of galaxies? From where have they gathered these representational powers? What makes cellular automata so compelling?

A note I have struck throughout this chapter is the appeal of cellular automata. I turn, now, to examine the experience of cellular automata more closely, and reflect upon how their enactive representations gives rise to their aesthetic qualities. What we will see is that the appeal of cellular automata is due, in large part, to their underlying structure.

Observable

Cellular automata are highly observable. Because the internal model state is so tightly unified with its visible manifestation, it is easy to look inside and see what is happening. This is in stark contrast to the opacity of system dynamics. There, state trajectories are represented as graphs, and causal chains must be explained with prose—stories written by an expert interpreter and translator. In a cellular automaton, on the other hand, everything takes place on the surface—there are no hidden depths, layers of information, or action.

Causality visibly unfolds across space. The matrix of local couplings that connects adjacent cells gives rise to a vivid spatiality. You can see phenomena ripple, cascade, and propagate through space. Flows of causality and information undulate across the surface of a cellular automaton. And in Wolfram's 1d automata, time takes on a similarly vivid spatial clarity.[60]

[60]The *spatial* nature of cellular automata—an n-dimensional lattice of identical cells—engenders an easy to observe surface upon which impressionistic figures can be apprehended and described. Cells are woven into a space via local couplings; all interactions are spatially local. A simple set of rules, applied homogeneously throughout space, transform the state of each cell in response to local conditions. That the space is made of repeating identical units means that the space is *homogenous* (or isotropic), and the same transition rules are applied everywhere, blind to space (where) and state (what). This makes programming/implementing and understanding a cellular automaton simulator very straightforward. This is not a modeling paradigm in which distinct objects require special programming, accounting for all possible interactions, or continuously move through space and require complicated collision detection and resolution—like the original kinetic model Ulam's cellular

The cellular picture plane affords both macro- and micro- views. One can simultaneously take in an overview, and see the whole of a cellular surface, as well as attend to discrete individual cells or areas. Enumerated colors render the discrete states of a cell highly legible. On a computer with real time display, time takes on a similarly telescoping quality. The discrete microscopic steps of a simulation can be apprehended in addition to its lively and unpredictable macroscopic evolution across longer spans of time.

Discreteness enhances observability. Space, time, state, and transformation take on a schematic clarity. Differences and transformations are highlighted. These qualities, enhanced by the two dimensional picture plane, give cellular automata a highly diagrammatic quality.[61]

I refer to this collection of easy to apprehend qualities as *observability*. Observability, in combination with the inherent abstractness of cellular automata, yields an impressionistic and easy to describe medium, an enactive material with a high degree of representationally plasticity.[62]

Representational Plasticity

One of the most striking qualities of cellular automata is the diverse representations they have been pressed into, from snowflakes and microorganisms to cities and sand dunes. As in sys-

---

formalization abstracted out of existence. No objects, just a universal substrate. A function is simply applied to every cell in space. Homogeneity owes a debt to von Neumann's original idea of using a model with identical functional units. The spatial homogeneity of cellular automata are a natural fit for representations of the natural world. Another corollary to the spatial quality is the *locality* inherent to cellular automata dynamics; cells only interact with their neighbors. Global dynamics occur only as the result of a cascade of local interactions.

[61] *Discreteness* means that specific and definite places, states, times, and transformations are easily identified, comprehended, described, and articulated. Discreteness feeds into the schematic quality of cellular automata, giving rise to worlds which are easy to see, comprehend, and act upon. This discreteness, according to Burks, is characteristic of von Neumann's automata theory, in contrast to the continuous nature of Weiner's cybernetic approach (Burks 1970).

[62] This underlying substrate—spatial, discrete, and homogenous—was Ulam's original contribution to von Neumann's project; an introduction of elegant and expedient formalism into what had been a messy kinematic problem: robot parts floating atop a lake.

tem dynamics, the mapping between simulation and represented phenomena is very flexible, and is established through discursive practices that narrate, annotate, and describe the underlying abstract phenomena. But the lively patterns of cellular automata are also concrete and suggestive in ways that abstract graphs are not.

Cellular automata worlds are mimetic and impressionistic. In their spatial dynamism we see stylized figures move, die out, grow, dissolve, and bounce. Observable and suggestive phenomena—poised at a sweet spot between abstraction and concreteness—invite and fan the flames of description. Consider the evocative visuality of Schelling's neighborhood segregation model, whose diagrammatic clarity strongly suggests a racially divided neighborhood. Forrester's systems, by contrast, lean more heavily on figures and text—the simulation phenomena is less impressionistic. Cellular automata are inherently more evocative, and better suited to representing lively spatial systems.

Stephen Levy compares the practice of labeling *Game of Life* phenomena to the naming of constellations (Levy 1992). The comparison is apt, as cellular automata, like constellations, yield vivid, impressionistic, and evocative figures—graceful igniters and receivers of description.

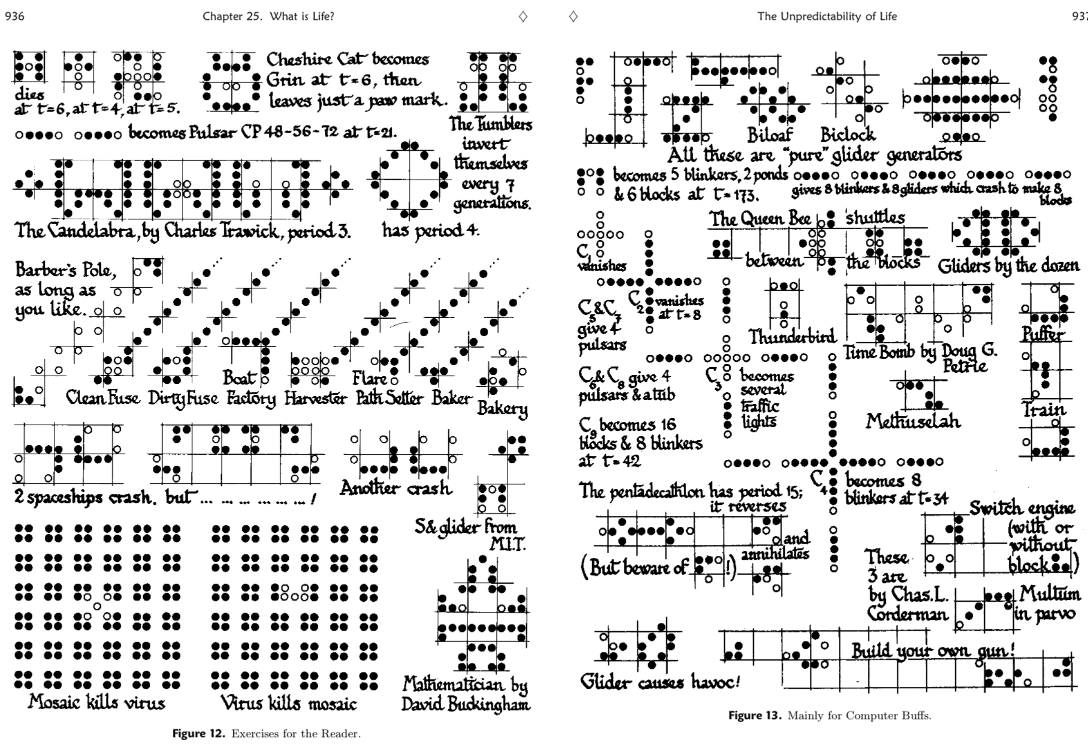


Figure 4.29: A couple pages from Berlekamp, et al. which demonstrate the diversity of representational labels attached to the suggestive figures and dynamics of Game of Life. The richness of the names underline the evocativeness of the patterns and dynamics (Berlekamp et al. 2004).

The intrinsic discreteness and abstractness of cellular automata—discrete states, spaces, rules, and transformations—offer many things to label. Continuous system dynamic graphs, by comparison, don't offer neatly segmented things to describe.[63] Cellular automata are a rich substrate around which to weave a "hermeneutic network" (Zhu 2009) of diagrams, descriptions, and narrative. Visual and textual glosses ground and connect abstract and evocative forms to both the familiar and fantastic. The abstract figures of cellular automata can come to mean anything, but it is description that imbues them with particular meanings. Our pretense is scaffolded, bringing us into the shared make belief of a miniature universe.

The critical role of these discursive practices, and the fluid boundary between phenomena,

[63]In a system dynamics model description, on the other hand, the tables are turned. Forrester's diagrams of levels and rates offer many discrete things to label, and have an easy to apprehend schematic quality.

perception, and description, is apparent when one considers *Game of Life* patterns. Stable patterns emerge out of and dissolve back into an entropic background, calling attention to the perceptual contingency of these forms. It is through conjoined description and perception that we see a stable pattern of blinking cells as a distinct figure, for example a glider moving through space.[64] Illustration and annotation emphasizes the integral nature of a particular set of cells. Multiple cells become a chunk: a brain, nerve cells, muscle cells, or a tail.

Emergence

The *Game of Life* and cellular automata more broadly are prototypical examples of emergence. *Life's* rules are quite simple, yet give rise to a mesmerizing assortment of dynamic phenomena. The aesthetic magic of emergence lies in the linkage of opposites: the yoking of simplicity and graspability to complexity and surprise. Cellular automata are ideal instruments for highlighting this articulation between the simple and the complex. This, after all, was von Neumann's motivating interest: how a complex organic process such as reproduction emerges from the organization and interaction of many simple elements. Emergence is hard wired into the foundational abstractions of cellular automata.

While system dynamics models also exhibit emergence, for example when they exhibit oscillatory steady states, cellular automata offer a more profound vision of emergence. In mathematical terms, the infinite cellular lattice is combinatorially more explosive than the scalars that make up a system dynamics model. We can see inside higher order stable figures, like

[64]The glider exemplifies a higher order gloss: the glider is a narrative about a pattern moving through space. A collection of cells, of course, must be identified as a glider, and its movement is an emergent quality we perceive and narrate out of a spatiotemporal coherency. There is no glider object in the system, only a blinking pattern of cells which retain a perceptible coherence across time and space. Cellular automata model lively activity, affording the representation of dynamic process. More broadly, the glider demonstrates how a stable process—even if it weren't a pattern translating through space—can itself become the subject of narration.

gliders, and see the simple individual units that make them up. Such figures are readily apparent as contingent granular phenomena. Cellular automata are like a new kind of telescope, an instrument that brings the universe and its granular elements into focus.

Cellular automata offer a vision of the world in which distinct objects with stable identities are an illusion; objects, rather, are contingent sections of a continuous substrate. In *Life*, as in the ancient board game Go, entities fluidly break apart, rejoin, are born, die. Figures can be in states of partial construction and destruction; they fluidly emerge and recede from the background, merge, transform, and dissolve. There are no individuals, just a single continuous universe substrate. Low level building blocks comprise higher order elements of interest—as well as unanticipated mutations and variations. The universe is modeled at a level of granularity one step below the objects of primary interest.[65]

The cellular lattice gives rise to a combinatorial explosion of possibilities, making possible the linking of simple rules to surprising outcomes. That cellular automata afford simultaneous macro- and micro- views encourages seeing and thinking across scales, from micro-scale interactions to their emergent macroscopic effects.

Cellular automata bring the atomic particles of the world and the edifices they form into simultaneous focus. One can grasp the emergence of *Life* through manual simulation, by

[65]This is one of the most powerful characteristics of the cellular automata formalism, and its genesis is directly traceable to von Neumann's original project. In order to design a self-replicating automaton, von Neumann needed the automaton to be able to incrementally build a copy of itself. A parent automaton would construct a new automaton out of the nothingness of the universe. The stuff out of which the child automaton would be built would be the same stuff out of which the parent automaton was made. For this reason, von Neumann postulated a world made of identical functional units. The same universal stuff constituted parent, child, and void. Rearrange these building blocks, and the automata "creatures" could also mutate, another trait von Neumann was after. It would have made no sense for von Neumann to choose as the level of granularity of the world a complete Turing machine: mutations, incremental construction, and the structure of his proof would have been impossible. The project forced von Neumann to model the universe, not the inhabitants of that universe. Building blocks were designed which could then be patterned into a self-replicating figure. That forms can emerge out of and dissolve back into the background transform, divide, and recombine is a direct result of von Neumann's need for a representation in which complex things could be born; a medium in which reproduction could take place.

performing the simple rules that produce surprising outcomes. Simple rules entice would be players into manual simulation, reward them with delightful surprises, yet frustrate them with tedium; inducing them, in turn, to program autonomous computer simulations. And simple rules, after all, are easy to program. Computerized simulations emblazon the complexity embedded within a system's simple seeds. A telescopic view is afforded, bringing the simple and the complex together into a comprehensive vision. Cellular automata offer a glimpse of both the subatomic particles and the complex living world they give rise to.[66]

Agency

Cellular automata offer a hybrid model of agency, a collaboration between the automaticity of the world and the interventions of the player. Pleasure lies in not just watching them, but poking and tinkering with rules and configurations.

Life is described as a "no-player game," a formulation that calls attention to the question of agency in cellular automata (Berlekamp et al. 2004). If these are zero player games, then how does one play? What is a player to do? What agency do these worlds afford? Despite being characterized as a "no-player game," *Life* isn't really a game. It lacks turn taking, competition, a win condition, and so forth. But people do *play* with cellular automata worlds, experimenting with rule sets, initial conditions, and sometimes intervening and interacting with an evolving

[66]As we have seen, computers catalyzed the creation, playing, and exploration of these cellular worlds (Codd 1968; Burks 1970; Gardner 1971; Levy 1992; Wainwright 2010; Wright 2011). Owing to von Neumann and Ulam's use of an automaton formulation, they run easily on computers. The *Game of Life* isn't just easy to play by hand, but is also relatively easy to program—and is encouraged by the fact that running it by hand is rather tedious.

The computer performs a critical role in not just making emergence apparent, but bringing these systems to life in another sense—easing the burden of simulating them, performing experiments, and perhaps most critically, enabling their players to run them in real time, and to quite literally *see* them as dynamic vivacious worlds. Gosling, for example, discovers the Glider Gun with a computer. Codd and Wainwright extol the benefits of computer simulation. The simple rules of the world, codified as programs, easily yield their complex outcomes through computer simulation.

cellular automata world. In system dynamics and cellular automata, players design the models, establish initial conditions, and then see what happens. Then, they tinker with the model or initial conditions, and run it again. It is as if they push a cart to the top of the hill, let it go, and watch it tumble down. Toffoli and Margolus, in this chapter's epigraph, cast cellular automata creators as *composers,* authors of universes which are then *played,* underlining a powerful and rewarding sense of agency in world building. (And iterated rebuilding.) Players design and wind up the clockwork world, step back, and watch it unfold.

But sometimes the makers of these clockwork worlds intervene, tinkering with its rules, or state, while it is still unfolding (e.g. Codd 1968; Burks 1970). One of Wright's key appropriative transformations was to produce a system dynamics and cellular automata model in which players were expected to intervene in the unfolding drama. Rather than have the tax rate automatically set by a simulated political process, players in *SimCity* set the taxes—whenever they want. Players zone, demolish, and build while the automaton is running. Taken together, player agency and automatic simulation constitute a hybrid agency, a two part improvisation between human and machine.[67]

Agency is heightened by observability and emergent dynamics. Causal effects swiftly ripple across a discrete spatial landscape, giving agency an immediacy and apparentness lacking in system dynamics. Cellular automata are extremely sensitive to rules and changes in state. Each cell is a butterfly vigorously flapping its wings, often causing storms elsewhere—a quality that is more true in a discrete cellular automata than in a continuous model like a fluid simulation. Even small interventions in rules, initial conditions, or the cellular landscape can

[67] Computation magnifies this sense of agency. Codd, as discussed earlier, observed that the "[r]apid interaction between man and machine" heightened intuition, perceptivity, and excitement (Codd 1968). Gardner's follow up column on the *Game of Life* noted its popularity with computer scientists—in other words, those with access to computers (Gardner 1971). The vision of highly reactive cellular automata drove the creation of the Cellular Automata Machine (Toffoli and Margolus 1987a).

have enormous and highly apparent results. Sensitivity plus observability imbues the players of cellular automata—whether they tinker with the rules, the world, or both—with a strong sense of agency.

Chapter 5

SimBusiness

"And are not we as well, if you examine us physically, mechanistically, statistically, and meticulously, nothing but the miniscule capering of electron clouds? Positive and negative charges arranged in space? And is our existence not the result of subatomic collisions and the interplay of particles, though we ourselves perceive those molecular cartwheels as fear, longing, or meditation? And when you daydream, what transpires within your brain but the binary algebra of connecting and disconnecting circuits, the continual meandering of electrons?"

"What, Klapaucius, would you equate our existence with that of an imitation kingdom locked up in some glass box?!" cried Trurl. "No, really, that's going too far! My purpose was simply to fashion a simulator of statehood, a model cybernetically perfect, nothing more!"

… As they circled it, preparing to land, they beheld a most amazing sight.

The entire planet was covered with countless signs of intelligent life. Microscopic bridges, like tiny lines, spanned every rill and rivulet, while the puddles, reflecting the stars, were full of microscopic boats like floating chips…. The night side of the sphere was dotted with glimmering cities, and on the day side one could make out flourishing metropolises, though the inhabitants themselves were much too little to observe, even through the strongest lens. Of the king there was not a trace, as if the earth had swallowed him up.

—Stanisław Lem, from "The Seventh Sally or How Trurl's Own Perfection Led to No Good" (Lem 1974).[1]

[1]This story is one of Wright's many sources of inspiration for *SimCity* (Lew 1989; Wright 2016).

Introduction

This chapter and the next are a case study of *SimCity* as a play design artifact. This chapter is about *SimCity's* social history, how and why it was made, and which biographical and historical influences shaped it. My ethnographically spirited account builds upon existing journalistic and historical sources, extending these with extensive original interviews I conducted with many of the key actors involved in *SimCity's* creation. The next chapter is a close reading of *SimCity* itself, and deconstructs its software representations, examining how they scaffold play.

SimCity has its origins in Will Wright's play. A highly synthetic and creative player, Wright scaffolded his own computer play with a variety of tools. Appropriating the enactive representations of system dynamics and cellular automata, various traditions of game making, and inexpensive home computers, he built fascinating playthings first for himself, and eventually for others. Wright's creative, synthetic, and playful vision is evident in the broad range of materials he appropriated into his computer play. Despite Forrester's dismissal of games and play, and his earnest belief in system dynamics as a serious practice, like engineering, for solving real world problems (Forrester 1961; Forrester 2016), Wright was able to turn it into a plaything for his own amusement and learning.

SimCity eventually become a software toy for more than just Wright. This translation was predicated upon a writer, additional programmers, an artist, package illustrator, packaging designer, manufacturing, a distribution network, marketers, and business expertise. The entrepreneur Jeff Braun engineered this social assemblage, setting the stage not just for *SimCity*, but for additional Will Wright software toys. But the company they founded, SimBusiness (more commonly known as Maxis), exhibited an inescapable tension between the inward creative play of Wright that generated software toys, and the business of selling those software toys—the outwardly facing constraints of player-consumers and capitalism. The social assemblage Braun had engineered, SimBusiness, eventually fell apart, resulting in its sale to Electronic Arts. This sale, however, made possible Wright's next commercially successful software toy: *The Sims.*

Because it was such an accessible software plaything, *SimCity,* and Maxis by extension, became an interchange for traffic between popular culture and the rarified world of computer simulation and complexity studies. Popular publications and big companies came knocking, as did education innovators and researchers of complexity and artificial life (Braun 2015; Braun 2016a). As an evocative prototype, *SimCity* provoked thought and debate about educational technology, complexity theory, simulation, and games, transforming these ideas and questions into a tangible and playable form.

SimCity looms large not just in popular culture, but in academia, too. This thesis is the latest work in what is now a long tradition of *SimCity* studies. *SimCity* figures prominently into discussions of politics, games, definitions of games, education, modeling, politics, emergence, semiotics, urban design, and cognitive mapping (e.g. Angus 1989; Tanner 1993; Kelly 1994a; Kolson 1994; Resnick 1994; Starr 1994; Bleecker 1995; Friedman 1995; Kolson 1996; Resnick 1996; Adams 1998; Friedman 1999; Bos 2001; Johnson 2001; Squire 2002; Bittanti 2004; Bleecker 2004; Juul 2005; Seabrook 2006; Gaber 2007; Kay 2007; Lauwaert 2007; Ito 2009; Wardrip-Fruin 2009; Sample 2011; Bittanti 2014; Koebler 2015; Wark 2015). These discourses seize upon *SimCity* as a tangible and playable model of a complex real world system.

A seminal work by the influential software toy designer Will Wright, *SimCity* is an ideal object for studying play design. Analyzing *SimCity* from a play design framework in which it sits at the comfortable center, rather than the awkward margins of games, demonstrates the utility of such a framework. In *SimCity* we can see the crystallization of Wright's approach to software toys, the play qualities inherent to the traditions Wright borrowed from, and strategies for approaching computers as a resource for play. While Wright's follow-on works were met with mixed commercial and critical reception, comparing them to *SimCity* reveals important play design lessons. As we will see in this chapter and those that follow, the design of *SimCity* harbors many lessons in the design of software toys, and design for play more broadly.

Will Wright at Play

Will Wright

Will Wright (born 1960) grew up building and playing with models: "ships, cars, planes—I loved to do that." At 10, he made a balsa-wood model of the Star Trek Enterprise flight deck. Both his father and grandfather were engineering graduates of the Georgia Institute of Technology. As a teenager he played complex military board games like *PanzerBlitz* (Seabrook 2006).

Wright thrived in the self-directed, discovery based, and problem solving atmosphere of the Montessori school:

> Montessori taught me the joy of discovery. It showed you can become interested in pretty complex theories, like Pythagorean theory, say, by playing with blocks. It's all about learning on your terms, rather than a teacher explaining stuff to you. SimCity comes right out of Montessori—if you give people this model for building cities, they will abstract from it principles of urban design (Seabrook 2006).

Wright's penchant for creative self-directed play and learning harks back to his Montessori upbringing, and is echoed not just in his own unusual trajectory, but in the kinds of experiences he would design for others—a reflection of his practice of designing for himself.[2]

[2]Wright attended Montessori school until about the 6th grade. It wasn't until later in his career, well after *Sim-*

His appetite for self-directed learning through a wide range of subjects is reflected in his formal education. Although he enrolled in both Louisiana State University and the New School, to study robotics, and excelled at subjects he was interested in—from architecture and economics to mechanical engineering and military history—he never finished a college degree (Seabrook 2006). He studied programming, but found punch cards and printouts to be cumbersome and tedious; looking back, he sees it lacked the responsiveness and interactivity that would eventually hold his attention (Wright 2016).

His aptitude for model making, dynamic systems, and creatively synthesizing diverse materials and ideas is also reflected in his robotics pursuits. He repeatedly made robotic models of one of the most complex biomechanical organs: the human hand. At 13, he built a hydraulic robotic hand out of repurposed syringes. Later, living in New York City, he built a 100 pound robotic arm out of repurposed machine parts, landing gear hydraulics from a military aircraft. Not long afterwords, he built a dynamic scaffold for the injured hand of Joell Jones, whom he would eventually marry, out of "metal and rubber bands" (Seabrook 2006; Wright 2016).

A Medium for Microworlds

Wright's lifelong interest in robotics led him to computers. To control his robot arm, he bought an Apple II, learned to program it in Pascal, and devised a radio modem that used walkie-talkies

City, that Montessori's educational philosophy became more than a subconscious influence on his work. Wright connects his design approach to the constructivist tradition of Friedrich Fröbel, Maria Montessori, Seymour Papert, and Alan Kay. "Let's give toys that are creative to kids, and let them play and learn geometry and math" (Donovan 2011).

> I think I just realized I enjoyed making things, and I thought other people would enjoy making things, as well, on the computer, and having them come to life. But then after that, I started reading more about the pedagogy of her [Montessori's] approach to education and stuff, so it became a little more self-conscious to me as a designer. But I think I always gravitated towards that process anyway (Donovan 2011).

for computer-robot communication.[3] He learned about artificial intelligence, which would be necessary for controlling robots. From there, his insatiable curiosity brought him to artificial life and Conway's *Game of Life* (Wright 2016).

Fascinated by the amazing patterns that emerged out of "utter simplicity," Wright describes himself as losing almost a year of his life to Conway's *Game of Life*. It could, Wright says, "drive you crazy." It was through *Life* that Wright became a more proficient programmer. "I started writing a little simple version of *Life* in BASIC, and then I wanted it to go faster." Driven by his love for Conway's cellular automaton, Wright learned low level assembly language programming, bringing his ability to create computer software representations to new depths of sophistication (Wright 2011; Donovan 2011).

The Apple II magnified his interest in the games and simulated worlds he had enjoyed his entire life. He became "infatuated" with computer conversions of the complex board games he already knew, such as Avalon Hill's *Ambush!* These computer games offered a unique twist: computer opponents that relieved the burden of finding or teaching people to play these complex games.[4] Such games were packaged and sold in simple Ziploc bags, a sign of the embryonic computer game industry (Rouse 2001; Wright 2011; Donovan 2011).

[3] "Actually, what got me into games was robots, because I was building robots as a teenager. Kind of these weird mechanical things out of random parts, and it kind of went from models, to robots, to games. I bought my first Apple II computer to connect to my robots to control them as I was building, and that's the point at which I really taught myself to program" (Donovan 2011). One of his rally racing friends had a business buying Apple II's in bulk, converting them to medical computers, and reselling them. The friend talked Wright into getting one, selling it to him at a discounted price (Wright 2011; Wright 2016).

[4] Having "enjoyed games my whole life," he "was intrigued by playing games on the computer — having the computer as an opponent" (Rouse 2001; Donovan 2011). Given their complexity "[i]t was really hard to find anyone else who could learn the rules, because there was so much investment in that." But they were ideally suited to the computer, as it "was always ready to play with you. You didn't have to go find somebody and spend hours teaching them how to play the game" (Donovan 2011).

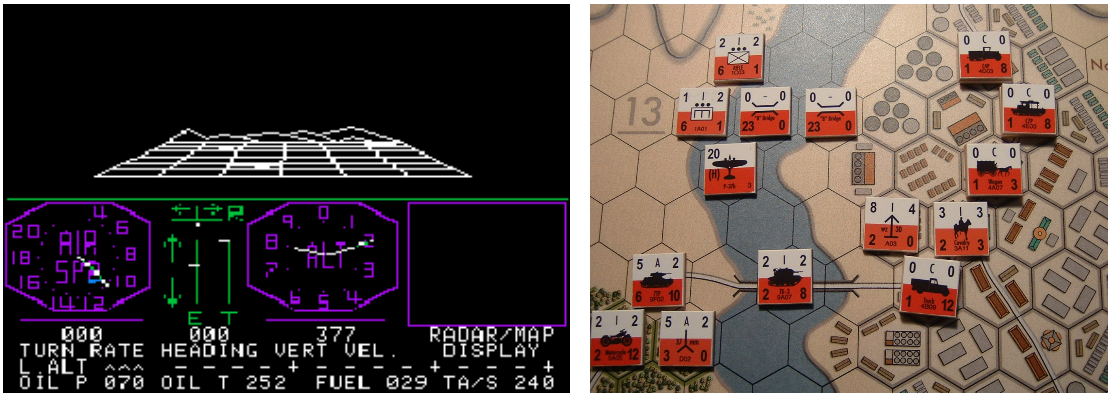


Figure 5.1: At left, Artwick's Flight Simulator *for the Apple II (1979) (Image from Wikipedia). At right, detail from a* PanzerBlitz *game (Dunnigan 1970). Photo by Ward McBurnery, from boardgamegeek.com. Caption accompanying photograph on board game geek reads: "Polish units make a river crossing at a major city while engineers construct a second bridge; minimal air cover."*

He also found himself in a new world of software toys and digital microworlds. He played Bill Budge's *Pinball Construction Set* and Bruce Artwick's *Flight Simulator*[5] (Wright 2011). "I was fascinated by the ability to have this little microworld inside the computer," he says. "And that world had its own little rules and physics that you could interact with" (Kushner 2005). Despite *Flight Simulator's* low resolution wireframe graphics, Wright was amazed by

> …this little self-consistent world that you could go fly around in and interact with, in sort of limited ways. … It was very open-ended and I could do whatever I wanted to in it. The first thing I did was I went in and started exploring the behavior space. Trying all the different things with the airplane. What happens if I go straight up? How far can I go? What happens if I crash? What happens if I do this that and the other? So I could carry out experiments in this world. And in running those experiments I could get a more accurate view of what the internal model was. So it's kind of a scientific process. It's kind of a "hypothesize, experiment, change your hypothesis" type cycle that was going on (Pearce 2001).

These titles turned Wright onto the possibilities of microworlds as mediums of representation. "But just the idea that you could build your own little micro-world inside the computer intrigued me. So I saw it as a kind of modeling tool" (Rouse 2001). Someone had designed

[5]It would later be purchased by Microsoft and become *Microsoft Flight Simulator* (Wikipedia, History of Microsoft Flight Simulator, 2016).

those worlds, carefully crafting their rules in assembly or Pascal, replicated the software on floppy disks, packaged these into plastic bags, and sold them as a product. And he, after all, had already been programming in assembly, as close as you could get to the computer's circuitry, and knew a thing or two about model making and complex game rules. Why not make and market a microworld of his own?

Building a Microworld

Wright decided to make a game. At the very least his computer expenses would be "tax deductible" (Wright 2011). He began working on a game around the time he moved to California to be with Joell Jones, his girlfriend and future wife. The game he made, *Raid on Bungling Bay*, established his identity as an entertainment software developer, a maker of computer microworlds. His approach reflected his interests in the computer as a medium for intricate model making, as well as his strategic capacity for creatively synthesizing diverse interests, influences, and constraints.

Consider his choice of platform. Figuring that he was "behind the pack" on the Apple II, for which virtuosically programmed games already existed, made by programmers who had mastered the craft of coding for Steve Wozniak's brilliant and challenging computer design (e.g. Bill Budge and Nasir Gebelli), Wright settled upon and bought the recently released Commodore 64 (C64) computer (Wright 2011), "figuring that it would be better to start on a new machine where everybody was on a level playing field, because other people had learned the Apple II years before I decided to do this. So I bought a Commodore as soon as it came out and just dove into it, and learned it as quickly as I could" (Rouse 2001).

The platform choice had creative implications. Wright paid attention to things the Commodore 64 could do which the Apple II could not. The VIC-II graphics chip allowed programmers to define custom character sets, and produce "smooth scrolling worlds." The SID audio chip enabled, for the first time on such an inexpensive personal computer, complex sounds to be generated (Wright 2011). Wright would take advantage of both of these opportunities.

The theme emerged from Wright's lifelong interest in helicopters and war games. A self-directed learner, project motivation followed interest in a particular topic, a pattern evident in his future projects and past educational experiences. In his youth, he once rode a helicopter at the state fair, and had watched "Mutual of Omaha's Wild Kingdom"—a television program that always ended with a helicopter chase.[6] This inspired him to make a game in which you played a helicopter pilot (Wright 2011). His interest in war games, not to mention the conventional video game tropes of piloting a shooting vehicle, account for the combat scenario. Of course, the Vietnam War unfolded on television during Wright's youth and that was full of both helicopters and combat.

Raid takes place in a clockwork world, reflecting Wright's interest in and aptitude for intricate model making. Despite the fact that *Raid* superficially resembles traditional shooters—and despite Wright's characterization of *Raid* as "a really, and I mean really, stupid video game" (Kelly 1994a)—its divergence from genre conventions reflects Wright's interest in computer microworlds, foreshadowing both *SimCity* and his emerging design sensibility. Wright credits Conway's *Game of Life*, cellular automata, and "things like orreries" which simulate "the world in very mechanistic fashion, very precise little things that have a lot of kind of unpredictable complexity emerging from them," for his desire to construct a "clocklike world" (Wright 2011). Since childhood, of course, Wright had been involved in model making and playing finely de-

[6]Note that *SimCity, SimCity 2000*, and *SimCopter,* all of which Wright worked on, contain or concern themselves with helicopters.

tailed strategy simulation games (Seabrook 2006), but it was computer games and software toys that ignited his interest in the computer as a medium for microworlds that one played with and learned through by creating, exploring, and tinkering.

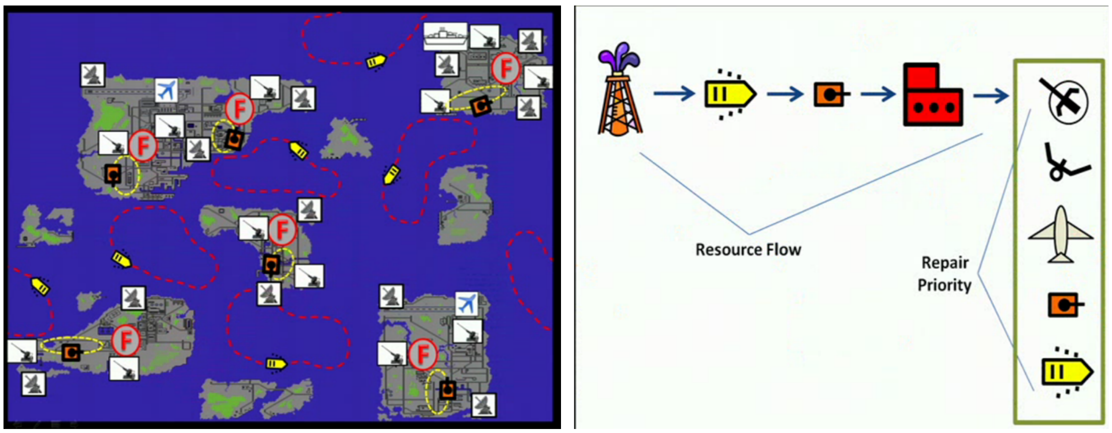


Figure 5.2: Slides from Wright's Raid on Bungeling Bay *postmortem talk. At left, the game map annotated with enemy military installations. Note boat and tank paths. At right, a schematic of the Bungeling Empire's resource flow and the priorities of how those resources are invested in repairs (Wright 2011).*

A clockwork model of military-industrial infrastructure animates the enemy: boats circulate on predefined routes, ferrying harvested resources from the water onto land, where tanks transport those resources to factories; factories then consume resources to repair the anti-aircraft guns, radars, tanks, and boats that detect and defend against the player's attack. Looking back, Wright observed that this simulation detail was somewhat wasted on the players, as it was barely legible. He notes that introducing a visible iconography for the resources, such as barrels, would have been a huge improvement, rendering the resource flow readily apparent (Wright 2011). In *Raid,* Wright synthesized influences from two quite different war game traditions: the slow and intricate turn based military strategy game, and the fast paced arcade shooter. The arcade war game tradition is evident in the player's goals and means: to pilot a helicopter and destroy the factories which sustain the Bungeling empire's industrial war machine.

He wanted a sprawling and immersive microworld: "a game where the world was large enough to get lost in." The Commodore 64's novel capacity for smooth scrolling enabled Wright to distinguish *Raid* from the "simple little worlds" that characterized contemporary computer games (Wright 2011). The hardware inspired and informed the game design: a large smoothly scrolling world composed of a quilt of redefined character graphics.[7]

As is customary in game development, he made special tools. He built a character editor program, *Chedit*, for designing the repeating graphical elements out of which *Raid's* world would be built, and a world editor, *Wedit*, for composing the individual graphical characters into a world. Unknown to him at the time, the world editing tool would eventually evolve into *SimCity*. He made a third tool that accelerated his mastery of the C64: a custom hardware interface which linked his older Apple II to the new machine. This allowed him to develop software on the Apple II and download it into the C64, speeding up development (Mechner 2011; Wright 2011; Budge 2013).

Unlike most computer game designers, Wright strove for naturalistic representation. Wright eschewed the conventional trope of video game levels to ratchet difficulty, as he desired "one coherent world." Instead, the Bungeling empire's defensive response escalated as its six factories were destroyed, creating a sense of ratcheting difficulty. The game climaxes when the Bungeling empire builds and deploys a battleship that seeks out and destroys the player's

---

[7]Wright says he was able to

> …redefine a character set, make it look like graphics, and then smoothly scroll it around the screen. So you could give the impression that you were scrolling over this huge bitmap, when in fact all you were doing is moving ASCII characters around on the screen. And when I saw that feature, I thought that would be really cool looking, because I knew the Apple couldn't begin to move that much in the way of graphics around the screen that smoothly. So I designed the game around that feature in a way (Rouse 2001).

home base, an aircraft carrier. Throughout the game, the player must balance attacking the enemy with defending her own carrier. Wright describes this continuously escalating difficulty as a spring that pushes back harder and harder as you press into it (Wright 2011). As opposed to a typical arcade style video game, *Raid* is a more systemic and naturalistic representation.

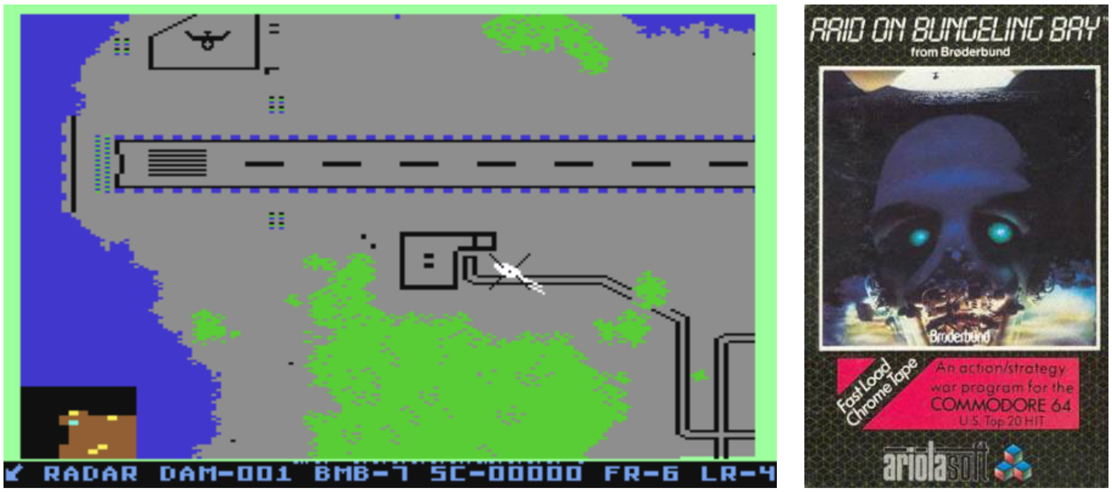


Figure 5.3: Left, screenshot of Raid on Bungeling Bay *(1984). Right, its box design (Images from Wikipedia).*

Published by Brøderbund in 1983, *Raid on Bungeling Bay* foreshadowed and laid the groundwork for *SimCity*.[8] Financially, royalties from the game enabled Wright to take a few years off and cultivate what became *SimCity*, as well as pitch in for Maxis's startup costs after his future business partner, Jeff Braun, ran out of money (Seabrook 2006; Wright 2011; Braun 2015; Braun 2016a; Wright 2016).[9] Creatively, *Raid* contained within it the seeds of Wright's

[8]Wright shopped around his game, driving around to the small number of game publishers in the San Francisco Bay Area. Eventually, he partnered with Brøderbund (Donovan 2011). Wright is very grateful for the mentorship and guidance offered by Brøderbund, whose pacifist values moved them to narrate the enemies of *Raid* as the fictional Bungeling robot empire, an antagonist also shared by the titles *Choplifter!* and *Lode Runner*. This fictional move meant the player would not be harming humans. He was surprised by Brøderbund's box design, describing it as "the ugliest thing I had ever seen" (Wright 2011). The working title until Brøderbund's renaming was *Carrier Patrol* (Foley 2015).

[9]Despite selling only around 30,000 copies in the U.S., Wright reports that just about everybody with a C64 had played it due to piracy. The game, however, was one of the first American titles converted to Nintendo's Japanese Famicon system, where it sold around 800,000 units (Rouse 2001; Wright 2011; Donovan 2011). (Sales figures

future software toys. Wright had made a sprawling and naturalistic clockwork world, as he would again in *SimCity*—even if he hadn't quite figured out yet how to surface its behavior to the player.[10] While *Raid* the game was the product people played, it was Wedit, the world editor tool Wright had made for himself, that captured his attention. While players were bombing the landscape of *Raid,* Wright found that he couldn't stop building that landscape, and elaborating his world editing tool. And it was this tool that evolved into *SimCity.*[11]

City Planner

Wright lived off the *Raid on Bungeling Bay* royalties for a few years, taking a year off after Jones gave birth to their daughter Cassidy, and it was during that time that he began elaborating *Wedit*, *Raid's* world editing tool, tinkering with it, and adding features (Donovan 2011).

The world editing tool gained new qualities of augmentation and automaticity, qualities Wright found satisfying, and which motivated him, in turn, to continue elaborating it (Kelly

vary depending on the source. Figures quoted tend to lie between 800,000 and a million units.) Nintendo's cartridge based systems made piracy much more difficult, and Brøderbund had a lucrative royalty arrangement with Nintendo. "Almost all the money I made on *Raid on Bungeling Bay* came from Japan," says Wright (Wright 2011). *Raid* was ported to the Famicom and NES by Hudson Soft, and was also distributed as a Nintendo branded arcade cabinet in Japan. Wright had one of these cabinets in his home (Foley 2015).

Wright's family also supported his ability to follow his interests. A trust fund established by his late father had funded his meandering college education, and his family eventually sold his father's plastics company after Wright declined to take it over (Seabrook 2006; Braun 2016a; Wright 2016).

[10] The clockwork world is composed of an interrelated network of actors. Resources, for example, flowed across the map, from boats and tanks to factories and defenses, and the player's role is to disrupt this system at critical points. An intricate system provides both *Raid's* naturalistic difficulty ramp—a systemic pushback Wright compares to a spring whose stiffness ratchets—and means for the player to intervene: the player must intervene in the system, targeting the factories that lie at the center of the enemy's defenses, and balancing defense of their own carrier with offensive tactics.

Naturalism is evident in Wright's eschewal of game tropes such as difficulty levels, choosing instead to create a single coherent world whose difficulty smoothly ratcheted upward—making the game trope of increasing difficulty diegetic. Part of *SimCity's* appeal is in the smooth blending of naturalism with the characteristic irreality of games (e.g. its madcap disasters).

[11] "I found that I was having so much more fun with the paint program than I was with the game that after I finished the game I kept playing with the paint program" (Rouse 2001).

1994a; Rouse 2001; Wright 2011).[12] Laying down a road had been like mating dominoes. To make a road you had to choose exactly the right characters for each tile, so the graphics of the entire road matched up and appeared continuous. Wright automated that, so all you had to do was define the road's path, and *Wedit* would automatically pick out the right character art.[13] This was satisfying, as players of *SimCity* can attest. He added more automaticity. He wanted traffic on the roads, but to do this the microworld needed places for the people to go. He wanted automatic building placement, which led him to make a zoning tool. But automatically picking the appropriate building for a zone required a model of urban growth and decay. He accomplished this with a land value model that eventually became the heart of a spatial city simulation. At some point, he named the program *City Planner* (Kelly 1994a; Braun 2016a; Wright 2016). All of this, in a sense, was an elaboration on the "clocklike world" Wright had begun in *Raid*.

Reflecting on why he was still playing with this tool, Wright realized he "was fascinated by bringing a city to life" (Kelly 1994a). His interests in the computer as a representational medium, living clockwork worlds, and appetite for self-directed learning culminated in the research and development of a simulation that led, according to Wright, "layer upon layer to a whole city" (Kelly 1994a).[14] Wright wanted to "see the world kind of come alive and be more dynamic,"

---

[12] "…I started adding more and more features to it. Making it a little bit more automatic and more satisfying" (Wright 2011). Also: (Kelly 1994a; Rouse 2001).

[13] According to Wright,

> … I kept going back to the darn thing and making the building utilities more and more fancy. I wanted to automate the road function. I made it so that when you added each connecting piece of island, the road parts on them would connect up automatically to form a continuous road. Then I wanted to put down buildings automatically, so I built a little menu choice for buildings (Kelly 1994a).

[14] Wright says that

> I started asking myself, why am I doing this since the game is finished? The answer was that I found that I had a lot more fun building the islands than I had destroying them. Pretty soon I realized that I was fascinated by bringing a city to life. At first I just wanted to do a traffic simulation. But then I realized

and what had begun as a toy for himself become a hybrid city planning tool and living world.[15] The microworld fed his autodidactic impulses, motivating him to learn "about urban planning theories, urban dynamics and simulation." He plowed his learnings back into the model, which "became a kind of test-bed." Wright would "read urban planning theories, then go code them up in the simulator. What was a very dry subject in books became very fascinating when I had this guinea-pig city that I could do these experiments on" (Phipps 2005). Research and model making became Wright's play, a pattern that would repeat in the future; he delighted in learning about cities and crafting a miniature world.

Luckily, his friend across the street was a city planner. Just as Wright consulted with James Lovelock, the originator of the Gaia hypothesis, in the development of *SimEarth*, and *SimAnt* was inspired and informed by the work of seminal myrmecologist Edward O. Wilson (Murray 1992; Holthaus and Allenby 2015), Bruce Joffe catalyzed early development of *SimCity*. Joffe worked for Bechtel, where he was part of a team that designed mining towns around the world.[16] His interest in urban planning was originally sparked while working for the Johnson Administration's Office of Economic Opportunity, which sought to increase the availability of low income housing.[17] After OEO's funding dried up in the 1970's, Joffe took a scholarship from MIT, where he earned masters degrees in both architecture and city planning.[18] While there, he also took a class from Jay Forrester on system dynamics, and a class on assem-

> that traffic didn't make a lot of sense unless you had places where the people drove to…and that led layer upon layer to a whole city (Kelly 1994a).

[15] "First, it was just a toy for me. I was just making my editor more and more elaborate. I thought it would be cool to have the world come to life" (Donovan 2011).

[16] He also developed an early GIS style technology to map the infrastructure of Riyad, Saudi Arabia so it could be better managed.

[17] This agency was started in response to the urban crisis, and was one aspect of Johnson's Great Society agenda.

[18] 1974-1977

bly language programming. Although he never built computer simulations, he modeled urban growth policies with the aid of geographical mapping software he wrote himself. Now, thanks to Wright's newfound interest in cities and maps, they had a new shared enthusiasm; in addition to playing Go and talking about Douglas Hofstadter's *Gödel, Escher, Bach,* programming, and the Apple II, they chatted about maps and cities. Joffe recalls being part of a conversation with Wright in which the idea of a city planning game emerged. Although the specifics of their interactions around *City Planner* are hazy, its development clearly benefited from Wright's interaction with Joffe, who received a special thanks in the manual (Wright 1989; Braun 2015; Foley 2015; Joffe 2015; Wright 2016; Joffe 2016).[19]

Wright's voracious, discipline crossing, and imaginative learning and appropriations fed him the raw materials with which to make *City Planner*. His search for modeling techniques led him to the work of Jay Forrester, who Wright describes as "one of the very first people to use a computer for simulation," and "the first person to try to simulate a city." Wright found system dynamics to be a versatile tool, describing it as "a way to look at a system and divide it into, basically, stocks and flows. Stocks are quantities, like population, and flows are rates, like the death rate, the birth rate, immigration. You can model almost anything just using those

[19]Mick Foley recalls that Joffe, after seeing Wright's map editor, pushed him in a new direction, telling him that "You could do something interesting with that" (Foley 2015). Jeff Braun told me, after our first interview, that "without Bruce Joffe there would be no *SimCity"* (personal recollection, May 19, 2015). John Seabrook's *New Yorker* profile of Wright claims that "[a] neighbor suggested that Wright take a look at a 1969 book called Urban Dynamics…" (Seabrook 2006). The neighbor could only be Joffe, except that both Wright and Joffe reject this particular story. Even if not literally true, the story nonetheless signifies a creative exchange between Joffe and Wright, and the mischaracterization captures the impracticality of clear credit assignment in such interactions. Joffe recalls that Wright has generously credited him for his contributions.

Given their friendship, intellectual camaraderie, and respective backgrounds, it's hard to believe that Joffe, who is given special thanks in the *SimCity* credits, didn't play an important catalyzing role in Wright's conceptualization and development of *SimCity.* Wright recalls showing Joffe an early version after the simulation was up and running, and that Joffe grew very excited and started bringing him books he didn't know of, for example the work of people like Jane Jacobs and Paulo Soleri. Joffe characterizes Wright as chasing his own intellectual interests. Wright didn't remember that the program was ever entitled *City Planner* until I mentioned that Braun had told this to me, after which Wright confirmed it was true (Joffe 2015; Wright 2016). This, of course, underlines the fact that these events are three decades in the past, and recollections are spotty and don't always concord.

two features" (Rouse 2001).

Wright compares system dynamics to a spreadsheet; it had no map. To remedy this shortcoming, he returned to the cellular automata he had played with earlier, which enabled him to model "propagation, network flow, proximity, and so forth." For Wright, like other practitioners of cellular automata, the appeal of systems such as "[the ancient game of] Go and Conway's *Life* or cellular automata in general" lay in how "there's some underlying aspect" of "reality and complex systems" that "they capture." The combination of system dynamics and cellular automata gave rise to exciting new dynamics. In addition to this fruitful hybridization, Wright layered cellular automata so they could "interact on the third dimension. So the layers of crime and pollution can impact the land value layer." Like a sculptor who creates by aggregating heterogenous materials, Wright created a complex simulation assemblage by combining many different techniques and subsystems into a new complex enactive representation (Rouse 2001; Donovan 2011).

He took inspiration from Stanisław Lem's lyric science fiction story *The Seventh Sally.* In this short story, quoted in the chapter's epigraph, a exiled king is given a "microminiaturized society" in a portable box, "very like a child's toy," crafted by the master inventor Trurl, a "constructor-benefactor." Peering through a "thick glass lid," the king, unseen by his subjects, intercedes and controls their lives by "manipulating the control knobs, which were carved with imperial eagles and regal lions." (This description echoes that of *SimCity's* box design.) Wright was both the constructor-benefactor and the king; his play was in inventing the box as well as in manipulating the world it projected. Beyond the metaphor of the box, Wright took two key ideas from the story: agency and empathy. The player controlled a world in a box. Very much like Lem's king, and unlike most computer games, the player was in an "undefined role … a mayor or a king … almost like a god." The inhabitants were "unaware of your presence," but you, the player, were "connecting at some level, empathizing with these little simulated data points." Lem's story also conveyed the magical feeling of "fiddling with it while it was running" (Lem 1974; Lew 1989; Wright 2016).

Tinkering with a clockwork world in motion was a new take on simulation agency. In the simulation traditions Wright drew from, modelers would design a clockwork world, wind it up, and then watch it go. It was like programming with punch cards, which Wright disliked; hand over the stack of cards, and only later would you would receive the results. This lacked the satisfying responsiveness of a computer game like *Raid on Bungling Bay.* Conway's *Life* was better; at least you could watch a vivacious animated display. But *Life* didn't offer the player a role in the ongoing action. Even Bill Budge's seminal software toy construction set *Pinball Construction Set,* one of Wright's inspirations, modally separated editing from playing. You designed the table, and when you were ready, played pinball on it. To modify it, you had to stop playing. Early versions of *City Planner* were the same. You'd pause, edit the map, and then push play. Eventually Wright merged these modes, a significant technical feat, which enabled playing and editing to unfold simultaneously (Wright 2016).[20]

Wright "wasn't trying to build something that people would play for entertainment value. It's more like I was just having fun doing this on my own" (Rouse 2001). His marvelously intricate toy city was the product of autotelic and exploratory play, expertise in intricate model making, self-directed and wide ranging research, and an ability to synthesize diverse influences and materials. Wright transformed a game development tool into a pastime all its own.[21]

---

[20] As simulation complexity increased, it became harder for the Commodore 64's 1Mhz computer to keep up. This required implementing a time sharing feature that would have been readily available in more sophisticated operating systems and higher end computers, but not on personal computers at the time. "I actually had to write a little multitasking operating system to get it to run on the Commodore because there were so many routines that were slow to run like power scan" (Wright 2011).

[21] In this process we see the crystallization of many of Wright's prior interests: the large, naturalistic, scrolling clockwork world of *Raid*—made possible by the C64 graphics architecture; Conway's *Game of Life* and cellular

Wright never thought *City Planner* could be popular: “I thought it might appeal to a few architects and city planner types, but not average people” (Donovan 2011).[22] He would later self-deprecatingly describe *City Planner* as “awful”—it wasn’t a game, and people didn’t like it—to Jeff Braun, the entrepreneur he would eventually partner with (Braun 2015; Braun 2016b). But by building an animate microworld, Wright had made the arcane theory of “urban dynamics and city behavior” accessible. His “toy simulated version … brought the whole subject to life.” He found building and playing his toy to be more engaging than just reading a book, because he could construct a model and then experiment with it. His own enjoyment caused him to start wondering whether others might enjoy it, too (Donovan 2011).

City Planner raised some big questions: could a city design tool and clockwork world be a finished experience that others might want to play with and pay for, as they did with computer games? How could the interface for such a complex system be made approachable, not just for Wright, but for others? Could it ever be as accessible as a toy? More than a miniature world, it was also a kind of construction set, an open ended plaything. Rather than making a serious city planning tool, he would make a software toy. But to understand software toys and construction sets we need to turn to a landmark 1983 release Wright took inspiration from: Bill Budge’s seminal *Pinball Construction Set*.

---

automata; his childhood interest in model building and systems (e.g. robots, war games); computer games, and in particular their ability to bring miniature worlds to life for exploratory play (e.g. Artwick’s *Flight Simulator*, Montessori). His appetite for self-directed learning, play, fascination with computer microworlds, synthetic creativity, and aptitude for model making all gave rise to *City Planner.*

[22] Indeed, both Joffe’s recollections and the working title suggest that Wright may have thought, at one point or another, that he was making a tool for city planners (Joffe 2015; Joffe 2016).

Software Toys and Construction Sets

Wright's play had been scaffolded by the materials he appropriated from system dynamics, cellular automata, and theories of urban development and city planning. He borrowed the enactive simulation representations that others had invented, and turned them into a new kind of simulation, a hybridized and layered assemblage of enactive representations. Through his creative play he transformed himself into both the player-king and the designer-inventor of a miniaturized toy city. But seen through the lens of a video game maker and player, it was the experience of *another player* that mattered most. And so he would share with other players the experience of city planning, ruling over the world, and tinkering with it while it ran, but the pleasure of researching and programming the microworld would be reserved for him. In this design turn, he was inspired by the emerging genre of the software toy construction set, which was defined by Bill Budge's landmark 1982 release, *Pinball Construction Set*.

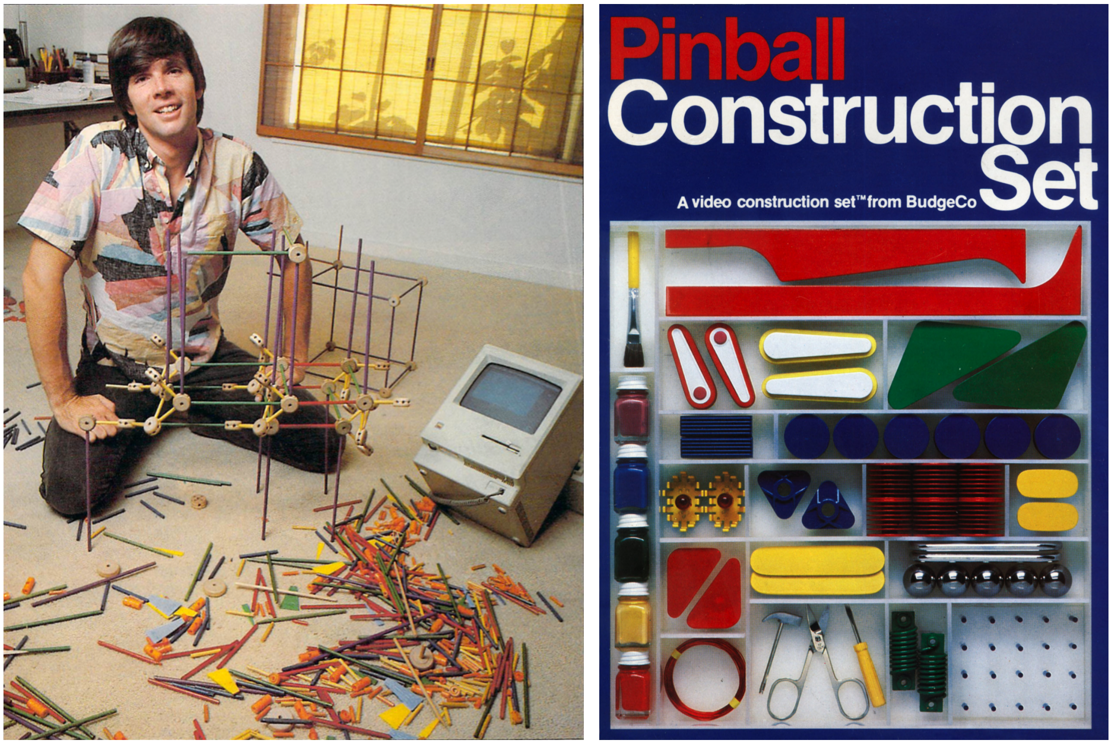


Figure 5.4: At left, Budge with tinker toys and a Macintosh (Meyers 1985). Photo credit: Ed Kashi. At right, the original BudgeCo box design for Pinball Construction Set*. Budge notes that the box design and color scheme mimic the stylings of the construction toys of his youth, reinforcing that* Pinball Construction Set *is "meant to be a toy" (Budge 2013). Budge recalls the box was designed by the art director for* The Berkeley Monthly*, and featured some parts of a wrecked pinball machine he had salvaged, but a model builder fabricated the rest of it (Budge 2016).*

Budge's creation gave Wright the license to make a software toy, and it showed him how to do it. *Pinball Construction Set* introduced the phrase *software toy*, and alongside it, the software toy genre of the *construction set*. Budge did this by employing the landmark graphical user interface design conventions that had crystallized at Xerox PARC in the 1970's. It also marked the first time a popular audience came into contact with the interface conventions that would incite and come to define popular computing. *Pinball Construction Set* demonstrated how an arcane computer assisted design tool could become an accessible construction set: with a graphical user interface (GUI). GUI conventions afforded direct manipulation of a microworld, scaffolded participation, and brought to users the same degree of tangibility that players of graphical computer games enjoyed (chapter 2). A properly designed construction set would scaffold construction based play.[23]

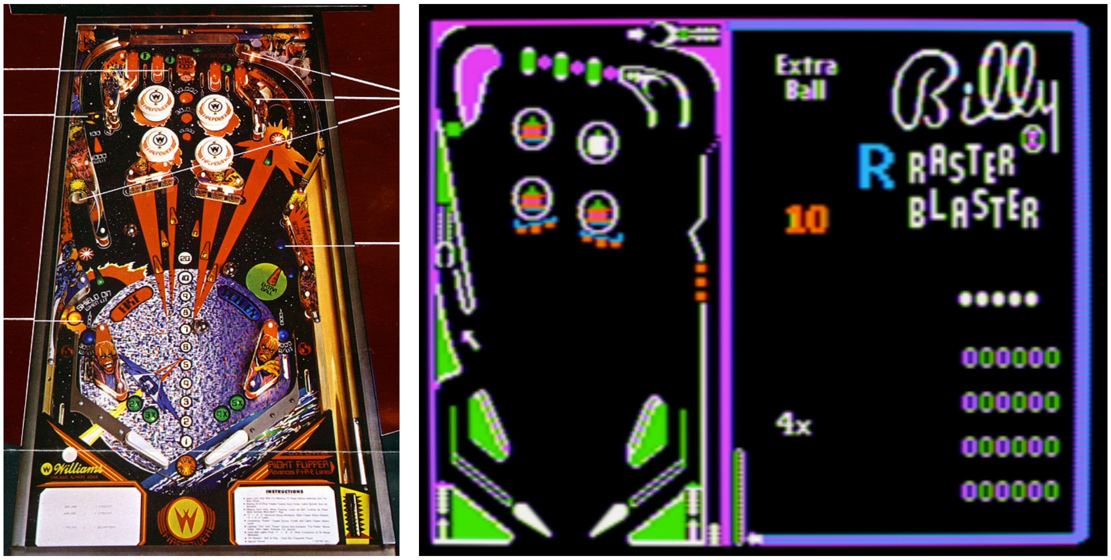


Figure 5.5: Left: Firepower *(1980) pinball table whose layout Budge copied with tracing paper. Image is a detail from a promotional flyer, source: ipdb.org. Right: a screenshot of Budge's Apple II pinball game* Raster Blaster *(1981). Source: Wikipedia. Note similar table layouts.*

Pinball Construction Set was the latest in a long tradition of bouncing ball computer demos and games that extend as far back as Whirlwind's *Bouncing Ball* demo, except that Budge

[23]Wright was influenced and inspired by (Pearce 2001; Pitts 2012) another, massively influential, 1982/1983 release he admired: Bill Budge's *Pinball Construction Set* for the Apple II (1982, BudgeCo; 1983, Electronic Arts). Budge's virtuosic game programming brought him a reputation as one of the Apple II wizards, an Apple II coder whose depth of knowledge and expertise Wright wisely decided to not compete with by taking up the relatively new Commodore 64 computer. The reputation of Budge's *PCS* is such that Steve Wozniak—the brilliant engineer who designed the Apple I and II and co-founded Apple Computer Inc.—is quoted, perhaps apocryphally, as calling it "the greatest program ever written for an 8-bit machine" (e.g. Maher 2013).

(It's possible this is simply a product of EA's excellent marketing. The quote appears, attributed to Wozniak—who sat on EA's board of directors, after all (Levy 1984)—in an EA ad for *Pinball Construction Set* entitled "Bill Budge wants to write a program so human that turning if off would be an act of murder," as well as the back of EA's box design for *PCS*—without the attribution to Wozniak.)

Not only was *PCS* a technical tour de force, but it was Budge who introduced, to my knowledge, the phrase "software toy" in order to describe his (game inspired) non-game, a phrase that Maxis (Jeff Braun and Will Wright's eventual company) would use to describe their products. And it was *PCS* that established the "construction set" genre which *SimCity* can be seen as apart of. *PCS* also disseminated the graphical user interface (GUI) design concepts which had crystallized at Xerox PARC—interaction conventions which very much lie at the heart of *PCS* and *SimCity*.

had introduced a twist, and made a construction set.[24] Budge, like Wright, had become enchanted with his game making tools, the pleasure of shaping those tools, and the possibilities afforded.[25] Like Wright, Budge's interests in computers and computer games took flight as a result of the Apple II, which was particularly well suited towards game playing and game making on account of its design. Steve Wozniak, the machine's brilliant designer, knew that supporting games would be key to a successful personal computer, and that BASIC, an easy

[24] Players wouldn't just watch a bouncing ball (in the case of the Whirlwind demo), or use a ball to play a game (as in Pong, Breakout, and its progeny), or just play pinball (as in Budge's 1981 *Raster Blaster* for the Apple II, or the 1978 pinball game for the Xerox Alto). Players would design, and then play with, their own pinball machine layouts (Maher 2013).

[25] See (Maher 2013) for an excellent historical account of Budge's programming career.

to learn programming language popular with game hobbyists, was the key to games.[26] In the case of Budge, Wright, and many others, this design strategy paid off magnificently.

Pinball Construction Set, like *City Planner,* grew out of earlier game projects. Budge's preliminary successes with creating games on the Apple II, mostly clones of preexisting games,

[26]Growing up, Bill Budge (born 1954) enjoyed construction toys like Fischer Technik, Lego, Lincoln Logs, Tinkertoys—which he credits as the inspiration for *Pinball Construction Set* (Budge 2013). He learned to program in a "Computer Math" high school course, which enabled him to program versions of games such as tic-tac-toe, checkers, and Go (Maher 2013). After graduating with a computer science degree from U.C. Berkeley he continued, in 1979, into its Ph.D. program. Budge decided to spend his money on an Apple II computer after seeing one belonging to a classmate in his supercomputer class (Budge 2013). This classmate was Andy Hertzfeld, a future key contributor to the Apple Macintosh ("Bill Budge Pioneer Award Panel" 2011; Budge 2013). The Apple II, introduced in 1977 for $1,298, was much more expensive than other personal computers on the market, and was in short supply—it took three months for his order to arrive from Village Electronics in San Francisco. Its disk drive and graphic capabilities—features which would be key to the success of both the Apple II and Budge's software—convinced Budge to borrow the money he needed from his parents in order to purchase it (Budge 2013; Maher 2013; Wikipedia, Apple II, 2015). The design of the Apple II reflected Steve Wozniak's mastery of both hardware and software, which enabled him to design the computer in such a way as to save money on hardware components by compensating with software. This meant that the Apple II was able to reach a broader market, both in terms of cost and capabilities, but it meant that programming it was challenging (Budge 2013).

The Apple II personal computer Budge now owned was particularly well suited towards games. Wozniak's design for the Apple II hardware and BASIC implementation reflected a desire for the machine to support games broadly, and *Breakout* in particular. (Wozniak, famously, had created the original Breakout for Atari, a deal which Steve Jobs engineered and took an unfair share of.) Believing that his computer would "have to play games" in order to be successful, and "the key to games was BASIC," Wozniak engineered the Apple computers to support sound, color graphics, and come equipped with BASIC (Wozniak 2014). This idea paid off, luring enthusiasts in with games, and then encouraging them to delve deeper into the machine's arcana. This is clearly evident in the case of both Budge and Wright, who followed these inviting stepping stones, first as enthusiasts playing games, then as BASIC programmers, and then eventually as assembly coders. Freeing game logic from the rigidity of a hardware representation, as was the case with arcade games at the time, and moving it into software was transformative:

> I called Steve Jobs over to my apartment to see what I'd done. I demonstrated to him how easily and instantly you could change things like the color of the bricks. Most importantly, in one-half hour I had tried more variations of this game than I could have done in hardware over 10 years. Steve and I both realized how important it was going to be now that animated (arcade style) games could be software. More than that, being in BASIC meant that anyone of any age could program it (Wozniak 2014).

Games written in BASIC were more fluid, easily made, and amenable to tinkering (e.g. Ahl 1973; Albrecht 1972; Albrecht 1975; Wozniak 1986; Wright 2011; Maher 2013; Albrecht 2014). One of these tinkerers was Budge, who cut his teeth on Apple II programming by modifying Wozniak's BASIC implementation of *Breakout* printed in the Apple II technical manual (Budge 2013; Maher 2013). Budge became so good at the game that he modified it, making up new rules and increasing its difficulty. Budge then reprogrammed the game in assembly, giving it high resolution graphics. This ultimately led him down the path of commercializing derivatives of popular games like *Pong* and *Asteroids*, first trading a collection of games to Apple Computer for a $700 printer, and then partnering with Al Remer's California Pacific Computer company, which packaged and distributed his games in plastic bags to stores (Kohler 2011; Budge 2013).

brought him notoriety amongst Apple II enthusiasts, eventually landing him a job at Apple. There, his learning accelerated, transforming him into an expert Apple II programmer. Budge also absorbed two powerful enthusiasms then swirling about the company: pinball fever and the graphical user interface.[27] Soon afterwards, he left Apple and released a pinball game program, *Raster Blaster,* whose table design was also a clone, this time of Williams's *Firepower* pinball machine. It was his most masterful programming effort to date.[28]

---

[27]Budge's reputation was such that Apple recruited him to work as a graphics engineer on the doomed Apple III computer, a job he began in 1981 (Hague 1997; Maher 2013). At Apple, he absorbed not just the cultural enthusiasm for pinball games, but also the Xerox PARC inspired graphical user interfaces that Apple would release in 1983 (Lisa) and 1984 (Mac) (Budge 2013; Maher 2013). Budge found his work on the Apple III boring, but eventually switched over to another ill fated computer development project, the Lisa. The Lisa team was Apple's first effort in building a computer in the image of Xerox PARC's GUI efforts. (The Macintosh was the second and successful effort). Budge describes his exposure to the interface paradigm that would revolutionize computing as

> Kind of amazing. I had heard about the Xerox user interface but it didn't really make a big impression on me. I didn't realize how great it was. Not everyone did. But, see, Jobs and the rest of Apple were pretty blown away by it. … I saw a lot of stuff with a lot of modern user interface. Absolutely had a huge impact on me (Budge 2013).

In addition to absorbing the conventions of the GUI, Budge came under the sway of the pinball fever many of his fellow engineers at Apple were in the grip of:

> I was working at Apple in 1981, and there was a pinball craze going on among the engineers there. Everyone was interested in playing pinball. So I decided to do a pinball game in hi-res. I saw an opportunity to do a game that hadn't been done and at the same time would be interesting to program, since there were graphics and physics problems that would have to be solved (Hague 1997).

Every day at lunch the engineers would play pinball. Excited to have a game developer on their team, his peers encouraged him to program a pinball game for the Apple II. Hertzfeld recalls Budge using tracing paper to copy the layout of the "*Firepower*" pinball machine (Budge 2013).

[28]Budge credits the leap in technical mastery evident in *Raster Blaster* to how his programming ability at Apple "went into hyperdrive as far as my learning, as opposed to working on my own" (Budge 2013). Mainly, Budge "wanted to see if I could build it as a technical exercise … Making it fun to play was kind of secondary" (Kohler 2011). Rather than market his new game through California Pacific, Budge quit his job and, along with his sister, formed BudgeCo to market *Raster Blaster* (Maher 2013).

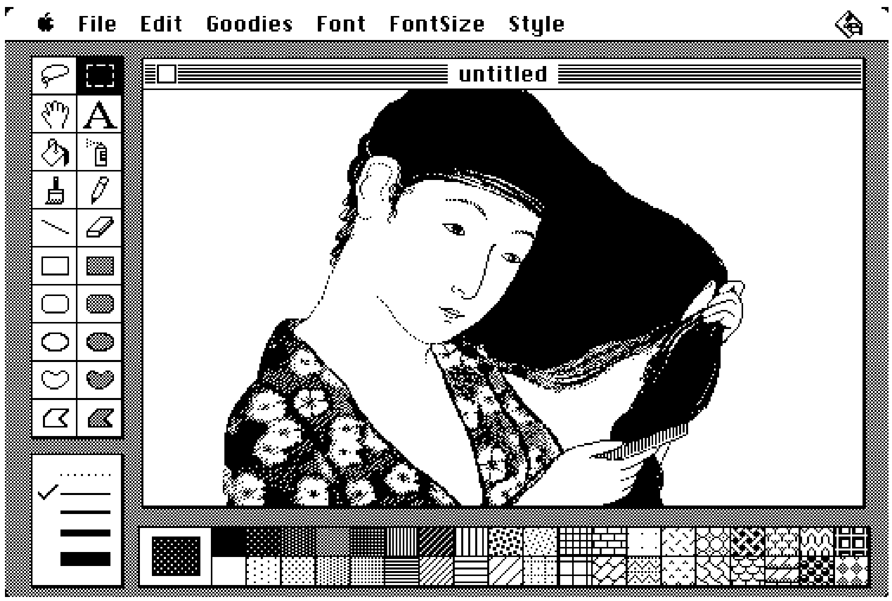


Figure 5.6: Bill Atkinson's MacPaint *(1984). Released with the Apple Macintosh. Note tool palette at left. Illustration by Susan Kare, based on a Japanese woodcut (Kare 2001). Source: Wikipedia. The overall interface design builds upon GUI conventions cultivated at Xerox PARC. These GUI conventions—developed with children as a primary audience—were invented and refined in order to produce software that was both easy to use and facilitated creative activity (Hiltzik 1999; Waldrop 2001).*

Budge was an expert coder who specialized in cloning games. Game programming was more interesting to him than game design, which led him to try something different for his next project. Using the GUI conventions he had seen at Apple, he would produce a tool for making and playing simulated pinball machines. He realized "that creating tools for others to make games was a way for me to indulge my interest in programming without having to make games" (Hague 1997; Maher 2013). The project grew out of the development tool he used to create *Raster Blaster,* in particular the background paint tool.[29]

---

[29]*Pinball Construction Set* is a kind of pinball game meta-clone; a tool for authoring and playing software pinball machines. According to Budge, he "wasn't that interested in playing or designing games," but "writing fast graphics code." For his new project, Budge didn't want to write another video game. He wanted to try making something different: a toy (Budge 2013).

To aid the development of Raster Blaster, Budge had developed a tool to help him paint the background, which contained colorful Apple logos and a handsome script "Billy" logo after that of "Bally," a well known pinball game manufacturer. The magnifier tool, which resembled the Xerox PARC inspired GUI he had seen at Apple, allowed him to zoom into a part of the background and easily edit individual pixels. The magnifier tool was the seed that

Figure 5.7: BudgeCo advertisements for Pinball Construction Set. *Note reference to the GUI that would one day become common: "…has a user-interface which until recently has been available only in expensive systems like Apple's LISA®," the idea of a "software toy," and, of course, a "construction set." Eventually the game would be published by Electronic Arts, whose marketing and packaging materials gave a different inflection to the ideas expressed here.*

Pinball Construction Set, which he and his sister self-published in 1982, transformed pinball table design, his advert claimed, from hard work into fun. The box depicts a pinball machine pieces, neatly organized, suggesting model kits and the fantasy of designing and playing one's own pinball machine. The pleasure of game creation was now more widely accessible;

Pinball Construction Set grew out of (Budge 2013). This transformation, from a developer's tool into a player's construction set and toy, resembles *SimCity's* future outgrowth from *Wedit*, the development tool used to edit Raid's landscape. GUI design conventions allow a technical design craft to be transformed into an accessible and tangible practice.

And so, armed with what he had seen of the GUI and his existing pinball game, and search of new technical challenges, Budge undertook to make a pinball construction set:

> I was exposed to GUI's at Apple, and I had the pinball simulation from "Raster Blaster." I saw that it would be a small step to do a construction set. This was the kind of program I liked, since there was no game to write. But it was a lot of work, since I had to implement file saving, a mini sound editor and a mini paint program (Hague 1997).

you didn't have to learn how to program or fabricate a pinball machine. Synthesizing the traditions of game design and development with the interface conventions of direct manipulation authoring tools, Budge had invented "The First Software Toy." Design became play. It was the prototypical construction set software toy, inspiring and influencing *SimCity* as well as many other titles.[30]

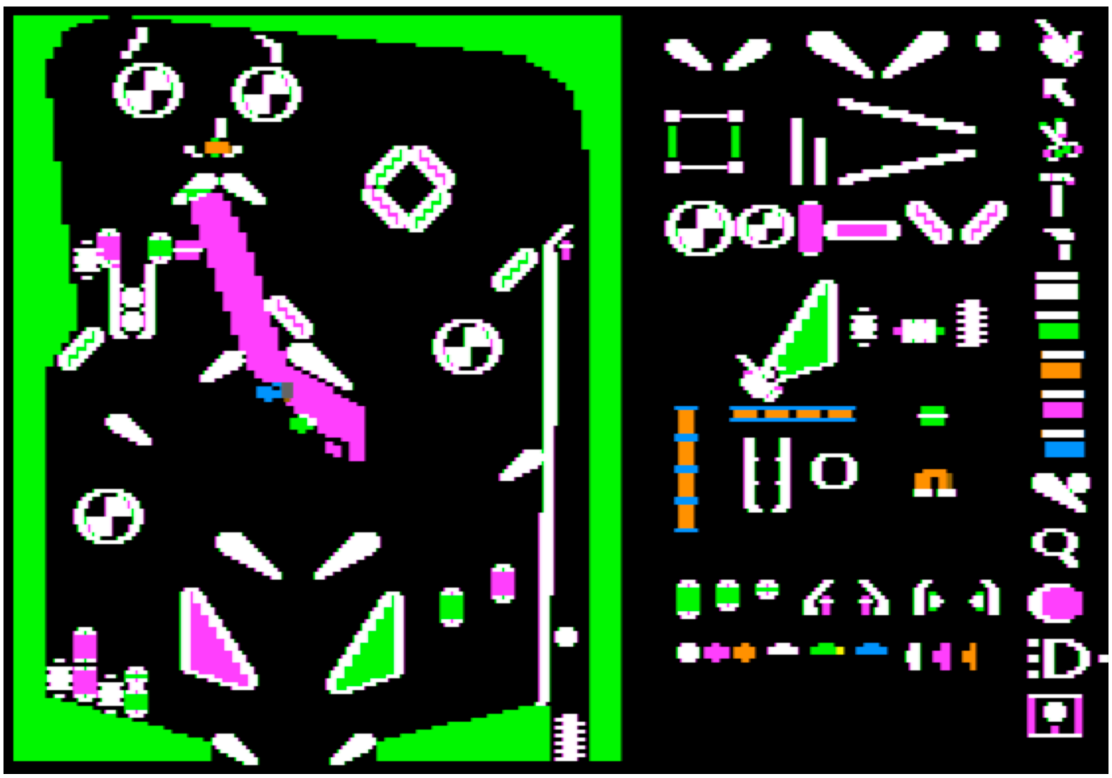


Figure 5.8: Pinball Construction Set *screenshot taken with an Apple II emulator. Parts and tools occupy the right side of the screen, while the pinball table sits on the left.*

[30]*PCS* was originally released by Budge in 1982 for the Apple II under the self-published BudgeCo imprint, and was later picked up and published by Electronic Arts for the Apple II, Atari 800, and C64 (all 1983) where it achieved strong commercial success. Maxis (Jeff Braun and Will Wright's eventual shared endeavor) advertised their products as "software toys," which directly followed in the footsteps of *PCS*, which was advertised in its self-published form as "The First Software Toy … the simplicity and freedom of interaction of a toy. You don't use this program—you play with it."

EA and others followed it with titles such as Music Construction Set (1984), Racing Destruction Set (1985), Adventure Construction Set (1985), Girlfriend Construction Set (Scheffler 1989), Tale Construction Set (1991) (Barton and Loguidice 2009). SimCity also fits into this mold, and might as well have been called "City Construction Set." *Pinball Construction Set* was a tool that inspired, guided, and amplified one's creative impulses in a particular domain, bringing the creations to life.

Pinball Construction Set is a wizardly achievement of design and technology. Players can place parts via a drag and drop interface, paint the background, shape the table's contours with a polygon tool, and wire points and sounds to triggers. It even supported multiple balls, and you could save tables to disk so your friends could play them.[31]

Many play design lessons are evident in *Pinball Construction Set*. Giving players this much flexibility meant they would surely push the boundaries of what was possible, or at least anticipated by the developer. Budge recognized that people would make "crazy stuff," and decided to just let them do it. The program gracefully and robustly accommodates and condones boundary pushing behavior.[32] Although the GUI eased the act of creation, table design still required patience and expertise. To satisfy impatient players and scaffold new designers, sample tables are offered, and players are always greeted by a suggestive and partially complete table. These templates guide and inspire play.[33]

[31]A GUI style hand cursor allows a panoply of parts to be dragged from a palette into the table layout. A bitmap editor allows the painting of rich background imagery. In addition to paint tools and drag and object editing, *PCS* features a surprisingly sophisticated vector shape editor, enabling table layouts to be comprised of freeform polygons. A logic wiring mode enables different points and sounds to be linked to various triggers on the table. On top of all this, the pinball machine designs, which can be quite complex, are fluidly simulated, even with multiple balls in play. Finished designs can be exported to a floppy disk as fully playable games, so that creations can be shared and played by others who don't own *PCS*.

[32]Putting all of this expressive power in the hands of players can lead to complications. While one can never anticipate exactly how players will play, and what they will make, it is certain that they will push the boundaries of such a tool, test its limits, and in general wreak havoc, pushing it into apparent failure states. And how will the program respond? By breaking, misbehaving, or failing in some way? There is an inherent tension between granting players creative agency and the limits of what a programmer and designer can anticipate or even handle. A proper toy must respond gracefully to unanticipated play, a possibility that increases with creative tools that allow players to explore a vast quantity of states, possible arrangements of parts, pieces, and designs.

Budge recognized this problem, which is perhaps most obvious with the polygon tool. As Budge puts it, with the polygon tool it was "really hard to prevent people from making crazy stuff. I just let people make crazy stuff." He was forced to invent an algorithm for dealing with poorly formed polygons (Budge 2013). Although the manual warns about poorly formed polygons, in other cases it condones and encourages such boundary pushing behavior. For example, the manual explains how you can hide objects by painting out their pixels, transforming what might in one sense be a bug into a feature. Wright would later propose accommodating this kind of play, which I argue is inherent to open ended software toys, as designing accounting for failure states and making them enjoyable (Wright 2016).

[33]It does take some effort to get up and running with a playable table. *PCS* comes with a handful of finished sample tables, for those who might not have the patience to design a table and just want to play one. Also, these

Figure 5.9: Front (left) and back (right) of Pinball Construction Set's *packaging, redesigned by Electronic Arts for the version they published (1983).*

An exquisite degree of *manipulability* is offered. It wasn't just the gloved hand that served as the player's cursor, but the *direct manipulation* it symbolized and mediated. On the cover of the edition Electronic Arts published[34], hands cradle a giant floating pinball.[35] The ways in

demos serve to illustrate the kinds of things one can make. Like the illustrations of finished pieces on the back of a construction set toy like Lego, they feed the fantasy of what one might make, scaffolding creation through guidance and inspiration. Beyond that, since these tables are fully editable, they also function as starting points for tinkering, and range from bare bones to quite complex. The default blank table one gets on launch, in fact, isn't entirely blank, but an incomplete and suggestive layout that scaffolds the design of a table. It contains no parts, but functioning very much like a blank Mr. Potato Head face, strongly hints—to those with even a passing familiar with pinball—where the flippers, ball launcher, and other parts should go. Such a design move is right out of the Xerox PARC GUI design playbook, which notes that "Creating something out of nothing is a difficult task … it is easier to modify an existing document or program than to write it originally" (Smith et al. 1982). I followed a similar tack in the design of the *Spore Creature Creator* (2008).

[34]Trip Hawkins eventually prevailed upon Budge to allow Electronic Arts to publish *PCS*, which had seen only modest sales with BudgeCo. On all platforms, *PCS* sold over 300,000 copies (Hague 1997; Budge 2013; Maher 2013). Budge's involvement with Electronic Arts also exemplifies the transformation of a nascent consumer software industry into a more mature form—although "mature" might be the wrong term for this particular industry. EA unsuccessfully marketed Budge himself as a kind of rockstar "software artist," and modeled their marketing and packaging after the music industry. Games weren't sold in plastic bags, but came in handsome album-like enclosures, and prominently featured the game creators (Maher 2013). Norman Seef was hired for their famous "Can A Computer Make You Cry?" advertising campaign, in which Budge can be seen wearing an out of character studded leather gauntlet which Susan Kare had helped pick out with him for a punk rock themed Apple party ("Bill Budge Pioneer Award Panel" 2011).

[35]The packaging, like all of EA's early titles, mimics that of a music album. Whereas BudgeCo's marketing

which manipulability was marketed points back, deep into the design history of the graphical user interface. While Budge and his sister had marketed *Pinball Construction Set* as having "the simplicity and freedom of interaction of a toy," Electronic Arts's packaging was more aggressive, and described it as "Power. Pure, sheer and unadulterated. A nearly telepathic link between you and the machine." This latter formulation directly echoes Licklider's "Man-Computer Symbiosis," hinting at the design trajectory of direct manipulation interfaces, from Whirlind and SAGE—a flight simulator and then a handheld light pen interface—through ARPA and PARC, and from there to Apple and then Budge. The conventions and ideology of direct manipulation had been cultured for decades, manifesting in military flight simulators, light pens, and mice. It took on a toy-like quality through the efforts of people such as Seymour Papert and Alan Kay, who turned it towards children, self-directed learning, and creativity. It was through this latter form, in the conventions of the graphical user interface, that it reached Budge. Budge had created the most popular manifestation of a tradition that had until then existed only in rarified form.[36]

conveyed playful ease of creation with both implicit (primary colors and shapes) and explicit ("software toy") references to toys, EA's visual design and copy offers a different inflection: "Power. Pure, sheer and unadulterated. A nearly telepathic link between you and the machine." In this design, the hands reaching into the lattice emphasize direct manipulation, power over the ball and board, and human-computer intimacy. This man-machine symbiosis articulated in image and text describes the power to shape a digital world and change its rules—from the rules of the game to the laws of physics. In contrast to BudgeCo's materials, which describe a toy-like model kit, play is described here as an unfettered power fantasy. This move, also reflected in the physical packaging of games—from plastic baggies to nicely crafted boxes—marked a broader shift underway in the game industry, from amateur to professional marketers. It also foreshadows the hyper-masculinized way in which video games would come to be marketed.

[36]*PCS* was the first time a wide audience interacted with a direct manipulation graphical user interface, a fact that the BudgeCo marketing materials emphasize materially ("…has a user-interface which until recently has been available only in expensive systems like Apple's LISA®.") and experientially ("the simplicity and freedom of interaction of a toy.").

It's beyond the scope of this chapter to delve into the full spectrum of historical influences Budge was drawing from, but I want to call attention to what was inherited from the graphical user interface paradigm. The tradition and design conventions of manipulability carries within it the mutually reinforcing impulses towards both *toy*-ness as well as *power*.

The fundament of this work rests, in part, with the research and institution building work of J.C.R. Licklider, who sowed the seeds of computer science, and advocated a particular vision of computing as interactive, per-

One of the most important legacies of *Pinball Construction Set* is that by pushing genre boundaries and achieving commercial and critical success, it gave license to developers such as Will Wright to confidently pursue their interest in creating *software toys,* something that wasn't a game, and couldn't be won or lost, as well as *construction sets,* experiences grounded in scaffolded construction based play. The latter was made possible by drawing upon the direct manipulation conventions of the graphical user interface. By allowing players to easily and enjoyably make pinball tables, bringing them to life, and encouraging their sharing, Budge had transformed design into play. It was a Tinkertoy for the digital era. *Pinball Construction Set* harbored the profound idea that an experience based upon open ended construction play could commercially thrive amidst a landscape of computer games. Players didn't need explicit goals. They just needed to be gently scaffolded with the right templates and tools.

In taking inspiration from *Pinball Construction Set,* Wright adopted the framing of a software toy, the idea of a construction set, and the specific user interface design conventions that made all this possible. He decided "to design a fancy GUI [graphical user interface] on the Commodore … these little icons at the bottom. I thought I was so cool at the time." Arcane keyboard keys were replaced with an icon palette. In truth, it was an immature effort. *City*

---

sonal, networked, and graphical, and received his first taste of interactive computing while working on Forrester's Whirlwind project, specifically the light pen. Licklider's vision owes a deep debt to the interactive graphical computing of Whirlwind, which grew directly out of a flight simulator, a specific genre of military play. The interactive quality of that project—humans and computers convening via a graphical display which responded to a special "light" pen—is directly attributable to its genesis as a flight simulator. It was the flight simulator's need for responsiveness—unusual for a digital computer at the time—that set the stage for the development of an interactive graphical interface.

While the flight simulators and light pens reflected a military-academic orientation, learning and children became part of the tradition of computational manipulability through the work of people such as Seymour Papert (MIT, Logo), who then inspired Alan Kay to work with children and build computer systems for children. These child oriented designs were playful, accessible, and sought to enable self-directed creativity and learning. All of these systems, from the military flight simulators to the work of Kay and Papert reflect a deep entanglement with play (Redmond and Smith 1980; Kay 1987; Waldrop 2001; Small 2013).

All of this is to say that it should not be surprising that the GUI was perfectly suited for a digital toy like construction set, as the GUI itself was shaped for precisely such playful, creative, embodied, and child-like uses at Xerox PARC.

Planner lacked the concreteness and tangibility *SimCity* would eventually have. There was no mouse support. The graphics were too abstract. And the graphical user interface, despite being inspired by Budge's example, was hampered by both the Commodore 64's hardware and Wright's inexperience with these new interface design conventions. At the time, for Wright, the "GUI was the row of icons." Nonetheless, a seed had been planted. Later, when *City Planner* was transplanted onto the Macintosh platform, its interface, along with the feeling of manipulability, would sprout and flourish (Wright 2011; Wright 2016).[37]

SimCity

Toy or Game?

Proceeding with the idea that *City Planner* might be something others might enjoy, he began showing it to others, and made two surprising discoveries. First, that people didn't just like to build; they loved to destroy. Early players took great pleasure in using the bulldozer tool to destroy and trifle with cities. This led Wright to implement disasters as well as the iconic

---

[37]This early prototype lacked the concreteness that *SimCity* would later have. There was no mouse support, and the graphics were simpler. Although both *City Planner* and the Commodore 64 version of *SimCity* didn't have mouse based interfaces, as the canonical Mac version later would, even this early version was informed by the graphical user interfaces emerging into the public at the time. Wright attributes his use of graphic icons for tools in the Commodore 64 prototype to *PCS* (Wright 2011). Budge's self-published *Pinball Construction Set* came in 1982, one year before the 1983 release of the Apple Lisa, the predecessor to the Macintosh, and *Raid on Bungling Bay*. *Pinball Construction Set* would have been where most people would have first encountered the graphical user interface conventions that have now became commonplace, as both the Lisa and Macintosh were very expensive machines.

Although Wright has cited *MacPaint* as a big inspiration for SimCity (Donovan 2011)—clearly evidenced in the canonical Macintosh release's user interface—the Macintosh wouldn't be released until 1984. (The precursor Lisa machine, released in 1983, cost $10,000 in 1983. The most readily accessible prototype of what would become the widely popular graphical user interface was in *PCS*, originally self-published in 1982, as BudgeCo's marketing materials claim.

Wright explains that even to build the Commodore 64 icon based interface required some programming wizardry. He had to use an interrupt to switch graphics modes between the character and high resolution graphics modes that drew the map and tool menu's icons (Wright 2016).

Godzilla-ish monster that would later come back to haunt him. Second, not everyone got it. Despite their initial enthusiasm, Brøderbund, the publisher of *Raid on Bungeling Bay,* his first game, kept pushing him to turn his construction set into more of a traditional game, which he resisted. This led Wright to shelve the project, only to be revived after he met Jeff Braun, an entrepreneur who saw great potential in Wright and his unusual *City Planner*.

While *City Planner* had renounced the violence of a game like *Raid*, the impulse to destruction and mayhem reemerged in a new form. Players would initially be cool to Wright's toy, but their interest would perk up when they discovered the bulldozer tool. The bulldozer was there "to erase mistakes," but players would invariably "blow up a building with it by accident. And then they would laugh. And then they would go and attack the city with the bulldozer" (Rouse 2001). "[T]hey'd start running it all up and down the downtown area with this maniacal laugh. They just loved it" (Silverman 1999). While most contemporary games, he reasoned, "were about destruction," the bedlam he was witnessing was actually a form of learning. The violence "really intrigued me, because it was like someone coming across an ant pile and poking it with a stick to see what happens" (Rouse 2001).

> I don't think it really has a lot to do with violence. It's about exploring the dynamics of the system. When they start an earthquake in SimCity and see fires and rubble, they see how how alive and fragile the system is. It builds the illusion in the player's head that the simulation is real (Silverman 1999).

Destruction was about the thrill of agency, the illusion of aliveness, and cultivating empathy and understanding. The city's fragility inspired players to nurturance. Having unearthed the city's vulnerability, players would become hooked by the prospect of rebuilding. Easy mayhem gave way to thoughtful creation (Rouse 2001; Wright 2016). But it wasn't just players who learned from their failure, but Wright, who "finally decided, 'Well, I might as well really let them get it out of their systems, I'll add some disasters to the game.' And that's what gave me the idea for the disaster menu." With the menu in place, randomly occurring disasters were an obvious next step (Rouse 2001).

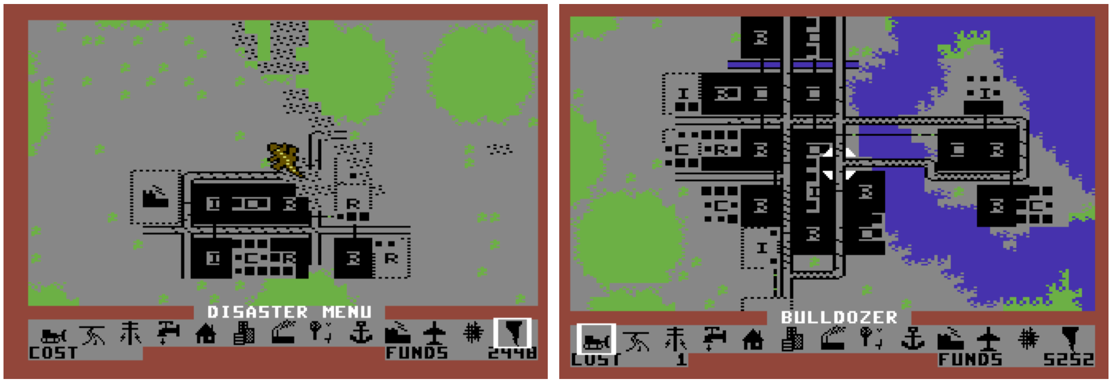


Figure 5.10: Screenshots from the Commodore 64 release of SimCity, *which is virtually identical to the version he would show Jeff Braun (Braun 2016b). Taken with a C64 emulator.*

Around 1985, after about six months of working on *City Planner,* Wright returned to Brøderbund. He showed them his new project, and "[t]hey agreed to pick it up, and we had a contract for it and everything." Brøderbund and Wright, however, had very different ideas about what the finished product would be. Whereas Brøderbund expected a traditional game, Wright was pursuing a vision of a software toy and construction set. "And I worked on it for about a year to the point where it was where I wanted it to be. And they kept thinking it wasn't finished. They kept saying, "When is it going to be a game? When is it going to have a win/lose situation?" (Rouse 2001). In response, Wright experimented with adding an election. Periodically, if your numbers sank too low, you would be booted from office. (The residue of this design is evident in *SimCity's* purely informational approval rating.)

But this didn't work for Wright. He envisioned something "much more open-ended, more of a toy" (Donovan 2011), which "felt more like Legos to me than a standard game" (Kushner 2005). Whereas one *fails* at a game and loses, Wright was fascinated by failure. Who says that failure is bad? Or that it isn't enjoyable? And who, after all, gets to define failure and success?

In the toy he was making, players decided for themselves. Besides, for Wright, "most effective learning is failure-based … 90% of winning is failing." Failure is "usually interesting," as it helps you "understand why" things went wrong. "You're elaborating that model, you're building the system in your head with every failure." Wright wanted his players to be fascinated by, delight in, and learn from their failures (Wright 2016).

After about a year of this, the arrangement with Brøderbund ground to a halt. According to Wright, they "didn't see how they could possibly sell it. And I just left it there, and they left it there, and that was that" (Rouse 2001). Wright thought the project was complete, but Brøderbund didn't like it. As Brøderbund's interest waned, so did Wright's (Wright 2016). "So it just sat on the shelf for several years" (Donovan 2011).

City Planner was a kind of failure. Luckily, it was a fascinating failure, the kind you might learn from. Fortunately, since Brøderbund had never paid for any of its development, Wright had no financial obligation to Brøderbund, and the rights, at least in theory, remained with him. But to grow into its finished form and reach players, the project needed a software publisher willing to take an enormous creative risk. A publisher whose aesthetic sensibility was sensitive to the creative promise of both Will Wright and his unusual city construction set, yet possessed of the business savvy to shepherd it to completion without ruining it.

Jeff Braun

Jeff Braun (born 1955), the son of a grocery story clerk and elementary school art teacher, grew up in Los Angeles. From a young age, he was interested in technology. "I knew … that if it buzzed, beeped, flashed, whirred, had a button—I'm in," says Braun. His first contact with computers, at the age of six, was through a family friend who worked at the aerospace company TRW and showed him the room sized computers that calculated the trajectories for the NASA moonshot. Braun took home ASCII art diagram printouts of the trajectories, which he kept in a drawer for years. He loved the California Museum of Science and Industry, where he remembers playing an electronic tic-tac-toe machine that always won. His mother, an artist, took the family to many museums. As a teenager, he fell for Oldenburg's kinetic *Ice Bag* sculpture at the Los Angeles County Museum of Modern Art (LACMA).[38] He saved a poster from the exhibit, which hung in the Maxis office for many years. "I always kept that poster because I thought that it's really about art and technology." He loved how *Ice Bag* groaned, exuding emotion and life (Braun 2015; Braun 2016a).

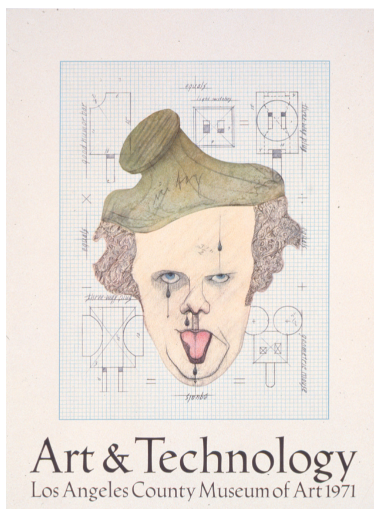


Figure 5.11: Braun's cherished poster from the LACMA Art and Technology exhibit. The image is a Claes Oldenburg self-portrait, and is related to his kinetic Ice Bag *(1970) sculpture.*

As a student at the liberal arts Pitzer college he interacted with computers for the first time. *Eliza* and *SpaceWar!* made a huge impression on him. "I thought it was too cool. We were going through outer space and we were talking to our psychiatrist." He discovered Philip K. Dick, which he found profoundly affecting, eventually amassing a massive collection of

[38] Oldenburg's sculpture emerged from the *Technology + Art* project (1969-1971, curated by Maurice Tuchman), which matched artists with technology corporations (Tuchman 1971).

manuscripts.[39] He took courses in a wide variety of fields, but, like Wright, lacked the patience to learn to program computers with punch cards. "I'm not gonna sit here with this stack of cards. … I don't have the patience to sit and knit a blanket." Braun moved to Lake Tahoe to ski, eventually graduating from Lake Tahoe Community College.

Around 1981 Braun moved to San Francisco and found work at a company that repaired pinball machines and video games, at a warehouse full of broken coin-op machines, which is where he first picked up the scent of video gaming's commercial possibilities. "And so I knew all the games and I played all the games. And of course, you had to test everything. And so that sort of built my first little introduction to "This is gonna be a huge damn industry."" He loved the arcade video games *Defender* (1981), *Robotron* (1982), *Q*Bert* (1982), and *Tempest* (1981), the pinball machine *Eight Ball Deluxe* (1980), and the text adventure games made by Infocom, such as *Zork* (1981, for personal computer). Because it was a cash business, according to Braun, organized crime was involved in the operation and maintenance of coin-op amusements: "eventually I got hired away by the Mafia and then the Hell's Angels."

Braun's first entrepreneurial effort was in barcoding, helping organizations like San Francisco MUNI, AC Transit, and the US Army do inventory management. Then, in 1985, Commodore released the Amiga 1000 personal computer, a machine with impressive multimedia capabilities. Excited by the possibilities of this computer, Braun decided he was "really into fonts" and hired the programmer Edward Kilham, who he met at an Amiga enthusiast meeting, and together they designed a product called *Calligrapher*. A font-editing system that enabled system-wide color font support, *Calligrapher* enabled the Amiga to compete with the special purpose Chyron video titling hardware that cost tens of thousands of dollars.

About a year later, Braun sold his small font company, Interactive Softworks, to a friend

[39] Braun eventually donated this to California State University, Fullerton, where Dick was a professor.

who wanted to enter the software business. Meditating on his experience with coin-op games and multimedia personal computing, he came to believe "that games were going to be the next big thing." Braun lacked a plan as to what shape his new games oriented venture would take. "I took that money and decided I was going to do video games, but I had no clue what that meant or what it would be, or how it would look, or who would do what, or anything. I had no clue" (Braun 2015).

Braun asked his friend A.J. Redmer, who had a stronger connection to the game industry, how he should go about getting into the game business and meet game programmers. Redmer's answer was beer, pizza, and parties for playing computer games like *M.U.L.E.*—the kinds of events that Braun and Redmer would attend together. Just as Braun had meet Kilham, the programmer for *Calligrapher*, at an Amiga enthusiast meet up, Braun sought out game developers in the same way. He started attending the pizza parties thrown by his friend Chris Doner, where guys would hang out, eat pizza, drink beer, and play computer games. It was the kind of party where you could meet John Draper, the infamous phone hacker nicknamed Captain Crunch. Wright's friend, a high schooler named Mick Foley, dragged him to some of these parties, where he met Braun (Braun 2015; Foley 2015).[40] Braun recalls asking Wright, shy and reticent, what he was working on:

> "Well, you're not gonna like mine. It's really awful."
>
> "Why wouldn't I like it?"
>
> "No one likes it."
>
> "Why doesn't anyone like it?"
>
> "Well, it's not a game."

[40] Will Wright's friend and neighbor, a high school student named Mick Foley, who had playtested *Raid* during its development, had brought Wright to the party. Foley would later build the Bern and Detroit scenarios of *SimCity*, and go on to become a programmer at Maxis. Foley recalls meeting John Draper and Braun talking a lot about barcode scanners (Foley 2015).

"Reallly? What do you do?"

"Well, you build a city."

"Really? What kind of computer?"

"Commodore 64."

"No way."

"Yeah."

"So which city?"

"San Francisco."

"So… do you have a bridge?"

"Yeah."

"Do you have the Transamerica Building?"

"Yeah."

"Do you have Market street?"

"Yeah."

"Commodore 64? Can I come and see this?" (Braun 2015)

Intrigued, Braun visited Wright's home, which contained his newborn daughter Cassidy from his recent marriage to Joell Jones in a bassinet, to see his demos. (*SimCity* is dedicated to Cassidy in the manual.) The first demo Braun saw was called *Probot* (*probe* and *robot*), an artificial intelligence inspired game that Wright had made before turning in earnest to *City Planner*. It was another incarnation of the hydraulic arms he was fascinated by, except this time his medium of representation was a software model, a game. Using a joystick, the player drove a robot arm mounted on a four wheeled platform through an abstract isometric landscape filled with platforms, passageways, and ramps. Rivulets of water coursed through, emerging from three different sources, each of which occasionally spawned a different shape: a cube, a sphere, and a pyramid. The robot arm, controlled by holding down the joystick button, allowed the player to collect and organize the shapes while negotiating and trading with three legged alien robots that were attempting to do the same. Braun wasn't interested. Then, Wright showed him *City Planner,* which he described as a paint program for a city (Braun 2015; Foley 2015; Wright 2016). Braun recalls his excited response:

> "Oh my god!" … Look, what he's doing! I mean seriously, to this day, go look what he did on the [Commodore] 64. … He had it so that you could hit a button and a monster would come in, and he had another button and it was an earthquake, and there's fires, and there's a little helicopter going around. I was like, "No way!" (Braun 2015)

Unlike Brøderbund, which wasn't interested in Wright's city building program, Braun was floored by what Wright was evoking with the C64's meager hardware. More importantly, Braun didn't share the game-centric reservations voiced by the game publisher, whose conservatism had trouble fathoming the appeal of an experience one couldn't win or lose. Braun had no trouble with the project not being a game. "So it's something else. So I just said, "I don't care." … I didn't have any issues with not being a game or being a game. I wasn't trying to draw a semantic [distinction]. … Like I care."

Because of Wright's relationship with Brøderbund, Braun wanted to ensure the rights for the project were clear. This should have been the case since Wright had invested all of his own money into the project. Braun sent Wright to Don Daglow, an executive at Brøderbund, along with a release letter for him to sign acknowledging Brøderbund didn't own it.[41] To be safe, Braun instructed Wright not to tell Daglow that anyone else was interested in the project. Wright acquired the release, setting the stage for a partnership. Braun wanted to run a game

[41] Incidentally, Daglow, earlier in his career, had produced an early simulation style game, a competitive two player game called *Utopia* (1981) (Wallis 2006; Loguidice and Barton 2009).

publishing house, and the business arrangement they struck was that Braun, operating in the role of a publisher, would license Wright's project and pay for its conversion to Macintosh, Amiga, and PC (Braun 2015).[42]

Building Micropolis

City Planner needed to be translated into a form which would reach as wide an audience as possible. Everyone, not just Braun and Wright, should be able to get what it was and play it. Part of this meant finding a better name. Braun didn't like "City Planner," and so he renamed the project "Micropolis," which was better, but still awkward (Braun 2016a; Braun 2016b). Translation also meant supporting newer platforms, like the Amiga, Macintosh, and PC. The user interface that enveloped and made tangible Wright's simulation would need to be vastly improved. Better art was needed, as well as a manual, packaging, distribution, and sales. Players needed to apprehend what kind of thing *Micropolis* was, and understand what was happening when they played it, to see and feel and act within it, and empathize with the inhabitants of Wright's miniature world as well as he did. Wright had transformed the esoteric subjects of city planning, system dynamics, and cellular automata into his own private plaything; now he, in partnership with Braun and a small team, would transform it into a plaything for everyone else.

A growing development was installed in Braun's two bedroom Moraga condominium. The second bedroom became the office, in which Braun's cherished LACMA Art and Technology poster hung, and the living room and kitchen were transformed into work spaces. Wright

[42]Wright received royalties for *SimAnt*, *SimEarth*, and *SimCity*. Later, when Maxis took on venture capital funding, Wright would become more of an owner of SimBusiness, and relinquish his royalty arrangement (Kennedy 1997; Wright 2016). According to Braun, Wright's royalty percentage was translated into an ownership percentage of Maxis—since "it was all his products anyhow" (Braun 2016a). Maxis did sell products by other developers, but *SimCity* was the cash cow.

continued work at his home, preferring to work late at night when he wouldn't be disturbed, revising and rewriting his hand written C64 assembly code in the C programming language, which could then be recompiled for other hardware platforms (Rouse 2001; Braun 2015; Wright 2016). From the condo, the development team could hike around the Moraga hills in the middle of the day, and return in time to get a bowl of soup from the Chinese restaurant just before it closed (Bremer 2015).[43]

Figure 5.12: Developers at Braun's condo/workspace. From left to right: Jeff Braun, Richard Bagle, David Caggiano, Leda Zudowski, Michael Bremer, Will Wright. (Names and photograph courtesy of Braun; additional help with names from Bremer.)

Creative authority emanated from Wright. Braun built a team around him, and describes

[43]Michael Bremer, who wrote the manual and conceived of the name "SimCity," recalls a late night session with Richard Bagel, when the two of them were working in their underwear because of the 95° heat. Bremer recalls telling Bagel, "Rick, remember this night. Someday, we're gonna think of this as the good old days." He continues, "[w]e both just like got it and smiled and just kept working. And I often do think, "Yep. Y'know, those were the good old days" " (Bremer 2015).

the organization he engineered as a "wagon wheel and Will [Wright] was the hub." If Wright needed help with something, or didn't want to do something, then Braun would plug someone in, a spoke, and see if it worked. He describes his role as an impresario. "My focus was really about trying to extract the best. I knew Will was a genius. … I knew we had to get him out in the best way possible." Braun didn't want to leave a mark on *Micropolis;* it was Wright's project. "I don't think I have a single suggestion that's mine in the game—intentionally" (Braun 2015). While Wright receives most of the credit for titles he has worked on, the teams he works with are integral and often unacknowledged contributors (Cambron 2002).

Additional people were hired to flesh out the interface between the simulation and the player. Wright handled the simulation, but other programmers would do the user interfaces and support each platform's conversion. At first, they focused on the Macintosh, which became the starting point for conversions to the Amiga and PC. Front end programming was done by Robert Strobel, who is also credited in the manual for design contributions—probably for the user interface. To make the graphical characters that were quilted together to form the landscape, Braun again tapped his network, and hired Don Bayless, an art therapist living in Kansas who had sent Braun design feedback on *Calligrapher* and made shareware fonts with it. Designing the characters of a city was not unlike designing the characters of a font.[44] The game programmer and sound designer Steve Hales made the sound effects. (Of course there were more contributors.)[45]

[44] He and Wright collaborated closely over email (Bayless 2016).

[45] A young programmer, Brett Durrett, helped out with some of the front end programming. Durrett worked on things like drawing the supplementary city maps and the world editing tools. Dan Goldman did the PC port. Brian Conrad completed the Amiga port of the Mac that Brian Witt began, and introduced native multitasking to it (Conrad 2016). Wright's friend Mick Foley worked on the Bern and Detroit scenarios. He recalls Wright giving him a map of Detroit, and tasking him with creating a scenario based upon it (Foley 2015). (The manual does not credit Mick Foley.) There are more people to credit; I'll discuss some more of them later, or you can consult the credits for each version.

Conrad, in fact, had initially connected with Braun with the idea that he would do a water ski game for Maxis.

Thanks to the platform affordances of the Macintosh, the city became more vivid and manipulable. One of Wright's brilliant accomplishments was getting his sophisticated simulation to run on the Commodore 64's now meager 1Mhz 8-bit CPU. But with the newer platforms they were targeting, with their high resolution graphics and beefy 7Mhz 16-bit CPU's, they could do so much more. The core of the simulation would stay the same, but many things would change (Rouse 2001; Wright 2011).

Fine detail was added, as well as better schematic overviews. The awkward user interface of *City Planner* gave way to a lovely Macintosh style graphical user interface. The world become more touchable, easier to control, and its visualizations easier to navigate. It became easier to see and understand Wright's microworld. The Commodore 64 version, while evocative, was quite abstract. You had to look at secondary maps and graphs to really understand what was happening in your city. In the new version, those secondary informational sources stayed, but also became more integrated with the main view. Demand for residential, commercial, and industrial buildings—the guts of the simulation Wright had adapted from Forrester—could now be shown in the main window. Buildings now looked like what they were, thanks to higher resolution graphics and multi-character building art. Belching smokestacks suggested pollution, and residential buildings looked like housing. The style of buildings indicated the land value in the underlying simulation. Messages, pre-made make believe scenarios, and more detailed evaluation feedback guided play and scaffolded interpretation of the city.[46] The underlying representation also became more finely detailed. The map got bigger. Waterways went away, but fire stations and police departments were added, along with new

---

This never came to fruition, and Conrad ended up completing the Amiga *SimCity* port after Brian Witt quit (Conrad 2016).

[46]It was *PanzerBlitz* that had introduced the idea of scenarios and open-ended play to the complex simulation board games Wright loved (Wikipedia, PanzerBlitz, 2016).

simulation layers. There was now a stadium and nuclear power plant. You could name your city. Disasters automatically happened, there were speed controls, and you no longer had to manually bulldoze if you wanted to build on a forest or waterfront. To gracefully handle all of this complexity, and allow simulation, representation, and user interaction to smoothly unfold, additional technical wizardry was required.[47] It looked and felt more like a living city, yet retained its schematic clarity. The microworld became more supple, vivid, tangible, and appealing.[48]

Braun sent a development Mac build to his childhood friend Michael Bremer in L.A., hoping to interest him in joining the fledgling effort. Bremer, an aspiring writer, had been working on a science fiction story about artificial intelligences who hung out online in a place called "SimCity." Bremer suggested the title "SimCity" to Braun, who still wasn't satisfied with the name "Micropolis."[49] For Braun, *SimCity* was what he was looking for. It was a place that Sims lived. For Bremer, the name was perfect not just because it led to many wonderful puns,

[47]Wright had to not just creatively allocate CPU time amongst the various city simulation tasks, but between the simulator and the user interface. The power scan simulation was especially costly. Interrupts were used to switch between the simulator and the graphical user interface. To make everything smooth, Wright implemented a primitive preemptive multithreading system; i.e. there was "one stack for the simulator and one for the UI." This design decision would later need to be revisited and refactored for the Macintosh Multifinder and other platforms (Foley 2015; Wright 2016). Wright has said that the Commodore version also had a kind of multitasking operating system to keep everything running (Wright 2011), but that version still tended to pause during power scan, which became quite slow as the city grew complex. In the Commodore version, a special dot is shown to indicate that it is stuck doing a power scan, which suggests that Wright must have elaborated upon the multitasking system for the Mac version, which uses a preemptive multithreading style system to allow the simulation and user interface to smoothly proceed. The black dot was eliminated. (Don Hopkins would optimize the power scan code at some point while porting *SimCity*.)

It was Brian Conrad who replaced the emulated multitasking code Wright had written for the Mac with the Amiga's native multitasking abilities. Conrad points out that Wright carried forward the interrupt based multitasking techniques used by Commodore 64 programmers to the Macintosh (Conrad 2016).

[48]The fact that the final Commodore 64 release of *SimCity* is a virtually unchanged version of *Micropolis*—aside from the title screen—means that I can compare the final version to this early prototype, and see what changed (Braun 2016b). The manual of the Commodore 64 version, in its discursive practices, resembles an artifact from Forrester, working overtime to explain what everything is, what's happening, and how it works. The Mac version is more self-explanatory.

[49]And in any case, there was a hard drive manufacturer called that.

but because it meant that it was no longer "Micropolis," an anonymous city, but a specific city, a city filled with people that you presided over as mayor. Bremer was hired as the writer, and wrote the manual. His title became "Voice of Maxis," and for the first time in his life, his quirky, creative, and funny ideas were welcomed with open arms. Bremer recalls Braun exhorting him to "get weirder" (Braun 2015; Bremer 2015).[50]

Braun hadn't yet incorporated his new business. To do this, he needed a lawyer, and again his friend Redmer came through with a suggestion: that he hire his friend Bob Derber. Braun describes Derber as "probably one of the most confident, trustworthy, loyal lawyers I've ever met. And also, the word psychopath or maniac comes up in my mind when I think of Bob." Wright compares him to Hunter S. Thompson. Derber, whom Braun describes as "brilliant," handled the incorporation, but to do this he needed a business name, and the name he chose was SimBusiness (Braun 2016a; Wright 2016).[51]

---

[50] This new title had the added benefit of not having the same name as a disk manufacturer named Micropolis. As an aspiring fiction writer with some experience in technical writing, it makes perfect sense that Bremer would come to see his documentation as both "part of the experience" as well as "customer service." Through careful testing and writing, they could save tens of thousands of dollars in tech support call costs (Bremer 2015).

[51] According to Braun, incorporation happened about six months after beginning his partnership with Wright. Braun recounts a story about Derber:

> And so, in the middle of the meeting, he'd just stop the meeting and he'd say to somebody who doesn't know about Bob Derber and—because he comes across as a hard nosed lawyer—he'd just point to the person and say, "Excuse me, but how do you think I'd look in a blue chiffon dress?" And they'd look at him. And they're like, "Did I just hear what I think I heard coming from this hard nosed lawyer?" Because up until that moment, he's like totally like in their face lawyer. … And it always disarmed them, and then they'd laugh. And then things would get easier (Braun 2016a).

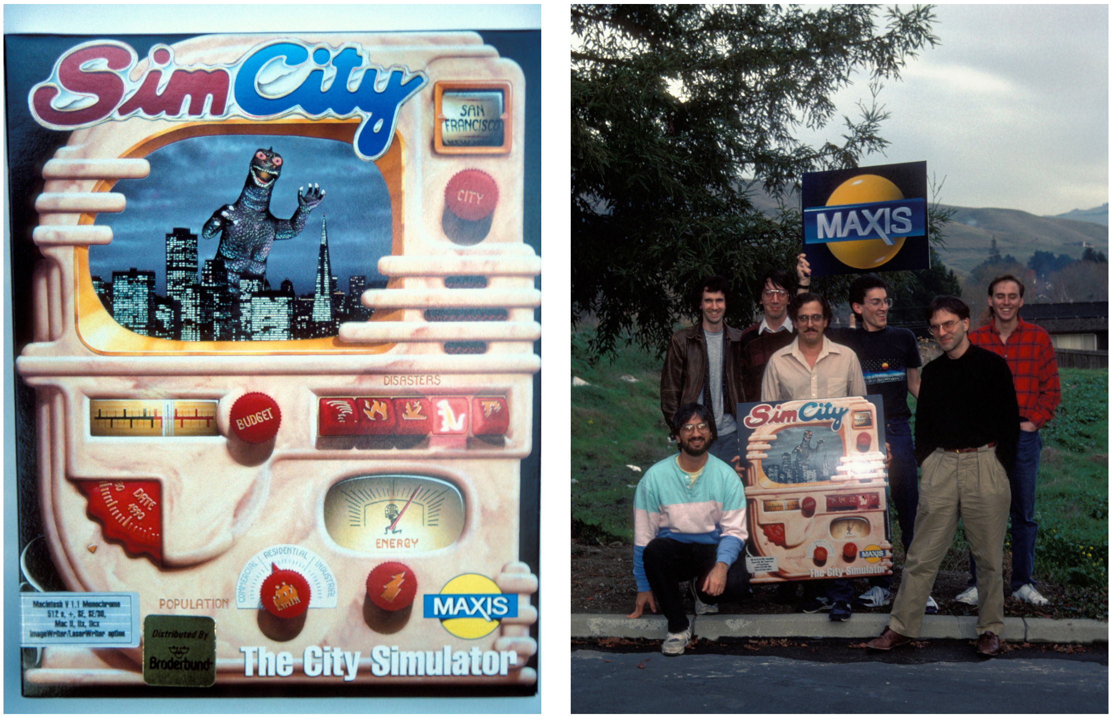


Figure 5.13: Left, original SimCity *box. Right, some members of Maxis. From Left to Right: Jeff Braun, Daniel Goldman, Will Wright, Michael Bremer, Michael Paterson, David Caggiano, Tim Johnson. Illustration by Kurt West, and package design by Richard Bagel (Photos and names courtesy of Braun; additional help with names from Bremer.)*

Braun knew that his company's name would be on the box, but the name Derber had come up with, "SimBusiness," wouldn't do. Braun asked his dad, a word game aficionado, to help him come up with a name, specifying that it needed to be meaningless, have six or fewer letters, a small number of syllables, and be in the same vein as Atari. "Maxis" was the name he came back with, which he recommended because having an "x" in the middle was cool, it had mother and sister in it ("ma", "sis"), and it spelled "six a.m." backwards. Braun loved it, and Maxis became the doing business as (DBA) title of SimBusiness (Braun 2015).

Another one of Braun's childhood friends, Richard Bagel, was contracted to do the packaging design. Bagel hired Kurt West for the illustration, which depicted a retro bakelite radio city control panel. Of the box, Braun says, "the idea was to cause questions more than answers.

Like, what is this bakelite radio thing with this monster coming out of the screen? What's going on here?" (Braun 2015).

The project eventually went over Braun's budget, which is how Wright came in as a co-owner. Recognizing that the project was special and that software development was unpredictable, Braun didn't impose a development schedule. Development proceeded for about a year and a half, but eventually Braun ran out of money. With Derber handling the legalities, Wright invested about $50,000 into *SimBusiness* in exchange for a 15% equity in Braun's company (Braun 2015; Braun 2016a; Wright 2016).[52]

As *SimCity* approached completion, Braun sought a larger publisher to produce, sell, and distribute the physical product. He returned to Brøderbund, but Gary Carlston, one of Brøderbund's founders, rejected a publishing deal, as *SimCity's* design still seemed too exotic and risky. The Carlstons, however, were seeking to grow their company in anticipation of going public, and Carlston offered Braun a favorable consignment deal—part of a new program they were doing.[53] Feeling that *SimCity* would be big, and not wanting to cut his teeth as a publisher on it, Braun first published a different title, Ralph Russell's *Sky Chase*, a two player dogfighting game for the Amiga. Brøderbund loved *Sky Chase*, and it became the first title Maxis published, and the first title Maxis distributed on consignment with Brøderbund (Wallis

[52]Braun recalls that Wright invested about $50,000 in exchange for a 15% stake. (He had already invested at least that much.) Braun says he later bumped Wright's share up to about 30% after *SimCity* became a hit. Braun held about 50% of the company—together Braun and Wright owned about 80% of the company—with the rest reserved for employees. Braun was always the controlling shareholder of Maxis, even after the venture capitalists were brought in, and then later when the company went public in 1995 (Braun 2015; Braun 2016a).

[53]Maxis would assume the risk of manufacturing and ship finished boxes to Brøderbund. Instead of the usual 20% royalty rate, Maxis would receive 80%. Sensing a big opportunity, Braun asked Carlston for an invoice for an initial shipment (a relatively small order of around 50 thousand units), which he then took to a new software box fulfillment manufacturing company in Concord. On the strength of the invoice, the ambitious upstart manufacturer advanced the cost of manufacturing (building the line, manufacturing, and warehousing). Braun recalls getting 45 day terms from the manufacturer, and 30 day terms from Brøderbund, so he could just take the revenue and repay the manufacturing advance without borrowing or risking more money. Effectively, Maxis risked little and received the lion's share of the revenue. Braun claims that Maxis grew to be the majority of the manufacturer's business (Braun 2015).

2006; Braun 2015).[54]

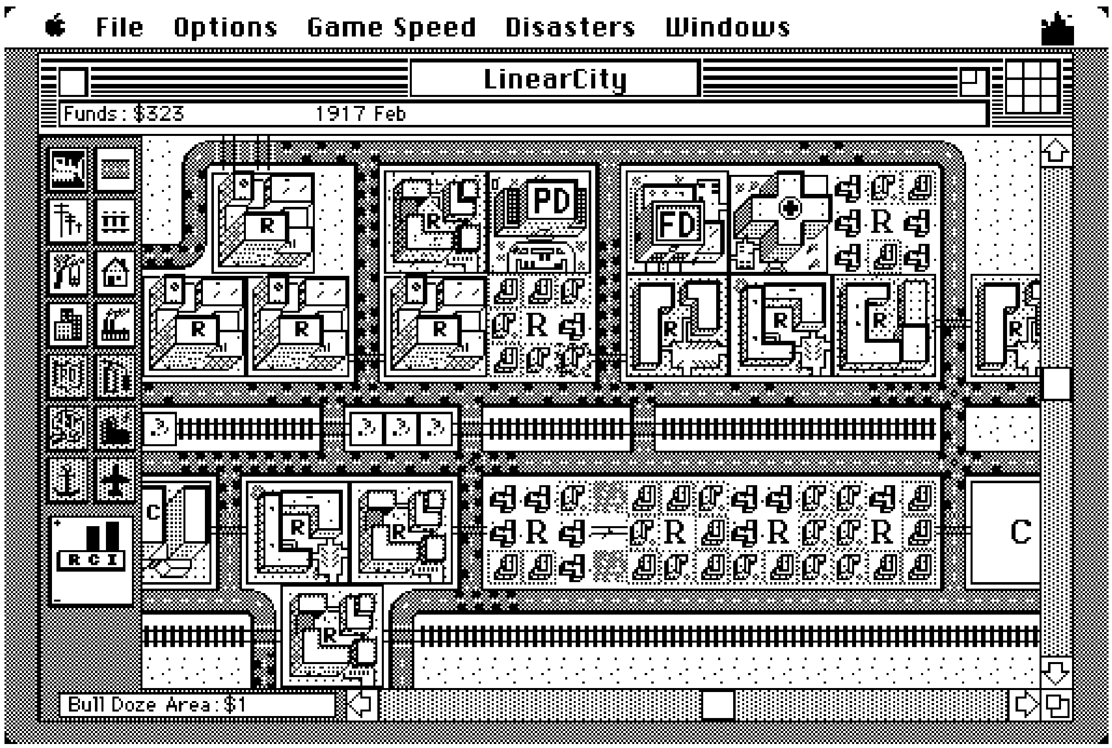


Figure 5.14: SimCity *for the Macintosh (Source: Wikipedia).*

SimCity was launched in early 1989 for the Macintosh. The Amiga version followed a few months later, with the PC release a year later. After shipping the Mac, Amiga, and PC versions, Maxis decided to sell a version for the C64. The code for the prototype C64 version, however, which Wright had written in assembly for its 8-bit CPU, was lost. In order to emblazon "SimCity" on its title screen they resorted to editing the object code directly. It is for this reason that the C64 release is a historically important snapshot into a developmentally early state of the project. Like an insect trapped in amber, the Commodore 64 release depicts Wright's *City Planner* before it blossomed into *SimCity* (Braun 2015; Braun 2016b).[55]

[54] Although Brøderbund plays the part of a foil in this story, both Braun and Wright express an enormous amount of gratitude and admiration to Brøderbund's founders, the Carlstons, and in particular Gary Carlston for support, guidance, and mentorship (Braun 2015; Wright 2011).

[55] My release date timings here are based on an interview with Braun (Braun 2015). An interview I did with

Reception

Wright credits the positive reception of *SimCity*, in part, to the demographic match between Macintosh computer owners and the audience for *SimCity*. While it was the PC version that would eventually bring in the bulk of revenue (Braun 2016a), the Macintosh and Amiga versions ushered in, according to Wright, a whole new audience open to such experiments. The mayor of Portland wrote a letter to Maxis, expressing how he wished running a city was as easy as it was in *SimCity*. A high school student orchestrated a play-off for the Democratic primary race for Providence's mayor (Juba 2015; Koebler 2015). This new demographic of players, according to Wright, skewed towards press, which reviewed and promoted *SimCity*. Sales of *SimCity*, in fact, didn't fully take off until a favorable full page review in *Newsweek* (Greco 1994; Donovan 2011; Wright 2016). There were a couple snags, like when Maxis was sued by Toho for infringing upon Godzilla.[56]

SimCity, like future Maxis titles, appealed to a different kind of audience than traditional

---

Wright roughly concurs with this sequencing (Wright 2016). More authoritative precision on the timing would require archival research. This sequencing makes sense given the fact that the Amiga (Conrad 2016) and PC versions were ports of the Mac version, and the Commodore 64 version was a repackaging of Wright's prototype.

[56]The Godzilla looking monster on the box and in *SimCity*, in the form of a giant monster attack, would give Maxis trouble. According to Braun, "The doll that was in the thing wasn't Godzilla. It was a Tyrannosaurus Rex, *we* say." The Japanese film company Toho, which created Godzilla, saw things differently and sued Maxis soon after the PC version launched. It didn't help that the press saw things as Toho did. Although Maxis never called the monster Godzilla in its materials, the press did, which led both Toho and Maxis's lawyers to believe Maxis culpable. Braun and Bob Derber, his lawyer, went to Los Angeles to meet with a Toho lawyer, and "pleaded poverty and ignorance." Braun claims that although Toho's lawyer felt sorry for them, he insisted he couldn't return to his boss empty handed. Braun's offer of a $50,000 dollar settlement was accepted, and the disaster on the box was changed from a monster to a tornado. The game remained unchanged (Braun 2015; Braun 2016a). Even if *SimCity* and ancillary narrative instruments (e.g. documentation and packaging) don't describe the monster as Godzilla to the player, its source code clearly refers to the monster as Godzilla. According to Foley, the box art and sprite art were from the same plastic Godzilla toy, and the sound effect for the monster was an audio sample from Godzilla (Foley 2015).

The irony of this is that in the late 1990's Toho approached Maxis about building a web site for an upcoming Godzilla film (Conrad 2016).

Later, the Red Cross called and told them the crosses on *SimCity's* hospitals should be blue, the actual color of hospital crosses, and not red, which was trademarked (Foley 2015).

games, which typically featured, says Wright, “arcade style action, graphics, very intense kinds of experiences. There were very few games that were laid back, more complex” (Pearce 2001). Maxis would go on to make

> the kind of games that I would want to play, and the kind of games that Jeff [Braun] would want to play, too. … We were twenty-something years old, and a lot of these games felt like they were targeted towards fifteen year-olds. There weren’t that many games targeted towards adults with different sensibilities (Juba 2015).

This sensibility dovetailed perfectly with Braun, whose ambition in starting a game publisher was to “do something different. I don’t wanna do the same thing. I was more into the PC. It was my thing. I wanted to appeal to everybody. Console was kind of like a kid thing” (Braun 2015).

SimBusiness

Whose playthings?

The marriage of play to commerce is always awkward. There is, at heart, a tension between the intrinsic quality of play—something done for its own purposes, in this case the crafting of a plaything—to the extrinsic nature of commerce. Software developers need the creative freedom to invent playthings for their own amusement. But within the context of a business, the developer-as-player must answer to external financial realities, production constraints, and the market of consumer-players. Reconciling the dynamically opposing forces of the developer, the play of the constructor-benefactor, with that of the business, whose king is the player and the capitalist, is a fraught challenge (e.g. Katayama 1996). Brøderbund wasn’t interested in taking a commercial risk on *City Planner.* They saw a failed game and a financial hazard. But Braun, the artistically inclined entrepreneur, saw in Wright’s plaything the potential for *SimCity,* a magnificent software toy construction set that could appeal to a broad audience. Braun's characterizes his objectives as often being "independent of profit," which, in any case, "is really sort of meaningless. It's vapor." What mattered to him was how you transformed people (Braun 2016a).

When people played *SimCity* they saw the name of the company, Maxis, but underneath this artifice it was SimBusiness, an enterprise predicated on the tenuous arrangement of transforming Wright's play into something the wider world might share in. The artifact of the constructor-benefactor would also have to fascinate the king. "No one in their right mind would make a game about city planning for everyone else," says Braun. When Wright made *SimCity*, says Braun, he was "making a game for himself." Braun saw in *City Planner* "something that others will enjoy," and in the case of *SimCity,* at least, "what Will [Wright] wanted lined up with the customers in a really nice way." But this wouldn't always be the case (Braun 2016a).

Maxis pursued a vision of software toys. According to Braun, "we always had this issue of: it's not a game, you know, win or lose." Following in the footsteps of Bill Budge, they described their creations as software toys. Braun and Wright divided responsibilities: "I'd run the business side of things and he'd run the programming side of things. And I never really told him what to do and he never really told me what to do" (Braun 2015).

Maxis was a quirky and free spirited place to work. The llama became the company mascot after Wright put it to a vote—a llama with a jetpack versus a tapeworm—immediately after dinner at the company Christmas party in Tilden Park's Brazilian Room. If you called Maxis and waited through the list of phone tree options, you'd get an option to "order decomposing meat products," which launched you into a Choose Your Own Adventure style story, complete with high quality sound effects, in which you serve tainted meat at a party, and the guests become ill and threaten to sue you. Eventually, you could choose to speak to an attorney or flee the country, which actually routed callers to Maxis's attorney and travel agent—who did not find it so funny (Foley 2015).

Wright followed his muse. As he had throughout his life, he followed his interests, learned about new fields, and built models. For Wright, each "game was almost an excuse to do years of research on a subject that I like" (Juba 2012). Immediately following *SimCity*, Wright developed *SimEarth* in consultation with James Lovelock, the originator of the Gaia hypothesis, which conceptualized Earth's organic and inorganic matter as a complex integrated self-regulating system. Released in 1990, *SimEarth* is one of the most sophisticated simulations Wright would make, and one that he remains extremely proud of (Rouse 2001). Despite his belief that *SimEarth* wouldn't be a commercial success like *SimCity*, which it wasn't, Braun believed in creating an environment in which Wright could engage his interests and creativity. Braun's job was to "do the best I can to make this work," which meant the manual, and generous amounts of packaging and marketing. Braun's guiding vision was to

> to protect him, in a sense, and create a structure where he can do his thing. It really was not about me and what I thought would be successful. It was about Will and getting his creative thing out (Braun 2015).

Braun knew it wouldn't sell, but wanted to support Wright as a creator. Braun describes his role as "nurturing talent" and "trying to get Will [Wright] to express himself like an artist." He adds that "talent like Will really needs someone to help them navigate their world … someone who believes in them and that is trying to grow them as an individual."

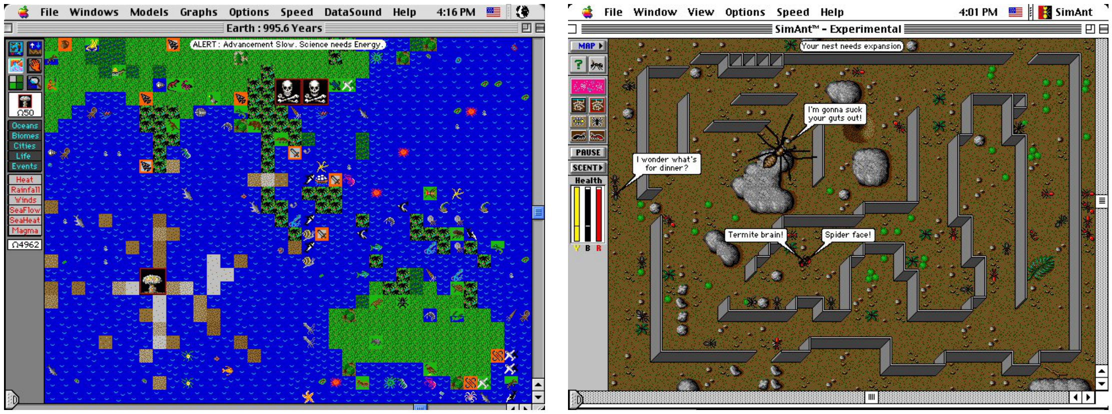


Figure 5.15: Left, SimEarth*. Right,* SimAnt*, which could be quite whimsical.*

SimEarth, released in 1990, reveals the commercially permissive atmosphere at Maxis. Very likely the continuation of Braun's receptiveness to *SimCity's* quality as a sui generis object, and a lucrative one at that, Wright was given free reign to experiment. A discomfiting artist and commercial patron relationship unfolded, a setting which nurtured new and boldly experimental works into existence, yet simultaneously threatened the financial viability of the entire enterprise.

While *SimEarth* pushes the boundaries of interactive simulations, it betrays the importance of the *construction set* quality it lacks. The problem with *SimEarth,* according to Braun, was that "it self-modified so you go and design your Earth, but then it decides it wants to be something else" (Braun 2015). It was a software toy, but it wasn't a construction set. While its design deserves close attention and admiration as another bold, ambitious, and ingenious simulation object, it also foregrounds how *SimCity's* aesthetic and commercial success is due less to the quality of sophisticated simulation, and more to something else. Braun's use of the phrase "self-modifying" provides an important clue, and that is that *SimCity's* success wasn't so much about the research and simulation design Wright loved, but its *construction set* quality. *SimCity* was about player creativity, a quality which wouldn't be heavily emphasized again until

The Sims. Eventually, both Braun and Wright became highly attuned to the critical role of this construction set quality; tools which inspired and scaffolded player creativity.[57]

SimEarth also lacked thematic accessibility. Cities are familiar objects, but entire planets are not. Wright's next project, *SimAnt* (1991), was co-designed by both him and Justin McCormick. While its subject matter was familiar, it mostly fascinated and sold to the undesirably narrow demographic of 10 to 13 year olds (Rouse 2001). Maxis had zoomed all the way out to the entire earth, and zoomed way in, to insects, but the most familiar scale of all remained elusive until *The Sims.* There was an additional accessibility problem with *SimAnt* and *SimEarth.* Wright classifies his simulation works into the biological and economic. The economic ones, like *SimCity* and *The Sims*, offer interactions grounded in earning and buying, and have been far more legible and successful that those based upon squishy and hard to understand biological systems like *SimAnt* and *SimEarth* (Rouse 2001).

[57] For example, see Braun and Wright's comments in (Eisenhart 1996) and (Rouse 2001).

One way to think of the relative stability of *SimEarth* and *SimCity's* worlds is like the clutch power of Legos, the force which grips connected bricks together, striking an ideal balance between easy recombination and structural stability. *SimCity* was labile, reactive, and alive, and could be directed and nurtured into particular forms; *SimEarth* was less yielding, and vastly more dynamic, unpredictable, and autonomous. As a construction set, *SimCity* had an ideal Lego-like stickiness, while *SimEarth* was like trying to build a structure out of animate oatmeal. *SimCity* struck more of an ideal balance. Its parts remained more or less stable over time, and could be saved and loaded, allowing the player to design and shape stable cities (see chapter 9's discussion of manipulability for more on this point).

Figure 5.16: SimCity 2000. *Source: http://www.theregister.co.uk.*

Wright had never wanted to make another *SimCity* game, but after the lackluster commercial reception of *SimEarth*, *SimAnt*, and *SimLife* (1992), Braun implored Wright to do a sequel to *SimCity*, or Maxis would fold (Braun 2015).[58] Wright, in fact, had been working on the first prototypes for what would become *The Sims* for about six months before becoming "dragged into," as he put it, *SimCity's* sequel.[59] Fred Haslam, who had contributed to the simulation design and coding on *SimEarth*, had been working on the *SimCity* sequel, but wasn't mak-

[58]Braun: "SimLife was even the worst."

[59]In 1991, prior to working on *SimCity 2000*, Wright had begun a new project whose working title, at various times, was Dollhouse, SimWorld, Project X, and Jefferson—"life, liberty, and happiness" (Kelly 1994a; Eisenhart 1996; Wright 2016). *SimCity 2000* contains code from Dollhouse, echoing Wright's transition from the project (Rouse 2001). And *SimAnt* informed the artificial intelligence design of *The Sims,* whose characters made decisions in response to their environment—a direct inspiration from the pheromone model used for simulating ant intelligences (Rouse 2001; Wright 2016).

ing headway as fast as had been hoped. In collaboration with Haslam and other developers, Wright created *SimCity 2000*, the most celebrated incarnation of *SimCity*. The final product, *SimCity 2000,* was released in 1993 to massive critical acclaim and commercial success, underlining how the *SimCity* franchise underpinned the financial health of Maxis.[60] Braun fondly describes *SimCity 2000* as the "crowning achievement" of Maxis. The rest, he says, "was like a train wreck" (Braun 2015).

Unraveling

Wright's muse was separating from Maxis's business needs. SimBusiness had tried to align the sometimes divergent and sometimes congruent play of Wright and Maxis's audience, but these were coming apart. The delicate alliance that Braun and Wright had brokered was coming undone.

Financially, Maxis was a rising star. First year sales, in 1989, were $3 million, and grew quickly. By 1991, sales were $10.6 million. By 1992, sales had begun to level off, and Maxis brought on venture capital for a 30% stake in the company.[61] They hired a vice-president of sales, Sam Poole, away from Disney's software division.[62] With the 1993 release of *SimCity 2000*, sales shot up to $23.3 million (Greco 1994; "Maxis Inc. ready to go public" 1995). The new funding came along with professional management and a board of directors, who made preparations to take Maxis public.

[60] Wright and Haslam worked well together, and the project only took an incredible twelve months to complete, building upon some of the code Wright had written for what would become *The Sims* (Rouse 2001). Braun describes the collaboration between Haslam and Wright as "like magic, and that product came out unbelievable" (Braun 2015). If it wasn't for Haslam, says Braun, then *SimCity 2000* would never have happened (Braun 2016a).

[61] As Maxis took on funding, Wright's financial stakes shifted; he would own more of Maxis, but he would no longer receive royalties for *SimCity, SimEarth,* and *SimAnt* (Wright 2016).

[62] And their eccentric and brilliant lawyer, Bob Derber, came in house (Greco 1994).

In May of 1995, Maxis became a publicly traded company. This decision to go public "was just sort of the thing to do, and it just fell right," according to Braun. He identifies the decision to take Maxis public, made by himself and not Wright, as "a big mistake." In this, he was emulating software publishers like Brøderbund and The Learning Company. "It was me. Greed, you know, that was the path back then. I regret it now." The venture capitalists had advised Braun, the CEO, that it had been the right time to go public. Braun had agreed, even though he knew that the products they would need to make their numbers weren't there (Braun 2015; Wright 2016).

These transformations deeply affected the creative side of things. According to Wright, being a publicly traded company "drives a lot of dysfunctional decision-making" (Juba 2012), and "it was just obvious that we were driven towards short-term thinking" (Wright 2016). Braun describes Maxis as losing its focus. In order go public, they tried "to build up as much product as possible." Maxis established a studio in Monterey that built simulations for clients such as PG&E, Chevron (*SimRefinery*), and Hillary Clinton (*SimHealth*, 1994). Braun says that they did many ill-advised experiments, and instead should have "just stuck with what worked." He "had the golden goose and I'm looking for other golden gooses" (Kelly 1994b; Braun 2015).

One symptom of Maxis's lack of focus was an inability to get behind Wright's massively ambitious project to build a computerized dollhouse. Focusing on *The Sims* could have rescued Maxis's financial fortunes. Braun had always believed that it "would be the big one," and regrets not focusing upon it. But *The Sims* was a bold and experimental idea that required a huge research investment, and a massive amount of labor to produce the sounds, art, animation, and objects it needed. But most importantly, it was met with skepticism and ambivalence within Maxis, and never received the support it deserved.

According to Braun, skepticism and hostility to *The Sims* within Maxis came from everywhere, from developers to board members. Braun describes people's skepticism: "A doll house? That's for little girls." And little girls, of course, don't buy computer games. According to Braun, some found Wright's idea for a dollhouse simulator to be so absurd that they considered it an instance of Wright's prankish playfulness, like his "Willvis" Elvis impersonation, or a Magic 8 Ball jukebox he had built that answered any question in the form of an Elvis song (Braun 2015). Even Wright was skeptical that "a doll house for adults may not be very marketable" (Kelly 1994b).

Figure 5.17: Wright's box design for Home Tactics: The Experimental Domestic Simulator. Source: will-wright.com.

Having labored to create an environment in which Wright would thrive, Braun had inadvertently undone his work by taking Maxis public. According to Braun, "it was that conflict between the stress of this business and the investors that come in from that public process, and Will and his creative thing." The increased pressure of being a public company caused things at Maxis to go haywire. As Braun had predicted, they missed the first quarter numbers, and he got slammed. Braun lasted as CEO for about a year, after which Poole, who had become general manager and then president, took over. Poole wouldn't last long much longer, and began looking for a solution (Braun 2015).[63]

Maxis was going to crumble. They entertained the possibility of merging with some other mid-level publishers of their size, such as Bull Frog. Ultimately, they decided to sell the firm to a larger company, and held talks with Infogrames, Electronic Arts, and Activision. In 1997, a financially struggling Maxis was purchased by Electronic Arts (EA).[64] Braun left the company: "I couldn't stay at EA. I couldn't stay with that. It had to be its own thing. And that was a little hard, it's like a family." EA backed *The Sims*, if not with the fervor its eventual blockbuster success deserved, then enough for the project to come to fruition.[65] The hugely ambitious project could only have gestated in the permissive culture of Maxis, a place that was also, by

[63]In large part, Maxis worked on questionable products and ports in an attempt to project good (but dubious) numbers. Despite the fact that it was unprofitable, *SimCopter* was ported to a new platform in an attempt to show that more products were in the pipeline. These shenanigans quickly came to an end when EA took over (Foley 2015). Wright recalls *SimCopter*, *SimGolf*, *SimTunes*, and one other title (whose name he did not remember, but is likely *SimPark*) being part of a "Four Sims Initiative" to make good numbers (Wright 2016). *SimCopter,* according to Braun, had been rushed out, and simply wasn't ready for release—plus it was "sabotaged" by Jacques Servin, who would go on to co-found The Yes Men. Servin inserted an easter egg that forced Maxis to recall the product, and embroiled Maxis in a lawsuit. (They fired him, and he sued.) Servin, apparently, was protesting the lack of gay characters in video games, and inserted, unbeknownst to Maxis, an easter egg in which men in swimsuits would, on certain days, take over the city and kiss one another. This was especially hurtful given that Maxis offered same-sex partner benefits to employees, didn't engage in the sexist design practices that dominate the industry, and included homosexual relationships in *The Sims* (Masaki 2007; Braun 2015; Braun 2016a).

[64]Electronic Arts (EA) purchased Maxis for $125 million in the form of a stock swap. According to SEC filing documents, Braun owned 28.5% of Maxis (roughly twice the size of the next smallest holdings), and Wright owned 11.5% (Kennedy 1997; "Electronic Arts Will Buy Maxis in Swap" 1997). Over the next seven years EA's stock price would skyrocket from around $10 a share to a high of around $60.

[65]Braun claims the expansion pack architecture of *The Sims*, key to its incredible financial performance, was motivated in part because of Wright's lack of interest in working on sequels (Braun 2015). Wright agrees with this characterization, but claims that his underlying motivation was a desire to support the kinds of fan modifications then happening with titles such as *Quake*. Ultimately EA would simply milk the expansion packs as a business, and not encourage fan modification in the way Wright had hoped (Wright 2016). Of course, the motivations of Braun and Wright are congruent, and this story is an example of how a software design decision can satisfy the goals of multiple stakeholders, in this case satisfying both creative and business agendas.

then, too conservative and unfocused to back it to completion. *The Sims* captures the essence of Maxis's trajectory, oscillating between creative permissiveness and financial conservatism. Maxis could have completed *The Sims*, according to Braun, but lacked the necessary focus; the fiscal resources were there, but were squandered on other efforts (Braun 2015; Wright 2016).

Wright appreciated the renewed creative freedom he enjoyed at Electronic Arts. No longer at the center "of this small public company that was struggling to survive" (Juba 2012), he finally received the leeway to pursue *The Sims* in earnest. EA is not known for accommodating creative developers, and Braun recalls being asked for advice on what to do with Wright, to which he responded "Let him do whatever the hell he wants." The real question, Braun recalls saying, is "What does Will do with you?" Braun admires the executive team that took over Maxis, Luc Barthelet and Lucy Bradshaw, and their shepherding of *The Sims* to completion. *The Sims* became a blockbuster financial success despite EA, skeptical of its commercial viability, giving it a lackluster launch (Braun 2015).

The delicate fusion between art and technology, between playthings Wright made for himself—as opposed to others, finally became undone. The creative permissiveness of Maxis's had nurtured Wright's creation of unusual and fascinating toys that would have otherwise never come to be; titles such as *SimCity, SimEarth, SimAnt*, *SimLife*, and ultimately *The Sims*. Despite their uneven financial performance, they are masterfully crafted playthings, fashioned by expert simulation makers for their own playful ends. These playthings, are, in a sense, unusual toys the world received as gifts, byproducts of Maxis's curious fusion of self-directed play to commercial enterprise.

Future Cities

Open Sourcing SimCity

The next chapter looks closely at the code to *SimCity*, which is possible only because it has been open sourced. There are few instances in which a company has open sourced the code to a commercial game, which makes the story of how this happened remarkable for a number of reasons. Recounting this story not only explains the provenance of my research materials, but reveals how social forces, in this case a heterogeneous collection of agents and agendas, shape software.

The process by which *SimCity* was open sourced underlines its status as an evocative object. *SimCity's* polymorphism, its seemingly simultaneous playful, serious, educational, commercial, abstract, and naturalistic qualities are precisely what enabled it to thrive in so many different contexts. This bundle of appealing contradictions is what enabled different agents to see in *SimCity* what they wanted, a perceptual process which recalls Brian Sutton-Smith's *Ambiguity of Play,* and its thesis that play is susceptible to radically diverse essentializing rhetorics (Sutton-Smith 1997).

In 1991 the computer manufacturer Sun Microsystems, as part of an initiative to address the market for GIS computing (geographic information system), became interested in bringing *SimCity* into their platform. Sun's business was rooted in selling Unix workstations, and to that end, would often support the development of applications for specific market verticals. In the summer of 1991, Sun circulated a message to gauge interest in *SimCity*:

> The GIS and Desktop Applications Segments have a window of opportunity to get the #1 selling game "SimCity" ported to Sun. The opportunity also allows us to open up the software so that it can become a front end to GIS applications by including ToolTalk

> and/or import/export file capability to the standard GIS formats. This and networking will differentiate it from the PC version. The Sun SimCity version will allow Sun to offer a true "desktop" GIS solution as well as a leading entertainment package!
>
> The game is currently being used by many government offices to train their city planners. We have already received high support from ESRI [a developer of commercial GIS products] and URISA [a GIS education and advocacy non-profit] to move the game over and now are seeking Sun field encouragement that will concur with our direction.
>
> Please respond with any comments or customer interest that you think would support the project.[66]

The potential for a serious application, such as GIS, is overblown—it is hard to imagine *SimCity* enabling Sun to offer a "true 'desktop' GIS solution." The email, however, speaks to the appeal of *SimCity* as a compelling representation of a city. Not only was it serious enough for GIS, but it was fun, qualities which enabled it to become absorbed into the current of Sun's marketing agenda.

One person who responded enthusiastically to this proposition was Don Hopkins, a programmer at Sun who first played *SimCity* in 1989, while studying computer science at the University of Maryland, where it made a deep impression upon him. While a student, Hopkins had worked for the influential user interface researchers Ben Shneiderman and Marc Weiser, where he contributed to the development of pie menus (see Callahan et al. 1988), an unconventional graphical user interface mechanism that would eventually figure prominently in *The Sims*, on which Hopkins would be a key contributor.[67]

Hopkins began work on the *SimCity* to Unix conversion while working at Sun, but the project was eventually completed under the aegis of DUX Software. On July 19th 1991, DUX

[66] This message, an email sent to an internal Sun mailing list, was made available to me by Don Hopkins.

[67] Pie menus play a critical role in *The Sim's* user interface design, dovetailing perfectly with the object and AI architecture. Objects advertise verbs to character AI, so it is natural for the verbs to be arranged in a radial menu about objects. I can't imagine an alternate design that would have had the same widespread usability, and therefore appeal, without them. It is difficult to imagine *The Sims* without pie menus.

signed a ten year contract to convert and market SimCity on Unix, with Hopkins brought on as a contractor about six months later.[68] Using the Macintosh version of *SimCity*, Hopkins performed the conversion, rewrote the remaining sections of assembly language to platform neutral C, and changed the platform from Macintosh to X11. Throughout the 1990's, Hopkins did the programming necessary to convert *SimCity* to a wide variety of Unix platforms, adding pie menus and multiplayer support (Perkins 1993; Hopkins 2013; Joffe 2015).

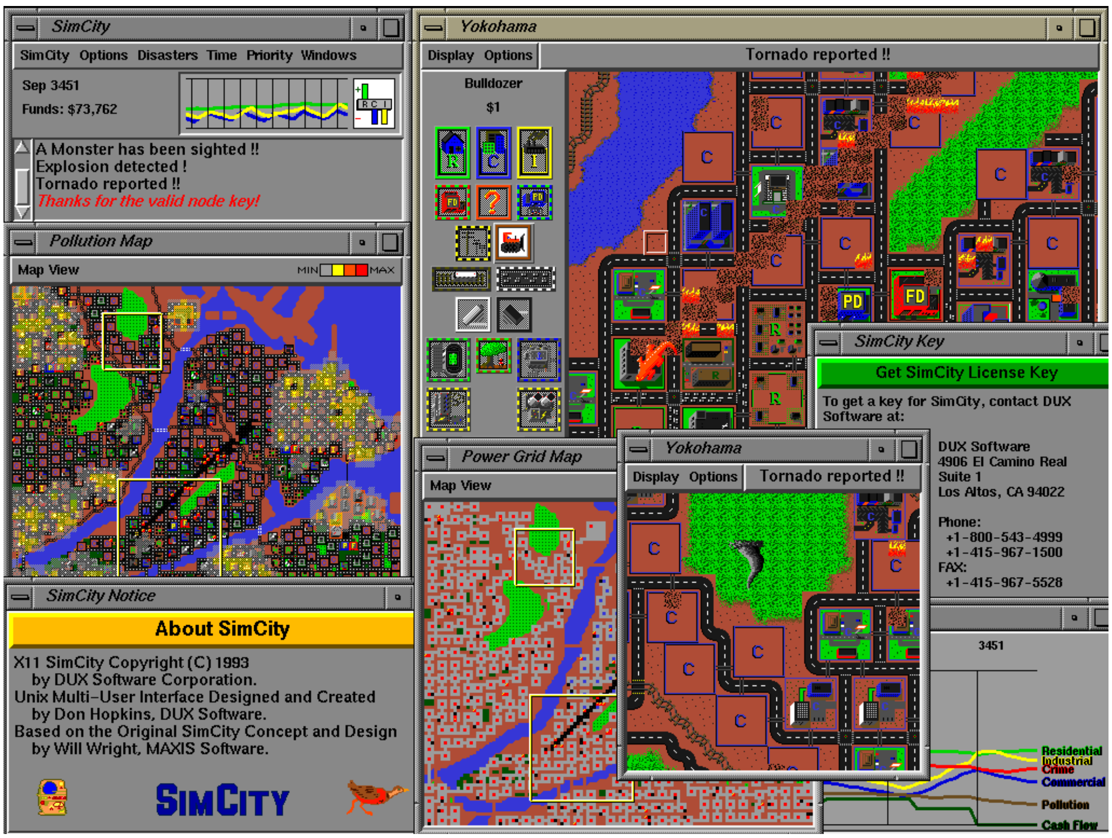


Figure 5.18: Screenshot of X11 SimCity *from* UnixWorld *review, via Hopkins's archive of (Perkins 1993). Retrieved from http://www.donhopkins.com/home/catalog/simcity/simcity-review.html, April 5, 2016.*

But it was the humanitarian One Laptop Per Child project that would provide the impetus

[68]Bruce Joffe, Wright's neighbor and friend with a background in city planning, GIS, and computer simulation, was almost brought onto the DUX project, via Braun and Wright, in order to advise on its conversion to a more serious simulation (Joffe 2015; Joffe 2016).

for open sourcing Hopkins's conversion. In 2005 Nicholas Negroponte presented the idea for an inexpensive laptop for children in developing countries that came to be known as OLPC (One Laptop Per Child). The effort traded upon the significant influence and prestige of Negroponte and others.[69] Reflecting both a desire to minimize cost and a particular ideological bent, the OLPC operating software was a Unix (Linux) variant with a GUI called *Sugar*—all of it open source and free software. This, in turn, set the stage for acquisition and development of open source, free, educational, and child-appropriate software that could run on such a low cost computer. *SimCity*, ever the evocative and fluidly perceptible object, was a perfect candidate, aside from one snag: it was propriety software owned by Electronic Arts, the company that had bought Maxis.

In November 2006 John Gilmore, one of the founders of the EFF (Electronic Frontier Foundation) and an early Sun Microsystems employee, asked Don Hopkins if he could ask Will Wright, whose prestige was at a zenith within Electronic Arts, about open sourcing or buying the original *SimCity*. Gilmore knew that Hopkins had completed the Unix port, was a close and long time associate of Wright's, and that the original 1989 *SimCity* would have presently been making very little, if any, money for EA. Hopkins, believing the request to be futile, nonetheless passed it along to Wright, who was enthusiastic about the idea, and introduced it to some other EA executives (Rod Humble and Scott Evans). Within EA, Chuck Norman was deputized for shepherding the process along (Hopkins 2013).[70]

Surprisingly, Electronic Arts agreed to the arrangement. Their legal counsel, in consultation

---

[69]This philanthropic agenda built upon decades of thinking about children, computers, and education, for example the work of Alan Kay and Seymour Papert, whose seminal work is specifically called out as part of the history of OLPC on its official web site (*http://www.laptop.org/en/vision/project/index.shtml*, accessed June 10, 2015). Trained as an architect, Negroponte founded the MIT Architecture Machine Group, later cofounding and directing the MIT Media Lab, both of which were hugely influential research sites.

[70]My account is largely based upon an interview I conducted with Hopkins as well as emails he has made available to me (Hopkins 2013).

with Eben Moglen (Columbia Law Professor, general counsel to the FSF, and OLPC advisor), worked through the legal logistics. This effort was aided by Hopkins's discovery and copying of the original Maxis/DUX licensing agreement, on a lark, while working on *The Sims*. Hopkins did the coding work of the conversion.[71] EA executives approved of the endeavor, no doubt aided by Will Wright's legendary persuasiveness and considerable prestige, not to mention the prestige of the OLPC project itself.

There were a couple of notable stipulations. First, that EA's quality assurance (QA) test and approve any release titled "SimCity." For this reason, Hopkins renamed the project "Micropolis," an earlier working title for *SimCity*. Second, as a post-9/11 consideration, EA requested that the airplane crash disaster be removed from the game. Regard for publicity, both negative and positive, must have driven these stipulations as well as the decision to move forward with the project.

SimCity began life as Wright's private toy, and went on to be continually recast as commercial software (Maxis), serious simulation (Sun), and educational plaything (OLPC). It's polymorphic quality allowed it to be perceived and used by diverse agents, who turned it towards satisfying their own particular agendas and contexts. This interpretive versatility, which can productively be understood through Sutton-Smith's notion of play's ambiguity, is crucial to *SimCity's* evocativeness, and ultimately led to its availability as free open source software.

[71] The Unix and then open sourced versions have undergone multiple iterations. The OLPC version is based on the original open sourced version, which is written in C for X11 with TCL/Tk. Hopkins introduced some optimizations and, I believe, some bugs in this version (the tile animation system). Hopkins would go on to rewrite Micropolis in C++, cleaning up and organizing the code. Java and JavaScript translations now exist, too.

Versions of SimCity

Before moving on to my close analysis of *SimCity* as software artifact, it is crucial to point out exactly *which* code I am looking at. What version of *SimCity*, exactly, am I studying? There are many different versions of *SimCity*.

First, I want to call attention to what I am calling the *canonical* version of *SimCity*, which is the original 1989 release for the Macintosh. While even this version went through various point releases, minor updates, and bug fixes, these Macintosh releases collectively stand as the original, canonical version of *SimCity*. (For this research, I used *SimCity* 1.4c for the Macintosh, running on Mac OS 9, within the *SheepShaver* emulator on OSX.)

Second, there is the original Commodore 64 *prototype* version of *SimCity.* This version, entitled *City Planner,* was originally written in assembly language for the C64 and had to be rewritten in C in order to be ported to the Macintosh and other platforms. This is the version Wright originally showed Braun, and differs in many significant ways from the canonical release. Since it was later released, virtually unchanged, for the C64, this prototype offers a glimpse into the early development of *SimCity*. Wright has referred to the prototype as the "first version" of *SimCity* (Wright 2011; Braun 2016b).

Third, the canonical Mac release served as the starting point for *conversions* to the Amiga, PC, and Unix. These conversions resemble one another very strongly, but often exhibit different user interfaces owing to the particular platform and developers doing the conversion. There are other more liberal conversions, for example, a 1991 version made by Nintendo for the Super Famicon that takes many creative liberties. Hopkins's various Unix variations, including the open sourced versions of *SimCity/Micropolis*, can be considered in this category of conversions.

Fourth, there is the *successor* to *SimCity, SimCity 2000*, considered by many to be the most refined version of *SimCity* (e.g. Braun 2015). This is one of many *sequels* that have been made.

Taken together, the prototype and successor versions of *SimCity* offer before and after snapshots of *SimCity's* design. The close analysis in the next chapter focuses on the canonical *SimCity* release.[72] When interpreting code, I look at the source code for the canonical Mac version, but also look to the PC conversion, and Hopkins's early C Tcl/TK version, for interpretive guidance.[73]

[72]When playing and taking screenshots, I use version 1.4c on the *SheepShaver* emulator.

[73]Don Hopkins has graciously made available to me the original Mac/PC source code that he received from Maxis in order to perform his porting work. It's not self-evident to me which released version this Mac code corresponds to, but given the comments, it appears to be from at least May 29, 1991.

Chapter 6

SimCity

Introduction

This chapter completes the *SimCity* case study. While the previous chapter focused upon the social forces that gave shape to it, this chapter attends to it as artifice and experience. Here we will see how the *SimCity* software transforms the universal computer into a miniature animate city, and how the resulting enactive representations scaffold play.

By deconstructing the artifice of these representations, we will again see the story about Wright's appropriations, but this time in the form of a technical tale that parallels the social and historical one. By paying attention to *SimCity's* code, we can see the building blocks that compose it. Traditions of simulation making, interface design (graphical user interfaces), software toys *(Pinball Construction Set),* and strategy and action games *(e.g. Raid on Bungling Bay)* offered strategies of representation and design—building blocks—that Wright repurposed to construct a software toy first for himself, and then for others.

Beginning with the packaging, we will see how *SimCity* signals and invites a playful orientation. The box is a discursive representation of *SimCity* that carefully establishes expectations, setting the stage for play. Next, we open this box and look at how *SimCity* affords interaction. Its graphical user interface enables players to lean in and see, feel, and touch a miniature city. By looking underneath the miniature city, we see the clockwork machinery that animates it. A complex software program, whose code I map and translate into diagrams, is the artifice that produces the illusion of a living world. I conclude by analyzing how *SimCity's* enactive representations, the building blocks of a miniature city construction set, scaffold play.

Packaging

Figure 6.1: The front and back of an early SimCity *box. (After Maxis was sued by Toho for infringing upon Godzilla, the monster was replaced with a tornado.) Despite the Commodore sticker, only Mac and Amiga screenshots are on the box, which are the first two platforms* SimCity *released on.*

The packaging initiates the framing of *SimCity* as a plaything. The box, along with the manual, advertising, and user interface are discursive practices that co-constitute *SimCity*. To solicit players and market *SimCity,* a sui generis object, it needed to be framed. Skepticism, for example from Brøderbund and their imagined consumer-players, needed to be overcome. To sell Wright's *City Planner*, Braun needed to translate it into a form that a broad audience would be receptive to. *SimCity's* discursive practices—the box, manual, marketing, and software—are a critical part of this translation. The packaging explains what *SimCity* is, kickstarting the scaffolding of player engagement.

Paradox signals play.[1] As was Braun's intent, *SimCity's* box, ripe with absurdity and paradox, arouses curiosity. Incongruities abound: the anachronistic bakelite material, the weirdly combined radio and television console. It is clearly some kind of a control panel, seemingly for a city, except we aren't to take this city seriously, as it's being menaced by Godzilla—one of many possible disasters we might inflict. The truncation in the title, *SimCity*, is explained at the box's bottom: "The City Simulator." *SimCity* is both city and simulation, reality and fantasy, belief and make believe.

A contradictory tone is struck. *SimCity* is silly and serious, full of whimsy as well as gravitas. In the program, players deal with funding levels and city budgets, in addition to giant monster attacks. *SimCity's* imaginative box design establishes this bipolarity. The bakelite control panel is an antiquated vision of the future, a retro-science fiction radio console. Or is it a television? Whatever it is, it looks both technical and fantastical. There are knobs, dials, and buttons for a city name ("San Francisco"), energy, budget, date, and population. An array of push buttons offer disasters, and on the screen a giant monster menaces the San Francisco skyline. The manual carries the contradictory tone forward. While it strikes an occasional

[1]See chapter 1 for information about paradox, play, and metacommunication (Bateson 1955).

note of whimsy, referring to the “simulated citizens” as “Sims,” it mostly affects a neutral and serious tone, offering “an essay on The History of Cities and City Planning, and a Bibliography for serious City Planners.”[2]

The box’s overall theme is one of power and control, of transforming a vast city into a manipulable plaything. A photographic San Francisco skyline is placed under the dominion of Godzilla. This image lives within the bakelite control panel, whose homeliness suggest that the topsy-turvy world contained within it is safely—to some degree—under the player’s dominion. Agency and aliveness are linked: the box promises to put traffic on the roads you make, and churn smokestacks you electrify. As the box suggests, the player is a god-like ruler of the city.

The box frames *SimCity* as an entertainment product. Despite the knobs, buttons, dials, and display that suggest power and control, this is clearly not a productivity tool, like a word processor or spreadsheet. Bakelite controls are anachronistic, promising not high-tech efficiency, but paradox. Godzilla belongs to the world of filmic entertainment, a harbinger of playful and risky transformation and destruction. This is not “City Planner,” a rational productivity tool. This is paradox, parody, and high jinks. In distinguishing *SimCity* from a productivity tool, autonomy is signified. One uses productivity tools to accomplish extrinsic goals, but entertainment products are used to satisfy intrinsic goals. Tools are for work, but *SimCity* is an instrument of fun.

There are many layers of discursive practices framing and constituting *SimCity.* Peripheral descriptions, such as the box, manual, and advertising establish expectations for how to interpret the software phenomena. Inside the box, *SimCity* elaborates upon the representational strategies of cellular automata and system dynamics, whose blinking colored squares

[2]The bibliography includes Jane Jacobs’s *The Death and Life of Great American Cities* (Jacobs 1961), one of the books that Joffe had recommended to Wright (Wright 2016). Michael Bremer, the manual’s author, notes that the historical essay was simply given to him for inclusion (Bremer 2015).

and numerical plots acquire representational force through *external* discursive practices that describe these as life, death, and unemployment. *SimCity,* on the other hand, employs many *internal* discursive practices. The running software describes itself via its user interface. *SimCity's* software *shows* the simulated world, while its secondary software descriptions *tell* you about it. Traffic, buildings, and monsters are shown with animated graphical artwork. *SimCity* also tells you about monster sightings and power outages through messages and maps that report on the world, narrating and interpreting it. All of these mechanisms scaffold make believe participation and play.[3]

SimCity exists within a discursively constructed play frame. Like a travel brochure that sets expectations and guides you to your destination, a news program that tells you what is happening, or a coach that guides your performance, *SimCity's* discursive practices enjoin you

---

[3]We know from chapter 2 that discursive practices co-constitute software and simulation, from the prose, diagrams, and graphs that manifested Forrester's simulated cities, and the stories told about computer simulated artificial life and intelligence, to the descriptors that transform Schelling's coins and Conway's tokens and shells into living beings and segregated cities.

In *SimCity,* many of these discursive practices take up residence in the user interface. Consider how *SimCity* describes its cellular automata with representational art. Cellular roads, traffic, buildings, and so on are visually depicted, whereas the interpretation of cellular automata is traditionally crafted through peripheral narratives. The black and white cells of Conway's *Life* are invested with the weight of "life" and "death" via language. The cellular automata literature is replete with such written narratives that transform abstract cellular states into vast catalogs of representations (e.g. Toffoli and Margolus 1987b; Berlekamp et al. 2004). *SimCity*, by contrast, directly undertakes some of the narration itself, via the user interface, delegating less to peripheral materials like journal articles, books, and packaging. Road, tree, and building cells are represented as such with naturalistic art; no peripheral gloss is needed to help us interpret abstract colored squares. By wedding a cellular automata to the C64 tile graphics system, a natural fit between two cellular lattices, Wright created a computational fabric whose fictional framing was conveyed without recourse to prose, as had historically been the case. While much of the flexibility and evocative abstractness of cellular automata is lost, *SimCity's* representations acquire force, specificity, and immediacy. *SimCity* describes and narrates itself through the user interface.

The user interface shows and tells. The world is *shown* via a diegetic representation of phenomena, and we are *told* about via descriptions in messages.

SimCity describes a primary phenomena, for example a city attacked by a monster, and then tells you, in a secondary narrative layer, about this phenomena, e.g. a message that a monster has been sighted. Sometimes this secondary non-diegetic description is the only evidence of a phenomena, as when players are told that citizens demand a stadium. Furthermore, *SimCity* is reflective, not only in its seemingly self-aware agents (e.g. helicopters seeking out and commenting upon high traffic), but also in its self-interpretation (e.g. power outages reported). *SimCity* interprets and comments upon your city through approval ratings, citizen ranked problems, and messages.

to engage *SimCity* as a plaything, enter its play frame, believe in its fictive world, and become a player.

Interface

Players lean in, see, touch, and manipulate a miniature city. The city manifests to the player in this tangible form through *SimCity's* user interface, which mediates between the clockwork machinery within, the city simulator itself, and the player. We can conceptualize *SimCity* as a conversation between player and simulator, an interaction that unfolds over numerous communication channels.[4]

Figure 6.2: Magnavox Odyssey (1972) television overlays. Source: Wikipedia.

Almost all modern computer games rely upon internal description and narration, a discursive practice constituted by the user interface. Historically, this was not the case. The Magnavox Odyssey, the first home video game console, designed by Ralph Baer and released in 1972, came with television overlays**.** The simple graphics shown on the television passed through these literal descriptive layers, which supplied a visual gloss on the simple glowing light patterns that emanated from below. These overlays call attention to discursive practices, and the gulf between abstract representation (as found in cellular automata, for instance) and figurative depiction. Once game consoles became more sophisticated, such peripheral descriptions migrated, for the most part, into the software itself. Some of these outermost layers, such as a physical box, cartridge, instructions, and advertising still persist, although some of these manifestations have disappeared or been transmuted into electronic forms, such as web sites, in the transition to digital distribution.

[4]In this framing, I follow Chris Crawford's conceptualization of interactivity as a conversation. The metaphor is limited, I think, but in this case it helps us to schematize and map out the interface (e.g. Crawford 2003).

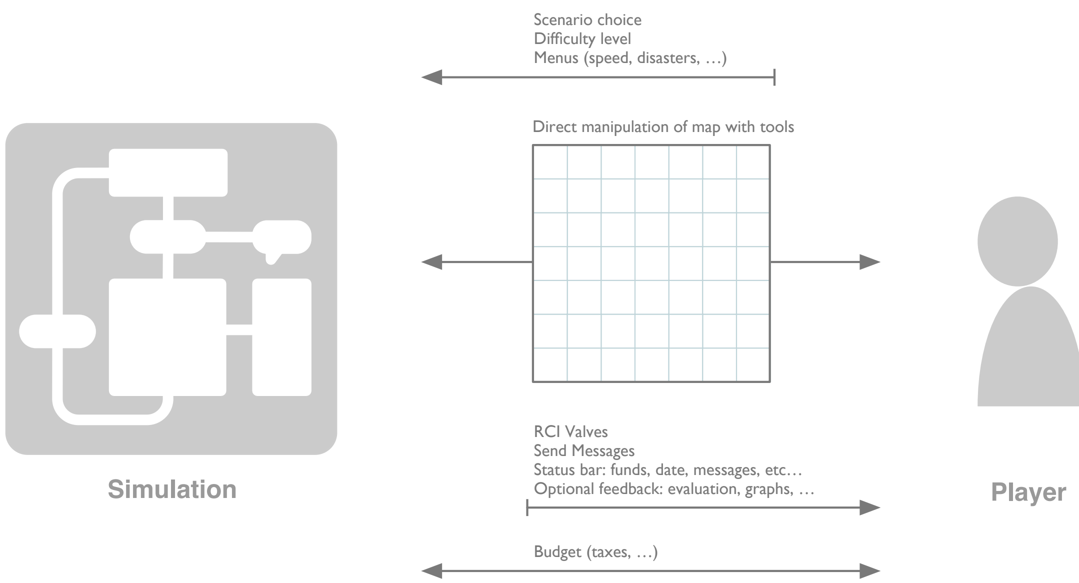


Figure 6.3: How the player convenes with the simulation.

As the diagram suggests, the map is the central representation. It is one of the few bidirectional communication channels, constituting the main shared grounding between player and simulator. The map is a tangible representation of a living city that the player directly manipulates. As far as the player and the simulator are concerned, this map is the city.[5]

[5]The map is a fairly direct representation of one of the system's fundamental data structures: the Primary Map. This map is the authoritative representation of the landscape—what is where: trees, buildings, roads, power lines, population, animation state, fires, flooding, debris, etc… Players visually see this map as a landscape composed of graphical character tiles, and manipulate it directly with the paint program style interface tools.

Furthermore, because of how development labor was distributed between the simulation backend and interface front end, the Primary Map abstraction and communication channel was also the critical point of contact between Will Wright and the programmers he collaborated with. (Rob Strobel, in the case of the Macintosh.) The Primary Map data structure is a critical piece of shared grounding between programmers, computational processes, and player.

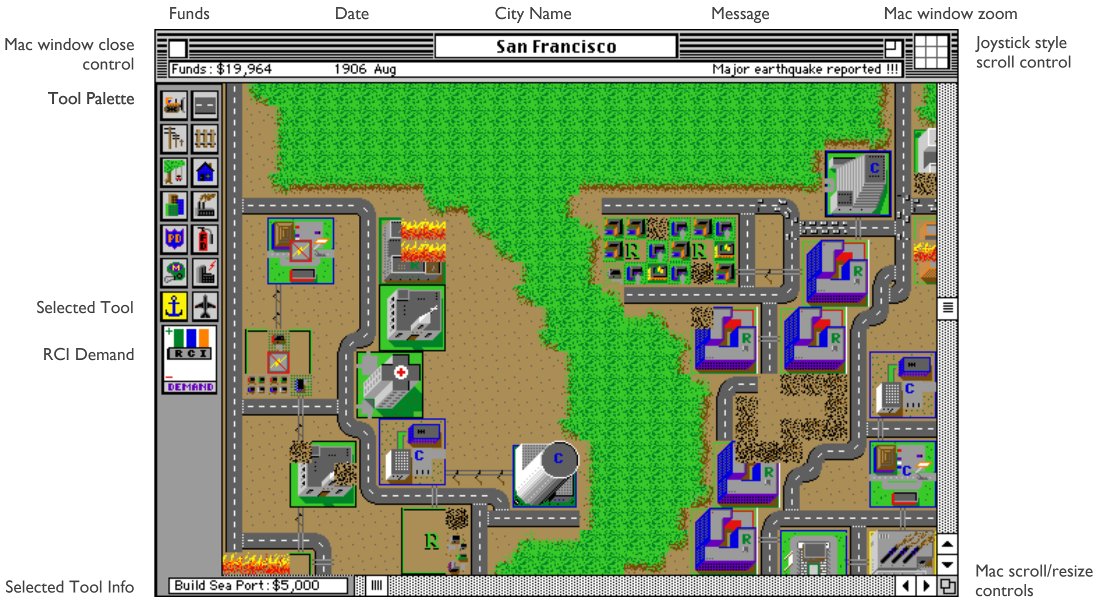


Figure 6.4: SimCity's *main window, which it calls the Edit Window. Native Mac GUI controls (scrolling display, scroll bars, window resize, window close) are intermixed with those inspired by Mac UI conventions (tool palette, a custom window title bar—expanded to make room for status information and joystick control), as well as creative additions such as the joystick style scroll control.*

This map representation dominates the Edit Window, which is the main focus of *SimCity's* user interface. This window allows players to pan around the city and manipulate it with paint program style tools. The map is, in effect, a giant canvas upon which players paint, zoning space and building infrastructure.[6]

[6]Using tools costs money, and a description of each tool, alongside its cost, is displayed in the Selected Tool Info box in the window's lower left corner.

Players only ever see part of the city at once in the Edit Window. A scrolling map display, similar to the one found in *Raid* and computer and video games more broadly, has been wedded to the Mac GUI's scrolling window and scrollbar conventions. A custom joystick style control, controllable by the player's mouse, has been added to the conventional GUI scrolling elements. These controls indicate where, relative to the overall map, the player is looking and offer a means for scrolling the display.

The layout and operation of the tool palette is clearly inspired by Bill Atkinson's *MacPaint* (screenshot in prior chapter). Pointing and clicking into the Edit Window's map employs the selected tool.

Figure 6.5: Detail of Edit Window's tool palette, annotated with their names and prices. A small menu attached to Build Power Plant allows the player to select a coal or nuclear plant.

While the map *shows* the city, other interface elements *tell* you about it in overview. The Edit Window's header displays the city name, available funds, date, and messages about the simulation.[7] Below the tool palette sits the RCI Demand indicators, which reveal the city's overall demand—positive and negative—for residential, commercial, and industrial buildings.

[7]It is an unconventional Macintosh window title bar, and has been expanded to include the quirky game inspired joystick-style scroller as well as informational feedback.

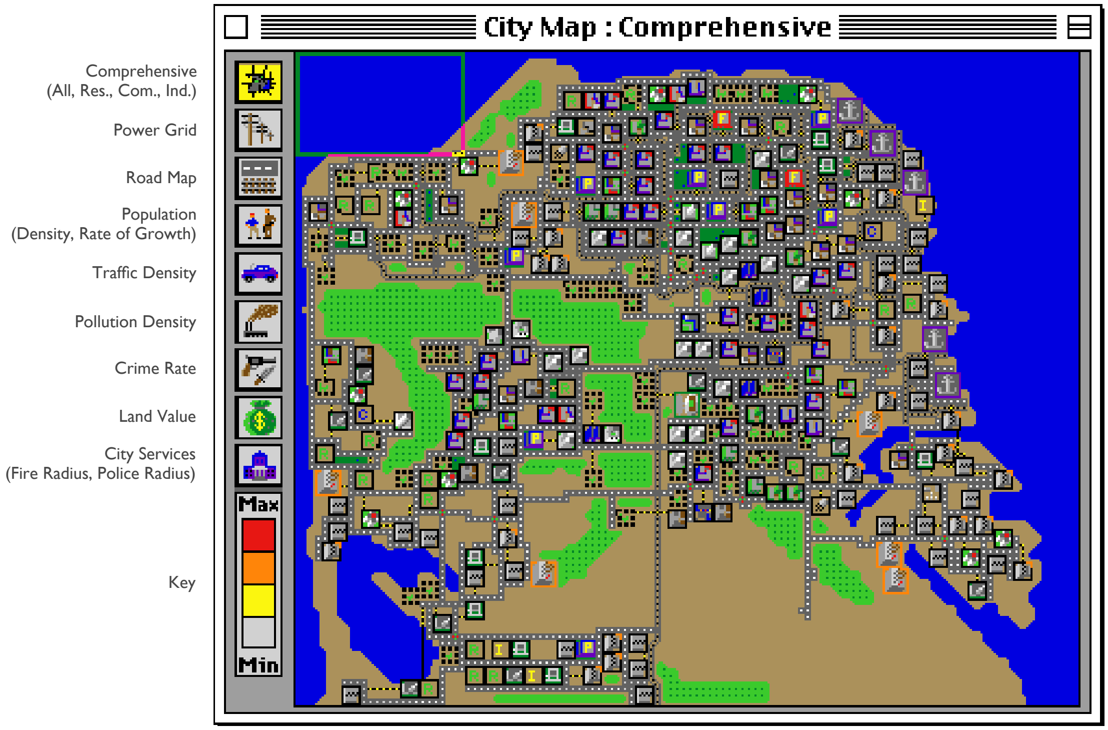


Figure 6.6: Maps Window.

Secondary data maps also tell about the city. In the Graphs and Evaluation windows, players can review historical data, statistics, and public opinion. Taken together, these secondary descriptions *about* the city are multifaceted and highly schematized overviews that complement the city map. While the main map shows a densely layered mimetic representation of the city, the secondary displays give abstracted overviews of it.[8]

[8]In these city maps the entire city is shown in extreme miniature. At the left side, a vertical stack of icons affords choosing a particular map to look at. Some icons encapsulate multiple related layers, exposed via a pop-up menu. A draggable rectangle indicates where in the overall map the Edit Window is focused. These maps offer relatively direct representations of many underlying data maps used to simulate the city.

The informational details of all these displays isn't important right now—we'll return to particular points as needed later on. For now, the main point to take away are the multifaceted overviews and descriptions offered of the city: across time and space, as graphs (time varying variables), maps, as a fictional city's name, available funds, and date, as messages, and in the form of public opinion and statistics.

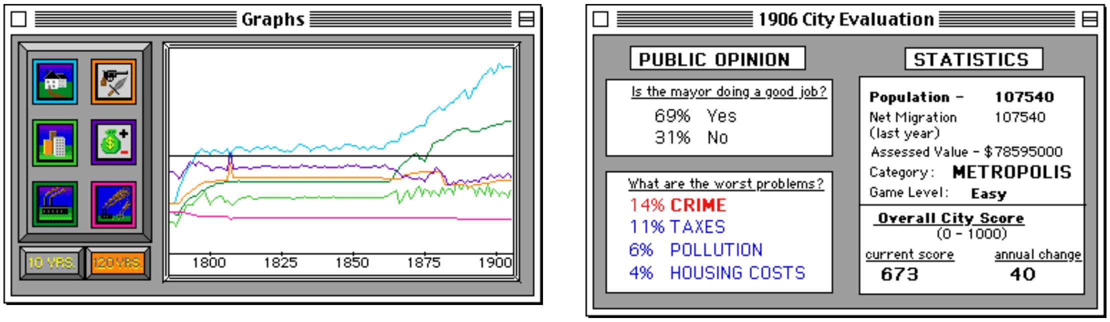


Figure 6.7: Left, Graphs window. Offers 10 or 120 year histories of six different variables: residential, commercial, and industrial populations, crime, cash flow, and pollution. Graph lines can be toggled on and off with the buttons. Right, Evaluation window.[9]

The Budget window is the only representation, besides the map in the Edit Window, that the player directly interacts with. It automatically pops up every year.[10] In it, players set the tax rate, allocate funds to city services, and review cash flow.

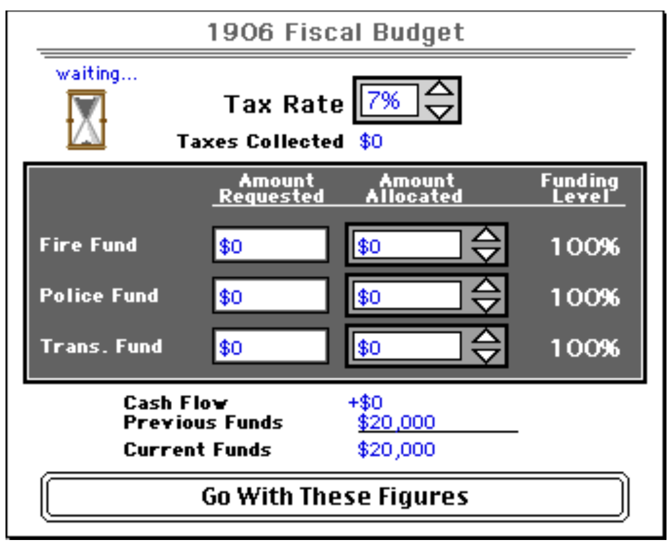


Figure 6.8: Budget window.

Play is configured through meta-level controls. Through menus, players can create, save, and load cities, adjust options, change the game speed, unleash or prevent disasters, or nav-

[9] Crime and pollution histories are smoothed, and residential populations are normalized (ResPop/8).

[10] Players can also view the Budget whenever they like via the Windows menu.

igate among the different windows.

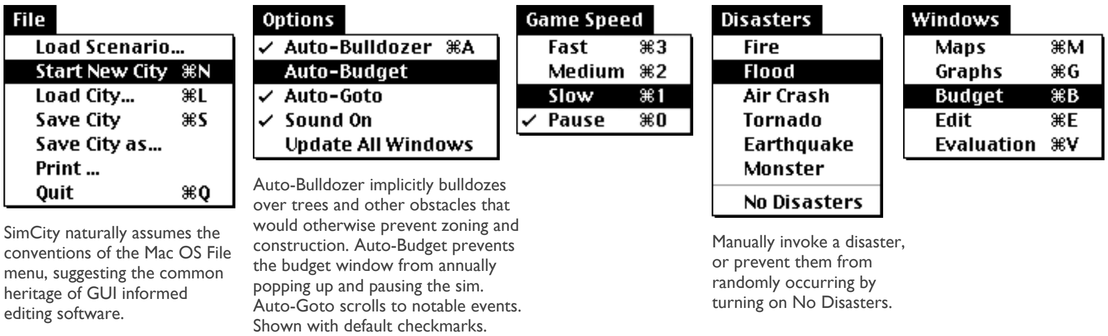


Figure 6.9: Composite image of SimCity’s *pull down menus. Meta-level operations on a city are afforded, as well as access to the different feedback windows.*

To begin play, a scenario must be established. Players either make a new city in a pristine landscape, load an old one, or play one of the built in scenarios.

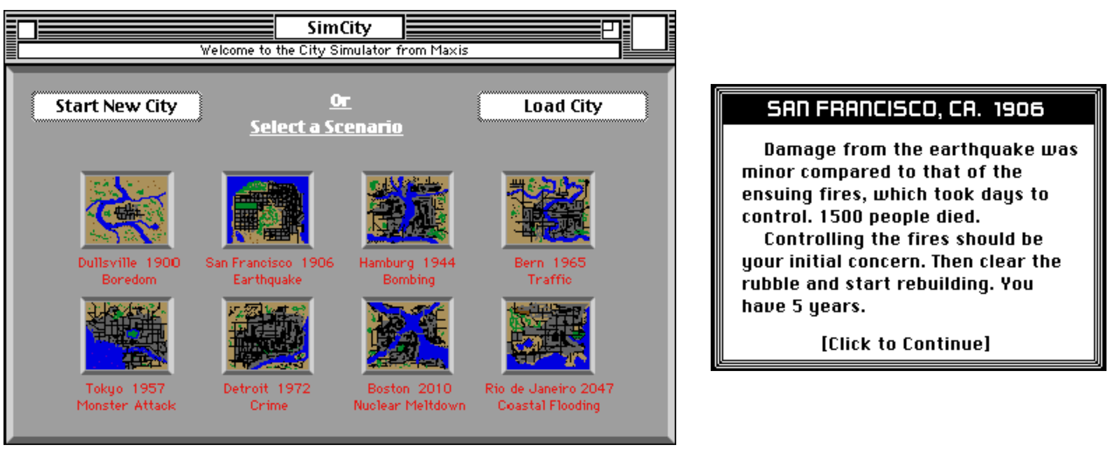


Figure 6.10: Left, the window presented at SimCity’s *launch. (I’ve skipped the splash screen). Right, an example scenario description.*

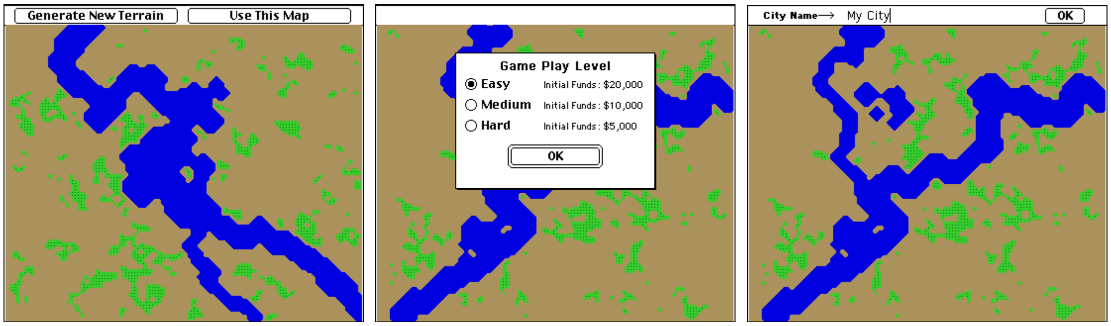


Figure 6.11: Start New City sequence. Players generate landscapes until they are satisfied (left), choose a game play level (middle), and then name their city (right).

These are the communication channels that link the player to the simulator. The interface puts enactive representations, such as a budget and editable city map, into the hands of players. These enactive representations are the products of software artifice, whose underlying mechanism we turn to next.

The Illusion of a Living City

One of *SimCity's* chief accomplishments is in bringing a stylized city to life, and doing so on rather simple computing hardware. We turn now to *SimCity's* simulation, to deconstruct its clockwork machinery, and see how it builds up a set of enactive representations that constitute a living city. These dynamic representations, a cellular city map, zones, buildings, an economy (demand valves, a budget), agents, and descriptions of the city, emerge from a carefully orchestrated hybrid of simulation subsystems.

The Simulation Assemblage

Systems

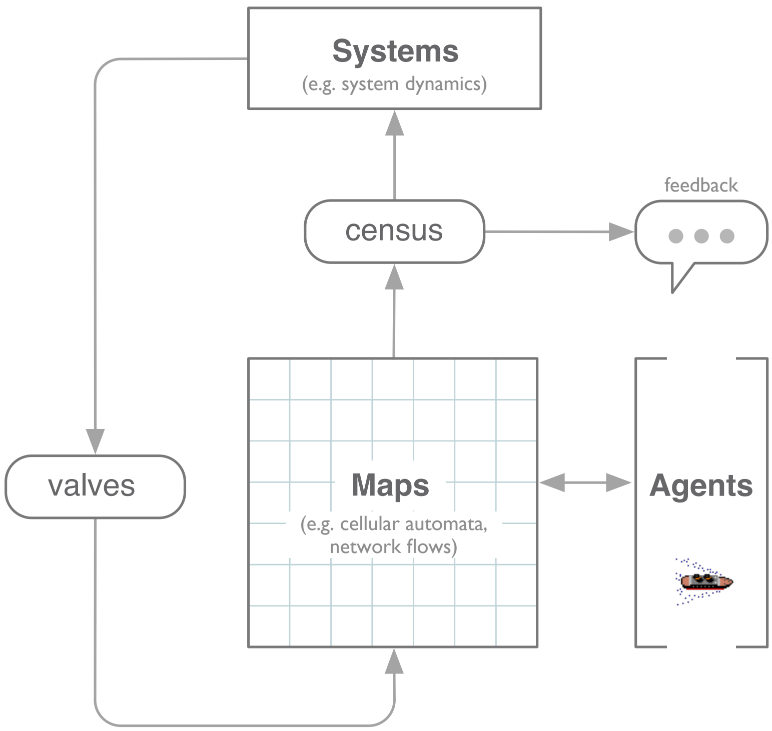


Figure 6.12: SimCity's *major simulation systems.*

The simulation is composed of a small number of major components. There is a spatial simulation in the cellular automaton tradition. An agent simulation models objects like a ship, plane, and helicopter that move around the map. The map is summarized into a set of census tallies that provide an overview of the map, enabling systemic processes to react to it. Foremost among these is the system dynamic model that simulates a citywide economy. This module sets valves that then govern growth and decay throughout the map. The census overviews also enable some other systems to react to the map: a messaging system which coaches play, agents (Godzilla emerges when pollution is very high), and evaluative feedback (approval rating).

Orchestration

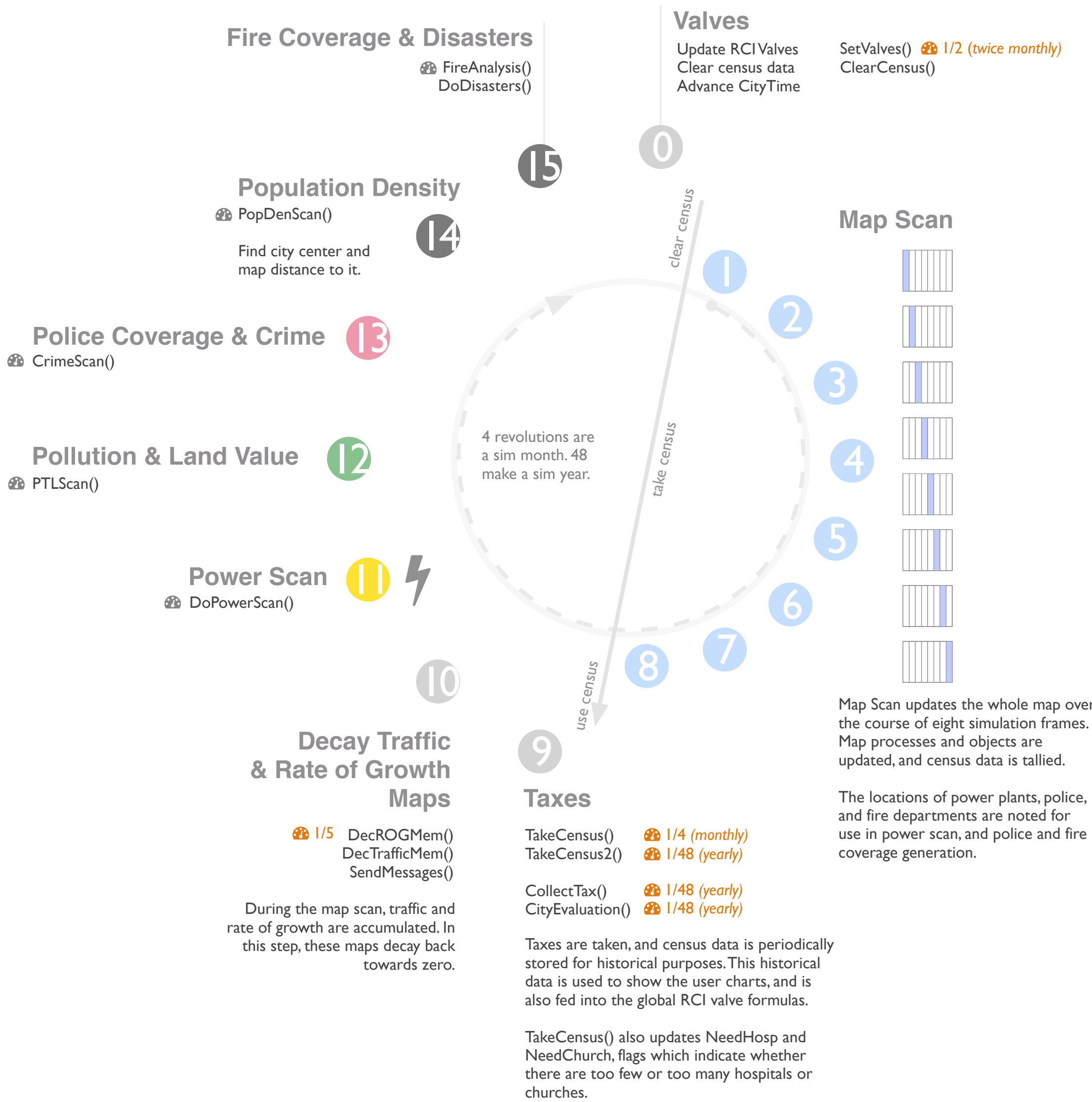
This ensemble is orchestrated by a multitasking scheduler that coordinates the performance of each simulation subsystem. This scheduler, called Simulate, allocates computer time to the simulation subsystems, and synchronizes communication between them. Although the computer can perform only one task at a time, swiftly interleaving these tasks yields the illusion of simultaneity, smoothness, and responsiveness.[11]

---

[11]The Simulate function coordinates and schedules virtually all simulation activity. As a schedule, Simulate is an excellent overview of the heterogeneous activities that sustain the city illusion. Recall Wright's comment that he had to "write a little multitasking operating system" to get everything to run on the meager C64 hardware. Simulate coordinates most of that multitasking, scheduling time for various simulation processes, ensuring that they happen in the proper order (e.g. taxation happens after a new census has been tallied), and allowing all this activity to be interleaved with other processes, such as the computations that keep the user interface updated and responsive to the player. As in a time-sharing multitasking OS, many activities are juggled.

City simulation is broken down into 16 steps. During every frame of computation, one of these 16 steps might be performed. Interleaved between these 16 steps *SimCity* performs activity that requires higher frequency updating, such as responding to user input, updating the various displays, simulating agents (e.g. tornadoes, monsters, boats; handled in MoveObjects), and tile animation (e.g. traffic animation and smoke billowing from stacks; handled in animateTiles). Taken together, all of these interleaved processes produce the illusion of simultaneous activity as well as reactivity. Each revolution advances the city time by 1, and is equivalent to 1/4 of a simulated month. 48 revolutions are equivalent to one year of simulated city time.

There are many reasons to distribute work over time like this. I've already mentioned the need to keep things responsive to the player. Breaking down work into chunks like this means that the simulator isn't trying to do too much all at once. Some of these processes could take a long time (at the scale of interactive computing), so interleaving means the player won't perceive any hiccups or pauses while playing. Furthermore, the overlapping rhythms generated by this scheduling has important aesthetic qualities I take up later, when I discuss the illusion of aliveness.



	Slow	Medium	Fast
DoPowerScan()	1/2	1/4	1/5
PTLScan()	1/2	1/7	1/17
CrimeScan()	1	1/8	1/18
PopDenScan()	1	1/9	1/19
FireAnalysis()	1	1/10	1/20

To keep up with faster simulation speeds (selected by the player in the Speed menu), certain processes run less often.

Some steps always run at a fixed, lower frequency.

Figure 6.13: Simulate(). Diagram of main simulation loop scheduling logic.

Simulation is broken down into 16 steps.[12] Between them, real time tasks that require high

[12]More detail on what happens in each of these 16 steps:

0. *Valves*. Update citywide demand for residential, commercial, and industrial buildings using a model inspired by Forrester's *Urban Dynamics*. These values are expressed to the player in the RCI Demand

frequency updating—the user interface, map animation, and agent system—have a chance to update.[13] As you can see in the figure, a broad range of processes are updated. And if you look closely, you can see that information flows cyclically, between them.[14]

display in the Edit Window. Internally, these variables are referred to as valves, nomenclature which betrays the debt to Forrester's system dynamics. This step consumes and then clears the census data that will be accumulated during Map Scan.

1–8. *Map Scan*. Scan and update the entire cellular map. It takes 8 steps to update the entire map, and each step covers 1/8th of it. A side effect of Map Scan is a census tally about what is in the map (and sometimes where).

9. *Taxes*. Collect tax revenue; store census data for historical graphs (shown in Graphs Window); update city evaluation (shown in Evaluation Window); update system dynamic style hospital and church population regulator.
10. *Decay Traffic and Rate of Growth Maps*. Map Scan adds traffic and rate of growth activity to their respective maps. This step decays those maps back towards zero. Also, messages might be sent to the player about the city's overall condition, appearing in the Edit Window's Message area.
11. *Power Scan*. Power is distributed throughout the grid. This uses the location and capacity of power plants gathered by Map Scan.
12. *Pollution and Land Value*. Update Pollution and Land Value maps.
13. *Police Coverage and Crime*. Police Radius and Crime maps are updated. This uses the locations and efficacy of police stations noted by Map Scan.
14. *Population Density.* Update Population Density map; find distance to city center, and precompute distance to city center in Commercial Rate map.
15. *Fire Coverage & Disasters*. Using the location and efficacy of fire stations recorded during Map Scan, the Fire Radius and Crime maps are updated. Spawn disasters according to scenario and settings of Disasters menu.

[13]Earlier revisions of *SimCity* relied on an interrupt based mechanism to interleave updating of the simulation and user interface, constituting a kind of primitive preemptive multitasking scheme. This later had to be improved upon because it wasn't portable and didn't work well with the Mac OS Multifinder. (Also, the Amiga natively supported such concurrent processing.) Remnants of Wright's preemptive multitasking scheme, and possibly support for Amiga's native concurrency, can be found in the form of archaic mutex guards, synchronization flags that seem to serve no purpose, and tests within the simulator for input events that cause simulation to stop (Foley 2015; Conrad 2016; Wright 2016).

[14]Two high level qualities of Simulate are worth attending to. First, the scope of processes that comprise the bulk of city simulation, from updating the power map to sending the player messages about the city. Second, the cyclical flow of information between cellular and systemic processes. For example, Map Scan sweeps the city map, tallying up information that will be consumed and used in systemic processes such as tax collection and updating the RCI valves. In turn, this systemic information (e.g. the RCI valves) is consumed in Map Scan's updating of map cells. The 16 steps of Simulate not only facilitate the interleaving of parallel activity, as in a multitasking operating system, but orchestrate the flow of information across systems and scales.

Hybridization

One of the most crucial cyclical information flows is between *SimCity's* spatial and systemic simulation processes. By coupling cellular automata and system dynamics style simulations, behavior unfolds at both micro and macro scales, giving the microworld both finely detailed texture as well as an overarching organic coherence.

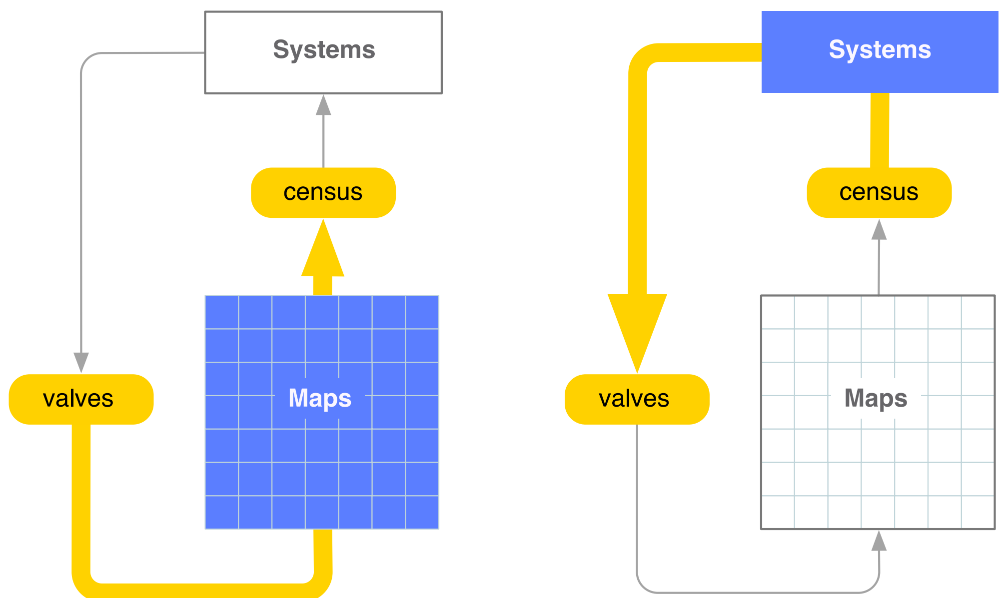


Figure 6.14: Coupling of systemic and spatial systems.

These two systems are coupled in a cyclical feedback loop. To talk to one another, these different simulation traditions—founded on different principles and schematic abstractions of the world, one spatial and one aggregate—need intermediate representations. This is accomplished via census data, which summarizes the map as population counts, and valves, which allow the map to be guided by systemic processes. The map tallies census counts for the system dynamics model, which uses them to compute citywide demand for residential, commercial, and industrial zones. These demands, or valves, go on to regulate the cellular automata. As a zone is scanned, the probability of growth and decay is modulated by local spatial circumstances as well as citywide demand.[15] And so, these different simulation styles are unified into a hybrid model. Like dance partners, local and global dynamics revolve about one another, taking turns leading and following.

Cellular Automata: The Urban Tapestry

SimCity offers the enactive representation of a living, tangible, spatial world. Like *Raid on Bungeling Bay*, the world is a sprawling scrolling landscape. Unlike *Raid,* the landscape is far more dynamic. Fires burn and spread, multicellular buildings grow and decay, and transportation and electrical networks interconnect the city. The landscape is animated by spatial processes that do everything from spread pollution to power, but all of these diverse enactive representations are organized around a single authoritative representation of the city: the Primary Map.[16]

Primary Map: A Supple Living World

When peering through the Edit Window one perceives a complex urban landscape. It doesn't take long to see that this landscape possesses a cellular granularity. Each cell looks like a chunk of road, part of a building, or unit of water. This mosaic, which measures 120 tiles wide and 100 tiles tall, depicts a city. The mosaic's combinatorics gives *SimCity's* landscape its suppleness. Players of *SimCity* think of the city as a mosaic, aware of both individual tiles and

[15]This is the implementation of one of the ideas Wright took from Walter Christaller's *central place theory.* Another idea is distance to city center. According to Wright, internally focused commercial development thrives near the city center, while outwardly focused industrial production thrives at the outskirts. From my reading of the code, only the commercial zones receive a slight bonus from being near the city center; residential and industrial ignore this proximity. When I shared my reading with Wright, he said it must be a bug (Wright 2016).

[16]This *Primary Map* terminology is my own. The code simply refers to this map as Map. I refer to this map as the Primary Map not just to disambiguate it from the many secondary maps, which we will come to momentarily, but to emphasize its authoritative nature.

the granular whole they compose.

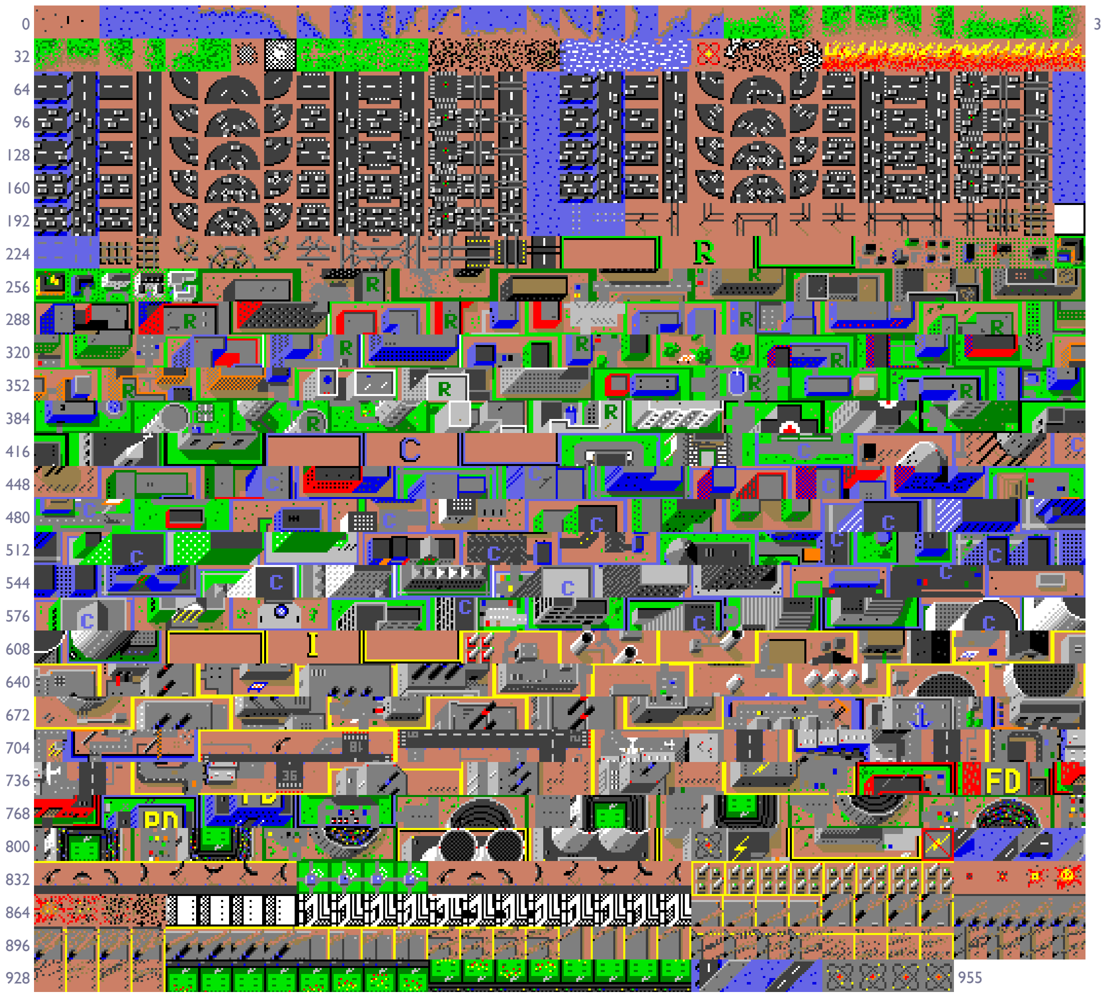


Figure 6.15: The 956 characters which are assembled into cities.[17]

The world players see is constituted by a mosaic of 956 possible characters. When combined, these characters form roads, power lines, fires, bodies of water, and buildings. Changing them over time produces animations.[18]

[17]This character set comes from the OLPC port of *SimCity*. Some characters are unused (e.g. black and white characters), and a subset of those contain variations introduced by Don Hopkins (e.g. an additional spinning radar dish.) As is visually apparent, their ordering is far from arbitrary, enabling semantically meaningful operations to be performed via arithmetic operations on tile indices. For a full accounting of the characters and their meaning see the appendix.

[18]*SimCity's* internal map representation can be traced back to the C64 character graphics hardware, which

The authoritative representation of a city is encoded in the Primary Map, which stores which characters are where. But the Primary Map is more than simply a visual representation—it can be thought of as storing the state of a cellular automaton.

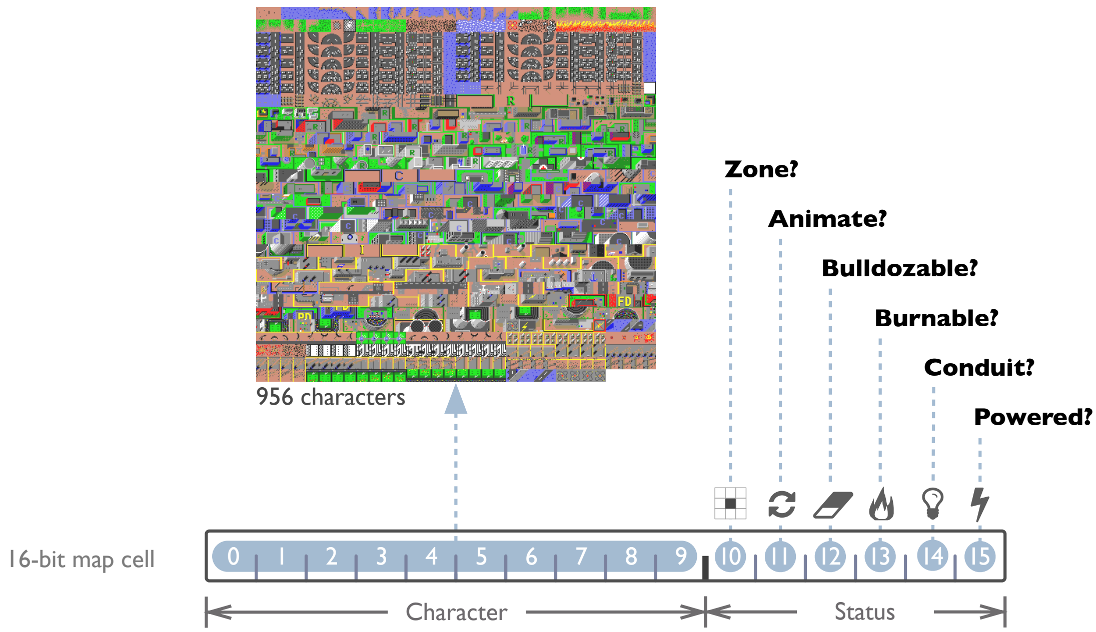


Figure 6.16: Anatomy of a Primary Map cell.

The character occupies the low 10 bits of a 16-bit integer, and is easily translated into the graphical representation shown in the Edit Window. (If the cell contains a 0 or 1, then dirt character graphics are shown; if the cell contains a 2, then the water graphic is shown; etc…) The leftover high 6 bits—referred to as status bits—are used to encode functional information such as whether the tile is the center of a zone, is animating, can be bulldozed, is burnable, conducts electricity, and has electrical power. These 6 status bits can be though of

can generate scrolling landscapes out of program specific graphical "characters." This hardware enabled large play fields to be displayed, updated, and scrolled with an efficiency (and hardware cost) that would have been otherwise impossible at the time (e.g. with bitmap graphics). Hardware affordances of the C64 graphics platform, such as the aforementioned graphical character map as well as its sprite system, are reproduced and extended by *SimCity* in software. Mac and PC/DOS did not offer these graphical platform affordances (tile maps and sprites), but were beefier than the C64, and so enabled these abstractions to move to software. The elaboration of these abstractions reflects *SimCity's* deep commitment to the structural patterns afforded by the C64.

as secondary map data layers.

The Primary Map is updated through a simulation process called Map Scan. As Map Scan sweeps over the map, it examines each cell, and decides what to do based upon its contents.[19] The resulting transformations imbue the map with aliveness. To understand everything Map Scan does, let's zoom in and look at how it handles one type of character: fire.

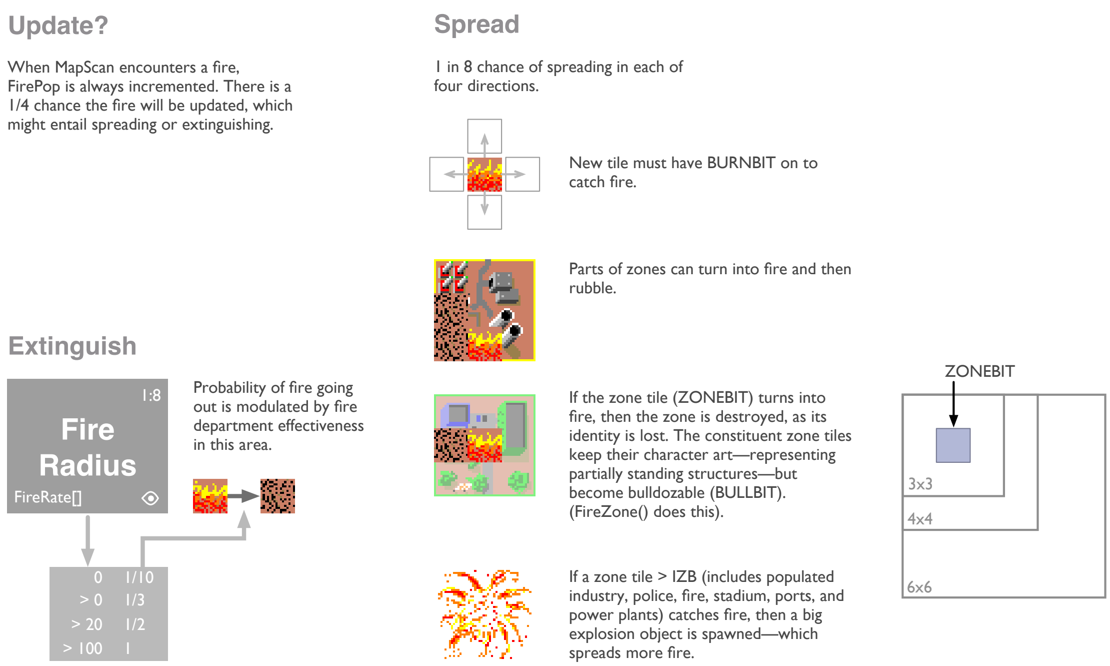


Figure 6.17: DoFire()

When Map Scan encounters a fire character, it first increments the fire census, and then it updates the fire. But fires don't always do something; there is only a 25% chance the fire

[19]Simulate allocates 8 simulation steps to Map Scan, which it then uses to sweep over and update 1/8th of the Primary Map. This cellular update is very much in the spirit of cellular automata, even if it does not adhere to a strict definition of cellular automata. There are many differences between a pure cellular automata simulation and *SimCity*, and this depends on whose definition of a cellular automaton you are using. According to Burks's rather free spirited definition, *SimCity* would count as one (Burks 1970). One key difference is that *SimCity* does not use double buffers. Two buffers are typically employed to ensure that an unfolding world state deterministically follows only from the state of the entire world on the prior time step. *SimCity's* heavy use of (pseudo-)random numbers eliminates a key problem that can arise without double buffers, which is cell update ordering strongly biasing outcomes. The element of chance ensures that certain phenomena, such as the appearance of hospitals and churches, are distributed evenly across the map instead of clustered together according to the order in which suitable development sites are found and updated.

will be updated. If it does, the fire might spread to a random adjacent cell or extinguish. The probability of going out is based upon the fire coverage map. Although fire updating is relatively simple, it illustrates the basic kinds of things that happen during Map Scan: tallying population, probabilistic updates and transformations, reading from data maps, and modifying the Primary Map.[20]

---

[20]Map Scan first increments the fire tally, so that when Map Scan is complete FirePop (fire population) contains a reasonably accurate census of the number of active fires. After tallying the fire, there is a 1 in 4 chance that the fire will actually update. If the fire is to be updated, then DoFire() is called, which does the actual work of extinguishing and spreading fires. DoFire() first tries to spread fire to each of the four adjacent cells (1/8 chance for each direction), and then uses the Fire Radius map to try and extinguish the fire. Stronger fire coverage means a higher probability the fire goes out and is replaced with a rubble character.

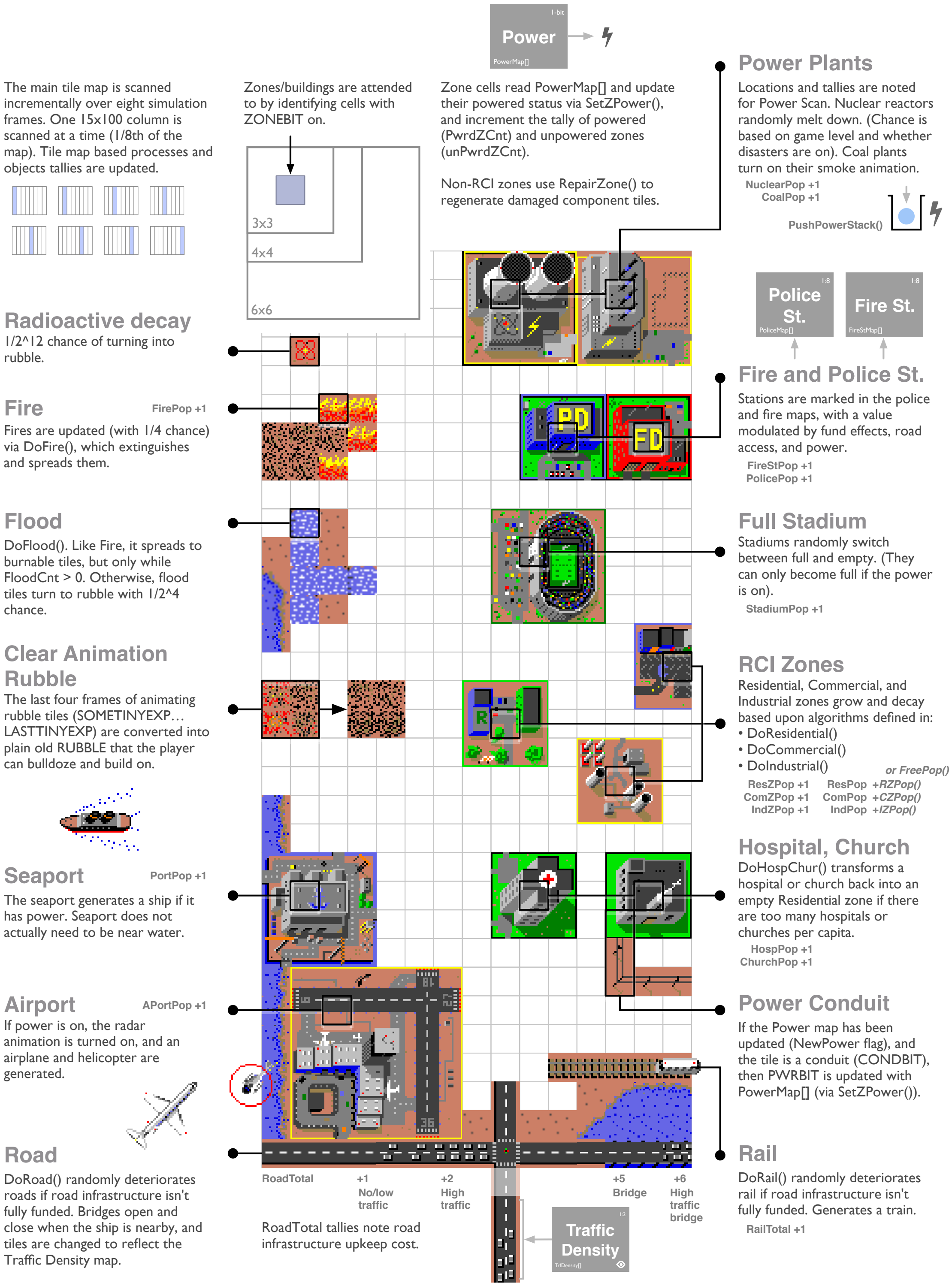


Figure 6.18: Map Scan.

Now, let's zoom out and get an overview of everything Map Scan updates. As you can see, many different entities embedded in the map are updated and tallied. Some of these entities are multicellular buildings, like the airport and power plants. Map Scan apprehends these by attending to a special cell marked with the Zone Bit.[21]

Almost everything, from buildings and roads to fires and floods, is tallied up for the census. These population tallies, as we have already seen, are fed into *SimCity's* system simulation processes, bringing its spatial processes in conversation with its overarching systemic ones.[22] Sometimes a simple tally isn't enough, and spatial information is needed. The locations of fire departments, police departments, and power plants are noted because other simulation subsystems—such as the fire coverage map generator—care about not just how many, but where these things are.[23] Map Scan, as you can see, sits as the nexus of many simulation processes.[24]

[21]While a cluster of fire cells or a transit network perceptually appear to be multicellular, Map Scan treats these as single cells, leaving our perception, for the most part, to grasp them as multicellular wholes. Buildings and zones, however, constitute multicellular wholes tracked in the Primary Map via the Zone Bit. Map Scan attends to these multicellular structures by identifying cells with the Zone Bit set to true. The Zone Bit does the bookkeeping for these larger objects—they are simultaneously cellular, composed of atomic cells, and have an identity as whole buildings. If the zone cell is destroyed then the structure has lost its identity as an integral whole, even if some of its cells persist. An example of this can be seen in the DoFire() figure, where a zone has been partially destroyed by a fire.

[22]Why does Map Scan so studiously tally the objects dispersed across the Primary Map? These census totals bridge the gulf between the diffuse cellular automaton style simulation and systemic city wide processes. Census data is what enables systemic city wide processes to respond to local circumstances, bringing the systemic and the cellular into conversation. City wide processes include everything from sending messages to the player about things like power outages, evaluating player performance in scenarios, and the system dynamic mechanism that modulates global RCI growth. Zone development, in turn, responds to both local and global (RCI valves) circumstances. Census data is the crucial link in the feedback loop that connects the local simulation, which is spatial and cellular, to the global simulation, which is systemic and indifferent to space.

[23]Fire and police department locations are stored as numbers in a map (scaled for efficacy), since a further cellular operation (smoothing) will be used to generate the coverage maps. Power is different, as a network algorithm must be used to determine the flow and reach of electricity. Therefore, the locations of power plants are noted in a stack (the Power Stack).

[24]As you can see in the figure, many maps are read from and written to, objects spawned, and global variables are read. Traffic Density map data is used to choose the appropriate traffic characters for roads; Fire Radius map effects the rate at which fires are extinguished; Power map is used to check if a cell is powered; rail spawns trains, airports spawn planes and helicopters, and seaports generate ships; road and rail decay based upon

Secondary Maps: Depth

Cities are composed of a Primary Map and many secondary maps. These other maps are like scratch paper, and are used to calculate and store things like land value, the location of police stations and fire departments, pollution levels, fire coverage, and traffic density. Encoding the city in many different map layers allows different spatial processes to be elegantly combined.

---

infrastructure spending, fire and police station efficacy is scaled according to funding levels.

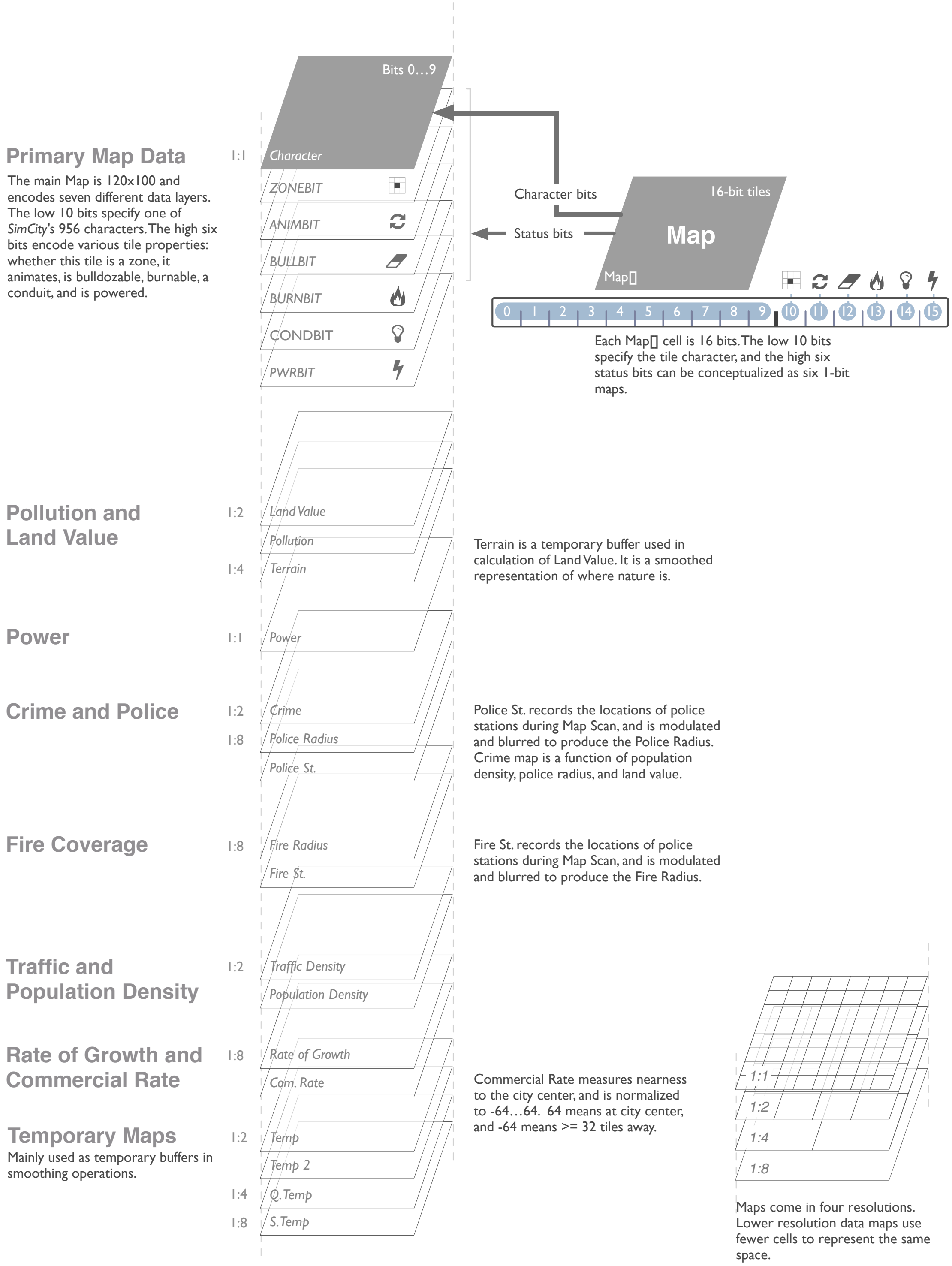
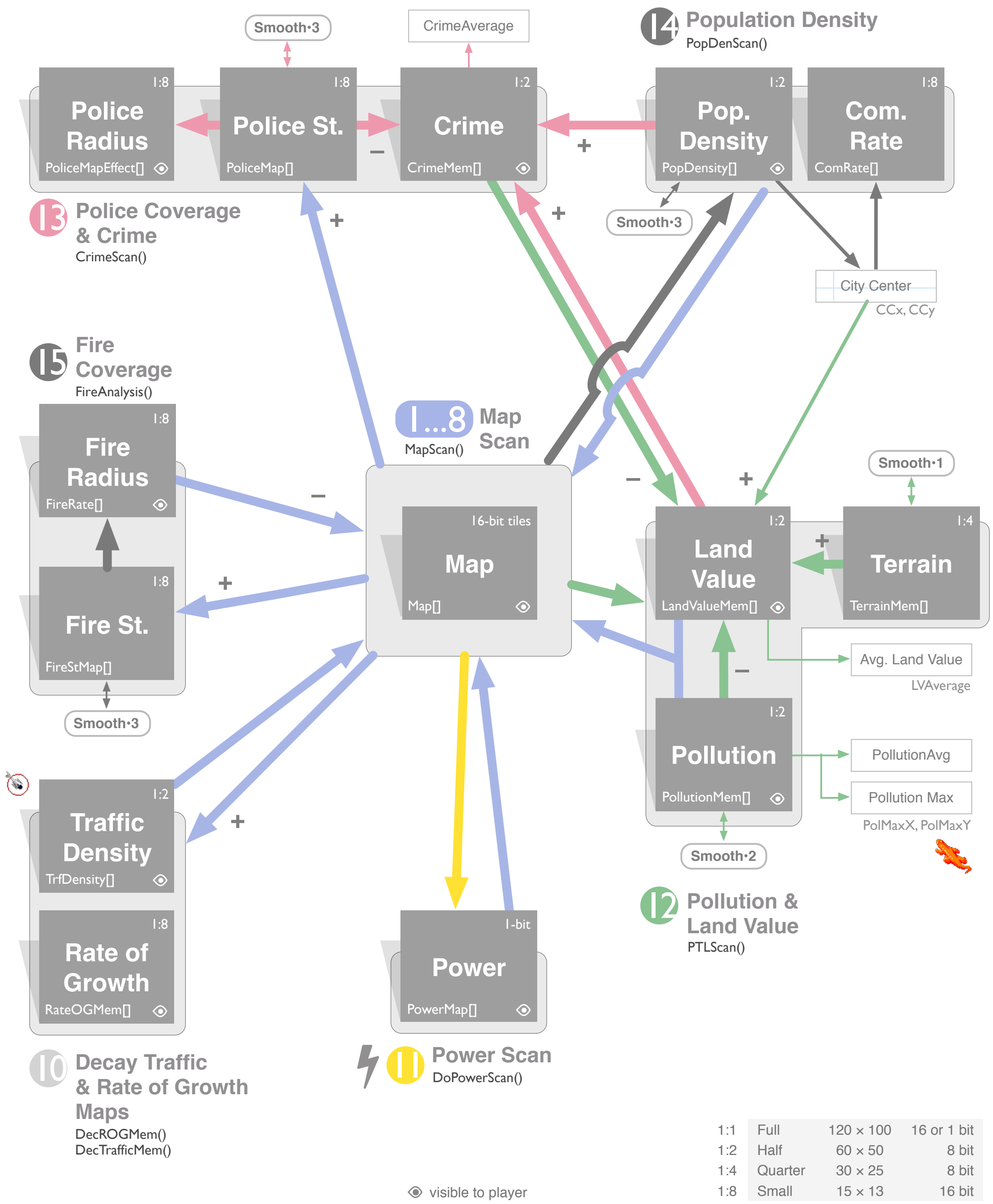


Figure 6.19: Map layers. SimCity's *spatial data is modeled in multiple maps that can be thought of as overlaid upon one another.*

Secondary maps facilitate communication. These scratch workspaces are where simulation subsystems carry out their computations, represent the result of their work, and convey information to one another. Many secondary maps are visible to the player, who can see them in the City Maps Window, giving a glimpse into the internal machinations of their simulated cities.

We can chart the flow of data between map layers. By overlaying information about which simulation process moves and transforms the data, and on which beat of the master schedule, we can see how space, time, and simulation subsystems interrelate. Visualizing time and space as correlated suggests that layers are chambers through which information courses, like the pumping of blood through a heart.



1:1	Full	120 × 100	16 or 1 bit
1:2	Half	60 × 50	8 bit
1:4	Quarter	30 × 25	8 bit
1:8	Small	15 × 13	16 bit

Figure 6.20: How data flows between map layers and simulation processes. Arrow colors and numbers correlate processing with the steps of Simulate. Positive and negative feedback relations are indicated when straightforward. (Some relations are more complex.)

This data flow diagram contains many feedback loops. Let's consider the example of fire coverage. Map Scan, as we already saw, records fire department locations. When it is time for the fire coverage to be computed, these locations radiate a coverage effect that is stored in the Fire Radius map. Then, when Map Scan runs again and wants to know the probability of a fire extinguishing, it consults this map.[25] Coupling multiple processes and maps gives richness and depth to the simulation while maintaining modularity.[26]

While there are many feedback loops in this diagram, almost all of them flow through the Primary Map, underscoring its central and authoritative role. When you save a city, it is the

---

[25]A more technical explanation of the fire feedback loop follows. Map Scan (steps 1-8) notes the locations and efficacy of fire stations, storing these as integers in FireStMap[]. (Fire station efficacy is modulated by whether each station has road access, electrical power, and citywide fire department funding levels). Step 15 computes fire department coverage by smoothing FireStMap[] and then storing the results in the coverage map (FireRate[]). Smoothing the location of fire stations produces a coverage which radiates outwards from each station, attenuated by distance and efficacy. Imagine looking at a map and squinting, and you have a pretty good idea of what smoothing accomplishes: the image is blurred. The smoothing distance is magnified by the fire coverage maps' 1/8th scale, as each cellular unit smoothed across corresponds to 8 cells of the Primary Map. That's how the city's fire departments' effectiveness is computed, but how is it used? The effectiveness, called Fire Coverage, slows and stops the spread of fire. Fire Coverage is read from by Map Scan when it updates fires, and used to adjust the probability that fires go out.

The use of two buffers allows fire coverage information to be simultaneously read from and written to during Map Scan. (Counting the temporary map used to do the smoothing, three buffers are used in total. These are not explicitly shown in this diagram, but are implied by the smoothing operations. See the appendix for a diagram explaining how the temporary map buffers are used.)

[26]Multiple interlinked systems have a powerful illusionistic quality, but can be difficult to author. Modular design is used to manage complexity. As we can see in the data flow diagram, simulation processes keep their own working models of the world, allowing them to be elegantly fire-walled from one another. Such separation is key for managing complexity, allowing new processes to be inserted into the flow of the simulation. For example, the police and fire coverage map computations are simple feedback loops that hang off the main map, and were added later in development (chapter 5; Wright 2016).

To maintain separation yet facilitate communication, different simulation processes "see" the same underlying data differently. Power Scan does not see a power line, hospital, or road, but simply cells which are conductive and those which are not. This is enabled by the Conduit Bit, a secondary map layer, which frees Power Scan from having to know anything else about the map. Secondary map layers are abstractions of the Primary Map, and offer parallel interpretations of the city. Such process specific representations enable processes to easily attend to and communicate about the phenomena they care about without getting caught up in all of the complexities of the simulation or map representation.

Besides secondary map layers, there is another way in which the map is easily abstracted, and that is through the physical organization of character indices. (See the appendix for a diagram of characters organized by index.) Logically related characters are close to one another, enabling the general type of a character to be determined with indexical operations. The Land Value computation, for example, is interested in detecting nature characters, all of which are clustered together in the landscape section, and can be tested with a simple inequality.

Character indexing, abstracted multiple views, and process separation enable loose couplings between subsystems, allowing them to interlock and work together together yet remain independent. Such modularity is expedient for an author, allowing the building of a complex system that is nonetheless scalable and separable. In other words, the system can grow complex yet remain manageable.

only map saved to disk, since the others are derived from it. And since the Primary Map is the main thing the player sees, it constitutes the underlying shared ground between the player and the simulation, and between simulation subsystems, ensuring coherency across all of these actors. The player sees and touches the same data map the simulation does, giving the landscape a high degree of tangibility and manipulability.[27]

[27] The linchpin role of both Primary Map and the Map Scan process is visually apparent in the data flow diagram. Primary Map is the clearinghouse through which information about the world flows—almost all feedback loops between maps are mediated via Primary Map. For example, Traffic Density does not directly contribute to Pollution. Rather, Traffic Density causes animated traffic to appear on Primary Map, and it is these traffic characters which then contribute to Pollution.

As the authoritative representation of the world, Primary Map plays a critical mediating role. First, a central representation of the city lends coherency to the design and unfolding of multiple, parallel, and multifaceted simulation processes. All of the secondary maps are derived, either directly or indirectly, from Primary Map. To see just how central it is, consider that Primary Map is the only layer *SimCity* writes to disk when saving a city. The secondary maps (not including the status bits) are rebuilt when a city is loaded, and are recalculated from scratch. Second, since the Edit Window depicts Primary Map, through which most spatial information propagates, the player's sense of direct access to *SimCity's* internal simulation is heightened.

The authority, directness, and centrality of the Primary Map is illustrated by considering how Traffic Density affects Pollution. Because of the Simulate schedule, it takes time for effects to propagate across systems. Requiring that animated traffic cars manifest in the Primary Map before contributing to Pollution ensures that the maps are synchronized in a way the player can see. If Traffic Density directly affected Pollution, the player might see pollution from simulated traffic (stored in Traffic Density) that has not yet been shown to the player as traffic characters. This mediation also contributes to coherency because of representational shortcuts the traffic generator takes, such as placing traffic along railroad lines (mass transit does not contribute to pollution), and encoding at 1/2 scale. Routing through the Primary Map ensures that automotive pollution only manifests where roads actually exist and traffic can be seen.

Also, look at how Land Value is at the center of a tangle, while the fire and police systems are relatively simple. This is because Land Value is one of the oldest and most important simulation systems Wright built—it is a central driver of growth and decay, while police and fire coverage was added later, as the simulation was elaborated (chapter 5; Wright 2016). This evolution underlines the modularity of this data flow network.

With the exception of Land Value, all feedback loops are quite simple and flow through Primary Map. Why does Land Value look like a tangled knot? First and most simply, Land Value is determined by multiple factors: the presence of nature (the unbuilt environment), pollution, crime, and distance to the city center. Second, Land Value is important, affecting both taxes collected (and thus player agency) and zone development. Land Value's importance means that *SimCity* takes special pains to clearly represent it to the player. While Land Value can be seen in the City Maps Window, it manifests most directly in the Edit Window via the kinds of buildings which develop in a zone. Residential and Commercial buildings span the gamut from run down to expensive looking, directly and visibly manifesting Land Value.

Animate Tiles: Liveliness

While Map Scan periodically spreads and extinguishes fires, handling their simulation logic, it is Animate Tiles that updates the flip-book animation for each fire, making it appear to burn. Map Scan is slow, beating out a slow and intermittent set of simulation transformations, but Animate Tiles is fast, producing the appearance of lively traffic, burning fires, curling smoke, spinning radar dishes, and stadium sporting events. The overall appearance of a lively simulation is sustained in large part by the real time animation and activity of both agents and Animate Tiles, which provide a high frequency polyrhythmic cover for *SimCity's* slower processes.[28]

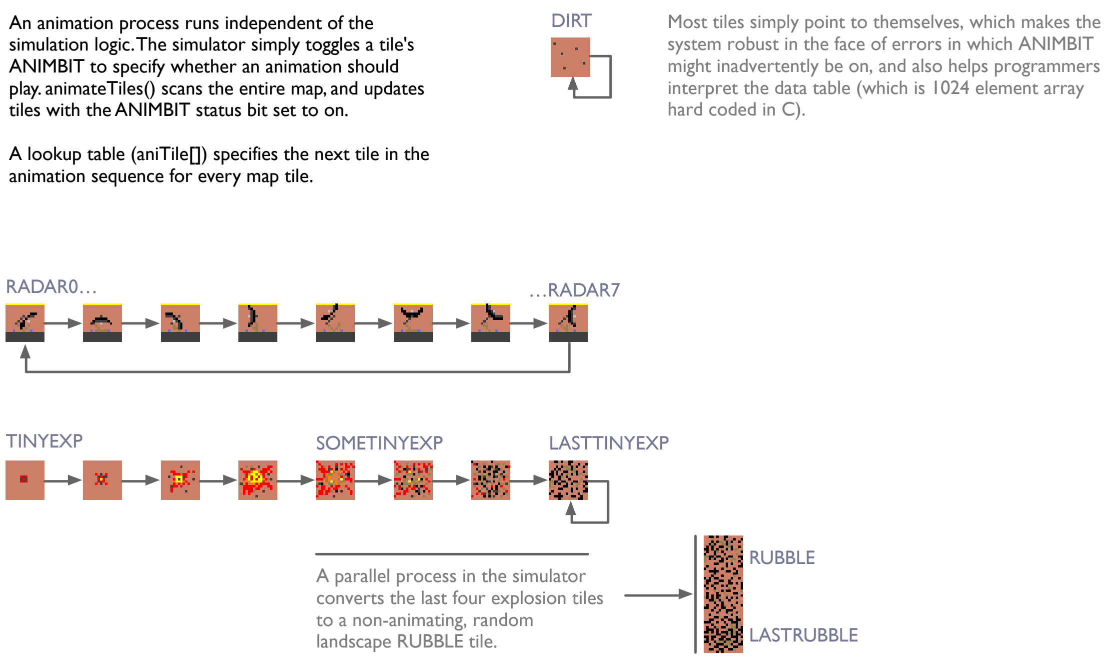


Figure 6.21: Animate Tiles: flip-book.

If a cell's Animate Bit is on, then Animate Tiles swaps out its graphical character, producing a flip-book animation. As these characters flip from one to the next, traffic appears to move

[28]Animate Tiles is one of the real time processes, like the user interface, that runs between the steps of Simulate. Animate Tiles's ability to update the entire map in real time has to do with its simplicity and elegance. Its inner loop is 14 lines of 68000 assembly code, and most cells—those which aren't animated—require only 5 instructions.

and fires appear to burn. This animation isn't a superficial gloss; it modifies the actual state of Primary Map.[29] This modular design allows Animate Tiles and Map Scan to run independently of one another. Coordination is easy: Map Scan simply flips the Animate Bit on or off.[30] Despite this simplicity, Animate Tiles must synchronize animations across cells. The landscape, after all, is a quilt of separate characters. Joined together, these characters represent road networks and multicellular buildings, and their individual animations must be synchronized to maintain visual coherency.[31]

---

[29]Animate Tiles transforms the Primary Map, swapping out the characters stored in each cell to produce flip-book animations. Each time Animate Tiles is invoked the entire Primary Map is scanned, and each cell is inspected to see if its Animation Bit is on. If it is, then the cell is animated.

Animating a cell is simple. As you can see in the Animate Tiles flip-book figure**,** each character points to the character which succeeds it. Combinations of links work together to define linear and looped animation sequences. (Link information is encoded in a simple lookup table, i.e. new = aniTile[old].) Characters that don't animate point to themselves, so they won't change even if their animation bit were to inadvertently turn on. These succession links are a simple, fast, and easily scalable way to encode animation sequences that are both linear and looped.

[30]The animation is not a separate representational layer or system, but transforms, through animation, the authoritative Primary Map. This shared representation enables Animate Tiles to talk to and coordinate with Map Scan and other simulation processes. This talking happens in terms of the Animation Bit and character swapping. Yet Animate Tiles is also elegantly decoupled from the rest of the simulation, running in parallel and at a different pace. This elegant modularity affords some interesting interplay between Animate Tiles and Map Scan. Map Scan can "fire and forget" animations, simply by flipping the Animation Bit on. It need not concern itself with multicellular synchronization, which Animate Tiles takes care of. Map Scan can also react to the animation system, as you can see in the explosion sequence**.** Tiny explosions are created by Map Scan, perhaps in response to a building exploding, or maybe a rampaging Godzilla. (This tiny explosion, modeled in a cellular automaton, is not to be confused with the big explosion modeled via the agent system.) These explosions are easily created: just set a cell character to TINYEXP and flip its Animation Bit on. Animate Tiles takes over, flipping the explosion animation characters one after the other. A tiny explosion animation finishes by turning into a one frame animation loop of rubble (LASTTINYEXP). Map Scan then detects that character, and then places some non-animating rubble into that cell, bringing the animation to an end. Map Scan and Animate Tiles are independent yet work hand in hand.

[31]Multicellular animations require synchronization to maintain coherency across cells. Many animations span cells: smokestacks are multi-celled, the football game spans two cells, and in the animated traffic tiny cars drive off one cell and into another (as long as the traffic density is equivalent). To ensure that such animations are synchronized across cells a separate synchronization lookup table is used. This table defines which step of a global 8-frame synchronization clock each animation frame synchronizes to. When AnimateTiles() fetches the next character in an animation sequence, it skips animation frames until it finds the next frame that matches the synchronization clock. Some animations, such as fires, aren't synchronized at all, producing visual variety.

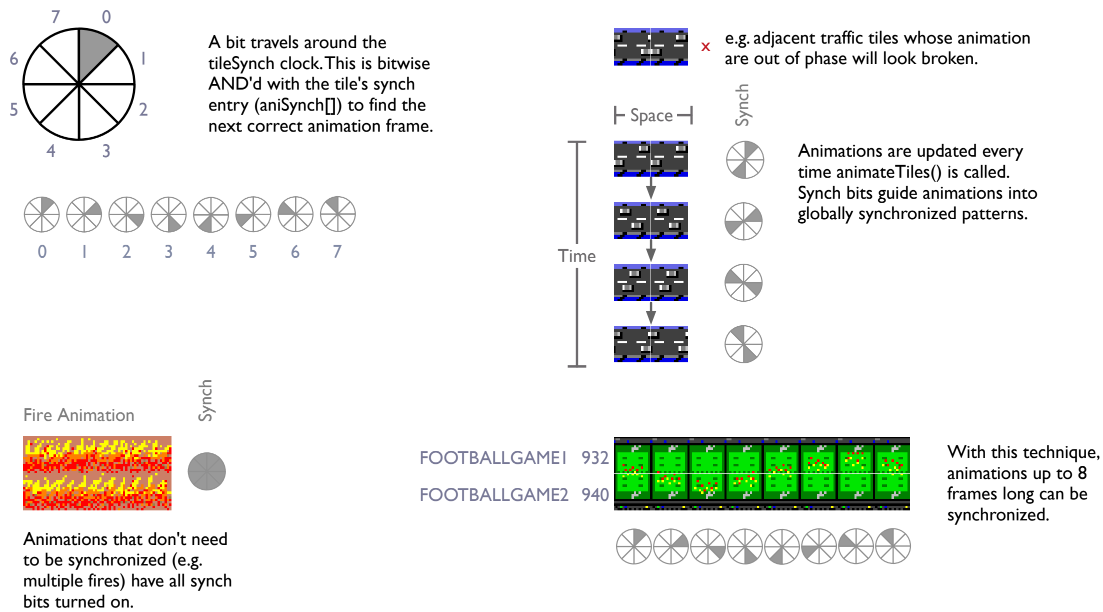


Figure 6.22: Animate Tiles: Synchronization. An 8-state clock (tileSynch) controls the phase (synchronization) of animations.

Animate Tiles provides a seamless cover for the fragmented update of Map Scan and Simulate, creating an illusion of smoothness and quickness that Map Scan and Simulate do not. It also provides cover for the cellular nature of the landscape, orchestrating the flip-book of each tile so that, seen together, the illusion of multicellular entities is sustained.

Multicellularity

Buildings, transportation networks, and forests are all examples of multicellular representations. As we have seen, large spatial entities are modeled at a level of granularity finer than the objects themselves.[32] The illusion of multicellular wholes is produced through graphical

[32]Modeling a world at such a fine level of granularity, as we saw in chapter 4, is a hallmark of cellular automata. Cellular automata makers employ graphical and textual descriptions to ground abstract states into specific representations, a discursive practice that also facilitates the representation of multicellular entities—at least in the mind of the viewer. *SimCity* carries forth this tradition, employing detailed bitmap artwork, shifting the burden of narrative work from peripheral text and diagrams into intrinsic images. What *SimCity* loses in evocative abstraction, it gains back in clarity, specificity, and approachability. Furthermore, *SimCity's* careful employment of multicellular visual representations—never a feature of traditional cellular automata, aside from secondary anno-

character artwork, synchronized animation, simulation behavior, and editing tools.

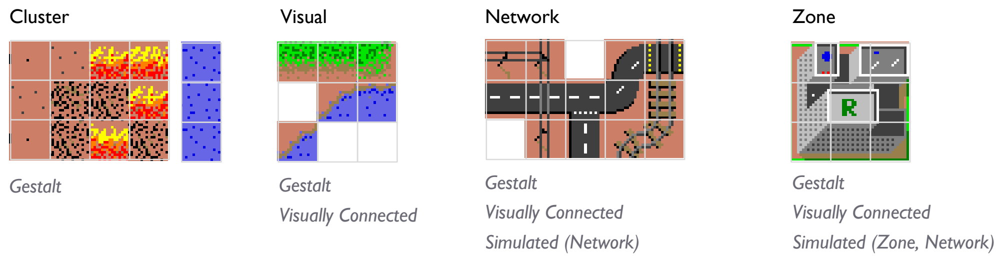


Figure 6.23: Typology of multicellular structures.

Almost everything in *SimCity* is multicellular in one of four senses:

- **Clusters** are multicellular through perception only. A cluster of fires, water, or dirt read as multicellular wholes through gestalt perception. *SimCity* aids this process only indirectly, if at all, either through the world generator, which establishes coherent bodies of water and land, or the emergent patterns of spreading fire.

- **Visual.** Coherent coastlines and forest edges are reinforced by placing visually connected characters next to one another, forming figures that span multiple cells.

---

tations, as in Kemeny's Scientific American article (chapter 4)—helps players see and experience the multicellular formations, and by extension the simulated world and its emergent patterns.

Cellular automata makers break the world down into granular cells yet retain a keen interest in how collections of cells function as wholes. Stories of cellular automata play a crucial role in putting the pieces back together. This reassembly, often traveling under the banner of "emergence," is aided by textual and visual narration. While the simulation may in fact spin new wholes out of its cellular mechanisms, viewers and players of cellular automata systems need guidance in order to see, experience, and appreciate this reassembly. And sometimes this reassembly is simply a descriptive gloss or some other kind of artifice. If, in such cases, emergence is not revealed to be a kind of conjuring trick, then such narratives at least complicate the idea of emergence.

SimCity often figures in discussions of emergent systems, but the Zone Bit complicates matters. While a zone is in some ways the sum of its parts (e.g. cells in a power network), its behavior as a whole emanates from the zone cell in top down fashion. The Zone Bit is an excellent figure for how a sense of bottom up emergence is cultivated while employing top down controls, which is a theme that repeats throughout *SimCity*, whose simulation is a hierarchy of top down and bottom up processes and representations. *SimCity* is simultaneously a cellular universe and a world filled with logical, higher level objects like buildings. From the Zone Bit to the system dynamic valves that guide the entire city's evolution, many invisible hands are at work, artfully shaping and guiding a sense of emergent activity.

- **Network** entities not only appear connected, but are simulated as such. Network simulation processes treat road, rail, and power line cells as multicellular wholes. Power Scan, which distributes electricity across the power grid, interprets anything with the Conduit Bit on to be part of the electrical grid, combining buildings, zones, and power lines into a multicellular whole.[33]

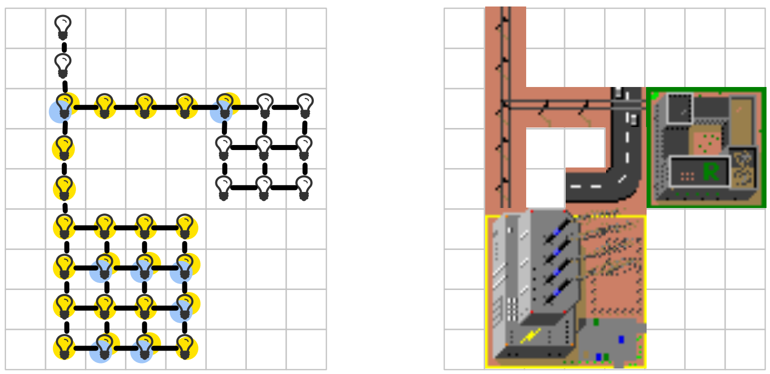


Figure 6.24: Power Scan network. Power Scan activates cells as a simulated network, interpreting the map in terms of new multicellular formations: a network through which power flows.

- **Zone**. Like network entities, zones appear multicellular, are visually connected, and are simulated as coherent wholes. If the Zone Bit is on, then a cell will that act and react on behalf of the entire multicellular zone.[34]

[33] More examples: the trip generator, which generates traffic and determines connectivity, interprets adjacent road and rail cells as constituting a multicellular network. Likewise, the train agent navigates adjacent rail cells as if they were multicellular wholes. The ship agent traverses water cell networks as navigable waterways.

[34] As should be clear, multicellular membership is not exclusive. Cells concurrently form the elements of different kinds of multicellular wholes. They can be simultaneously simulated as individual cells, part of a network, and part of a multicellular zone. Cells can be concurrently simulated at three different levels:

1. *Unicellular*. Cells are the flexible universal substrate out of which cities are simulated. Single cells can, for example, be independently destroyed. As we have seen, fires can burn and destroy individual cells, and Animate Tiles operates only upon individual cells, employing a synchronization table to ensure coherency in multicellular animation. (Also, traffic destinations are also concerned only with whether they have reached a particular character. The trip generator doesn't care whether it has reached the appropriate multicellular building; a single destination cell stands for the whole.)
2. *Multicellular Zone*. The cells of a building grow and decay as a multicellular whole, the census tallies zone cells, and the locations of police and fire are tracked through zone cells. Many zones simply explode if

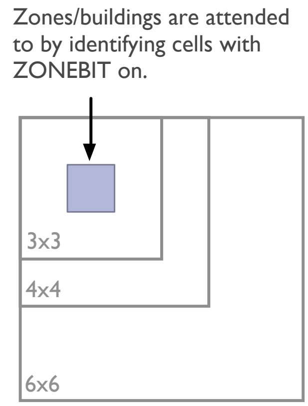


Figure 6.25: Zone cell.

the zone cell is damaged. As we saw in Map Scan, the zone character plays the special role of acting for the entire zone. Zones act and are activated as orchestrated wholes through the special Zone Bit. So, if the zone cell is destroyed then the entire zone is effectively destroyed, but if a different cell is damaged then the zone persists.

3. *Multicellular Network*. Elements of a power or transport network. Unlike zones, whose multicellularity is governed by special zone cells, networks are emergent multicellular formations.

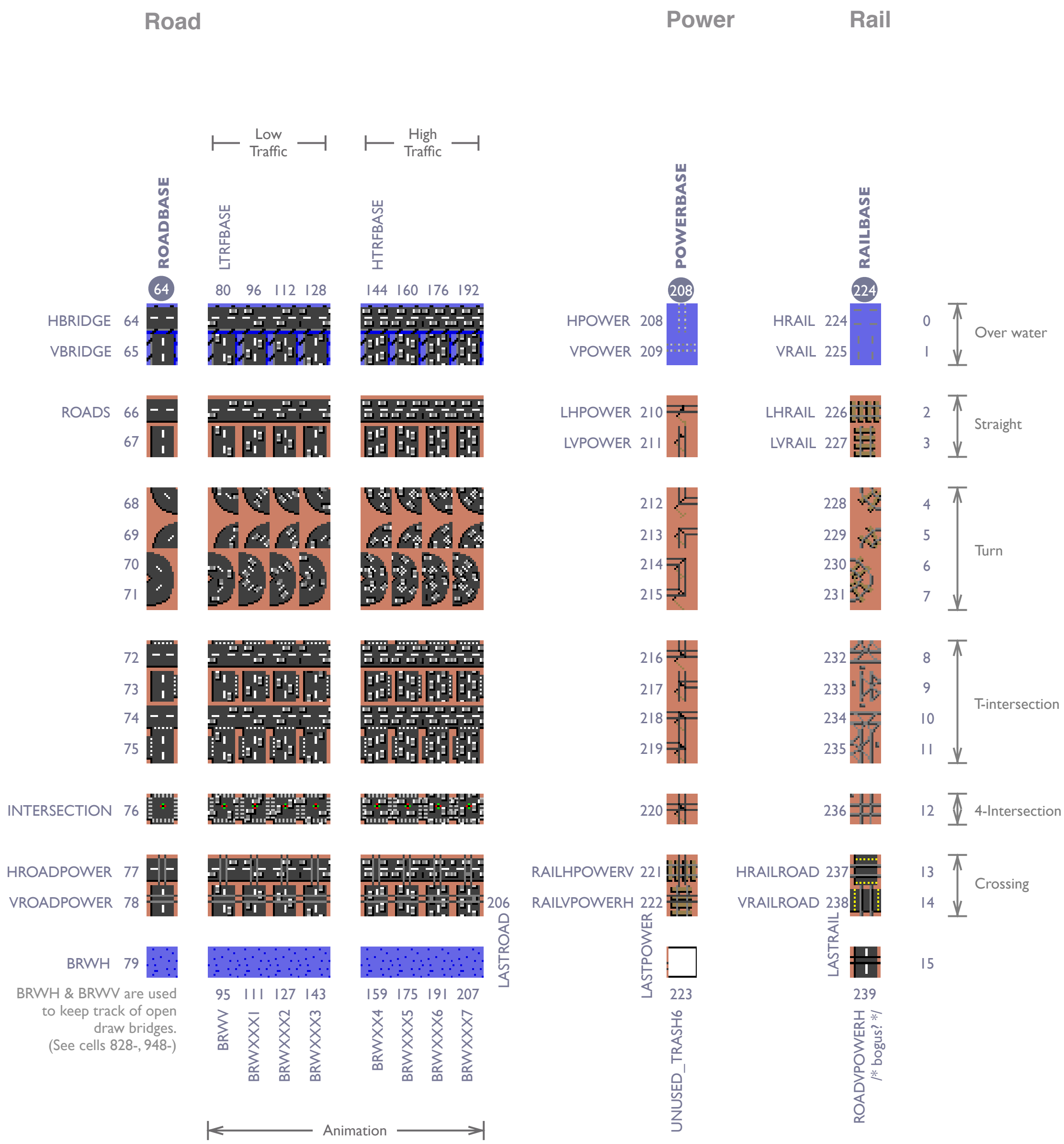


Figure 6.26: Network Characters.

Visual presentation reinforces the simulation's interpretation of network cells as interconnected.[35] By selecting the appropriate character, multicellular tapestries are produced from

[35]There is an exception to this, notable as an example of how the simulation logic diverges from its visual presentation (and therefore player expectations). Although roads and rails only appear to cross at intersections, the traffic generator conceives of road and rail as forming a single navigational mesh. Trips can jump wily nilly between adjacent rail and road cells—even if they don't visually appear to connect. Also, player expectations afford the (correct) interpretation of the ring roads that surround buildings as interconnected to the building itself,

the recombination of individual characters, giving rise to road, power, and railways networks that traverse the map. By doing this as the player lays down zones, road, rail, and power line cells, the network's multicellular coherency is established, along with a magical feeling of augmented authorship.

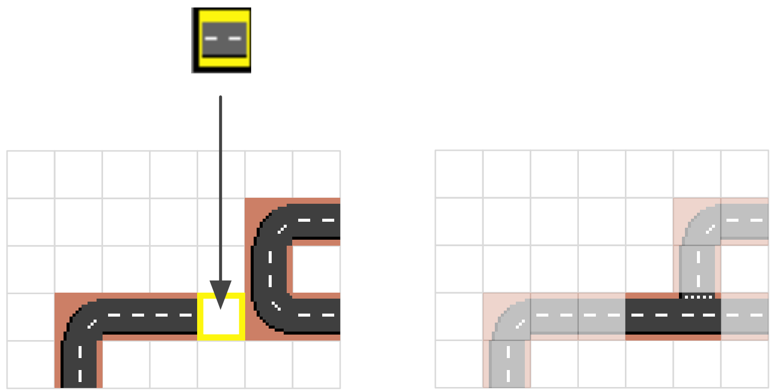


Figure 6.27: Connect Tile. When the player places a road, Connect Tile updates the tile and its neighbors to ensure they look connected.

System Dynamics: A Coherent Whole

In addition to spatial representations, *SimCity* employs systemic models of the city. Systemic models operate at a city-wide scale, imbuing the city with globally coherent behavior. This allows player, simulation, and simulation maker to see, think, and act at an overview level. One example is tax collection, which is based upon the tax rate, average land value, population, and game difficulty.[36] If the map provides a shared grounding for the city in detailed plan, then

even if the road itself does not visually cross into the zone.

[36] In addition to the RCI Valves, and the regulators for hospital and church population, the city budget operates as an overarching system in conversation with the cellular automata. This system is shown to the player in the Budget window, from which the player can set the tax rate and funding levels for fire, police, and transportation infrastructure. Population, average land value, tax rate, and game level are combined to determine taxes collected. Upkeep costs of infrastructure is determined by the census data tallied by Map Scan. RoadTotal, for example, tallies the funding needed for transportation upkeep. Each rail and empty road counts for +1, and bridges and higher traffic roads cost more. When fed back into the cellular automata, incomplete funding levels causes transport infrastructure to randomly decay, and diminishes the efficacy of police and fire departments.

the systemic processes provide a shared grounding for the city in overview.[37]

Churches and Hospitals

By far the most important system model is of citywide zone demand, a system dynamics model whose valves regulate the growth and decay of zones across the map. To understand this model, it will be helpful to first look at a simpler and functionally analogous system, the growth and decay of churches and hospitals.

Whereas tax and funding policies in *Urban Dynamics* are baked into the model—the "player" doesn't play it as much as design, wind up, and watch it unfold—in *SimCity* players can adjust these parameters and see what happens.

[37]This shared grounding is between the player, simulation subsystems, and simulation maker(s).

Reconciling cellular automata and system dynamics in this fashion places the system dynamic process in a privileged position. The system dynamics process, by collating populations and setting the valves which govern and guide zone growth, exerts a strong influence upon the evolution of the entire city. Furthermore, the modular design of *SimCity's* simulation processes means that this influence is focused in a single highly leveraged subsystem. This leverage has implications for both the creator and players of *SimCity*.

For the simulation creator, this focus affords centralized control over the unfolding simulation. System dynamics, acting like an invisible hand that guides the entire city, is itself guided by the simulation author. Wright could, from a single place, tune values and rules with global implications. You might be surprised to learn that *SimCity*, a simulation often used as an example of bottom up emergence, is strongly governed by top down mechanics. Given my personal experience working with Wright, this is not surprising at all. When working for Wright on an early *Spore* prototype, in which I was writing rules to generate spiral galaxies, I recall being surprised at his encouragement to be more "heavy handed." Wright was telling me, in other words, not to worry about exerting top down authorial pressure in order to achieve the desired results. Such focused governance affords design of a system's overall evolution, even if that system is comprised of many independently interacting and emergent behaviors. This grants the author a strong position from which to sculpt the player's experience. In Set Valves, for example, the growth of residential, industrial, and commercial zones is stopped if the city's demand for a stadium, seaport, or airport was not met. Those caps are enforced in the system dynamic process, into whose global logic they neatly and logically fit.

Players also make use of this focused leverage. The game level, chosen by players when creating a city, and the tax rate, adjustable in the budget window, strongly influence global demand. Players thus manipulate the simulation at local and global scales. Not only does the system dynamics process give cities coherent dynamics as a whole, but it affords top down control for both the simulation creator and players.

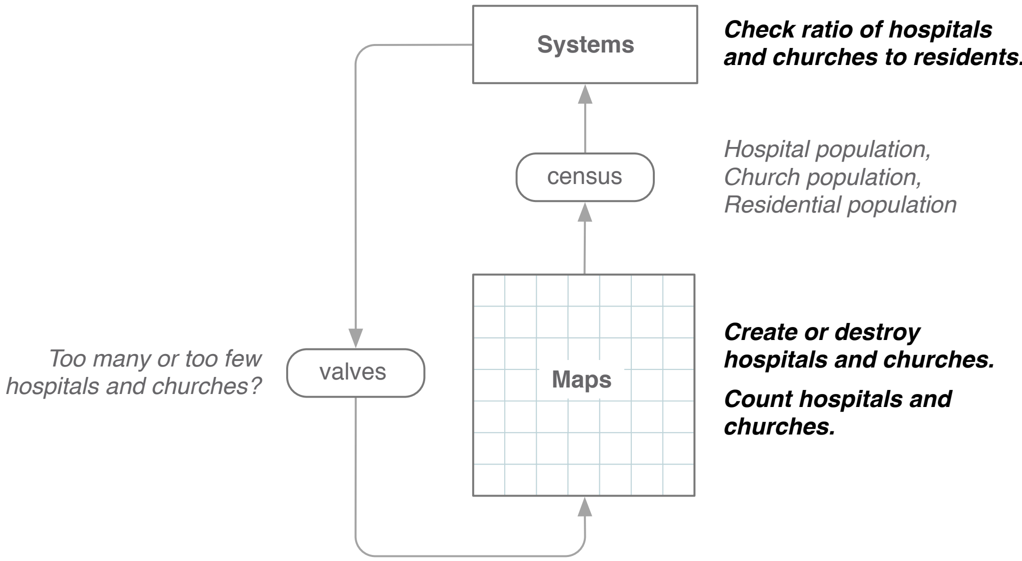


Figure 6.28: Coupling system dynamics and cellular automata to regulate hospitals and churches.

Globally, the city strives to maintain a balanced ratio between residents and the hospitals and churches that serve them. Map Scan, by collecting census tallies for residents, hospitals, and churches, allows the system model to decide whether more or fewer hospitals and churches are needed. After the systemic calculation is done, Map Scan knows, when evaluating each residential zone, whether it should try to erect or tear down a church or hospital.[38] And so, two incongruous simulation techniques—system dynamics and cellular automata—are reconciled. A highly elaborated version of the same basic idea animates the growth and

[38]Every once in a while empty residential zones become hospitals or churches, and sometimes these hospitals and churches transform back into empty residential lots. This design answers these questions: How does the simulator decide where churches and hospitals go? And how many? Hospitals and churches sprout in proportion to the city's residential population. The simulator seeks a population ratio of 1 hospital and 1 church for every 128 residential citizens. This is accomplished with a very simple system dynamic style mechanism.

As Map Scan performs its cellular updates it keeps a running tally of the residential (ResPop), hospital (HospPop), and church (ChurchPop) populations. This tallying transforms the cellular automaton representation of a city—composed of cells and states and space—into a system dynamic style representation: populations. The system dynamic mechanism now has the information it needs to work. It computes the ratios and updates the valves NeedChurch and NeedHosp to reflect whether more or fewer churches and hospitals are needed. These valves serve as a signal from the system dynamic governor to the cellular automata. If this signal says there are too many churches and hospitals, then there is a random chance that a church or hospital will transform into an empty residential zone; if there are too few, then a developing residential zone has a chance of becoming a hospital or church. RCI Valves are a highly elaborated variation upon this basic idea.

decay of residential, commercial, and industrial zones.

Demand Valves

A system dynamics style simulation governs citywide demand for residential, commercial, and industrial development. These are described to the player as *demand*. Internally, they are labelled as “valves,” signaling the debt to Forrester.

Figure 6.29: RCI Demand. Detail of Edit Window. Player facing representation of the RCI valves.

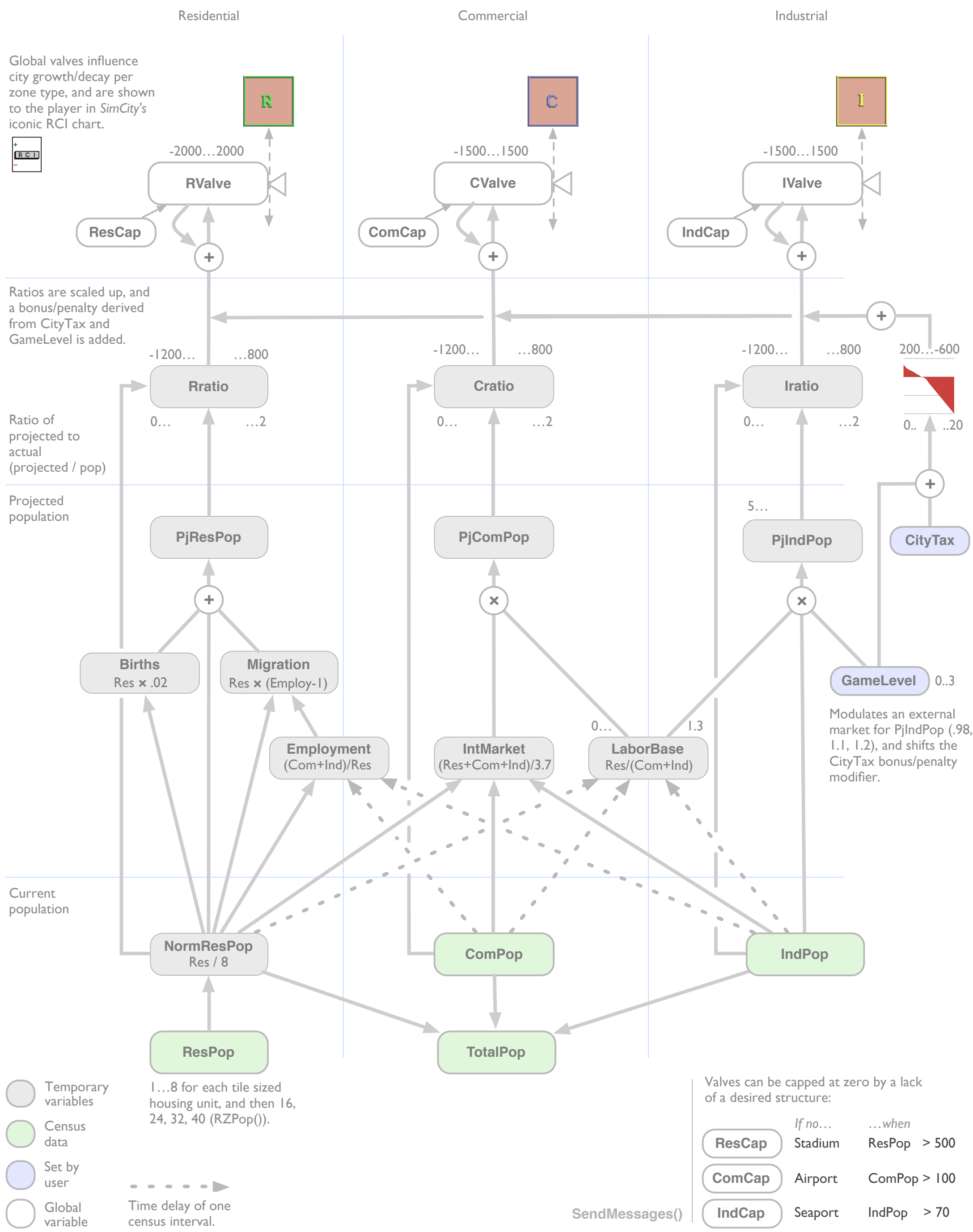


Figure 6.30: SetValves(). Census data (bottom) is input, and valves (top) are output.

This logic is more complex than that used for hospitals and churches, but the basic idea, of using population ratios to guide growth, is the same. These valves are updated according to formulas that represent population fluxes (migration, births, deaths), the labor pool, job availability, the employment rate, the market for goods (internal and external)[39], and the inclement affect of taxation on development. The city's current census is taken in, projected populations are computed, and then the resulting values, the valves, will then act as a "spigot" (Wright 2016) that guides the development of zones across the map.[40] As in Forrester's *Urban Dynamics* model, time lags and inertias are intentionally built in. In addition to these system dynamics style formulas, growth is halted if the city nurses an unsatisfied desire for a stadium,

---

[39]The "external market," which is a constant factor determined by the Game Level (difficulty), manifests in code only as a comment within SetValves(). The comment says "New ExtMarket," indicating this is a revised external market. There is evidence of a prior external market, in the form of the global variable EMarket, which occupies some of the lowest (and presumably oldest, in terms of design history) space in the miscellaneous data stored when saving a city. EMarket is saved and loaded, but the simulation never uses it, indicating a vestigial simulation variable.

[40]While there is a lot going on in this diagram, its overall structure is quite simple. Information flows upwards from the bottom, in the form of census data gathered from the cellular automata, towards the top, where it emerges transformed as three valves: RValve, CValve, and IValve. These valves then go to on influence the cellular automata simulations.

How is population data transformed into growth controlling valves? Set Valves begins with the population data gathered by Map Scan. These populations are transformed by processes that model births (and implicitly deaths), migration, employment, an internal product market, and labor pool into projected populations for each of the three sectors. Since Set Valves cannot directly change the actual populations—the homes, industry, and commercial populations, remember, are represented in the Primary Map—it must instead signal to the spatial zone development processes, just as NeedChurch and NeedHosp did. To send this signal, the ratio between the actual and desired (projected) populations are computed. These ratios are added to RValve, CValve, and IValve. Thus, if a population is projected to go down, then the valve will begin to close—its numeric value goes down, potentially going negative and causing decay. If a population is projected to go up then the valve opens—its numeric value goes up, stimulating growth. The tax rate (and game level) apply a bonus/penalty to demand in all sectors; high taxes retard growth, while low taxes encourage it.

As you can see, the SetValves() figure is roughly divided into three columns, each of which corresponds to computations for the residential, commercial, and industrial sectors. While the hospital and church feedback loops discussed earlier were independent of one another (hospital and church populations didn't affect one another), we see here that the residential, commercial, and industrial sectors are deeply interlinked and influence one another.

Let's zoom in and see how projected populations are actually modeled. Projected residential population is the sum of three values: births, existing population, and migration. This formulation can be traced to Forrester's *Urban Dynamics* model, in which the city's relative attractiveness—largely a function of jobs—to its environment plays a crucial role in determining population flux. Migration in *SimCity* is driven by job availability, which is a straightforward ratio of jobs (commercial and industrial population) to residential population. Labor base is the inverse of job availability, and contributes to both the projected industrial and commercial population. Commercial population is also determined by the city's internal market for goods and services, derived from the combined residential, commercial, and industrial populations. An external market also contributes to the industrial demand.

airport, or seaport.[41]

Zones: Automatic Building Placement

A large part of *SimCity's* magic is in how residential, commercial, and industrial zones are automatically built up by unseen simulated citizens. The inhabitants that move into your city, building it up and bringing it to life, respond to global demand (the system dynamic valves) as well as local circumstances (the map).

Empty lots zoned by the player blossom into different kinds of factories, housing, and commercial areas. Each zone represents populations of residents, industry, and commerce. As a zone grows or decays, its population ratchets up or down. Population density as well as land value are represented by the graphical characters used.[42]

[41] The messaging system, if it determines that one of these facilities is needed, sets one of three flags (ResCap, ComCap, IndCap), which Set Valves then uses to prevent demand for that sector from ever going positive.

[42] Each time a zone grows or decays its population density notches up or down. Residential zones, for example, initially grow and decay at the scale of single cell homes (up to 8), progressing to multicellular 3x3 zone transformations (at populations above 8). Land Value and Pollution are reflected in the particular characters chosen to represent that population density.

This representation is more than visual, as the characters are ground truth for the simulation. As far as the simulation is concerned, the characters are the zone type and population density.

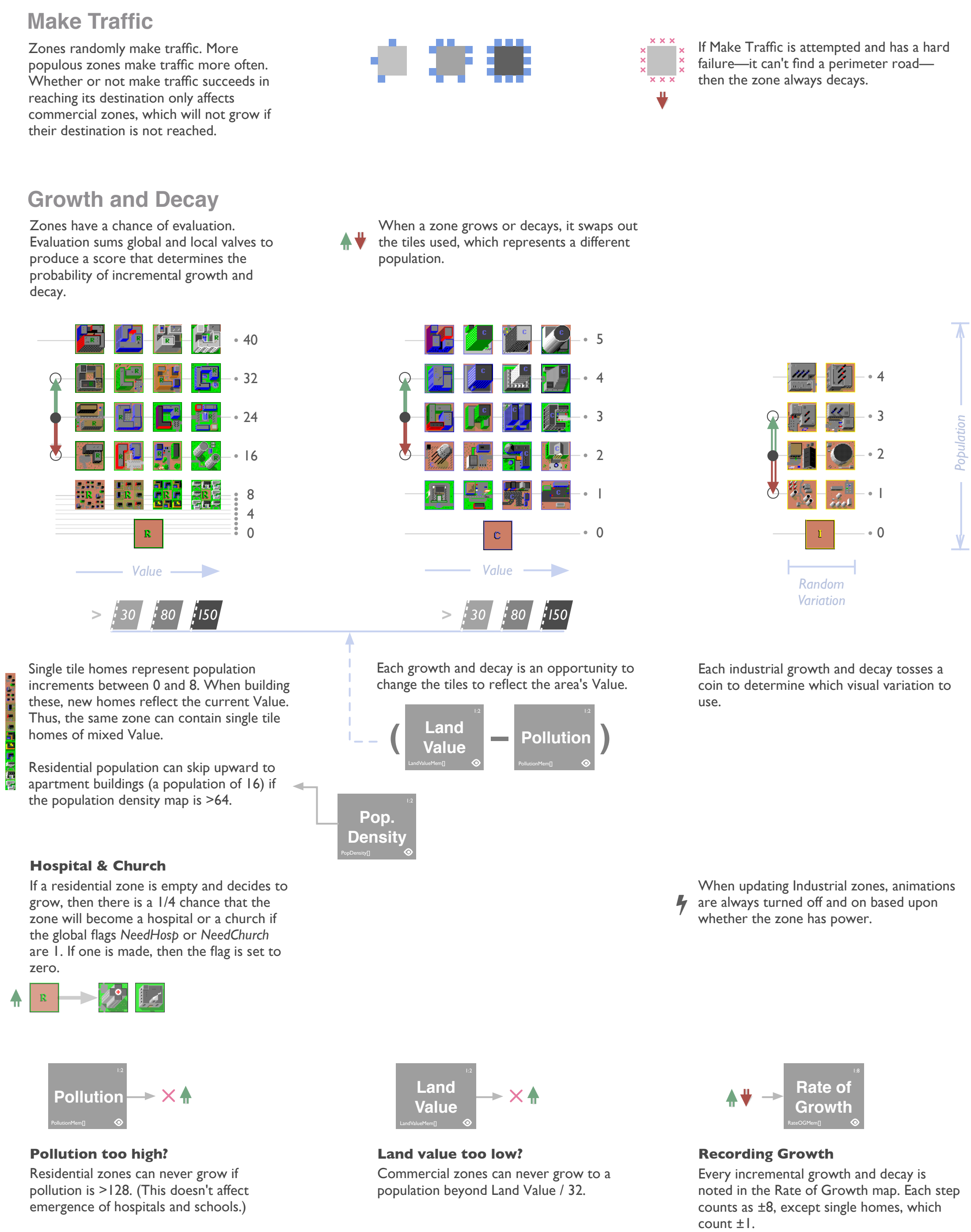


Figure 6.31: Zone Development: Overview

Many factors affect the evolution of a zone: random chance[43], global citywide demand

[43]Besides adding a satisfying unpredictability to growth, random chance removes an undesirable spatial bias

(whose logic we have just seen), and local circumstances such as electrical power, transportation connectivity, land value, pollution, population density, and distance to the city center. As zones are updated, they generate traffic, which will lead to cars on the road and pollution in the air. Wright likens citywide demand to a spigot that modules the growth of zones throughout the map (Wright 2016).[44]

from processes like Map Scan. The same is true of other simulation processes, such as the spread of fire.

[44]The first step of zone updating is trip generation, which puts traffic into the map and allows the zone to react to the local transit network and nearby zones. Zones probabilistically generate traffic with a frequency proportional to their population; more populous/developed zones make traffic more often. Make Traffic generates random trips through the transportation network, trying to reach a particular destination based upon the origin zone type. Make Traffic can either succeed (successfully reach the desired destination type), fail (fail to reach the destination type), or fail hard (origin zone isn't adjacent to any road or rail). Hard failures means the zone always decays.

As long as a hard trip failure hasn't forced decay, zones have a 1 in 128 chance of updating. (Residential zones with a population less than or equal to 8 always update.) When zones update they have a random chance of growing, decaying, or staying the same. These probabilities are governed by a score, and this score is determined by summing what the *SimCity* code refers to as local and global valves. This probabilistic score, then, is where the system dynamics valves that represent global demand touch each zone. Rather than influencing a single global population flux, as would be the case in traditional system dynamics, these valves influence the probabilistic population flux in each and every zone.

Each zone type is influenced by a different mix of local and global factors. While residential zones are guided by local and global forces according to almost equal proportions, commercial and industrial zones are guided almost entirely by global demand. While local factors hardly figure into the score, and thus the growth probability for commercial and industrial zones, these zones are nonetheless strongly affected by local factors. For example, a lack of power automatically forces every zone type to have a score of -500, which means that it can only decay, and with a probability of 10.5%. And as shown in the Zone Development Overview figure**,** traffic and land value also modulate commercial zone development.

The combination of local and global interactions interweaves individual simulation elements into an organic whole. SetValves(), by cross-correlating many variables, brings individual zones in contact with the rest of the city. The residential valve, for instance, is primarily governed by migration, and this migration is mostly a function of employment: the ratio of jobs (commercial and industrial zone population) to residents (residential zone population). Global demand is also modulated by factors such as taxes. The net effect of all these information flows, focused through the system dynamics process, is that the city acts as a coherent whole, with individual zones responding to both their local environment and global citywide circumstances such as the tax rate and the relative mix of zone populations.

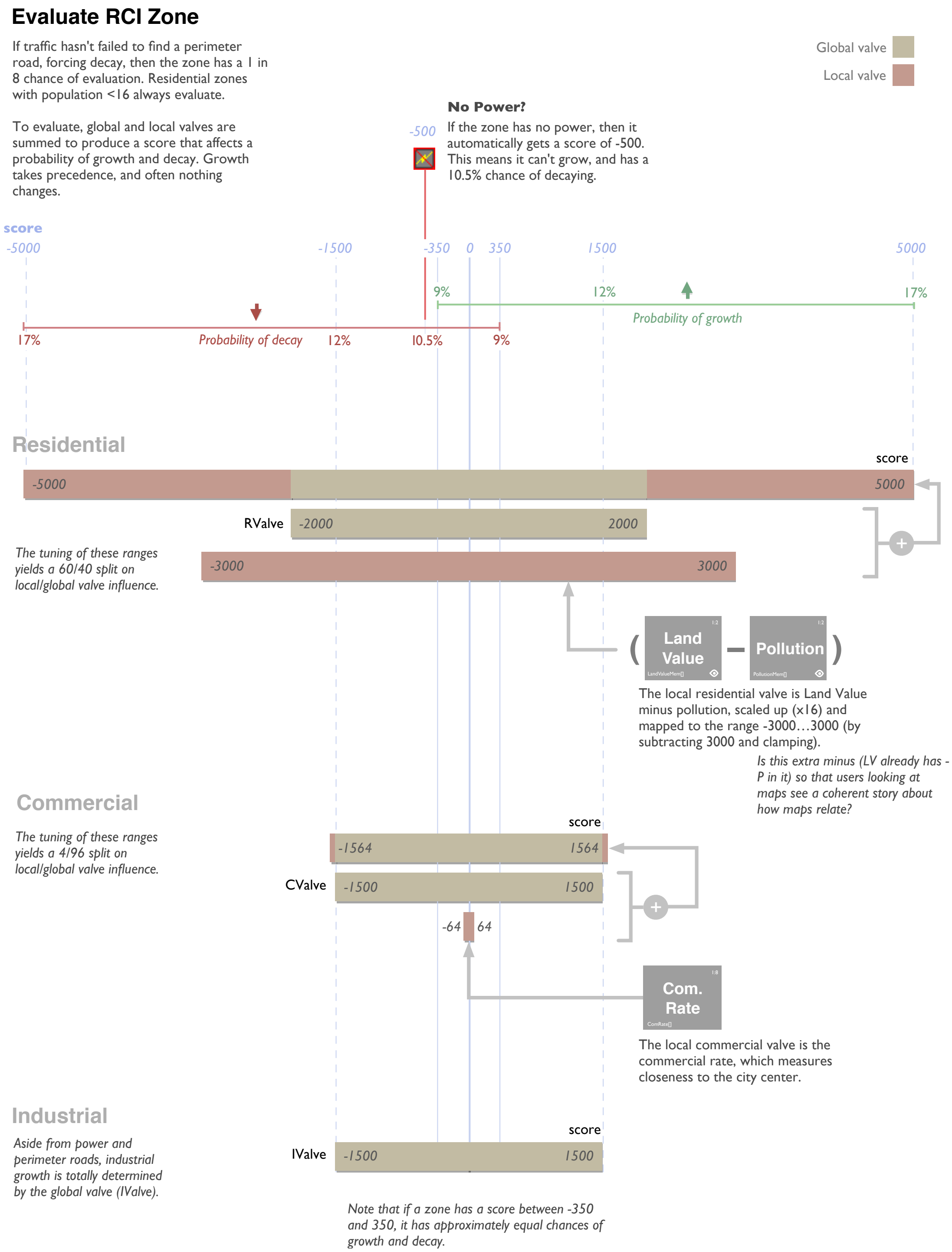


Figure 6.32: Zone Development: Evaluation

Agents

In addition to conjuring a world out of simulation materials like cellular automata and system dynamics, *SimCity* employs an agent based simulation of a train, helicopter, airplane, ship, monster, tornado, and big explosion.[45] These animations and behaviors update at a higher frequency, outside of the multitasking scheduler, imbuing the agents, and the world they romp in, with spots of vivacity.

[45]The sprites reflect a debt to the Commodore 64's graphics hardware, which supported 8 simultaneous sprites. Also, while all of these can be active at once, only one of each can exist at a time.

TRA Train

Spawned by train tracks, it randomly wanders the map.

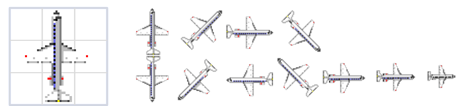
COP Copter

If it encounters heavy traffic (>160) in the traffic density map, it will make a report. High traffic tiles encountered during traffic generation will become the copter's destination.

When spawned, it goes to a random destination. When a destination is reached it returns to the airport.

AIR Airplane

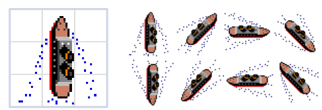
Takes off from the airport, and randomly flies around the map. Can crash into the copter (destroying both of them in an explosion object).



SHI Ship

Spawned by the Port, and randomly wanders the map. Can disappear off the map, or run aground and destroy map tiles and itself.

Makes sounds.



GOD Godzilla

Emerges out of water and heads towards the city's most polluted location. After reaching this destination, it returns to its origin (and disappears). Can randomly spawns if disasters are on and average pollution exceeds a threshold (80).

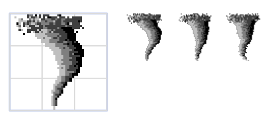
Can collide with and destroy airplanes, copters, ships, trains, and map tiles.

Makes sounds.

TOR Tornado

Lasts for 200 ticks. Wanders the map, and can disappear off the map.

Can collide with and destroy airplanes, copters, ships, trains, and map tiles.



EXP Explosion

Spawned during some object collisions and zone explosions. Spreads fire to some nearby map tiles.

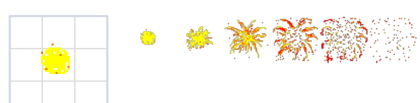


Figure 6.33: Agents. Internally SimCity's *code refers to these as "objects." Note that the monster art used to look more like Godzilla, but legal action with Toho forced Maxis to rework it into the giant orange newt shown here.*

Agents add believability, detail, and liveliness to the landscape. They organically emerge from the world—planes and helicopters from the airport, a ship from the seaport, Godzilla from the sea—and interact with it as you would expect. A train rolls along the tracks, the ship navigates the waterways, causing drawbridges to open and close, and Godzilla rampages, leaving a wake of destruction. Agents give the city a sense of self-awareness, as when the helicopter reports on high traffic. They also illustrate the importance of robustly handling edge cases: collisions often result in explosions, as when the ship runs aground or Godzilla smashes the helicopter.[46]

Send Messages: Guiding Make Believe and Play

SimCity not only shows you the city, but tells you about it, too. In order to make his toy more accessible to others Wright added "more clear feedback, in terms of what people need and why they need it" (Wright 2016). A messaging system not only tells players about the city, describing what is happening, but often supplies interpretation and guidance, directs attention to problems, and suggests possible solutions. This message system is a crucial part of *SimCity's* discursively constructed play frame, scaffolding players into fluent interpretation

---

[46]These agents are spawned endogenously by the simulation itself, often triggered by a disaster (which might, in turn, be caused by the player using the Disaster Menu), and some are triggered by the cellular automata. The airport spawns a plane and helicopter, the seaport a ship, train tracks spawn a train, and when some buildings are destroyed they explode. Agents affect one another and the map. If objects collide with one another, for example Godzilla and a train, or the helicopter with the plane, then they explode. This explosion is itself an agent simulation, and might in turn set the map on fire. Tornadoes, Godzilla, and crashing ships can damage the map, smashing cells into debris or inserting fire into the map. Not only do agents modify the cellular automata, but the cellular automata can respond to the agent model: the ship's approach can cause bridges to draw open.

Agents also respond to other parts of the simulation, adding to the city's sense of aliveness. The monster is spawned when the pollution average is high, and seeks out the most polluted part of the city; trains follow train tracks; the ship navigates waterways; the traffic generator causes the helicopter to seek out high traffic parts of the map, which then reports on high traffic as it flies overhead.

Collisions and explosions are a humorous, simple, and robust way of handling the unexpected collisions and overlaps which are certain to occur in such an open ended system in which players are invited to create and play with juxtapositions and permutations. Players, for example, might be tempted to trap and attack the ship with bridges, and an explosion—the result of a ship-bridge collision—is a fitting response to such an edge case.

of, and competent play with, their miniature cities.[47] While some messages directly emanate from simulated events—insufficient funds to use a tool, an earthquake, or monster sighting—a reflective system called Send Messages, operating like a coach, observes the simulation and supplies a running commentary. By reflecting upon things like the ratio of residences to roads and fire stations to populace, whether there are brownouts, and tax rates, messages are generated that interpret what is happening, explain why, and advise on what to do about it.[48]

Artifice and Belief

SimCity transforms a computer into a city, and the player into a super powerful city planner. It invites you to play in a world of fantasy, to join its playful pretense. Like an older sibling, parent, or playmate, *SimCity* scaffolds the player's make believe. The discrete space and states of

[47] The City Evaluation, feedback from citizens—approval scores and problem rankings, also scaffolds interpretation of and play with the city. Unlike the messages, headlines which pop up above the city, the evaluation feedback is passive, providing secondary information about the city that players can consult.

[48] Messages are periodically sent by the simulation to the player, alerting them to disasters, highlighting problems, and offering advice. These messages originate within the simulation and are typically shown at the top of the Edit Window, although some present in an illustrated dialog box. Some messages are accompanied by sounds, and those that occur at a particular place on the map are accompanied by a "GoTo" button. If "AutoGoto" is on then the Edit Window automatically scrolls to that location. Some simple prioritization logic acts as a kind of message traffic controller, ensuring that simultaneously firing messages are properly prioritized, ensuring, for example, that messages which respond to user actions always take precedent and that the player doesn't see a rambling or flickering message display.

Messages are dispatched from two kinds of places: organic simulation events and a system commentary.

Organic simulation events originate in discrete simulation happenings. They are triggered by events such as disasters, insufficient funds for tool use, and insufficient electricity in Power Scan. Such messages are dispatches from within the simulation, enabling the running simulation code to narrate itself.

The second kind of messages, system commentary, is generated by a subsystem called Send Messages. Send Messages inspects the city's statistics and, if the appropriate conditions are met, sends messages to coach, coax, goad, and guide the player. Sometimes Send Messages makes infrastructure demands for a stadium, airport, or seaport, signaling to Set Valves to constrain zone demand until the demand is met. The demands themselves arise within Send Messages, constituting the only actual simulation taking place in Send Messages. By making such demands, Send Messages works to narrate and place obstacles of escalating difficulty in the player's path, forming a kind of dramatic arc. Send Messages does not respond to discrete simulation events. It simply reflects upon the simulation state and sends the player messages when a set of conditions are triggered. To strike the right balance of messaging frequency, and not inundate or starve the player of this organic narrative, each condition is checked on a different beat of a 64 period cycle of CityTime. (Population check is done more frequently, on every 4th beat.)

a cellular lattice become, through mimetic representation, a city. Just as a playmate might scaffold and support the transformation of a banana into a telephone, *SimCity's* artifice cues players to interpret its imagery, sounds, and behaviors as a city.

The dynamic cellular map, system dynamics, agents, budget, and messaging systems constitute the enactive representations of *SimCity*. These are the vibrant parts and pieces that players play with. Carefully constructed software illusions, their artifice is starkly evident when we look at what might appear to be bugs in the simulation code[49]—places where mechanism diverges from player belief.[50] For example, only one power plant needs to be connected to the power grid, but all power plants supply it with electricity.[51] Such divergences shed light on how illusion making works. The larger the gulf between interpretation and mechanism, the more clearly we can see how illusion is conjured.

---

[49] Are they bugs? What is a simulation bug, anyway? It's all artifice, after all (Foley 2015).

[50] These divergences also complicate Wardrip-Fruin's reading of *SimCity* as presenting a transparent representation of its inner machinery, which he calls the "SimCity effect" (Wardrip-Fruin 2009).

[51] Some examples where mechanism contradicts player belief:

- Only one power plant needs to be connected to the power grid. The other power plants can feed power the grid as long as one of them is connected.
- Fire and police stations don't really need to be connected to the transport network to function properly. As long as they are adjacent to one transport cell they function at full capacity.
- Commercial zones must be properly connected to the transport network in order to grow. But failed transport interconnect will not block the growth of residential and industrial zones. As long as they are adjacent to a single road or rail tile, these zones can grow.
- The seaport does not need to be placed near water. Even if placed inland, the citizenry will be satisfied and it will spawn a ship.

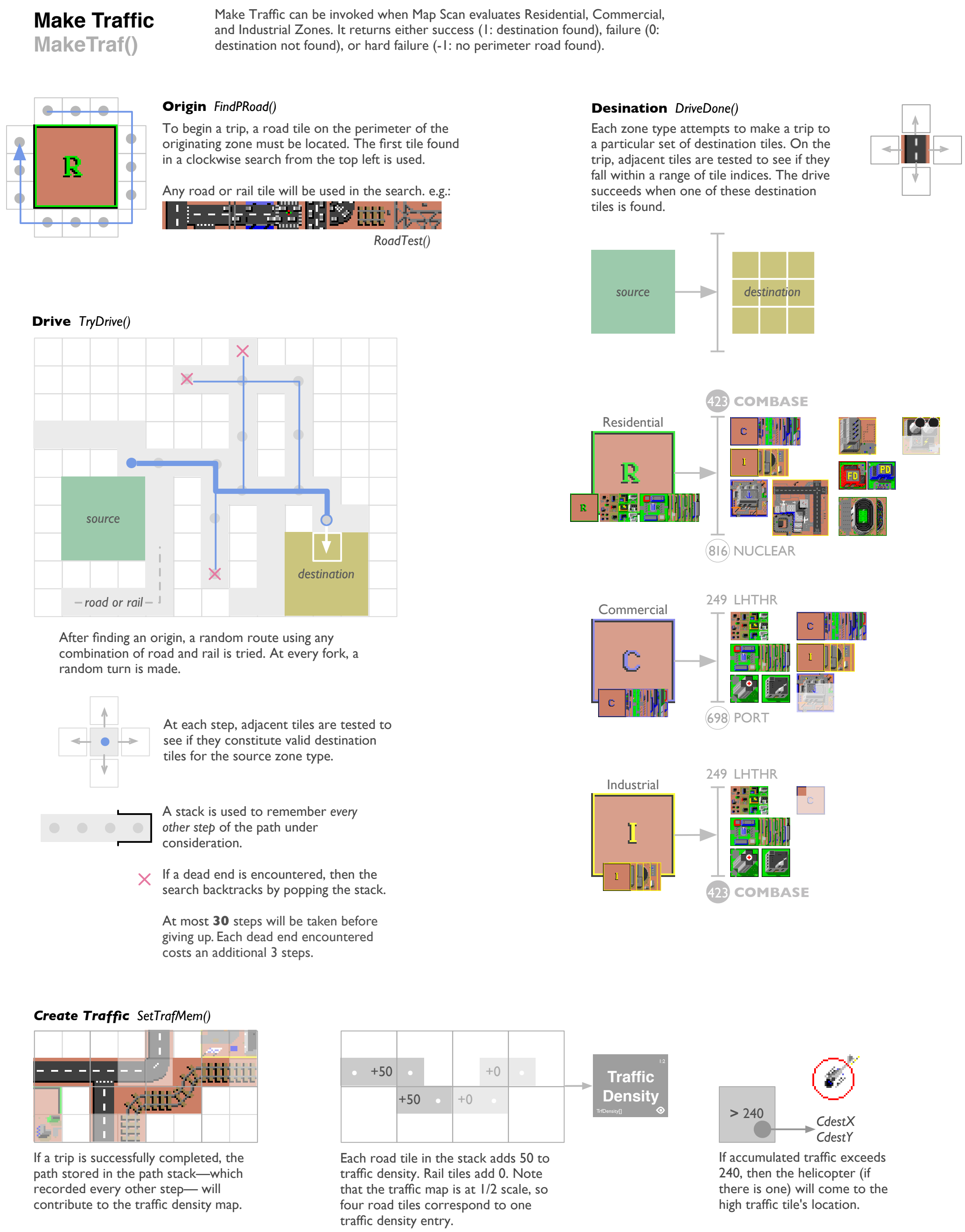


Figure 6.34: Make Traffic

The traffic system employs an illustrative legerdemain. Randomly generated trips, a stochastic process, is how traffic is simulated in software. But players don't see these sporadic trips.

Peering through frosted glass, they see only the blurred and smoothly animating veneer of traffic.[52] Through artful misdirection and abstraction, *SimCity* implies processes that aren't actually taking place.[53] It hides and it reveals; it is transparent and opaque. According to Wright,

> there are certain things we just cannot simulate on a computer, but on the other hand that people are very good at simulating in their heads. So we just take that part of the simulation and offload it from the computer into the player's head (Pearce 2001).

Hopkins describes this as "simulation by implication" (Hopkins 2013). The player co-constructs the simulated phenomena, a process Murray describes as "the active creation of belief" (Murray 1997). All of this highlights the extent to which we dine on the expectations we ourselves bring to the simulation table, and the artful setting of that table. We believe power plants should be connected to the power grid, and indeed the simulation supports that belief. Summoning our expectations about power grids, the simulation requires that the first plant be connected, and so we continue to do so for the 2nd and 3rd—even if it is not necessary. A seaport suggests the sea, and so we place it on the shore. The simulation is indifferent, and we, happily, remain in the dark, imaginatively filling in and actively creating belief. *SimCity* implies and we simulate.

[52] A single trip traverses the transit network by taking random turns. It succeeds if it finds a zone of a particular type, determined by the originating zone. Although road and rail look like different networks to the player, trips hop wily nilly between them, as Make Traffic treats all of these characters as constituting a single unified transit network. The combination of many individual random trips is presented as a veneer of animated traffic cells. This helps us to imagine that connectivity is important, but this is largely a facade, an approximation in multiple senses. First, it is stochastic approximation. Rather than plan a path to the destinations, like a GPS navigation system does, sporadic and randomly generated trips are used as an approximation. Second, the animated traffic players see is a veneer that misdirects players about what is actually transpiring. Residential and industrial zones don't actually care if their trips reach their destinations, only that they are adjacent to transit cells. Nonetheless, traffic between industrial and residential zones is interpreted by players as signifying a functional connection.

[53] Withholding information and detail catalyzes empathy and imagination, as Scott McCloud has pointed out (McCloud 1993). It also fosters a sense of autonomous behavior (Zhu 2009).

Miniature City as Play Set

How do *SimCity's* enactive representations scaffold play? We've seen how, through software artifice, the enactive representations of a city are built up: an urban landscape, full of buildings and infrastructure, tools, an economy, and agents. While there are many perspectives from which one could analyze *SimCity's* enactive representations, I now focus upon my overarching interest in design for play.[54]

Stylized

SimCity is separate from ordinary life; it is not a real city, but a simplified miniature. It is very much like a toy. Its operation is easy to see and grasp, it yields easily to the desires of players, and it signifies, through its stylized otherness, that it is an instrument of play.

Simplification and caricature make plain what is going on. Like a toy car, whose schematic abstraction foregrounds mechanism, inviting comprehension, power, and assimilation, the cellular lattice, iconic art, and scale of *SimCity* make its world easily graspable.[55] If schematization affords the internalization and appropriation of a material for play, as Sutton-Smith has argued, then *SimCity* is an ideal instrument for players to use as a toy, and turn to their own ends.[56]

As with any easily grasped miniature, players are granted a huge degree of power. The

[54]One could, for example, do an analysis of *SimCity's* politics (e.g. Sample 2011), design ideology (e.g. Kay 2007), or perhaps recover evidence of Wright's research sources in his simulation design. In addition to play design, there are a few other points I make. I point out the commitments to Wright's simulation influences, cellular automata and system dynamics, as well as lessons in the craft of illusion making and simulation design.

[55]Cellular automata schematize space, state, and causality. Valves, graphs, and problems clearly articulate dynamics and problems. Rich feedback and reportage offer insight into what the model is doing. As in a toy, you can clearly see what is happening, why, and by what mechanism.

I use the term *grasp's* multiple meanings, to signify both conceptual comprehensibility and tangible manipulability.

[56]As Sutton-Smith has argued, schematic stylization and clarity seeds the comfort, familiarity, and feeling of power that precedes appropriation and play (Sutton-Smith 1986).

player is more than a city planner—they have the power to bulldoze, zone, and build infrastructure wherever they like, as well as unleash disasters, and via "cheat," inject cash into the treasury.[57] And player actions are amplified, heightening the sense of expressivity, agency, and autonomy. Clarity and power foster self-directed activity.

Transforming the ordinary world into a stylized representation signals play and invites a playful attitude.[58] The schematic city is simultaneously real and imaginary, puffing up and deflating itself, enacting a paradox that sends the signal: this is play. Many of the signals that mark *SimCity* as play, and guide the player into a playful orientation, are intrinsic to its stylization, which distorts and recombines cities into miniaturized counterparts. This paradoxical quality helps *SimCity* to become everything from a frivolous diversion to an evocative philosophical object, a child's toy as well as a serious model.[59]

Stylization gives license to play. *SimCity's* familiar theme, stylization, and manipulability engender a feeling of safety that encourages play. Miniaturization puts players in command, buffering make believe from ordinary reality. Irreality feeds bacchanal, granting players the license to be silly, playful, and destructive. Rescued from responsibility, players are free to plunge their make believe citizens into catastrophe.[60] As Brian Sutton-Smith has pointed out,

[57]It is a well known but somewhat obscure—it isn't made plain—violation of *SimCity's* dominant ludic order (Gingold 2003). It is an inversion within an inversion. The documentation refers to it as embezzling, reinforcing its transgressive quality.

[58]Transformation requires a baseline, a ground against which the figure of inversion can stand. For something to appear different, exaggerated, inverted, or grotesque, we need to be familiar with what is normal (Sutton-Smith 1986). The represented materials of *SimCity,* an urban landscape, is familiar to most people.

[59]To those that might critique it for being a poor and unrealistic model, *SimCity* can respond that it is "only playing." Recall Sun's desire to use *SimCity* as a serious GIS tool. *SimCity* wields the power of mimesis while disavowing the responsibility of accuracy. The same problem came up for me when developing *Earth: A Primer*. I knew that the simulation was a schematic abstraction of geological processes, and the challenge lay in signaling that, yes, this is real and representational, except that it's not, and you shouldn't take it too seriously.

[60]Play is inconsequences and harmless, a safe space inviting experimentation and play. Destruction is bloodless and free of suffering, enabling players to indulge in make believe pandemonium. Destruction is enacted, simultaneously real and unreal, a duality that enables players to enjoy wreaking real havoc while absolving them, through unreality, of its responsibilities. On top of this, players can enjoy the absurd contradictoriness of this

play inverts, mocks, and destabilizes an order. Silliness can only unfold against a backdrop of seriousness; in order to invert something, there must first be an order to turn upside down (Sutton-Smith 1986). *SimCity* models and mocks real cities, but is not immune to its own playful inversions. Godzilla comes to destroy the simulated city, poking fun at the model itself—tearing down any pretensions of accuracy and seriousness. These acts of destruction are prankish. Not only can the player prank a city by bulldozing and visiting disasters upon it, but *SimCity* pranks the player by randomly unleashing disasters and problems upon them.[61]

Play delights in inversion, transformation, and stylization. A full scale city is turned into a miniature model. This scale model then undergoes further grotesque distortions, subjected to the destructive whims of players and monsters, as well as the unpredictable dynamics of an animate world.

Alive

As *SimCity's* software machinery clicks and whirs to life, the delightful illusion of a miniature world is produced for the player. *SimCity* feels alive. Much of *SimCity's* pleasure comes from watching the unfolding of a vivacious and mimetic world. As with a model train set, we are enchanted by moving cars, transforming neighborhoods, animated smokestacks, and burning fires.

Timing is everything. A sense of aliveness emerges not just from the quickly animating traffic and agents, but also from the many delays built into the simulation. This design is

enactment.

[61] The quality of recursive inversion distinguishes *SimCity,* and Maxis titles more broadly, from other kinds of simulations. Serious simulations as well as simulation games don't exhibit such multi-leveled inversion. In them, the world is transformed into a stylized model, and that is it. They don't go on to recursively mock themselves and their players.

computationally expedient, allowing *SimCity* to perform complex calculations on very simple computers. But delays also introduce important aesthetic payoffs. Delays and instabilities are mimetic, evoking the slow transformations and vivacious polyrhythms of urban activity. They are also dramatic, delighting the player with anticipation and surprise.[62]

SimCity is alive—oscillatory, surprising, and delightful—qualities that signify play. It is full of mystery and wonder. Like a playmate or bouncing ball, it amplifies, surprises, delights, and frustrates. The oscillating valves that govern citywide demand introduce an element of caprice and topsy turvy variety. Instabilities and variations also take the form of traffic, trains, and tornadoes. The world pulses with activity, vivacious and playful, pushing and pulling upon

---

[62]What gives *SimCity* its vivacity, responsiveness, and believability? The answer, in large part, stems from the many delays in *SimCity's* simulation. We have seen the importance of speed in creating an illusion of vivacious life: quick burning fires and fast moving agents. How then does slowness also contributes to a feeling of life?

The interleaving of processes (by the multitasking scheduler), the pipelining of data (e.g. across map layers and between cellular and system processes), smoothing of variables (e.g. in Set Valves), and using pseudorandom random numbers to time updates (e.g. updating fires) all produce delays. Delayed processing is not only computationally expedient, allowing *SimCity* to juggle many parallel processes with limited computational resources, but it produces mimetic delays and polyrhythms that produce a feeling of vivacity, heightening the sense of believability. Delays are critical to an illusion of inner life, autonomy, and aliveness (Shannon 1953a; Johnston and Thomas 1986/1995; Liversidge 1993; Tognazzini 1993; Zhu 2009; Horgan 2016). Delays and polyrhythms are also mimetic. Bruce Joffe, Wright's former neighbor who works as a city planning consultant, notes that the time delayed consequences of player action in *SimCity* is naturalistic. As in a real city, responsive change unfolds slowly, and activities transpire at multiple overlapping time scales (Joffe 2015).

Combining and staggering many time delayed processes, as *SimCity* does, creates a polyrhythm evocative of a city pulsing with activity and life. Interleaved and overlapping activity unfolds at different scales, from detailed train movements and billowing smoke, to the development and decay of zones that unfold over sim weeks, all the way to the yearly budget cycle. Activity at one scale or layer covers for a lack of motion in another, creating a sense of constant movement, scale, layered interest, and telescoping complexity. Broad action unfolds at a giant scale, slowly tracking the evolution of the overall city, while other, more finely detailed activity provides local embellishments and more immediate flourishes. For example, the temporally ongoing and spatially diffuse activity of Animate Tiles—updating drawbridges, fire, smoke, and traffic—sustains an illusionistic sense of life in the map even while Map Scan is asleep, inactive and waiting to run again. Agent simulation offers a similar degree of local embellishment, and together with detailed cellular activity, function as fine filigree and textural variation.

Delay techniques, furthermore, are dramatic. Making the player wait delays gratification and stirs curiosity. What will happen next? How will the city respond to my actions? In combination with feedback loops, delays produce instabilities, a behavior Forrester intentionally set out to model with system dynamics. These instabilities are surprising, intensifying player curiosity and delaying gratification. Surprise is also caused by the simulation's use of pseudorandom numbers, which further adds sprinkles and wrinkles of variety.

Simulation delays and instabilities lie very much at the heart of *SimCity's* feeling of aliveness, autonomy, and responsiveness. To summarize: delays are computationally expedient, mimetic, and dramatic.

the player.

The miniature city seems to act on its own. This illusion is established by obscuring its animating machinery, an act which invites us to perceive it as an autonomous system (Zhu 2009). We don't really see how traffic works, but its opacity and autonomy invites us to conjure a belief in miniature drivers stuck in tiny traffic jams. We place road tiles, and unseen architects and construction workers stitch them together into coherent road networks. And then, miraculously, traffic appears. We zone lots, and then—amazingly—developers, industrialists, and families build and inhabit apartment buildings, factories, and shopping malls. The world seems to be very much alive, inhabited by an unseen populace, pursuing its own agendas.[63]

Manipulable

SimCity is a world made of tiny parts you play with. Players are brought into contact with highly sensitized, sensorily rich, and animate representations. By transforming the real world into a stylized miniature, *SimCity* sets the stage for the player's transformative touch, to which it robustly responds. Causality unfolds with schematic clarity, heightening the sense of agency and power.

Like magnetic poetry or emoji, these are labile and expressive materials whose inherent

[63]Lifelike believability results from how *SimCity* sets and then fulfills expectations. The back of the box sums it up: "Lay roads and traffic moves. Supply electricity and smokestacks churn." More examples: train tracks cause trains to appear, the airport generates an airplane and helicopter, the helicopter seeks out and reports on traffic, drawbridges open for the ship, traffic manifests on roads, problems lead to complaints and poor approval ratings, and unhappy sims move out. Cause and effect. These examples fit into Murray's design principle of "scripting the interactor," deliberately calibrating player expectations so that they can then be fulfilled. Players must be coaxed into playing their part, so that the player and designer can meet one another on the field of play (Murray 1997). Coherent cause and effect evokes a powerful sense of agency and believability: what you make comes to life, and behaves as you expect. The city, in turn, reacts to itself in a plausible feedback loop. Many phenomena, for example the helicopter and Godzilla—autonomous agents that respond to the simulation itself—imbue the city with a self-awareness, conjuring a world which reacts to itself with seemingly lifelike, intelligent, and intensional activity. *SimCity* leverages our familiarity with cities, gracefully meeting and reshaping expectations we bring to it, from the simplistic (that police stations stop crime) to the fictional (Godzilla).

combinatorics entails a huge variety of possible configurations.[64] A granular world, in conjunction with the right level of abstraction, enables the same enactive representations to become San Francisco or Tokyo.[65]

Robustness encourages manipulation. *SimCity* doesn't break in the face of unexpected transformations. As in Budge's *Pinball Construction Set,* the awkward configurations that inevitably arise are handled with grace.[66] The ship might find itself stuck at the periphery of the water, its navigation logic unable to recover. The train lacks the smarts to avoid crashing into other agents, such as Godzilla. Such impasses are gracefully handled through explosions. The ship runs aground, and Godzilla destroys the train.[67] *SimCity* is highly robust, fulfilling its

---

[64] Like magnetic poetry or emoji, powerful recombinant meanings emerge from the juxtapositions of evocative elements. The proper set of components, when recombined via the ingenuity of human play and interpretation, lead to ingenious juxtapositions and novel interpretations. Within Wright's oeuvre, *The Sims*, with its recombinant of everyday matter—people, needs, relationships, and domestic furnishings—gives rise to the most diverse and highly charged human situations and dramas. In this, *The Sims* is similar to *Grand Theft Auto* and *Katamari Damacy*, meaning recombination and generation machines that permute ordinary things with ordinary meanings—cars, cats, coffeepots, people, and relationships—into creative, nonsensical, whimsical, transgressive, and unlikely configurations.

[65] *SimCity* has combinatorics on its side. Cities are made out of reusable components. The urban tapestry is woven from 956 different characters that recombine into a practically endlessly set of layouts. But variety and permutation alone aren't enough. Components themselves strike the proper balance between specificity and abstraction. If the individual buildings and roads were more resolved and specific, our imaginations would have a harder time flexibly assigning them to Tokyo as well as Rio de Janeiro. If they were too abstract, it would be harder to see the cellular lattice as a city. Although *SimCity* leaves behind the highly evocative yet minimalist representational abstraction of traditional cellular automata, it retains some of their representational generality. Stories are still spun around the cellular automata campfire, but those stories are focused upon a more limited domain, the urban landscape. There is also the tremendous power of the discursive practices that surround these model cities. Their names, whether in the included scenarios and their descriptions, or as player given names, stimulate the imagination.

The granularity of these materials matters a great deal. Magnetic poetry, for example, would be quite a different experience if the recombinable elements were letters instead of well chosen words. It would no longer be magnetic poetry, but magnetic words.

The expressive range, however, is not as broad as a title like *The Sims*, with its modeling of the familiar world of western people and the mundane details of their lives and personalities. The rich collection of captured stories that players tell with *The Sims* are a testament to its recombinant, expressive, and evocative powers.

[66] Budge's *Pinball Construction Set*, for example, put a general purpose polygon editor into the hands of players, allowing them to build arbitrarily shaped courses. But polygons don't make logical sense when their outlines self-intersect. Given that it was "really hard to prevent people from making crazy stuff," Budge "just let people make crazy stuff," and invented algorithms that robustly dealt with the such poorly formed polygons (Budge 2013).

[67] The drawbridge opens and closes despite the massive number of possible layouts it might find itself in, and

half of an unwritten contract between program and player: you can do whatever you like, and I promise that I will keep working coherently. This unwritten contract underpins a feeling of safety, encouraging play.

Hybrid agency amplifies player manipulations. *SimCity* is more than a passive material for expression, and goes far beyond something like *MacPaint*. Like a genie, it scaffolds, amplifies, and frustrates the player's transformations. The world is alive, animate, and responsive, interpreting and elaborating upon your actions. *SimCity* saves you from the painstaking labor of assembling a world out of delicate parts. From sketchy directions—a winding path for a road, some zones—a detailed urban tapestry unfurls.[68] The player is amplified, their agency

---

this logic works because the drawbridge makes few assumptions about what surrounds it and what state it might be in. And the fact that multicellular structures such as roads and buildings can be damaged one character at a time, at a cellular granularity, means that the simulator must be able to deal with all of the weird resulting configurations. That so much of the simulator logic operates in cellular automaton fashion—a single transformation function applied homogeneously throughout space—means that these situations are gracefully accounted for. For instance Animate Tiles, as we saw, updates the map one cell at a time, blind to damaged wholes. Since multicellular wholes like roads and buildings are contingent phenomena, they happily proceed even if some cells drop out of the picture. Like the Internet, which can keep working even if particular nodes burn out, the power and transportation grids, composed of contingent cellular networks, robustly adapt to the loss of individual cells.

The simulator itself robustly adapts to any number of buildings and networks. Any number of structures can be embedded within the cellular representation, so memory isn't an issue. (It may be possible to contrive a power grid that causes the Power Scan algorithm to fail to properly light up the entire grid.) The multitasking architecture that juggles so many parallel simulation processes ensures that even as the simulation gets busier it does not stutter or stall.

Failures can unfold within the fictional world, as when a power outage occurs or there is too much pollution. As we saw with the explosions resulting from agent collisions, robust failure handling can do more than proceed with grace, it can entertain. Wright's design philosophy, as we saw in chapter 5, is to indulge such failure states. This is most clearly seen in the disasters and destruction players can wreak upon cities. Failure is fascinating, as the scenarios, which often put failing cities into the hands of players, attest to. It is more fun for a city stricken with problems to be attacked by Godzilla (he arrives in response to high pollution) than for the mayor to be kicked out of office and play to end. The key is ensuring an enjoyable mess. A partially destroyed city is fun, and as Wright learned during development, players enjoy wreaking havoc, especially if the city responds dramatically and appropriately. Destruction and failure are also teaching and engagement opportunities, revealing the city to be alive and what makes it tick. This often motivates further play, to repair the damage (Rouse 2001). *The Sims* is perhaps one of the best examples of such design. Sims who fail to get to a bathroom pee themselves, exhausted sims collapse on the floor, and sims trapped in malicious player-designed enclosures eventually die. Failure is fun, making it not failure, but something else.

[68] *SimCity* performs a lot of the labor in realizing a detailed model city. It is much easier, even, than assembling a model train set out of prebuilt parts. Players put down a few things, a zone here, a road there, and a world begins to wake. Sims build homes, drive cars, smokestacks billow, traffic snarls, pollution spreads, industry grows, and infrastructure decays. Connect Tile keeps roads, train tracks, and power lines connected, freeing the player from

hybridized with that of the machine.[69] Much of this amplification stems from the world's autonomy. Wright compares *SimCity* to gardening:

> So you're kind of tilling the soil, and fertilizing it, and then things pop up and they surprise you, and occasionally you have to go in and weed the garden, and then you maybe think about expanding it, and so on. So the actual process of playing SimCity is really closer to gardening (Pearce 2001).

The most striking example of such hybrid agency is in autonomous zone development. Players allocate residential, commercial, and industrial zones, but it is *SimCity* which determines the pace and course of their development and decay. Hybrid agency means that *SimCity* amplifies and challenges player agency. Hybrid agency also helps *SimCity to* establish common ground with the player, scaffolding a synchronized understanding of its enactive representations. Seeing a residential zone blossom into a hospital or apartment building helps players to learn the meanings of things.

A manipulable, stylized, and animate world leads to the pleasures of problem solving. The combination of agency, clear causality, and the push back of an autonomous world leads to tactics and strategy—figuring out how to address high traffic, mitigate pollution, grow a neighborhood, or advance an intention. Strategic questions arise, such as whether to invest in short or long term solutions: road versus rail, coal versus nuclear.

SimCity's representations are tangible.[70] Using the paint program style tools, players reach

---

worrying about such details, and producing a powerful feeling of agency.

[69]Cellular automata contain within them the seeds of this hybrid agency. The practice of cellular automata was transformed by the introduction of computer simulation, as we saw in chapter 4. Unleashed from the static world of mathematical proofs and thought experiments, cellular automata sprang to life, surprising their makers, and prompting research into them as models of emergence. While their empirical study as surprising systems had been done by hand, as with Conway's *Game of Life,* it was automatic computation that caused the creation, study, and play of cellular automata as autonomous systems to flourish. The combination of machine agency, which breathed life into cellular automata as surprising, emergent, and autonomous systems, with human agency gave rise to a hybrid agency combining both machine automaticity and human intervention.

[70]As Wright's work evolved, and moved to increasingly sophisticated hardware, players gained increasingly direct perceptual access to simulation innards. *Raid on Bungeling Bay's* antagonist was animated by a simulation

in and touch the landscape. Similar to how Conway and his collaborators were able to directly handle the physical counters and shells that constituted the *Game of Life* universe, *SimCity* players are granted direct access to the underlying cellular map that constitutes a city.[71] When we peer through *SimCity's* Edit Window we see the naked Primary Map data dressed in character art. Where we see cells as graphical representations, algorithms see that same numeric cell data as enactive encodings. Not only are these representations sensually concrete, but they constitute the shared grounding of player, simulation, and software developer.[72]

Guide and Inspire Play

SimCity strikes a balance between open-ended and guided play. Like an older playmate, it scaffolds play through guidance and inspiration. It does this through scenarios, roles, calls to action, narration, and suspended judgement. Open-ended autonomy can be overwhelming,

---

of resource harvesting, transport, and consumption. In his postmortem on *Raid*, Wright noted that the simulation was overly opaque, so players were never able to see and understand what it was doing. He suggests that the addition of some moving oil barrel icons would have gone a long way towards improving simulation legibility (Wright 2011). Looking at the trajectory of Wright's work, from *Raid* to the C64 *SimCity* prototype, and even beyond into *SimCity 2000*, we see not just increasingly complex simulations, but more transparent views into them. On the C64, *SimCity* offers graphs and maps, but none of the real time contextual messaging and RCI demand indicators. *SimCity 2000* went even farther, offering a city newspaper, and a tool for querying map objects.

[71] The player and simulation subsystems see and touch the same representation. A feeling of directness is intrinsic to the cellular automata tradition. The visual manifestations of cellular automata are practically indistinguishable from their state encoding. When Conway gazed at or held a *Game of Life* token he perceived and touched the world representation itself, and in digital versions of the *Life* black and white colors map one to one to dead and alive states. Internal representation and external presentation, if not one and the same, are tightly bound to one another.

The discrete nature of cellular automata mean that they are easy to perceive. States, transformations, places, boundaries, parts, and wholes are clearly delineated and schematic, easily perceived, identified, and narrated. Causality is easily seen. Since cells can only interact with their neighbors, spatial locality is intrinsic to the simulation. Cause and effect lie near one another, and transformations ripple across the face of the world. All of these qualities, when combined with a strong sensitivity to rule sets and states, mean that cellular automata are highly responsive to player intervention. They are a highly direct simulation material.

[72] I've mentioned earlier how development labor was divided between the front end user interface and the back end simulation, which means that this enactive representation functions as a shared grounding between many actors: player, simulation subsystems, Will Wright, and *SimCity's* multiple interface developers. The map, as a software structure, functioned as a social-technical shared grounding between Wright and Strobel.

burdening players with the existential dilemma of construing meaning and purpose in their activity. Guidance counterbalances the tremendous autonomy offered to players, leading to a feeling of safety. The player is welcomed into a world that simultaneously teaches them how to play, gently suggests activities, and provides a dramatic experience.

First, players must learn how to interpret and play with *SimCity's* dense and complex urban system. The volume of concurrent activity means that it is difficult to attend to and comprehend, especially for new players. *SimCity* helps with this by scaffolding perception and play. By allowing players to build a city from scratch, there is a gradual buildup of complexity, which facilitates assimilation. Even so, there is much to learn and understand. The message system,

---

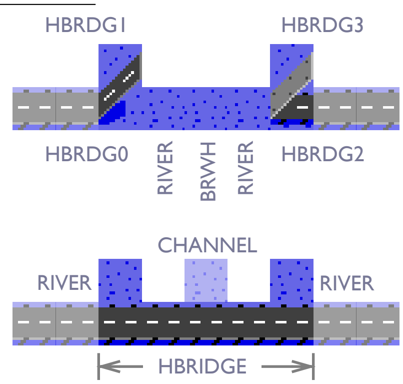


Figure 6.35: Drawbridge.

The characters which populate map cells are neither just art nor state information, but simultaneously both. Internal representation and external presentation are tightly coupled. The drawbridge, which opens when the ship is nearby, illustrates the implications of this tight coupling. Bridge cells will transform into open drawbridges as the ship approaches. (As long as there is enough room; a horizontal drawbridge requires a CHANNEL cell above it, which marks the center of a big river. In theory a drawbridge can manifest anywhere along a bridge, retaining no coherent identity across openings. Ships only navigate through CHANNEL, BRWH, and BRHV cells.) A drawbridge opens and closes by toggling between two multicellular character configurations. Although RIVER and BRWH look the same, they mean different things to the simulation. BRWH marks a cell as navigable by the ship, and bookmarks the open drawbridge for the simulation. When Map Scan updates BRWH, which is how it knows where the drawbridge is, it closes the drawbridge if a pseudorandom number comes up and the ship has passed (e.g. is no longer close). So while RIVER and BRWH look the same to the player, their meaning is quite different to the simulation. An inverse case, where characters look the same to the simulation but different to players, can be seen in the character animations. Different animation frames, e.g. traffic and smoke, are treated as functionally equivalent by the simulation, while the player perceives these as different animation frames. These examples underline how the same representation serves two different participants, the player and the simulation. They both have direct access to the same data, and must accommodate one another. Cell characters are neither pure art nor state, but both, conjoined.

by focusing attention on phenomena, and narrating it, teaches players how to interpret and react to the world. Causality is also explained passively, through the secondary data maps and informational feedback. One clearly sees, for instance, that Land Value and Pollution are opposed. (More sophisticated or subtle simulation rules would be harder to learn in this way.)

Descriptions and templates set the scene and frame player expectations, establishing a role for the player. Scenarios are the best example of this, evoking representations of specific places with their names (San Francisco, Detroit, Tokyo, Bern) and layouts.[73] But scenarios and their descriptions do more than evoke particular cities; their descriptions and behaviors challenge players to solve particular problems. Players are enjoined to fix Detroit's crime, Bern's traffic, or rebuild in the wake of San Francisco's 1906 earthquake or Tokyo's Godzilla attack.[74] Like the *Pinball Construction Set* boards, or the random simple starter creature bodies of the *Spore Creature Creator*, the scenarios are simultaneously scripts, guides, and calls to action that eliminate the creative challenge of the blank page (Smith et al. 1982). *SimCity* never stages a completely blank starting point. Play always begins within a scenario that challenges, pushes, or inspires—even if that scenario is an empty landscape inviting settlement. Scenarios are a balm for the existential problems posed by open ended play materials. The reassurance of a script, and the calm of a unsettled landscape, create a relaxed field that is conducive to play.[75]

[73]Scenario names help set the stage for play, calling forth particular cities in the player's imagination. Players also partake in this narrative act. One of the first acts of a new mayor is to name their city, setting their own scene, imagining what their city is, might become, or just being silly; but even a senseless playful name becomes invested with meaning through play. *SimCity 2000* elaborates this feature, allowing players to name and place billboards across the landscape.

[74]The Commodore 64 version of *SimCity* calls attention to where and how the disasters are role-played; the manual includes instructions for manually invoking disasters, and warns players not to save their scenario game over the built in scenarios—which, on the C64 version, are just saved cities. This, presumably, is because the scenarios were a later design addition, and Maxis never revised the earlier C64 prototype, aside from updating its title screen (chapter 5).

[75]*SimCity* begins in a state of calm, offering a choice between starting a new city or a playing a scenario. New

Players are called to action by instabilities, problems, and feedback. Disasters are the most salient calls to action thrust upon players, but many other problems push back, prompting and scripting involvement. Poor approval ratings, citizen problem rankings, unintended consequences such as traffic, pollution, and crime, as well as space constraints, budget constraints, and citizen demands, are many of the ways in which *SimCity* pushes back upon players, calling them to action, and disrupting their plans. Fluctuating zone demand invites players to participate in a push and pull dynamic, addressing an over- and under- supply that will lead, in turn, to a new imbalance. As in *Tetris*, players find themselves building up incomplete patterns, from zone supply imbalances to unfinished urban layouts, that they then strive to complete. The result is ongoing improvisational play. Even if stability is achieved in one dimension (e.g. a complete urban pattern or demand balance), the balance in another dimension will be disturbed.

Goals are not imposed upon players. The stylized problems, scenarios, feedback, and fictional world are materials that ask, suggest, and cajole the player but never demand or require. You are not kicked out of office for not complying with *SimCity*, and you do not fail to win by not following a particular script. Judgement is suspended. Instead, feedback and guidance function as materials to appropriate.[76]

---

cities begin in a landscape bereft of human settlement, with no apparent pressures or demands. While developed cities are highly stimulating, complex, and demanding affairs, players slowly build up these cities themselves, journeying from a state of calm simplicity to complex chaos. Scenarios, on the other hand, drop players into cities in the grip of severe problems. While not calm, scenarios familiarize players with the potentialities of *SimCity*, and function like *Pinball Construction Set's* demo boards, or the instructions that come with *Lego*, prompting and cueing activity.

[76]While a game's interpretive commentary is evaluative towards some goal, oriented around concerns such as winning, losing, and progress, *SimCity's* reflective commentary is more open ended. Your approval rating might be low, but contra Brøderbund's request and perplexity at Wright's design, this doesn't constitute losing and being kicked out of office. The meaning of this reflective commentary isn't closed, as is the case with games, but open to the player's creative use. This largely has to do with specificity and enforcement. Play is never demarcated as "failure" or "success," and play is never halted, as games typically do when one has "won" or "lost." Normative expectations, of course, follow mayoral approval ratings, but it is up to the player whether they wish to respect those norms. The player does not face a challenging existential vacuum of interpretation, such as

What does my play mean?, How am I doing?, or *What is going on?*, but a concrete prompt harboring a specific, yet open ended meaning. Rather than dictating to players what their play means, *SimCity* offers an evocative yet flexible starting point. A goldilocks zone is achieved, somewhere between dictating an agenda and leaving the player with nothing to go on. We find this basic pattern repeating throughout *SimCity*. *SimCity's* many rich feedback channels—maps, messages, polls, charts—supply the raw materials out of which players formulate and measure their own agendas. Maxis titles, according to Wright, force players

> to determine the goals. So when you start SimCity, one of the most interesting things that happens is that you have to decide "What do I want to make? Do I want to make the biggest possible city, or the city with the happiest residents, or the most parks, or the lowest crime?" Every time you have to idealize in your head, "What does the ideal city mean to me?" It requires a bit more motivated player. What that buys you in a sense is more replayability because we're not enforcing any strict goal on you. We could have said, "Get your city to 10,000 people in ten years or you lose." And you would always have to play it that way. And there would be strategies to get there, and people would figure out the strategies, and that would be that. By leaving it more open-ended, people can play the game a lot of different ways. And that's where it's become more like a toy (Rouse 2001).

Chapter 7

City Building Education

> [Design based learning] DBL teaching is rooted in the idea of a **community**, because it is composed of individual parts that fit together. A community is also immediately understandable, unthreatening (everyone has an opinion about it) and flexibly open-ended. Depending on grade level, the community could be a house, a city, a village, a business, a government, a civilization, an outpost, or a Utopia.
>
> Students learn about the community's systems and organizations by figuring out what the community needs to function properly. What individual parts does it contain? What does each part need, and how does it interact with the others? What happens if there's a breakdown? Who can help, and how? ("How Does Design-Based Learning Work?")[1]

Introduction

Play has a complex relationship to what is not play. Depending on who you ask, *SimCity,* the software toy, is either a frivolous diversion or an earnest model—and sometimes both. Right from the start, *SimCity* had appeal as an educational tool, a quality that Maxis tried to capitalize on.[2] According to Braun, "It was never our intention to go into the education market, but the

[1] "How Does Design-Based Learning Work?" web page (*https://www.cpp.edu/~dnelson/methodology/index.html*). Accessed in 2015. Web page updated in 2009.

[2] Evidence of this can be seen in the many discussions about education, semiotics, simulation, and so on that *SimCity* figures into as an evocative object. Sun Microsystems, as we have seen, thought to use *SimCity*

education market came to us and said: ‘This is what we need if you’re gonna work with us.’ ” What the education market wanted was teacher’s guides that translated and adapted *SimCity* for classroom use. It didn’t hurt that Brøderbund, Maxis’s publishing partner, was deep into the then hot educational software market, and that along with the investment Maxis received from venture capitalists in 1992, came a hunger for aggressive growth into new markets.[3] Wright, of course, was busy making titles like *SimEarth* and *SimAnt* for an uncertain market. Maybe that market was education?

But in order to sell into the education market, *SimCity* would need to undergo some kind of translation.[4] What began, in 1989, as a simple pamphlet about *SimCity* for teachers became, with the 1992 infusion of venture capital, an earnest effort to meet the desires of educators. Claire Curtin, an experienced educational software producer, was hired away from Brøderbund, and became Maxis’s educational project manager. Curtin, who would later co-design *The Sims* with Wright and Roxana Wolosenko, recruited teachers into an advisory board that reviewed teacher guides, and hired teachers to write those guides. One of the teachers Curtin hired was Doreen Nelson, a brilliant and innovative educator who had developed a pedagogy called City Building Education, in which students collaboratively built cities out of craft materials and role play. Nelson become a regular visitor to Maxis, and Curtin made some trips to Los Angeles to

as a way into the Geographic Information System market (chapter 5). The *SimCity* manual, in introducing and explaining itself—a new kind of software that didn’t belong to any established genre—described *SimCity* as “the first of a new type of entertainment/education software, called SYSTEM SIMULATIONS.”

[3]The inherent affinity between *SimCity* and education was noted in the manual Bremer had written, which called it a hybrid “entertainment/education” title. But *SimCity* cohered with the education world beyond even the subject and treatment. Brøderbund, *SimCity’s* distributor, was deeply focused on the education market, which created an immediate linkage between Maxis and schools. Brøderbund, after all, published some of the most famous and successful titles in schools: *Where in The World is Carmen San Diego?* and *Print Shop* (Braun 2015). The educational software market in the 1980’s was, unlike now, a vibrant business dominated by players such as MECC (*Oregon Trail, Lemonade Stand*), Brøderbund (*Carmen Sandiego, Print Shop*), and The Learning Company (*Reader Rabbit, Rocky’s Boots*) (Shuler 2012).

[4]Translation, in this sense, into a form where *SimCity* would fulfill the agenda of teachers, principals, and school districts. I use this term in the sense that Latour does (Latour 1987).

see City Building in action, where she found the experience of "watching a classroom actually go through a couple of days worth of creation" to be "very inspiring. … I will never forget that experience" (Curtin 2015; Nelson 2015).[5]

Figure 7.1: Photographs of City Building Education in action. Photographs from Compton, California courtesy of Doreen Nelson.

[5]This translation took the form of a short teacher's guide, a pamphlet, really, written by Michael Bremer, and published by Maxis in 1989—the same year *SimCity* was released, explaining the limitations and applications of *SimCity*, and offering curricular questions and scripts. Within a few years, Maxis became more serious about tackling the education market, and hired Claire Curtin, in 1992, as their first educational product manager, charging her with finding ways to package *SimCity*, *SimEarth*, and *SimAnt* for the school market. Prior to joining Maxis, Curtin had been the senior producer of Brøderbund's hit educational franchise, *Where In The World Is Carmen Sandiego?*, a job she had started in 1988, immediately after finishing graduate studies at NYU's Educational Communication and Technology program, where she had studied with the noted education technology researcher Roy Pea. Over the course of her career at Maxis, Curtin shifted roles and projects, a result of Maxis's fickle focus and its inability to produce hits beyond *SimCity* (chapter 5). Later, when Maxis defocused on a hard to reach education market, Curtin would go on to co-design or co-produce the kids' titles *SimTown* (1995) and *SimSafari* (1998). Curtin collaborated closely with Roxana ("Roxy") Wolosenko, and after Maxis decided not to do any more kid specific titles, the two of them were shifted to Wright's "Dollhouse" project—a title that was not spoken out loud due to its gender connotations—where they were instrumental, as Wright's co-designers, in evolving the design focus away from time management and towards people and interactions inspired by everyday life. It is this more human centric vision of Dollhouse that eventually saw release as *The Sims*, which became, at long last, the second commercially successful Sim title (Curtin 2015).

As educational product manager, Curtin was charged with supplying Maxis's sales force with the curricular materials needed to compete with titles from established players like MECC, The Learning Company, and Brøderbund. She contracted curriculum writers, and recruited California teachers for a teachers' advisory board, who reviewed and field tested guides. In a sense, her job was to translate Maxis products into a form that was appealing to the school market. As part of this effort, Braun introduced Curtin to a teacher based in southern California named Doreen Nelson, who would not only go on to pen teachers' guides for multiple Maxis products in collaboration with Bremer and Curtin, but became a regular visitor to the Maxis offices. Curtin recalls Braun saying "We should work with this woman. She's really smart and she's got good ideas about cities." Nelson, according to Curtin, was "another wonderfully smart person" who circulated through Maxis (Curtin 2015). Nobody I spoke to can quite recall how Nelson initially got in touch with Maxis.

Curtin introduced me to Nelson's work when we were collaborating on *Spore.*

This chapter is another case study about city play, but this time the cities are made of construction paper and cardboard, not computer software. City Building Education is a beautiful counterpoint to *SimCity*. They both involve cities, design, planning, construction, disasters, problem solving, and simulation; but one is made of tangible craft materials, multiple human players, and flexible rules, while the other is designed as a solitary experience that runs on a computer rigidly following a software program. The challenge in studying Nelson's methodology is that it spans her career, and is a highly socially situated affair, making it difficult to summarize. (It now goes under the banner of "Doreen Nelson's method of Design-Based Learning," or DBL.) To circumvent these difficulties, I first sketch an account of how City Building evolved, and then describe an early and prototypical City Building activity: the Purium Instant City.[6]

The deep resonances of Nelson's City Building Education are felt not just in her working with Maxis, but in her collaboration with the Vivarium project, a joint experiment by The Los Angeles Open School and Apple Computer.[7] Computers, constructionism, experiential learning, simulation, and complex systems thinking were brought together under the umbrella of

[6] My account of Nelson's teaching methodology and biographical details are drawn from her publications, interviews and conversations with her, materials and ephemera she shared with me (such as: her curriculum vitae, newsletter clippings, emails), various publications, and two days I spent with Nelson at a professional development institute for teachers on her Design Based Learning methodology hosted at Art Center in Pasadena. Although the title "City Building" wouldn't come about until a Smithsonian workshop in the early 1970's, and "Design Based Learning" until later, I use these labels interchangeably, favoring "City Building," the title of Nelson's 1982 book. (Nelson also uses City Building Education Program and its acronym, CBEP, in (Nelson 1974a).) Nelson now oversees a graduate degree in education at California State Polytechnic University at Pomona.

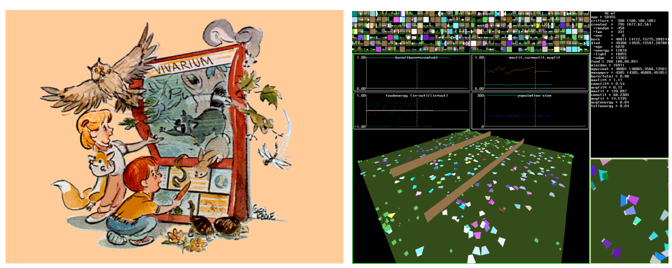
[7] City Building entered the orbit of the Vivarium program via Rachel Strickland, a filmmaker and architect employed by Apple on the Vivarium project. Strickland had been working with Kay since Atari Research Labs, and now, as a researcher, videographer, and designer at the Open School, sought a method for teaching architecture to kids. She looked for techniques within the library of the School of Architecture in Los Angeles (SCI-Arc), an experimental and well regarded architecture school at which Nelson had taught, where a librarian introduced Strickland to Nelson and her work (Strickland 2014; Nelson 2015). Nelson's City Building was a natural fit for the Vivarium project; City Building and the Open School Vivarium project were both committed to creativity as a means for learning how to live in a world of complex systems—from natural ecology to the urban world.

an idealistic educational experiment.[8]

[8]The Open School was an ideal site for Apple's Vivarium experiment. The school was founded in 1977, one of the public magnet schools created to integrate the Los Angeles Unified School District. From the start, the Open School was predicated on parental involvement, experiential and constructivist learning, and educational experimentation. Apple donated almost 200 computers, which were integrated directly into classrooms, and placed within custom desks, rather than being relegated to a separate computer lab. Ann Marion liaised between Apple and the Los Angeles school board, administered the program's two million dollar budget, and directed the research project and its staff members, graduate students, and teachers. The project ran from 1985-1992 ("The Los Angeles Open Charter School"; "School History"; Ann Marion C.V.).

The Vivarium project was conceived of by Ann Marion while working at Atari Research under Atari's chief scientist Alan Kay. Kay had moved to Atari from Xerox PARC in the early 1980's, where he directed a research lab whose far thinking agenda sought to design products that would only become practical many years into the future. The Atari research lab included notable researchers in human machine interaction such as Brenda Laurel and Susan Brennan (whose work we encountered in chapter 2). Marion's idea for what Yaeger describes as "the ecology-in-a-computer concept" emerged within the context of research into interactive fantasy environments, and the simulation of believable cartoon characters, such as Bugs Bunny and Elmer Fudd. (Warner Communications, whose subsidiary Warner Bros. owned Bugs and Elmer, had bought Atari in 1976.) Initially, Kay had trouble understanding Marion's project, which he has described as a simulation world about "ecological and social communication." Marion focused on an aquarium in which animated fish interacted, chased and ate one another, and emoted. The Atari lab was not long lived, however, and was shut down in the wake of a collapsing video game market (Brand 1987; "Vivarium History" Yaeger; Bell and Kennedy 2000; Wikipedia, Atari, 2016).

A couple years later Kay and Marion resurfaced at Apple Computer, where the Vivarium concept took on new life. Marion became a research director within the Advanced Technology Group, and Kay an Apple Fellow, an influential position from which he promoted the Vivarium concept into something much larger. Marion's vivarium project, under Kay's aegis, grew both institutionally and conceptually.



At left, Glen Keane's concept sketch for Vivarium (Source: Larry Yaeger). At right, image from Yaeger's *Polyworld*, one of the many diverse projects under the Vivarium umbrella. (Source: Yaeger 1994). The gulf as well as the connections between these two images is striking.

A vivarium implies ecology (community); independent agents—living creatures; observability—a glass enclosed and transparent environment; power—pets, a miniature world. It is a complete, bounded, living world. This evocative network of ideas became a meeting point for diverse institutions, agendas, and research projects. The Vivarium umbrella was flexible enough to house a broad range of interests, from artificially intelligent assistants to educating children. Vivarium's advisory board included one of Disney's nine old men of animation, Frank Thomas, human powered flight engineer Paul McCready, artificial intelligence pioneer Marvin Minsky, biologist Richard Dawkins, the novelist Douglas Adams, and Koko the gorilla ("Vivarium History" Yaeger). The Media Lab, with funding from Apple Computer, contributed research on artificial intelligence, computer animation, and interaction design. Evans & Sutherland—a computer graphics company founded by Kay's doctoral advisor Ivan Sutherland—collaborated, along with former Disney animator Glen Keane, and experimented with military grade computer graphics hardware (Brand 1987). Artificial life and artificial intelligence research Larry Yaeger built the noted artificial life simulation *Polyworld* (Yaeger 1994; Levy 1992).

Figure 7.2: Photographs from the Open School in Los Angeles. The source is Alan Kay's 1991 Scientific American article "Computers, Networks and Education" (Kay 1991).[9]

The above images—of children in gardens, real and metaphorical—capture the Vivarium project's high minded agenda: incorporating computation into a learning environment that would prepare children (and future adults) to understand, design, and inhabit complex biological, ecological, and urban systems.[10] An overarching goal of Vivarium, in Alan Kay's words, was to render invisible "contexts visible, make them objects of discourse and make them explicitly reshapable and inventable." Children would do so by creating "whole worlds," which

[9]In the images from the Open School you can see, in the photograph at left, Nelson's city building in action. Third graders built the city "after months of planning," according to Kay, and "the computers helped the students simulate the formation of smog in their city." Center photographs show third graders tending to an urban garden they designed "after creating and debating many models, often with the help of their computers." At right, the special desks designed to unobtrusively house the Macintoshes donated by Apple.

[10]According to Alan Kay, who championed the project, "much of the learning that will go on in the future will necessarily be concerned with complexity." But, echoing Forrester, Kay believed that "many apparently simple situations are actually complex," and the "contexts that give meaning and limitation to our various knowledges have been all but invisible. To make contexts visible, make them objects of discourse and make them explicitly reshapable and inventable are strong aspirations" (Kay 1991).

Why, Kay asks—echoing Papert, should we teach science, math, and music as spectators "when the children can more happily (and to better effect) actually create whole worlds?" (Papert 1980; Kay 1991). A transformation of learning environments and modes of knowing would lead not just to a newfound understanding of the complex worlds we inhabit, but an ability to reshape and recreate them. This agenda bears a striking resemblance to Nelson's, whose pedagogy enacted her belief that in order to reshape the city one must first come to know it as a whole, recreating it in simulation. And in order to transform one's community, one must first transform oneself.

was found to be both enjoyable and effective (Kay 1991).[11]

Following in the footsteps of Wright, Forrester, and multiple generations of cellular automata makers, children built simulations in order to come to grips with their world.[12] At the Open School, they did so with cardboard as well as computers. Forrester had addressed the urgent problems of humanity through an innovative hybrid of servomechanisms and computation. Wright, like one of Kay's ideal child learners, indulged his self-directed learning through constructionist play. But of these simulation makers it is Nelson who is most adamantly focused on empowering people—children, future adults—to reshape themselves and their world through art, imagination, and collaborative play.

This chapter explores Doreen Nelson's City Building Education pedagogy as play. We will see how Nelson, working from a very different set of influences than Wright, built a very different kind of city simulation grounded not in computation, but in role play and model making. This chapter culminates in a discussion of Nelson's Purium instant city game, which I analyze through the lens of play. To fully appreciate how Nelson's pedagogical techniques scaffold

---

[11]This echoes the thinking of Kay's role model, Papert (Papert 1980), but also Wright, Forrester, and the many players of cellular automata.

[12]Vivarium also included projects about education and programming languages, two very consistent themes across Kay's career. Kay had been inspired by Seymour Papert to work with children, and many of his influential projects—SmallTalk, Dynabook, KiddieComp—were envisioned and executed with children in mind. (If the design of new computerized media projects new types of users that do not yet exist (Woolgar 1991), then what better audience to test designs on than children, curious and adaptable—the adults of the future?) A "child-oriented" programming language called *Playground* was devised, which readily accommodated the central tenets of Marion's original Vivarium concept, as well as the expanded educational and programming interests Kay brought to it.

> Playground is inspired by our intuition that biology provides a good metaphor for understanding complex dynamic systems. Children will write programs by constructing artificial animals and turning them loose in an environment. Each object is a separate creature, with sensors, effecters, and processing elements, that can act of its own accord (Fenton and Beck 1989).

Playground was incorporated into the curriculum of the Open School, which provided "a living laboratory for study and experimentation" of the Playground system for its developers (Fenton and Beck 1989). The Open School proved to be a powerful "living laboratory" in which educators, alongside researchers from Apple Computer, could experiment with building a "model 'school of the future' " (Marion Resume).

play, I tell the story of how her methodology evolved over time, absorbing influences from art, architecture, and simulation gaming. This chapter is the first of two case studies of non-computerized forms of play, and provides valuable materials from which I generalize the play design principles in chapter 9. My account is informed by ethnographic study; in addition to reviewing published accounts of City Building Education by Nelson, I interviewed her and conducted some participant observation at a teacher training workshop.

Transformation

Doreen Nelson

From an early age, design, architecture, and art were constitutive experiences of Nelson's life. She has one sibling, the architect Frank Gehry (born 1929), who is ten years older. She considers him as a "second father figure." As a child, everything he said to her was "just wonderful … still is. He influenced me so much, I couldn't walk down the street with him without him saying, 'Look at that tree, and look at how that tree grows, and look at that building… And look at this, and look at that.' " (Nelson 2015).[13] Construction play with scrap materials was a formative experience for Gehry, who had, as a child, played with their grandmother, transforming bits of wood into "cities, bridges, buildings." It was, he recalls, "a license to play" that would go

[13] Her home is filled with art, many of Gehry's famous corrugated cardboard furniture pieces, including a dining room table, and a painting by Gehry's son.

on to serve him as an “emotional and intuitive” “anchor” (Isenberg 2009).[14] Gehry and Nelson were both born with the name Goldberg, but her brother, responding to antisemitism, changed the entire family’s name to Gehry.[15] It was 1954, and Gehry was 25, and Nelson 15. Nelson retains the last name of her first husband, because she had begun writing under that name, and neither Gehry nor Goldberg felt like appropriate or fitting names anymore (Nelson 2015).[16]

While Nelson was trained as a musician, a harpist, her mother encouraged the pursuit of a teaching credential so that she would have more economic opportunities. She graduated with a BA in teaching from UCLA in 1959, a credential that was invaluable when a family economic crisis precipitated her family members to all begin working. Nelson started teaching, and found that she loved it (Nelson 2015).

---

[14] According to Gehry,

> My grandparents had a wood stove, and I’d go with her [Gehry’s maternal grandmother] to shops around the neighborhood where we’d buy burlap bags filled with small, leftover wood pieces. When we got home, she’d open one of the sacks and fill the kitchen floor with pieces of raw wood in all kinds of shapes. We’d sit on the floor together and make things out of them—cities, bridges, buildings.
>
> When I was choosing a profession, I didn’t know what I wanted to be or what to do with my life, and I remembered us making those things on the floor. … I started thinking about what was the most fun I ever had in my life. What would be something useful for me to do? And I had this image of being on the floor with Grandma making things.
>
> Years later, I realized it was a license to play. … By the time you get there, you’ve been through a school system that tries to make everything rational, mathematical, and logical and all of a sudden you’re confronted with something that’s emotional and intuitive. You look for anchors, and my anchor was this memory of my grandmother (Isenberg 2009).

[15] While Gehry and Nelson were born with the family name Goldberg, her brother, struggling to begin his career and support his young family, changed the name of their entire family—his wife, sister, and parents—from Goldberg to Gehry. Gehry’s first wife Anita, then pregnant with his first child, lobbied for a name change in response to antisemitism; for example, not being accepted to join an architecture fraternity, or a colleague encouraging him to change his name so they could start a partnership (Isenberg 2009).

[16] “I just started writing stuff from doing stuff under Nelson. I should’ve gone back but Gehry didn’t really feel like my name either. It wouldn’t have made sense to go back to Goldberg because there were no Goldbergs anymore” (Nelson 2015).

A Crisis of Imagination

Urban planning and unrest precipitated Nelson's involvement with imaginative cities. In 1964, the Los Angeles city planning department began a project called the LA Goals Program. It was to envision the city's future, and advise the municipal leadership on social as well as architectural matters. With the 1965 Watts riots, hope turned to fear, and the program became part of LA's response to the urban crisis. Nelson joined a task force on city planning and education, and researched what inhabitants envisioned wanting decades into the future. She was stunned by their lack of imagination. Respondents leaned on prepackaged visions of the future, like the monorail, familiar from Disneyland. "Nobody could even think of anything. They had no idea. … They could think of Disneyland because it was already there. And they wanted a monorail" (Nelson 1984; Light 2003; Nelson 2015).[17]

[17] She worked on LA Goals from 1965-1967, but her contributions continued beyond 1967.

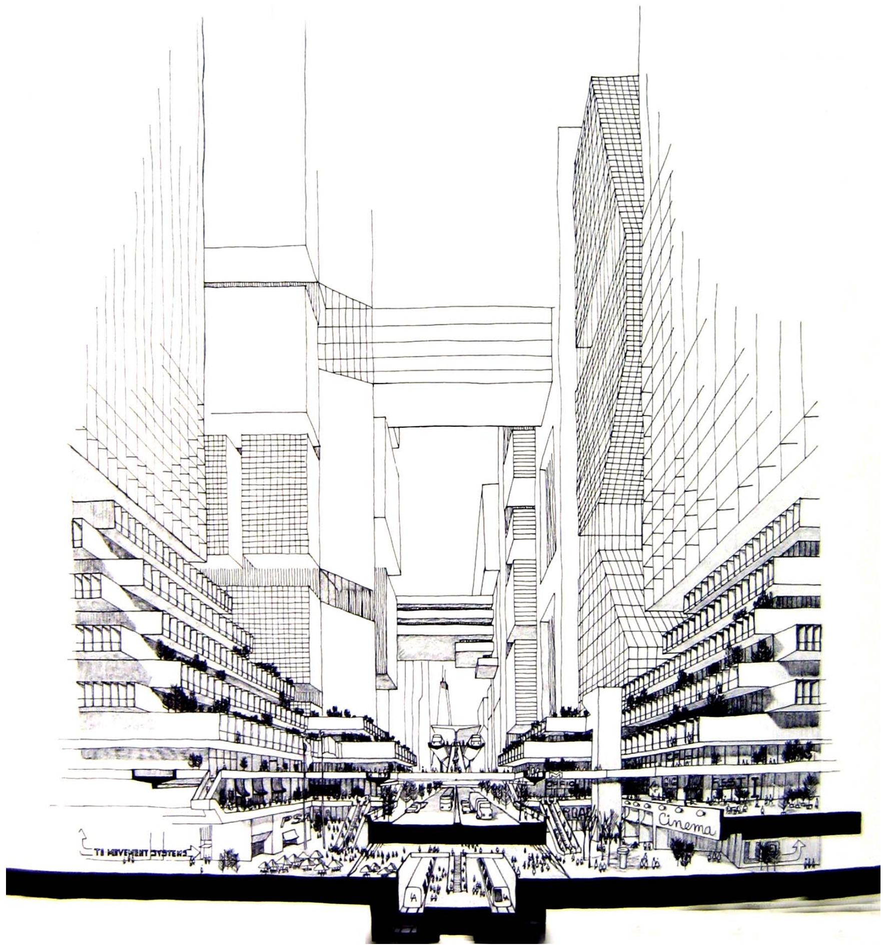


Figure 7.3: Architectural rendering from a follow on report to the Los Angeles Goals project ("Concept Los Angeles" 1970).

Unrest continued through 1967 in cities across the United States. After finishing with the LA Goals project, in 1967, another project committee approached her, imploring: "You're an educator, what do you think we should do? Because, clearly, the city of Los Angeles is in peril. Nobody has an imagination. They can't imagine what things should look like. How do we educate people?" (Nelson 2015). To educate people who "can't imagine what things should look like," Nelson responded that "it's worthless to train adults because it's too hard." Instead, "You educate children and then you hope that a fifth grade kid, in fifteen years, might be in a position to holler at somebody and say, 'Don't do that.' " To address a failure of civic imagination among adults, one needed to focus on children—a natural position for a school teacher to take (Nelson 2015).

Nelson began experimenting, taking a group from the LA Goals project to a classroom, where she

> gave kids pieces of roads … and I gave them a tabletop that I said was a city. … They built some buildings, and I gave them these pieces of roads, and asked them to put the roads down on the city. And it was a disaster because they kept wanting this part to join this part, so there was a curve part. It had to curve, and that one had to join, and the very thing that I thought I was doing, I was operating against, because I gave them the trigger device that was already a road.

The road pieces were overdetermined, shutting down what she wanted to encourage: imagination. A road piece was a "trigger" for roads, and all the expectations kids brought to them—such as precision of connection, which the pieces did not afford. But she persisted, and learned from her experiments: "So now I don't call them roads, I call them pathways." "And if you had a pathway, you could imagine a pathway meandering. You could imagine a pathway in the sky" (Nelson 2015). The road is denaturalized into a more abstract pathway, a technique that we will see repeat throughout her practice. For example, she prefers "shelter" to "house," nomenclature that points towards the design problem, rather than the familiar solutions we already inhabit. This denaturalization and abstraction creates a space in which imagination and creativity can take up residence. Rather than use the literal and familiar as a prompt, with its attendant commonplace associations, Nelson uses design problems as a prompt, drawing out creative and imaginative solutions (Gingold 2015).

City of Bridges

While pursuing a master's degree at California State University, and teaching at risk children at a school in Los Angeles's Venice neighborhood (1968-1969), the theme of the city resurfaced as the impetus behind her first classroom city.

> I had this crazy notion that they [the students] had to build a city. And that each kid had to have a piece of it, and that each kid had to fight with the other kids about what was on their pieces, and that somehow they would build this thing and it would represent a real place. A real location (Nelson 2015).

In September of 1969, Nelson, along with the teachers and students of grades 3–6, switched focus from the history of westward expansion, as programmed by the school district, and turned instead to the future of their own community. Rather than ask students to be "imitators and replicators," and enact "a numbing program that only fed the mind with facts," Nelson sought to "challenge them to think today about the world of tomorrow," and "apply creative thinking to everyday life" (Nelson 1984). The process began modestly, with observation of their neighborhoods and "a basic study of physical change" (Nelson 1974a).

> They began by taking common objects like forks and pencils and coat hangers and making simple transformations. In one exercise they picked everyday objects, studied them, and fabricated the objects to their scale as costumes, learning about structure and space as the object changed in size (Nelson 1974a, p. 38).

These transformations were initially done with paper and pencil, as drawings and lists, and then progressed into scale models. Eventually, they went on to document and reproduce Venice in miniature. Nelson writes that "[b]oundaries were formed around the community, and by decreasing the scale the students brought a large section of the Venice landscape into their classroom." Through a " 'land grab'," students took ownership of Venice and then enacted "a giant simulation of the community planning and government process" (Nelson 1974a).

Outside experts scaffolded classroom activity. She recruited architecture graduate students, who introduced specialized knowledge and craft about planning and model making. The teaching process became decentralized, and the presence of experts, in this case architects, heightened the sense of importance of what was transpiring. This practice, of bringing in outside experts to distribute, scaffold, and promote classroom activity is a theme that reappears throughout City Building, as experts from the community join the classroom activity (Nelson 2015).[18]

Nelson found that the city was a potent instrument for the transfer of abstract and powerful concepts, a central concern of hers as an educator: "For example, if I taught them a principle about movement. If I taught them about the heart, and the fact that things move around in the body. Could they now go out onto the city and build a movement system that reflected what they had learned about the heart? And yes they could." She was stunned by the model city's potency as a vehicle for conceptual transfer, as well as her inability (like Wright) to escape it. She had thought that after the city she and her students would build a "a marine biology world," and then a "business, but it's been over forty years and I'm stuck with the city… The city is like a classic metaphor somewhere in our psyche for our human thinking. We navigate in this thing all the time called the city and we obviously have a very deep connection [to it]" (Nelson 2015).

But one kind of conceptual transfer was foremost in her mind: agency. She wanted her students to be both empowered and imaginative, to feel in control of their lives, and exert imaginative agency over their built environment. Through transformation, the surrounding city—big,

[18]Rather than enlist help from her older brother, who was then struggling to support his family, Nelson asked an artist friend, who suggested she contact USC and asked if they could send some architecture students to help. Some architecture graduate students were excited about the possibility of working on a future, imaginative city, and stayed on to help for a year (Nelson 2015).

imposing, and filled with problems and possibilities—could be appropriated, brought into the classroom, wrestled with, and reinvented. Nelson writes that as they "began building a city, I kept telling my students that "it's got to be bigger." I thought that if they did something big—something they couldn't put in their desks—then they might be empowered to do some big things in their lives" (Nelson 1984). But to feel powerful and creative, they would first need to feel safe. And to do this, they would need to transform themselves.

Transforming Self

Just as her older brother, born a Goldberg—but now Gehry—had, by enrobing his family in a new name, garnered a degree of protection from antisemitic prejudice, she, too, sought to transform herself and her students into more powerful, protected, comfortable, and flexible forms. Nelson was keyed into and took inspiration from the cultural moment of the late 1960's, which, she observes, was all about change, power, and powerlessness. Her students felt powerless. This disempowerment holds true today—I watched Nelson deliver this story to a roomful of teachers who unconsciously nodded in unison at this point. She felt powerless, as a teacher and as a woman. She asked herself: How do I make my students powerful? How do I become powerful? Her answer was transformation, to cover up, and become something new. She compares transformative costuming to "a new skin." To enable her kids to be able to "fool around," play, make things, and transform their environment, they first needed to be comfortable and feel empowered. Transforming themselves into new forms made this possible (Gingold 2015).

The impetus for object dress up was roadblocks Nelson encountered in getting her students to learn about and reimagine their surroundings. She had initially intended for her kids to document their own homes, and convinced a wealthy patron to buy the cameras and cover photo development costs. The idea was for the kids to think about their home environment, and how they might change it. The kids, however, failed to take the photographs, and instead produced endless excuses: "I don't want to take a picture of my house; it's so ugly" (Nelson 1974a, p. 38). Nelson says that "[i]t dawned on me that they were embarrassed of their houses. They didn't want to take pictures of them" (Nelson 2015).[19]

She was stuck. How could she get them to reimagine their city if they couldn't even imagine rearranging their bedroom? Her solution was to "cover them up. They needed to become something besides who they were. Instead of being you, you needed to be a hammer. And then if you were a hammer, you could think about where you lived, and who lived with you, and what kind of a building it would be, and how it would be arranged." A child transformed into coke bottle, for example, would be free to imagine her environment: a vending machine, a refrigerator, or something else (Nelson 2015).

[19] Even after recruiting a friend, a talented photographer, to take pictures for them, the kids didn't recognize their own homes in the artful black and white photographs. According to Nelson, kids would say something like: "Yeah, it's my house but it doesn't look like that" (Nelson 2015).

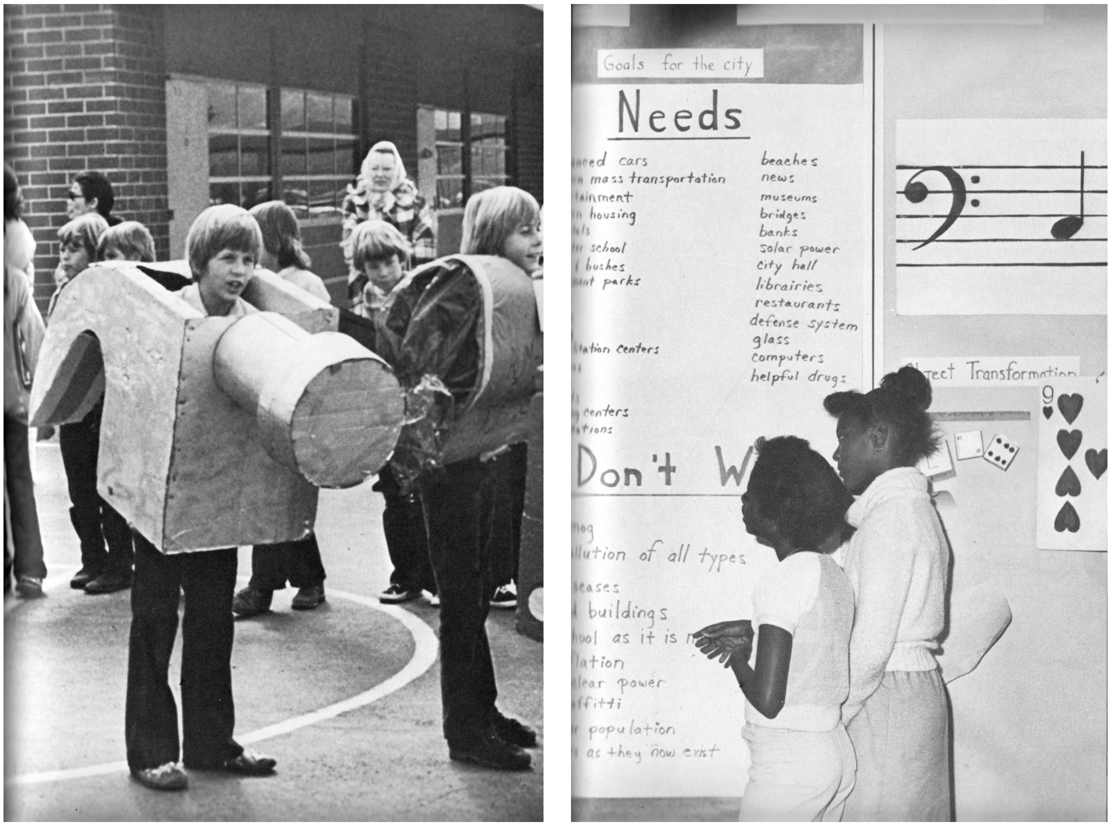


Figure 7.4: Images from City Building Education *(Nelson 1982). At left, a boy in his hammer costume. At right, two girls stand next to a list of "Goals for the city," categorized into "Needs" and "Don't Want." Many of the photographs in this book were taken by Charles Eames, as well as Grant Taylor and Bobbi Mapstone.*

Because it was close to Halloween, Nelson thought the kids would be comfortable with the idea. Indeed, the kids wanted to march at the school Halloween parade in their new costumes. The safety created by self-transformation was apparent in the teacher training I observed. Trainees chose objects to represent themselves, such as a stapler or lipstick. They then elected a mayor for their town by having their *objects* campaign for the mayorship. Debate and discussion took place *on behalf* of the objects up for election, which served as masks and proxies for the people they symbolized. The dress up procedure introduced a layer of abstraction and comfort to what might otherwise be a fraught and combative process (Gingold 2015).

Transformation also fed conceptual transfer. Making and donning costumes not only fostered the comfort and elasticity she desired, but helped the children to understand the history and meaning of Halloween, and why particular costumes are traditionally made and worn. Later on, her students would dress up as their favorite toys and design holidays for themselves. This, she found, helped them understand the cultural specificity of holidays such as the Chinese New Year (Gingold 2015; Nelson 2015).

They role played their new selves. Inspired by games that actors play to develop character, and an interest in teaching symbolism, Nelson had the students interview one another "as the character."[20] One might ask of "King Lear, Who are your friends? What are you afraid of? Where do you sleep at night? What is your history? Who are the people you like to hang out with?" Nelson had "the kids interview themselves as the object." For example, one kid dressed up as a yellow M&M candy:[21]

> So, the kid is dressed as this M&M and the other kids are interviewing him and they say, "What's your favorite thing to do?" And his favorite thing is when his owner "reaches into the bag and chooses me." A yellow M&M. I mean, the kid, this kid is waxing on. You know, how wonderful this feels. And then they go on: "Who are your friends?" "The other M&Ms." "Where do you sleep at night?" And all these kind of questions: what's your history, what was your origin? One of the questions is "When were you born?" and "When do you die?" "And where were you born?" "I was born in a candy factory." "When do you die?" This kid puts his hands on his head like this and he said, "Oh my god," he said. "I was born to die." [laughter] It's that kind of stuff that's so strong (Nelson 2015).

In Nelson's analysis, these character transformations, a way of "making new us's," were profound. By shifting contexts, they allowed the exploration of the multidimensionality of people, and contributed to the building of new communities. Stylized character transformations

[20] She learned of this technique while working with the Mark Taper Theatre, on an NEA grant. Note that the NEA grants wouldn't come for another couple years, and so I am compressing some of the later elaborations into the story here.

[21] This particular story is from her work at the Vivarium school, which came later in her career.

were highly evocative, symbolic, and charged. The kid who made the hammer costume went on to become an architect. The kid who dressed up as a yellow M&M became an emergency room doctor at Princeton. (Nelson is satisfied that most of her kids went on to college.) One student dressed up as a blade because he liked cutting things; later in life, he committed suicide. Nelson remarks that she wasn't a psychologist, and wasn't equipped to work with everything that surfaced amidst the transformations she had provoked (Nelson 2015; Gingold 2015).

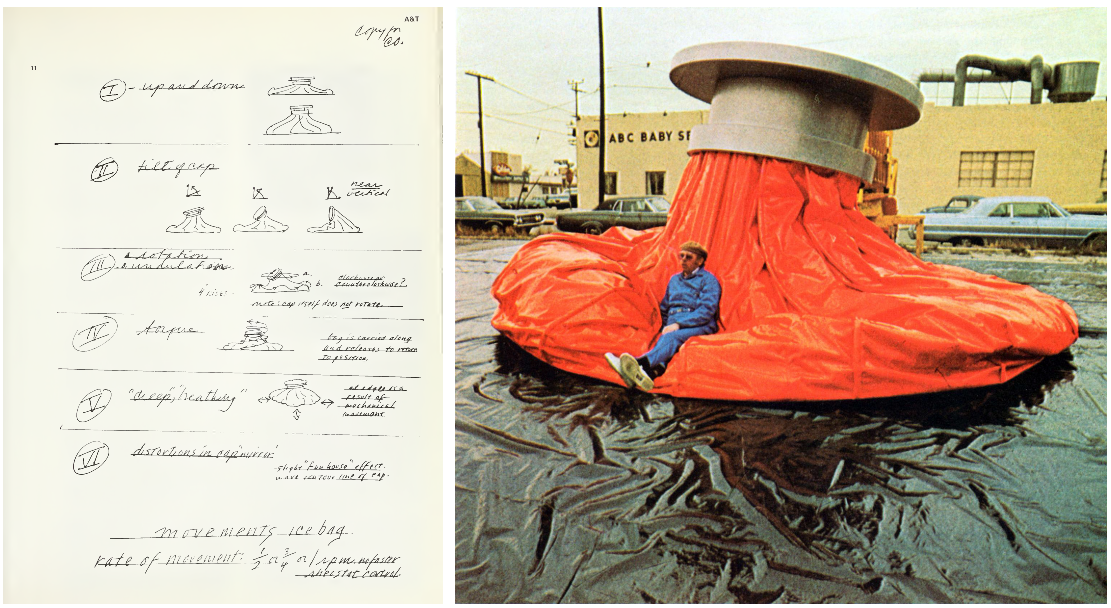


Figure 7.5: Left: Oldenburg sketches for Ice Bag's *movement (Tuchman 1971). Right: Claes Oldenburg on his* Ice Bag *(Johnson 1971).*

Nelson and her students took inspiration from Claes Oldenburg's *Ice Bag,* first exhibited at the Osaka 1970 World Exhibition, as a beacon of transformation and multiplicity of meaning. Like many of Oldenburg's sculptures, it was an ordinary object transformed through enlargement and softening. But *Ice Bag* is unique in that it is animated.[22] In Nelson's analysis, *Ice*

[22]Its animated-ness informs the project's atypical level of technological ambition. Oldenburg conceived of *Ice Bag* while installed as a kind of artist in residence at Disney's WED Enterprises, the organization responsible for making Disneyland, through LACMA's art and technology program (chapter 5; Tuchman 1971). The technical,

Bag was about change: it was pliant, fluidly rotated and animated, and bowed. It was about healing as change, and in particular the postwar healing between the United States and Japan. In *Ice Bag,* as well as in her classroom work, "meaning is in the ambiguity of these objects." Her students, she recounts, were fascinated that an artist would make something ambiguous, something which harbored multiple meanings (Gingold 2015).

Ice Bag is an example of the kinds of influences that Nelson drew from. She saw trans-

---

artistic, and pop cultural orientation of Disney's amusement park enterprise clearly informed his thinking, as his initial ideas were called "Oldenburg's Ride," a direct reference to Disneyland. Nelson's use of *Ice Bag* as a touchstone about creativity, transformation, and ambiguity is remarkable in part because of its resonance for Braun, for whom it served as a totem for the synthesis of technology and art.

LACMA's A&T program was conceived in 1966 by curator Maurice Tuchman, who sought to place artists into LA's futuristic coastal industries—from NASA's Jet Propulsion Laboratory and The Rand Corporation to Universal Film Studios—where they might operate as if "in their own studios." *Ice Bag* emerged from an initially uneasy partnership—one of the first to come to Tuchman's mind—between Oldenburg and WED Enterprises, the Disney subsidiary responsible for creating Disneyland and Disney World. Oldenburg became a kind of artist in residence at WED, and situated amongst designers and craftsmen working on Disneyland's haunted mansion ride, was inspired to research and think in terms of metamorphosis, mirrors, magic, and the theme park ride. Asking himself how he could take advantage of WED's techniques and resources, Oldenburg collected and invented ideas for kinetic and illusionistic sculptures, collectively conceived of as belonging to a theme park ride entitled Oldenburg's Ride. The A&T catalog captures the theme of these sculptures—e.g. a giant undulating jello mold, giant eggs cracking, scrambling, and reconstituting—as "disintegration, transmogrification, and reconstitution," qualities that Oldenburg compared to the unsettling phantasmagoric transformations of dreams "where your teeth fall out, but on awakening you find out they didn't" (Tuchman 1971).

Oldenburg was eager to focus on fabricating one design, his Giant Ice Bag, for the Osaka 1970 World Exhibition. Although WED had encouraged Oldenburg to conceive of models that they might fabricate, they ultimately backed out, unwilling to commit the resources to see any of Oldenburg's designs through. Under the direction of Ken Tyler, Ice Bag was ultimately fabricated by Gemini G.E.L. in time for the 1970 World Exhibition. This was a tremendously challenging project, and required overcoming immense hurdles in logistics, programming, materials, kinetics, and transportation. One model was destroyed when, according to Tyler, "the fabric got caught in the gears and it ate itself." The piece stands as a parable about the wonder and impossibility of combining art and technology. The production of an animated film, theme park ride, or video game—which Lantz compares to taking everything that's hard about making a bridge and combining it with everything that's hard about making an opera—all subsume serious and sophisticated technology for pointless, senseless, artistic, and playful ends. These are, as the story of Ice Bag makes clear, exercises in absurdity, masochism, and enchantment. One confronts the need to plan, indirectness, a heaviness of materials, dealing with things you don't understand, specialization; the wonder of automation and the illusion of life; the audacious intention, in Oldenburg's words, "to give birth to nature yourself through the machine" (Tuchman 1971; Lantz 2014b):

> I think perhaps my approach to technology is to remove the difficulty of technology, as to take something which is formidable in its complexity, and make it do some very foolish thing—and I sort of like the idea that all this time and effort was spent on the Icebag [sic]. I'm creating something which really doesn't do very much. It just does something very simple; and it doesn't do anything more really than a leaf does in the wind (Tuchman 1971).

formation as the central act of the creative dramas and processes that artists and designers used to produce their “intuitive” “leaps of insight,” an observation borne of conversations with artists like Oldenburg and designers like Buckminster Fuller (Nelson 1984).[23]

Purium

City Simulation Games

“The children,” writes Nelson, “were very proud of their work, and the number of people who came to see their fantasy city of Venice gave increased weight to their ideas” (Nelson 1974a). This process was true of Nelson herself—as more people came to observe her evolving ideas and teaching methodology, they took on greater importance and weight.

It didn’t hurt that the miniature cities were visually arresting. One gifted photographer who become enamored with the imaginative cities taking shape in Venice was the legendary designer Charles Eames, whose studio sat across the street from the school. Eames helped Nelson secure funding from the National Endowment for the Arts (NEA), and took thousands of photographs, many of which populate Nelson’s publications. Along with David Olney and Grant Taylor, Eames made multiple films and slide shows about her pedagogy, which led to even more widespread attention, including national news coverage (Nelson 1984; Nelson 2015).[24]

---

[23]Fuller wrote the short glowing preface to *City Building Education* (Nelson 1982) and *Transformations* (Nelson 1984).

[24]According to Nelson, Charles and Ray Eames found her work in 1971, and shortly thereafter—Charles Eames served on the National Endowment for the Arts (NEA) advisory council from 1971 until 1976—Nelson received an NEA grant, the first of many, for City Building Education. Within six months, Nelson had an office within the Eames studio, which made multiple short films about City Building. (According to Nelson, Charles Eames could not put his name on the films because he was on the NEA advisory council, and so other contributors, such as David Olney took directorial credit.) Charles Eames took thousands of photographs of City Building Education classrooms, many of which populate Nelson’s books, and along with Ray Eames, advised Nelson; Ray Eames,

As a result of this media attention she became acquainted with Richard Duke's city simulation games. These emerged from the tradition of military and business simulation games.[25] Unlike Forrester, who produced non-interactive city simulations, Duke produced simulations that relied heavily on human actors, that were, in other words, playable simulation games.[26] In *Gaming: The Future's Language*, Duke argues that "games are a form of communication" amongst players (Duke 1974), ideally through which " 'men of good will,' each holding a unique perspective, sincerely seek to communicate and/or negotiate a compromise" thus revealing their "collective intuition" (Duke 2011). While Nelson was inspired by art, architecture, and education, Duke's simulations came from the tradition of war gaming. Contact with simulation games altered Nelson's methodology, and she began to see her make believe cities as a kind of "game simulation" (Nelson 1974a; Nelson 2015).[27]

---

for example, offered design feedback on Nelson's book *Transformations* (Nelson 1984; Nelson 2015; NEA annual reports give Charles Eames's service years).

At the end of 1970, a national news network ran a special on the future of the world, and included a piece on Nelson's City Building methodology. Nelson recollects that the program included segments on Stewart Brand and the New Games Movement, the authoritarian rehab program Synanon's "games" (Gelder 1997), and Richard Duke's seminal city simulation games (Nelson 1974a; Duke 1995; Nelson 2015).

[25]Through Duke, Nelson learned about city simulation games such as METROPOLIS, METRO-APEX, and CLUG, and came to see her work as part of this simulation game tradition (Nelson 1974a; Nelson 2015). Duke invited Nelson to Michigan to share her work with him and his colleagues, who had, since the early 1960's, been designing and running urban simulation games for students, city council members, and planners. Like Nelson, Duke was focused on cities and experiential learning, but his work emerged from a different tradition. Whereas Nelson gathered influence from art, architecture, and education, Duke's practice emerged from the tradition of World War II era war games used for logistical planning and the business games that followed (Bellman 1957; Moncreiff 1965; Duke 1995; Duke 2011; Peterson 2012). In METROPOLIS, for instance, participants enacted a city budget negotiation procedure. The game was used, to resounding acclaim, by the Lansing Michigan city council to repair their dysfunctional budget negotiation process. Later, with a grant from the Ford Foundation, Duke began incorporating computers into his simulation games (Duke 2011).

[26]The Ford Foundation funded work on cities throughout the 1960's, for example at the RAND corporation, at which Schelling worked, and at MIT, where Forrester used Ford Foundation grant money to fund the computer time used to make his *Urban Dynamics* model (Light 2002; chapter 3).

[27]She began to think and write, at least for a while, about her methodology as a type of "game simulation" (e.g. Nelson 1974a). Nelson credits Duke with giving her the conceptual framing and terminology with which to understand her work as a form of simulation. "He taught me a lot," she says (Nelson 2015).

City Building Education

As a result, Nelson was inspired to create “a short-term game simulation based on some of the ideas” that she “had been working with in the Venice classroom” (Nelson 1974a). The game, called “Purium,” would eventually lose the “game” moniker, and simply become known as the “Purium Instant City” (e.g. Nelson 1984). It was this instant city that Maxis was particularly interested in, and that Claire Curtin found particularly inspiring and unforgettable (Curtin 2015).

Purium took shape at a series of workshops hosted at the Smithsonian Institution. It was co-created by her older brother Frank Gehry, who had already been running workshops for disadvantaged children called “Fantastic Cities.” Although Nelson was initially “extremely reticent” to share “these children’s games” with “adults,” and “was especially intimidated by the idea of professionals, such as architects, playing city building games with styrofoam and cardboard,” the workshop was a smashing success (Nelson 1974a). It was here that the name “City Building” was chosen “to relate the invisible fabric of social, political, and economic relationships to the built environment” (Nelson 1984).[28]

[28]“In 1971, Susan Hamilton invited me to the Smithsonian Institution, where we developed the basic teacher training model” (Nelson 1984, p. x). The Smithsonian workshop was important in that it was where City Building crystallized, in name and form. The first summer, in 1970, involved children from grades 2-10, and functioned as a research site for City Building, while the second summer, in 1971, functioned as a 5-week live in teacher training seminar, and added teachers from across the country to the mix of students (Nelson 1984). The workshop start dates given in these two publications (Nelson 1974a; Nelson 1984) don’t precisely concur.

My Weekly Reader News Parade

Edition 4 • Vol. 53 • Issue 1 Sept. 15, 1971

Mini-cities are springing up in many parts of the country. The cities are models being built by girls and boys. They are learning about city problems.

The young people below had fun building their model city. They were guided by a city planner who has worked on many problems in large cities.

These junior planners were faced with questions about a city of the future. How would they get people to and from their jobs? Where would they build an airport? What would they do about electric lines in a city?

Pupils used paper, cardboard, and plastic foam blocks to make the cities. They decided what they would do about old buildings, crowds, cars, noise, and pollution. They planned homes, schools, hospitals, and stores.

A masking-tape monorail was built so that people wouldn't have to drive cars into the city. Some buildings were put on posts to make room for parks below. Building a mini-city helped the pupils see what could be done in big cities of the future.

CHILDREN PLAN FUTURE CITIES

The first issue of *Weekly Reader News Parade* has been mailed early. The second issue, dated Sept. 22, will arrive on schedule for use that week.

A city planner (above) helped pupils as they built their model city at the Smithsonian Institution in Washington, D.C. Some buildings were held up by cardboard tubes. (See front-page photo also.)

At the Smithsonian, students were allowed some make-believe things in their city. For example, they could mine from the mountains make-believe stuff that could make the city's trash rot easily. They could dangle buildings from a balloon over the city. Do you think that's a far-out plan? Remember, this was a city of the future that they were planning.

Read Think for Yourself in the box below. Maybe you'll get ideas for a mini-city.

Check Yourself

After reading this week's front-page news story, answer the following questions.

Get the Facts

1. Why are some students building mini-cities of the future? ______
2. What did a group of students decide that they must plan for in their model city? ______
3. What did they do about cars? ______
4. Where did they plan to have some parks in their city? ______

Think for Yourself

Imagine that several thousand people will soon be moving to your community. What changes might be needed? What will this mean for fire, police, hospitals, and the water supply? What could be done about cars, streets, and highways? Would you need more playgrounds and parks? How would you plan for these things in a city of the future?

GLOSSARY

monorail—a single rail used as a track for cars and trains to travel on

mini—a beginning for a word that makes the word mean a small copy, or a model of something larger.

R4-1

MY WEEKLY READER NEWS PARADE Education Center, Columbus, Ohio 43216

Class rate: 35¢ per semester per student; 70¢ per school year in quantities of 10 or more sent to one address. Foreign: add 30¢ per student. Single subscription rate on orders totaling less than 10 copies: $1.40 a year, payable in advance. Foreign: add $1.50 per subscription.

Figure 7.6: An article about Nelson city building workshop hosted at the Smithsonian in 1971. In most top right photo, Doreen Nelson with her brother Frank Gehry. (Newsletter supplied to me by Nelson, who also identified herself and her brother for me.)

A newsletter about the first summer workshop summarizes the activities:

> These junior planners were faced with questions about a city of the future. How would they get people to and from their jobs? Where would they build an airport? What would they do about electric lines in a city? Pupils used paper, cardboard, and plastic foam blocks to make the cities. They decided what they would do about old buildings, crowds, cars, noise, and pollution. They planned homes, schools, hospitals, and stores. … students were allowed some make-believe … they could mine from the mountains make-believe stuff that would make the city's trash rot easily. They could dangle buildings from a balloon over the city ("Children Plan Future Cities" 1971).

Purium is a scale model of Nelson's methodology. Playing it is a way to experience "the totality of City Building, a whirlwind of activity that incorporates all the processes and principles of this way of learning," preparing both students and teachers for the City Building program at large (Nelson 1984). By zooming into this model, we can catch a glimpse of City Building Education as a whole.

Purium Instant City

Figure 7.7: Example of a Purium landscape (Nelson 1984).

Scenario

Purium begins in a pristine landscape, typically composed of a mountain range and a valley. A stream tumbles from the mountains, feeding a lake at the center of the valley in which the city will grow. A story sets the scene: "An old prospector fleeing the discontents of modern urban life comes to a tranquil oasis of natural beauty in the wilderness."[29] This prospector

[29]This tale seems to be a fantasy enactment of the back to the land movement of the time, with its concomitant desire to escape and reinvent American civilization (Turner 2006). But unlike the 20th century back to the land movement, or Henry David Thoreau's *Walden*, Nelson does not idealistically negate the civilization whose fruits

discovers purium in the mountains, a remarkable material "which makes all wastes biodegradable. This discovery has such far-reaching implications that the federal government" initiates mining. Players, acting on behalf of the government, are responsible for planning and governing a mining boom town, and will do so under various pressures enacted by the teacher, who acts in the fictional role of federal government, as well as the functional role of game master. One of these pressures is time: after planning is complete and the city begins to grow, its population will double every five minutes, which is equivalent to one month of simulated time (Nelson 1984). As in a game, there is setup to do. In multiple stages, participants will establish criteria, organize into a community, and plan out the city.

Criteria

First, criteria are set. Students are guided by the teacher into "asking questions, discussing, and describing their ideal visions for community life." They brainstorm what they "Need" and "Don't Want" in their city. The teacher-facilitator asks questions that probe at qualities of the city often taken for granted. For example: Where does trash go? What would you like to change about your city? What do you like? The resulting criteria are posted prominently, and used at the game's conclusion to evaluate the resulting city. These criteria will also live beyond Purium Instant City, and become an evolving set of design requirements for other City Building activities.

---

make these utopian dreams possible. City Building reimagines and enacts, in miniature, the interconnectedness of urban civilization.

Figure 7.8: Criteria (Nelson 1984).

Having students set their own criteria establishes a new relationship between students and teacher, and an orientation that will underpin the entire City Building enterprise. Students are encouraged to "attempt projects without waiting for the guidance of external commands. Their own authority becomes internalized, and the teacher becomes a helper, assisting the class to see how it measures up to its own yardstick." Purium begins the transformation of students into agents that can take initiative and set their own goals (Nelson 1984). Designs are encouraged to be "never-before-seen," as opposed to *already been seen* (Nelson 1984). In the teacher institute I witnessed, these concepts were of central concern, and called "NBS" and "ABS" (Gingold 2015).

Community and Roles

Now that criteria are established, the community is organized. The map is gridded up into districts, and players, invested into the role of citizens, are assigned to them. Each district elects its own district commissioner to represent it in the city council. Then, a mayor is elected who will liaise between the inhabitants and the federal government. Based upon the Need list, city commissioners are designated—for example: Education, Parks and Recreation, Culture, Aesthetics/Natural Beauty, and Underground.[30] Each player wears one or more badges indicating their simultaneous status as a citizen, representative, or commissioner. "These multiple badges emphasize that in an interrelated entity like a town, each member has more than one role to perform" (Nelson 1984).

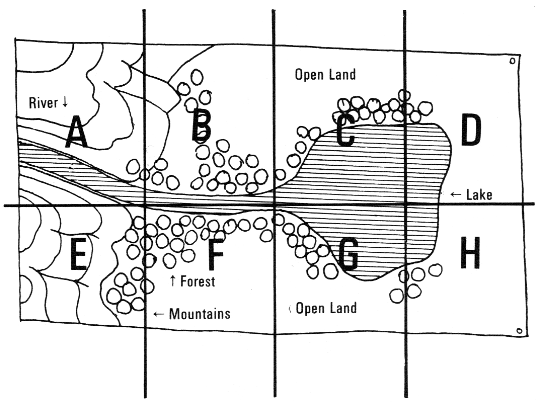


Figure 7.9: Grid plan of Purium, and photo of participants wearing badges representing their multiple roles (Nelson 1984).

Planning

Next, planning commences. Players of Purium have a great deal of imaginative leeway, as it is set 100 years in the future (Nelson 1984). "[C]hildren were told they were 'future people'

[30]It is possible for the city to develop underground, which can be modeled by building on the underside of the tabletop/landscape.

(meaning technology could accomplish anything they wanted to build)" (Nelson 1974a). Like any good science fiction author, "[t]hese citizens of the future may build anything they envision, but the rule is that *they must invent and identify the technology to carry it out.*" Innovations are made by announcing them, but their physical manifestations and implications, including new problems resulting from the inventions, must be worked through (Nelson 1984).

Council districts take 15-30 minutes to draw aerial plan views of their neighborhoods. Commissioners facilitate planning, ensuring that the concerns they represent are accounted for, and exercise final approval. Representatives meet every five minutes to discuss progress. After the planning time is up, district plans are reconciled into a unified plan which, like the criteria, is posted on the wall for reference (Nelson 1984).

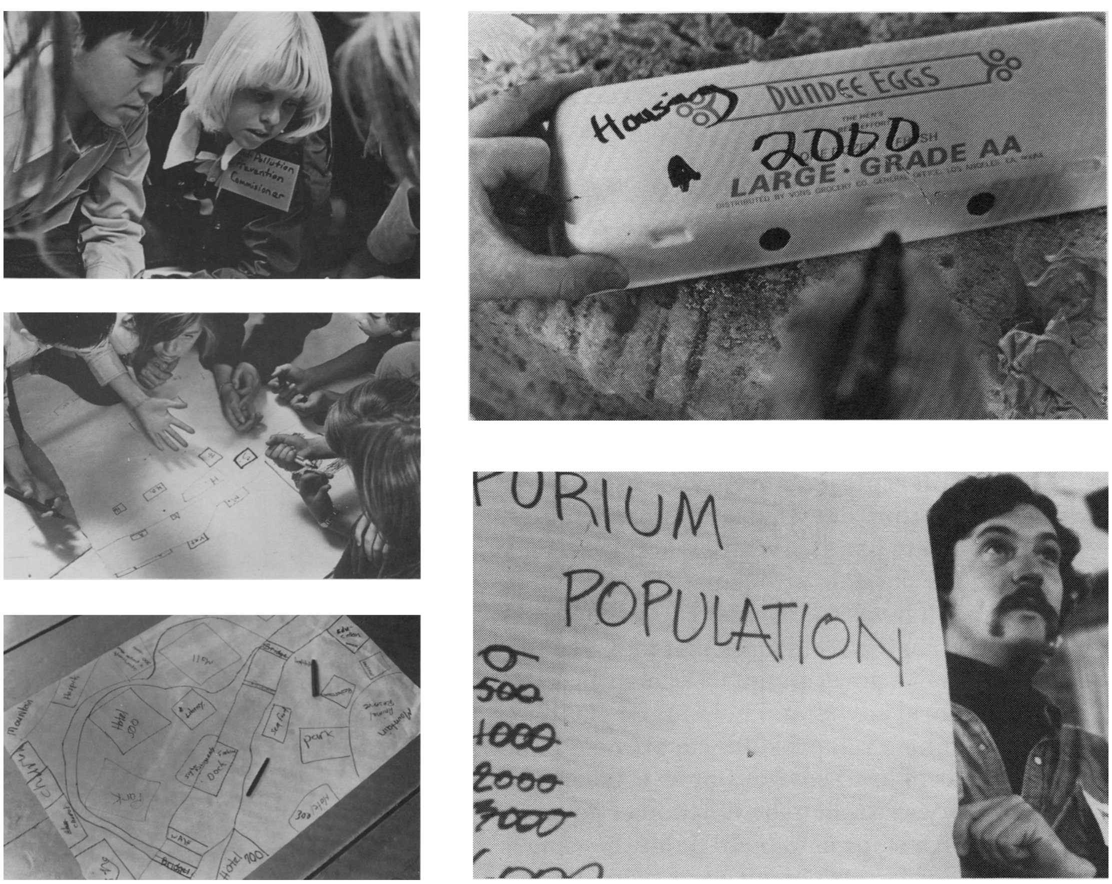


Figure 7.10: Planning and growth (Nelson 1984).

Growth

Once all of the planning is complete, and criteria, governance, and land use plans have been established, then play—the building phase—can begin. The clock begins to tick, Purium rapidly grows, and "students meet the unplanned variable of fate head-on." The carefully considered plans and criteria will be put under tremendous pressure (Nelson 1984). Pressure is enacted by the teacher-facilitator, who performs in one of two guises. As the game master, the facilitator helps brings the world to life. As the federal government, she role plays a powerful agent to whom they are responsible.

The facilitator helps to sustain the simulated city and enforce consequences. Every five minutes of play (equivalent to one month of simulation time) the population doubles, bringing added pressure for housing and infrastructure. Nelson gives the following examples:

> "The population has just doubled; we need housing for 2,000 more people." "A woman is having a baby; where's the hospital?" "The president is arriving by plane; we need an airport" (Nelson 1984).

These high pressure events are enacted by the facilitator, who acts in the guise of the game master. If housing needs aren't met in time, then the facilitator, this time performing the role of the federal government, might step in and build something. Memos from the federal government often identify problems and suggest solutions. If excess materials aren't properly disposed of—say, in a landfill—then the facilitator, acting as game master, might dump them into the lake. In the exercise I observed, the instructor would threaten to intervene as the federal government and make a decision if the city did not come to a decision within the allotted time.[31] This was observed to be highly motivating—threatening to take away the feeling of agency that is a central pleasure of City Building. I was told that instructors often make

[31] It was not Purium, but nonetheless shared many features with it.

decisions that they know students won't like.

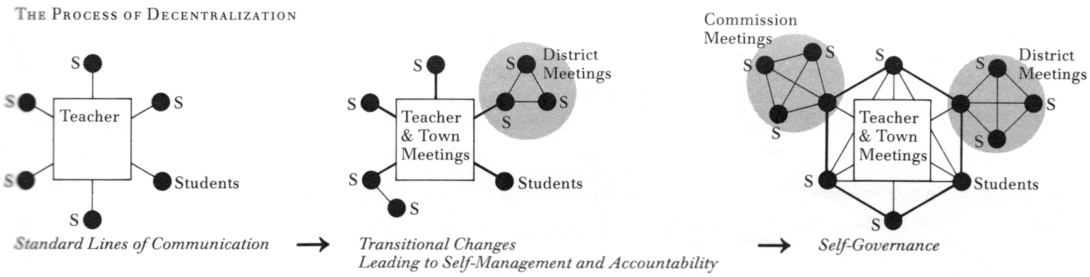


Figure 7.11: "The Process of Decentralization" (Nelson 1984).

And so the facilitator, assuming various forms within the simulated world, looms over play. She breathes life into the world and antagonizes its inhabitants. This pressure shapes the unfolding simulation in desirable ways, forcing tradeoffs and decisions to be made, and also restores authority to the teacher who has otherwise relinquished much of the power that she normally wields.[32]

As a result of all of these pressures and events, tradeoffs result, and players find that they have designed a city with qualities from their "Don't Want" criteria list. This can be "painful and frustrating," and Nelson writes that "this anger catalyzes" them to solve problems and imagine new possibilities. "City Building asks the students over and over again: 'What are you going to do about it?' " Another result of this investment and provocation is that in future City Building exercises, which aren't so tightly time constrained, students are eager to seek advice from community members, for example the mayor, mail carrier, firefighter, and real estate developer (Nelson 1984).

[32]City Building leans heavily on the strong bond between classroom governance and city governance, whose power structures mirror one another. Teachers can exploit this conceptual transfer, role play, and emotional investment to simultaneously give more agency to their students, while governing classroom behavior more closely than would otherwise be possible. This correspondence and governance effect can pervade all classroom activity.

Evaluation

Finally, the city is evaluated, and the facilitator guides all participants in a discussion about the city and its creation. The president arrives, and a tour is given by the mayor and commissioners. The students answer questions about the city, for example about housing, recreation, and pollution, and engage in critique, reviewing what they have made against their own criteria. An honest discussion about the planning process, successes, and mistakes follows. Individual and collective experiences are recorded, and they create a history wall documenting their collective experience.

City Building as Play

City Building Education is more than play. But if we look at the design of City Building Education, and examine how it scaffolds engagement and activity, play is everywhere. Play is in the scaffolding's braces, boards, and brackets—it gives strength to these structural elements. Scenarios and stylization guide the climb up into play's risky heights, making it feel safe. When students walk the boards, their eyes and feet are guided by make believe. And as the photographs make plain, designing, building, and inhabiting these imaginary cities is deeply appealing and pleasurable. Charles Eames took thousands of slides, reveling in the imagery that City Building Education provided. The city is a familiar and potent domain that takes on a profound holding power when transformed into a stylized miniature. The models are evocative, colorful, and inspiring. They captivate, not only as art, design, and architecture, but as living worlds one moves to and invests in.

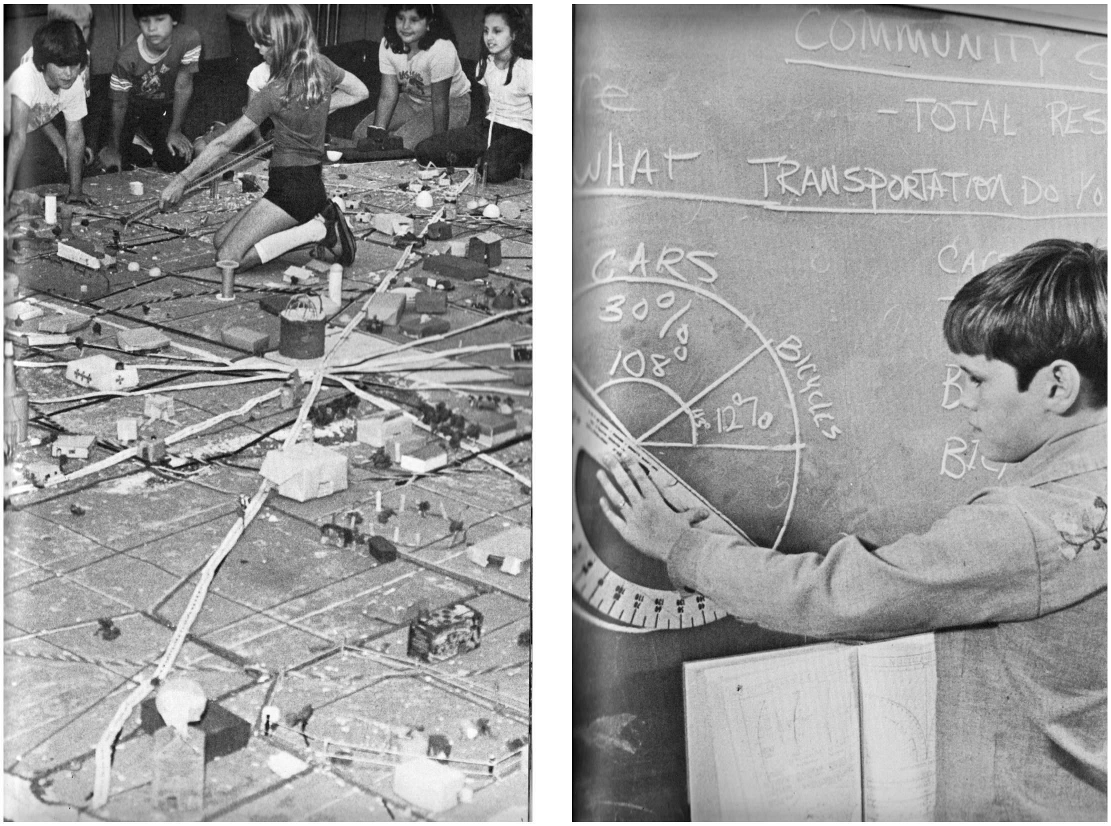


Figure 7.12: City Building Education photographs (Nelson 1982).

The Shared Threshold

Visiting a teacher training workshop at Pasadena Arts Center, I was struck by how the model city often sat at the room's periphery, yet the imagined city grounded and permeated all activity. I pointed out this paradox to Nelson, who affirmed it: "The city is nothing and the city is everything" (Gingold 2015). Immersion, of course, takes place in the mind—we actively create belief. Make believe is socially sustained and enacted. And yet the model city, a prop, performs a vital role. If the city exists in the mind, then what function does the model serve?

The model is the threshold of transformation. It is a gateway and anchor of immersion, linking reality and fantasy. The model city is like the fairy dust that casts the spell of immersion upon us, leading us into a shared trance of actively created make believe.[33] A new world is conjured by the stylized models, costumes, and roles. The model cities are mesmerizing, inviting, and appealing. Looking at photographs of them, it is hard to resist the pull across their imaginative threshold.

The model empowers and connects. The outside world, unseen in whole, is transformed into a miniature over which the students have power. An inversion takes place, linking the small classroom model to the city outside. And since the city is familiar, the model fuses the mundane with the novel.[34] Parts of the model city can be taken home, bringing your most intimate and familiar self into contact with new worlds. The make believe city is an aperture through which new subjects can be introduced and made relevant. Because students give shape to it, the model links the present to the possible, the here and now to what might be (Nelson 1984). As in Venice, a city of bridges and canals, a multiplicity of beings, places, and frames are connected. The model links imagination and reality, you and me, class with home, the here and now of the class with the built world outside, and worlds that might be.

---

[33]The model city seems to function as a *threshold object* of immersion (Murray 1997), as well as a *boundary object* (Star and Griesemer 1989), *boundary infrastructure* (Bowker and Star 1999), *liminal phenomena* (Turner 1982), and *transitional object* (Winnicott 1953).

[34]Nelson critiques the Vivarium project's use of fish. She asserts that cities are more familiar and exciting than fish (Nelson 2015).

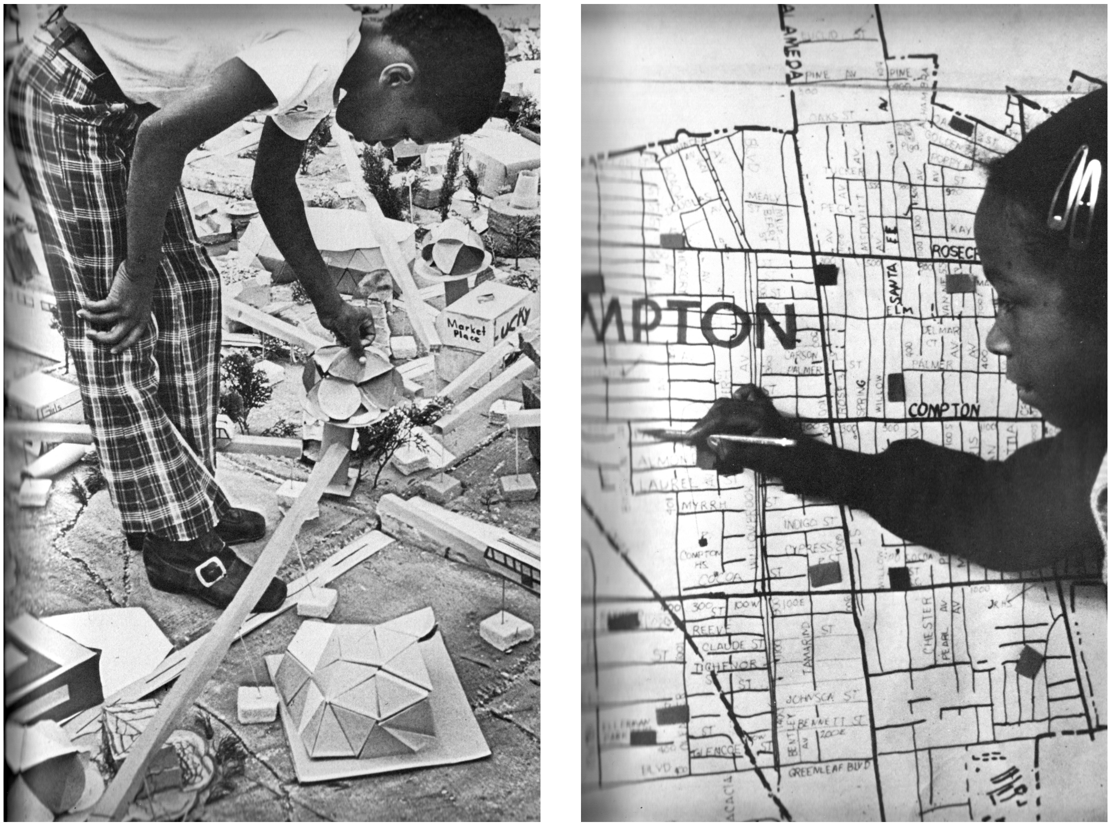


Figure 7.13: City Building Education photographs (Nelson 1982).

The tangible and embodied qualities of City Building create further linkages. Physicality integrates the imagination, attention, and activities of multiple players. The model anchors an imagined world by providing a concrete shared grounding. Multiple players design and role-play the city, fostering a sense of community and collective ownership. Individuals are linked into a community; the city is a place where people meet and live together.

The self is transformed. Costumes, like the model cities, integrate attention, pretense, and playful intent. A costume, whether a hammer outfit worn over the whole body or simply a name tag, synchronizes the meaning of transformations and enactments. Students become hammers, coke bottles, and M&Ms; citizens, commissioners, and mayors. An embodied and situated transformation of self is enacted by costumes, songs, dances, and interviews. Being another gives license and safety to further transformations. The city and the costumes are gateways: I am linked to a new me and a new you; together, we become a new community (Nelson 1984; Nelson 2015; Gingold 2015).

Nelson, as we have seen, was delighted to discover the profound degree of conceptual transfer the model cities afforded. Cities are hubs. Made of fixed infrastructure and mobile elements, they are places where goods, people, and ideas circulate. The model city also facilitates conceptual travel. The circulation of city traffic could be mapped to the circulatory system. The city as model helps us to enter, move about, and understand new worlds. The unknown is linked to the known. As an educator, this quality was deeply important to Nelson. It is learning. At a meta-level, City Building Education, the methodology itself, with all its evocative appurtenances, linked Nelson into new worlds. It not only connected Nelson, her students, and the surrounding community in new ways, but it also brought Nelson into contact with Maxis, the Smithsonian, the Eames Studio, and the Vivarium project. City Building Education's evocativeness allowed it to move between the worlds of education and art, schools and studios, museums and software publishers.

Open-Ended Scenarios

City Building Education strikes a balance between open-ended and guided play. Over and over, I was struck by the facilitator's restraint in guiding play. Nelson writes that

> [t]he importance of the children's discussion and resolution of their own problems, without the intervention of adult authority, cannot be overstressed. We addressed this entire program to the examination of decision making and the conflict or struggle it involves and to each person's potential as a teacher of others (Nelson 1974a).

The teacher's role is to provide "direction and leadership without rigid discipline" (Nelson 1974a). Scenarios, roles, and facilitation, like those we saw in Purium, function like an older playmate, guiding and inspiring make believe. At a meta-level, Nelson herself was open to scaffolding from outside experts—the architecture graduate students, the city simulation maker Richard Duke, and the architect Frank Gehry.

Design criteria are used to balance open-ended and guided play. Students enjoy the creative flexibility to craft open-ended solutions that meet the design criteria. Criteria are set by the students, who are guided through the process of identifying problems and establishing values, which allows them to bring their own perspectives into the mix. Evaluation is correspondingly open-ended. The facilitator simply guides students through measuring what has happened against their own stated criteria.

Open-ended play is guided by the facilitator's provocations. Nelson embeds problems that must be solve into the world, enacting disasters, pressures, and events. She asks students to reflect on their designs: "Whatever the kid builds, the question you ask is why? Why is this? Why is that? Why is this? How does this work?" One of Nelson's favorite things about *SimCity* is that the software enacts cause and effect, persistently asking the player *Why?* (Nelson 2015). This creates a strong "need to know … learning flows directly into application" (Nelson 1984; Nelson 2015).

Construction materials are not overdetermined. Students don't build out of roads and buildings, but scrappy craft materials like cardboard and tape. The familiar is defamiliarized, making space for new meanings. Through make believe and arts and crafts, they build a world, solve problems, and create new ones. In Nelson's eyes, this is the biggest weakness of *SimCity,* that players don't have the creativity to invent never before seen solutions to problems. *SimCity* doesn't allow players to invent new kinds of shelters or transportation networks.

City Building's open-ended qualities are also a liability. It challenges traditional school environments and power structures. Teachers, students, parents, and administrators must be on board, as it can be difficult to accommodate the time, space, and flexible focus of City Building (Nelson 1974a; Gingold 2015).

Emergence

City Building Education is a lever for appropriation and transformation. Like a prism, it refracts life. Players transform craft materials, scraps of cardboard, and odds and ends into shelters and transportation networks. People turn into new selves and new communities. The open-ended quality of City Building means that it flexibly takes on new craft materials, meanings, players, and topics. The model city weaves participants into a tapestry, bringing together diverse perspectives, knowledge, and interests.

City Building is emergent. Elementary units of space, individuals, structures, and roles are knit into complex interconnected systems. The city is a collaborative enactment. Because the model is divisible, you can take part of it home; and yet, the city remains a unified whole. Through transformation and collaborative role play, disparate agents are interconnected into emergent wholes. Heterogeneous perspectives, desires, and expertise are interlinked. The emergent whole, here, is not apprehended from the outside by an external and omniscient simulation maker and player, but from the inside, as one of many loose parts at play.

Chapter 8

Adventure Playground

> Adventure playgrounds are perhaps the most revolutionary experiment we know for absorbing the interest and releasing the energies of young people. Children the world over have a deep urge to experiment with earth, fire, water and timber. They need to be masters of the materials to hand and be free to move them around to suit their own desires and to create their own order out of seeming chaos. They delight to work with real tools, to use them in their own way and at their own pace without criticism or censure. Their love of freedom to take calculated risks is recognised and welcomed in adventure playgrounds for these qualities bring their own exhilarating sense of independence and adventure. As Ibsen has said 'There is always a certain risk in being alive, and if you are more alive there is more risk'. The children feel liberated in an adventure playground, especially those who live in crowded cities or in over-regulated and over-tidy housing estates. Adventure playgrounds are places where they can test themselves against new challenges in complete freedom and where they can learn to come to terms with the responsibilities of freedom.
>
> —Lady Allen of Hurtwood (Bengtsson 1972)

Introduction

In this chapter we turn from play with miniature cities—*SimCity* and City Building Education—to play within real ones. This chapter looks at an unusual playground, the Berkeley Adventure Playground, in which children engage in activity that is prohibited elsewhere. They dig, build with scrap materials, paint, saw, play with mud, and engage with materials and risks that are typically withheld from them. If the city robbed children of developmentally beneficial and natural play, then the playground, and especially the adventure playground variant, was compensation.

The adventure playground genre, which originated in 1930's Denmark, offers an excellent case study of play in the wild. Such play isn't just wild in the sense that it transgresses a collective sense of appropriateness. Wild means more than mud and danger. Wild also means out there, in the world, untethered from thought experiments, carefully controlled laboratory settings, and reflective thought about play and experience. Adventure playgrounds are a play tradition that is analyzable empirically, historically, and within specific social and cultural contexts.[1] In addition to drawing upon existing accounts by scholars and practitioners, I visited the Berkeley Adventure Playground as an ethnographer and interviewed key participants.

Such playgrounds have popped into and out of existence for over 70 years, yielding ample examples for analysis. They reveal a playground concept braided with contradictions. In looking at any particular playground, we see that its fate and composition is inextricably intertwined with the broader community. Playgrounds are play spaces set apart from their context, yet the boundaries that surround them also link them into the wider world, buffering and protecting them. These qualities, while present in all playgrounds, come to the fore in adventure playgrounds.

Adventure playgrounds vividly illustrate design techniques for scaffolding play that is embodied, messy, risky, autonomous, and imaginative. These playgrounds call attention to the

[1] My use of "in the wild" follows Hutchins (Hutchins 1995), who analyzes the distributed cognitive work of navigation on a Navy ship. This is in contrast to the *in the laboratory* psychological experiments that typify cognitive science. My point here is that adventure playgrounds are useful case studies for studying how play is situated in and participates in the broader world. These are not, in other words, thought experiments or laboratory examples of play.

problems that play design must address, and illustrate design techniques for doing so. As evocative phenomena, they help us to think about central themes of play design: separateness and boundaries, risk and safety, the balance between autonomy and guidance, and the nature of appropriation.

Berkeley Adventure Playground

Figure 8.1: Berkeley adventure playground. Photo courtesy of Patty Donald.

The Berkeley adventure playground sits on the city's edge, on a marina shoreline overlooking the San Francisco bay. It lies between the ragtag U.C. Berkeley sailing club—a motley collection of industrial shipping containers, sailing craft, and port-o-johns—and a typical rubberized playground set within a pretty green space flecked with picnic tables. When active, the adventure playground is filled with the noise of hammering and squealing. Multiple languages can be heard, and the rambling yard proudly deserves the “junk playground” title once given to such playgrounds. Five star reviews on Yelp celebrate it as “like the backyard of a white trash Brady Bunch[;] Awesome,” or “Lord of the Flies and a kid-version of Survivor,” or “a little shanty town.”[2]

The earth is packed from the footsteps of countless children. Ramshackle wooden structures populate the place, beautiful in a post-apocalyptic sort of way. Tall overgrown plants, weeds, and flowers grow in exactly the right places. From here you can smell the salty ocean air, and on some days the savory smells of barbecue from picnickers outside. On my first day at the marina, just outside the playground, I meet a sanitation worker, a young man in his late 20’s. He is dressed in an official park uniform and a formidable pirate hat, and speaks with an Italian sounding accent.

The playground is part of the Berkeley Shorebird Park Nature Center, whose program coordinator is Patty Donald. Donald is a tall, slender, middle aged woman with the outgoing, energetic, and emotive bearing of an actress—an avocation of hers. I first met her in person on a warm, sunny morning at the Nature Center, where she has scheduled a simultaneous interview with a young Taiwanese man studying industrial design in San Francisco, who has brought styrofoam scale models of playground equipment he is designing, and myself. We are surrounded by aquariums, folding chairs, and the accessories of natural history education. Donald’s dog wanders in and out of the building while we talk. It’s a quiet, sunny day, probably because it’s a pre-summer Monday, and bird chatter fills my interview recording. From this building Donald runs educational programs and administers the adventure playground that she has been supervising since 1979 (Gingold 2012; Donald 2015).

[2]Yelp review quotes were pulled in spring of 2012. This was also the time of my initial visit to the playground.

I have come here to interview Donald about her experience. My account of adventure playgrounds in general, and the Berkeley playground in particular, draws upon interviews, ethnographic observations, historical research, and various published sources. In the course of researching the Berkeley Adventure Playground I interviewed Donald a couple times, and received her permission to volunteer at the playground to make ethnographic observations. I was able to speak with her staff as well as her daughter, who was then, conveniently for me, a college student at the University of California Santa Cruz. (Donald had suggested I do so, since she had grown up on the playground.) Donald also generously shared her photographs (but only those with non-identifiable subjects) as well as other ephemera that she has collected.

Figure 8.2: Climbing net at Berkeley adventure playground. Photo courtesy of Patty Donald.

One admiring Yelp reviewer, reminiscing about childhood experiences at the playground, writes that

> …all your child life you are told to not run with scissors. You're banned from the garage when your dad is working with his tools. You're weren't allowed to play with sharp objects. No parent would ever allow their child to play with tools. … Then one day my father took my brother, sister and I to Adventure Playground. …wondrous of wondrous places!!! …Are you a little girl and want to roll around a play in the mud and imagine your life as a bikini clad wrestler? Shit…you can do that here too!! Or practice your James Bond moves by flying across the park on cable only to land on a pile of dirt.

Not everyone approves of such playgrounds, however. A dissenting Yelper writes that "The parents who praise this place seem to share a cultish, abstract notion of "creativity" that is divorced from any child development theories I know of." In fact, the Berkeley playground is one of the last such remaining playgrounds in the area, not to mention the United States, and might be one of the few adventure playgrounds in the world that still grants this level of agency to children. Adventure playgrounds have disappeared due to concerns about liability and NIMBY (Not In My Back Yard)—property owners and developers concerned about property values (Wiederholt 2006; Donald 2012; Frost 2012). These playgrounds have died out because of the risks and junk aesthetics that constitute them.

The fragility of these playgrounds is a longstanding issue, dating back to their genesis in occupied Denmark. But to fully understand the motivation, operation, and appeal of the adventure playground concept, we must look at its formation. And to understand what makes an adventure playground a special kind of playground, and why they were created in the first place, we must first know something of playgrounds in general.

History of Junk Playgrounds

Playgrounds

Playgrounds are a response to the problem of children, modernity, and urbanization. They are motivated as much by social justice as social control. In the 19th and early 20th century, childhood came to be seen as a special and separate time of life. Rather than put them to work in factories or homes, they were given room to play and learn. But there was a shadowy side to this newfound freedom. In cities, free ranging children engaged in dangerous street play and disruptive activity (juvenile delinquency). In response to changing attitudes and circumstances, schools and playgrounds were created, interventions that sought to protect, nurture, and control children. Reforming elites set up such special zones to accommodate modern childhood, completing the buffering of children from mainstream life. Free ranging children needed to be reined in, their activities directed towards uplifting ends.[3]

---

[3]The 19th and early 20th century ushered in many changes to the perception and treatment of childhood, ultimately setting the stage for public playgrounds. These changes arose from shrinking birth rates—more attention was devoted to fewer children—and a concomitant shift in attitudes. Children were seen as needing special protection, shelter, and cultivation, and childhood came to be seen as a distinct phase of life with its own requirements. Play began to be seen as developmentally healthy, natural, and necessary, and parents began to withdraw children from work so they could play and learn. Child labor diminished, and as a result children had more unstructured free time. Children became economically and socially detached, setting the stage for interventions like schools and playgrounds, which Sutton-Smith describes as a "colonization of children into the main stream of American life."

Experts advised the harnessing of play with toys, books, and activities towards morally uplifting and developmentally beneficial purposes. To the extent possible, children appropriated all these toys, materials, and newfound freedoms to their own ends. We will see this basic pattern repeat: children are given more autonomy, but this autonomy must then be kept in check and directed towards safe, healthy, and uplifting ends. In turn, children appropriate and challenge the socially acceptable play scaffolds given to them.

Children, and boys especially (girls had more work obligations), received unparalleled freedom to play and run amok in cities. But this brought with it a host of anxieties about juvenile delinquency and street play—seen as dirty, dangerous (e.g. traffic accidents), and a nuisance. A rational recreation movement arose in response to such anxieties about industrialization, urbanization, poverty, and the leisure of the urbanized poor (e.g. drinking and street play). The efforts of these reforming elites gave rise to Sunday schools, the temperance movement, municipal playgrounds, youth camping movements (e.g. boy scouts), social clubs (e.g. YMCA), public libraries, family oriented music halls (replacing male-only signing saloons), cultural complexes (e.g. museums), and so

Sutton-Smith compares the modern playground to a zoo "that has failed to keep in its animals." He writes that

> playgrounds, first seen as an anti-thesis to the state of was [sic] in the streets, ultimately achieved much of their purpose in taming their animal members. Children who had ravaged the streets and been everywhere and underfoot were by playgrounds, (etc. etc.), rendered marginal in the ongoing life of the larger culture… (Sutton-Smith 1980)

Reformers sought to appropriate, contain, and channel play through playgrounds in public parks, school yards, and factory grounds. While play was seen as intrinsically natural and developmentally healthy, it was undesirable when it impinged upon modern life. In this sense, playgrounds are like zoos or wildlife reserves. Just as a wildlife reserve offers sanctuary to animal life threatened by urbanization, and protects city dwellers from wild beasts, playgrounds offered a special buffered zone in which a natural urge could safely express itself. Playgrounds scaffold the expression of natural play behavior, but in a way that mitigates the threat both to players and non-players. They do so by appropriating and regulating play perceived as problematic.

But wildlife reserves and playgrounds are designed by experts. Opposites are fused: playgrounds are both natural and designed, free yet directed. But how does one guide an au-

---

on. Playgrounds can be seen as part of a broader reformist movement to rationalize play, and direct it towards positive ends.

Playgrounds were one of many ways in which reformers sought to rationalize play—especially that of poor immigrants to American cities—appropriating problematic play, containing it, and directing it towards safe and developmentally normative ends. Unrestricted urban play by children was problematic, but playgrounds could, it was hoped, contain and direct child's play into developmentally and socially appropriate forms. The playground as a concept descends in large part from the work of Froebel, the inventor of kindergarten, whose students engaged in free and directed play, with a particular emphasis on play with the natural world. Playgrounds were attached to schools, placed in parks, and sometimes even on the premises of factories—outlets for a play drive identified by reformers and psychologists (Sutton-Smith 1980; Cross 1990; de Coninck-Smith 1999; Chudacoff 2008; Frost 2010; Frost 2012; Latané 2013).

Social justice and social control can be seen as entwined. Playgrounds, according to Roy Kozlovsky, "are very much about censoring and restricting types of play deemed undesirable and displacing them from places deemed dangerous or corrupting, such as the street." Through playgrounds, modern societies seek to reconcile "the subjective experience of play … an attribute of the autonomous, individual self" with a desire "to rationalize and shape children's play from the outside to advance social, educational, and political goals" (Kozlovsky 2007).

tonomous activity? Like a wildlife reserve, a playground is both natural and contrived. How to make it both wild and reserved?

Junk Playgrounds

The playground is haunted by the question of what counts as appropriate play. If the problem, at least in part, was that the city was dirty, then the solution should incorporate some form of nature.[4] In 1931 the Danish landscape architect Carl Sørensen published a booklet that articulated an answer to the question of what particular form urban play should take (Sørensen 1931). Instead of taking the urban child to the countryside, the country and its opportunities for play would be brought into the city. In his booklet, entitled "Parkpolitik" (Park policy), Sørensen proposed what he called a "junk playground."

> Perhaps we could try to arrange some sort of junk playgrounds on fairly large and suitable areas where the children were allowed to utilise old cars, packing cases, brushwood and that sort of thing. It is possible, of course, that there will have to be attendants there, both to hinder bad instances of intolerance and to prevent possibilities of the children getting hurt. Probably, however, such attendance would be unnecessary (Sørensen 1951).[5]

Sørensen sought to reconcile the wildness of children's play to the limits of the city. He wished to legitimize play activity he had observed children engage in on junkyards and construction sites (Kozlovsky 2007)[6], the kind of play afforded to country children, but which city

[4]This is how some reformers in the United States and Europe saw things, taking a cue from the emphasis on natural play articulated by Dewey, Froebel, and Rousseau. They took city dwelling children, often poor, on trips to the countryside, which was seen as uplifting, and, in the case of Denmark, sometimes seen in nationalistic terms (Cross 1990; de Coninck-Smith 1999; Chudacoff 2008; Frost 2012).

[5]This quote is from (Sørensen 1931), but is quoted in (Sørensen 1951).

[6]In the Sørensen materials I have accessed I do not see him directly drawing an arrow between his observations of illegitimate urban play on marginal sites like junkyards and the junk playground, but it is commonly attributed to him in secondary and tertiary sources (e.g. Allen 1968 p. 54 and Kozlovsky 2007). Even if it isn't literally true, as a fable it reflects the values and history held as true by playground and adventure playground advocates.

children were deprived of. "[C]ountry children," wrote Sørensen, could play with "twigs, old boards and bits of wood, bricks, tiles, drain pipes, discarded automobiles, boats, waggons [archaic], wheelbarrows and other kinds of junk." They could "dig in the earth," "work with houses and caves," and "build towers." Illegitimate city play was really just a distorted expression of a natural play drive that manifested more easily in the countryside. Such play could be brought into the city, safely contained in a playground, and supervised by attendants (Sørensen 1951).

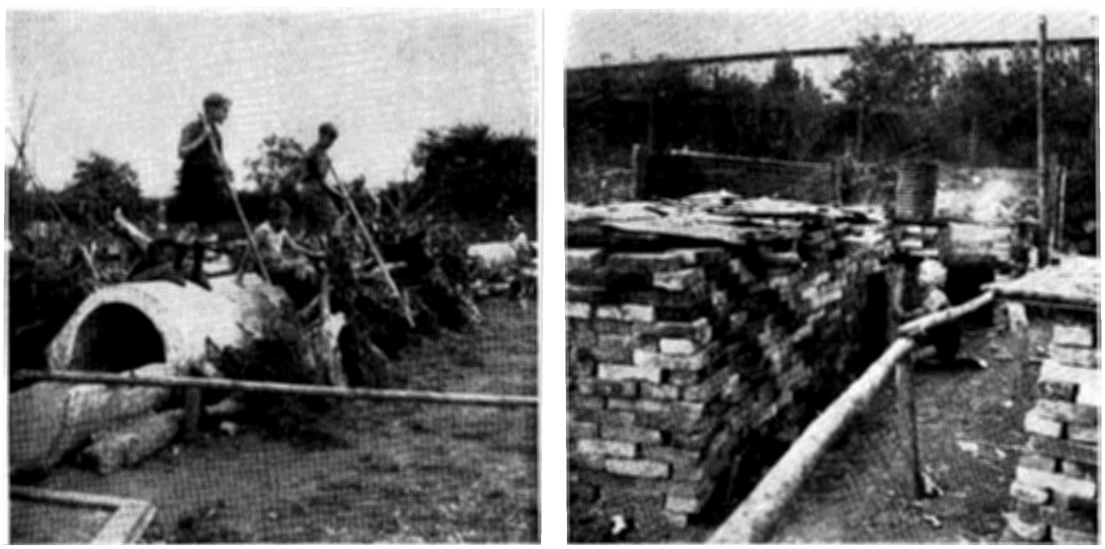


Figure 8.3: Scenes from Emdrup (Sørensen 1951).

The junk playground concept was first put into practice in 1943, during the German occupation, at a housing estate called Emdrupvænge, in Copenhagen. The residents insisted on an attendant, and so John Bertelsen was hired: the first junk playground play leader (Sørensen 1951; Kozlovsky 2007).

Florid creativity gushed into and gave body to the playground. The children, according to Bertelsen, built "a little town of small stone huts" and "small bungalows" with watertight roofs made with grass and soil. An old car was used for make believe: "six mouths produce the sound of a humming motor–and they are already a good way down the high road of imagination on their way to distant and unknown destinations." The children loved to dig caves. Some had roofs, antechambers, and fires for cooking and brewing tea that one might drink out of a "sooty metal mug." Caves were inaugurated through many "curious" "celebrations" and rituals.

> At some of these celebrations wax candles are lit, whilst oatmeal is fried over the fire and served boiling hot with homemade rhubarb jam. Others go so far as to serve roast pork and potatoes, which are eaten straight from the pan and pots by forks made out of flat pieces of wood. One thing is common to all inaugurations, whether of a cave, a house or a tower, and that is the pleasure shown by the children at the results of their own efforts (Bertelsen 1953).

Here they could "make mankind's basic discoveries" (Bertelsen 1953). Ning de Coninck-Smith writes that indulging the desire to "build dens, climb, fight, throw things and hunt" grew out of contemporary developmental psychology. Recapitulation theory claimed that children, as they develop, enact man's trajectory through more primitive evolutionary states. Children, she writes, were acting out an "urge to be like the ancestors of humanity" (de Coninck-Smith 1999).[7]

Sørensen's concept emphasizes the imagination of the child. He liked how the sandbox enabled children to create and shape their environment, and envisioned a radical expansion of this idea. He wanted a playground in which children could "dream and imagine and make dreams and imagination a reality" (Sørensen 1951).[8] Emphasizing the creativity of the child

---

[7]This was in line with the advice and philosophy of experts such as Stanley Hall and Luther Gulick, who advised on healthy play and development, and advanced a "recapitulation theory" in which individual development recapitulated humanity's transformation from the primitive to the civilized.

De Coninck-Smith quotes Bertelsen, Emdrup's play leader, who wrote in an article that "[i]t is now not easy to be a child in the city when you feel the urge to be a caveman or a bushman" (de Coninck-Smith 1999).

[8]The sandbox suggested an environment that offered "children possibilities for creating and shaping," but was, Sørensen noted, "rather limited." And the "seesaws, swings, roundabouts and such appliances" of a typical playground were also quite limited.

diverged from contemporaneous playground designs that emphasized the creativity of the playground's architect (Sutton-Smith 1980; Solomon 2005; Kozlovsky 2007; Frost 2012).[9] Sørensen, instead, analyzed children's play functionally, and designed his playground around their behavior (Kozlovsky 2007). What resulted was "an antithesis to the routine character of both cities and schools" (Sutton-Smith 1980). By appropriating the junk play of children and giving it a legitimate outlet for expression, the imagination of the child, rather than the architect, is expressed. The aesthetic implications are significant: instead of the simple and abstract forms that typify playground designs, the junk playground looks, well, like junk.

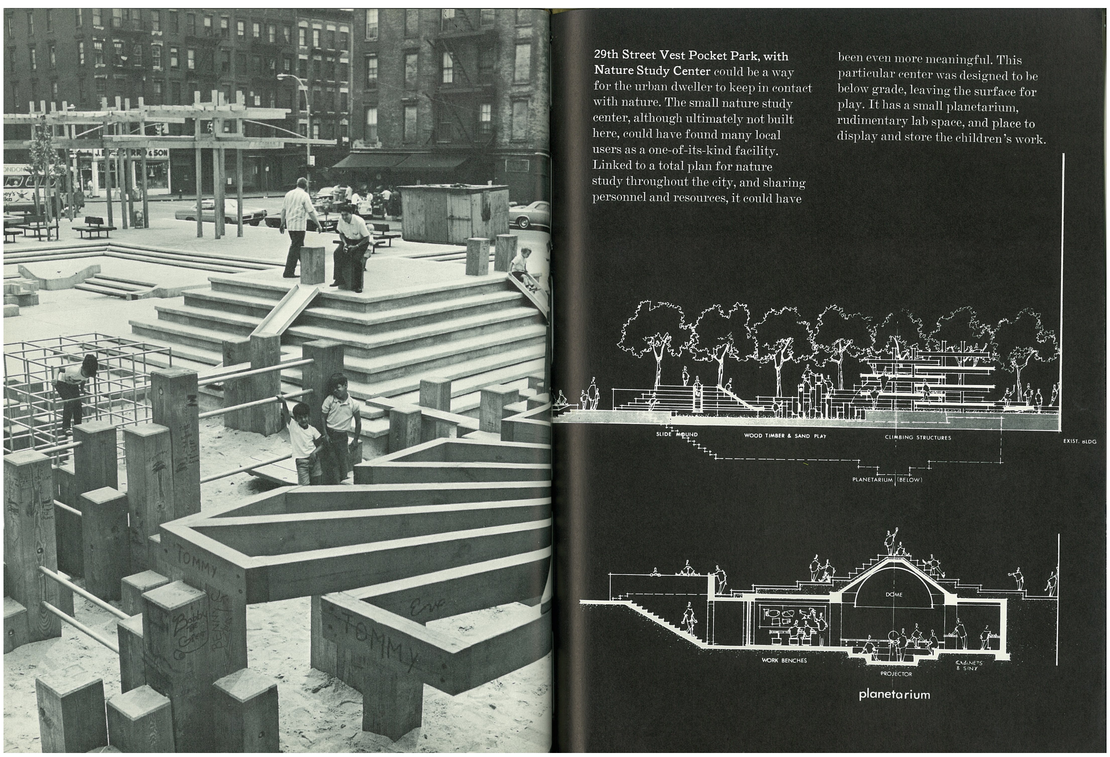


Figure 8.4: Example of an modernist playground design by M. Paul Friedberg. A geometric and sculptural order prevails (Friedberg 1970).

[9]Around the time that Sørensen proposed his junk playground, architects and sculptors such as Isamu Noguchi and Aldo van Eyck designed playgrounds in the modernist "aesthetics of abstraction" (Kozlovsky 2007), examples of "painting in concrete … a communication from architects and designers about the way the structured world might look" (Sutton-Smith 1980).

This, of course, caused problems, as not everybody appreciated the aesthetics of junk playgrounds, or the fact that their kids got dirty and muddy (de Coninck-Smith 1999). To make space for what has been forbidden elsewhere is still to engage the prohibited. Sørensen was well aware of this, and wrote that "of all the things I have helped to realize, the junk playground is the ugliest; yet for me it is the best and most beautiful of my works." Anticipating the protests of neighbors that would forever haunt future junk playgrounds, Sørensen, a landscape architect, included in his design for Emdrup a two meter high dike that encircled the playground, topped with a fence, and planted over in a "thicket" of "wild roses, hawthorn and acacia." The children got their wild play, and the neighbors, instead of an unsightly mess, got wild roses (Sørensen 1951).

Adventure Playgrounds

Following Emdrup, the junk playground was replicated, spread by proponents such as Lady Allen of Hurtwood and Arvid Bengtsson. Like Sørensen, Allen was sensitive to critiques of the junk playground concept, and rebranded it the "adventure playground" (Kozlovsky 2007; Frost 2012).

Allen had married the British politician and conscientious objector Clifford Allen. Following her husband's death, she worked throughout World War II on behalf of international pacifist efforts, and promoted the interests of children, who she saw as a group that transcended divisions of class and nation. It was while pursuing this work that she visited the junk playground in Copenhagen. Trained as a landscape architect herself, Allen was taken with the idea, especially its emphasis on democratizing creative agency (Kozlovsky 2007; Wilson 2013).

Back in England, Allen promoted the concept, and advocated the conversion of bomb sites to junk playgrounds. The debris, after all, was an ideal material for creative play. In a 1946 newspaper article entitled "Why Not Use Our Bomb Sites Like This?," Allen writes that

> Juvenile delinquency and the death of young people in road accidents both arise, in part at least, from the inadequate and unimaginative manner in which local authorities try to meet the need for creative play. . . . The best the Borough Engineer can do is to level the ground, surface it with asphalt, and equip it with expensive mechanical swings and slides. His paradise is a place of utter boredom for the children, and it is little wonder that they prefer the dumps of rough wood and piles of bricks and rubbish of the bombed sites, or the dangers and excitements of the traffic (Allen 1946; as quoted in Kozlovsky 2007).

These playgrounds enacted a narrative and process of reconstruction (Kozlovsky 2007). The detritus of war became the soil in which children's play took root, the budding growth of a postwar renewal. As in Copenhagen, the playground was a balm for the deprivation, wreckage, and psychic wounds of wartime. Allen's rhetoric reveals a belief that the freedom afforded to children on such playgrounds went hand in hand with democratic citizenship. "Adventure playgrounds," she wrote, "are places where they [children] can test themselves against new challenges in complete freedom and where they can learn to come to terms with the responsibilities of freedom" (Bengtsson 1972).[10]

[10]Kozlovsky applies a Foucauldian frame to this idea, arguing that the mix of oversight and freedom on such playgrounds mirrored broader social strategies for cultivating liberal democratic citizenship. Children and local communities were given both supervision and agency in rebuilding, inculcating them with an internalized set of social controls and desires (Kozlovsky 2007).

Legislation passed in 1944 converted bombed sites into playgrounds. (Many of these playgrounds lasted until the 1950's, when property owners redeveloped the sites.) Sensitive to the needs of working class families, in which older children supervised younger children, Allen was interested in play environments that were not overly specific in terms of age and ability (Kozlovsky 2007). The British playworker Penny Wilson calls this "graduated access"—offering play spaces and activities with overlapping meanings, challenges, and skill requirements (Wilson 2010).

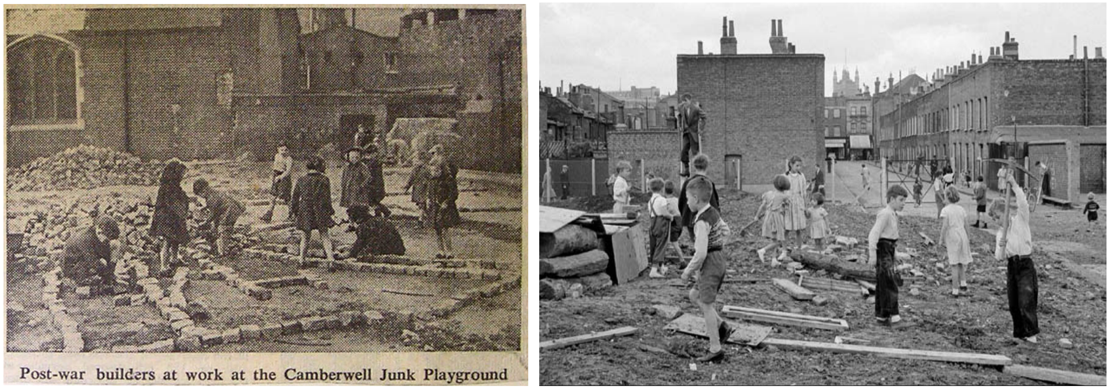


Figure 8.5: English adventure playgrounds on the sites of a bombed school (left) and church (right) (Kozlovsky 2007).

Berkeley

The first American adventure playground, dubbed “The Yard” or “Yardville,” was established in Minneapolis by the magazine McCall’s. Built in 1949, and inspired by the Scandinavian examples, the playground was the focus of a 1950 McCall’s cover story, and received a visit from President Truman (Steller 2014). The playground was shut down after a year—neighbors feared the kids’ shanty town was ruining real estate values (Sutton-Smith 1980).

Figure 8.6: Huntington Beach adventure playground (“Adventure Playground: Park Design Modification University Community Park”.)

The adventure playground that sprang up in Huntington Beach California was built in a quarry that children referred to as the “pit” that had housed illicit play activity, such as hiking and fishing. By creating an adventure playground, the parks and recreation department appropriated the site and its illicit play activities, legitimizing, structuring, and supporting them. Inspired by the Huntington Beach example, the city of Irvine dug their own pit in a city park, and, as in Emdrup, used the excavated material for a berm (Vance 1982).

But the American adventure playground is an endangered species. As of 2012, only three exist in California (Frost 2012). According to experts, the Berkeley adventure playground is the epitome of contemporary adventure playgrounds. In 2012, the Berkeley playground saw 65,000 visitors a year, almost twice the number of visitors ten years past.[11] On the first day I visited, a summer Sunday in 2012, the playground counted 427 visitors (Donald 2012; Latané 2013).

Berkeley offered a unique combination of circumstances that was friendly not just to initiating an adventure playground, but sustaining it. It sprouted, in 1979, within the special Marina Enterprise Zone, nourished by a propitious confluence of forces. It was created, alongside a now defunct adventure playground in San Francisco[12], after a weeklong workshop hosted by the American Adventure Play Association, a non-profit founded in 1974 to promote and disseminate information about adventure playgrounds.[13] AAPA covered insurance for the first year of operation, helping to get the playgrounds off the ground, up and over the concerns of liability skeptics.[14] After a year, with concrete data on the low accident rate in hand, the City

[11]This number is almost twice that of the Irvine adventure playground during its heyday (Latané 2013).

[12]In Fort Miley (Donald 2016).

[13]Bill Vance, one of the founders of the organization, wrote the brochure that Berkeley city parks used in the design of its playground, a copy of which Donald gave to me.

[14]In doing so, they underwrote their own claims of low risk (e.g. Vance 1982).

of Berkeley brought the adventure playground under its own citywide insurance policy (Vance 1976; Hogan 1982; Vance 1982; Donald 2012; McManis 2013; Donald 2016).[15]

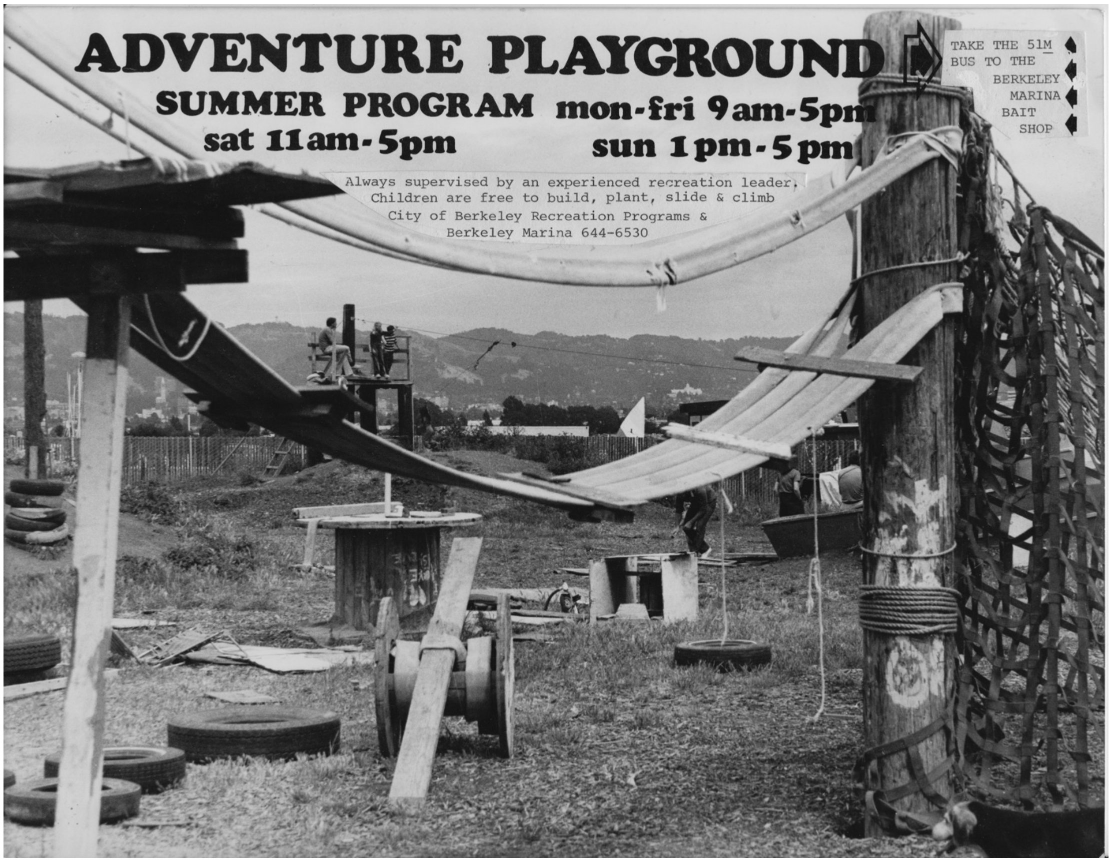


Figure 8.7: Flyer for Berkeley adventure playground, circa mid 1980's. Courtesy of Patty Donald.

The playground's structural liminality—neither part of Berkeley parks nor the marina—not only helped give rise to it, but sustains its survival. Like a plant growing through a sidewalk crack, the playground thrives in this in-betweenness. It sits on the cusp of the ocean, at the edge of the San Francisco Bay, in a special Marina Enterprise Zone. It is a joint endeavor between two bureaucracies: the marina, supervised by Bill Montgomery, and Berkeley parks and recreation, headed by Frank Haeg. (Donald was working for Haeg at the time, running after school programs, and has been administering the adventure playground practically since its

[15]It wasn't until 15 years in that participants had to sign a liability waiver (Donald 2016).

inception.) While the city covers insurance, operating expenses for the playground come from the Marina Fund, which is independent of the city, and belongs to the Marina Enterprise Zone. This silos the playground's finances from the city General Fund, giving it a degree of protection and independence. But Donald is an employee of the city, which pays for its insurance. Not only is she tasked with ensuring safety, but she must safeguard the city's liability by monitoring how she represents it.

Ideologically, the City of Berkeley supports the playground because they want to be on the cutting edge—outside of the norm. The marina's agenda also motivates the playground. In addition to amenities for boaters and diners, it is required by the state to provide recreational services for families. The playground is able to do so very cheaply, which is a powerful motivator of its creation and survival. In 1978, the year before it was established, California voters passed Proposition 13, the "People's Initiative to Limit Property Taxation."[16] All of a sudden, Donald says, "there was no tax money paying for recreation anymore, and so they had to come up with inexpensive ways to accommodate a lot of people" (Donald 2012; Kamen 2012; McManis 2013; Kamen 2015; Donald 2016).

Geographic marginality also helps to protect it. According to Donald,

> it doesn't have the issue that all the other adventure playgrounds have of being in the wrong place. It's always in a place nobody wants. And that's the best place for it. Except that people keep wanting their places. Nobody wants this land down here for anything at this point. So, it's not threatened. But most of the adventure playgrounds that were in neighborhoods were closed down because the neighbors didn't like the noise, and the kids, and the banging, and the parking, and the issue that they felt it brought down their property level, property worth. … the NIMBY issue (Donald 2012).

The playground's survival is predicated on being in a developmentally undesired area and

[16]Proposition 13, the "People's Initiative to Limit Property Taxation," had just passed in 1978, a revolt by suburban Californian homeowners against taxes and rising real estate values. Property values were reversed to 1975 levels, and held close to that level, throwing the balance of power in the state towards homeowners, especially wealthy white suburbanites (Davis 1990; Wikipedia, California Proposition 13 (1978), 2016).

a need to accommodate budget shortfalls. It is structurally siloed within the Marina Enterprise Zone, poised between two bureaucracies. It emerged from the overlapping agendas of multiple actors, from the AAPA to the City of Berkeley. The playground is buffered from the surrounding area, a design consideration present since the original Emdrup playground's rose planted dike. This buffering is not just spatial, but bureaucratic, and emanates from the playground's marginal status. Despite existing within a separate, bounded area, the playground is articulated with the outside world in complex ways.

Tensions and Compromises

If the playground—a response to the problems of children's free wheeling play across the city—contracted, focused, and controlled children's play, then the junk playground re-expanded it. It sought a new compromise between the city and children's play. In this expanded view of the playground, kids play around, dig, build, make a mess, play with fire, and cook once again—but this time within the confines of a play yard.

But this is a compromise. "Any playground," says Sutton-Smith "is always liminal" (Sutton-Smith 1980). Playgrounds, and adventure playgrounds in particular, attempt to resolve opposing forces. Due, in part, to the intensity of forces under negotiation, the compromises eked out offer many vivid lessons in play design. I divide these up into four themes:

- **Boundaries**. Boundaries mediate between inside and outside. The boundary, a response to leakage, points to an ongoing dynamic tension between players and non-players. The city and its residents are left in peace and safety, and the children are allowed to play.

- **Risk**. Risk is desirable, as is safety, and so the two must be reconciled. Risk means agency and excitement, which are desirable. But safety is necessary to begin and sustain play. Potentially disruptive and harmful play is partitioned off, keeping both children and the outside world safe, protected from perceived hazards and disruptions. Children, parents, and the City of Berkeley celebrate risk and liminality while simultaneously minimizing it.

- **Autonomy**. Free vs. guided play. How is it that play, an activity thought to be free, is guided? Someone, after all, must design the playground—in this case landscape architects, or at the very least allocate space for it. Playleaders (or playworkers) help accommodate freedom to constraints, and form an indispensable component of adventure playgrounds.

- **Appropriation**. The playground scaffolds the ingestion and imaginative reconfiguration of what is not play—like junk, a bomb site, landfill, or gravel pit—into play. Transcending play's mutation of detritus into a resource for play, appropriation also encompasses the resiliency and adaptability of the adventure playgrounds themselves.

I now take up these themes in more detail.

Boundaries

Collapse

Adventure playgrounds are always at risk of collapse. The playful inversion that transforms junk into beauty contains within it the seeds of its own destruction.[17] While Sørensen found

[17] "The life span for most American adventure playgrounds," writes Frost, "was short, due to concerns about junky appearance, expansion of safety regulations, fear of injury and liability, shortage of funding and play leaders,

the junk playground to be the "ugliest … and most beautiful" of his creations (Sørensen 1951), this sentiment isn't always shared. The surrounding communities might not see the beauty in the junk, seeing instead depreciating property values. Outside the inversion, people still see junk as junk. Allen's rebranding of the "junk playground" into an "adventure playground" is a clever bit of marketing, but the junk persists. The legitimization of forbidden and desirable play introduces powerful tensions between players and non-players. And so, there is the risk of collapse.

Figure 8.8: Dattner's "Adventure Playground" in New York City's Central Park. Photograph by Richard Dattner (Trainor 2012).

Some adventure playgrounds fail before they even begin. New York City's Central Park boasts an acclaimed playground designed by the architect Richard Dattner. Dattner's play-

and lack of support from community leaders. Despite their strong reputation among developers, child users, and involved parents, most disappeared but a few model examples remain" (Frost 2012).

ground was intended as an adventure playground, and is referred to as an “adventure playground,” but it is in name only. Although the funders and some community members, a group of mothers, wanted a proper adventure playground, the project foundered over the objections of other community members.[18] “[T]he doomed project,” Dattner writes, “yielded one very important lesson: the community must be fully involved in a project from its inception.” The celebrated playground that emerged from this process was, in the end, an adventure playground in name but not spirit. A fresh modernist architectural hymn to play, but not a junk wonderland (Dattner 1969 [1974]; Trainor 2012).

Figure 8.9: Protesting the closure of Irvine’s adventure playground (“Adventure Playground: Park Design Modification University Community Park”).

Sometimes adventure playgrounds devolve into more benign playgrounds. The landscape architect Claire Latané, working on behalf of EPT Design, took on the project of reimagining Irvine California’s shuttered adventure playground. Latané’s background in community participatory design and sustainability uniquely situated her to reconcile the conflicted desires of the Irvine community. While it may seem incongruous that an adventure playground would appear in Irvine, the prototypical suburban city, such experimentation was very much in the early spirit of suburbanization—which was, after all, an experiment in city planning. Innovative planners established at least two adventure playgrounds in the city. One of them began as a mud pit. In

[18]The Estée and Joseph Lauder Foundation became interested in underwriting an adventure playground, joined forces with Dattner, the New York City Parks Commissioner, and a civically engaged group of mothers hoping to reform a playground on the west side of Central Park.

the late 2000’s this last playground was temporarily closed so an adjacent playground could be renovated, but then advocates of the adventure playground noticed a bulldozer sitting on their beloved site. A public outcry followed, which ultimately resulted in EPT Design being brought in to mediate between advocates and the city council, with its concerns about environmental sustainability (stormwater drainage issues), accessibility (to people of all physical abilities), cost (popularity meant high staff costs), health codes (mud), and building codes.[19] The resulting playground design employs innovative ecological ideas, but the adventure has been defanged. The city council was adamant: no more mud (Latané 2013).

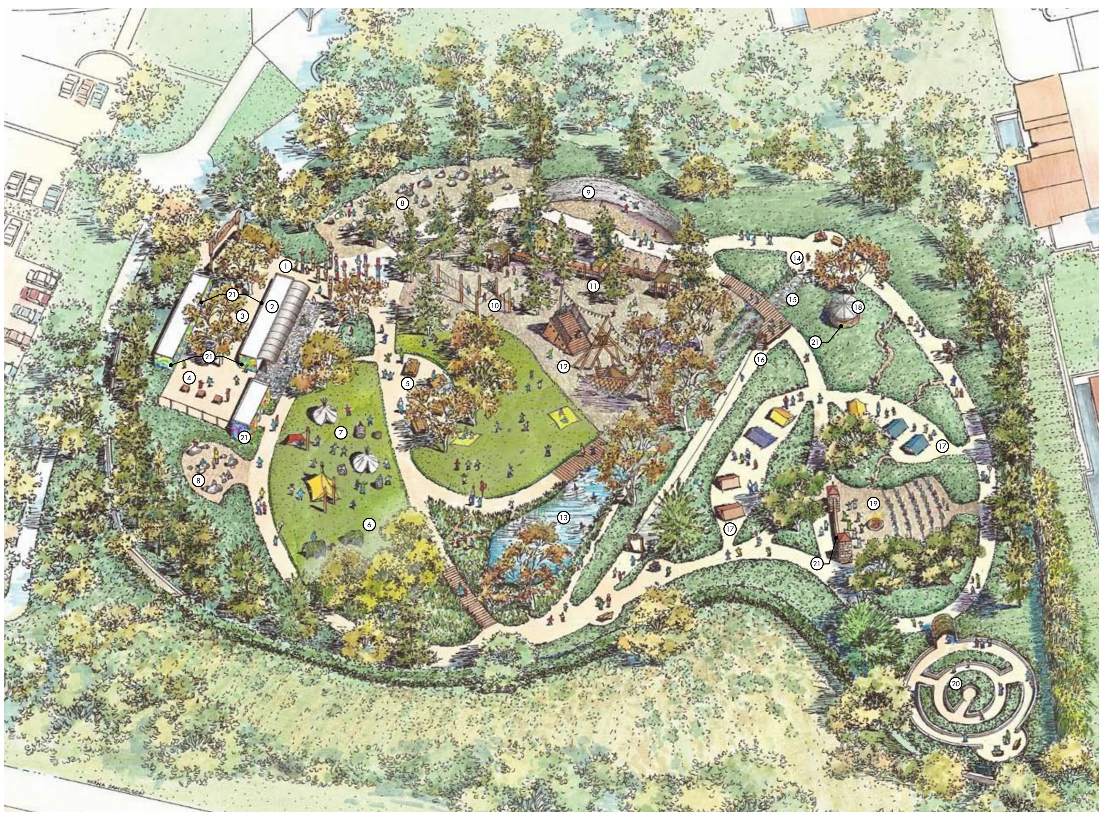


Figure 8.10: Planned redesign of Irvine adventure playground (“Adventure Playground: Park Design Modification University Community Park”).

[19]Advocates rallied and protested. They established a blog, “Defend Adventure Playground,” which can be found on *https://defendadventureplayground.wordpress.com* as of December 2015.

Donald points out that many Bay Area adventure playgrounds shut down because "the neighbors didn't like the noise and the kids and the banging and the parking" and its perceived diminishment of their property values. The Berkeley site, on the other hand, is safely ensconced on the civic marina. A fear of dropping property value is a commonly voiced concern (Sutton-Smith 1980; Donald 2012). "Modern civilization", writes Allen, "interferes with a hard and heavy hand in the spontaneous play of children"—a form of "psychological pollution. This arrogance, this paucity of invention, this regard of the worth and scale of the individual is a world wide disease and is one of the tragedies of affluence" (Bengtsson 1972).

As these examples indicate, sustaining adventure playgrounds requires ongoing meditation between the players and the broader community. Play means something different to players than non-players (Sørensen 1951; Jones 2003), and the differences, especially in public playgrounds, must be reconciled if they are to persist (Dattner 1969 [1974]; Spivack 1969; Vance 1982; Latané 2013; Kinoshita and Woolley 2015). One way to achieve this balance is through buffering. Sometimes the best way to involve the community is to keep it at arm's length.

Buffering

Buffering can take many forms. Beginning with Emdrup's rose planted earthen berm, and extending to Berkeley's fenced in marginal location, boundaries regulate the tension between players and non-players. Emdrup's first play leader, Bertelsen, wrote that the encircling earthen mound "takes some of the sound and view of the children's games … as some grown-ups find it difficult to tolerate this form of amusement" (Bertelsen 1953). Sørensen observed that the presence of such dikes distinguished whether or not a junk playgrounds was accepted by its neighbors (Sørensen 1951).

The Berkeley playground, as we have seen, is liminal. It sits in a marginal location—far from people's homes. Donald attributes the playground's longevity to this fact; there is no NIMBY issue at stake. The land is leased from the state, and the marina's charter provides a degree of bureaucratic separation and protection. The playground is siloed within the special Marina Enterprise Zone, which has its own Marina Fund separate from the city General Fund (Donald 2012; Kamen 2012; Kamen 2015; Donald 2016).

The Berkeley's adventure playground boundary is regulated in other ways. Rules govern who can enter, and how. Donald is understandably skeptical of me and my project when I first make contact, but she eventually comes to accept me as a volunteer/researcher. She remarks that I could help pull nails—whatever the staff might find useful—while doing my observations. In order to gain access to the playground I must pass muster with Donald and negotiate the IRB (Institutional Review Board) process at UCSC.[20]

[20] It takes two attempts to successfully define and delimit my relationship to the playground and its participants via IRB. There are IRB complications if one wants to work with children, boundaries to negotiate.

Figure 8.11: Berkeley adventure playground entrance signage. Photo courtesy of Patty Donald.

The area just inside and around the main gate is surrounded by a wall of notices, warnings, legal waivers, and explanations of group fees. A giant red stop sign greets you as you walk into the gate, arresting and directing your attention to this material. A waiver must be signed to gain entry, and children must be accompanied by adults. This rule is strongly enforced. I witnessed a young unaccompanied child brought to the attention of staff, who then escorted the child out and searched for his guardian.

All of these buffers, whether a fence, mound, thicket, waiver, or social patrol, are not impervious; they regulate access. Emdrup's gatehouse needed to be unlocked by the play leader in order for people to access the playground, its tools, and its lavatory (Bertelsen 1953; Bertelsen 1972). Berkeley operates similarly: the staff control the main entrance, and keep tools—as well as the zip line trolley—under lock and key when the playground is closed. Interlopers might violate these boundaries; this is suggested to me by spray-paint graffiti—a painting material the children aren't, to my knowledge, given access to.

Internal divisions further organize and buffer the Berkeley adventure playground. Besides the main staff area, with its locked storage container, tool exchange desk, material storage, and de-nailing area, there are other staff-only areas Donald refers to as "red zones." These four red zones, off-limits to kids, are places which staff can quickly move dangerous constructions into—taking them out of the mainstream of activity. These internal buffer zones staff also provide a place from which staff can unobtrusively observe behavior and police risk (Gingold 2013).

Outlet

All of this buffering, and its attendant access control, serves to give an outlet to what proponents of adventure play argue should be normal. Rather than ban certain kinds of play, it is incorporated, and given a protected place for expression. Adventure playgrounds, like wildlife reserves, are places in which children can still engage in wild forms of play. Playgrounds are play places protected against urban encroachment.

Bertelsen gives voice to this sense of displacement, of providing an outlet:

> In all big towns it is necessary to forbid children playing in unsuitable localities, but it is no good being content with a negative ban; something else must be produced instead. For this reason a junk playground is an important part of the playgrounds of the future, an essential for every big town (Bertelsen 1953).

Proponents advocate reclaiming something that has been lost to time and urbanization, of recovering outdoor play in an urbanized world. According to Sørensen, the "junk playground will give the town children something of the same thing that country children have" (Sørensen 1951). Vance, who co-founded the American Adventure Play Association, wrote that

> It has become evident to even the most casual observer that much of the "natural" play space enjoyed by youngsters of previous generations has disappeared. In many urban settings the vacant lot has been replaced by housing developments. Parks and playgrounds designed as alternatives to streets are often designed not so much for the play of children as for the aesthetic pleasure of adults. Fear of property damage or liability, even in rural settings, has deprived many of today's youngsters of the play spaces many of us enjoyed in our youth (Vance 1982).

Proponents search for lost play. This quest, often framed in ethical terms, is a rebellion against how childhood play has been circumscribed by playgrounds, urbanization, and an overly safety conscious culture. According to Donald, adventure play

> should be happening everywhere, its just normal, it's what should happen in a back-yard. It should happen with every family, and it always surprises me that everyone thinks it's so special. It's so easy to do on your own, in your own house … the thing that gives me joy about the playground is the creativity, watching the staff grow and develop and become better adults, and what kind of jobs they get and how it's very similar to this. … There's very rarely a bad day in the playground, its always a happy place, the kids are just thrilled to be there. I feel like it fits my personal philosophy of reduce, reuse, recycle, and rot; of caring for the environment and reusing things, and playing with the earth and getting in touch with the earth. That's why I created these environmental education programs because I was tired of watching kids collect crabs in a cup and leave them on the sidewalk when their mom says you can't take them home … I like providing things that challenge kids to be more than they can.

Donald's youth was filled with adventuresome outdoor play activities, like playing in creeks and building campfires. Her background orients her, and enables her to envision what is possible and desirable. She laments the transformation of children, who have, over multiple decades, become less proficient at creative play—a decline she attributes to the passive experiences of television and video games.

Donald estimates that the promotion of outdoor play over the past ten has had an effect on growing attendance, as well as an increasing desire to be closer to nature, which also finds expression in "Alice Waters and the edible schoolyard. Those sorts of things are getting people away from 'broccoli always comes in a plastic bag'. Those sorts of things have gotten people more in touch with the earth, and more in touch with ‘basics are better’.” After all, she says, “this is Berkeley” (Donald 2012).

All of this buffering, in other words, is so that natural, normal, and healthy activity can express itself.

Transformational Threshold

In order to gain access to the Berkeley playground, I first speak with Donald on the phone. I walk away from our initial phone conversation thinking she sees me as volunteer labor, which is fine with me, as it provides a cover story for my observations. The staff happily accept me as a volunteer/researcher (although I will need to reexplain myself). When I arrive the other playground participants easily accept me as a volunteer, completing my conscription. But in order to properly cross the boundary into the playground I must transform into a volunteer (as well as an ethnographic observer). From the staff members I learn how I should perform this role. I learn to shovel the sandpile that breaks the fall at the end of the zip line, pull nails from wood at the de-nailing station behind the front desk, and hand out tools (Gingold 2013).

As we have seen, the boundary of the adventure playground separates, or buffers, access to the playground, keeping non-players and players apart. The boundary also serves as a threshold. The entrance to the playground offers a waiver and explains the rules. Inside, a sign explains how tools like paint, the hammer, and saw are earned by exchanging them for a Mr. Dangerous, nails, and so on. To cross inside is to transform and assume a new role. Guardians become, in theory at least, careful observers of their children’s behavior. The playground’s design anticipates and reconfigures participants (Woolgar 1991; Murray 1997).

When new participants arrive, especially en masse, they are given special introductions by the staff. Two birthday groups are scheduled the first day I visit, and such groups get an introductory talk about tools, rules, and procedures from staff. I watch a staff member give the spiel to a birthday party seated on a picnic table just outside the playground. They are becoming configured as proper park participants. All these boundaries and transitional elements structure participation, and strive to ensure that if you are inside, then you know what's going on, have signed a waiver, and are properly supervised (or supervising) (Gingold 2013).

In addition to child players, their guardians, and staff, there are other roles. There are volunteers, such as myself. Some of these volunteers might be teenagers seeking an opportunity to participate in the playground without guardian supervision. Over long periods of time participants might shift roles. Kids who played in the adventure playground as children sometimes grow up to become staff members. For example, Donald's daughter grew up playing at the playground, and then joined the Berkeley Day Camp staff as a teenager, taking kids into the playground (Poock 2012; Gingold 2013).

Not all visitors neatly fit into the schema of roles provided by the playground. I witnessed a man come to the desk and ask about "Any parental advice for the zip line?" I didn't understand his question, but a staff member did, answering "go for it." Sometimes guardians use the big seesaw without any kids. I saw curious college age students wander through the playground. These are edge cases, people for whom the playground's boundaries and roles don't quite fit—or at least don't fit my perception of them (Gingold 2012; Gingold 2013).

Summary: The Function of Boundaries

Taken together, we see that the boundaries of the adventure playground perform a handful of functions. The playground itself is fragile, and the boundary offers a buffer that reconciles the tension between inside and outside, between players and non-players. The protection of play inside is seen as an outlet, a reserve for a kind of outdoor and creative play that is endangered. Finally, the boundary operates as a threshold, transforming non-players into players, turning children into proper adventure playground players, and adults into guardians that provide oversight.

Risk

Figure 8.12: St. John's Wood Adventure Playground in London (Allen 1968).

> One of the ways children earn the right to check out tools is by picking up "Mr. Dangerous." For example, pieces of wood with nails sticking out of them. When we were there, a child turned in just such a piece to the teenage attendant, who said, "Oh, thank you. We really need to pick up things like this. Especially if you see any nails sticking out of a ceiling in a playhouse."
>
> No kidding.
>
> —Excerpt from a negative Yelp review of the Berkeley adventure playground.

On each visit to the playground, children must earn tools by trading in nails, trash, splinters, or a Mr. Dangerous. Mr. Dangerous, as the disapproving review at top suggests, is a terrifying idea—a board with a nail you might step on, puncturing your shoe, flipping it up, or worse. But danger, here, has been managed. The playground self-regulates, and encourages children to keep their eyes peeled and clean up hazards. And this hazard, furthermore, has been made identifiable and approachable. According to Donald, Mr. Dangerous "got his own little name" from a staff member, which turned him into a "a kind of a character" (Donald 2012). A board with nail in it isn't just *dangerous*—it's a *Mr. Dangerous.* Just as Allen transformed *junk* into *adventure,* risk has been transformed into a character. Danger has become a playmate. Children, in other words, are directly in contact with risk, and are themselves responsible for its management.

Giving children contact with risk and making them responsible for its management is very much in the spirit of the adventure playground.[21] But safety is also of paramount importance. Risk and safety are both desirable, and so must be reconciled. Mr. Dangerous is one strategy for reconciling this tension. But it seems to be a quixotic task, to reconcile risk with safety.

[21] My use of the term *risk* instead of *danger* is intentional. Danger is simply a hazard, but risk is a rationalized and actuarial concept, a function of probability and loss. As in a car insurance policy, risk allows actors (you, your insurer) to rationally transact with uncertainty and loss. Risk, in other words, is a rational engagement with hazard.

Why do this? And how? Let's look more closely at how risk and safety is scaffolded by the playground, and why this is done.

Risk is desirable

Adventure playgrounds are specially sanctioned sites of transgression. Part of the inversion, a natural part of any kind of transgression, is an increase in risk. The playground offers a license to engage in risk, play in mud, and build with junk. Rather than strive to eradicate risk, a futile effort which sees its fulfillment in the sterile playgrounds that dominate the American landscape, risk is celebrated.[22] Advocates frame risk as pleasurable, natural, and developmentally necessary for cultivating responsibility, creativity, and citizenship.

Figure 8.13: Zip line at the Berkeley adventure playground. Photo by David Gilkey/NPR (Westervelt 2014).

[22]Many religious traditions share that uncertainty is part of life (e.g. Ecclesiastes 9:11). According to Buddhist tradition, striving for complete security and safety in the face of life's unpredictabilities is itself a futile and dangerous trap (Chödrön 2001).

Risk is pleasurable and natural. If you ask kids about their favorite thing in the playground, Donald says, "they'll all say the zip line." "The highest amount of risk is usually what they're favorite thing is because they're not allowed to do it anywhere else." The adventure playground is a site of organized rule transgression. "We tried to do a mud pit here until we realized the raccoons were using it for something else, so we had to close." Although it was a short lived experiment, mud play still might happen outside the playground, on the park shoreline.[23] Sometimes they make "mud angels."[24] "Kids love the gooey aspect of it, as long as the parents are forewarned" (Donald 2012; Donald 2016).

Donald laments the loss of the merry go round and the height one could achieve on swings when playgrounds were transformed by anxieties about liability and safety. Her personal history informs her philosophy of what is possible, what kids need, and what they deserve. Donald came

> from the background of a risky kid. I played in creeks, I was a tomboy, I did everything that girls weren't supposed to. … I did backpacking. We use to have a zip line at our cabin in Santa Cruz, and we'd dam up the creek and jump off a log, and zip down into and drop off into the creek. We'd go out into the canoes and throw rocks at each other, and try to sink each other. … We did all the stupid things you possibly could… (Donald 2012)

Advocates frame risk as developmentally important, helping children to cultivate self-reliance, citizenship, and creativity. Adventure playgrounds, writes Allen, are places where children "come to terms with the responsibilities of freedom" (Bengtsson 1972). According to Allen, the developmental benefit of self-reliance is inseparable from the pleasures of risk:

> It is a rewarding experience for children to take and overcome risks, and to learn to use lethal tools with safety. Life demands courage, endurance and strength, but we continue to underestimate the capacity of children for taking risks, enjoying the stimulation

[23]This also takes less labor and precious water.

[24]Like a snow angel: lie down and rotate your arms and legs.

> of danger, and finding things out for themselves. It is often difficult to permit children to take risks, but over-concern prevents them from growing up. This is all too clearly seen in the dull, 'safe' playgrounds that continue to be devised … There is an urgent need to help develop children's independence, self-reliance and ability to live on their own inner resources, and this cannot begin too early in life (Allen 1968, pp. 16-17).

As a ropes course leader, Donald gained experience with both accidents and proper safety precautions. "Those sorts of things," she says, brought her "to what is possible here." The underlying idea of the ropes course, "the reason they started ropes courses was because they would look at survivors, especially survivors out at sea. The young ones would give up hope because they didn't know they could survive," while the "old ones knew they could survive because they" had been in challenging situations. Donald sees "providing challenging situations" as integral; children learn to overcome problems and grow more resilient. "As long as they're given guidelines as to how far they can go with it. But I truly feel that kids deserve that challenge of risk, a safe challenge of risk" (Donald 2012).

A Yelp reviewer summarizes risk's developmental benefit of self-reliance and responsibility: "The lack of safety addressed in almost all of these reviews is one of the points of Adventure Playground … it helps kids develop common sense … Only a kid raised on plastic playgrounds would lack the common sense to check how splintery wood was before she grabbed or climbed on it." This line of reasoning turns the question of risk on its head, arguing that lack of exposure to risk is, in the long term, more risky. Like a vaccine, exposure to risk inoculates children against future hazards.

Risk is made to seem normal. Mr. Dangerous—characterizing a risky object as a playmate—works to normalize danger. Danger becomes natural. As a participant-observer, it is something I must come to terms with. On my first day at the playground, on a morning safety walk, I help a staff member deconstruct a structure identified as dangerous, and am told to leave the nails on the ground. This doesn't come easily. I try to think of the nails as a game designer thinks of coins, as treasure for kids to find, instead of the hazard I feel that they are (Gingold 2013).

Safety is desirable

Not only is risk celebrated—as pleasurable, natural, and developmentally necessary—but so is safety. While this might seem contradictory, the apparent paradox makes more sense when we consider the special relationship between safety and risk that holds true in play. As we saw in chapter 1, while risk is desirable as a component of play, safety is necessary to begin and sustain it. The adventure playground exemplifies this dynamic.

Proponents of adventure playgrounds claim that they are safer than they might appear. The heightened sense of risk is illusory. Vance cites a study by his American Adventure Play Association which concluded that adventure playgrounds, in terms of safety, are "equal to or better than conventional playgrounds, aquatics programs, and particularly youth sports programs, without exception." He attributes this to supervision and safe procedures of use (Vance 1982). Using a related argument, Allen was able to persuade Lloyd's of London to insure the Clydesdale adventure playground (Kozlovsky 2007). One of the staff members tells me that compared to being a lifeguard—a far more intense, focused kind of work—supervising an adventure playground is more relaxed. He tells me that here you don't have to worry about people drowning. While risk is part of the playground's attraction, risk must also be minimized, not just for the safety of participants, but because danger and liability threaten the playground's survival.

Play needs safety in order to begin. Safety is used to scaffold children into play that might include risky activity. Multiple times, I witness guardians cajole children to overcome their fear of the zip line and use it. Fear melts into delight, scaffolded by an encouraging adult who reassures them about its safety. Donald believes that setting boundaries, and making children feel safe is critical. If “there’s nobody setting boundaries,” then not only will people walk off with tools (which they still do), but kids won’t feel emboldened to engage in “creative play.” Kids need to “feel safe,” that “someone is there that’s looking out for them.” Only then will they be “taking risks, going to a place they’re not familiar with,” psychologically, physically, and creatively. “When you give kids guidelines sometimes they blossom a little bit better,” otherwise they’ll think “we can do *anything!?* and they short circuit” (Donald 2012; Gingold 2012).

While a certain amount of risk is experienced as pleasurable, desirable, and developmentally important, risk also threatens people, and threatens to shut down the playground with lawsuits. For these reasons, the playground also seeks to mitigate risks, fine tuning them to desirable levels. My first experience with this was in contacting Donald, who sought to protect the park and the kids from the risk I might pose to them, their privacy, and the operation of the playground. She is also concerned for my own safety, advising me to bring water, a hat (shade is minimal), warm layers, lunch, clothing that can get paint on it, and shoes that can take a nail through them (Gingold 2013).

Children are required to enter the playground with guardians. Oversight from guardians and staff mitigates risk. (Guardians are often on their cell phones, however.) At the opening of each day the staff perform a safety walk, looking for unsafe structures and hazards, risks to be minimized. We identify an unsafe structure to be torn down after the safety walk. In the language of Penny Wilson, a leading British playworker, this is an *audit.* Wilson advises playworkers, when performing a safety audit, to also assess whether a site will benefit from increased challenges and risks (Wilson 2010).

Figure 8.14: The storage container that serves as a tool shed/staff area (photo coursey of Patty Donald), and the front desk where building materials are received (photo from Yelp, accessed April 8, 2016).

Rules and procedures work to mitigate risk. At the front desk a sandwich board sign explains how tools and hazardous materials circulate in the park. In order to earn a tool you must recover five nails, five pieces of trash, five splinters (pinky or bigger), or one Mr. Dangerous. The tools on offer are a saw (and optional vice), a hammer and three nails at a time, and paint and brush. You can have one tool out a time, and trade them for others whenever you like.[25] These rules help maintain a homeostatic balance between chaos and order, risk and safety. In order to build, you must first do some safety maintenance work. Such rules also means that child participants contribute to park maintenance, which helps reduce the need for staff labor. These rules not only mitigate physical hazards on the playground and save labor, but teach children to attend to hazards (Gingold 2013).

These rules change over time, often in response to emergent problems—staff, for example, having to collect all the nails they pull out every day themselves. Now the children help, too. “All the things in the playground evolved over time,” says Donald. At the bottom of the zip

[25]There is also a big magnet attached to a four foot rope and wooden handle, to make nail collection easy. (This tool does not need to be earned.)

line lies a sandpile that must be continually rebuilt with a shovel. Donald would love to find a way for the sandpile maintenance to be done by kids, but hasn't found a way to do this (Gingold 2013). The playground evolves in response to emergent risks, a theme we will take up in greater detail later when discussing its adaptive qualities.

Summary: Risk and Safety

Risk and safety are engaged in an ongoing dance. Play needs risk, but play also needs safety; it is composed of complementary opposites. This contradiction puts adventure playground administrators in an awkward bind. Their playgrounds promise risk as well as safety; and so, they must satisfy the conflicting desires of children, parents, city administrators, and insurance underwriters. A certain amount of risk is desirable, as it is seen as pleasurable, natural, and developmentally beneficial. Risk must also be kept under control, managed and scaffolded by oversight, encouragement, and procedural rules. There is a self-maintaining homeostatic quality to the Berkeley playground's risk. Children and staff both participate in identifying and cleaning up hazards. The adventure playground, as an institution, adapts and evolves in response to emergent challenges.

Autonomy

Figure 8.15: At left, a garden from the Berkeley adventure playground, circa 2000's. Photo courtesy of Patty Donald. At right, original caption reads "The leader — interested, helpful but not interfering" (Allen 1968).

Sørensen's original proposal for the junk playground anticipated a need for attendants. Following the Danish precedent of "play uncles" and "play aunts" (de Coninck-Smith 1999), this came to pass in the form of a *play leader.*[26] Residents of Emdrup insisted on an attendant, and John Bertelsen served as the first junk playground leader. It quickly became apparent that his involvement was central to Emdrup's success (Sørensen 1951). But adventure play, as we have seen, emphasizes the role of autonomous behavior on the part of children. So what does a play leader actually do? Bertelsen describes his role:

> The children themselves create the playground during their games with junk, bricks and boards. This is best done if they have a popular adult leader in whom they have

[26]Such supervision is unusual in contemporary public playgrounds in the United States, but these playground attendants drew upon a Danish tradition of hiring park supervisors called "play aunts" and "play uncles." Ning de Coninck-Smith argues that the shared etymology of "park" and "parking" points to the function of parks as "a place where the parents could park their children." Play aunts and uncles facilitated the childcare function of parks and playgrounds, and the planning and hiring of such play attendants was motivated, at least in part, by welfare policy (de Coninck-Smith 1999).

> confidence. It is his task to obtain the necessary materials to enable the children to realise their projects. The leader must be an elder comrade of the children, one to whom they can safely confide their wishes and plans. He must have his finger on the pulse of the children and the playground all the time, in order that he can direct the growth of the playground, though in such a way that it is always the children's own Work. A junk playground must grow from inside and never be directed from outside (Bertelsen 1953).

This seems contradictory. How to reconcile the desire for autonomous activity with an adult player leader? It is for this reason that the *play leader* label—like the "junk" moniker—never sat well with Allen. In England, *play leader* morphed into *playworker,* a term that intentionally signals the contradictions inherent to the craft of playwork. Wilson sums up the central paradox: "The ideal playworker leaves the children free to play for themselves but intervenes in carefully measured ways to support the play process" (Wilson 2010). (In Berkeley the playworkers are simply called staff.) The opposing forces of autonomy and guidance are reconciled. But how to negotiate this tension? And why bother with it in the first place?

How to Scaffold

Let's look at some of the strategies playworkers and adventure playgrounds use to balance autonomy with guidance.[27]

A playworker might support, encourage, or direct activity. Guardians sometimes play this part in Berkeley. I witness a son and his father work on a wooden walkway. The father helps his son hold a piece of wood in place so the son can hammer. Guardians egg on children to overcome their fears and use the zip line, or subtly chide them when they spill paint (Gingold 2012). Staff in Berkeley might direct activity with their own enthusiasms. Donald set up a pipe

[27] My intent is not to offer an authoritative reference on playwork strategy, but rather to sketch a catalog of the diverse strategies employed. Penny Wilson's *Playwork Primer* (Wilson 2010) is a good starting point for those interested in the craft of playwork.

with holes in it so that kids had water to play in and grow plants with. One summer a staff member shared her enthusiasm with gardening by planting kale and chard. A staff member interested in fishing might take children out of the adventure playground on fishing expeditions (Poock 2012).

Children seek guidance. Children will sometimes ask staff: Where can I paint? Where can I nail? We tell them: anywhere. The autonomy can be a surprise and a challenge.

There are breakdowns, especially with parental scaffolding. Sometimes support is perceived as being overbearing. A boy leaves to get something, and warns his father not to touch anything. Parents are distracted, talk on their cell phones, chat with one another, or maybe use the play equipment themselves. There are negotiations between guardians and children about what to do and when to leave, and sometimes meltdowns ensue (Gingold 2012). I witness fewer breakdowns with staff. Perhaps this is because staff engage less directly with the children. (At the same time, Donald laments that she wishes her staff would engage and coach the children's play more directly.) Donald encourages parental involvement, but believes that the parents themselves might need guidance—maybe a class—on how to build things and play (Donald 2012; Donald 2016).

Figure 8.16: Distracted parent. Photo courtesy of Patty Donald.

Playworkers don't just support and direct; they also withhold their own judgment and involvement. On my first day at the playground, I must learn to suppress my desire to pick up all the nails myself; they are there for the children to find. I am struck by how the playground staff preside over a teeming playground without dominating it. This would, in any case, be impossible. Their posture is relaxed, which is something I have to learn how to do (Gingold 2012). Jack Lambert, an experienced play leader offered that "[d]oing nothing is one of the hardest things of all" (Kozlovsky 2007). Bertelsen wrote that "the initiative must come from the children themselves. … I cannot, and indeed will not, teach the children anything" (in Bengtsson 1972, quoted by Kozlovsky 2007).

Playworkers become invisible. Wilson describes playworkers as enrobed in a "Cloak of invisibility." A playworker might play with clay and mud, demonstrating its potential for play as well as validating “the messiness of it” (Wilson 2010). Donald compares staff, at times, to a fly on the wall, invisibly looking over the playground. Appropriation, a topic we will take up more fully later, is one way to guide invisibly. Donald and her staff manage the incorporation of donations into the playground (Donald 2012). Kozlovsky describes an episode in which Lambert dealt with disruptive children by discovering their interest in scooters, and starting a club in which broken scooters were fixed. They then became invested in and respectful of the playground (Kozlovsky 2007).

Some playwork, on the other hand, is overtly regulatory. Access to tools and the playground itself is controlled by staff in Berkeley and play leaders elsewhere (Bertelsen 1953; Gingold 2012). In Japanese adventure playgrounds established amidst natural and man made disasters, playworkers regulate earthquake play and tsunami play. This disaster play is encouraged as a form of catharsis, but is stopped if a child is overwhelmed with fright. Playworkers in these contexts must also accommodate the sensibilities of disaster stricken adults who are offended by children at play (Kinoshita and Woolley 2015). As we saw earlier, Berkeley staff audit the playground on a daily safety walk in which unsafe structures are identified and dismantled. They bear responsibility for risk management.

The rules that govern regulatory work are negotiable. In Berkeley, for example, clearly stated rules govern how construction tools are earned and exchanged by children. Five splinters, nails, or trash pieces, or one Mr. Dangerous, is exchanged with the front desk for a tool such as paint, a hammer, or a saw. Once a tool is earned it can be exchanged for another at the desk. These rules help regulate the population and flow of tools, as well as hazardous materials. But they aren’t strictly enforced. A guardian might wish to facilitate the play of their child, and become annoyed at the rules. There might not be enough nails or splinters or trash to find. In one case a grandfatherly looking figure, after failing to convey the procedural rules about tools to the young child he was caring for, simply took a tool from the front desk. Another volunteer happily accommodates some rule bending by accepting a tangle of dried foliage as splinters. This tempers my attitude towards enforcing rules, and I start to become more flexible. Guardians, at times, simply seek to obtain a particular experience for children. Some children break the rules about exchanging tools and deny they are doing so. An experienced staff member is far more relaxed about the rules than me, stating that the front desk mostly takes care of itself. He says that people will follow rules or they won't (Gingold 2012).

Balancing the Contradiction

Figure 8.17: Staff and children collaborating. Source: City of Berkeley web site, accessed April 8, 2016.[28]

[28]http://www.ci.berkeley.ca.us/Parks_Rec_Waterfront/Marina/Adventure_Playground_Summer_Building_Activity.aspx

Let's return to the contradiction between guidance and autonomy. What does it mean? Why invest so much energy in providing children with a carefully cultivated experience of risk and autonomy?

There is a political angle to this. Autonomy and risk are seen as constitutive of freedom and the construction of liberal democratic citizenship. The adventure playground, writes Allen, is a place in which "children feel liberated" and "can learn to come to terms with the responsibilities of freedom" (Bengtsson 1972). Kozlovsky argues that adventure playgrounds are representative of a World War II model of citizenship that emphasized "self-regulated" and "self-improving"

> individuals who internalized the obligation to be free. The adventure playground demonstrates how this new form of citizenship operated: it made the interiority of children observable and governable precisely because free play was conceptualized as a subjective realm of freedom requiring the participation of consenting, active subjects (Kozlovsky 2007).

Adventure playgrounds represent a form of citizenship that serves as an alternative to identification with a leader, a model that become bankrupt in the wake of fascism. Adventure playgrounds can be seen as examples of a *democratic surround.* Fred Turner characterizes the democratic surround as affording individuals the agency to explore and make their own meanings within a designed environment. Democratic surrounds were brought into being by intellectuals, artists, and designers that sought a liberal democratic response to fascist propaganda. They are a benign democratic form of propaganda, a gentle form of guidance for democratic subjects, and an inoculation against fascism (Kozlovsky 2007; Turner 2013).

Turner and Kozlovsky are keenly attuned to the contradictory quality of such environments, in which the designer's craft invisibly and subtly shapes individual autonomy. But is this a problem? Is producing liberal democratic subjects a bad thing? Is there a better way? As

Kozlovsky has it, play is made to seem more free, but is actually more controlled, and has become an instrument of policy. But this conflates what play means to its players with what it means to its observers. Interpretation of someone else's play is tricky, fraught with illusion.

As with risk and safety, the tension between autonomy and guidance is inherent to the adventure playground. But perhaps this tension is illusory—or at least desirable. Winnicott and Vygotsky offer an alternative conception. Play and autonomy are predicated upon a feeling of safety (Winnicott's transitional phenomena) and of guidance (Vygotsky's Zone of Proximal Development). Feelings of safety encourage autonomous behavior and growth. Guidance and scaffolding helps individuals to grow and try new behaviors (Winnicott 1971; Vygotsky 1978; Wilson 2014). Guidance, in other words, is not the enemy of autonomy, but a catalyst to its developmental. This is the ideology of adventure play proponents.

Summary

Adventure playgrounds exhibit a tension between guidance and autonomy. This is most clearly seen in the figure of playground staff who are sometimes called play leaders or playworkers. These titles indicate the contradictory nature of the role, which is to cultivate and guide autonomous play. Playwork is carried about by supporting children in activities, which takes the form of helping, directing, and modeling. But this is balanced by withholding judgement and involvement. Playwork, at times, strives to become invisible. At other times playwork is overtly regulatory, controlling access to tools and the playground itself. But these rules are negotiable, leaving room for adults and children to adapt the environment to their particular needs. Finally, we see that the apparent contradiction between guidance and autonomy might not be contradictory; according to the developmental theories of Vygotsky and Winnicott, guidance and safety stimulate growth and autonomy.

Appropriation

Figure 8.18: Original caption reads: "An ostrich, the product of junk and imagination" (Bertelsen 1953).

The adventure playground is a place where junk is transformed, through play, into something else. Junk is invested with imagination, and becomes something new. The "most intriguing" quality of play, writes Garvey, is that it has "certain systematic relations to what is not play" (Garvey 1970/1990). In this last section we look at how appropriation negotiates the relationship between play and not-play.

The playground's appropriation is, at heart, an inversion. Bertelsen describes the junk playground as giving rise to "[t]he noble art of junkology" in which "[a]ll pedagogical and occupational ideas were quickly turned upside down[.]" Lowly junk is elevated into noble "science." A playground opens amidst "war and the German Occupation." Waste, trash, nonsense, disorder, war, and chaos are turned on their head, transmuted into beauty, meaning, and delight. Floating upon a whimsical reverie of junkology, Bertelsen imagines it as a legitimate area of study accompanied by its own " 'Junkological Society', whose members should explore and declinate the future framework of the science." Sørensen, he says, would be president. The transformation of junk has become a legitimate activity, performed by a "junkologist," with an attendant area of study, "junkology." These musings underline the playground's appropriative inversion, which "makes legal children's games on unbuilt plots, building sites, old business premises etc" (Bertelsen 1953). Sørensen's description of the junk playground as both his most beautiful and most ugly creation further points to this inversion. Kozlovsky describes this as an "inversion of social values," and connects the aesthetics of the junk playground to those of Dada—"a playful and collective reassembling of the leftovers of a machine civilization" (Kozlovsky 2007). This is the redemption of junk, debris, the inappropriate, and the marginalized.

The Flux of Appropriation

Junkology points directly to play's appropriative quality. The outside world is ingested, and junk is imaginatively transmuted into something else. Bertelsen paints a vivid picture:

> On our tour round the grounds we keep on seeing how junk is transformed into material for games and employment. At one place boys and girls have built a high hut on poles; round it a whole negro village has arisen with kraals and totem poles, open fires and monotonous jungle song. …
>
> At one place the children have made a high Indian wigwam out of some old spruce rafters and some straw mats. Inside the Wigwam sits the Indian chief, Oklatava of the Hard Heart, and practises war cries with his family and friends. They are soon to go on the war path, and preparations are many. Some are being painted on face and body, whilst others howl about all the dreadful things to come when, shortly, they are to crawl forwards over the grassy Steppes.
>
> We reach a high tower, solidly constructed, the building of which has taken many preparatory experiments. … It has now been realised at the cost of much toil, but it stands as a monument to the creative ability of the children (Bertelsen 1953).

The detritus of urban civilization becomes loose parts, unglued from their normal associations, and recombined into new patterns and meanings.[29] Loose parts can reimagined as parts of new wholes. The adventure playground is a powerful lever, a mechanism for appropriation.

The playground sites themselves are acts of appropriation. The Huntington Beach playground was established in a gravel pit, taking in both the site and its unsanctioned play activity, giving license to both. The Berkeley playground is in Shorebird Park, which is built on clean construction landfill Donald describes as "concrete, bricks, old sidewalks and broken streets." The adjacent Cesar Chavez park sits atop a former municipal waste landfill. The entire marina park complex is built on junk and waste.[30] At the Berkeley and Huntington Beach playgrounds,

[29] We'll dig into the concept of "loose parts" more fully in the conclusion, but the idea comes from Simon Nicholson, who decried the elitist ossification of the world into fixed parts (Nicholson 1971).

[30] The Albany Bulb is another East Bay park built on waste. See Schwartz for a delightful history on the creative evolution of Berkeley's landfills (Schwartz 2012).

the city provided staff, and donations of raw materials were easily rallied (Vance 1982; Donald 2012; Schwartz 2012; Donald et. al 2015).

The playground is predicated upon the circulation of people and materials, both inside of the park, but also between the park and the outside world. The adventure playground concept is built atop historical ruins, junk and scrap—from the debris of war torn Europe and the economic tragedy of Prop 13, to the literal landfill on which the Berkeley site sits. The materials that circulate into the playground are a waste stream from the outside world. Here, the junk of civilization and anxieties about childhood are spun into imaginative new play possibilities.

Bill Vance's 1976 guide states that "your community is a gold mine of raw materials," and advises not to "be too selective in front of the donor. Next time around he will provide you with a really valuable load" (Vance 1976; Vance 1982). Materials flow into these playgrounds; donated scrap materials are their lifeblood. The Berkeley adventure playground web site calls for both material donations and volunteers. The Berkeley playground's web page reads:

> **Donations**
>
> We welcome donations of scrap materials that stimulate creative play: fabric, sturdy old wooden furniture, 8D, and 16D nails, washable tempera paint, large 3/4" or 1" thick plywood pieces, 2 by 4's, or very small blocks of wood. Please no pressed wood or particle board.**Please do not leave any materials that could hurt a young child with bare hands- no metal, sharp, splintery wood or house paint. No old or thin wood please, The dumpster fees for hauling away the materials are very high. Call first, if you are planning on dropping items off so we can let you know if we can use them. 510 981-6720. We need 2x4s and small wood!**[31]

The flux and appropriation of materials suits Donald's "philosophy of reduce, reuse, recycle, and rot" (Donald 2012), and fits into the larger ecological-educational function of Shorebird Park, the playground's site, and the conservation ideology that pervades Berkeley.[32]

[31] http://www.ci.berkeley.ca.us/adventureplayground/, accessed 2012.

[32] The Berkeley city web site touts The Nature Center's new "environmentally sensitive building" which opened

Figure 8.19: Creative reuse in Berkeley. Note old pianos (in whole and part), a punching bag, tire, and slide. Photos courtesy of Patty Donald.

Imagination can repurpose anything. Inside, one finds everything from boards and logs to old boat hulls and pianos. Donald donated her old theatre stage sets, which become part of the playground. Some central structures in the playground are built around toxic creosote poles (which need replacing), originally sunk in exchange for a case of beer with someone from the telephone company, back in the "wild, wild, west days." Colorful octagonal cubes—materials from a different kind of creative playground—were not appropriate for sawing (they are made of fiberglass), and so were repurposed into cubbies. Some large Lincoln Log style objects were donated without the connectors and were adapted onto mock cannons, so "we had pirate ships that happened." And small pieces of wood "are really magic" because everybody "wants to take something home." Donated fire hose became hammocks. Tires (not steel belted) are great. Some of the best activities have involved taking donated bits and pieces of electronic gadgets and using them for crafts: "be inspector gadget … become a spy … create an object and tell us how its going to be used." Children use the donated materials to enact make believe transformations of themselves (Donald 2012).

in 2004. "The programs in the new building teach children about the ecology of their estuary, watershed, and the Bay, while the building's construction serves as a model for the entire community on environmentally conscious building practices. As a significant and accessible project, this building showcases resource efficient and waste minimization practices and heightens awareness about alternative building techniques." (Accessed in 2012).

Donations can also pose problems. “Spools tend to be a challenge,” and had to be “incorporate[d] into objects that don’t move,” since adults got on them and rolled them into kids. Old electric toy cars posed a hazard that needed to be addressed. People often donate dangerous materials to the playground, like lead based paint, indoor pressed wood products, old things full of splinters and nails. “What do you do with that?” asks Patty. “It’s toxic waste.” Getting rid of the stuff, “that’s our biggest cost, dumping the dumpster” (Donald 2012). Materials, just like volunteers, must be filtered.

Figure 8.20: Berkeley’s donation bin and dumpster. Photos courtesy of Patty Donald.

Unlike a normal playground, whose materials are relatively inert and comprised of few loose parts (Nicholson 1971), this is a place of dynamic transformations. The world is remade in grotesque form, deformed and refracted through the materials of junk and adventure play.

The playground itself is continually being broken down and reconstructed, a form of internal appropriation. De Coninck-Smith quotes the Danish psychologist Anne Marie Nørvig: “The destructive urge … could … just as well be called constructive play” (de Coninck-Smith 1999). Bertelsen writes of the repurposing of an old car in Emdrup:

> Like all other material on the playground the car is also an article of consumption, proceeding slowly but surely towards its destruction. In the end it will be resurrected as a number of other things. There is the old story of the boy who dismantled an alarm clock, and was delighted to find that the cog wheels could be used as tops (Bertelsen 1953).

Materials from outside and inside are continuously appropriated, transformed, and given new life. As in the fall of a tree or death of a whale, a buzz of activity springs up to nourish itself. This appropriative–regenerative function is seen most distinctly in the adventure playground's relationship to disaster and ruins. As in trickster stories, creation proceeds upon the ashes of destruction, detritus, and death (Hyde 1998). Disaster, dirtiness, and ruin become the stage for play, creativity, and regeneration.

Figure 8.21: Repurposed car. Before being donated to the playground, where children took it apart until it became dangerous and had to be disposed of, the car had been used as a flower bed on Cedar Street in Berkeley. Photo courtesy of Patty Donald.

Playing with Ruins

Disaster is where destruction, junk, and creation intermingle. In ruins the world becomes loose parts for imaginative play. Because of this, adventure playgrounds have a special relationship to disaster. Through appropriation, adventure play turns the junk and detritus of disaster into loose parts for regenerative and healing play.[33] Junk helps us to imagine new futures.

The Berkeley playground, for instance, is built on the ruins of California's proposition 13.[34] Constricted public resources interlinked naturally with the agenda of play reformers to produce an adventure playground. A marginalized location protects the Berkeley site from the fate of other adventure playgrounds. Economic downtown stimulates increasing use of the Berkeley playground; Donald believes that the state of the economy is leading to more staycations, and thus more attendance (Donald 2012). Donated junk constitute most of the raw materials for play. The playground is a vivid illustration of thriving on rot, decay, constraints, and limited funds.

In post-war England, bombed out buildings became the sites of adventure playgrounds. Allen argued in "Why Not Use Our Bomb Sites Like This?" that Emdrup, with its use of "waste materials" for play, showed the way forward (Allen 1946). England's wartime debris became the raw materials of adventure playgrounds, constituting a "narrative of reconstruction" (Kozlovsky 2007). The adventure playground accommodates and appropriates disaster, reconstructing civilization out of its own detritus.

[33]In describing the adventure playground as play (and regeneration) within ruins I borrow framing from Anna Tsing's *Living in Ruins* project. I briefly encountered her project while checking out her Fall 2011 seminar. She explores this theme in *The Mushroom at the End of the World: On the Possibility of Life in Capitalist Ruins* (Tsing 2015).

[34]In 1982, an author of a playground manual wonders if the "new supply-side economics will be a boost" for the adventure playground movement (Hogan 1982).

Figure 8.22: Japanese adventure playground built following the Tohoku triple disaster (Kinoshita and Woolley 2015).

In “Children’s Play Environment after a Disaster” Kinoshita and Woolley describe the use of play interventions, and adventure playgrounds in particular, as a means to social healing in the aftermath of disaster. Adventure playgrounds were created in the wake of the 1995 Kobe earthquake, as well as the Tohoku 2011 triple disaster—a combined earthquake, tsunami, and nuclear meltdown (Fukushima). Children engaged in disaster “tsunami play” and “earthquake play,” which was stopped if children became frightened. Through play, children experienced that it is easier to destroy than build. But there were problems, because “there was such a contrast between the children’s play and the sad, mourning atmosphere of people who had lost families, friends, houses and all their resources.” Despite this, play was ultimately accepted as having cathartic value for the children and the broader community (Kinoshita and Woolley

2015). A Japanese playworker nicknamed Buncha offers a vivid description:

> Tsunami play was seen often at Asobi-ba. With a handmade equipment slide, a child gliding from the top plays the part of the tsunami. A child standing below is drenched, and may die or not, and at the side another child is positioned in the role of announcing "a major tsunami warning is announced". They all survived on this occasion. Hideaki Amano (who sometimes came from Tokyo to support us as the headquarters of the adventure playgrounds network JAPA), said that the big slide had been made with tsunami play in mind. In the summer time, children played by flushing water, brought from a keg, from the top of the slide, as tsunami play. There was another child who was frightened feeling the reality of the Tsunami by watching this play. Then I told the children to stop this tsunami play. We (the playworkers at Asobi-ba) have never made the kids play tsunami play. However, another adventure playground organization said that they had tried to involve children in tsunami play. They had tried to use a blue sheet and balls to imitate a tsunami. At Asobi-ba, we let children decide by themselves whether to play tsunami play or not. There is a discussion whether it is good or not for play-workers to introduce this kind of tsunami play for children so that they can release their stress and trauma (Kinoshita and Woolley 2015).

Urbanization can be seen as another disaster that adventure playgrounds respond to. Adventure playgrounds are advanced as compensation for play possibilities lost to encroaching urbanization (e.g. Sørensen 1951; Bertelsen 1953; Allen 1968; Vance 1982; Donald 2012). "The fact has to be faced," writes Allen, "that modern civilization interferes with a hard and heavy hand in the spontaneous play of children. The use made of land around buildings is still, almost always, totally unsuitable for children" (Allen 1968). The pressures of modern life also deprive children of the time and license to play (Kinoshita and Woolley 2015). Adventure play, proponents argue, should be normal, remedying some of the ills inflicted on children by modern society.

Why are adventure playgrounds well suited to disasters? First, their appropriative powers—inversion, junkology, call it what you like—is voracious enough to consume and repurpose the flotsam and jetsam of civilization. In fact, these playgrounds thrive on the dissolution of the fixed world, which is converted back into loose parts, ideal materials for play. Schools bombed in war become heaps of bricks and other loose parts. Not much else, besides a degree of safety, is needed beyond the space to play: junk play spontaneously appears and can even thrive, like a desert plant, in zones of depletion. Second, adventure playgrounds are cheap. They can spring up in occupied Denmark, post-war England, and in post-Proposition 13 California. Vance writes that adventure playgrounds offer "the least expensive, most exciting, and by far the most heavily frequented playground program in town" (Vance 1982). Finally, adventure playgrounds are often seen to harbor healing and restorative powers. They hold out the possibility of pleasure in times of impoverishment, and an opportunity for coming to terms with risk and destruction. In them, the world is creatively remade.

Robustness

Adventure playgrounds are continually changing. Not only do individual playgrounds morph and change in the hands of children, but so does the playground concept itself. Operating principles vary across place and time. Adventure playgrounds, Allen writes, are "significantly different from one another, for waste material playgrounds are influenced by the country, the nature of the site, the wishes of the children, the imagination of the leader, and the amount of money available" (Allen 1968, p. 54). New things are tried and thrown out if they don't work. The travels and metamorphosis of the junk playground concept reveals an underlying adaptability and robustness. It has thrived in Europe, the United States, and Japan. It continues to change. It becomes not a junk playground, but an adventure playground. It is localized and adapted to local circumstances. Robustness underpins the concept's resiliency and appropriative essence.

"All the things in the playground evolved over time," says Donald. She describes the changes she introduced as a matter of survival. For example, having the kids "help with the safety elements" by collecting nails, for example, arose as a response to the challenge of having staff carefully collect every nail while pulling 50–70 a day.

In order to survive, the playground must robustly and imaginatively respond to changing risks, needs, and donations. Absorbing donations from the outside forces change within. Many of the playground's materials come from donations, some of which turn into safety challenges to be assimilated. The spirit of supervised risk, and the Berkeley playground's spirit of adaptability, is related in a story about donated toy cars that kids could roll around in. Parents put "three year olds in there … it was just a nightmare." Supervision couldn't be left to the parents. The question for Donald was, how "do we change the flow of the playground to accommodate this use?" The cars were stored in a shed, but "people [would] just reach into the staff area and help themselves to stuff, which is a no-no." Someone helped themselves to one such cart, and a kid pinned a woman against a boat, resulting in a lawsuit. But, "because it's so popular we're going to create a day, or a couple days, and create cart day, … and assign people to be traffic cops. … We have to figure out a way to accommodate this joy of 'I'm barreling down the hill and out of control!' to something we can make not hurt people." Robustness enables Donald's playground to creatively adapt to unanticipated materials and problematic behaviors. The joy of the carts isn't denied, but appropriated.

Risk and litigation is kept at bay by transforming the playground's operation. "Lawsuits that happened before usually happened during chaos hour, between 4–5" when kids know it is time to leave. Tired and overheated kids, for example, jumped onto the side of a fort, and a plywood board that wasn't properly nailed in fell on them. This was an "accident, not anything anyone could have foreseen." "Now there is no loose wood—staff is constantly checking to make sure." (We saw this in the safety walk.) As a result, staff perform more safety audits and attend more intently to kids during closing hour.

"Another one where a woman was reading a book and walking up the zip line path, reading a book. A kid gets on the zip line everyone's yelling: *lady get off the path!*" A mishap resulted in a more of a boundary—made of objects, fences, and weeds—being built up around the zip line area. With the zip line often it's parents who "break the rules, and let their little kids go on it" before they are six years old, which has resulted in broken legs, and parents slinking way without notifying them "because the parents didn't want us to say we told you so." If the opportunity presented itself, she would restructure the zip line so the low point is in the middle, eliminating the need for both the sandpile, its maintenance, and safe landings in it (Donald 2012).

From new materials and behaviors arose problems, which prompted successive reimagining of the playground's operation. The openness of the playground, its ability to take anything in, means that it must be commensurately flexible, and robustly adapt to challenges. Donald's flexibility, imagination, and commitment to the core values of adventure play have helped sustain the Berkeley playground. We see not just how robust the playground is, but how robustness breathes life into it.

Summary

Appropriation is central to the adventure playground. It scaffolds and legitimizes the imaginative transformation of junk, its conversion from not-play to play. The world—especially its junk, refuse, and waste—become the raw materials for play. This orientation makes the playground particularly well suited to ruins. But in order to sustain a voracious appropriative appetite a commensurate degree of robustness is needed; adventure playgrounds must respond to changing circumstances and materials.

Conclusion

Playgrounds are contradictory and liminal places that Sutton-Smith compares to zoos that have failed to keep their charges (Sutton-Smith 1980). In this chapter we looked at how adventure playgrounds tangle with the contradictions inherent to the playground itself, manifesting these tensions in caricatured form. In contrast to an aestheticized landscape designed by architects and artists, adventure playgrounds scaffold a different kind of play, offering children the freedom to dig in the earth, build, imaginatively reshape the world, and "make mankind's basic discoveries" (Bertelsen 1953). Such play is seen as compensatory for what has been lost to urbanization. (The popularity of the computer game *Minecraft* suggests that the desire for such play is now compensated in new ways.) These playgrounds, however, are quite fragile, and employ a variety of design strategies to absorb the tensions that continually threaten to destroy them. Boundaries mediate between inside and outside, keeping players and non-players at a safe distance. Danger is normalized and adapted to, reconciling contradictory desires for risk and safety. The craft of playworkers seeks to reconcile the tension between free versus guided play. Imaginative appropriation and robustness, finally, mediate between play and not-play.

Adventure playgrounds show how play transcends constraints and materials, overcoming ruin, bureaucracy, and danger. Play is not in the materials we have designed, but in our desires, imaginations, and performances. Play transforms a nail from a danger to a delight. Everything can be swept up into play—from a tiny nail to the wreckage of civilization, a landfill, the stringencies of urban and bureaucratic life, economic wreckage, the ruins of war, tsunamis, nuclear meltdowns, and earthquakes. The adventure playground can be thought of as a rambling and voracious appropriation machine that thrives at the margins of civilization, transforming not-play into play. It is a powerful lever for the intention to play, scaffolding it, and enabling it to transcend ruin.

Chapter 9

Play Design

To see a World in a Grain of Sand

And a Heaven in a Wild Flower

Hold Infinity in the palm of your hand

And Eternity in an hour

—William Blake (first four lines of "Auguries of Innocence")

Introduction

In chapter 1 we saw that play is an *attitude,* a special disposition towards resources, whether physical, social, or symbolic. It is pursued for *intrinsic* purposes, and requires a feeling of *safety* to proceed. Actions are *separate,* buffered from their usual meanings and consequences. Despite this detachment, play refigures what is not play. It is *transformational,* and it is social, enmeshing one or more players in a *shared* play frame. Play is *structured variability,* marrying stricture, free movement, and agency.

In the case studies, chapters 3–8, we saw how play is solicited and supported by material things (a computer, plastic Lego bricks, software package, or model classroom city), social practices (the evolving adventure playground tradition), and behaviors (the stylized play bow of a dog, an older playmate's make believe guidance, the fluctuating and unstable demands of a stylized city simulation). Collectively, I refer to these supporting entities as *scaffolds* (chapter 2).[1] I employ the term to designate both the supporting entity (scaffold as a noun), and the act of support (scaffold as a verb). Computer software (playthings), adventure playgrounds, and playmates all scaffold play.

Taking our case studies as examples of play scaffolds, certain design commonalities become apparent. Gathering these up reveals design patterns which invite, support, and sustain the characteristics of play. This chapter takes up and enumerates these design patterns. These patterns are understood to be both generative and descriptive, useful for design as well as analysis.

To obtain these design principles and techniques, I looked for patterns across the case studies. I was able to identify many shared design techniques which I further explored by gathering supporting evidence from both the characteristics of play and additional examples. I then reflected over this collection of techniques, condensing them into broader categories, and organizing these under three banner design principles: encourage appropriation, afford transformation, and entrain. Throughout this process, I attended to how the design techniques engender the characteristics of play distilled in chapter 1.

Encourage Appropriation

In play, we approach the world with a special orientation, and look to it as a set of diverse resources—language, people, objects, activity—to play with. To encourage appropriation is

[1] I follow a somewhat idiosyncratic usage of scaffold, after (Clark 1998). For more background on this see chapter 2.

to catalyze this process. It is to help players reframe and see themselves, others, and the world around them as play, to offer stylizations that fit the world into their grasp, and offer pivots that stimulate and receive the player's meanings and purposes.

Reframe

Appropriation is scaffolded by reframing what is not play as play. Just as art galleries contextualize what they contain as art, Burning Man provides a context in which scenery, objects, art, and activity appear as festival play. Haul it to the desert, and it crosses a magical threshold, and joins a carnivalesque city. This threshold beckons us to bring things across it, inviting us to see what is not Burning Man as potential carnival play, and happily accepts donations and installations as part of Black Rock City. Reframing can also be seen at work in alternate reality games, such as the Jejune Institute. Not only did the urban landscape of the San Francisco Bay Area become appropriated as the sprawling play field of an epic narrative adventure game, but players, in turn, came to see everything in their environment as enchanted (McCall 2012).[2] Japanese gardens, by carefully framing views of scenery outside the garden, such as the distant mountains, the sky, or forests, incorporate scenery from outside the garden's perimeter, making it appear to be a contiguous part of the garden.[3]

Reframing establishes a perspective from which the outside appears to be part of the

[2]Because, in part, of how boundaries were blurred, this aesthetic effect could persist beyond a play session. Everything became a potential clue, shred of narrative, or part of an enchanting world. Ordinary life, to many players, took on an enchanted quality. The Jejune Institute caused players to continue transforming their environment, appropriating it into their private play.

[3]This technique is called *borrowed scenery* or *captured alive.* The garden is made to feel more expansive, layered, and contextualized (Higuchi 1983, p. 80; Slawson 1991, p. 83; Ketchell 2013; Deane 2015). Deane writes that "[t]he idea of borrowing a backdrop from surrounding scenery is known as shakkei"—"borrowed scenery." This "replaced the original Japanese term of ikedori"—"captured alive." Deane continues, "[a]s Günter Nitschke observes, the term ikedori "makes it clear that shakkei is more than just a view of a section of the distant landscape. It is the art of 'capturing alive' both natural features, such as mountains, hills and plains, and man-made structures, such as temples and pagodas" " (Deane 2015).

inside, which beckons players into appropriating, and provides leverage for them to do so. The Berkeley Adventure Playground reframes junk as a material for play. It accomplishes this by filtering junk deposited at the playground, hauling away the unwanted parts (just as a Japanese garden wall might hide the unwanted street scene, opting for the mountains beyond), and taking what remains and inserting it into circulation. Players, then, encounter materials for building and play. The overall aesthetic of the playground, unpolished, coarse, and heterogeneous, invites and receives highly worn and imaginatively mismatched materials. The roughness gives license to imaginative appropriation. Marvelous things appear: a painted piano, grotesquely out of tune and damaged, but playable. An old boat. Hammocks made of fire hose. The grotesqueness which naturally results from creative repurposing and appropriation signifies play. The adventure playground, like Burning Man, establishes a context, a social practice, that reframes the world as a resource for play. Standing atop these scaffolds, we see the world differently. The city becomes enchanted. The leftover junk of civilization becomes a transcendent material for construction, play, and celebration. The world radiates new potentialities.

Reframing can also support the make believe play frame, drawing players into the active creation of belief, and structuring their participation. This is accomplished by reframing activity as happening within a diegetic world. As we saw in chapter 2, make believe participation is scaffolded through commentary that provides roles, establishes scenes, narrates, and directs. A parent, for instance, might frame activity as make believe through commentary: "Is the dolly tired?" (Garvey 1970/1990). Participants are scaffolded into proficient make believe.

City Building Education and *SimCity* both frame make believe in this way, appropriating the player into their imagined worlds. Players are given roles, framing them as a mayor, representative, or citizen. Both Purium's facilitator and the *SimCity* simulation narrate and frame activity as happening within a make believe city: this is the Purium city, the president is arriving, a woman is pregnant and needs a hospital. *SimCity* frames play as happening within a make believe city by offering the player the role of the mayor, describing the city via mimetic artwork, commenting upon public sentiment in such a way as to constitute it, narrating events such as an earthquake, staging scenarios, and directing the player to build infrastructure such as a seaport in response to the demands of an unseen citizenry. In City Building Education, participants create and wear costumes, and invent songs and dances that they perform. These threshold objects and actions reframe players as part of a make believe world.

The genre of play called games utilizes such framing practices to construct participants as players. Through reframing, players come to see themselves, each other, their actions, and playthings—cardboard, tokens, balls, and fields—as settling virgin landscapes or competing on sports teams. Games provide players with roles (e.g. black or white, motivations in the form of win conditions), scenes (a board, a set of pieces, a backstory), plans (game phases, turns, objectives, a script), make believe transformations (this bit of wood is a rook, this one a queen), forceful directions (you must capture in this circumstance), questions (turns), choices (a repertoire of valid moves), and commentary and narrative (valorized outcomes, captures and wins). Games frame people, things, and activity as part of an imaginary situation.[4]

Reframing also operates at a meta-level, giving rise to lasting play competencies, and encourages people to see themselves as players. Just as older playmates legitimize and guide make believe play, boosting the fluency and willingness of players in the short and long term (Garvey 1970/1990; chapter 2), Forrester's publications guide make believe and validate a simulation practice. Functioning like an older sibling or parent, these texts ushered Wright's entry

[4]As Vygotsky has pointed out, rules underly imaginary situations, just as imaginary situations underly rule based games. Fantasy enactment requires the following of rules, just as a game like chess projects an imaginary scenario full of pieces, moves, capture, and winning. Chapter 1 contains more detail on this point.

into city simulation and the make believe of system dynamics. By providing a book, *Urban Dynamics,* filled with enthusiastic commentary, narration, objects (housing stocks, migration rate), and roles (that of the simulation builder), Forrester brought Wright into the pretense of his city simulation, validated a particular practice of computerized simulation making, and framed Wright's participation as a simulation maker. Wright learned how to pivot the levels and rates of system dynamics towards his own simulation purposes, first with *SimCity*, and then with more liberal adaptions like *SimEarth* and *The Sims*. Software programs and their attendant discursive and social practices—books, articles, and contests in the case of cellular automata (chapter 4)—have worked to perpetuate and scaffold the make believe practices in which some objects (cellular automata and system dynamics) come to stand for others. Such framings construct and legitimize the role of simulation maker and player, leading others, like Wright, to play the part and perpetuate the practice.

Stylize

Stylization facilitates appropriation. Stylized things are easy to apprehend and assimilate. Miniaturized and simplified things are easily grasped by "interior thoughts," "imagination," and "interior feelings"—and thus more easily appropriated for play, which is itself a schematic form of representation and enactment (Sutton-Smith 1986, pp. 138 and 249).

Toys are typically miniatures, rendering them susceptible to playful appropriation. Miniaturization diminishes the world to a scale that feels safe, is easily understood, grasped, and placed under the reign of a player's thoughts, imagination, feelings, and manipulations. "Worldy phenomena," writes Sutton-Smith, are reduced "to our own scale and our own terms" (Sutton-Smith 1986). "Since [play's] voluntary controls are hard to achieve, play takes place most easily in a setting established for that purpose and with scaled down objects," such as dolls (Sutton-Smith 1972) and video games (Sutton-Smith 1986). City Building Education places not just a miniature urban landscape into the classroom, but a miniature society, a scale model of civic governance. *SimCity* is a miniature world on a computer screen. Cellular automata function as scale models of everything from ant colonies to spiral armed galaxies. All of these scale models make entire worlds subject to our imaginations, and gently fit them to the palms of our hands.

Figure 9.1: Stylization in Minecraft *(left) and* LittleBigPlanet *(right). Coarseness encourages diverse representation and use. Source: mojang.com and mediamolecule.com*

Stylized and coarse representation encourage creative reinterpretation. As a sculptural material, the coarse cubes of *Minecraft* are fine enough to allow everything from dragons and castles to spaceships to be made, but they aren't so detailed as to be an onerous and persnickety material for representation. The overall abstracted and low fidelity quality of *Minecraft's* visual world accommodates a wide variety of aesthetic styles and creations. Like the whimsical handicraft universe of *LittleBigPlanet*, components and styles are specific enough, in terms of visual appearance and behavior, to fire up and egg on players' imaginations, yet are abstracted enough to leave breathing room for creative repurposing.

To schematize something is to simplify, abstract, compress, caricature, or exaggerate it. What is ordinarily complex and finely shaded becomes simple, sharply graduated, and crisply delineated. By schematizing a phenomena, it becomes easier to perceive, parse, grasp, assimilate, and appropriate. Games, toys, and cellular automata are all characterized by a high degree of schematization. Cellular automata—"synthetic universes defined by simple rules much like those of a board game"—are schematic and easily assimilated worlds (Toffoli and Margolus 1987/1988b). The cellular automaton formalism produces worlds constituted by clearly enumerated states (alive or dead), sharply demarcated spaces (square cells), and actions (transition rules). Such schematic stylizations make cellular automata extremely legible. In the *Game of Life,* we see formations live, die, and locomote.[5] Clearly delineated parts and activities lend themselves to reframing, assimilation to interior thoughts, and appropriation.

SimCity is schematic in part and whole. A discrete cellular structure gives the landscape an abstracted and clearly defined form. But all of these stylized details, when combined, don't become a jumble, but manifest as an abstracted whole. Top down simulation dynamics and representations—the demand valves and data maps—give the city a schematic and easy to apprehend quality in overview.

Schematization affords social appropriation and interchange, by facilitating the construction of a shared play frame. This is best exemplified by games. Games carve, out of the formless chaos of the world, carefully delineated space, time, turns, moves, conditions, teams, goals, winners, losers, and time. This is obvious in the case of a board game like Monopoly, with its carefully demarcated space, tokens (the player markers, money, and property cards), players, goals, rules, and moves. The game materials function as highly stylized represen-

---

[5]Although system dynamics is also a highly schematic scaffold—think of the diagrams of levels and rates that serve as practically complete functional descriptions of models—the output, unlike that of a cellular automaton, is in the form of continuous graphs. These graphs, although they are schematic in that they abstract complex industrial and social phenomena into numbers, are hard to parse for action and causality. It is easier, in other words, to assimilate the design of a system dynamics model, which is highly schematized, than its output, which is not. Cellular automata, on the other hand, are highly schematic in behavior. Their specifications, however, are not as schematized as in system dynamics.

tations that synchronize the attention, action, and negotiations of the players. Players can point to and unambiguously speak of specific places, property ownership, whether a move is allowed or forbidden, and who has won. Sports, although less discrete, employ schematized elements such as teams, balls, moves, stripes on a field, turns, and conditions. Participants, including spectators and gamblers—who appropriate the play of athletes for their own purposes—can speak of who is on first, the offense and the defense, teams, roles, fouls, downs, quarters, points, winners, and losers.

Stylization results from the ways in which playthings and play practices refract and filter the world. Just as water refracts light, producing a distorted image of what is above or below the surface, play materials and practices superimpose schemas that filter the world, transforming it into grotesque, inverted, distorted, and stylized forms. Lego bricks scaffold the production of jagged edged, primary colored, and reconfigurable forms; cellular automata afford the creation of evocative and lively two dimensional systems; system dynamics scaffolds the production of unstable, highly abstracted models of systems as levels and rates; you can build any city you like in *SimCity*, but it will be caricatured in a cartoonish rectilinear style; the repurposed craft and junk materials of City Building Education and Berkeley Adventure Playground lend them a coarse, improvisational, and weathered quality; the slightly out of control tools of *KidPix* leads to cheerfully exaggerated and out of control paintings. You can make the world out of any of these representational schemas, and it will be refracted and stylized accordingly.

Stylization also helps to send the metacommunicative signal that this is play. Smallness signifies play. So do the schematic simplifications of toy cars and dolls (Sutton-Smith 1986). The caricatured form of children's toys, cellular automata, *SimCity* cities, Minecraft worlds, and professional sports, all send the signal that *this is play*, encouraging their appropriation as a resource for play.

Pivots

Pivots are key to scaffolding appropriation. They are enactive props—objects that can be flexibly used to stand for other objects (Vygotsky 1978; chapter 1). Pivots let players insert their own meanings into a scene. They might be processes, objects, places, or people. The stick becomes a horse, a pattern of cellular automaton states becomes a glider, and a banana becomes a telephone; the living room becomes a castle, and the family dog becomes a dragon. Pivots are essential tools in make believe play, in the "voluntary transformation of the Here and Now, the You and Me, and the This or That, along with any potential for action that these components of a situation may have" (Garvey 1970/1990).[6]

What makes a good pivot? At least in theory, any object might be adopted, through play, to stand for something else. But what makes a pivot amenable to imaginative appropriation?

One consideration is the pivot's place on the continuum between specific and general. A good pivot evokes specific uses and meanings on the one hand, while leaving room for the player's imagination. The cells of a cellular automaton suggests space, a cardboard box suggests a building, and a stick a horse. The representations are specific enough to enact these meanings. But a pivot must be general enough to absorb and carry new meanings assigned to it. Wright used the generic and evocative qualities of cellular automata to enact a city, but *SimCity's* specificity of representation makes it hard to imagine it as something else.

[6]There is a relation here to the idea of an "operational logic," but the pivot idea is broader (Mateas and Wardrip-Fruin 2009). Operational logic is defined to exclude the interior structures of a computing system, such as file descriptors, and focuses on ones that the user encounters. The idea of the pivot is broad enough to encompass both the internal (hidden) and exterior (visible) objects of computation and their enactive properties. One can conceive of internal computational constructs, such as sprites, valves, cellular states, and file descriptors as pivots; they are specific enough to constrain and afford particular uses, but abstract enough to carry a wide range of meanings assigned to them through use. A sprite can become a ball, Mario, or an explosion; a valve can represent the birth rate or the rate at which widgets are manufactured, a cellular state becomes alive, and a file descriptor becomes a repository of geographic coordinates.

Too much specificity shuts down appropriation. Nelson, for example, was disappointed that when she gave kids props that looked like roads that they used them only as literal roads. To circumvent this triggering problem, Nelson gave her students bits of string and asked them to build *paths* instead of *roads.* The string and the word *path* were abstract enough to become an imaginative transportation network, yet specific enough to cue a pathway. For this reason, Nelson prefers *path* to *road* and *shelter* to *home* (chapter 7). Denaturalization strips away existing meanings and associations, clearing the way for imaginative reuse.

Pivots have evocative potency. Will Wright and Doreen Nelson both found that the theme of the city is resplendent with potential meanings. As a pivot, the city triggers and accommodates many potential meanings. It is a rich domain for conceptual transfer, handily mapping and manifesting ideas such as circulation, part and whole, and growth and decay.

Pivots have enactive potency. Their enactive behaviors can also be specific, triggering certain associations, and functioning as receptacles for new make believe meanings. A box will stand in place, like a building. Levels and rates, the basic elements of system dynamics, enact the holding and moving of materials, and enact feedback fluxes when combined. Levels and rates, because of how they behave, are potent representational materials for inventory, manufacture, sales, population, pollution, employment, taxes, births, and deaths.

Constellations of pivots create networks of meaning. It is only when levels and rates are combined that their enactment takes flight, which illustrates the systemic quality of pivots—that they, like Lego bricks, draw enactive power in combination. A single Go stone takes on very different meanings when combined with other black and white stones, a latticed board, some rules, and a couple players, giving rise to complex phenomena that assume, in the minds and discourse of players, dramas of life and death, territorial contests, bold invasions, and desperate defenses. Like the feedback loops of system dynamics, Go as a system is enactive, evocative, and flexible; its dynamic formations and activity serve as pivots which evoke and receive the metaphors of battle, life, and death. Computing and simulation practices are resplendent with pivot networks—cellular automaton cells, rules, and states; system dynamics's levels and rates; file systems; graphical user interface icons, windows, and pointers—that function as an ensemble cast, ready to enact the scenes imagined by players.

Afford Transformation

Play transforms. After reframing, stylization, and pivots have turned the world into a resource for play, transformation can take place. In play, players push the world into new meanings and configurations, construct and reshape, give stylized performances, turn upside down, and make believe. Such transformations are enabled and encouraged by manipulability, robustness, and the celebration of change.

Manipulability

Transformation is made possible by manipulability. Plastic, pliant, and supple materials yield to the probing and shaping of our hands, bodies, and minds. Our manipulations can be sure and skillful or tentative and naive; directed by focused intent or absent mindedness.

Loose parts are variables that afford transformative manipulation. Drawing inspiration, in part, from adventure playgrounds (Taylor 2008), Simon Nicholson's "The Theory of Loose Parts," argues that "in any environment, both the degree of inventiveness and creativity, and the possibility of discovery, are directly proportional to the number and kind of variables in it." This variability is more than simple combinatorics, encompassing a wide variety of transformations:

> all children love to interact with variables, such as materials and shapes; smells and other physical phenomena, such as electricity, magnetism and gravity; media such as gases and fluids; sounds, music, motion; chemical interactions, cooking and fire; and other humans, and animals, plants, words, concepts and ideas. With all these things all children love to play, experiment discover and invent and have fun (Nicholson 1971).[7]

Loose parts afford freedom of motion, the pleasure of agency, and the experience of risk. Emergent meanings can take shape, both nonsense and sense. We see loose parts in *SimCity's* cities: the character library, moving agents, and valves; the nails, wood, paint, and people of the Berkeley Adventure Playground; the players and craft materials of City Building Education; the letter tiles of Scrabble and magnetic poetry; the recombinant elements of *The Sims*, *Katamari Damacy*, and *Mr. Potato Head*.

[7] I've lifted Nicholson's phrase "loose parts" but left alone a lot the other material in the essay, which is, overall, more evocative and polemical than practical. In addition to taking up variability, multimodality, discovery, and invention, Nicholson's essay has a strong political and activist bent. (It cites the *Whole Earth Catalog* and rails against the closing of Berkeley's People's Park.) Loose parts are potentially everywhere: "the most interesting and vital loose parts are those that we have around us every day in the wilderness, the countryside, the city, and the ghetto." We live, however, in a world with insufficient loose parts. Nicholson's polemic is intrinsically political, as "loose parts … are at present controlled and fixed by an inflexible education system and cultural elite." This elite thrives on the myth that creativity is a specialized skill. He advocates de-schooling and other interventions so that "a new generation will be able to invent new systems with the parts." My aim in focusing on variability, the real heart of his essay, is not to pry the political themes from transformation, but rather to bring focus to one design strategy. Nelson's City Building methodology underlines the political potential of transformation quite clearly.

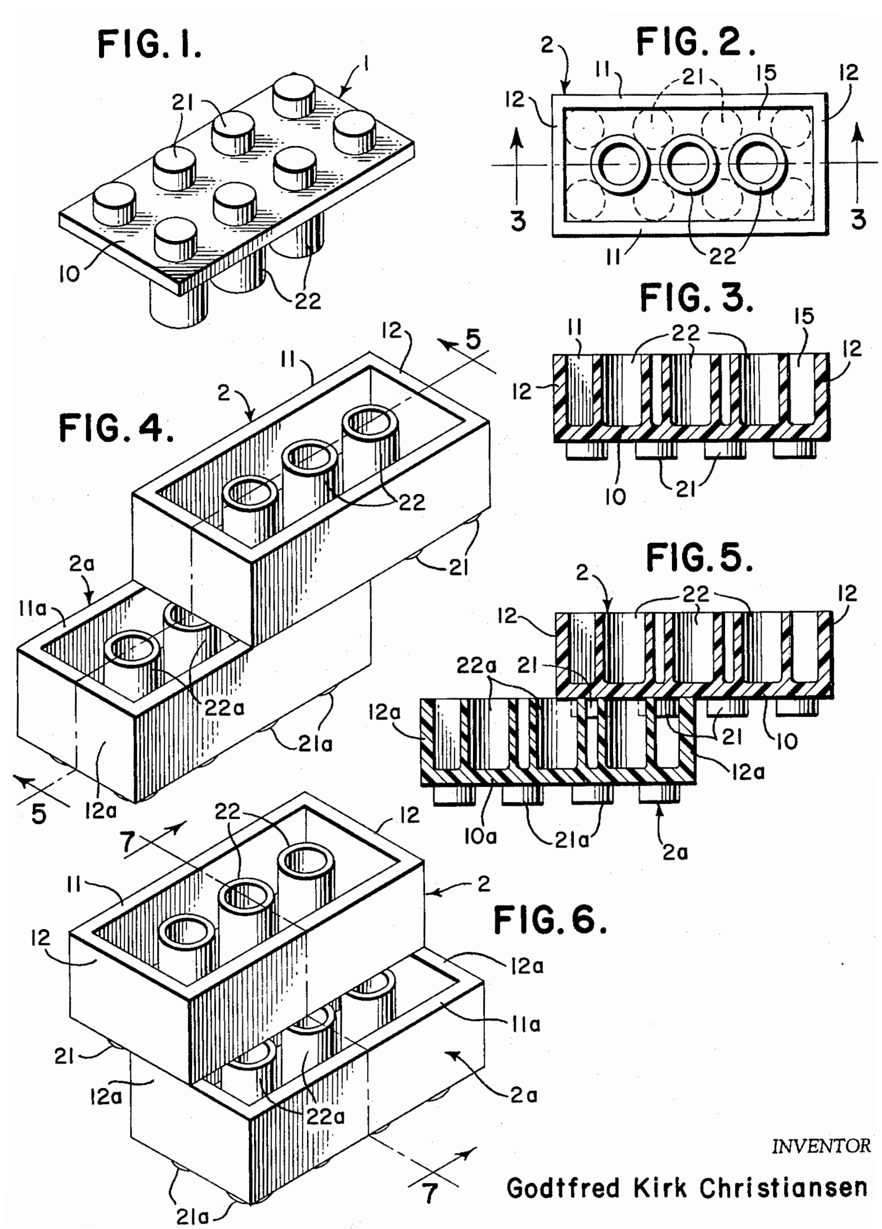


Figure 9.2: Figure from Christiansen's original Lego patent (US Patent #3,005,282), filed in 1958. "[T]he principal object of the invention is to provide improved coupling means for clamping such building bricks together in any desired relative position thus providing for a vast variety of combinations of the bricks for making toy structures of many different kinds and shapes" (Christiansen 1961).[8] *Lego's manipulability stems from innovations in form as well as plastic materials (Lauwaert 2008).*

[8]The original patent emphasis the benefits from the internal column projections. First, they provide the "clamping" power whose importance the patent repeatedly emphasis (see the text quoted in the caption above). Second, the projections allow for "a vast number of possible combinations of adjacent bricks." They don't need to simply stack on one another, but can be offset and still hold on, as the figure illustrates. Third, the "clamping effect …

The variability of loose parts gains force through its opposite: fixity. It is DNA's fixity as well as its variability that give it power as a medium for genetic code. One of Lego's key innovations was improving how well toy building bricks clasped one another, a quality referred to as *clutch power*.[9] The bricks don't merely stack, but grip one another, allowing complex forms to be built up. They are sticky parts. Variability and fixity exist in a figure/ground relationship to one another, each giving meaning to the other.[10] The finely tuned stickiness of Lego bricks strike an ideal balance between ossification and tenuousness, affording their transformation by hand into new stable configurations.

Sticky parts engender manipulability. In Scrabble and magnetic poetry, the board, letter trays, boxy letters, and magnets allow configurations to be built up and persist. In the Berkeley Adventure Playground, architectural structures persist over time, and children can take constructions home. Aggregations of material—of wood, paint, and nails—are constantly built up and torn apart in a kind of internal ecological flux, a process predicated upon sticky parts.[11] *SimCity* is more manipulable than *SimEarth* because its parts are stickier. *SimEarth* lacked

---

is, fundamentally, independent of the presence of the side and end walls," which are provided only to provide a "brick-like appearance." This is also illustrated in the figure in the form of a brick with no walls. The walls, of course, provide additional clutch power, and more configurations for brick coupling, allowing a stud to be gripped between a projection and a wall, as shown in the figure's bottom configuration.

[9]The original source of this phrase is unclear to me, but it abounds in literature by and about Lego (e.g. Lauwaert 2008). The original patent uses the term "clamping" repeatedly.

[10]Consider the loose part of a door knob. The knob rotates relative to the door, which provides a stable ground for the rotation of the knob; the knob itself does not melt in your hand as you turn it, but holds a stable shape.

I'm channeling Sutton-Smith's observation that play's inversions and transformations can only exist against a backdrop of stability (Sutton-Smith 1986). An analogous idea can be found in Langton's argument that the possibility of computation (and life) emerge at a phase transition between fixity and transformation (chaos) (Langton 1990).

[11]It's important to keep in mind that clutch power is deeply contextual; it is relative to the player's tools, skills, and culture, as well as the particular variables one wishes to clamp and the desired scope of the clutching. For example, in the Berkeley Adventure Playground one needs special tools, like hammers and nails, to clamp and unclamp wooden structures materials, but this is desirable within the scope of structures strong enough for people to play with, build, and climb on.

The overall theme of loose parts and clutch power existing in a dynamic with one another reproduces Sutton-Smith's characterization of play as enacting a disequilibrial equilibrium (Sutton-Smith 1999).

fixity—it was, in Braun's words, "self modifying"—frustrating player desire to design and author a world (Braun 2015).[12]

Tangibility allows us to touch, feel, see, and know the materials we manipulate. Tangibility yields the sensual pleasure of touch, of instant feedback, both of a thing at rest and in flux. It is the feel of a plastic brick, its sharp edges and round studs, as well as the click it makes when we join it to another. This click is felt in the hand, seen with the eye, heard with the ear, and experienced as a unified event. Even software has a touch and feel, which computer game designers carefully craft and evoke by interweaving multiple sensory channels (Swink 2009). Players of *SimCity* directly access the main map, virtually touching the simulation's internal map data structure. A hand moves a mouse on a table, a mouse cursor moves across the screen, and a finger is used to click the mouse. A road is built, and the ears and eyes take in the transformation of the landscape. All of this embodied experience is unified into a single sensual phenomena of touching the map.

Tangibility also has social implications. Physical embodiment means that materials have a social life—they can be pointed to, shared, and talked about. The little pewter tokens used in Monopoly—the car, shoe, thimble, battleship, terrier, iron, and canon—are manipulable pivots used to represent players. Manipulability means that a pivot can be touched and talked about, focusing shared attention. We can see the same Lego bricks, hand them to one another, put them in particular places, and talk about them. Tangibles are the primitive elements of shared attentional scenes, as well as the materials of embodied cognition (Hutchins 1995; Clark 1998; Johnson 2007; Tomasello 2010). *SimCity's* main map not only constitutes a tangible shared ground between player and computer, but between players. I can look over your shoulder as you play. Two or more people can look at the same screen, pointing at the same map with

[12] Sticky parts are critical for manipulation generally, but construction play in particular.

their fingers, the mouse cursor, and their words. Tangibility helps create a shared play frame.[13]

Robustness

Appropriation, inversion, and change are central to play, so it is natural that players test, push against, and transform boundaries. Robustness means that boundary pushing is celebrated rather than prevented, punished, or flopped.[14] Robustness means resiliency, gracefully handling extreme manipulations and appropriations.

One can jump outside of the level frame of *Super Mario Bros.* and discover a warp zone; prevent a Sim from going to the bathroom or eating in *The Sims*, which results in them peeing themselves or starving to death; or wreak havoc on a *SimCity* city, causing it to tumble into pandemonium. When Bill Budge gave his players access to a polygon tool in *Pinball Construction Set*, he knew they would construct mathematically degenerate polygons. Rather than worry about how to prevent this, Budge ensured that these broken polygons were gracefully handled.[15]

---

[13]Game tokens illustrate how manipulability affords pivoting. Monopoly's whimsical miniatures serve as pivots for player identity, marking their location on the board. Their manipulability—that they are easily moved and collectively perceived—makes them excellent pivots for player identity. They are easily moved by hand and occupy clearly demarcated spaces, all of which is socially apparent. Specificity, that the car is clearly different from the battleship, means that they can distinguish and represent the identity of different players.

The activities of a City Building classroom are focused upon a shared scene, the tangible model city, which anchors a great deal of thinking, imagining, planning, designing, negotiation, and discussion—even when students are drafting plans, their backs turned to the model.

[14]This includes breaking gracefully. Breaking can be a pleasurable transformation in itself. Such an inversion isn't necessarily bad. Glitches in video games and melted Lego bricks can be marvelous materials for play. These breakages are distortions that give rise to a new set of loose parts. Part of their pleasure is as a subversive transformation, an inversion of the dominant order. But this inversion only exists as a figure against a background of stability. Robustness is concerned with how scaffolds maintain stable play frames, and engender the feeling of safety needed for players to play.

[15]*SimCity's* distributed cellular architecture and agent system gracefully handle contrived situations players might create: buildings can exist in states of partial destruction—the simulation and animation logic carry on just fine. And if the ship's navigation logic fails and it crashes—then it explodes. City Building facilitators must robustly handle the whimsical technological innovations imagined and introduced by players, improvising outcomes and constraints.

The Berkeley Adventure Playground robustly accepts a diverse range of donated play material, some of them quite dangerous. (Most are filtered out and disposed of.) Eclectic donations harbor both risks and possibilities, and must be ingeniously accommodated by the playground. The exuberant and potentially dangerous constructions made by children are robustly handled by staff. Every morning they inspect the playground for unsafe structures, tearing down those they deem unsafe. Children are encouraged to pick up nails and Mr. Dangerous, returning these to the staff, who scrap them, returning many back into raw materials. In this way the freedom of play with loose parts and the undesirable hazards that might result are reconciled. The playground robustly absorbs variability, mitigating risk both to children and to itself as an ongoing institution.[16] Transformation and safety and harmonized.

Robustness engenders a feeling of safety and gives license to transformation, which encourages play. If *Pinball Construction Set* crashed when players made a degenerate polygon, or *SimCity* behaved erratically when players forced contrived cities into existence, or the City Building fictional reality came crashing down when a far reaching intervention was made, players would not feel comfortable in their transformations. Players must be confident that things will hold up.

Celebrate Change

Players are also emboldened to play if their transformations are invited, accepted, and celebrated. Improvisational actors use a technique called “Yes, and…” to accept and build upon each other’s performances. The enveloping acceptance and safety disarms self-censorship, an internalized fear of judgement from ourselves and others. This heightens freedom and

[16]Entrants also sign legal waivers, an effort to mitigate the risk of play which makes the playground more robust in the face of accidents and litigation.

creativity.

Suspending judgement allows players to explore and relish transformation, finding their own meaning and feeling in play. Activity is not micromanaged with closely defined winning and losing states. Players, rather, are given autonomy. This is in contradistinction to games, which define and valorize outcomes (Juul 2003). In play, we set our own agendas and criteria.

SimCity, contrary to Brøderbund's desire, does not boot you from office if your approval rating is low. Instead of judging and refereeing your performance, *SimCity* provides non-binding feedback. Graphs, maps, problem lists, and an approval rating give players the space to define their own agendas, and the materials for measuring and interpreting their own play. The same happens in City Building Education. Facilitators guide students into defining their own criteria and help them to see their city's activity and consequences, providing students with the raw material to evaluate their performance. Students are allowed to fumble, make mistakes, and learn from these. This is made possible by the facilitators, who withhold judgement and intervention.[17]

When volunteering at the Berkeley Adventure Playground I found myself in an awkward position of having to suspend judgement. I helped a staff member disassemble an undesirable structure, and carefully collected the nails I extracted. I judged loose nails to be dangerous, but this was wrong; the correct place for the nails was simply on the ground, in the grass, so the kids could collect the nails themselves and exchange them for tools. Suspending judgement in this way made room for adventure play. Similarly, the staff don't take down unsafe structures immediately, as they are built, but in the morning. How, otherwise, would a safe structure ever

[17]It is only when certain limits are exceeded that the facilitator steps in, for example when a decision goes over time the "federal government" steps in and makes the decision itself. But the consequence reinforces the underlying point: this is punishment in the form of loss of autonomy, which students find highly motivating. At the end of building, players of Purium Instant City reflect upon and discuss what happened, a process guided by the facilitator, but conducted by the players.

get built? By suspending judgement, transformation and variability attain room to breathe.

Another way to suspend judgement and celebrate change is by suspending social norms, giving license to the performance of actions that might otherwise be considered inappropriate. As in festival play, the rails of self- and social- censorship come off. *Katamari Damacy,* for instance, is resplendent with stylized inversions that license and encourage players to roll up whole cities into colorful balls of junk.

Transformation is also celebrated by rich and generous feedback. Disproportionate feedback, such as *SimCity's* automatic embellishments of player input, *KidPix's* exuberant responses, or a bouncing ball's unpredictable antics, all produce a heightened sense of agency and reward. Vibrant sound and animations accompany Mario's transformation from small to big, his smashing of bricks, and his deaths. Such generous and overflowing feedback is sometimes referred to as *juiciness.* We are egged on, surprised, and delighted. Buoyant feedback inspires players to revel in transformation (Morris 1966/1999).[18]

Entrain

To entrain players is to draw them into play. Aliveness, a sense of rhythmic change and surprise, works to signal, solicit, and regulate play. Once inside, we find ourselves guided by plans, roles, and rules—structures we can follow, negotiate with, and rebel against. Boundaries frame and envelop us, focusing and coordinating activity into a shared play frame, buffering not-play from play.

[18]Manipulation can be augmented, amplifying and automating play's transformations. *KidPix* introduces interventions that exaggerate and surprise. It amplifies and distorts the pleasure of painting introduced by *MacPaint*. Disasters and destruction in *SimCity*, as well as disasters inserted by the facilitator of City Building, celebrate the transformation and transformability of the model cities. *SimCity's* hybrid agency amplifies player initiated transformation, and its mini-maps present alternate transformed views of the city: the city recolored according to crime, traffic, pollution, or land value.

Figure 9.3: Canid play bow (Bekoff 1974b).

Aliveness

The polyrhythms and buzzing activity of *SimCity* evoke a living world that arouses curiosity, anticipation, and focused engagement. So does the effervescent activity of cellular automata, perched between predictability and surprise, lifelessness and chaos. City Building worlds are alive. The federal government intervenes. Students leave class and return the next day, and find their city in ruins, struck by disaster. A dog jerks and runs back and forth, signaling and soliciting play. This is transformation and change for the sake of transformation and change, a dance of uselessness that signifies and enjoins play. Aliveness evokes wonder, curiosity, compulsion, and delight. It is to be moved by music, art, and animation; to find your hips moving to the music, or feel the exhilarating fear and thrill of flying down a zip line. It can be found in the caprice of a bouncing ball, whose laughing trajectory—just on the cusp of predictability—teases and provokes.

Play's paradoxical inversions manifest in alternating repetition:

> The most elementary forms of play involve successive repetition of alternating states

> of up and down, to and fro, in and out, back and forth. You climb up and you jump down. You throw the ball and it comes back. You fill up the bucket and you empty it out. With increasing maturity these antitheses are absorbed into larger cultural play patterns of approach and avoidance (peek-a-boo), order and disorder (knocking down block structures), chase and escape, attack and defend, score and score, win and lose (Sutton-Smith 1986).

Alternation and uncertainty solicit engagement, signal playfulness, and hold our attention. Stories, jokes, games, and play build anticipation and then surprise and delight us (Huizinga 1955; Lammers 1986; Murray 1997; Stanton 2012). The Japanese garden design principle of meigakure, or "hide-and-reveal," is one such way to tease and evoke. A watercourse or pathway winds its way into and out of view, ducking behind trees and rocks, only to surprise us later.[19] The master Japanese garden designer David Slawson writes that "[a] device like this almost demands that the designer have a genuine sense of play, and revel in the way the garden can at turns entice and surprise the observer" (Slawson 1991). Aliveness—dynamic and unexpected change—entices and surprises us with secrets, revelations, and surprises.

The computer's aliveness exudes a "holding power" that delights and entrances (Turkle 1984). We see the hold of aliveness in the random reward schedule of video games, which along with regret, invites and compels play (Loftus and Loftus 1983). Just as an older playmate might adjust their play in order to adapt to a less sophisticated partner, aliveness can engender flow. A delicate balance between challenge and ease welcomes us into the pleasurable zone of flow (Csikszentmihalyi 1991).[20] The carefully calibrated reinforcement patterns of video

[19] Or perhaps a viewer is led along a pathway whose destination and vistas are periodically obstructed, enticing our imagination and involvement. This technique is used in video games as well as gardens, for example the sprawling adventures of *Super Mario Bros.*—with their hidden vines, warp zones, and pipes that interconnect unseen spaces (Gingold 2003), or in Marc ten Bosch's forthcoming 4-dimensional puzzle game entitled *Meigakure*. Tiny spaces are instilled with wonder and expansiveness.

[20] Differences in skill level can be harmonized by self-handicapping, which is evident in the game Go's use of handicap stones, and in how some animals adjust their play to accommodate younger and less experienced playmates (Bekoff and Allen 1998).

slot machines invite players into a delicately constructed zone of obliteration, an escape—a compelling immersion into nothingness, a kind of dark flow (Schüll 2012).

Negotiable Structure

Play liberates us from ordinary constraints, but this freedom also comes with a psychic cost. We can become paralyzed and overwhelmed by too much choice (Sutton-Smith 1972; Schwartz 2009). Structure is the existential balm for an overwhelming freedom of choice. Rather than throw the player into all possible activities and affordances, structure constrains, gives meaning, and eases play.

Themes and plans frame play, providing a structure to cling to. *SimCity's* scenarios place players into scenes, and give them a part to play, for example rebuilding Tokyo in the wake of a Godzilla attack. Lego scaffolds make believe engagement via box art that establishes an imaginative scenario and detailed action plans in the form of step-by-step assembly instructions. Semi-assembled starter boards in *Pinball Construction Step*, like the partially constructed defaults of the *Spore Creature Creator*, serve as plans in outline, templates that invite completion. Through challenges, prompts, and guidance, players are entrained into play.

Structures guide us through play. Gifts are a structure that reward and gradually reveal possibilities. The term comes from Friedrich Fröbel, the inventor of Kindergarten. Over the course of multiple years, children are given a succession of gifts to play with, beginning with a soft ball and ending with peas-work, a construction material made of soft joints (peas) penetrated by toothpick-like objects, which can be used to construct complex forms (Brosterman 1997). *SimCity*, by setting the cost of some infrastructure objects very high, like the stadium or seaport, effectively gates their use, turning them into gifts. Tools in the Berkeley Adventure

Playground are gifts. To earn a saw, hammer, or paint one must first perform some actions, exhibiting a base level of mastery and understanding. In the tradition of game design, gifts are thought of as something to unlock, or tech trees to traverse. Power ups such as the mushrooms that make Mario big, or the fire flower that enables him to shoot fire, can be thought of as gifts. So can the levels and spaces that unfold during the course of Mario's journey. Gifts are a technique for structuring play. They gradually introduce affordances, ensure fluency, guide attention, and reward play.

Structure also guides and entrains multiple players with one another, precipitating shared understanding and activity. The rules and materials of a game, for instance, enable multiple players to undertake the shared activity of game playing.

In play, we adhere to structure and follow rules, but we also invert and negotiate them. Structure provides a stable backdrop for inversion—it is something to invert, to transform, to play with. *SimCity's* naturalism is a structure that a fantastical monster rebels against; its orderly cities can be smashed down; the city treasury can be subverted by typing "fund," which injects $10,000 into the treasury. (The documentation refers to this as embezzlement). *SimCity* invites inversion, but also pushes back. Use this cheat code many times and *SimCity* will "punish for cheating," (as a code comment says), and unleash an earthquake.

But unlike Monopoly, whose open structure invites and accepts house rules, it is virtually impossible for *SimCity* players to negotiate with the underlying simulation mechanics.[21] Monopoly and City Building Education are simulated by people who can compromise with one another, but *SimCity's* inner structure is closed, opaque and non-negotiable, a fact bemoaned

[21] Monopoly contains many semi-conventional house rule variations, especially surrounding free parking. In the official rules nothing happens when players land on free parking, but in the variations players collect money—a jackpot of penalty fees, taxes, or some other amount of money. Monopoly is a packaged product, but it is negotiable because its guts are visible, and it is animated by people one can negotiate with.

by critics such as Alan Kay (Kay 2007). This stems from the complexity of the program, which requires sophisticated tools and expertise to modify, as well as the intellectual property considerations that animate software enterprises.[22]

Boundaries

Play is separate. Sometimes the boundaries that buffer it are manifestly clear, as in the fence that demarcates and controls entry to the Berkeley Adventure Playground, or they are gossamer, as in the Jejune Institute's porous separation of non-play and play. Boundaries perform a number of important functions, which the metaphor of a sandbox helps to illustrate. Sandbox walls demarcate space, focusing the sand inside so it can be played with. The walls also protect the outside, keeping sand from getting out and annoying everyone else.

A boundary is a threshold, bridging play and not-play. Ordinary things and events take on new meanings, are transformed—we become new selves, a "fictitious "I" " with new desires (Vygotsky 1978). This threshold might be a mask, a team jersey, the fence of a playground, lines on a sports field, the social context of a tournament, or the reframing of life through the lens of an alternate reality play experience. But the boundary as threshold does much more

[22] Kay has advocated for more open and tinker-able systems like SmallTalk throughout his career, but such designs run counter to the capitalist desire to turn code into a sealed product, its intellectual property protected through opacity and closed-ness.

In theory computers offer a great deal of flexibility for rule rewriting. The computer, after all, is a strictly rule driven machine. In practice, powerful limitations stand in the way. First, software sellers don't want you to look inside their products. Software products are typically black boxed, their precious intellectual property sealed up, packaged, and sold as a product. Room for use beyond the seller's intentions is circumscribed. Mods, of course, are a counterexample to this, but even then a clear separation is made between closed and open parts. Second, negotiating with computers is hard. Complex software is animated by intricate software code that is hard to modify, even for developers. Even at the diagrammatic level of abstraction at which I have presented *SimCity*, the system is complex and hard to understand. Third, computers are not human players. While City Building is animated by people following rules described in human terms, *SimCity* is animated by rules described in computer code. The computer knows nothing about energy, pets, and new technologies—not unless you code it. It is much easier to improvise make believe rules with a person than with a machine, as one can negotiate with a person in terms a person understands.

than reframing not-play as play.

Thresholds engender safety, a precondition of play. They make entering and leaving, if not easy, then at least clear. Garvey notes that in the excitement of make believe play "children may suddenly feel uneasy and in need of reassurance." This uneasiness is dispelled by reaffirming the boundary between pretend and real (Garvey 1970/1990), the distinction between not-play and play. We are comforted by the knowledge that we can step outside, or take off the mask. To engage in the active creation of belief, one must feel safe, comfortable in the scope of transformation.

Boundaries focus attention and establish a shared grounding for play. The walls of the Berkley Adventure Playground tell us where we can build, which materials to use, what we can climb on, and who is there for similar purposes. Just as a sandbox focuses grains of sand, keeping them in place so we can dig tunnels and build castles, boundaries help us to see and agree on who and what is in play together. We can see who stands within the lines of the play field, who wears the uniforms, and who stands just outside. In City Building Education, the extent of the miniature city marks the model as a separate world, one that focuses everyone's attention. This then enables the imagined city to extend beyond the physical model itself, into role play. Students can bring sections of the city model home with them. The immersive threshold is in our imaginations, but it is scaffolded by a concrete perimeter, visible to all.

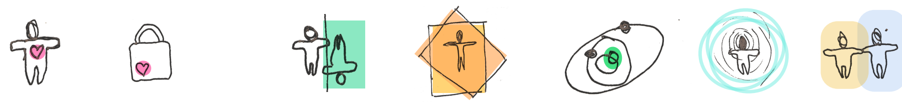
Boundaries also protect the outside of play experiences, those who are not playing. As Victor Turner points out,

> Playfulness is a volatile, sometimes dangerously explosive essence, which cultural institutions seek to bottle or contain in the vials of games of competition, chance, and strength, in modes of simulation such as theatre, and in controlled disorientation, from roller coasters to dervish dancing… (Turner 1986).

Without proper boundaries, play leaks out, disturbing those who are not playing. Boundaries protect those who are not playing from its disorder and unruliness, bottling it up and giving it a protected space for expression. The original junk playground was surrounded by a two meter high dike covered by a fence, and was decorated on the outside with "wild roses, hawthorn and acacia," shielding its unruly and ugly interior from the disapproving view of neighbors (Sørenson 1951). Displeased neighbors have forced the closure of many such junk or adventure playgrounds (Spivack 1969; Donald 2012). To cross the threshold of the Berkeley adventure playground one must do more than walk through an open gate, one must sign a liability waiver, and guardians must attend to their children in the playground at all times. Festivals are confined in time and space, bottling them up, and limiting their effects. The annual Burning Man bacchanal began in 1986, a small affair on San Francisco's Baker Beach, but in 1990 the police prevented the effigy from burning, ruling it a fire hazard, which prompted relocation to the Black Rock Desert in Nevada. A festive conflagration of Burning Man's scope requires an appropriately scaled buffer, which the Black Rock Desert, one of the biggest sandboxes imaginable, provides. Crossing the threshold is not an easy task—one must make a pilgrimage into the harsh desert, a journey that marks and enacts a commitment to the festival's deep and hazardous transformations.[23]

Boundaries are always porous. Play can have ulterior motives (Sutton-Smith and Kelly-Byrne 1984; Schechner 1988). The Jejune Institute appropriates the urban environment into its pretend world, and because this appropriation is deliberately porous, unclear about the interface between pretend and not pretend, a transformational leakage occurs, enchanting everything. We leave, but we and the world we live in are transfigured, changed by play (Holland et al. 1998). We leave the sandbox, but enchanted granules of sand trail us home, embedded in our shoes, suspended in our hair, or caught in the folds of our clothing.

[23]http://burningman.org/timeline/, accessed Nov. 5, 2015.



	Intrinsic	Safe	Transformational	Separate	Structured Variability	Attitude	Shared
Encourage Appropriation							
Reframe			recontextualize	context	guidance	enchantment	validate and teach play
Stylize	ease interiorization	easily grasped	refractive stylization	non-ordinary		paradox signals play	clear shared attentional scene
Pivots	subject to imagination		provoke and absorb player meanings				
Afford Transformation							
Manipulability	tactile pleasure		recombination		loose + sticky parts		embodied shared scene
Robustness		resilience inspires confidence				license to transform	
Celebrate Change	encourage self-directed activity	mitigate self-censorship	transformation is celebrated	no such thing as failure	variation is celebrated	joyful, exuberant	
Entrain							
Aliveness						arouse engagement	soliciting playmate
Negotiable Structure	rules bend to desire	balm on overwhelming choice	structure is something to transform		guide play		mutual entrainment
Boundaries		clear scope + threshold	threshold of transformation	buffer	structure		shared grounding

Figure 9.4: Play design principles and techniques mapped to the characteristics of play.

Chapter 10

Conclusion

In this brief conclusion I summarize the research contributions of the dissertation. The contributions come in three primary flavors: case studies, methods, and theory.

Case Studies

Six case studies make up the bulk of the dissertation. Chapters 1 and 2 establish the theoretical framework and methods I use. Chapter 9 generalizes over the case studies, articulating principles and techniques of play design. The case studies are contributions in themselves, which I summarize below.

System Dynamics

System dynamics is a representational technique that has been used to model everything from social systems to spiritual experiences, and has its origins in servomechanisms built in the service of war.

This case study is a contribution to the history of science. I show that while Jay Forrester thought he was producing a serious scientific instrument, the element of play and make believe was a vital animating force in his project. While Forrester's work has already been critiqued and historicized, this case study looks at it from a new perspective: as an imaginative simulation practice. I show how this representational practice formed, and explain why it lends itself so well to creative appropriation.

Cellular Automata

This case study is a contribution to our understanding of how simulations represent. I explore why cellular automata are such an appealing instrument for representation, and how it is that they have come to stand for such a diverse set of phenomena—everything from galaxies and sand dunes to social insects and ecosystems.

While many surveys of cellular automata have been written over the years, these are positioned from within the field. This analysis stands outside of the field, and looks at it as a representational practice. I argue that in order to understand the appeal and spread of cellular automata, we must understand their underlying commitments, as well as the forces that shaped their evolution.

Along with the system dynamics chapter, these two case studies make the argument that it is by attending to the evolution of a software representation that we can understand the underlying commitments that give these representational practices their power and appeal.

SimBusiness

This case study is a contribution to the history of games and software. I argue that appropriative and self-directed play powered not just the creation of *SimCity* and the rise of Maxis, but also led to Maxis's downfall and sale to Electronic Arts in 1997.

While there are plenty of popular histories about the creation of computer games and computer game companies, these are mostly journalistic in nature. I argue that play and capitalism are at odds with one another, and must be creatively reconciled. Play fuels the creation of new products, but also challenges the institutions that try to channel and market those products. Will Wright's creative life is marked by self-directed playful appropriations. For a while, Maxis successfully reconciled Will Wright's internally directed play with the desires of capitalism and players, but this tension ultimately caused the company to unravel.

It is also, to my knowledge, one of the most comprehensive accounts of the formation of an entertainment product and an entertainment software company. It revises the popular cliched histories of Maxis, with Will Wright as a singular hero, and shows how it was only through the efforts of an entire team assembled by Jeff Braun that Wright's private plaything was brought to the wider world. This case study has an ethnographic bent, and is informed by well over ten hours of original interviews, especially from voices that have not been widely heard.

SimCity

The case study is a close reading of the original *SimCity*. It is a contribution to our understanding of how code mechanisms give rise to believable complex simulation models. It also explores how the design of computer simulations afford play.

As a contribution to software studies, it is the most comprehensive close reading of a complex software artifact I know of. Other studies tend to look at much simpler programs (Mateas and Montfort 2005; Montfort et al. 2012), or do less comprehensive readings (Wardrip-Fruin 2009). In undertaking this study, I have also made methodological contributions to the study of software, which I summarize later.

The Software Case Studies

Just as *SimCity* amalgamates a diverse collection of representational techniques, the aforementioned case studies—system dynamics, cellular automata, SimBusiness, and *SimCity*—build upon one another, and become something greater in combination. Taken together, these four case studies are a substantial contribution to the humanistic study of software.

Their scope is uniquely comprehensive:

- **Time.** I look at how serious simulation practices that arose in the mid 20th century influenced a popular entertainment product released in 1989. We can see specific representational techniques arise, evolve, and spread over multiple generations of simulation makers.

- **Depth**. *SimCity* is a non-trivial piece of software. Most close software analyses tend to take either toy programs as their objects of study, or lightly touch upon multiple complex programs. This is the first time, to my knowledge, that such a complex piece of software has been interpreted so closely and comprehensively. I descend not just into the intricacies of *SimCity's* technical design, but deeper still, to probe the nuts and bolts of the simulation techniques it builds upon.

- **Breadth.** These case studies span history, code, representational practices, and experience—an atypically broad approach. *10 Print CHR$(205.5+RND(1)): Goto 10* demonstrates how a single software program can be studied from all these different avenues (Montfort et al. 2012). The program, however, is very simple, and such broad multi-pronged approaches are rare.[1]

City Building Education

The City Building Education case study is a contribution to the history of this particular pedagogical technique. I analyze it through the lens of play, which provides a valuable case study of non-computerized play. By generalizing over computerized and non-computerized playable city simulations, I am better equipped to draw comparisons and articulate play design principles. This is the first of two case studies about non-computerized play, which helps ensure my play design principles are sufficiently general. Like the adventure playground and SimBusiness chapters, this case study draws heavily on interviews and ethnographic methods.

Adventure Playground

This is a close reading of a playground, and uses ethnography, interviews, and historical research. The adventure playground chapter is a contribution to our understanding of what makes these unusual playgrounds both fragile and appealing.

I argue that adventure playgrounds succeed and fail because of how they reconcile oppositional forces between safety and danger, players and non-players, free and guided play, and play and not-play.

These tensions, I argue, are present to a degree in all playgrounds, and in all kinds of play. Attending to these concrete examples helps us to reinterpret basic theoretical points

[1] Humanistic studies of software code, or computer design more broadly, are relatively new, and it is rarer still to find those that also attend closely to history. Notable examples include (Agre 1997; Edwards 1997; Montfort et al. 2012). Each work has its own particular disciplinary inflections: Edwards delves into computer design, but not software, while Agre is focused on the "history of ideas" more than the history of people who shaped those ideas. The platform studies series in the MIT Press also includes examples of works that span such disciplines (e.g. Montfort and Bogost 2009; Altice 2015).

about play, as well as understand how to design for it. For example, the adventure playground enriches our understanding about the function of boundaries and the separateness of play.

Methods

Software Studies

When I began this project with the idea of studying *SimCity*, I was directed by Michael Mateas and Noah Wardrip-Fruin to the emerging field of software studies. As I came to learn about this emerging discipline, I found an inconsonant set of approaches and theories about why one should study software, what it means to do so, and how to go about it. The methodological framework I lay out in chapter 2 is the map I wish I had at the onset.

The framework surveys and synthesizes a diverse set of perspectives on the study of software. The regularization of what I found to be an awkwardly diverse set of case studies, theories, and methods constitutes a contribution to the nascent field of software studies. Seen as ethnography, the case studies provide the data from which I generalize the categories of analysis. My methodological framing, in other words, articulates between the existing literature on software studies and my own case studies.

Reverse Diagrams

I contribute one very specific methodological approach to the study of software, and that is the *reverse diagrams* technique I describe in chapter 2, and demonstrate in chapter 6 as well as the appendix.

The trouble with code is that it is long, sprawling, and complicated. It evolves organically, and so tracks the evolving thought of developers. This complicates analysis, making it difficult to get an overview of code in order talk and think about it. I introduce a technique called *reverse diagrams*, which is inspired by Stone Librande's one page design documents, and show how to use reverse diagrams to translate and map code for analysis. This is in the spirit of Latour, who argues that figures allow scientists to reduce and embed large and complex phenomena into flat documents, such as this one, and then disseminate them.

Beyond making more diagrams, there are two kinds of future work that logically follow in these methodological footsteps: live diagrams, generative diagrams, and tools that support each of these. The diagrams I have made are static, and are meant to be embedded in a print document. It would be nice to be able to hook up a diagram, say the Data Flow Map, to a running instance of *SimCity,* and see the diagram animate in response to the program. (In this example, the program could live inside of a Macintosh emulator.) A live diagram like that would be in the spirit of Bret Victor's dynamic visualization of Dan Amelang's Nile graphics renderer (Kay 2012; Ohshima et al. 2012; Victor 2013). The diagrams become, in this case, a kind of highly domain specific debugger. The diagrams could also afford agency over the running program. By interacting with the diagram, one could control the flow of execution, pausing and stepping through time, or manipulate variables. Now that *SimCity* has been translated and mapped as a set of diagrams, it would be nice to have a tool that would allow diagrams to be imported, marked up, and elaborated with bindings to an executable.

Generative diagrams push the idea one step further. If a diagram is a good overview of a program's internal organization, then why not use living diagrams in the specification of a program? In this further elaboration I am imagining, the diagrams house and organize the logic that constitutes a program. Unlike Librande's original one page design documents, which are read only by people, diagrams became the essential organizing structure of a program description, and are used to design, configure, and interpret programs. Rather than living in the temporal periphery, in the form of design documents or reverse diagrams, diagrams move to the center of software development, linking design, construction, and interpretation.

Theory

Software

The framework for studying software I have knit together out of the existing literature on software studies and my own case studies constitute not just a methodological contribution to the study of software, but a theoretical framing for conceiving of software simultaneously as a social practice, medium, and artifact. While a growing body of studies seek to interrelate software as social process, history, code, representation, and experience (e.g. Agre 1997; Edwards 1997; Wardrip-Fruin 2009; Montfort et al. 2012; Altice 2015), my contribution is in offering a theory which fits these heterogeneous perspectives together (chapter 2).

Characteristics of Play

Play is a challengingly ambiguous subject that has been studied from a wide variety of disciplines (Sutton-Smith 1997). Following in the methodological footsteps of Jesper Juul's framing of games (Juul 2003), I survey the literature on play and distill it into seven characteristics (chapter 1). My sources are drawn from influential conceptions of play and come from a wide variety of disciplines, and include everything from animal behavior and philosophy to psychology and performance studies. This conception of play is then used throughout the dissertation, giving me firm ground to stand on. This review, reconciliation, and distillation is a theoretical contribution to the theory of play.

Play Design Principles

The play design principles constitute a contribution to theory as well as practice. These play design principles are synthesized from the six case studies, and describe three top level design principles—encourage appropriation, afford transformation, and entrain—and nine techniques for designing for play (chapter 9). This theory is analytical, and helps us to understand how and why certain play designs work. It is also intended to be generative, and function as a guide for designers. This analytical-generative pairing should also afford design transfer, helping us to perceive the design at work in a particular plaything or play practice, abstract it, and generatively apply it to a new project.

Game Studies

I have deliberately positioned this research outside of the field of game studies. Nonetheless, this research offers contributions to game studies. Methodologically, I demonstrate the value of deeply examining a single toy/game, including attending to its code and history. My methods range from ethnography (extensive developer interviews) and history, to attending to experience and code (software studies).

By turning the table on games and play, and seeing games as a genre of play, rather than seeing play as a consequence, ingredient, or outlier of games, I offer an alternative theoretical framing for games. To support play, games employ particular play design principles (chapter 9). Zooming out of games, and seeing them in proper relation to the parent category of play, gives us a new way to think about the characteristics of games. For example, while rules loom large in discussions of games, and iconoclasts have mounted challenges to this framing, these challenges still lean heavily on rules in accounting for the affinity between play and computation (Sicart 2011; Sicart 2014). As I have shown in chapter 2, there is much more than procedural rule following to the affinity between computers and play. As I touch upon in chapter 9, the theory of play design can help us to more fully appreciate the family resemblances between games and play, and reconsider and re-theorize the function of game elements such as boundaries, rules, and stylization.

References

1wErt3r. *A Comprehensive Super Mario Bros. Disassembly*. URL: HTTPS://GIST.GITHUB.COM/1WERT3R/4048722 (visited on 11/19/2014).

A Report on the Art and Technology Program of the Los Angeles County Museum of Art, 1967-1971 (1971). SBN: 87587-044-9. Los Angeles County Museum of Art, 1971.

Adamatzky, Andrew, ed. (2010). *Game of life cellular automata*. Springer, 2010.

Adams, Douglas (1979). *The Hitchhiker's Guide to the Galaxy*. Gollancz, 1979. 192 pp.

Adams, Paul C. (1998). "Teaching and learning with SimCity 2000". In: *Journal of Geography* 97.2 (1998), pp. 47–55.

Adventure Playground: Park Design Modification University Community Park (2012). Reference File No. 00541581-PPD. Irvine, California: City of Irvine and EPT Design, June 6, 2012.

Agre, Philip E. (1997). *Computation and Human Experience*. Cambridge University Press, July 28, 1997. 392 pp.

Ahl, David H. (1973). *101 Basic Computer Games*. Digital Equipment Corporation, 1973. 249 pp.

Albrecht, Bob (1972). *My Computer Likes Me: When I Speak in BASIC.* Dilithium Press, 1972. 61 pp.

——— (2014). *Interview by Chaim Gingold*. May 4, 2014.

Allen, Lady Marjory Gill (1946). "Why Not Use Our Bomb Sites Like This?" In: *Picture Post* (Nov. 16, 1946), pp. 26–27.

——— (1968). *Planning for Play*. Norwich: Jarrold and Sons Ltd., 1968. 140 pp.

Altice, Nathan (2015). *I Am Error: The Nintendo Family Computer / Entertainment System Platform*. MIT Press, May 15, 2015. 439 pp.

Anderson, John (1983). “Who Really Invented The Videogame?” In: *Creative Computing and Arcade Games* 1.1 (1983), pp. 8–.

Apple Human Interface Guidelines (1987). *Apple Human Interface Guidelines: The Apple Desktop Interface*. Addison-Wesley, 1987–1988. 164 pp.

Apple II (2016). In: *Wikipedia, the free encyclopedia*. Page Version ID: 702520540. Jan. 3, 2016.

Asimov, Isaac (1951). *Foundation*. Vol. 1. Gnome Press, 1951.

——— (1952). *Foundation and Empire*. Vol. 2. 1952.

——— (1953). *Second Foundation*. Vol. 3. 1953.

Atari (2016). In: *Wikipedia, the free encyclopedia*. Page Version ID: 705289295. Feb. 16, 2016.

Avedon, Elliott M. and Brian Sutton-Smith (1971). *The study of games*. J. Wiley, 1971. 552 pp.

Badham, John (1983). *WarGames*. Feature film, 114 minutes. MGM, June 3, 1983.

Ball, Philip (2001). *The self-made tapestry: pattern formation in nature*. Oxford University Press, July 1, 2001. 316 pp.

Banks, Edwin Roger (1971). “Information processing and transmission in cellular automata”. Ph.D. thesis. Boston, MA: Massachusetts Institute of Technology, 1971.

Bartle, Richard (1996). “Hearts, clubs, diamonds, spades: Players who suit MUDs”. In: *Journal of MUD research* 1.1 (1996), p. 19.

Barton, Matt and Bill Loguidice (2009). *The History of the Pinball Construction Set: Launching Millions of Creative Possibilities*. Feb. 6, 2009. URL: http://www.gamasutra.com/view/feature/132316/the_history_of_the_pinball_.php (visited on 06/25/2013).

Bates, Joseph (1994). “The role of emotion in believable agents”. In: *Communications of the ACM* 37.7 (1994), pp. 122–125.

Bateson, G. (1955). “A theory of play and fantasy”. In: *Psychiatric research reports* 2.39 (1955), pp. 39–51.

Bayless, Don (2016). *Interview by Chaim Gingold*. Mar. 23, 2016.

Beecroft, Stu (1983). *Episode 1: Getting Started*. Vol. 1. 12 vols. Bits and Bytes. Television program, 28 minutes. 1983.

Bekoff, Marc (1972). “The Development of Social Interaction, Play, and Metacommunication in Mammals: An Ethological Perspective”. In: *The Quarterly Review of Biology* 47.4 (Dec. 1, 1972), pp. 412–434. JSTOR: 2820738.

——— (1974a). “Social play and play-soliciting by infant canids”. In: *American Zoologist* 14.1 (1974), pp. 323–340.

——— (1974b). “Social Play in Coyotes, Wolves, and Dogs”. In: *BioScience* 24.4 (Apr. 1, 1974), pp. 225–230. JSTOR: 1296803.

——— (1977). “Social Communication in Canids: Evidence for the Evolution of a Stereotyped Mammalian Display”. In: *Science*. New Series 197.4308 (1977), pp. 1097–1099. JSTOR: 1744147.

Bekoff, Mark and Colin Allen (1998). “Intentional communication and social play: how and why animals negotiate and agree to play”. In: *Animal play: Evolutionary, comparative, and ecological perspectives* (1998), pp. 97–114.

Bell, David and Barbara M. Kennedy (2000). *The Cybercultures Reader*. Psychology Press, 2000. 810 pp.

Bellman, Richard, Charles E. Clark, Donald G. Malcolm, Clifford J. Craft, and Franc M. Ricciardi (1957). “On the construction of a multi-stage, multi-person business game”. In: *Operations Research* 5.4 (1957), pp. 469–503.

Bengtsson, Arvid (1972). *Adventure Playgrounds*. Ed. by Arvid Bengtsson. New York: Praeger Publishers, 1972. 167 pp.

Bentham, Jeremy (1871). *Theory of Legislation*. Ed. by Etienne Dumont. Trans. by R. Hildreth. Trübner, 1871. 496 pp.

Berlekamp, Elwyn R., John H. Conway, and Richard K. Guy (2004). *Winning Ways for Your Mathematical Plays, Volume 4*. 2nd edition. Natick, Mass: A K Peters/CRC Press, Mar. 30, 2004. 224 pp.

Bertelsen, John (1953). “The daily round on a junk playground”. In: *Danish Outlook* 6.6 (1953), p. 690.

Bertelsen, John (1972). “Early Experience from Emdrup”. In: *Adventure Playgrounds*. Ed. by Arvid Bengtsson. New York: Praeger, 1972.

Bhatnagar, Ranjit (2012). *@Pentametron*. Twitter bot. 2012.

Bleecker, Julian (1995). “Urban Crisis: Past, Present, and Virtual”. In: *Socialist Review* 24.1-2 (1995), pp. 189–221.

——— (2004). “The Reality Effect of Technoscience”. Ph.D. thesis. Santa Cruz, CA: University of California Santa Cruz, Sept. 2004.

Bogost, Ian (2005). “The rhetoric of exergaming”. In: *Proceedings of the Digital Arts and Cultures (DAC)* (2005).

——— (2007). *Persuasive Games: The Expressive Power of Videogames*. MIT Press, June 22, 2007. 463 pp.

——— (2008). “The rhetoric of video games”. In: *The ecology of games: Connecting youth, games, and learning* (2008), pp. 117–140.

——— (2009). *Persuasive Games: The Proceduralist Style*. Jan. 21, 2009. URL: HTTP://WWW.GAMASUTRA.COM/VIEW/FEATURE/3909/PERSUASIVE_GAMES_THE_.PHP (visited on 05/30/2012).

Bolt, R. A. (1979). *Spatial Data-Management*. DARPA Report. Cambridge, Mass.: MIT, Architecture Machine Group, Mar. 1979.

Bos, Nathan (2001). *What do game designers know about scaffolding? Borrowing SimCity design principles for education*. Ann Arbor, MI, USA: University of Michigan and CILT, PlaySpace Project, 2001.

Bowker, Geoffrey C. and Susan Leigh Star (1999). *Sorting Things Out: Classification and Its Consequences*. MIT Press, 1999. 377 pp.

Brand, Stewart (1972). “Spacewar: Fanatic life and symbolic death among the computer bums”. In: *Rolling Stone* 7 (1972), pp. 50–58.

——— (1987). *The Media Lab: inventing the future at MIT*. Viking, 1987. 336 pp.

Braun, Jeff (2015). *Interview by Chaim Gingold*. May 19, 2015.

——— (2016a). *Interview by Chaim Gingold*. Feb. 6, 2016.

——— (2016b). *Message to the author*. E-mail. Feb. 2016.

Bremer, Michael (2015). *Interview by Chaim Gingold*. July 17, 2015.

Brennan, Susan E. (1991). "Conversation with and through computers". In: *User modeling and user-adapted interaction* 1.1 (1991), pp. 67–86.

Brooks, Rodney A. (1986). "A robust layered control system for a mobile robot". In: *Robotics and Automation, IEEE Journal of* 2.1 (1986), pp. 14–23.

——— (1991). "Intelligence without representation". In: *Artificial intelligence* 47.1 (1991), pp. 139–159.

Brosterman, Norman (1997). *Inventing Kindergarten*. In collab. with Kiyoshi Togashi. H.N. Abrams, Mar. 1, 1997. 166 pp.

Brown, John Seely, Allan Collins, and Paul Duguid (1989). "Situated cognition and the culture of learning". In: *Educational researcher* 18.1 (1989), pp. 32–42.

Bruner, Jerome S. (1986). *Actual Minds, Possible Worlds*. Harvard University Press, 1986. 220 pp.

Budge, Bill (1981). *Raster Blaster*. Apple II program. BudgeCo, 1981.

——— (1982). *Pinball Construction Set*. Apple II program. BudgeCo, 1982.

——— (1983). *Pinball Construction Set*. Apple II program. San Mateo, CA: Electronic Arts, 1983.

——— (2013). "Classic Game Postmortem: Pinball Construction Set". Game Developers Conference. San Francisco, CA, 2013.

——— (2016). *Message to the author*. E-mail. Feb. 2016.

Burghardt, Gordon M. (2005). *The genesis of animal play: Testing the limits*. The MIT Press, 2005.

——— (2010). "The comparative reach of play and brain". In: *Am. J. Play* 2 (2010), pp. 338–355.

Burks, Arthur Walter, ed. (1970). *Essays On Cellular Automata*. University of Illinois Press, 1970.

Bush, Vannevar (1934). “Structural analysis by electric circuit analogies”. In: *Journal of the Franklin Institute* 217.3 (Mar. 1934), pp. 289–329.

Caillois, Roger (1957). “Unity of Play: Diversity of Games”. In: *Diogenes* 5.19 (Jan. 9, 1957), pp. 92–121.

——— (1961). *Man, Play, and Games*. New York: Free Press of Glencoe, 1961. 208 pp.

Cambron, Melanie (2002). *A Chat with Will Wright*. June 2002. URL: https://web.archive.org/web/20020603144735/http://www.gignews.com/goddess_wright.htm.

Campbell-Kelly, Martin and William Aspray (2009). *Computer: A History of the Information Machine, Second Edition*. Westview Press, Apr. 27, 2009. 362 pp.

Chandler, David L. (2003). “In the beginning was the Rule”. In: *New Scientist* (June 21, 2003).

Charney, Jule G. and N. A. Phillips (1953). “Numerical integration of the quasi-geostrophic equations for barotropic and simple baroclinic flows”. In: *Journal of Meteorology* 10.2 (1953), pp. 71–99.

Christiansen, Godtfred K. (1961). “Toy building brick”. US Patent #3,005,282. Asignee: Interlego Ag. Oct. 24, 1961.

Chudacoff, Howard P. (2008). *Children at Play: An American History*. New York: NYU Press, Sept. 1, 2008. 269 pp.

Church, Doug (1999). “Formal abstract design tools”. In: *Game Developer* 6.8 (Aug. 1999), pp. 44–50.

Chödrön, Pema (2001). *The Places That Scare You*. Shambhala Publications, 2001. 203 pp.

Clancy, John J. and Mark S. Fineberg (1965). “Digital Simulation Languages: A Critique and a Guide”. In: *Proceedings of the November 30–December 1, 1965, Fall Joint Computer Conference, Part I*. AFIPS ’65 (Fall, part I). New York, NY, USA: ACM, 1965, pp. 23–36.

Clark, Andy (1998). *Being There: Putting Brain, Body, and World Together Again*. A Bradford Book, Jan. 9, 1998. 291 pp.

Clark, H. H. and S. E Brennan (1991). “Grounding in communication”. In: *Perspectives on socially shared cognition* 13.1991 (1991), pp. 127–149.

Codd, Edgar Frank (1965). "Propagation, Computation, and Construction in Two-Dimensional Cellular Spaces". Ph.D. thesis. Ann Arbor, Michigan: University of Michigan, 1965. 159 pp.

——— (1968). *Cellular automata*. Academic Press, 1968. 144 pp.

Concept Los Angeles: The Concept For The Los Angeles General Plan (1970). Los Angeles, California: Department Of City Planning, Jan. 1970.

Conrad, Brian (2016). *Interview by Chaim Gingold*. Mar. 21, 2016.

Costikyan, Greg (1994). "I Have No Words And I Must Design". In: *Interactive Fantasy* 2 (1994).

Crawford, Chris (1984). *The Art of Computer Game Design*. Osborne/McGraw-Hill, 1984. 113 pp.

——— (2003). *The Art of Interactive Design: A Euphonious and Illuminating Guide to Building Successful Software*. No Starch Press, 2003. 410 pp.

Cross, Gary S. (1990). *A Social History of Leisure Since 1600*. State College, PA: Venture Pub, 1990. 297 pp.

Csikszentmihalyi, Mihaly (1975). "Play and intrinsic rewards." In: *Journal of humanistic psychology* (1975).

——— (1991). *Flow: The psychology of optimal experience*. HarperPerennial New York, 1991.

Curtin, Claire (2015). *Interview by Chaim Gingold*. Aug. 2, 2015.

Davis, Mike (1990). *City of Quartz: Excavating the Future in Los Angeles*. Vintage Books, 1990. 486 pp.

De Coninck-Smith, Ning (1999). *Natural Play in Natural Surroundings: Urban Childhood and Playground Planning in Denmark, c. 1930 — 1950*. Odense University, 1999.

Deane, Andrew R. (2015). *Chapter 36: Looking beyond: Borrowed Scenery*. 2015. URL: http://www.japanesegardensonline.com/Looking-beyond-Borrowed-Scenery/ (visited on 11/04/2015).

Donald, Merlin (2001). *A mind so rare: The evolution of human consciousness*. WW Norton & Company, 2001.

Donald, Patty (2012). *Interview by Chaim Gingold*. Apr. 30, 2012.

Donald, Patty (2016). *Interview by Chaim Gingold*. Mar. 1, 2016.

Donald, Patty and Susan Schwartz (2015). *Message to the author*. E-mail. Dec. 10, 2015.

Donovan, Tristan (2011). “The Replay Interviews: Will Wright”. In: *Gamasutra* (May 23, 2011).

Dreyfus, Hubert L. (1972). *What computers can’t do: a critique of artificial reason*. Harper & Row, 1972. 312 pp.

Duke, Richard D. (1974). *Gaming: the future’s language*. New York: Sage Publications; distributed by Halsted Press, 1974. 248 pp.

——— (1995). “Gaming: An emergent discipline”. In: *Simulation & Gaming* 26.4 (1995), pp. 426–439.

——— (2011). “Origin and Evolution of Policy Simulation: A Personal Journey”. In: *Simulation & Gaming* 42.3 (Jan. 6, 2011), pp. 342–358.

Dunnigan, Jim (1970). *PanzerBlitz*. Board game. Avalon Hill, 1970.

Edwards, Paul N. (1997). *The Closed World: Computers and the Politics of Discourse in Cold War America*. The MIT Press, Aug. 1, 1997. 462 pp.

——— (2010). *A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming*. MIT Press, Apr. 30, 2010. 547 pp.

Eisenhart, Mary (1996). “Building SimEmpires: Good Times At Maxis”. In: *MicroTimes* (1996).

“Electronic Arts Will Buy Maxis in Swap” (1997). In: *The New York Times. Business* (June 5, 1997).

Eno, Brian (1996). “Generative Music”. In: *In Motion Magazine* (June 7, 1996). A talk delivered in San Francisco, June 8, 1996.

Eno, Brian and Will Wright (2006). “Playing with Time”. Seminars About Long-term Thinking. June 26, 2006.

Ensmenger, Nathan L. (2010). *The Computer Boys Take Over*. Cambridge, Mass: MIT Press, 2010.

Fagen, Robert (1981). *Animal play behavior*. Oxford University Press, Jan. 29, 1981. 708 pp.

Fenton, Jay and Kent Beck (1989). “Playground: an object-oriented simulation system with agent rules for children of all ages”. In: *ACM SIGPLAN Notices* 24.10 (1989), pp. 123–137.

Foley, Mick (2015). *Interview by Chaim Gingold*. July 31, 2015.

Forrester, Jay Wright (1956). “Dynamic models of economic systems and industrial organizations: Nov. 5, 1956 Note to the Faculty Research Seminar.” In: *System Dynamics Review* 19.4 (1956–2003), pp. 329–345.

——— (1958). “Industrial Dynamics: A Major Breakthrough for Decision Makers”. In: *Harvard Business Review* 36.4 (1958), pp. 37–66.

——— (1960). “The Impact of Feedback Control Concepts on the Management Sciences”. In: *Collected Papers of Jay W. Forrester*. Wright-Allen Press, 1960–1975, pp. 45–60.

——— (1961). *Industrial Dynamics*. MIT press Cambridge, MA, 1961.

——— (1968). “Industrial dynamics—after the first decade”. In: *Management Science* 14.7 (1968), pp. 398–415.

——— (1969). *Urban Dynamics*. Cambridge, Mass.: M.I.T. Press, 1969.

——— (1971). *World Dynamics*. 2nd. Cambridge, Mass.: Wright-Allen Press, 1971–1973.

——— (1975). *Collected Papers of Jay W. Forrester*. Wright-Allen Press, 1975. 284 pp.

——— (1989). “The beginning of system dynamics”. System Dynamics Society. Stuttgart, Germany, July 13, 1989.

——— (1992). “From the ranch to system dynamics: an autobiography”. In: *Management Laureates: A Collection of Autobiographical Essays*. Ed. by Arthur G. Bedeian. Vol. 1. 1992, pp. 337–370.

——— (1995). “The beginning of system dynamics”. In: *The McKinsey Quarterly* 4 (1995).

——— (2016). *Message to the author*. E-mail. Jan. 4, 2016.

Forrester, Jay Wright and Robert R. Everett (1990). “The Whirlwind computer project”. In: *IEEE Transactions on Aerospace and Electronic Systems* 26.5 (1990), pp. 903–910.

Foundation series (2015). In: *Wikipedia, the free encyclopedia*. Page Version ID: 694220905. Dec. 7, 2015.

Frasca, Gonzalo (2001a). “Rethinking agency and immersion: video games as a means of consciousness-raising”. In: *Digital Creativity* 12.3 (2001), pp. 167–174.

——— (2001b). “Videogames of the Oppressed”. In: *First person: New media as story, performance, and game* (2001), pp. 85–94.

——— (2001c). “Videogames of the Oppressed”. M.S. thesis. Atlanta, GA: Georgia Institute of Technology, 2001.

——— (2007). “Play the message: Play, game and videogame rhetoric”. Ph.D. thesis. Copenhagen, Denmark: IT University, 2007.

Fredkin, Edward (1992). “A new cosmogony”. In: *Proceedings of the Physics of Computation Workshop*. Physics of Computation Workshop, October 2-4, 1992. IEEE, 1992.

——— (2003). “An introduction to digital philosophy”. In: *International Journal of Theoretical Physics* 42.2 (2003), pp. 189–247.

Freitas, Robert A. and William P. Gilbreath, eds. (1982). *Advanced Automation for Space Missions*. NASA Conference Publication 2255. 1982.

Friedberg, M. Paul (1970). *Play and interplay: a manifesto for new design in urban recreational environment*. In collab. with Ellen Perry Berkeley. Macmillan, 1970. 200 pp.

Friedman, Ted (1995). “Making sense of software: Computer games and interactive textuality”. In: *Cybersociety; Computer-Mediated Communication and Community. Thousand Oaks, Calif.: Sage Publications* (1995).

——— (1999). “The Semiotics of SimCity”. In: *First Monday* 4.4 (Apr. 5, 1999).

Frost, Joe (2010). *A History of Children’s Play and Play Environments: Toward a Contemporary Child-Saving Movement*. Routledge, Apr. 2, 2010. 343 pp.

——— (2012). “Evolution of American Playgrounds”. In: *Scholarpedia* 7.12 (2012), p. 30423.

Gaber, John (2007). “Simulating Planning SimCity as a Pedagogical Tool”. In: *Journal of Planning Education and Research* 27.2 (2007), pp. 113–121.

Ganguly, Niloy, Biplab K. Sikdar, Andreas Deutsch, Geoffrey Canright, and P. Pal Chaudhuri (2003). “A survey on cellular automata”. 2003.

Gardner, Martin (1970). “Mathematical games: The fantastic combinations of John Conway’s new solitaire game “life””. In: *Scientific American* 223.4 (1970), pp. 120–123.

——— (1971). “Mathematical games: Cellular automata, self-reproduction, Garden of Eden and Game life”. In: *Scientific American* 224.2 (1971), p. 112.

Garvey, Catherine (1974). “Some properties of social play”. In: *Merrill-Palmer Quarterly: Journal of Developmental Psychology* (1974).

——— (1977). *Play*. Enl. ed. The Developing child. Cambridge, Mass: Harvard University Press, 1977–1990. 184 pp.

Geertz, Clifford (1972). “Deep play: Notes on the Balinese cockfight”. In: *Daedalus* 101.1 (1972), pp. 1–37.

Gelber, S. M (1999). *Hobbies: Leisure and the culture of work in America*. Columbia Univ Pr, 1999.

Gelder, Lawrence Van (1997). “Charles Dederich, 83, Synanon Founder, Dies”. In: *The New York Times. U.S.* (Mar. 4, 1997).

Gingold, Chaim (2003). “Miniature Gardens & Magic Crayons: Games, Spaces, & Worlds”. M.S. thesis. Georgia Institute of Technology, 2003.

——— (2009). “The Human Play Machine”. Game Developers Conference. San Francisco, CA, 2009.

——— (2012). “Field Notes from Berkeley Adventure Playground 4/29/12 and 5/14/12.” Field Notes. Berkeley, California, 2012.

——— (2013). “Playing with Mr. Dangerous: An ethnography of the Berkeley adventure playground”. Mar. 4, 2013.

——— (2015). “Field Notes from Design Based Learning Institute”. Field Notes. Los Angeles and Pasadena, California, July 27, 2015.

Graetz, J. M (1981). “The origin of Spacewar”. In: *Creative Computing* 56 (Aug. 1981), p. 67.

Greco, Susan (1994). “Looking for Mr. Right”. In: *Inc.* (May 1, 1994).

Hague, James (1997). “Halcyon days: Interviews with classic computer and video game programmers”. In: *Self-published digitally in* (1997).

Harnad, Stevan (1990). “The symbol grounding problem”. In: *Physica D: Nonlinear Phenomena* 42.1 (1990), pp. 335–346.

Hart, Vi and Nicky Case (2014). *Parable of the Polygons*. 2014. URL: http://ncase.me/polygons/.

Hayles, N. Katherine (1999). *How we became posthuman: virtual bodies in cybernetics, literature, and informatics*. University of Chicago Press, 1999. 368 pp.

Hickman, Craig (2015). *Kid Pix - The Early Years*. 2015. URL: http://red-green-blue.com/kid-pix-the-early-years/ (visited on 09/16/2015).

Higuchi, Tadahiko (1983). *The Visual and Spatial Structure of Landscapes*. MIT Press, Jan. 1, 1983. 218 pp.

Hiltzik, Michael A. (1999). *Dealers of Lightning: Xerox PARC and the Dawn of the Computer Age*. HarperCollins, Mar. 3, 1999. 480 pp.

History of Microsoft Flight Simulator (2016). In: *Wikipedia, the free encyclopedia*. Page Version ID: 702419824. Jan. 30, 2016.

Hodges, Dan (2015). *Why do Pinky and Inky have different behaviors when Pac-Man is facing up?* Aug. 14, 2015. URL: http://www.donhodges.com/pacman_pinky_explanation.htm (visited on 03/28/2016).

Hogan, P. (1982). *The Nuts and Bolts of Playground Construction*. Leisure Press, Jan. 1, 1982. 348 pp.

Holladay, J. and S. Ulam (1960). “On some combinatorial problems in pattern of growth”. In: *Notices of the American Mathematical Society* 7 (1960), p. 234.

Holland, Dorothy C., William Jr. Lachicotte, Debra Skinner, and Carole Cain (1998). *Identity and Agency in Cultural Worlds*. Harvard University Press, 1998. 376 pp.

Holland, John H. (1959). “A universal computer capable of executing an arbitrary number of sub-programs simultaneously”. In: *Managing Requirements Knowledge, International Workshop on*. IEEE Computer Society, 1959, pp. 108–108.

——— (1960). “Iterative circuit computers”. In: Western joint IRE-AIEE-ACM computer conference. ACM, May 3–5, 1960, pp. 259–265.

——— (1962). “Outline for a Logical Theory of Adaptive Systems”. In: *J. ACM* 9.3 (July 1962), pp. 297–314.

——— (1968). *Hierarchical descriptions, universal spaces and adaptive systems*. Ann Arbor, MI, USA: University of Michigan, 1968.

——— (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, 1992. 232 pp.

Holthaus, Eric and Brad Allenby (2015). “Of Daisies and the Robot Apocalypse”. In: *Slate* (Mar. 30, 2015).

Hopkins, Don (2008). *Soul of The Sims, by Will Wright*. Program source code fragment, originally from Jan 28, 1997. Feb. 10, 2008. URL: http://www.donhopkins.com/drupal/node/148.

——— (2013). *Interview by Chaim Gingold*. Sept. 17, 2013.

Horgan, John (2016). *Claude Shannon: Tinkerer, Prankster, and Father of Information Theory*. Apr. 27, 2016. URL: http://spectrum.ieee.org/computing/software/claude-shannon-tinkerer-prankster-and-father-of-information-theory (visited on 05/16/2016).

Huizinga, Johan (1955). *Homo Ludens; a Study of the Play-Element in Culture*. 1st Beacon paperback ed. Beacon paperbacks 15. Boston: Beacon Press, 1955. 220 pp.

Hunicke, Robin, Marc LeBlanc, and Robert Zubek (2004). “MDA: A formal approach to game design and game research”. In: *Proceedings of the AAAI Workshop on Challenges in Game AI*. 2004.

Hutchins, Edwin (1995). *Cognition in the Wild*. MIT Press, 1995. 413 pp.

Hutchins, Edwin, J. D. Hollan, and D. A. Norman (1985). “Direct manipulation interfaces”. In: *Human-computer interaction* 1.4 (1985), pp. 311–338.

Hyde, Lewis (1998). *Trickster Makes this World: Mischief, Myth, and Art*. Farrar, Straus and Giroux, Jan. 1, 1998. 440 pp.

Isenberg, Barbara (2009). *Conversations with Frank Gehry*. First Edition edition. New York: Knopf, Apr. 21, 2009. 320 pp.

Ito, Mizuko (2009). *Engineering Play: A Cultural History of Children's Software*. Mass., 2009. 234 pp.

Iwata, Satoru (2010). *Iwata Asks: Game & Watch*. Nintendo, Apr. 2010.

Jacobs, Jane (1961). *The Death and Life of Great American Cities*. Vintage Books, 1961. 466 pp.

Joffe, Bruce (2015). *Interview by Chaim Gingold*. June 9, 2015.

——— (2016). *Message to the author*. E-mail. Feb. 2016.

John F. Collins, former mayor and MIT professor, dies at 76 (1995). Nov. 29, 1995. URL: http://web.mit.edu/newsoffice/1995/collins-1129.html (visited on 07/17/2013).

John von Neumann (2016). In: *Wikipedia, the free encyclopedia*. Page Version ID: 709133193. Mar. 9, 2016.

Johnson, Ellen H. (1971). *Claes Oldenburg*. Penguin New Art 4. 1971. 64 pp.

Johnson, Mark (2007). *The Meaning of the Body: Aesthetics of Human Understanding*. University of Chicago Press, Aug. 1, 2007. 338 pp.

Johnson, Steven (2001). *Emergence: the connected lives of ants, brains, cities, and software*. Scribner, Aug. 28, 2001. 302 pp.

Johnston, Ollie and Frank Thomas (1995). *The Illusion of Life: Disney Animation*. Rev Sub edition. New York: Disney Editions, Oct. 5, 1995. 576 pp.

Jones, Gerard (2003). *Killing Monsters: Why Children Need Fantasy, Super Heroes, and Make-Believe Violence*. Basic Books, May 2003. 272 pp.

Juba, Joe (2012). "Reflecting On Maxis: A Will Wright Interview". In: *Game Informer* (Nov. 1, 2012).

——— (2015). "25 Years of Maxis – Living the Simulated Dream". In: *Game Informer* 236 (May 2015).

Juul, Jesper (2003). "The game, the player, the world: looking for a heart of gameness." In: *DIGRA Conf.* 2003.

——— (2005). *Half-Real: Video Games between Real Rules and Fictional Worlds*. The MIT Press, Nov. 4, 2005. 248 pp.

Kare, Susan (2001). *Interview with Susan Kare*. In collab. with Alex Soojung-Kim Pang. Feb. 20, 2001. URL: http://web.stanford.edu/dept/SUL/sites/mac/primary/interviews/kare/trans.html (visited on 04/05/2016).

Kay, Alan (1972). “A personal computer for children of all ages”. In: *Proceedings of the ACM annual conference-Volume 1*. 1972, p. 1.

——— (1984). “Computer Software”. In: *Scientific American* 251 (Sept. 1984), pp. 3–9.

——— (1987). “Doing with Images Makes Symbols”. Higher Education Marketing Group, Apple Computer, Inc. 1987.

——— (1991). “Computers, networks and education”. In: *Scientific American* 265.3 (1991), pp. 138–148.

——— (2007). *Alan Kay’s ideas about SimCity for OLPC*. E-mail from Alan Kay to Don Hopkins. Nov. 9, 2007. URL: http://www.donhopkins.com/drupal/node/134.

——— (2012). “Extracting Energy from the Turing Tarpit”. ACM A.M. Turing Centenary Celebration. San Francisco, CA, June 16, 2012.

Kay, Alan and Adele Goldberg (1977). “Personal dynamic media”. In: *Computer* 10.3 (1977), pp. 31–41.

Kelly, Kevin (1994a). *Out of Control: The Rise of Neo-Biological Civilization*. Addison-Wesley, Jan. 1994. 534 pp.

——— (1994b). “Will Wright: The Mayor of SimCity”. In: *Wired* (Jan. 1994).

Kemeny, John G. (1955). “Man viewed as a machine”. In: *Scientific American* (Apr. 1955), pp. 58–67.

Kennedy, Ruth A. (1997). “FORM S-4: REGISTRATION STATEMENT UNDER THE SECURITIES ACT OF 1933”. Redwood City, California, June 25, 1997.

Ketchell, Robert (2013). *Borrowed Landscapes - Shakkei*. June 12, 2013. URL: http://robertketchell.blogspot.com/2013/06/borrowed-landscapes-shakkei.html (visited on 11/04/2015).

Kidder, Tracy (1982). *The soul of a new machine*. Avon, 1982. 308 pp.

Kinoshita, Isami and Helen Woolley (2015). “Children’s Play Environment after a Disaster: The Great East Japan Earthquake”. In: *Children* 2.1 (Jan. 28, 2015), pp. 39–62.

Kirschenbaum, Matthew G. (2004). “Extreme inscription: Towards a grammatology of the hard drive”. In: *TEXT Technology* 2 (Nov. 2, 2004), pp. 91–.

Kirsh, David (1995). “The intelligent use of space”. In: *Artificial intelligence* 73.1 (1995), pp. 31–68.

Kirsh, David and Paul Maglio (1992). “Some Epistemic Benefits of Action-Tetris, a Case Study”. In: *Proceedings of the 14th Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Morgan Kaufman, 1992.

——— (1994). “On distinguishing epistemic from pragmatic action”. In: *Cognitive science* 18.4 (1994), pp. 513–549.

Koebler, Jason (2015). *The Real Mayors of ’SimCity’*. May 8, 2015. URL: http://motherboard.vice.com/read/the-real-mayors-of-simcity (visited on 05/09/2015).

Kohler, Chris (2011). *Trailblazing DIY Pinball Game Snags Pioneer Award for Bill Budge | Game|Life*. Jan. 21, 2011. URL: http://www.wired.com/2011/01/bill-budge-pioneer/ (visited on 07/14/2014).

Koster, Raph (2004). *Theory of Fun for Game Design*. Paraglyph, Incorporated, Nov. 2004. 256 pp.

Kozlovsky, Roy (2007). “Adventure playgrounds and postwar reconstruction”. In: *Designing Modern Childhoods: History, Space, and the Material Culture of Children; An International Reader.* Ed. by Marta Gutman and Ning de Coninck-Smith. 2007.

Kuhfeld, Albert W. (1971). “Spacewar”. In: *Analog Science Fiction/Science Fact* (July 1971), pp. 67–79.

Kushner, David (2005). *Life Is a Game*. Apr. 1, 2005. URL: http://www.technologyreview.com/article/403898/life-is-a-game/ (visited on 04/30/2015).

Lammers, Susan M. (1986). *Programmers at work*. Microsoft Press, 1986. 398 pp.

Lane, David C. (2007). “The power of the bond between cause and effect: Jay Wright Forrester and the field of system dynamics”. In: *System Dynamics Review* 23.2-3 (2007), pp. 95–118.

Lane, David C. and John D. Sterman (2011). “Jay Wright Forrester”. In: *Profiles in Operations Research: Pioneers and Innovators*. Ed. by Arjang A. Assad and Saul I. Gass. Springer, 2011, pp. 363–386.

Langton, Christopher G. (1984). “Self-reproduction in cellular automata”. In: *Physica D: Nonlinear Phenomena* 10.1 (1984), pp. 135–144.

——— (1986). “Studying artificial life with cellular automata”. In: *Physica D: Nonlinear Phenomena* 22.1 (1986), pp. 120–149.

——— (1990). “Computation at the edge of chaos: phase transitions and emergent computation”. In: *Physica D: Nonlinear Phenomena* 42.1 (1990), pp. 12–37.

Lantz, Frank (2011). “Life and Death and Middle Pair: Go, Poker and the Sublime”. Game Developers Conference. 2011.

——— (2014a). “Go in the New World”. US Go Congress. New York, N.Y., Oct. 14, 2014.

——— (2014b). “Hearts and Minds”. Game Developers Conference. San Francisco, CA, 2014.

Lantz, Frank and Michaël Samyn (2009). *Tale of Tales » Interview with Frank Lantz*. 2009. URL: http://tale-of-tales.com/blog/interviews/interview-with-frank-lantz/ (visited on 11/20/2014).

Latané, Claire (2013). “The Irvine Adventure Playground: Grassroots Design from the Mud Up”. IndieCade. Culver City, CA, 2013.

Latour, Bruno (1986). “Visualization and cognition”. In: *Knowledge and society* 6 (1986), pp. 1–40.

——— (1987). *Science in Action: How to Follow Scientists and Engineers Through Society*. Open University Press, 1987. 274 pp.

——— (1996). *Aramis, or the Love of Technology*. Harvard University Press, Apr. 1, 1996. 336 pp.

Laurel, Brenda (1986). “Interface as mimesis”. In: *D. Norman & S. Draper, User Centered System Design. Lawrence Erlbaum, Hillsdale NJ* (1986), pp. 67–85.

——— (1993). *Computers as Theatre*. Reading, Mass: Addison-Wesley Pub. Co, 1993. 227 pp.

Lauwaert, Maaike (2007). “Challenge everything? Construction play in Will Wright’s SIMCITY”. In: *Games and Culture* 2.3 (2007), pp. 194–212.

——— (2008). “Playing outside the box – on LEGO toys and the changing world of construction play”. In: *History and Technology* 24.3 (Sept. 1, 2008), pp. 221–237.

Lazzaro, Nicole (2004). “Why We Play Games: 4 Keys to More Emotion”. Game Developers Conference. 2004.

LeBlanc, Marc (2004). “Mechanics, Dynamics, Aesthetics: A Formal Approach to Game Design”. Apr. 2004.

Lem, Stanisław (1974). *The Cyberiad: Fables for the Cybernetic Age*. Seabury Press, Mar. 1974. 310 pp.

Leonard, R. J. (1995). “From parlor games to social science: von Neumann, Morgenstern, and the creation of game theory 1928-1944”. In: *Journal of Economic Literature* (1995), pp. 730–761. JSTOR: 10.2307/2729025.

Levy, Steven (1984). *Hackers: Heroes of the Computer Revolution*. Anchor Books, 1984. 488 pp.

——— (1992). *Artificial life: the quest for a new creation*. Random House Inc., 1992.

Lew, Julie (1989). “Making City Planning a Game”. In: *New York Times* (June 15, 1989).

Lewin, Kurt (1951). *Field Theory in Social Science: Selected Theoretical Papers*. Harper, 1951. 376 pp.

Librande, Stone (2010). “One-Page Designs”. Game Developers Conference. 2010.

Light, Jennifer S. (2002). “Urban security from warfare to welfare”. In: *International Journal of Urban and Regional Research* 26.3 (2002), pp. 607–613.

——— (2003). *From Warfare to Welfare: Defense Intellectuals and Urban Problems in Cold War America*. JHU Press, Nov. 19, 2003. 312 pp.

Liversidge, Anthony (1993). “Profile of Claude Shannon”. In: *Claude Elwood Shannon: Collected Papers*. Ed. by Neil JA Sloane and Aaron D. Wyner. John Wiley & Sons, Inc., 1993.

Loftus, G. R. and E. F. Loftus (1983). *Mind at Play: The Psychology of Video Games*. Basic Books, Inc., 1983.

Loguidice, Bill and Matt Barton (2009). *Vintage Games: An Insider Look at the History of Grand Theft Auto, Super Mario, and the Most Influential Games of All Time*. Taylor & Francis US, 2009. 410 pp.

Maglio, Paul P. and David Kirsh (1996). "Epistemic action increases with skill". In: *Proceedings of the eighteenth annual conference of the cognitive science society*. Vol. 16. 1996, pp. 391–396.

Maglio, Paul P., Teenie Matlock, Dorth Raphaely, Brian Chernicky, and David Kirsh (1999). "Interactive skill in Scrabble". In: *Proceedings of the twenty-first annual conference of the cognitive science society*. Routledge, 1999, pp. 326–330.

Maher, Jimmy (2013). *The Pinball Wizard*. Feb. 1, 2013. URL: http://www.filfre.net/2013/02/the-pinball-wizard/ (visited on 07/14/2014).

Mahoney, Michael S. (1988). "The History of Computing in the History of Technology". In: *Annals of the History of Computing* 10.2 (1988), pp. 113–125.

——— (2005). "The histories of computing (s)". In: *Interdisciplinary Science Reviews* 30.2 (2005), pp. 119–135.

Maines, Rachel P. (2009). *Hedonizing Technologies: Paths to Pleasure in Hobbies and Leisure*. JHU Press, June 9, 2009. 225 pp.

Malaby, T. M (2007). "Beyond Play A New Approach to Games". In: *Games and Culture* 2.2 (2007), pp. 95–113.

Margolus, Norman (1984). "Physics-like models of computation". In: *Physica D: Nonlinear Phenomena* 10.1–2 (Jan. 1984), pp. 81–95.

Marino, Mark C. (2006). "Critical code studies". In: *Electronic book review* 4 (2006).

Masaki, Lyle (2007). *Whatever Happened To ... The man behind SimCopter's gay "Easter Egg"?* June 6, 2007. URL: http://www.newnownext.com/whatever-happened-to-the-man-behind-simcopters-gay-easter-egg/06/2007/ (visited on 04/13/2016).

Mateas, Michael (2002). "Interactive drama, art and artificial intelligence". Ph.D. thesis. Pittsburgh, PA: Carnegie Mellon University, 2002.

——— (2003). "Expressive AI: Games and Artificial Intelligence." In: *DIGRA Conf.* 2003.

Mateas, Michael and Nick Montfort (2005). "A box, darkly: Obfuscation, weird languages, and code aesthetics". In: *Proceedings of the 6th Digital Arts and Culture Conference, IT University of Copenhagen*. 2005, pp. 144–153.

Mateas, Michael and Noah Wardrip-Fruin (2009). "Defining operational logics". In: *Proceedings of the Digital Games Research Association* (2009).

"Maxis Inc. ready to go public" (1995). In: *SFGate* (Mar. 30, 1995).

McCall, Spencer (2012). *The Institute*. Documentary film, 91 minutes. 2012.

McCloud, Scott (1993). *Understanding Comics*. HarperCollins, 1993. 236 pp.

McCulloch, Warren S. and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

McDermott, Drew (1976). "Artificial intelligence meets natural stupidity". In: *ACM SIGART Bulletin* 57 (1976), pp. 4–9.

McGonigal, Jane (2012). *How Might Video Games Be Good for Us?* Oct. 15, 2012. URL: https://www.bigquestionsonline.com/content/how-might-video-games-be-good-us.

McKenzie, John A. (1974). *TX-O Computer History*. RLE Technical Report No. 627. Cambridge, Mass: The Research Laboratory of Electronics at MIT, Oct. 1, 1974.

McLuhan, Marshall (1964). *Understanding Media*. 1964. 368 pp.

McManis, Sam (2013). "Discoveries: At a park in Berkeley, kids wield hammers and saws with impunity ... while parents watch." In: *The Sacramento Bee* (Sept. 22, 2013).

Meadows, Dana. *Dear Folks: January 18, 1987*. URL: http://www.donellameadows.org/archives/dear-folks-january-18-1987/ (visited on 05/16/2015).

Meadows, Donella H., Dennis L. Meadows, Jørgen Randers, and William W. Behrens III (1972). *The Limits to Growth: A Report for the Club of Rome's Project on the Predicament of Mankind*. Universe Books, 1972. 212 pp.

Mechner, Jordan (2011). "Classic Game Postmortem: Prince of Persia". Game Developers Conference. San Francisco, CA, 2011.

Meyers, Susan (1985). “Computer Construction Worker: Bill Budge Wants to Help you Build the Ultimate Computer Game”. In: *Enter* 2.5 (Mar. 1985).

MIT History | Department of Urban Studies and Planning (2004). Oct. 2004. URL: http://libraries.mit.edu/mithistory/research/schools-and-departments/school-of-architecture-and-planning/department-of-urban-studies-and-planning/ (visited on 07/23/2013).

Mitchell, Melanie (1996). “Computation in cellular automata: A selected review”. In: *Nonstandard Computation* (1996), pp. 95–140.

Moncreiff, Bruse (1965). “The ”Sumerian Game:” Teaching Economics with a Computerized P.I. Program”. In: *Programmed Instruction* 4.5 (1965), pp. 10–11.

Montfort, Nick and Ian Bogost (2009). *Racing the Beam: The Atari Video Computer System*. MIT Press, Mar. 31, 2009. 193 pp.

Montfort, Nick, Patsy Baudoin, John Bell, Ian Bogost, Jeremy Douglass, Mark C. Marino, Michael Mateas, Reas Casey, Mark Sample, and Noah Vawter (2012). *10 Print CHR$(205.5+RND(1)): Goto 10*. MIT Press, Nov. 16, 2012. 328 pp.

Moore, Edward F. (1962). “Machine models of self-reproduction”. In: *Proc. Symp. Appl. Math*. Vol. 14. 1962, pp. 17–33.

Morris, Desmond (1966). *The Naked Ape: A Zoologist’s Study of the Human Animal*. Dell, 1966–1999. 252 pp.

Moss, Richard (2015a). *Episode 2: Mercy*. Podcast. Dec. 24, 2015.

——— (2015b). “From SimCity to, well, SimCity: The history of city-building games”. In: *Ars Technica* (Oct. 11, 2015).

Murray, Janet H. (1997). *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. First Edition. Free Press, July 16, 1997. 336 pp.

——— (2006). “Toward a cultural theory of gaming: Digital games and the co-evolution of media, mind, and culture”. In: *Popular Communication* 4.3 (2006), pp. 185–202.

——— (2011). *Inventing the Medium: Principles of Interaction Design As a Cultural Practice*. MIT Press, Nov. 23, 2011. 483 pp.

Murray, Laura M. (1992). “E.O. Wilson Inspires SimAnt”. In: *The Harvard Crimson* (Feb. 29, 1992).

Murrow, Edward R. (1951). *Jay Forrester and Whirlwind*. Television program. Dec. 16, 1951.

Nelson, Doreen (1974a). “The City Building Educational Program: A Decision-Making Approach to Education”. In: *Alternative Learning Environments*. Ed. by Gary J. Coates. In collab. with Teri Fox, Miller Ty, Benjamin Dan, Spiess Greg, and Smith Alexis. Dowden, Hutchinson & Ross, 1974.

——— (1981). *Classroom City*. Film, 16 minutes. 1981.

——— (1982). *City Building Education: A Way to Learn*. In collab. with Kirsten Grimstad. Center for City Building Educational Programs, Jan. 1, 1982. 84 pp.

——— (1984). *Transformations: Process and theory; a curriculum guide to creative development*. Center for City Building Educational Programs, 1984. 218 pp.

——— (2015). *Interview by Chaim Gingold*. July 13, 2015.

Nelson, Theodor H. (1974b). *Computer Lib: And Dream Machines*. Theodor Nelson, 1974. 127 pp.

——— (1977). *The home computer revolution*. Nelson, 1977. 232 pp.

Nicholson, Simon (1971). “How not to cheat children: the theory of loose parts.” In: *Landscape Architecture Quarterly* 62.1 (1971), pp. 30–34.

Norman, Donald A. (1988). *The Design of Everyday Things*. MIT Press, 1988. 257 pp.

Ohshima, Yoshiki, Dan Amelang, Ted Kaehler, Bert Freudenberg, Aran Lunzer, Alan Kay, Ian Piumarta, Takashi Yamamiya, Alan Borning, Hesam Samimi, Bret Victor, and Kim Rose (2012). *STEPS Toward the Reinvention of Programming, 2012 Final Report Submitted to the National Science Foundation (NSF)*. VPRI Technical Report TR-2012-001. Viewpoints Research Institute, Oct. 2012.

Olney, David (1976). *City Building Education: A Way to Learn*. In collab. with Doreen Nelson and Charles Eames. Film, 11 minutes. Eames Office, 1976.

Onishi, Kristine H., Renée Baillargeon, and Alan M. Leslie (2007). “15-month-old infants detect violations in pretend scenarios”. In: *Acta Psychologica* 124.1 (2007), p. 106.

Panksepp, Jaak (2004). *Affective Neuroscience: The Foundations of Human and Animal Emotions*. 1st ed. Oxford University Press, USA, Sept. 30, 2004. 480 pp.

PanzerBlitz (2015). In: *Wikipedia, the free encyclopedia*. Page Version ID: 690089405. Nov. 11, 2015.

Papert, Seymour (1980). *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books, 1980. 230 pp.

——— (1987). "Computer criticism vs. technocentric thinking". In: *Educational Researcher* (1987), pp. 22–30. JSTOR: 1174251.

Parlett, David Sidney (1999). *The Oxford history of board games*. Oxford University Press, Feb. 25, 1999. 410 pp.

Pearce, Celia (2001). "Sims, BattleBots, Cellular Automata God and Go. A Conversation with Will Wright". In: *Game Studies* 2.1 (Sept. 5, 2001).

Penrose, Lionel S. (1959). "Self-reproducing machines". In: *Scientific American* 200.6 (1959), pp. 105–114.

Penrose, Lionel S. and Roger Penrose (1957). "A self-reproducing analogue". In: (1957).

Perkins, Edwin C. Jr. (1993). "Bedlam in SimCity". In: *UnixWorld* (Apr. 1993).

Peterson, Jon (2012). *Playing at the World*. Unreason Press, Aug. 1, 2012. 720 pp.

Phipps, Keith (2005). *Will Wright*. Feb. 2, 2005. URL: http://www.avclub.com/article/will-wright-13911 (visited on 04/29/2015).

Pittman, Jamey (2011). *The Pac-Man Dossier*. June 16, 2011. URL: http://home.comcast.net/~jpittman2/pacman/pacmandossier.html.

Pitts, Russ (2012). *Will Wright cites Pinball Construction Set as major inspiration*. Mar. 8, 2012. URL: http://www.polygon.com/gaming/2012/3/8/2855177/will-wright-pinball-construction-set-sims-sim-city (visited on 05/28/2015).

Poock, Julia (2012). *Interview by Chaim Gingold*. May 15, 2012.

Preston, Frank (2003). "Vannevar Bush's network analyzer at the Massachusetts Institute of technology". In: *IEEE Annals of the History of Computing* 25.1 (2003), pp. 75–78.

Raz, Guy (2011). *Cow Clicker Founder: If You Can't Ruin It, Destroy It*. Nov. 18, 2011.

Redmond, Kent C. and Thomas Malcolm Smith (1980). *Project Whirlwind: the history of a pioneer computer*. Digital Press, 1980. 296 pp.

Reiter, Carla (1984). "Life and death on a computer screen". In: *Discover* 5.8 (1984), pp. 81–83.

Resnick, Mitchel (1994). *Turtles, Termites, and Traffic Jams: Explorations in Massively Microworlds*. MIT Press, 1994. 163 pp.

Resnick, Mitchel, Fred Martin, Randy Sargent, and Brian Silverman (1996). "Programmable bricks: Toys to think with". In: *IBM Systems journal* 35.3.4 (1996), pp. 443–452.

Richardson, George P. (1983). "The feedback concept in American social science, with implications for system dynamics". In: International System Dynamics Conference. Chestnut Hill, Massachusetts, July 27–30, 1983.

——— (1991). *Feedback thought in social science and systems theory*. University of Pennsylvania Press, 1991. 394 pp.

——— (2011). "Reflections on the foundations of system dynamics". In: *System Dynamics Review* 27.3 (2011), pp. 219–243.

Roberts, John M., Malcolm J. Arth, and Robert R. Bush (1959). "Games in culture". In: *American anthropologist* 61.4 (1959), pp. 597–605.

Roberts, John M. and Michael L. Forman (1971). "Riddles: Expressive models of interrogation". In: *Ethnology* (1971), pp. 509–533. JSTOR: 3773178.

Roberts, John M. and Brian Sutton-Smith (1962). "Child training and game involvement". In: *Ethnology* 1.2 (1962), pp. 166–185.

Roberts, John M., Brian Sutton-Smith, and A. Kendon (1963). "Strategy in games and folk tales". In: *The Journal of social psychology* 61.2 (1963), pp. 185–199.

Rouse, Richard (2001). *Game Design: Theory and Practice*. Wordware Pub., Jan. 1, 2001. 612 pp.

Ruberg, Bonnie (2006). "Big Reality: A Chat with 'Big Game' Designer Frank Lantz". In: (Aug. 10, 2006).

SAIL Timesharing System <SAI@SAIL.Stanford.EDU> (1991). *life as a computer for a quarter of a century*. E-mail. June 7, 1991.

Salen, Katie and Eric Zimmerman (2003). *Rules of Play: Game Design Fundamentals*. The MIT Press, Sept. 25, 2003. 688 pp.

Sample, Mark (2011). “Criminal Code: The Procedural Logic of Crime in Videogames”. Panel: ”Close Reading the Digital”. MLA convention. Los Angeles, California, Jan. 14, 2011.

Sarkar, Palash (2000). “A Brief History of Cellular Automata”. In: *ACM Comput. Surv.* 32.1 (Mar. 2000), pp. 80–107.

Schechner, Richard (1988). “Playing”. In: *Play & Culture* (1988).

Scheffler, Tom (1989). *Girlfriend Construction Set*. 1989.

Schelling, Thomas C. (1971). “Dynamic models of segregation†”. In: *Journal of Mathematical Sociology* 1.2 (1971), pp. 143–186.

School History. URL: http://www.opencharter.org/node/20 (visited on 09/30/2015).

Schrandt, Robert G. and Stanislaw M. Ulam (1960). “On patterns of growth of figures in two dimensions”. In: *Notices of the American Mathematical Society* 7 (1960), p. 642.

——— (1967). *On Recursively Defined Geometrical Objects and Patterns of Growth*. Los Alamos Scientific Lab., N. Mex., 1967.

Schwartz, Barry (2009). *The Paradox of Choice*. HarperCollins, Oct. 13, 2009. 304 pp.

Schwartz, Susan (2012). *History and Future of the Berkeley Waterfront*. Jan. 29, 2012. URL: http://www.berkeleypaths.org/walkhandouts/walk_waterfront.htm.

Schüll, Natasha Dow (2012). *Addiction by Design: Machine Gambling in Las Vegas*. Princeton University Press, 2012. 457 pp.

Seabrook, John (2006). “Game Master: Will Wright changed the concept of video games with the Sims. Can he do it again with Spore?” In: *The New Yorker* (Nov. 6, 2006).

Searle, John R. (1980). “Minds, brains, and programs”. In: *Behavioral and brain sciences* 3.03 (1980), pp. 417–424.

Shannon, Claude E. (1953a). *A mind-reading (?) machine*. Memorandum. Bell Laboratories, 1953.

——— (1953b). “Computers and automata”. In: *Proceedings of the IRE* 41.10 (1953), pp. 1234–1241.

Sheff, David (1993). *Game over: how Nintendo zapped an American industry, captured your dollars, and enslaved your children*. Random House, May 1, 1993. 470 pp.

Shneiderman, B. (1983). “Direct Manipulation: A Step Beyond Programming Languages”. In: *Computer* 16.8 (Aug. 1983), pp. 57–69.

Shuler, Carly (2012). *What in the World Happened to Carmen Sandiego? The Edutainment Era: Debunking Myths and Sharing Lessons Learned*. New York, NY: The Joan Ganz Cooney Center at Sesame Workshop, 2012.

Sicart, Miguel (2011). “Against Procedurality”. In: *Game Studies* 11.3 (2011).

——— (2014). *Play Matters*. MIT Press, Aug. 15, 2014. 171 pp.

Silverman, Amy (1999). “How to Win at Life”. In: *The New York Times Magazine* (Oct. 31, 1999).

Slawson, David A. (1991). *Secret Teachings in the Art of Japanese Gardens: Design Principles, Aesthetic Values*. Kodansha International, 1991. 228 pp.

Small, James S. (2013). *The Analogue Alternative: The Electronic Analogue Computer in Britain and the USA, 1930-1975*. Routledge, June 17, 2013. 331 pp.

Smith, David Canfield, Charles Irby, Ralph Kimball, and Eric Harslem (1982). “The Star user interface: An overview”. In: *Proceedings of the June* (1982), pp. 7–10.

Solomon, Susan G. (2005). *American playgrounds: revitalizing community space*. UPNE, 2005. 286 pp.

Spariosu, Mihai I. (1989). *Dionysus Reborn: Play and the Aesthetic Dimension in Modern Philosophical and Scientific Discourse*. Cornell University Press, Sept. 28, 1989. 352 pp.

Spivack, Mayer (1969). “The political collapse of a playground”. In: *Landscape architecture* 59.4 (1969), pp. 288–291.

Squire, Kurt (2002). “Cultural framing of computer/video games”. In: *Game studies* 2.1 (2002), p. 90.

Stanton, Andrew (2012). “The clues to a great story”. TED. Feb. 2012.

Star, Susan Leigh and James R. Griesemer (1989). “Institutional ecology,translations’ and boundary objects: Amateurs and professionals in Berkeley’s Museum of Vertebrate Zoology, 1907-39”. In: *Social studies of science* 19.3 (1989), pp. 387–420.

Starr, Paul (1994). “Seductions of Sim: Policy as a Simulation Game”. In: *The American Prospect* 17 (Dec. 1994), pp. 19–29.

Steller, Chris (2014). “When ‘The Yard’ was Minnesota’s most radical park”. In: *MinnPost* (July 25, 2014).

Stenros, Jaakko (2015). “Playfulness, Play, and Games: A Constructionist Ludology Approach”. Ph.D. thesis. Tampere, Finland: University of Tampere, School of Information Sciences, 2015.

Stevens, William K. (1969). “Computer Is Used as Guide for Expert Seeking Way Out of Labyrinth of Urban Problems”. In: *The New York Times* (Oct. 31, 1969).

Strickland, Rachel (2014). “Curriculum Vitae”. Oct. 2014.

Suchman, Lucy A. (1987). *Plans and Situated Actions: The Problem of Human-machine Communication*. Cambridge University Press, 1987. 224 pp.

Sudnow, David (1983). *Pilgrim in the Microworld*. Warner Books, 1983. 227 pp.

Suits, Bernard Herbert (1978). *The Grasshopper: Games, Life and Utopia*. University of Toronto Press, 1978. 200 pp.

Sutton-Smith, Brian (1959). “The Kissing Games of Adolescents in Ohio”. In: *Midwest Folklore* 9.4 (Dec. 1, 1959), pp. 189–211. JSTOR: 4317818.

——— (1972). *The Folkgames of Children*. Publications of the American Folklore Society. Bibliographical and special series, 24. Austin: Published for the American Folklore Society by the University of Texas Press, 1972. 559 pp.

——— (1973). “A Developmental Structural Account of Riddles.” In: (1973).

——— (1980). “The Playground As A Zoo”. In: *The Association for the Anthropological Study of Play* 7.1 (1980).

Sutton-Smith, Brian (1986). *Toys as Culture*. Gardner Press, Inc., 19 Union Square West, New York, NY 10003, 1986.

——— (1995). “Conclusion: The persuasive rhetorics of play”. In: *The future of play theory: A multidisciplinary inquiry into the contributions of Brian Sutton-Smith*. Ed. by Anthony D. Pellegrini. 1995, pp. 275–295.

——— (1997). *The Ambiguity of Play*. Cambridge, Mass: Harvard University Press, 1997. 276 pp.

——— (1999). “Evolving a consilience of play definitions: playfully”. In: *Play and culture studies* 2 (1999), pp. 239–256.

Sutton-Smith, Brian and Diana Kelly-Byrne, eds. (1984). *The Masks of Play*. Association for the Anthropological Study of Play 8. New York: Leisure Press, 1984. 200 pp.

Sutton-Smith, Brian, J. M Roberts, and R. M Kozelka (1963). “Game involvement in adults”. In: *The Journal of Social Psychology* 60.1 (1963), pp. 15–30.

Sutton-Smith, Brian and John M. Roberts (1971). “The cross-cultural and psychological study of games”. In: *International Review for the Sociology of Sport* 6.1 (1971), pp. 79–87.

Swink, Steve (2009). *Game Feel: A Game Designer’s Guide to Virtual Sensation*. Taylor & Francis US, 2009. 377 pp.

Sørensen, C. T (1931). *Parkpolitik (Park policy)*. 1931.

——— (1951). “Junk playgrounds”. In: *Danish outlook* 4.1 (1951), pp. 311–316.

Tanner, Clive (1993). “SimCity in the Classroom”. In: *Classroom: The Magazine for Teachers* 13.6 (1993), pp. 37–39.

Taylor, Chris (2008). “Playwork and the theory of loose parts”. In: *Foundations of Playwork*. Ed. by Fraser Brown and Chris Taylor. Open University Press, 2008.

The Los Angeles Open Charter School. URL: https://www2.ed.gov/pubs/EdReformStudies/EdTech/opencharter.html (visited on 09/30/2015).

Toffoli, Tommaso (1977). “Cellular Automata Mechanics”. Ph.D. thesis. Ann Arbor, MI, USA: University of Michigan, 1977.

——— (1984). “Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics”. In: *Physica D: Nonlinear Phenomena* 10.1–2 (Jan. 1984), pp. 117–127.

Toffoli, Tommaso and Norman Margolus (1987a). “Cellular Automata Machines”. In: *Complex Systems* 1.5 (1987), pp. 967–993.

——— (1987b). *Cellular Automata Machines: A New Environment for Modeling*. MIT Press, 1987. 284 pp.

Tognazzini, Bruce (1993). “Principles, techniques, and ethics of stage magic and their application to human interface design”. In: *Proceedings of the INTERACT’93 and CHI’93 Conference on Human Factors in Computing Systems*. ACM, 1993, pp. 355–362.

Tomasello, M., M. Carpenter, J. Call, T. Behne, and H. Moll (2005). “Understanding and sharing intentions: The origins of cultural cognition”. In: *Behavioral and brain sciences* 28.5 (2005), pp. 675–690.

Tomasello, Michael (2010). *Origins of Human Communication*. A Bradford Book, Aug. 13, 2010. 408 pp.

Trainor, James (2012 Spring). “Reimagining Recreation”. In: *Cabinet* 45 (2012 Spring).

Treanor, Mike, Bobby Schweizer, Ian Bogost, and Michael Mateas (2011). “Proceduralist Readings: How to find meaning in games with graphical logics”. In: *Proceedings of the 6th International Conference on Foundations of Digital Games*. 2011, pp. 115–122.

——— (2012). “The micro-rhetorics of Game-O-Matic”. In: *Proceedings of the International Conference on the Foundations of Digital Games*. ACM, 2012, pp. 18–25.

Tsing, Anna Lowenhaupt (2015). *The Mushroom at the End of the World: On the Possibility of Life in Capitalist Ruins*. Princeton University Press, 2015. 352 pp.

Tufte, Edward R. (1990). *Envisioning information*. Graphics Press, 1990. 136 pp.

Turing, Alan M. (1950). “Computing machinery and intelligence”. In: *Mind* (1950), pp. 433–460. JSTOR: 2251299.

——— (1952). “The chemical basis of morphogenesis”. In: *Bulletin of mathematical biology* 52.1 (Mar. 15, 1952), pp. 153–197.

Turkle, Sherry (1980). “Computer as Rorschach”. In: *Society* 17.2 (1980), pp. 15–24.

Turkle, Sherry (1984). *The Second Self: Computers and the Human Spirit*. New York: Simon and Schuster, 1984. 362 pp.

Turner, Fred (2006). *From counterculture to cyberculture: Stewart Brand, the Whole Earth Network, and the rise of digital utopianism*. University Of Chicago Press, 2006.

——— (2013). *The Democratic Surround: Multimedia and American Liberalism from World War II to the Psychedelic Sixties*. University of Chicago Press, Dec. 4, 2013. 374 pp.

Turner, Victor (1982). "Liminal to liminoid, in play, flow, and ritual: an essay in comparative symbology." In: *From Ritual to Theatre*. 1982, pp. 20–60.

——— (1986). "Body, brain and culture". In: *Performing Arts Journal* 10.2 (1986), pp. 26–34.

Ulam, Stanislaw (1952). "Random processes and transformations". In: *Proceedings of the International Congress on Mathematics*. Vol. 2. 1952, pp. 264–275.

——— (1958). "John von Neumann 1903-1957". In: *Bulletin of the American Mathematical Society* 64.3 (1958), pp. 1–49.

——— (1962). "On Some Mathematical Problems Connected with Patterns of Growth and Figures". In: American Mathematical Society Proceedings of Symposia in Applied Mathematics. Vol. 14. 1962, pp. 215–224.

Vance, Bill (1976). "A Guide to the Development of the Adventure Playground". 1976.

——— (1982). "Adventure Playgrounds: The American Experience". In: *Parks and Recreation* 17.9 (Sept. 1982), pp. 67–70.

Vichniac, Gérard Y. (1984). "Simulating physics with cellular automata". In: *Physica D: Nonlinear Phenomena* 10.1–2 (Jan. 1984), pp. 96–116.

Victor, Bret (2013). "Media for Thinking the Unthinkable". MIT Media Lab, Apr. 4, 2013.

Von Neumann, John (1945). *First Draft of a Report on the EDVAC*. June 30, 1945.

——— (1951). "The general and logical theory of automata". In: *Cerebral mechanisms in behavior* (1951), pp. 1–41.

——— (1958). *The Computer and the Brain*. Yale University Press, 1958. 82 pp.

——— (1966). *Theory of Self-Reproducing Automata*. Ed. by Arthur W. Burks. University of Illinois Press, 1966. 416 pp.

——— (1993). "First Draft of a Report on the EDVAC". In: *IEEE Annals of the History of Computing* 15.1 (1993). Ed. by Michael D. Godfrey.

Vygotsky, Lev S. (1978). *Mind in society*. 1978.

Wainwright, Robert (1972). "Life Saver". In: *Lifeline* 5 (Sept. 1972).

——— (1974). "Life is universal!" In: *Proceedings of the 7th conference on Winter simulation - Volume 2*. WSC '74. Washington, DC: Winter Simulation Conference, 1974, pp. 449–459.

——— (2010). "Conway's Game of Life: Early Personal Recollections". In: Adamatzky, A. *Game of Life Cellular Automata*. Springer-Verlag, 2010, pp. 11–16.

Waldrop, M. Mitchell (2001). *The Dream Machine: J.C.R. Licklider and the Revolution That Made Computing Personal*. Viking, 2001. 520 pp.

Wallis, Alistair (2006). *Column: 'Playing Catch Up: Stormfront Studios' Don Daglow'*. Oct. 19, 2006. URL: http://www.gamasutra.com/view/news/102180/Column_Playing_Catch_Up_Stormfront_Studios_Don_Daglow.php (visited on 05/01/2015).

Ward, John E. (1959). *MOUSE–Preliminary Instructions*. Jan. 16, 1959.

Wardrip-Fruin, Noah (2005). "Playable media and textual instruments". In: *The Aesthetics of Net Literature: Writing, Reading and Playing in Programmable Media* (2005), pp. 211–388.

——— (2006). "Expressive Processing: On Process-Intensive Literature and Digital Media". Ph.D. thesis. Citeseer, 2006.

——— (2009). *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. MIT Press, Sept. 30, 2009. 504 pp.

Wark, McKenzie (2015). *The Avery Review | Capture All: SimCity, Gamespace and Play*. Mar. 2015. URL: http://averyreview.com (visited on 03/08/2015).

Weizenbaum, Joseph (1966). "ELIZA—a computer program for the study of natural language communication between man and machine". In: *Communications of the ACM* 9.1 (1966), pp. 36–45.

Westervelt, Eric (2014). *Where The Wild Things Play*. Aug. 5, 2014. URL: http://www.npr.org/sections/ed/2014/08/04/334896321/where-the-wild-things-play (visited on 11/25/2015).

What to Do After You Hit Return Or P.C.C.'s First Book of Computer Games (1975). Hewlett-Packard / People's Computer Company, 1975. 167 pp.

Wiederholt, Kristin (2006). *'Adventure Playgrounds' a Dying Breed in the U.S.* Mar. 9, 2006. URL: http://www.npr.org/templates/story/story.php?storyId=5254026 (visited on 11/25/2015).

Wiener, Norbert (1961). *Cybernetics Or Control and Communication in the Animal and the Machine*. MIT Press, Jan. 1, 1961. 244 pp.

Wilson, Doug (2011). "In Celebration of Low-Process Intensity". Unpublished Manuscript. Nov. 14, 2011.

Wilson, Penny (2010). "The playwork primer". In: *Alliance for Childhood* (2010).

——— (2013). *'children are more complicated than kettles.' the life and work of Lady Allen of Hurtwood.* 2013. URL: http://theinternationale.com/pennywilson/38-2/ (visited on 04/07/2013).

——— (2014). *good enough: d.w. winnicott and playwork*. United Kingdom, 2014.

Winnicott, Donald Woods (1953). "Transitional objects and transitional phenomena". In: *The International journal of psycho-analysis* 34 (1953), p. 89.

——— (1971). *Playing and reality*. Basic Books, 1971. 200 pp.

Wittgenstein, Ludwig (1953). *Philosophical Investigations*. Blackwell, 1953. 268 pp.

Wolfram, Stephen (1983). "Statistical mechanics of cellular automata". In: *Reviews of modern physics* 55.3 (1983), p. 601.

——— (1984). "Universality and complexity in cellular automata". In: *Physica D: Nonlinear Phenomena* 10.1 (1984), pp. 1–35.

——— (2002). *A New Kind of Science*. Wolfram Media, 2002. 1288 pp.

Wolpert, Arlen (1992). “Application of System Dynamics to the Study of a Religious Experience”. In: *Proceedings of the 10th International Conference of the System Dynamics Society*. Utrecht, Netherlands, 1992.

Wood, David, Jerome S. Bruner, and Gail Ross (1976). “The role of tutoring in problem solving”. In: *Journal of child psychology and psychiatry* 17.2 (1976), pp. 89–100.

Woolgar, Steve (1991). “Configuring the user: the case of usability trials”. In: *A Sociology of monsters: Essays on Power, Technology and Domination*. Ed. by John Law. Routledge, 1991, pp. 66–75.

Wozniak, Steve (1986). *And Then There Was Apple*. Trans. by Jack Connick. 1986. URL: HTTP://APPLE2HISTORY.ORG/MUSEUM/ARTICLES/CA8610/ (visited on 02/05/2013).

——— (2014). *How Steve Wozniak Wrote BASIC for the Original Apple From Scratch*. May 1, 2014. URL: HTTP://GIZMODO.COM/HOW-STEVE-WOZNIAK-WROTE-BASIC-FOR-THE-ORIGINAL-APPLE-FR-1570573636 (visited on 05/28/2015).

Wright, Robert (1988). “Did the universe just happen?” In: *The Atlantic Monthly* 261.4 (Apr. 1988), pp. 29–.

Wright, Will (1989). *SimCity*. Computer program. Maxis Software Inc., 1989.

——— (2003). “Dynamics for Designers: Why I Hate Calculus... and why I Love Compression”. Game Developers Conference. San Jose, CA, 2003.

——— (2011). “Classic Game Postmortem: Raid On Bungeling Bay”. Game Developers Conference. San Francisco, CA, 2011.

——— (2016). *Interview by Chaim Gingold*. Feb. 8, 2016.

Yaeger, Larry. “Vivarium History”. Web page.

——— (1994). “PolyWorld: Life in a New Context”. In: *Proc. Artificial Life* 3 (1994), pp. 263–263.

Zenil, Hector (2012). *A Computable Universe: Understanding and Exploring Nature as Computation*. Singapore ; Hackensack, N.J: World Scientific Publishing Company, Oct. 30, 2012. 856 pp.

Zhang, Jiaje and Donald A. Norman (1994). “Representations in distributed cognitive tasks”. In: *Cognitive science* 18.1 (1994), pp. 87–122.

Zhang, Jiajie (1991). “The interaction of internal and external representations in a problem solving task”. In: *Proceedings of the thirteenth annual conference of cognitive science society*. Vol. 88. Erlbaum Hillsdale, NJ, 1991, p. 91.

Zhu, Jichen (2009). “Intentional systems and the artificial intelligence (AI) hermeneutic network: Agency and intentionality in expressive computational systems”. Ph.D. thesis. Atlanta, GA: Georgia Institute of Technology, Aug. 2009.

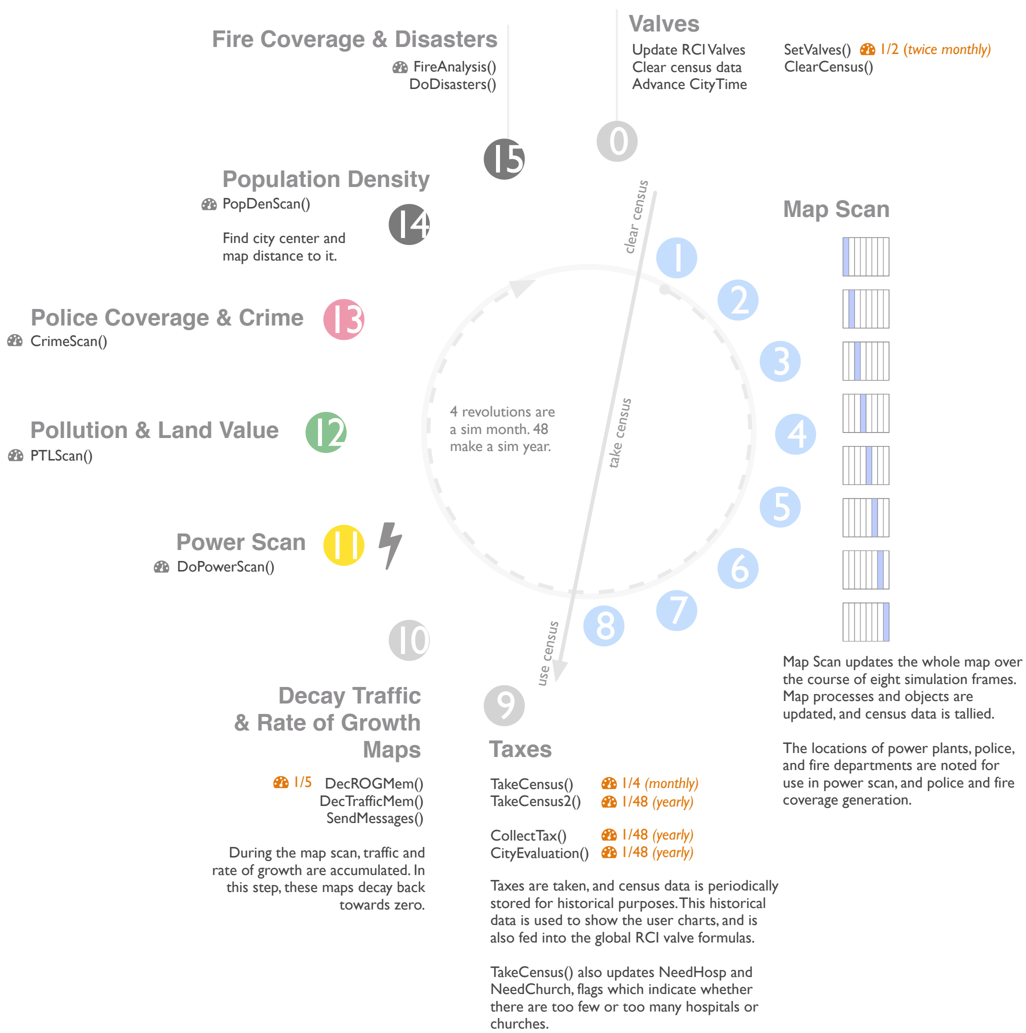
Zuse, Konrad (1969). *Calculating Space*. Massachusetts Institute of Technology, Project MAC, 1969–1970. 188 pp.

Appendix A

SimCity Diagrams

Simulate()

City simulation is broken down into 16 steps. Each revolution advances the city time by 1. Every frame of the game, one of these 16 steps is performed.



	Slow	Medium	Fast
DoPowerScan()	1/2	1/4	1/5
PTLScan()	1/2	1/7	1/17
CrimeScan()	1	1/8	1/18
PopDenScan()	1	1/9	1/19
FireAnalysis()	1	1/10	1/20

To keep up with faster simulation speeds (selected by the player in the Speed menu), certain processes run less often.

Some steps always run at a fixed, lower frequency.

Figure A.1: Simulate

Anatomy of a Map[] cell

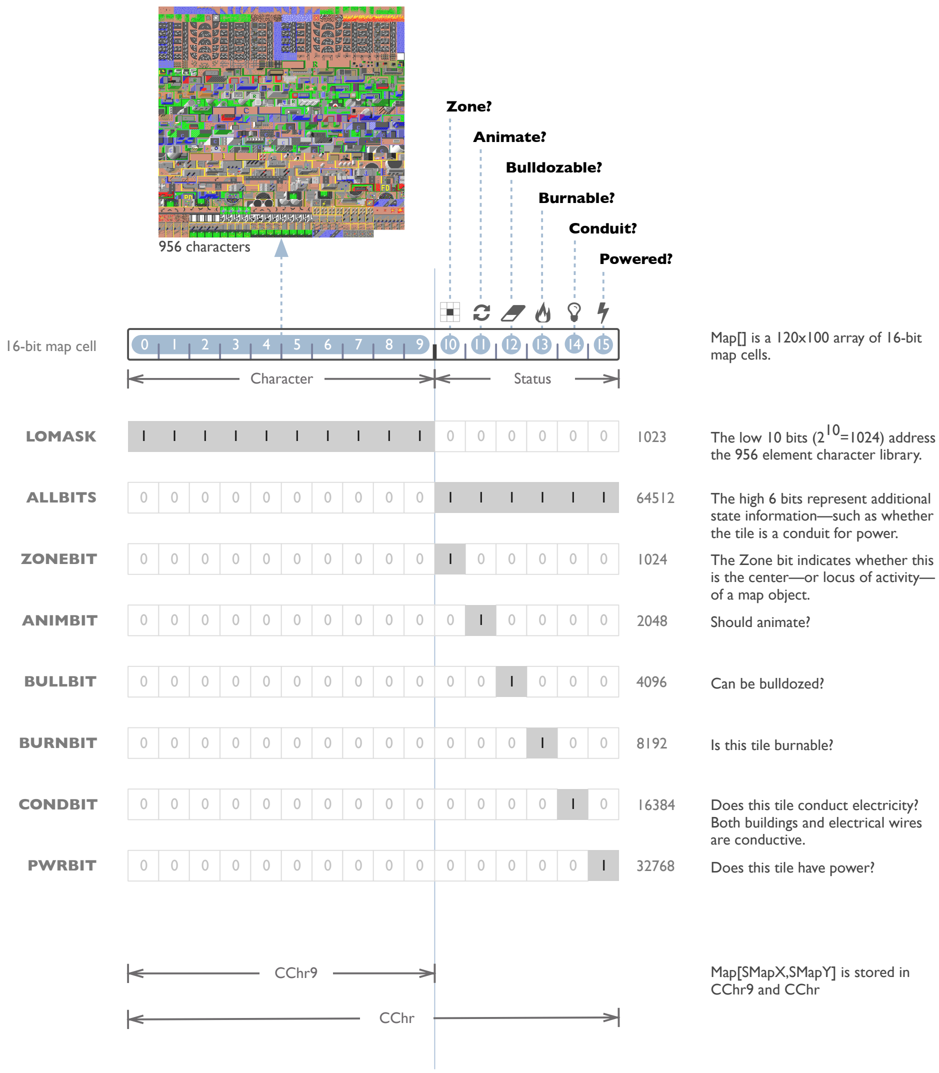


Figure A.2: Anatomy of a Map Cell

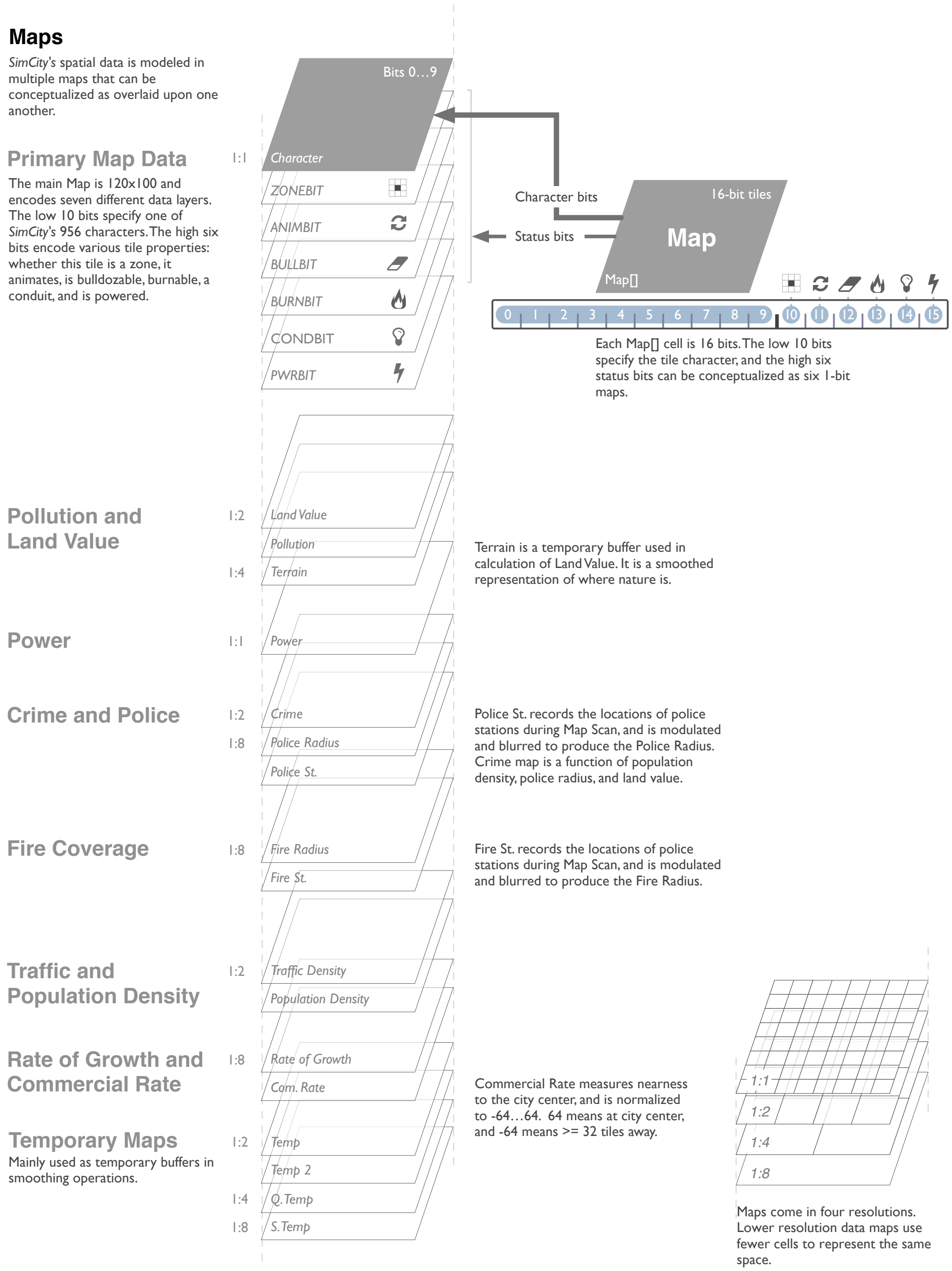


Figure A.3: Maps

Map Data Flow

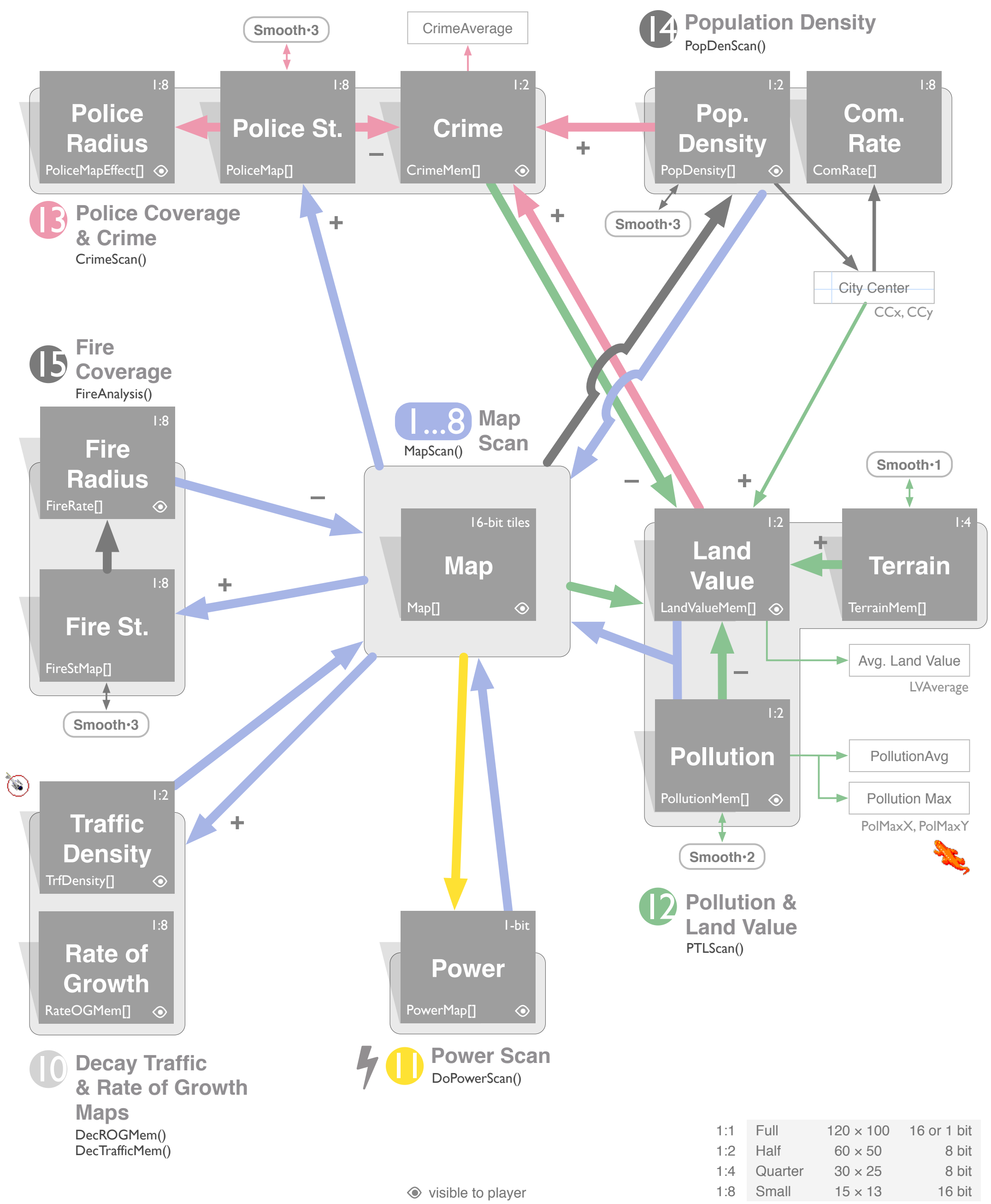
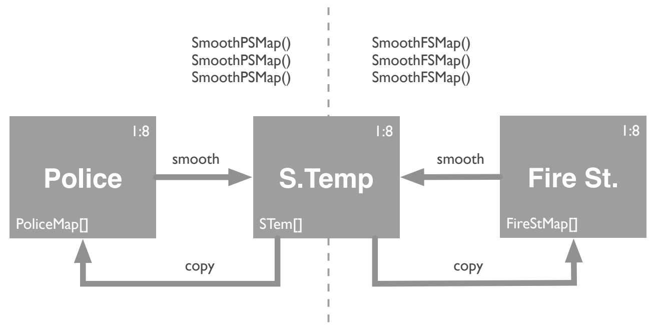


Figure A.4: Map Data Flow

Map Smoothing

Smooth·3

Each of the three iterations of the police and fire coverage map blurring smoothes into STem[], then copies back.



Smooth·3

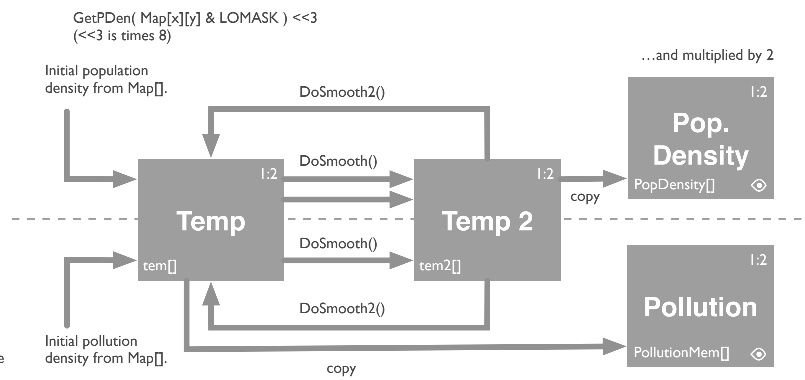
Population density is initially written into tem[], then smoothed three times using the two half scale buffers.

DoSmooth()
DoSmooth2()
DoSmooth()

Smooth·2

Pollution map is initially written into tem[], then smoothed twice using the two half scale buffers.

DoSmooth()
DoSmooth2()



Smooth·2

PTLScan() accumulates the natural landscape into QTem[], +15 for each tile < RUBBLE, which is dirt, water, woods, and park.

It is smoothed once, into TerrainMem[], and then used by the land value calculation. A time delay of one simulation cycle is inherent in this operation, as TerrainMem[] is consumed on the next simulation cycle.

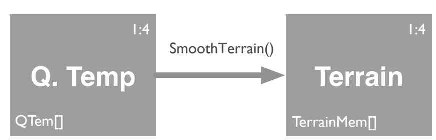


Figure A.5: Map Smoothing

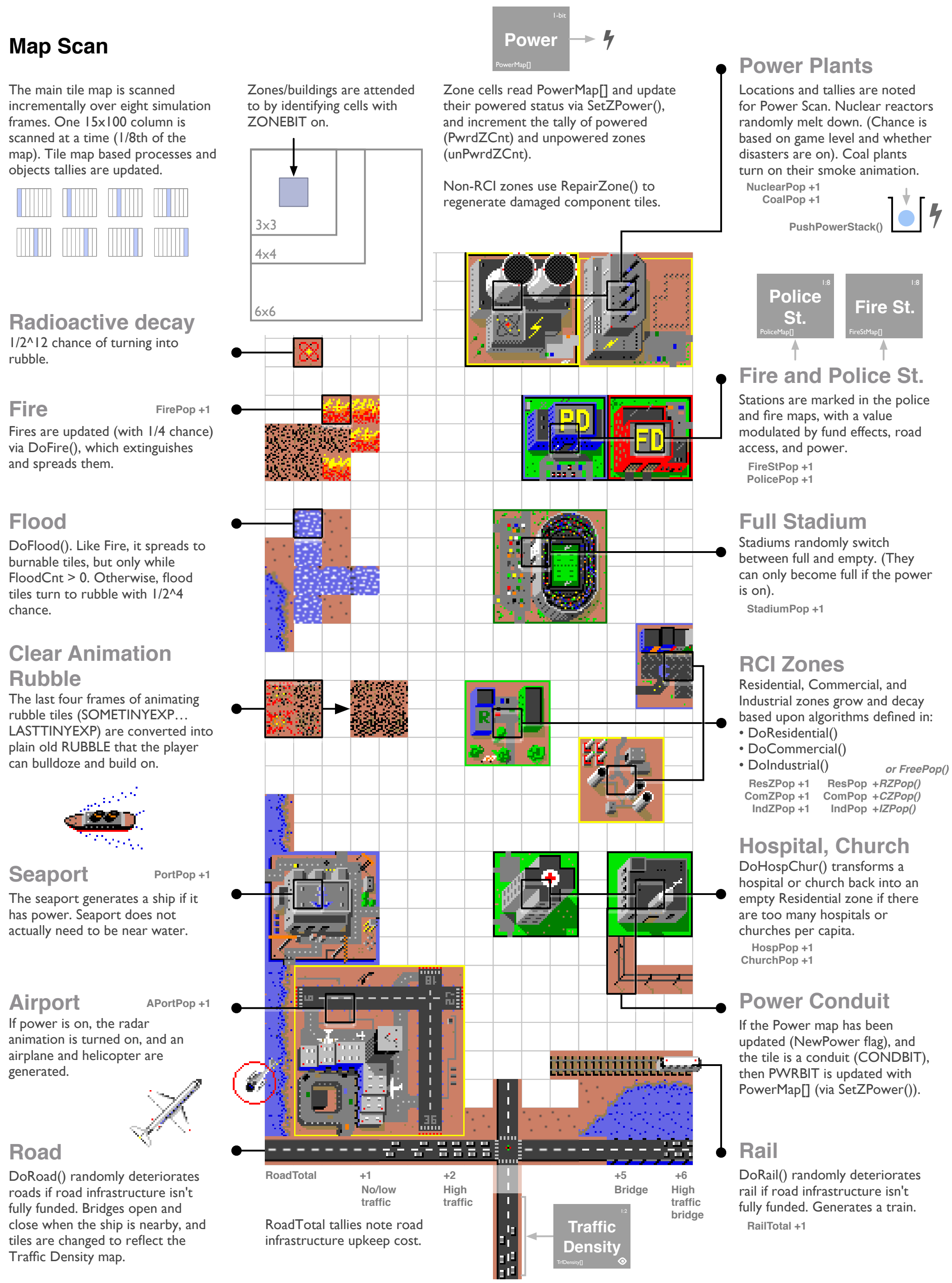
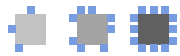


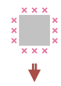
Figure A.6: Map Scan

DoResidential(), DoCommercial(), DoIndustrial()

Make Traffic

Zones randomly make traffic. More populous zones make traffic more often. Whether or not make traffic succeeds in reaching its destination only affects commercial zones, which will not grow if their destination is not reached.



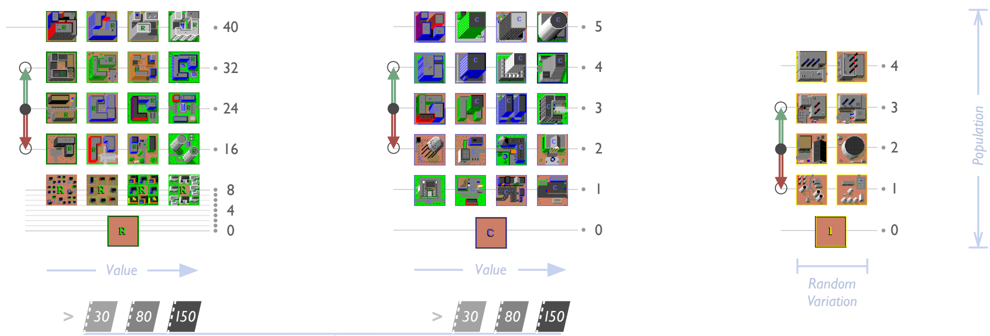


If Make Traffic is attempted and has a hard failure—it can't find a perimeter road—then the zone always decays.

Growth and Decay

Zones have a chance of evaluation. Evaluation sums global and local valves to produce a score that determines the probability of incremental growth and decay.

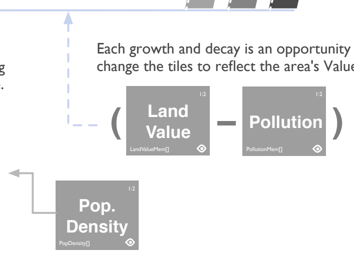
When a zone grows or decays, it swaps out the tiles used, which represents a different population.



Single tile homes represent population increments between 0 and 8. When building these, new homes reflect the current Value. Thus, the same zone can contain single tile homes of mixed Value.

Residential population can skip upward to apartment buildings (a population of 16) if the population density map is >64.

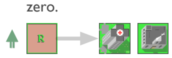
Each growth and decay is an opportunity to change the tiles to reflect the area's Value.



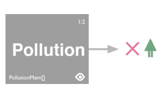
Each industrial growth and decay tosses a coin to determine which visual variation to use.

Hospital & Church

If a residential zone is empty and decides to grow, then there is a 1/4 chance that the zone will become a hospital or a church if the global flags *NeedHosp* or *NeedChurch* are 1. If one is made, then the flag is set to zero.

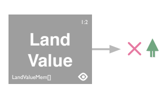


When updating Industrial zones, animations are always turned off and on based upon whether the zone has power.



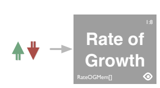
Pollution too high?

Residential zones can never grow if pollution is >128. (This doesn't affect emergence of hospitals and schools.)



Land value too low?

Commercial zones can never grow to a population beyond Land Value / 32.



Recording Growth

Every incremental growth and decay is noted in the Rate of Growth map. Each step counts as ±8, except single homes, which count ±1.

Figure A.7: Do RCI

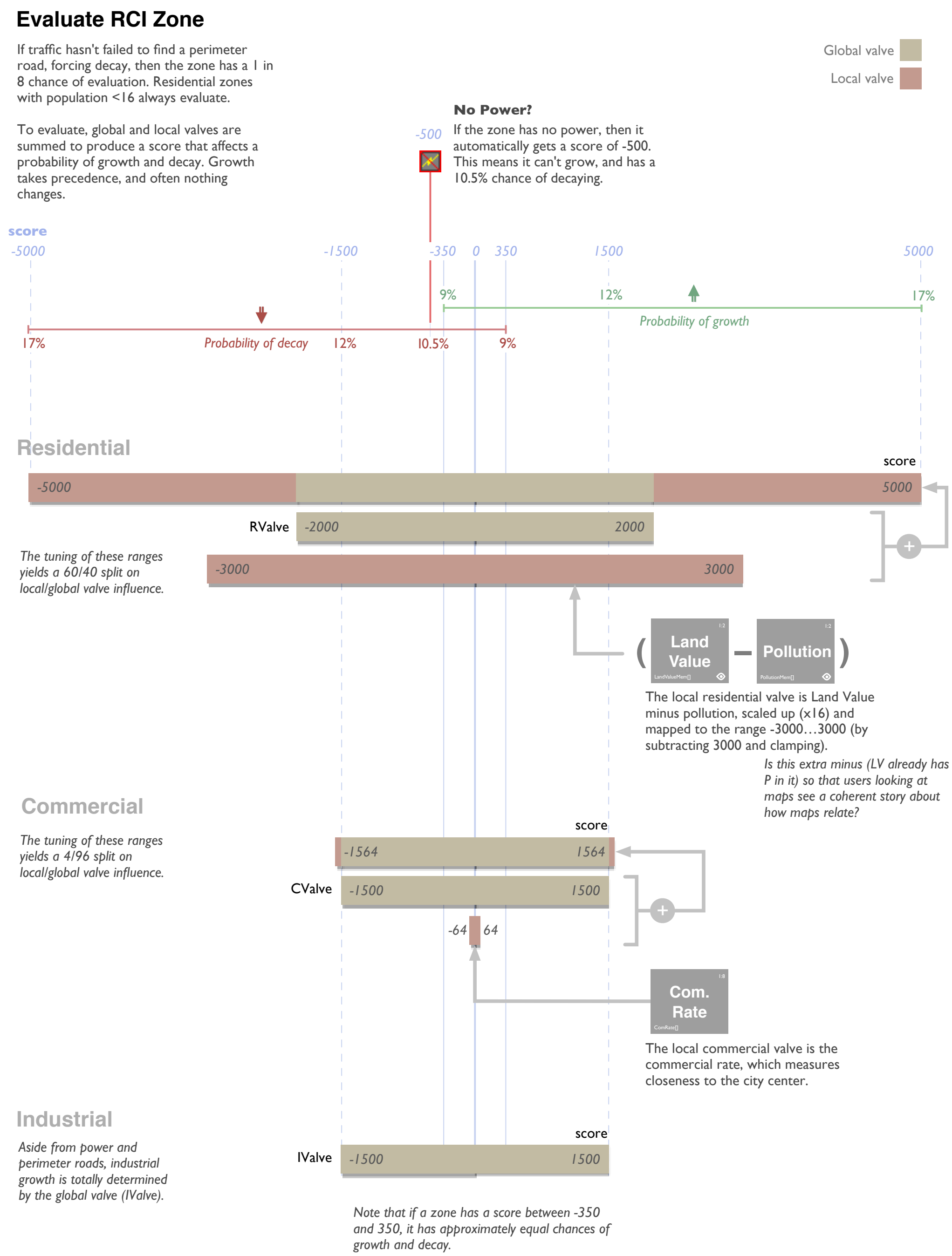
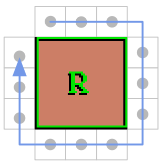


Figure A.8: Evaluate RCI

Make Traffic
MakeTraf()

Make Traffic can be invoked when Map Scan evaluates Residential, Commercial, and Industrial Zones. It returns either success (1: destination found), failure (0: destination not found), or hard failure (-1: no perimeter road found).



Origin *FindPRoad()*

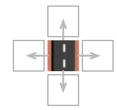
To begin a trip, a road tile on the perimeter of the originating zone must be located. The first tile found in a clockwise search from the top left is used.

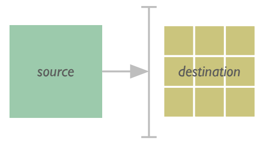
Any road or rail tile will be used in the search. e.g.:

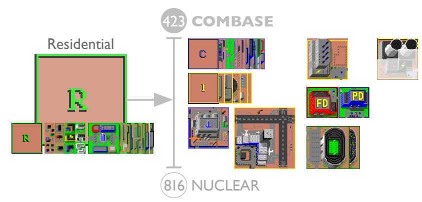
RoadTest()

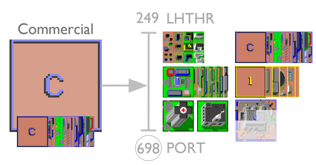
Desination *DriveDone()*

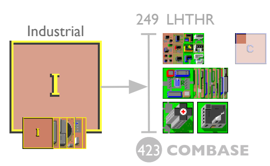
Each zone type attempts to make a trip to a particular set of destination tiles. On the trip, adjacent tiles are tested to see if they fall within a range of tile indices. The drive succeeds when one of these destination tiles is found.



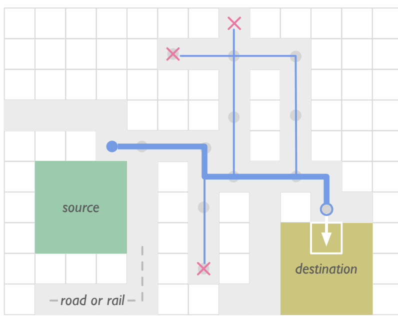




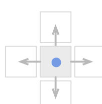




Drive *TryDrive()*



After finding an origin, a random route using any combination of road and rail is tried. At every fork, a random turn is made.



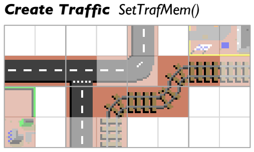
At each step, adjacent tiles are tested to see if they constitute valid destination tiles for the source zone type.

A stack is used to remember *every other step* of the path under consideration.

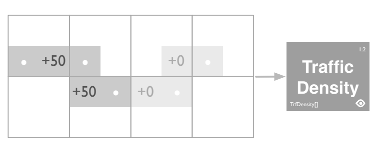
If a dead end is encountered, then the search backtracks by popping the stack.

At most **30** steps will be taken before giving up. Each dead end encountered costs an additional 3 steps.

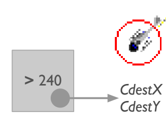
Create Traffic *SetTrafMem()*



If a trip is successfully completed, the path stored in the path stack—which recorded every other step— will contribute to the traffic density map.



Each road tile in the stack adds 50 to traffic density. Rail tiles add 0. Note that the traffic map is at 1/2 scale, so four road tiles correspond to one traffic density entry.

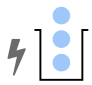


If accumulated traffic exceeds 240, then the helicopter (if there is one) will come to the high traffic tile's location.

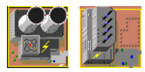
Figure A.9: Make Traffic

Power Scan

DoPowerScan()



Following MapScan(), the power stack has been seeded with the locations of all power plants. The power plant population has also been tallied.

NuclearPop ×  =
CoalPop

1 Nuclear = 2000
1 Coal = 700

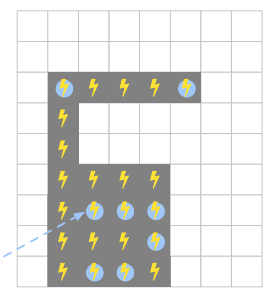
MaxPower
Count of total power available. One unit powers one tile.

NumPower

NumPower starts at zero, and will count total power consumed. Each conduit tile (power line and building tiles) scanned consumes one unit of power.

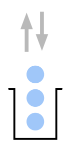
PowerMap is a 1-bit array at 1:1 scale to the main map. It represents whether a tile is powered. Power Scan begins by initializing this map to zero. During its search, it flips bits on and increments NumPower.

After Power Scan is done, MapScan will read PowerMap and set whether a tile is powered (its PWRBIT).



If power needed exceeds power available (MaxPower < NumPower), then Power Scan stops. All unscanned tiles remain marked as unpowered (zero) in PowerMap[].

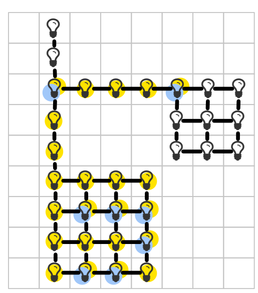
Search started here.

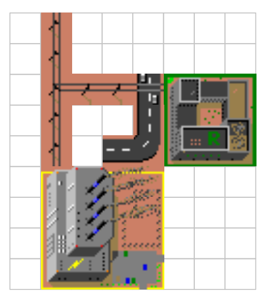


The power stack marks where the search will continue (or begin) from. As branches are encountered, they are pushed onto the stack. When the stack is empty, the search is done.

Power Scan attends to CONDBIT, and not Map[]'s tile characters. The one exception is detection and avoidance of power plant zone tiles (which is redundant, as powered conduits are ignored).

A tile's CONDBIT defines whether it conducts and consumes electricity. It defines the topology power scan traverses.





The algorithm employed means that only one power plant must be connected to the grid for all power plants to feed the power grid!

Figure A.10: Power Scan

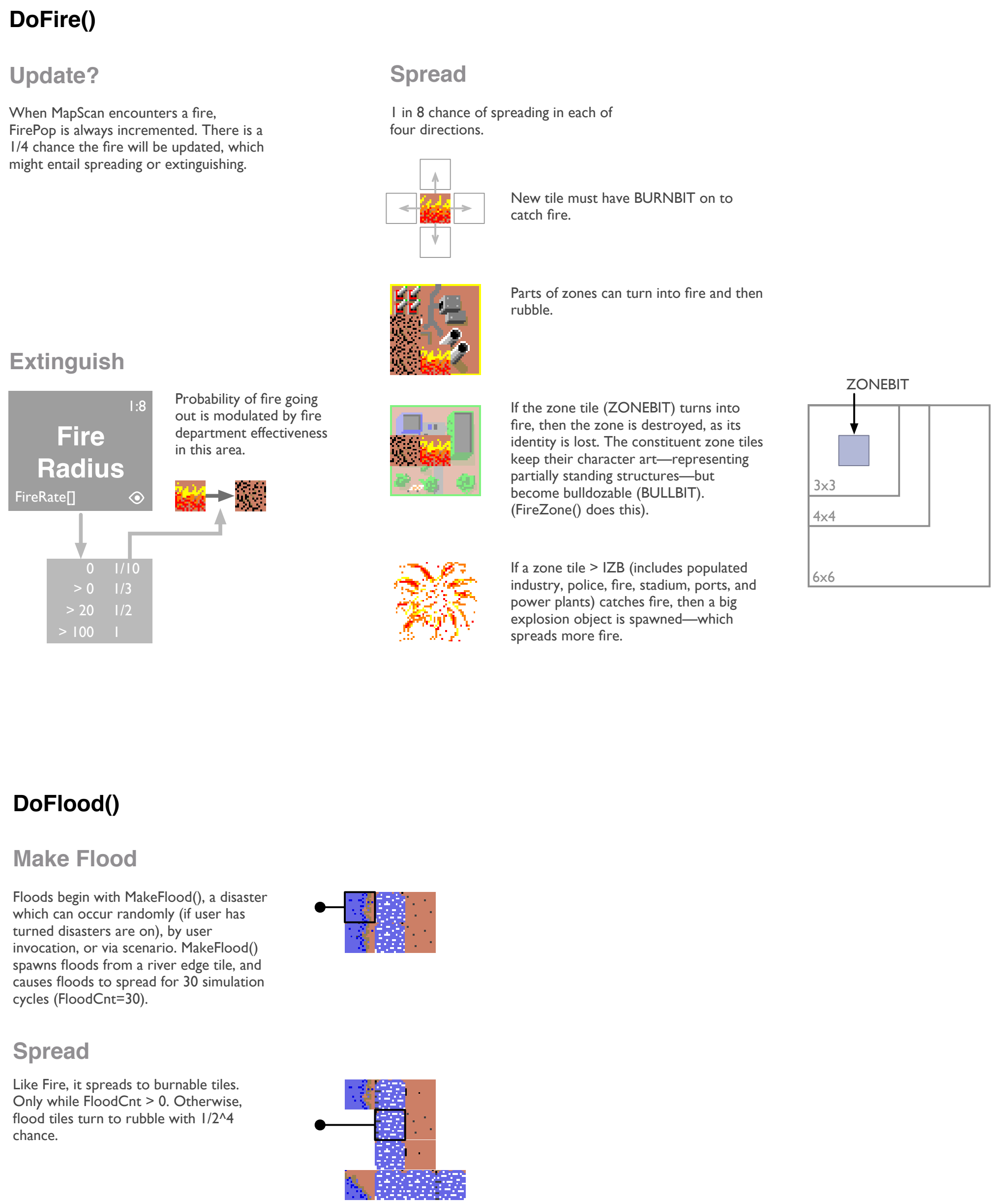


Figure A.11: DoFire and DoFlood

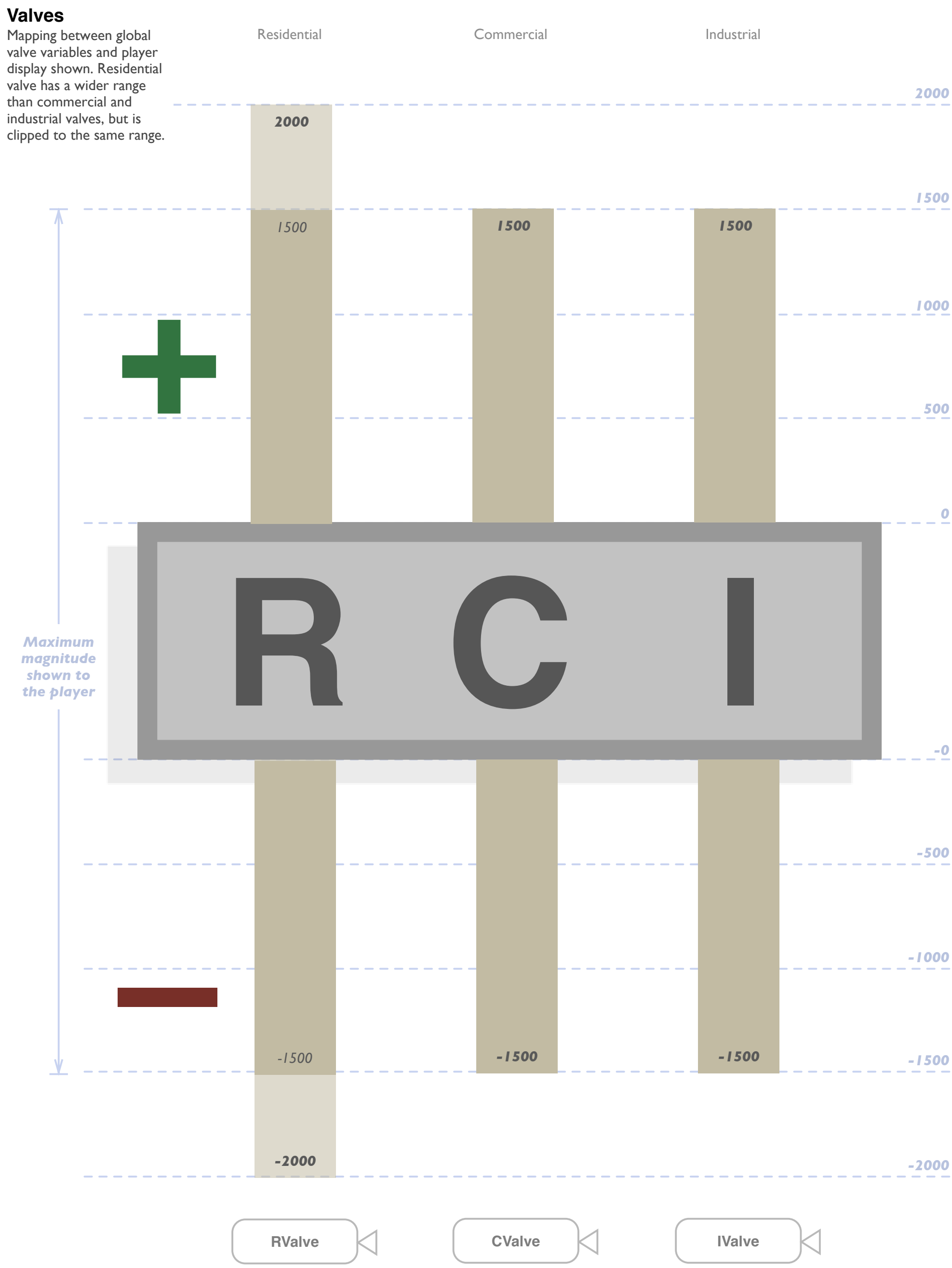


Figure A.12: Valves

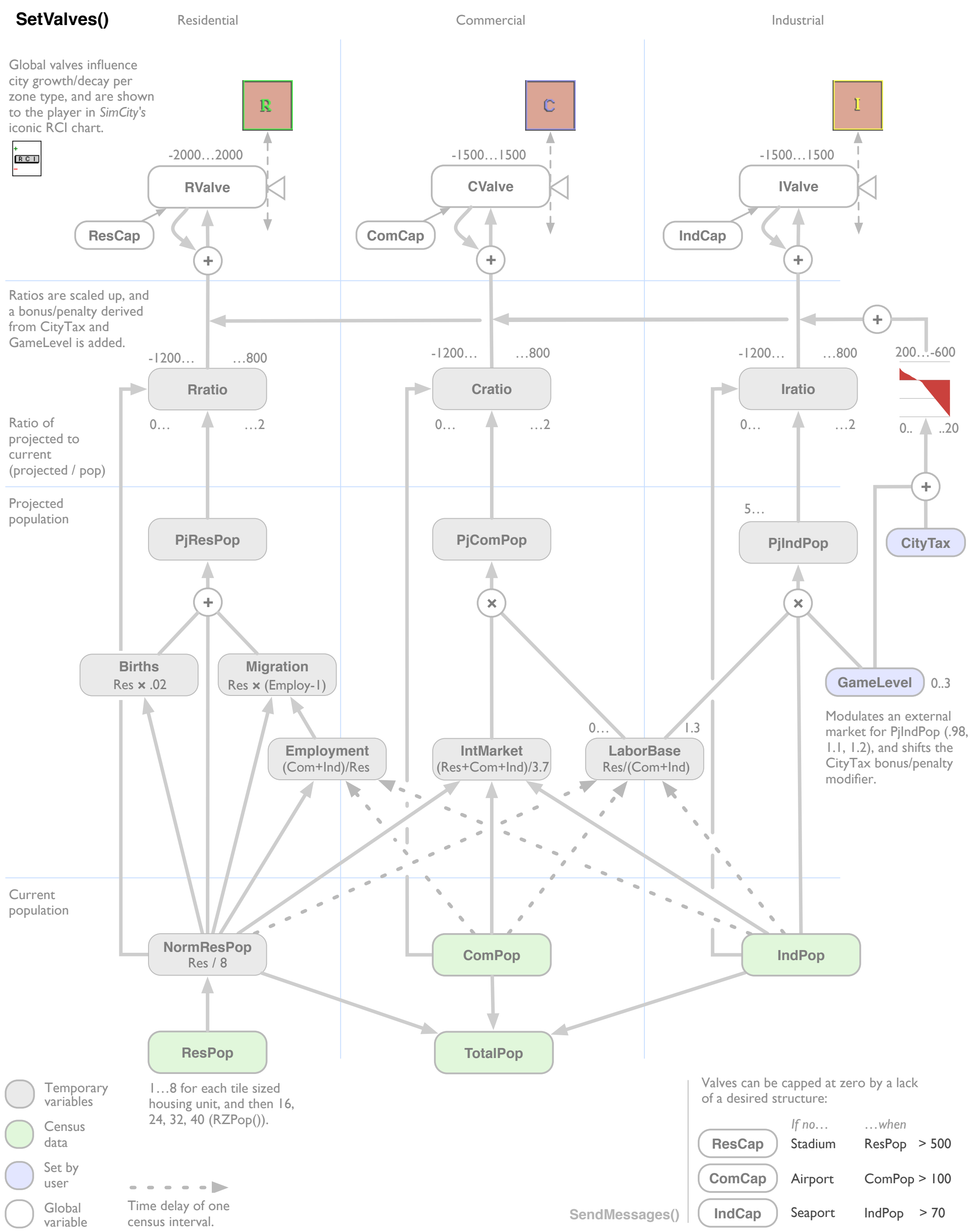


Figure A.13: SetValves

Agents

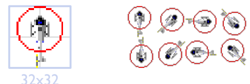
TRA Train

Spawned by train tracks, it randomly wanders the map.

COP Copter

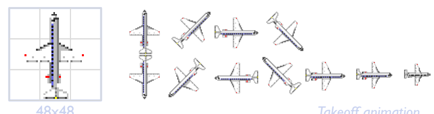
If it encounters heavy traffic (>160) in the traffic density map, it will make a report. High traffic tiles encountered during traffic generation will become the copter's destination.

When spawned, it goes to a random destination. When a destination is reached it returns to the airport.



AIR Airplane

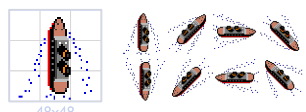
Takes off from the airport, and randomly flies around the map. Can crash into the copter (destroying both of them in an explosion object).



SHI Ship

Spawned by the Port, and randomly wanders the map. Can disappear off the map, or run aground and destroy map tiles and itself.

Makes sounds.



GOD Godzilla

Emerges out of water and heads towards the city's most polluted location. After reaching this destination, it returns to its origin (and disappears). Can randomly spawns if disasters are on and average pollution exceeds a threshold (80).

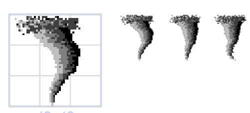
Can collide with and destroy airplanes, copters, ships, trains, and map tiles.

Makes sounds.

TOR Tornado

Lasts for 200 ticks. Wanders the map, and can disappear off the map.

Can collide with and destroy airplanes, copters, ships, trains, and map tiles.



EXP Explosion

Spawned during some object collisions and zone explosions. Spreads fire to some nearby map tiles.

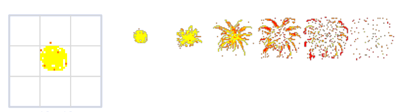


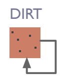
Figure A.14: Agents

Tile Animation

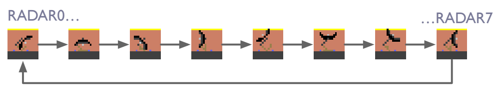
Transitions

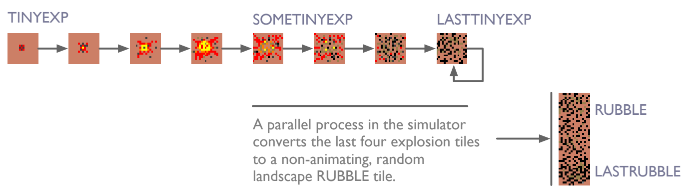
An animation process runs independent of the simulation logic. The simulator simply toggles a tile's ANIMBIT to specify whether an animation should play. animateTiles() scans the entire map, and updates tiles with the ANIMBIT status bit set to on.

A lookup table (aniTile[]) specifies the next tile in the animation sequence for every map tile.



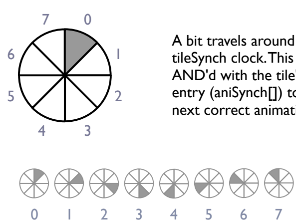
Most tiles simply point to themselves, which makes the system robust in the face of errors in which ANIMBIT might inadvertently be on, and also helps programmers interpret the data table (a 1024 element array hard coded in C).



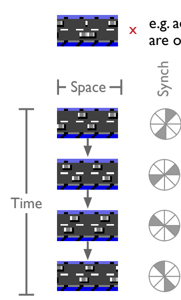


Synchronization

An 8-state clock (tileSynch) controls the phase (synchronization) of animations.

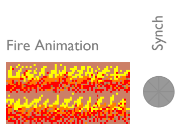


A bit travels around the tileSynch clock. This is bitwise AND'd with the tile's synch entry (aniSynch[]) to find the next correct animation frame.



e.g. adjacent traffic tiles whose animation are out of phase will look broken.

Animations are updated every time animateTiles() is called. Synch bits guide animations into globally synchronized patterns.



Animations that don't need to be synchronized (e.g. multiple fires) have all synch bits turned on.

With this technique, animations up to 8 frames long can be synchronized.

Figure A.15: Tile Animation

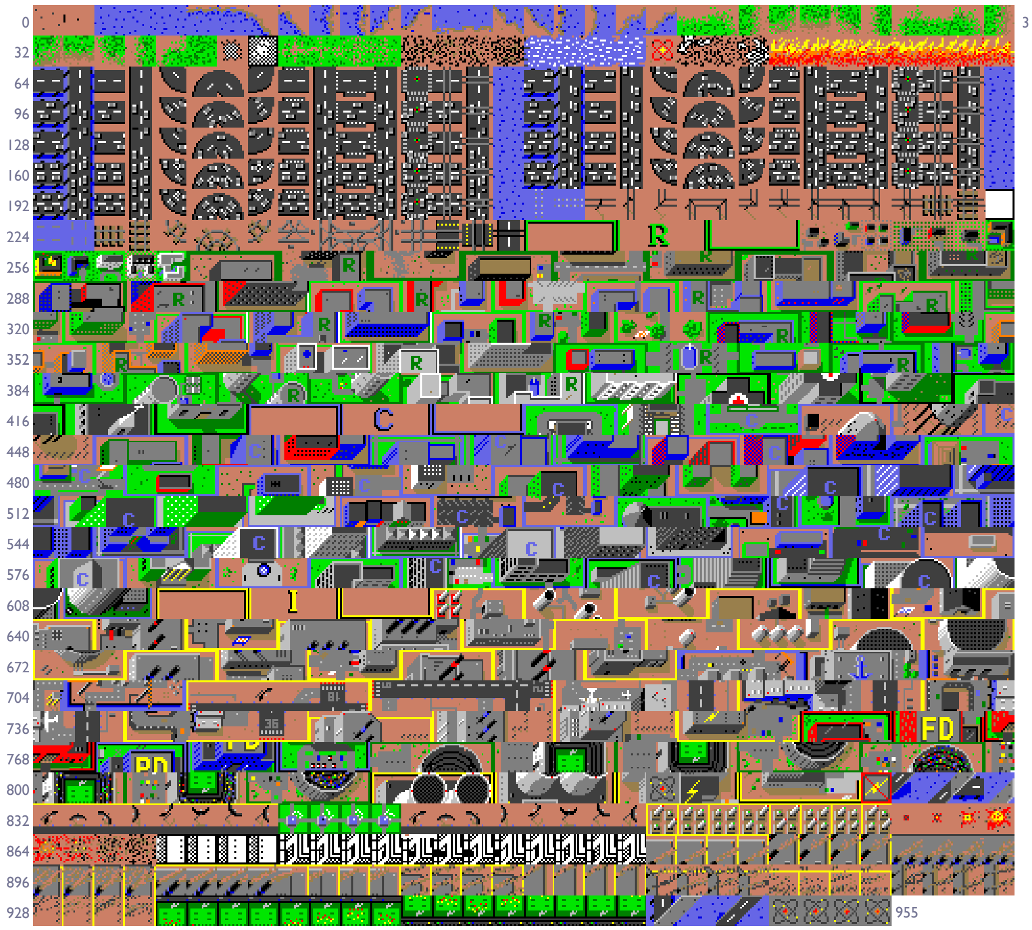


Figure A.16: All Characters

Character Overview **0-955**

Figure A.17: Character Overview

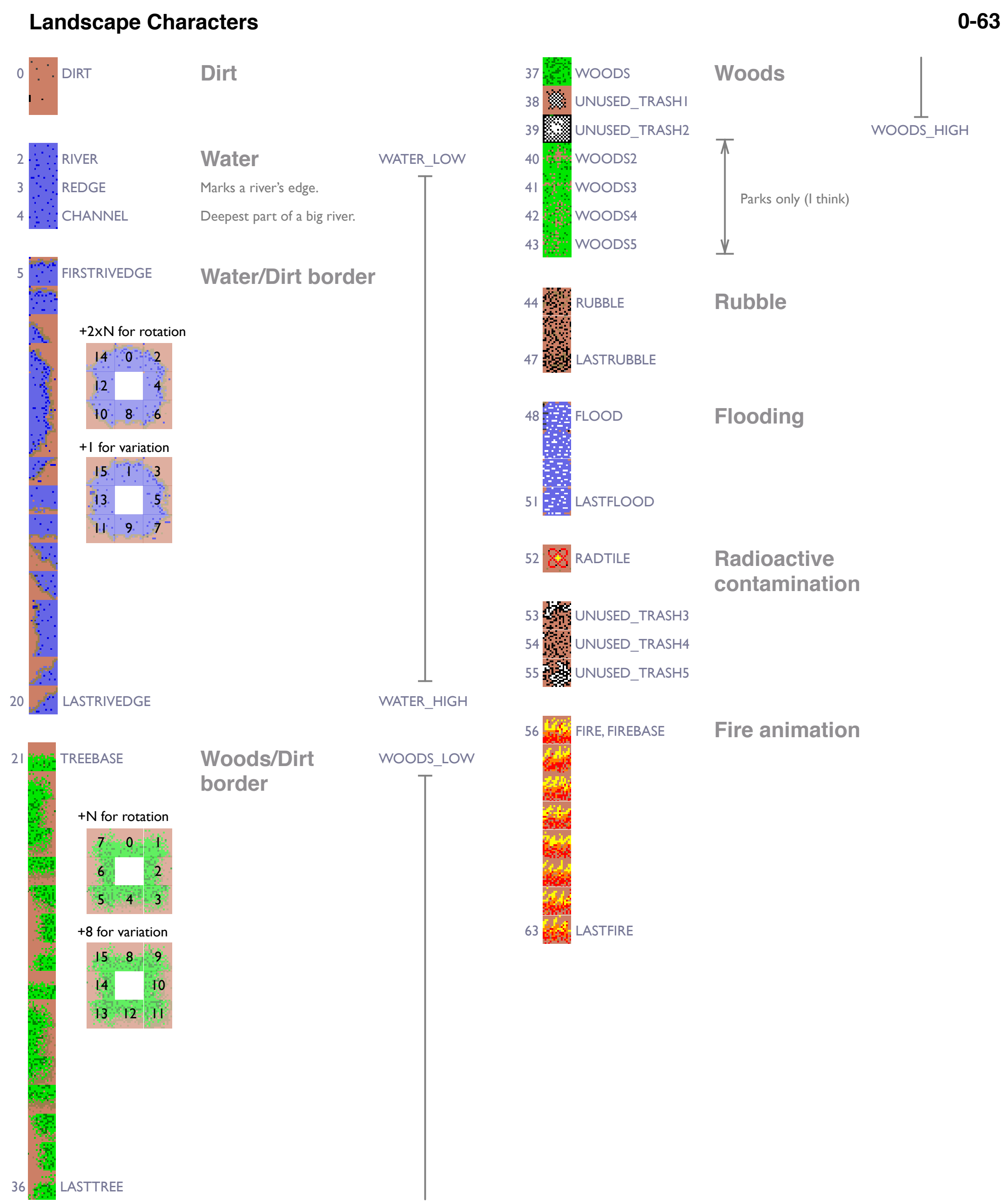


Figure A.18: Landscape Characters (0–63)

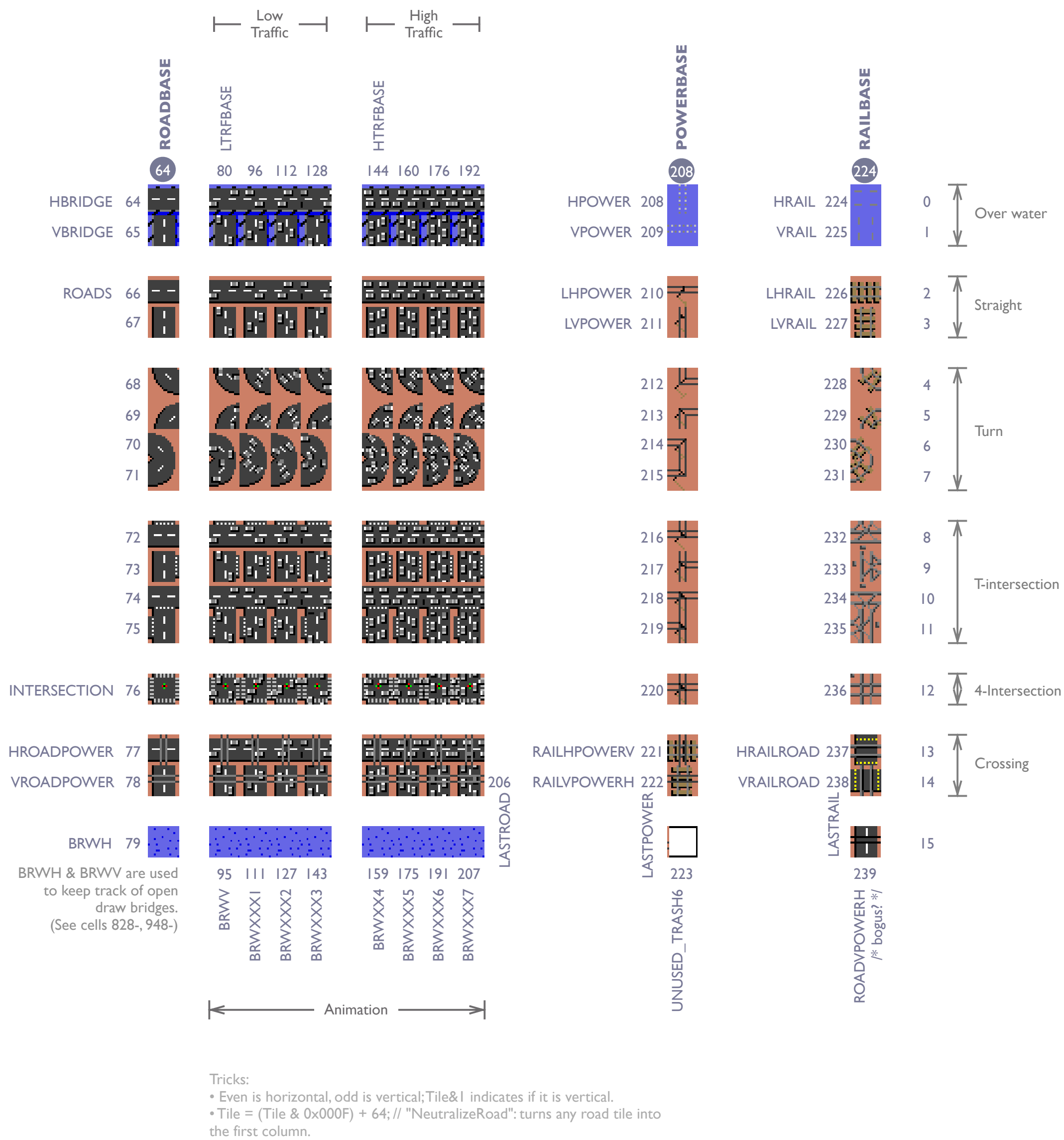


Figure A.19: Network Characters (64–239)

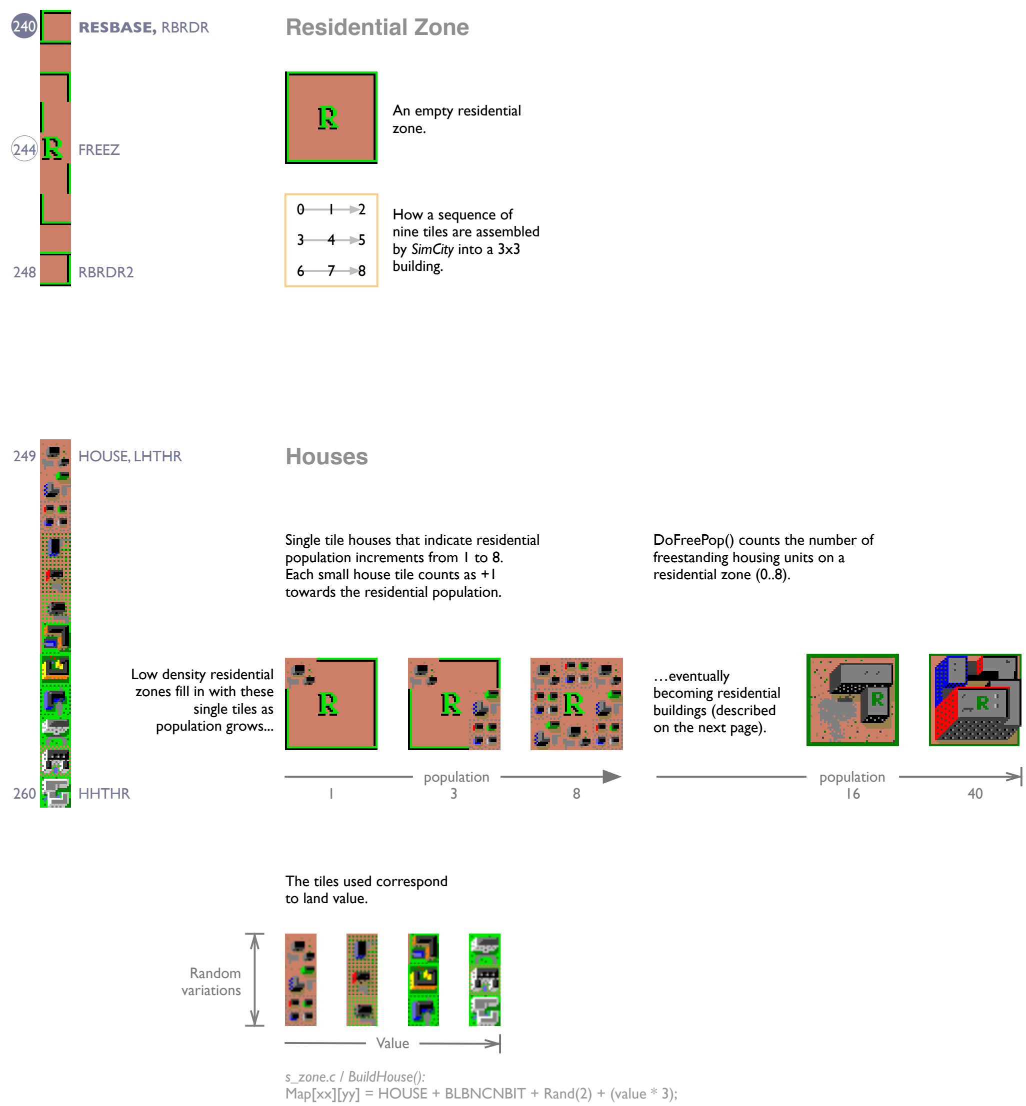


Figure A.20: Residential Characters (240–260)

Residential Buildings

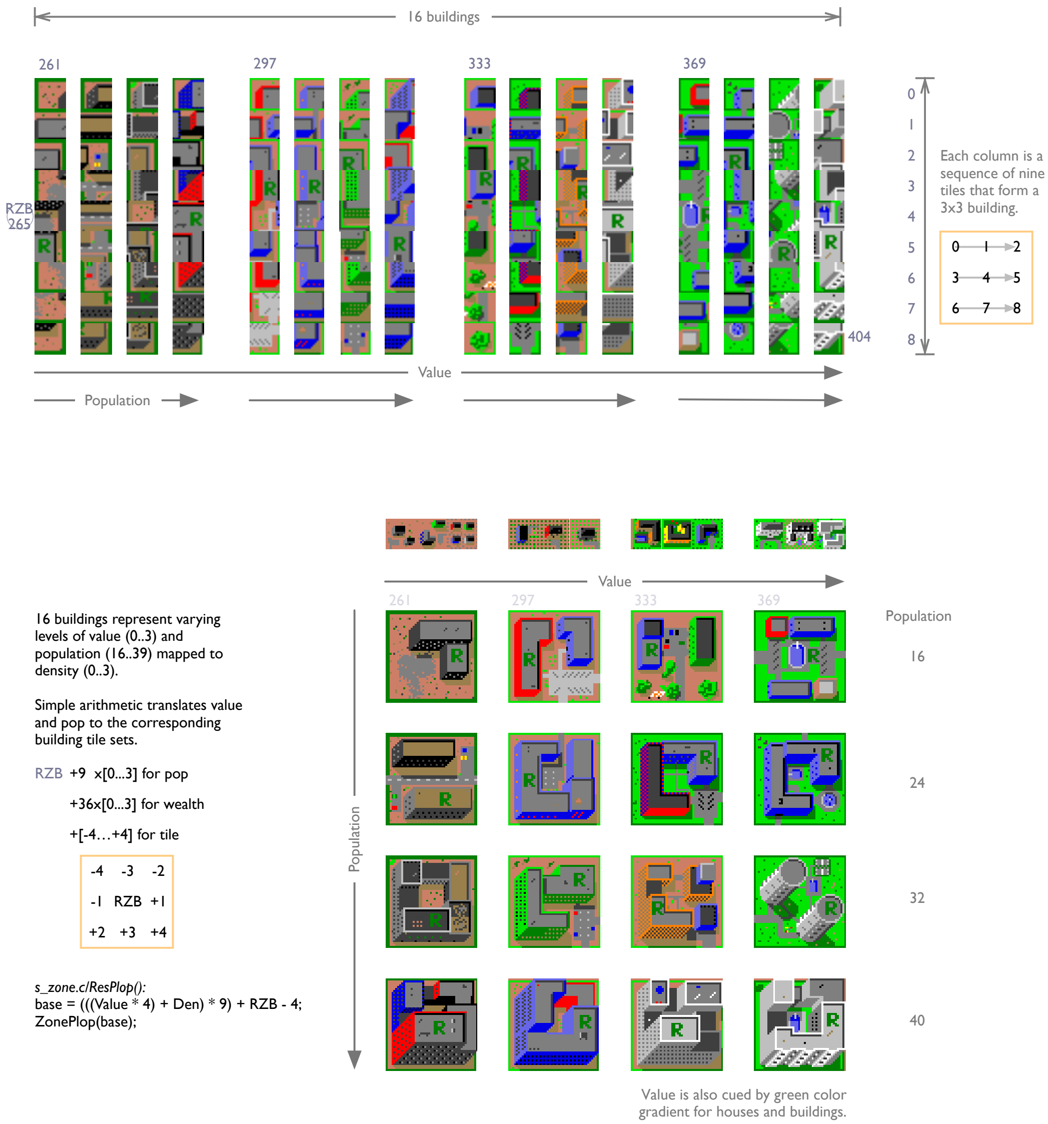
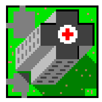
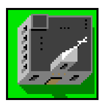


Figure A.21: Residential Characters (261–404)

Hospitals and Churches

405

409 HOSPITAL


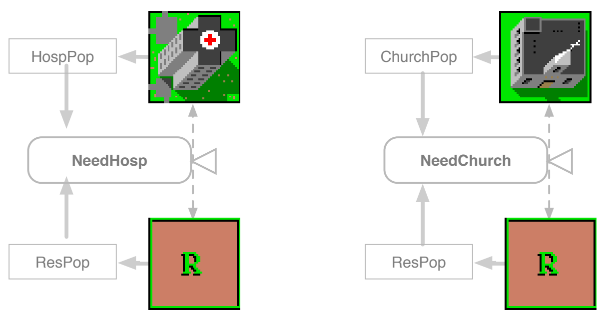

418 CHURCH

422

Cities sprout hospitals and churches in proportion to the city's residential population.

TakeCensus() updates the global variables NeedHosp and NeedChurch to reflect the per capita balance. A ratio of 128 residential citizens to a hospital and church is sought. -1, 1, 0 represent whether a hospital or church should be removed, added, or kept the same.

When DoResidential() attempts to grow an empty residential zone, there is a random chance a hospital or church will be developed if one is desired.

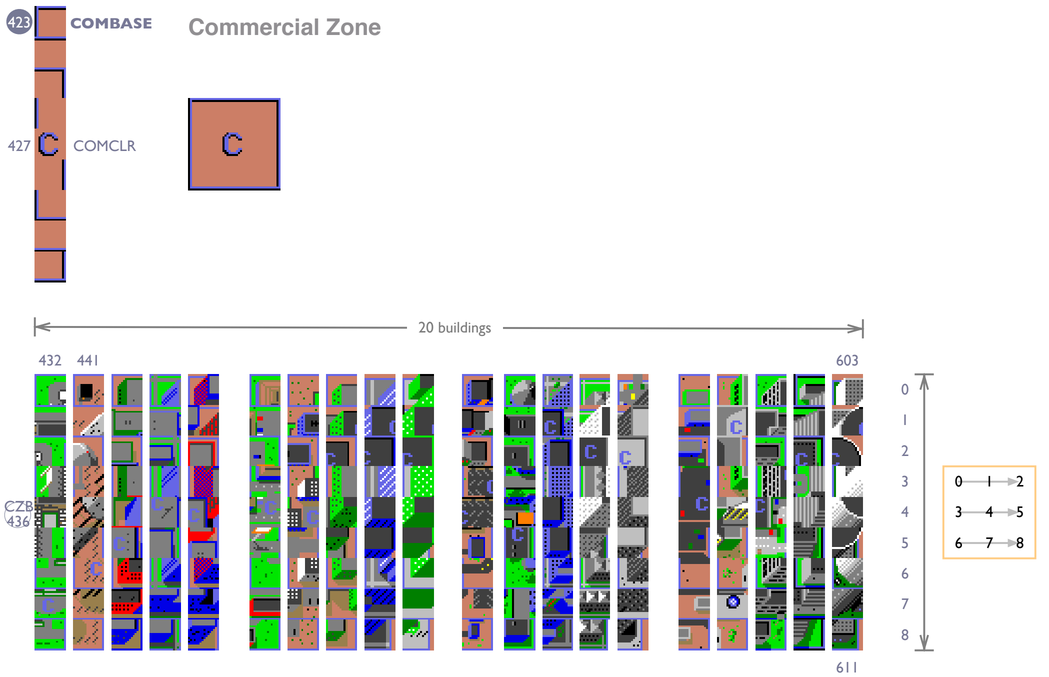
DoHospChurch() turns these back into empty residential zones if there are too many.



Hospital and church population regulation represented as a system dynamics style diagram.

A ratio of 128 residential citizens to a hospital and church is sought.

Figure A.22: Hospital and Church Characters (405–422)



20 buildings represent varying levels of value (0..3) and density (0..4).

Simple arithmetic translates value and population to the corresponding building tile sets.

s_zone.c/ComPlop():
base = (((Value * 5) + Den) * 9) + CZB - 4;

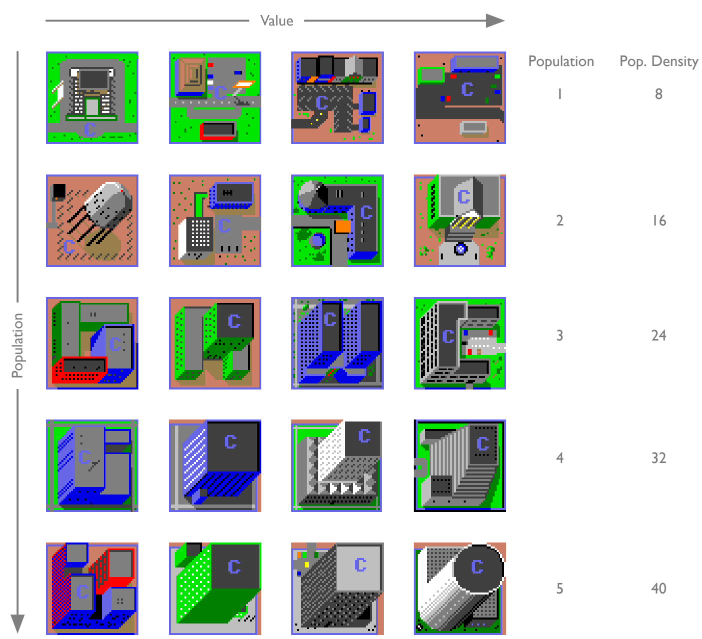
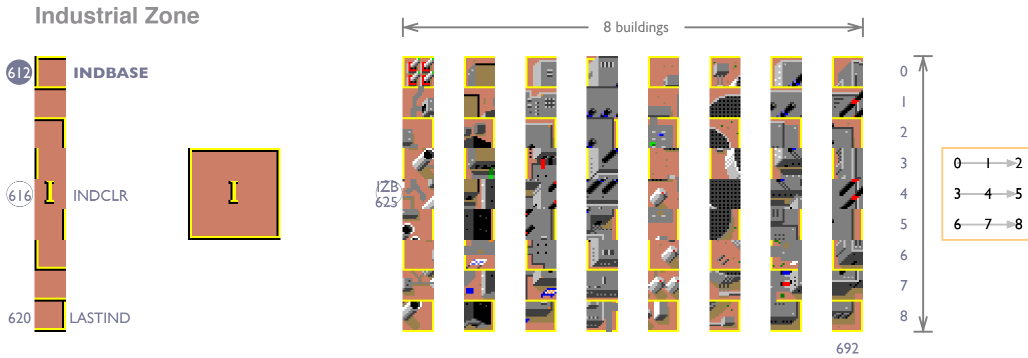


Figure A.23: Commercial Characters (423–611)



8 buildings represent varying population (0..4), with 2 random variations.

Simple arithmetic translates value and population to the corresponding building tile sets.

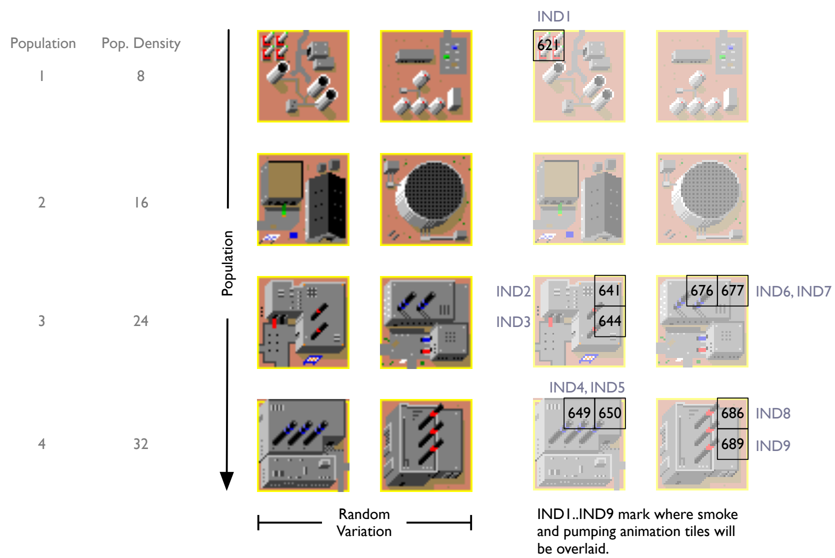
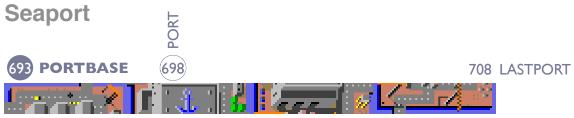
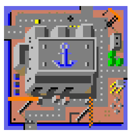


Figure A.24: Industrial Characters (612–692)

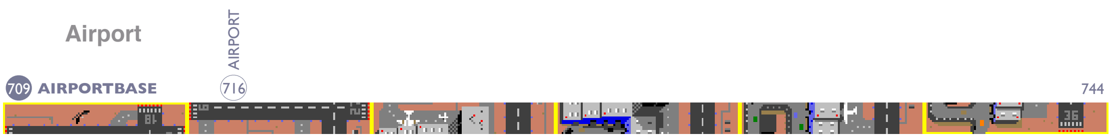
Seaport

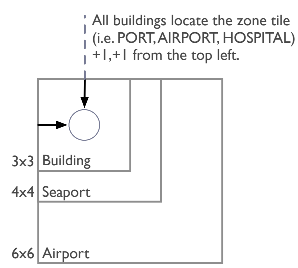




4x4

Airport





6x6

RADAR marks where the radar animation will be overlaid, which happens when the power is on.

Figure A.25: Seaport and Airport Characters (693–744)

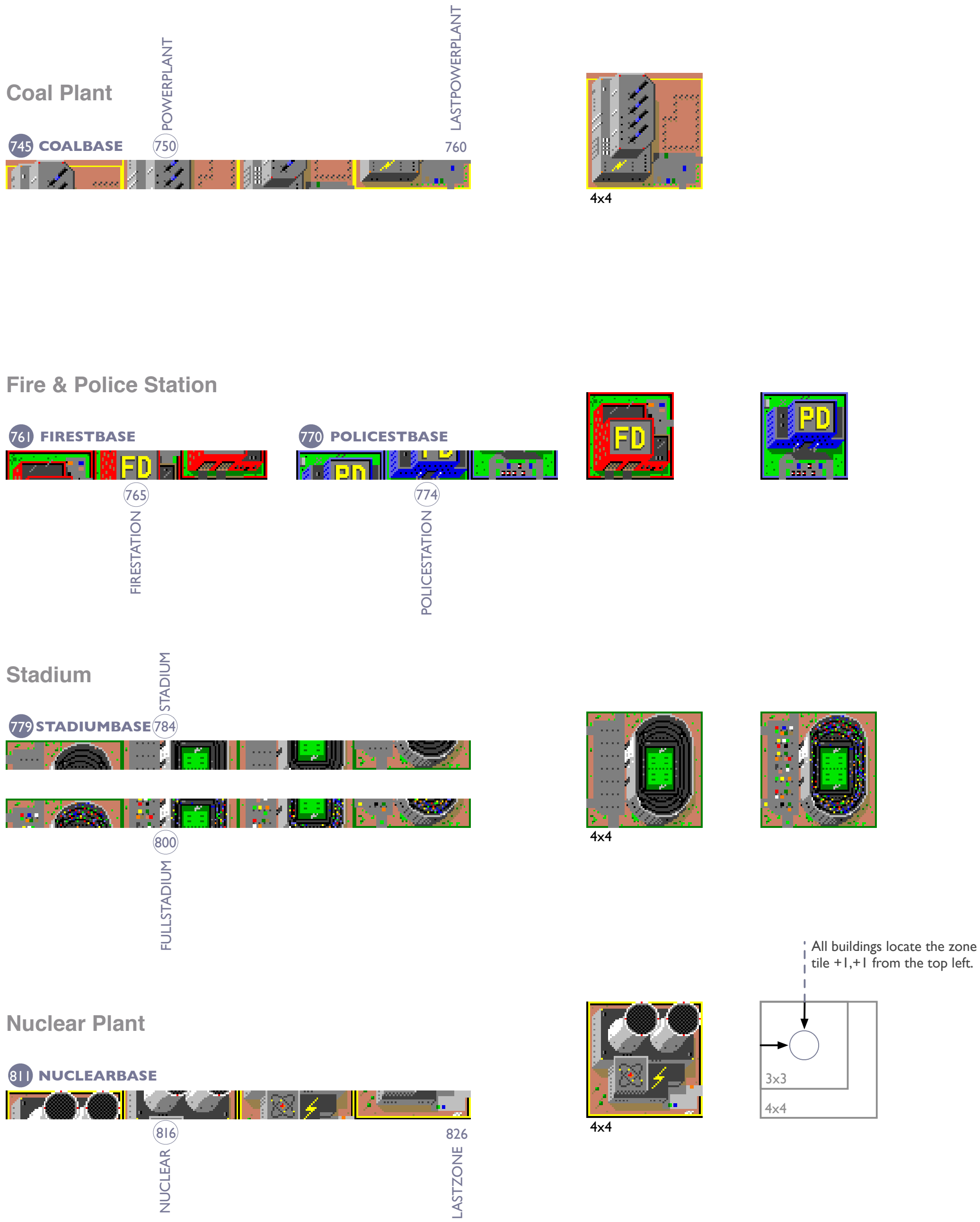
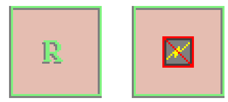


Figure A.26: Police, Fire, Power, and Stadium Characters (745–826)

Power Outage

827  LIGHTNINGBOLT



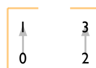
Blinking lightning bolt indicates a lack of power.

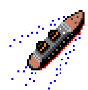
This blinking animation is not done through the tile character substitution technique, but is done while rendering the tiles to the screen.

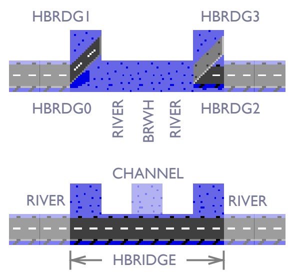
Draw Bridge—Horizontal

HBRDG0 … HBRDG3

828 



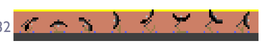




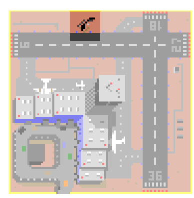
Bridges can open when boats are nearby, and close when they are far. BRWH bookmarks the open bridge.

s_sim.c/DoBridge() structures this logic as a toggle between two state configurations of a U shaped tile layout.

Airport Radar

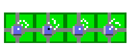
832 

RADAR0 … RADAR7



Airport radar spins when the airport has power.

Park Fountain

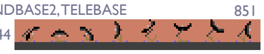
840 

FOUNTAIN

Placed parks have a 1 in 5 chance of being an animating fountain, and a 4 in 5 chance of being WOODS.

Telecommunications

INDBASE2, TELEBASE 851 TELELAST

844 

s_traf.c/FindPTele(void) /* look for telecommunication on edges of zone */

Added by Don Hopkins to OLPC *SimCity/Micropolis*.

Figure A.27: Animation Characters (827–851)

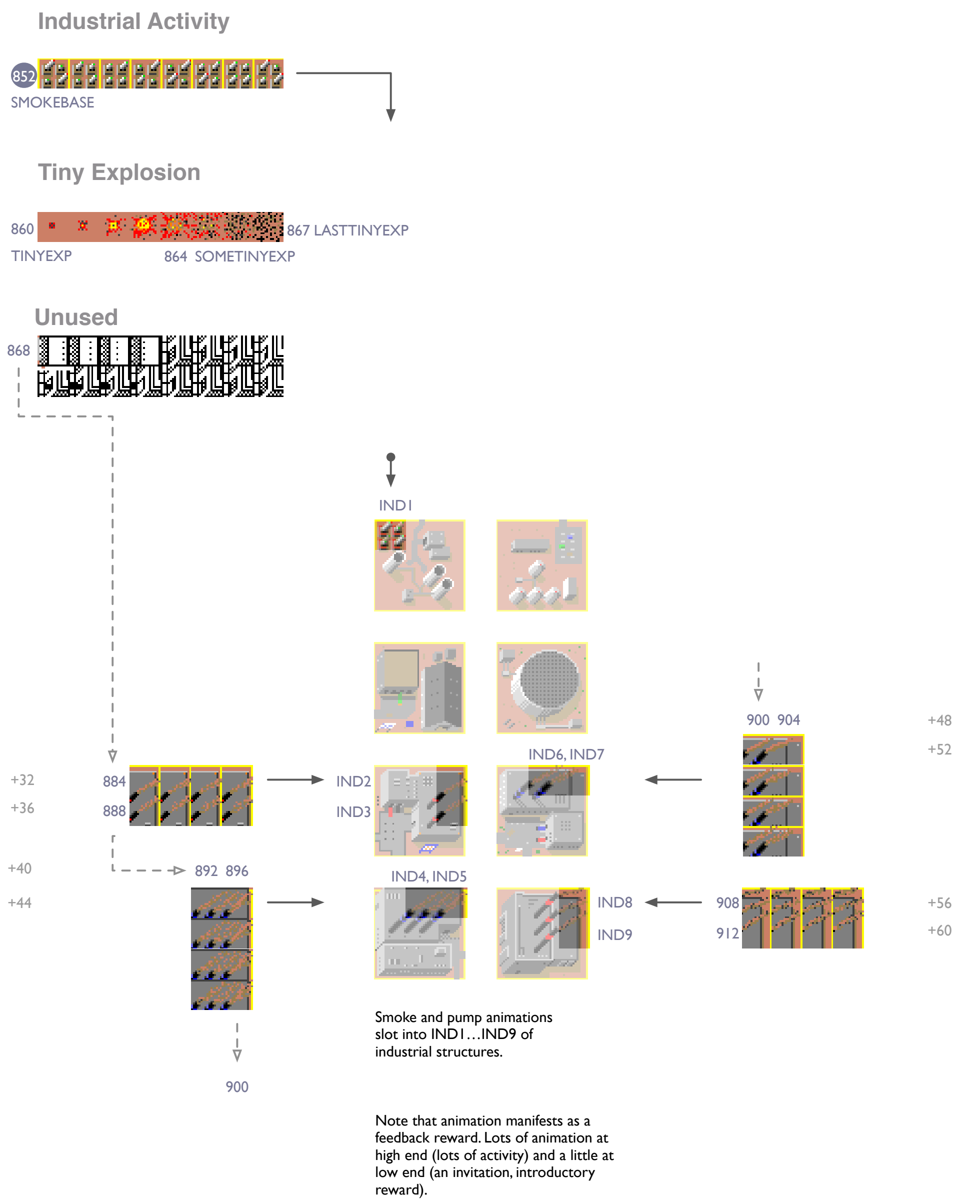
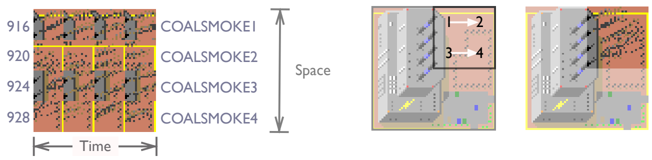
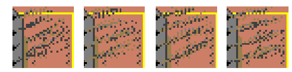


Figure A.28: Animation Characters (852–915)

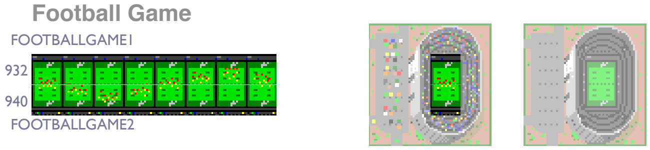
Coal Plant





Smoke for coal power plant.

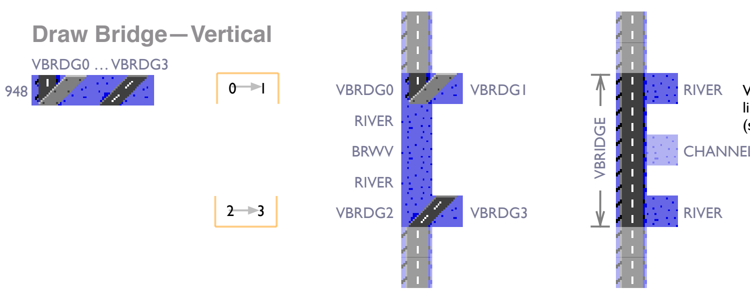
Football Game



Animation for football game, organized into parallel top and bottom sequences.

The stadium also has empty and full states, whose tiles are stored adjacent (779-810).

Draw Bridge—Vertical



Vertical draw bridge logic works like the horizontal draw bridge (see 828-).

Nuclear Plant

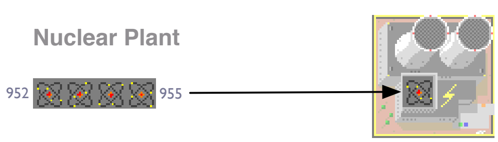


Figure A.29: Animation Characters (916–955)