UNIVERSITY OF CALIFORNIA

Los Angeles

Into the Black Box:
Using Data Mining of In-Game Actions to Draw Inferences from Educational Technology about Students' Math Knowledge

A dissertation submitted in partial satisfaction of the
requirements for the degree of Doctor of Philosophy
in Education

by

Deirdre Song Kerr

2014

ABSTRACT OF THE DISSERTATION

Into the Black Box:
Using Data Mining of In-Game Actions to Draw Inferences from Educational Technology about Students' Math Knowledge

by

Deirdre Song Kerr
Doctor of Philosophy in Education
University of California, Los Angeles, 2014
Professor Noreen M. Webb, Chair

Educational video games have the potential to be used as assessments of student understanding of complex concepts. However, the interpretation of the rich stream of complex data that results from the tracking of in-game actions is so difficult that it is one of the most serious blockades to the use of educational video games or simulations to assess student understanding, and there is currently no systematic approach to extracting relevant data from log files. This study attempts to determine whether data mining techniques can be used to extract information from log files that allows for the formation of testable hypotheses. The log files in this study come from an educational video game teaching students about the identification of fractions. The three data mining techniques used in this study were: cluster analysis, sequence mining, and classification.

Cluster analysis was used to examine the individual actions each student took in a given attempt to solve each game level. This led to the identification of valid solution strategies students used and mathematical errors and game-related errors students made as they tried to solve game levels. Sequence mining was used to examine the change in strategy use as each student moved through the game levels. This led to the identification of strategy sequences representing different paths students took to arrive at the correct solution. Classification was used to examine the change in the number of attempts each student required to solve the levels in each stage. This led to the identification of performance trajectories representing improvement, decline, or lack of change in performance over time.

To demonstrate the usefulness of the extracted information and provide initial evidence that the interpretation of the extracted information was valid, testable hypotheses from the results of each data mining technique were generated examining whether the grouping of students resulting from each of the three data mining techniques differed significantly on paper-and-pencil pretest scores, posttest scores, or the gain in scores between pretest and posttest. The groups resulting from each of the three data mining techniques differed significantly on pretest and posttest scores, with students in groups interpreted as representing lower performance demonstrating lower performance on the tests and students in groups interpreted as representing higher performance demonstrating higher performance on the tests. However, none of the three techniques led to the identification of groups of students that differed significantly on the gain in scores between pretest and posttest.

The dissertation of Deirdre Song Kerr is approved.

Eva L. Baker

Li Cai

Carlo Zaniolo

Noreen M. Webb, Committee Chair

University of California, Los Angeles

2014

This dissertation is dedicated to my grandfather, Richard Alexander Nelson (1921-2012), a proud graduate of West Orange High School, World War II veteran, and Bell Laboratories employee. He was the kindest man I've ever known, and I miss him greatly. Whenever I was unsure I'd be able do something, he always said, "Of course you will." Now, of course, I did.

It is also dedicated to my grandmother, Helen Nelson (née Magyar) (1923-2013), a proud graduate of West Orange High School, member of the Army Nursing Corps, and Knitting for Newborns volunteer, who never doubted.

Table of Contents

Acknowledgments

This dissertation would not have been possible without the support of many people. First and foremost, I would like to thank my supervisor Gregory Chung. Without him, none of this would have happened. Thanks for giving me the room to figure things out, for always supporting me, and for asking all the right questions.

I would like to thank my advisor, Noreen Webb, for sticking it out with me and for all her excellent advice. This dissertation is a better piece of work than it would ever have been without her. I would also like to thank the members of my committee: Eva Baker, Li Cai, and Carlo Zaniolo. They have proved invaluable on both personal and professional levels and I would not be the researcher I am without them.

Additional thanks go out to my colleagues, inside and outside CRESST. Particular thanks go out to Joanne Michiuye for making all my writing better and for being at least as neurotic about data as I am. Thanks also go out to Andre Rupp, Roy Levy, Kristen DiCerbo, Richard Almond, and Jan Plass for being interested in my work before I was sure there was any reason to be.

Finally, I wish to thank my family and friends. Particular thanks go out to Jesse for the repeated assurances that I am doing good work, Dawni and Chad for the much needed kitten exposure, Kat and Iggy for the occasional weekend away eating good food and playing video games, and Amber and A.C. and Zak for providing refuge on countless occasions.

Vita

PRIOR EDUCATION

The Evergreen State College	M.A.	Teachers of Native American Learners	2000
Carleton College	B.A.	English	1996

RESEARCH EXPERIENCE

2009-2013	National Center for Research on Evaluation, Standards, and Student Testing
2008-2009	LessonLab Research Institute

JOURNAL ARTICLES AND PUBLISHED PROCEEDINGS

Kerr, D., & Chung, G. K. W. K. (2013). Identifying learning trajectories in an educational video game. In R. Almond & O. Mengshoel (Eds.), *Proceedings of the 2013 UAI Application Workshops: Big Data Meet Complex Models and Models for Spatial, Temporal, and Network Data* (pp.20-28).

Kerr, D., & Chung, G. K. W. K. (2012). Identifying key features of student performance in educational video games and simulations through cluster analysis. *Journal of Educational Data Mining*, *4*, 144-182.

Kerr, D., & Chung, G. K. W. K. (2011). The mediation effect of in-game performance between prior knowledge and posttest score. In J. Matuga (Ed.), *Proceedings of the IASTED International Conference on Technology for Education (TE 2011)* (pp. 122-128). Anaheim, CA: ACTA Press.

SELECTED TECHNICAL REPORTS

Kerr, D., & Chung, G. K. W. K. (2013). *The effect of in-game errors on learning outcomes* (CRESST Report 835). Los Angeles, CA: University of California, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).

Kerr, D., & Chung, G. K. W. K. (2012). *Using cluster analysis to extend usability testing to instructional content* (CRESST Report 816). Los Angeles, CA: University of California, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).

SELECTED PRESENTATIONS

Kerr, D., & Chung, G. K. W. K. (2013, April). Using log data analysis to identify common misconceptions across games. In *Advances in Analysis of Process Data from Game-Based Assessment*. Coordinated session at the 2013 annual conference of the National Council on Measurement in Education, San Francisco, CA.

Kerr, D., & Chung, G. K. W. K. (2012, April). Using in-game performance to assess content knowledge. In A. Rupp (Chair), *Embedded Assessments in Digital Learning Environments*. Invited session at the 2012 annual conference of the National Council on Measurement in Education, Vancouver, BC.

CHAPTER 1: INTRODUCTION

Educational video games and simulations are often lauded as environments where students can be exposed to educational material in more interesting and motivating ways than are generally provided in the classroom (Garris, Ahlers, & Driskell, 2002; Lepper & Malone, 1987). However, the true benefits of educational video games and simulations go far beyond just the motivational. More importantly, they are environments where students can be exposed to authentic and interesting educational tasks (Edelson, Gordin, & Pea, 1999) and interact with and explore complex representations of serious academic content (Fisch, 2005; National Research Council, 2011).

Besides their ability to address complex content, educational video games and simulations also record every action taken by students as they play, rather than just the answers given. This means they can record the exact learning behavior of students (Romero & Ventura, 2007), allowing examination of thought processes that are often not captured in students' verbal explanations (Bejar, 1984). Problem-solving strategies and mistakes that can be impossible to capture on a paper-and-pencil test are easily recorded in educational video games and simulations (Merceron & Yacef, 2004; Quellmalz & Pellegrino, 2009; Rahkila & Karjalainen, 1999) in an unobtrusive manner (Kim et al., 2008; Mostow et al., 2011) that is both feasible and cost-effective (Quellmalz & Haertel, 2004).

This information can be used to provide detailed measures of the extent to which players have mastered specific learning goals (National Science and Technology Council, 2011) or to support diagnostic claims about students' learning processes (Leighton & Gierl, 2007). Educational video games and simulations provide the ability for direct assessments of complex tasks (Linn, Baker, & Dunbar, 1991) in a more authentic manner than standard multiple choice tests (Su, Hung, Hwang, & Lin, 2010), and because they are less text-intensive, they may be more valid assessments of English language learners, students with disabilities, and students with low reading ability (Kopriva, Gabel, & Bauman, 2009).

Educational games and simulations also have the potential to be used to identify the strengths and weaknesses of individual students (Mehrens, 1992), to provide individualized feedback (Brown, Hinze, & Pellegrino, 2008), to guide instruction that is optimal for each student (Bejar, 1984; Clark, Nelson, Sengupta, & D'Angelo, 2009; Radatz, 1979), or to allow students to track their own progress (Rahkila & Karjalainen, 1999). Additionally, they could be used to improve classroom instruction (Merceron & Yacef, 2004) by allowing for the identification of common errors or determining the relative effectiveness of different pedagogical strategies for different types of students (Romero & Ventura, 2007).

Therefore, it is not surprising that the government called for the research and development of educational video games and simulations to assess the complex skills identified in state and national standards (U.S. Department of Education, 2010) and the use of educational video games and simulations to determine the effectiveness of different instructional practices (U.S. Department of Education, 2012). Unfortunately, it is also not surprising that educational video games and simulations cannot currently be used as stand-alone assessments of student performance.

In standard testing formats, evidence of student performance takes the form of the answers students have given to a series of problems. In educational video games and simulations, on the other hand, this evidence takes the form of the specific actions students have taken while trying to solve each problem (e.g., "the student changed a fraction from 1/2 to 2/4"). Because the interpretation of these actions often depends on what else the student does while attempting to solve the problem, the relationship between each specific action and overall student performance is not immediately clear.

This means the process of identifying evidence of student performance in educational video games and simulations is incredibly complex due to the sheer number of observable actions and the variety of potential relationships each action could have with student performance (Frezzo, Behrens, Mislevy, West, & DiCerbo, 2009). Extracting relevant features from the noise in the data is crucial in such environments to make analysis computationally tractable (Masip, Minguillon, & Mor, 2011).

In educational video games or simulations, relevant features of student performance must be extracted from the log files that are automatically generated by the game or simulation as students play (Kim et al., 2008). Though log data are more comprehensive and more detailed than most other forms of assessment data, analyzing log data presents a number of challenging problems (Garcia, Romero, Ventura, de Castro, & Calders, 2011; Mostow et al., 2011) inherent when examining exact learning behaviors in highly unstructured environments (Amershi & Conati, 2011). These environments typically include thousands of pieces of information for each student (Romero, Gonzalez, Ventura, del Jesus, & Herrera, 2009) with no known theory to help identify which information is salient (National Research Council, 2011). Most of the difficulties associated with analyzing data of this type come from the sheer amount of information present. Log data consist of prohibitively large quantities of information (Romero, Gonzalez, et al., 2009), wherein a single student can generate over 3,000 actions in half an hour of game play (Chung et al., 2010).

In addition to the size of the data, the specific information stored in the log files is not always easy to interpret (Romero & Ventura, 2007) as the responses of individual students are highly context dependent (Rupp, Gushta, Mislevy, & Shaffer, 2010) and it can be difficult to picture how student knowledge, learning, or misconceptions manifest themselves at the level of a specific action taken by the student in the course of the game. Additionally, it can be difficult to determine which actions represent key features of student performance given that log files are generally designed to capture all student actions relevant to game play, and it is not until after analysis that one would know which actions were relevant to learning.

Due to these difficulties, there is currently no systematic approach to extracting relevant data from log files (Muehlenbrock, 2005) and the field is still in its infancy (Romero, Ventura, Pechenizkiy, & Baker, 2011; Spector & Ross, 2008). Despite the increasing availability of extensive fine-grained longitudinal information derived from educational technology (Koedinger et al., 2011), the interpretation of the rich stream of complex data that results from the tracking of in-game actions is one of the most serious bottlenecks facing researchers examining educational video games and simulations today (Mislevy, Almond, & Lukas, 2004). The task is so difficult that a government task force recently determined that the single biggest challenge to embedding assessment in educational games and simulations is determining methods of drawing inferences from log data (National Research Council, 2011).

Purpose

The purpose of this study was to develop techniques to extract salient information from log data from educational video games so that inferences about student performance can be made. Three separate data mining techniques were used: cluster analysis, sequence mining, and classification. The usefulness of these techniques was examined through the following research questions:

1. Can educational data mining techniques be used to extract information from log files from an educational video game that allows for the formation of testable hypotheses?
    a. Can cluster analysis be used to extract information leading to testable hypotheses?
    b. Can sequence mining be used to extract information leading to testable hypotheses?
    c. Can classification be used to extract information leading to testable hypotheses?

Additionally, testable hypotheses from the results of each data mining technique were generated and examined in order to (a) demonstrate the usefulness of the extracted information and (b) create initial evidence that the interpretation of the extracted information was valid. These hypotheses focus on the relationship between the information extracted from game log data using each data mining technique and student performance on paper-and-pencil measures of the same content.

CHAPTER 2: LITERATURE REVIEW

The Game as a Black Box

In the majority of studies examining the impact of educational video games or simulations, the game itself is a black box. The effects of the games are generally measured by differences on posttest scores or survey responses, and no information about in-game student performance is measured or analyzed (Tobias, Fletcher, Dai, & Wind, 2011).

Studies of the impact of educational video games or simulations on student perception look for differences between students who played and students who did not on surveys administered after game play. While students playing the *Incredible Machine* game were more bored and frustrated than students using the *Aplusix* math tutoring software (Rodrigo et al., 2008) and Hoffman, Pack, Zhou, and Turkay (2009) found no change in math motivation after game play, most studies found positive effects of game play on perception.

These studies found that students thought an electronics simulation was more interesting than a workbook (Ronen & Eliahu, 1999), were more interested in the topic after playing a game about chemical/biological/radiological defense (Ricci, Salas, & Cannon-Bowers, 1996) or math (Ke, 2008) than after traditional instruction, and enjoyed playing *SimCity* more than reading about city planning (Betz, 1995). Additionally, students playing games or simulations gained more self-efficacy than students who received traditional instruction (Tompson & Dass, 2000) and reported higher levels of engagement (Coller & Shernoff, 2009). No in-game measures of interest, engagement, or self-efficacy were reported in any of these studies.

While a number of other studies examined learning in games, those studies also largely used differences in posttest scores between students who played and students who did not to measure learning rather than using in-game measures. Though Wiebe and Martin (1994) found that playing *Carmen Sandiego*, a geography game, did not increase scores more than standard instruction, Whitehill and McDonald (1993) found no difference for Navy students playing a game about circuits, and Hart and Battiste (1992) found no transfer for students playing the flight simulators *Space Fortress* or *Apache Strike*, most studies found positive effects for games and simulations.

Positive posttest differences were found in a number of studies of applied knowledge. Students who played *ReMission*, a cancer education game, took more of their prescribed medication (Kato, Cole, Bradlyn, & Pollock, 2008), students who played a virtual putting game improved their golf performance more than students who did not (Ferry & Ponserre, 2001), and students who played a flight simulation improved pilot ratings more than students who did not (Gopher, Weil, & Bareket, 1994). Studies with preschool and kindergarten children found positive effects for games teaching classification skills (Sung, Chang, & Lee, 2008), rhyming and grapheme knowledge (Segers & Verhoeven, 2005), spelling and reading (Din & Caleo, 2001), and math (Laffey, Espinosa, Moore, & Lodree, 2003).

Students who played a *Quest Atlantis* unit on writing produced better writing than the control group (Warren, Dondlinger, & Barab, 2008), students playing a game about the food pyramid learned more about nutrition (Serrano & Anderson, 2004), and students playing a simulation of cell theory performed better on a biology test (Wekesa, Kiboss, & Ndirangu, 2006). Similar effects were found for games on electronics (Parchman, Ellis, Christinaz, & Vogel, 2000), computer memory principles (Papastergiou, 2009), economics (Gremmen & Potters, 1997), math (Kebritchi, Hirumi, & Bai, 2010), physics (Ravenscroft & Matheson, 2002), and business (Blunt, 2008).

Other studies did not have control groups that did not play the game. These studies gave students a pretest before the game and a posttest after the game, and inferred learning if there was a significant change between pretest and posttest scores. Chen and O'Neil (2008) used this process to determine that students who played a game about diabetes improved their self-care skills, and Wilson, Revki, Cohen, Cohen, and Dehaenel (2006) used this process to determine that students who played *The Number Race*, a game designed to teach number sense to struggling students, improved their mathematical skills.

Even studies of the effects of modifications to the game itself rarely used in-game measures of those effects, relying instead on posttest or survey measures collected after game play. For example, Baylor (2002) found that the presence of a constructivist agent in the game was related to higher self-reported planning activity, but did not examine whether it led to an increase in planning activity in the game. Similarly, other studies found that having an in-game agent give advice reduced self-reported anxiety (Van Eck, 2006), providing players with constantly updated information reduced self-reported mental workload (Hsu, Wen, & Wu, 2007), and personalizing the game led to greater self-reported engagement (Cordova & Lepper, 1996). Studies also found that personalizing the game led to more creative solutions on posttest measures (Moreno & Mayer, 2000), and that providing specific prompts rather than general prompts (Lee & Chen, 2009), reducing complexity (Lee, Plass, & Homer, 2006), and providing a paper-and-pencil pictorial instructional sheet (Mayer, Mautone, &Prothero, 2002) improved performance on posttest measures. However, none of these studies examined in-game performance measures in addition to or instead of the paper-and-pencil posttest measures. The lack of examination of in-game measures in these studies, even when the research questions would seem to dictate it, is largely due to the fact that such measures are extraordinarily difficult to extract from game data.

Basic Log Data Analyses

Due to the difficulty involved in analyzing log data of students' in-game performance (Frezzo et al., 2009; Mislevy et al., 2004), some researchers have resorted to hand-coding information from log files from video games. Trained human raters have been used to extract purposeful sets of actions from game logs (Avouris, Komis, Fiotakis, Margaritis, & Voyiatzaki, 2005) and logs of eye-tracking data (Conati & Merten, 2007) and to identify student errors in log files from an introductory programming environment (Vee, Meyer, & Mannock, 2006). One study even had the teacher play the role of a game character to score student responses and provide live feedback to the students (Hickey, Ingram-Goble, & Jameson, 2009). Amershi and Conati (2011) examined behavior patterns in an exploratory learning environment and categorized students as high learners, thoughtful low learners, or unthoughtful low learners by hand.

A number of other studies used basic aggregate information from the log data from online learning environments, without examining the specific actions taken by students. The aggregate information extracted from the log data were the number of activities completed by the student and the amount of time spent in the activity. The number of activities completed in the online learning environments *Moodle*(Romero, Gonzalez, et al., 2009) and *ActiveMath* (Scheuer, Muhlenbrock, & Melis, 2007) have been used to predict student grades. The time spent in each activity in an online learning environment has been used to detect unusual learning behavior (Ueno & Nagaoka, 2002). Combinations of the total time spent in the online environment and the number of activities successfully completed have been used to predict student success (Muehlenbrock, 2005) and report student progress (Rahkila & Karjalainen, 1999).

Studies examining log data from educational video games or simulations generally restricted themselves to basic summarizations or averages of pre-coded events such as deaths or resets. One such study counted the number of hints students requested and the number of failures they experienced to categorize students as hint-driven learners or failure-driven learners (Yudelson et al., 2006). A second study counted the number of deaths in each area of the game to determine which areas needed improvement (Kim et al., 2008). A third study counted the amount of money earned in a management simulation to determine effective or ineffective players (Ramnarayan, Strohschneider, & Schaub, 1997). A fourth study counted the number of errors, the average economy of motion, and the time it took students to finish a laparoscopic surgery simulation to determine performance (Gallagher, Lederman, McGlade, Satava, & Smith, 2004).

Cluster Analysis

Because the log files generated by educational video games or simulations are too large to analyze manually (Garcia et al., 2011), data mining techniques must be used to automatically identify and describe meaningful patterns despite the noise surrounding them (Bonchi et al., 2001; Frawley, Piateski-Shapiro, & Matheus, 1992). This automatic extraction of implicit, interesting patterns from large data sets can lead to the discovery of new knowledge about how students solve problems in order to identify interesting or unexpected learning patterns (Romero, Ventura, et al., 2009) and can allow questions to be addressed that were not previously feasible to answer (Romero et al., 2011).

Cluster analysis is one of the most commonly used educational data mining techniques (Castro, Vellido, Nebot, & Mugica, 2007). It is a density estimation technique for identifying patterns within user actions reflecting differences in underlying attitudes, thought processes, or behaviors (Berkhin, 2006; Romero, Ventura, et al., 2009) through the analysis of either general or sequential correlations (Bonchi et al., 2001). Because cluster analysis is driven solely by the available data and is ideal in instances in which little prior information is available (Jain, Murty, & Flynn, 1999), it is particularly appropriate for the analysis of log data. Cluster analysis can be used to identify the latent dimensionality of a data set (Roussos, Stout, & Marden, 1998) and to compress the data set into a manageable number of variables that are nontrivial, implicit, previously unknown, and potentially useful (Frawley et al., 1992; Hand, Mannilla, & Smyth, 2001; Vogt & Nagel, 1992). It has been used regularly in such fields as engineering, chemistry, physics, astronomy, law enforcement, and marketing to identify key features of large data sets (Frawley et al., 1992).

Cluster analysis partitions entities into groups on the basis of a matrix of inter-object similarities (James & McCulloch, 1990) by minimizing within-group distances compared to between-group distances so that entities classified as being in the same group are more similar to each other than they are to actions in other groups (Huang, 1998). Using these similarities, cluster analysis algorithms can identify the latent grouping structure of the data (Roussos et al., 1998; Vellido, Castro, & Nebot, 2011) and perform the necessary pattern reduction and simplification so the patterns present in large data sets can be detected (Vogt & Nagel, 1992). In educational settings, cluster analysis is most often used to identify sets of test items or types of learners (Castro et al., 2007; Vellido et al., 2011). In these studies, each cluster represents either a group of users with similar behavior patterns or a group of items with similar requirements (Mobasher, Dai, Luo, Sun, & Zhu, 2000). Once the clusters of items or students have been identified, logistic regression is used to identify the variables that differ between groups (Rodrigo, Anglo, Sugay, & Baker, 2008).

Because educational data mining has its roots in the analysis of web logs (records of web page visits), a number of cluster analysis studies focused on standard educational data presented in an online environment. For example, one study clustered readers of an online newsletter by the pages of the newsletter they clicked on in order to recommend relevant articles (Mobasher et al., 2000). Chen and Chen (2007) proposed clustering students in a virtual university based on their use of online resources in order to create a recommendation service for their digital library. Two other studies clustered students based on their page views in virtual universities to identify different types of library usage (Ferran, Casadesus, Krakowska, & Minguillon, 2007; Masip et al., 2011).

Roussos et al. (1998) used a simulation study to demonstrate that cluster analysis could be used to detect multidimensionality of test items. Properties of test items in the *Bridge to Algebra* cognitive tutor were clustered to form groups with different proficiency requirements (Pavlik, Cen, Wu, & Koedinger, 2008), and keywords in an online FAQ system were clustered to form a concept hierarchy (Chiu, Pan, & Chang, 2008). Sison, Numao, and Shimura (2000) clustered answers in an online programming environment to form an error hierarchy, and Hunt and Madhyastha (2005) clustered responses to items on the WASL state standardized exam into different types of misconceptions.

However, most cluster analysis studies grouped students rather than learning material. Students in a state university were clustered to identify different profiles of African American college students (Rowley, 2000). Students in the *Ars Digita* online learning environment were clustered into six different types of collaborators (Talavera & Gaudioso, 2004), students in an intelligent tutoring system for algebra were clustered into collaborative or solitary work patterns (Rodrigo, Anglo, et al., 2008), and students in an online distance learning university were clustered into low, medium, or high collaborators (Anaya & Boticario, 2009). Students playing *Prime Climb*, a middle school factorization game, were clustered based on differences in their biometric data to find different types of affective expression (Amershi, Conati, & Maclaren, 2006). Students playing *Alien Rescue*, a middle school science game, were clustered to identify different patterns of tool usage (Liu & Bera, 2005). Students playing *Magical Surprise*, an online math game, were clustered into different risk taking categories (Araya et al., 2011).

Only two studies used fuzzy cluster analysis rather than hard cluster analysis. One used fuzzy cluster analysis to group students into either homogeneous or heterogeneous clusters for in-class group formation (Christodoulopoulos & Papanikolaou, 2007). The other grouped students into different personality types based on survey responses in an online university (Tian, Wang, Zheng, & Zheng, 2008). Only one study used feature cluster analysis to cluster actions instead of students: Hershkovitz and Nachmias (2011) clustered "learnograms" such as time on task to identify different aspects of motivation. No educational studies used fuzzy feature cluster analysis.

This study uses fuzzy feature cluster analysis to identify the different strategies students use to complete levels in an educational video game.

Sequence Mining

Cluster analysis ignores certain salient aspects of the log data, such as timing and order (Perera, Kay, Koprinska, Yacef, & Zaïane, 2009). If timing or order is important to the analysis, then sequence mining must be used. Sequence mining is a technique for identifying patterns within user actions that frequently occur in the same order (Srikant & Agrawal, 1995).

Sequence mining can identify previously unknown misconceptions (Antunes, 2008) or patterns of behavior that can be used to support or improve decision making in education (Zhou, Xu, Nesbit, & Winne, 2011). However, it produces a prohibitively large number of sequences (Antunes, 2008) that are often redundant or difficult to understand (Garcia et al., 2011). Therefore, even though order is very important in educational data because it often indicates increasing difficulty (Cummins, Yacef, & Koprinska, 2006), time-based analyses of educational data are rare (Hadwin, Nesbit, Code, Jamieson-Noel, & Winne, 2007).

The most common use of sequence mining in education is to identify common access patterns in online learning environments. Pahl and Donnellan (2003) used sequence mining to identify common patterns of web navigation. Antunes (2008) found sequences of courses taken in a computer science program. Cummins et al. (2006) and Ksristofic (2005) found common sequences of materials accessed to provide recommendation services in online learning resources. This process was expanded by Shen and Shen (2004) to recommend materials related to the targeted content if students knew the targeted content, but to recommend materials related to the prerequisite if students did not know the targeted content.

Other studies identified sequences in online collaborative learning data and associated those sequences with group performance (Perera et al., 2009), identified sequences indicative of good teamwork (Kay, Koprinska, & Yacef, 2011; Kay, Maisonneuve, Yacef, & Zaïane, 2006), or detected action patterns reflective of different learning styles (Kelly & Tangney, 2005). Additional studies used the results of sequence mining to identify previously unknown student conceptualizations of physics (Madhyastha & Hunt, 2009) or determine when students deviated from desired behavior (McLaren, Koedinger, Schneider, Harrer, & Bollen, 2004).

Robinet, Bisson, Gordon, and Lemaire (2007) used sequences of student actions to assign students to predesignated mental models. Chika, Azzi, Hewitt, and Stocker (2009) used action sequences to model students' lab skills. Buckley, Gobert, and Horwitz (2006) used sequence mining to identify four groups of students (correct-systematic, correct-haphazard, incorrect-systematic, and incorrect-haphazard) that were indicative of student performance.

Only two studies used both cluster analysis and sequence mining. One study clustered students in an online learning environment into groups and then ran sequence mining to identify common sequences that could be used to provide recommendations in the system (Romero, Ventura, Zafra, & de Bra, 2009). The other study clustered only the good students and ran sequence mining to find frequently occurring learner patterns for each group of good students (Su et al., 2006).

This study uses sequence mining to identify changes in student strategies over time.

Classification

Classification is a supervised form of cluster analysis (Romero & Ventura, 2007) that predicts group membership (Ayala, Dominguez, & Medel, 2009) based on inherent characteristics of individual group members (Romero, Ventura, Espejo, & Hervas, 2008). Classification matches new members to predefined groups by testing the new members' similarity to a training set consisting of previously categorized members of each group (Tanner & Toivonen, 2010).

Classification is one of the most frequently studied machine learning techniques (Romero et al., 2008) and is considered to be one of the most useful data mining techniques for educational purposes (El Den, Moustafa, Harb, & Emara, 2013; Kotsiantis, Patriarcheas, & Xenos, 2010). However, classification algorithms are often supplanted by Bayesian networks or decision trees in educational research (Ayala et al., 2009), even though classification algorithms do not require a model because they mine information about group membership directly from known group members and are well suited to the noisy or incomplete data often found in educational environments because they do not rely on any assumptions about prior probabilities (Tanner & Toivonen, 2010).

Classification algorithms are most commonly used in educational research to predict student performance (El Den et al., 2013). For example, Kotsiantis, et al. (2010) used student assessment data from the prior year to predict current students' grades, while Romero et al. (2008) used a subset of current students as a training set to classify the remaining students into performance categories (excellent, good, or poor). Minaei-Bidgoli, Kashy, Kortemeyer, and Punch (2003) not only classified students into performance categories (high, medium, or low) but also used classification to predict whether students would pass or fail the course, similar to how Tanner and Toivonen (2010) used classification to predict student dropouts and failures.

Few studies used classification algorithms for purposes other than predicting performance. Shih and Lee (2001) used classification to group students in an online learning system so that they could be presented with appropriate materials based on their similarity to other students who had benefited from the material. Additionally, Baker, Corbett, and Koedinger (2004) used classification algorithms to group each action in an intelligent tutoring system as being either an example of gaming behavior or a valid action in the system so that the percentage of time each student spent gaming the system could be computed.

This study uses the *k* nearest neighbor algorithm to classify graphs of students' performance over time into different performance trajectory types, using a hand-coded subset of the students as a training set.

Validating In-Game Measures

Because of the difficulty inherent in extracting meaningful measures of student performance from in-game actions, there are few studies examining the validity of the extracted in-game measures. However, now that a body of work has been published regarding the potential of games as assessments, Gee (2011) has called for robust empirical studies testing specific hypotheses about the validity of educational games and simulations as mediums of learning and assessment. While thus far in-game measures have tended to be motivational rather than indicative of student understanding (Tobias et al., 2011), the term “stealth assessment” has been coined to refer to the ability of games and simulations to assess students’ abilities by examining the actions they perform while attempting to solve problems in these environments (Shute, 2011).

Despite calls for more robust studies, the most common method of validating games is by consulting experts to determine whether the game has face validity. For example, Thompson and Irvine (2011) interviewed content experts about the face validity of *CyberCIEGE*, a game teaching introductory computer security, and found that experts believed they could determine students’ level of knowledge of computer security from the amount of time the students spent on each problem in the game and whether or not they successfully completed the problems. Sherif and Mekkawi (2010) ran a similar study about *The Excavation Game,* a game teaching project management aspects of building construction, and found that both students and experts believed the game addressed important aspects of the curriculum.

Another way to validate games is to examine the correlation between in-game measures and other assessments or outcomes thought to measure the same construct. Though this process depends on the quality of the assessment the game is being compared to, the strength of the correlation between measures is considered to be a measure of the construct validity of the game (Cook & Beckman, 2006). This method of validating games is particularly common in medical education. For example, Hislop et al. (2006) examined the correlations of rater-identified measures of skill in an endovascular simulator with formal levels of training and prior experience, and between the time it took students to complete the simulation and their levels of training and prior experience, and found significant correlations for both measures. Additionally, Johnsen, Raij, Stevens, Lind, and Lok (2007) found significant correlations between interview skills with a virtual human in a medical simulation and interview skills with a trained actor. Outside medical education, Delacruz, Chung, and Baker (2010) found that paper-and-pencil pretest scores were predictive of last level reached in a game addressing middle school mathematics concepts (an earlier version of the game used in this study) and that last level reached in the game was predictive of paper-and-pencil posttest scores.

This study examines the relationship between the in-game measures of student performance extracted by all three data mining techniques and paper-and-pencil pretest and posttest scores.

CHAPTER 3: METHODS

Designing *Save Patch*

The educational video game used in this study is *Save Patch*, a game designed by the National Center for Research on Evaluation, Standards, and Student Testing (CRESST) at the University of California, Los Angeles, and the Game Innovation Lab at the University of Southern California. The development of *Save Patch* was driven by the findings of the National Mathematics Advisory Panel (2008) that fluency with fractions is critical to performance in algebra, which is, in turn, of central importance to performance and participation in science, technology, engineering, and math courses and careers (Malcom, Chubin, & Jesse, 2004). Additionally, the understanding of fractions is one of the most difficult mathematical concepts students learn before algebra (National Council of Teachers of Mathematics, 2000; Siebert & Gaskin, 2006) and misconceptions about the meaning of fractions are not only very common but are also associated with subsequent difficulty understanding and applying advanced mathematical concepts (Carpenter, Fennema, Franke, Levi, & Empson, 2000; McNeil & Alibali, 2005).

Once fractions concepts were identified as the subject area for the game, the most important concepts involved in fractions knowledge were analyzed and distilled into a set of knowledge specifications delineating precisely what students were expected to learn in the game (Vendlinski, Delacruz, Buschang, Chung, & Baker, 2010). The four main concepts to be addressed in the game included: the meaning of the unit, the meaning of addition as applied to fractions, the meaning of the denominator, and the meaning of the numerator. Each of these concepts was broken down into further specifications of what understanding of that concept would entail, as seen in Table 1.

Table 1

Knowledge Specifications for *Save Patch*

1.0 Does the student understand the meaning and importance of the whole unit?

- 1.1 The size of a rational number is relative to how the whole unit is defined.
- 1.2 In mathematics, a whole unit is understood to be of some quantity.
- 1.3 The whole unit can be represented as an interval on the number line.

2.0 Does the student understand the meaning of addition as applied to fractions?

- 2.1 To add quantities, the units or parts of units must be identical.
- 2.2 Identical units can be added to create a single numerical sum.
- 2.3 Dissimilar quantities cannot be represented as a single sum.

3.0 Does the student understand the meaning of the denominator in a fraction?

- 3.1 The denominator of a fraction represents the number of identical parts in a whole unit.
- 3.2 As the denominator gets larger, the size of each fractional part gets smaller.
- 3.3 As the fractional part size gets smaller, the number of pieces in the whole gets larger.

4.0 Does the student understand the meaning of the numerator in a fraction?

- 4.1 The numerator of a fraction represents the number of identical parts that are combined.
- 4.2 If the numerator is smaller than the denominator, the fraction represents less than a whole.
- 4.3 If the numerator equals the denominator, the fraction represents a whole unit.
- 4.4 If the numerator is larger than the denominator, the fraction represents more than a whole.

These knowledge specifications were the driving force behind game design decisions. For instance, the game area was represented as a line in 1-dimensional levels and a grid in 2-dimensional levels to reinforce the idea that a unit can be represented as one whole interval on a number line (Knowledge Specification 1.3). Units were represented as blue ropes on a dirt path with small red posts indicating the fractional pieces the unit was broken into (see Figure 1). Students were given ropes in the resource bin on the left side of the game screen labeled “Path Options” and had to break the ropes they were given into the fractional pieces indicated in the level and place the correct number of unit fractions (fractions with a numerator of one) on each sign to guide their character safely to the cage to unlock the prize.

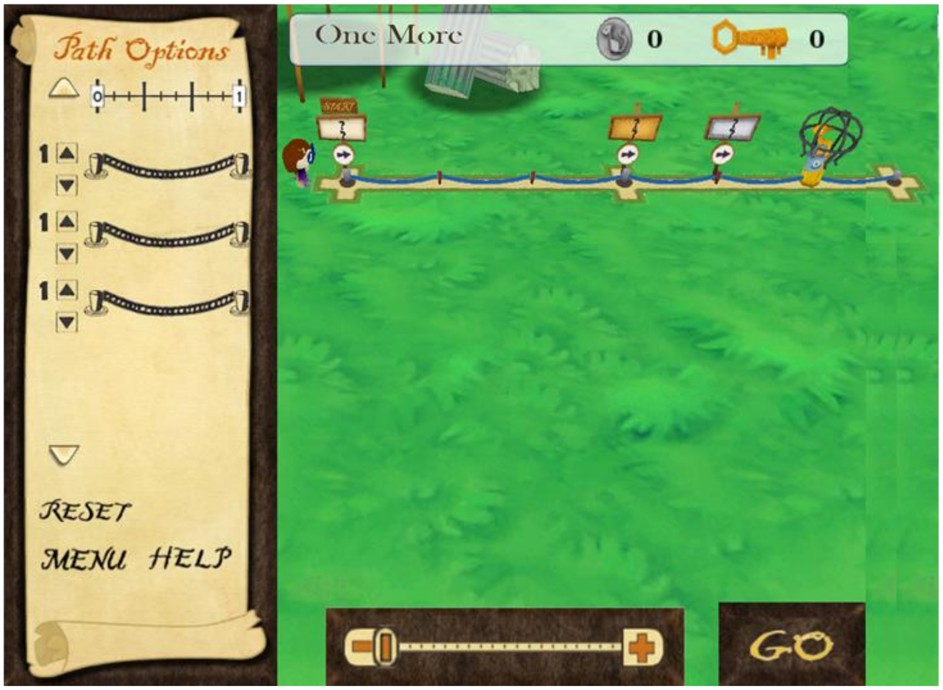


Figure 1.Screen shot from *Save Patch*.

The level pictured in Figure 1 represents two whole units (indicated by gray posts at intersections) broken into thirds (indicated by red posts in the spaces between gray posts). The distance between the first sign and the second sign is 1/1 (or 3/3), the distance between the second sign and the third sign is 1/3, and the distance between the third sign and the cage is 1/3. In order to correctly complete this level, a student would select the top rope in the “Path Options” and place it on the sign on the far left of the grid. (Alternatively, the student could break the first rope in the “Path Options” into thirds and place all three thirds on the sign.) The student would then break the second rope in the “Path Options” into thirds by clicking the down arrow to the left of the rope and place one of those thirds on the second sign and one of those thirds on the third sign. If the student made a mistake, he or she could either click on the “Help” button on the lower left to read through the help menu or click on the “Reset” button to remove all ropes from the signs and start the level over. The player could also click on the “Menu” button if he or she wanted to return to the main menu to change the game character to a different image. When the student felt that he or she had the correct ropes on each sign, he or she would click on the “GO” button. The game character would then move the distance represented by the ropes on the first sign. If that placed the game character on the second sign, it would then move the distance represented on that sign, and so on. Upon successfully reaching the cage, the cage would open, the game character would celebrate, and the student would move on to the next level.

This design allowed students to demonstrate knowledge of the meaning of the denominator of a fraction (Knowledge Specification 3.0) and the meaning of the numerator of a fraction (Knowledge Specification 4.0). Students could demonstrate understanding of addition as applied to fractions (Knowledge Specification 2.0) by adding rope pieces with the same denominator to a sign already loaded with rope pieces. Additionally, a number of levels in the game were designed to represent more than one unit, allowing students to demonstrate knowledge of the meaning and importance of the whole unit (Knowledge Specification 1.0).

Game play was constrained so that it was not possible to add two numbers with different denominators (Knowledge Specifications 2.1 and 2.3), rather than allowing the students to make the addition and having the game calculate the resulting distance. This means that the game did not allow students to add 1/2 to 1/3, instead forcing them to change the 1/2 rope to 3/6 and the 1/3 rope to 2/6 before allowing them to be added together. For the same reason, the game did not allow the creation of mixed numbers (e.g., 1½), forcing players to convert the whole number portion of the mixed number into the appropriate fractional representation (e.g., 2/2) before adding the fractional portion to the whole number portion.

Successful game play was intended to require students to determine the unit size for a given grid as well as the size of the fractional pieces making up each unit. The distance the character moved was a function of the number and size of rope pieces placed on each sign, where one whole rope represented a whole unit on the grid and each whole rope could be easily broken into fractional pieces of the desired size by clicking on the arrows next to the rope in the resource bin on the left side of the game screen. Therefore, a successful solution to a given level should indicate a solid understanding of the knowledge specifications underlying the game presentation.

In order to provide a logical progression through the knowledge specifications so that students would be exposed to problems that built on each other as they progressed through the game (Rupp et al., 2010), *Save Patch* was broken into seven stages (see Appendix A). The first stage was designed to introduce students to the game mechanics in a mathematical setting they were comfortable with, and therefore included only whole numbers. The second stage introduced fractions via unit fractions, requiring students to identify the denominator while restricting the numerator to one. The third stage combined concepts from the first two stages, with at least one distance in each level representing a unit fraction and at least one other distance representing a distance equivalent to a whole unit. The fourth stage was similar to the third stage, except that the distance representing a whole unit did not start and end on unit markers. Instead, the whole unit distance spanned a unit marker (e.g., extending from 1/3 to 4/3). The fifth stage dealt with proper fractions (e.g., the numerator was greater than one but smaller than the denominator), which required students to identify the numerator as well as the denominator of a fraction. The sixth stage completed the identification of fractions concepts by asking students to identify improper fractions (e.g., the numerator was larger than the denominator).

Logging Student In-Game Actions

In order to record student actions that might be indicative of their understanding of the knowledge specifications, the data generated by students while playing the game were stored in the form of a structured log written to a tab-delimited text file (Chung & Kerr, 2012). While educational video games are often designed to log every mouse click, we chose to ignore mouse clicks or drags that did not result in an action in the game (e.g., mouse clicks on the game background) and to capture only mouse clicks that represented game state changes or deliberate student actions such as clicking on a rope, dragging a rope to a sign on the grid, or clicking on the “Reset” button.

The intent was to capture student actions believed to indicate underlying understanding of the knowledge specifications at the smallest usable grain size to eliminate noise representing construct-irrelevant variance. However, such actions might not be fully interpretable without relevant game context information indicating the precise circumstances under which the action was taken (Koedinger et al., 2011). For this reason, each click that represented a deliberate action was logged in a row in the log file that included valuable context information such as the game level in which the action occurred and the time at which it occurred, as well as specific information about the action itself.

Additional structure was added by assigning codes to each of the different types of actions that could occur in the game, such as selecting a rope (code 3000) or adding a rope to a sign (code 3010). Codes 1000-1999 were used for general game information, such as game version or study condition; 2000-2999 were used for basic manipulation of objects, such as toggling a fraction to a new denominator; 3000-3999 were used for in-game mathematical decisions, such as adding a fraction to a sign; 4000-4999 were used for success or failure states such as player deaths or event-driven feedback; 5000-5999 were used for in-game navigation such as returning to the main menu or advancing to the next level; and 6000-6999 were used for the help menu system.

Using data codes allowed for the easy grouping of similar actions. Grouping specific actions such as the addition of a 1/2 rope to the first sign on the grid or the addition of a 1/8 rope on the third sign on the grid into a more general group (e.g., "adding fractions") made both evidence identification and evidence accumulation easier. For instance, the number of times a player reset a level could be determined by simply adding up all code 4010's appearing in the log data for that level, without having to determine post hoc which actions fell into this category. Additionally, if an entire category of actions (e.g., "scrolling ropes to a different denominator") proved to add little or nothing to the analysis, the whole category of actions could be easily ignored in later analyses.

The uniqueness of events was preserved by including columns in the log data capturing the specific detail of each event, along with a description of how to interpret the data. Thus, each action was captured at both a general and specific level, as can be seen by the logging in Table 2 of a set of student actions resulting in the addition of 2/3 to the leftmost sign on the grid. In this series of actions, the student breaks a 1/1 rope into three 1/3 pieces of rope (in the first row of Table 2) and selects one of those 1/3 ropes (in the second row). The student places that third on the leftmost sign on the grid (located at position 1/0), resulting in a value of 1/3 on that sign (in the third row). Then the student selects another 1/3 rope (in the fourth row) and places it on the same sign, resulting in a value of 2/3 on that sign (in the fifth row). The student hits the "Submit" button (in the sixth row) and the game character moves right 2/3 from the leftmost sign (in the seventh row). The student receives feedback congratulating him or her on passing the level (in the eighth row) and then advances to the next level (in the ninth and last row of Table 2).

Table 2

Hypothetical Log Data of a Student Adding 2/3 to the Leftmost Sign on the Grid

ID	Game time	Data code	Description	Data_01	Data_02	Data_03
1115	3044.927	2050	Scrolled rope from [initial value] to [resulting value]	1/1	3/3	
1115	3051.117	3000	selected coil of [coil value]	1/3		
1115	3054.667	3010	added fraction at [position]: added [value] to yield [resulting value]	1/0	1/3	1/3
1115	3058.443	3000	selected coil of [coil value]	1/3		
1115	3064.924	3010	added fraction at [position]: added [value] to yield [resulting value]	1/0	1/3	2/3
1115	3088.886	3020	Submitted answer: clicked Go on [stage] – [level]	2	3	
1115	3097.562	3021	Moved: [direction] from [position] length [value]	Right	1/0	2/3
1115	3106.224	4020	Received feedback: [type] consisting of [text]	Success	Congratu-lations!	
1115	3108.491	5000	Advanced to next level: [stage] – [level]	2	4	

In order for the log data to be used in cluster analysis or other data mining techniques, the log data in Table 2 first had to be transformed into a structure more amenable to those techniques. The first step in this process was to calculate a mnemonic that summarized all relevant information in each row. The mnemonic consisted of two parts. The first part was a word indicating the type of action being taken (e.g., ADD, SELECT, SCROLL, etc.) and the second part was one or more values with short indicators of the context of each additional value beyond the first (e.g., GET_1o3). All blank spaces were removed and all symbols in the values were replaced with letters to allow the mnemonic to function as a variable name in a variety of statistical software. For example, the symbol “/” was replaced with “o” (for “over”) and “-“ was replaced with “n” (for “negative”). See Table 3 for the mnemonics for the data in Table 2.

Table 3

Mnemonics for the Data in Table 2

ID	Game time	Data code	Mnemonic
1115	3044.927	2050	SCROLL_1o1_TO_3o3
1115	3051.117	3000	SELECT_1o3
1115	3054.667	3010	ADD_1o3_AT_1o0_GET_1o3
1115	3058.443	3000	SELECT_1o3
1115	3064.924	3010	ADD_1o3_AT_1o0_GET_2o3
1115	3088.886	3020	GO_WITH_2o3_AT_1o0
1115	3097.562	3021	MOVED_Right_2o3_FROM_1o0
1115	3106.224	4020	FB_Success_WITH_2o3_ON_1o0
1115	3108.491	5000	ADVANCED_TO_S2_L4

After the mnemonic was calculated, the data were transformed from long to wide format. In wide format, each unique mnemonic was a column and each attempt by a student to complete the level was a row with a value of “1” or “0” for each column indicating whether or not the action performed in that attempt corresponded with that particular mnemonic. The log data for each level were separated to create a series of data sets consisting of all actions made in each level in the game. This separation of the data was necessary because the same action might have different meanings in different levels. For example, ADD_1o2_AT_1o0_GET_1o2 (indicating the addition of 1/2 to the leftmost sign on the grid) would be a correct action in a level that was broken into halves, but an incorrect action in a level that was broken into thirds.

Within each level, the data were aggregated for each attempt (defined as the time between starting a level and either resetting the level or advancing to the next level), resulting in a matrix wherein one dimension consisted of all attempts to complete the level and the other dimension consisted of all actions made in those attempts. A hypothetical example of the resulting data can be seen in Table 4. The names of the mnemonics have been replaced by

“M1,”“M2,” etc. While this dataset had values for hundreds of mnemonics, only eight representative mnemonics are shown in Table 4 to illustrate the differences between attempts. The data in Table 2 and Table 3 correspond to student 1115’s third attempt to complete this level of the game.

Table 4

Hypothetical Example of Data Used in Cluster Analysis

ID	Attempt	M1	M2	M3	M4	M5	M6	M7	M8
1115	1	1	0	1	1	1	0	0	1
1115	2	0	0	0	0	0	1	1	0
1115	3	1	1	0	0	0	0	0	0
1116	1	1	1	0	0	0	0	0	0
1117	1	1	0	1	0	0	0	0	1
1117	2	1	1	0	0	0	0	0	0

Mnemonics: M1 = ADD_1o3_AT_1o0_GET_1o3, M2 = ADD_1o3_AT_1o0_GET_2o3,
M3 = CHNG_1o0_Right_TO_Down,M4 = ADD_WRONG_1o4_TO_1o3_AT_1o0,
M5 = CLICKED_HELP_WITH_1o3_ON_1o0, M6 = ADD_1o4_AT_1o0_GET_1o4,
M7 = ADD_1o4_AT_1o0_GET_2o4, M8 = RESET_WITH_1o3_ON_1o0

Cluster Analysis: Identifying Student Strategies

Groups of in-game student actions were identified using the fuzzy feature cluster analysis algorithm *fanny* (Maechler, 2012) in *R* (R Development Core Team, 2010). Cluster analysis operates over a distance matrix to group one dimension of the matrix based on similarities and differences in the other dimension of the matrix. Cluster analysis in educational contexts usually groups students based on similarities and differences in their answers to test questions or, in the case of educational video games or simulations, clusters student attempts based on similarities and differences in the actions taken in each attempt. In the hypothetical game level in Table 4, student 1115 attempt 3, student 1116 attempt 1, and student 1117 attempt 2 would be grouped together because the actions performed in those attempts were the same, while student 1115 attempt 2 and student 1116 attempt 3 would be in two different groups because the actions performed in those attempts were different. The distance calculation used when clustering answers to test questions is usually the Euclidean or Squared Euclidean distance because the data are either continuous or ordinal. However, in the analysis of *Save Patch* the data were binary, indicating whether or not a given student performed a given action in a given attempt, so the Manhattan distance was used instead (Cha, 2007).

Had the analysis of *Save Patch* used a standard cluster analysis approach, the result would have been groups of student attempts. While this would have resulted in the identification of distinct groups of students that differed in game play, it would not have been known what the distinguishing features of play were. For that reason, feature cluster analysis was run instead. Feature clustering differs from standard clustering techniques in that it groups descriptive features of entities (e.g., specific actions made by students) rather than grouping the entities themselves (e.g., students). Feature clustering does not require different algorithms from standard clustering, and can be run by simply transposing the matrix being operated on (Krier, Francois, Rossi, & Verleysen, 2007). In the data in Table 4, M1 and M2 would have been a group because the students who performed M1 (indicated by a 1 in that column) generally also performed M2 and the students who performed M2 generally also performed M1. A second group would have been made up of M3, M4, and M5, and a third group would have been made up of M6 and M7. Depending on what other students did, M3 and M5 might have been in a group of their own or they might have been in the group with M1 and M2. However, M5 and M6 would never have been in the same group because students who performed M5 (clicking the "Help" button with 1/3 on the leftmost sign) never performed M6 (adding 1/4 to the leftmost sign to get 1/4), and students who performed M6 never performed M5 because a student could never have two different denominators (thirds and fourths) on the same sign.

Having the output be a list of actions rather than a list of students meant that the groups could be identified by naming each group with the strategy the actions were representative of. For example, in Table 4 the group of actions including M1 and M2 could easily be identified as the *standard solution* to the level because adding 1/3 to the leftmost sign to get 1/3 (M1) and then adding a second 1/3 to the leftmost sign to get 2/3 (M2) is the standard way to complete the level.

The analysis of *Save Patch* also differed from most cluster analyses used in educational data mining in that it used fuzzy cluster analysis (Ruspini, 1969) rather than hard cluster analysis. Hard clustering forces each action to belong to a single cluster, whether or not all actions are easily classified in this manner. While hard clustering has been used successfully with test items to detect multidimensionality (Roussos et al., 1998), to develop a hierarchy of concepts (Chiu et al., 2008), and to find conceptual similarities among items (Madhyastha & Hunt, 2009), log files from educational video games are far more likely to require the use of fuzzy clustering algorithms due to the highly intercorrelated nature of the data they generate. For example, in the level in Table 4, hard clustering would force M1 to belong to either the group with M2 or the group with M3 and M8 (even though M1 sometimes occurs with M2 and other times occurs with M3 and M8). Fuzzy cluster analysis would allow it to belong to both groups, using probability theory to identify the degree of belongingness in each cluster. This allows for superior clustering results for data with problematic data points lying between otherwise easily identifiable clusters, such as the action M1 that could belong to two separate clusters (Ruspini, 1969). This is likely to be an issue with data from educational video games because different strategies for solving problems often include the same initial steps. While fuzzy clustering and hard clustering will return similar results if the data are not very fuzzy, the fuzzier the data are, the more imprecise the hard clustering results will be (Tian et al., 2008). Because the data from *Save Patch* ranged from moderately fuzzy to very fuzzy across different levels in the game (indicated by the Dunn coefficient in the fuzzy clustering output), fuzzy clustering was the most appropriate choice for clustering in-game actions into the various strategies students used to complete levels in *Save Patch*.

Clustering algorithms provide no single definitive method of determining the number of clusters present in the data. Rather, solutions with different numbers of clusters must be compared using both statistical and substantive criteria. In order to determine the correct number of clusters in each level in the fuzzy cluster analysis, each level was run with two clusters, then three clusters, then four, etc. until one of two things occurred: incorrect actions began to appear in the *standard solution* cluster, or the additional cluster provided no additional interpretive value (e.g., the additional cluster resulted in the split of an easily identifiable strategy into two parts). Once either of those outcomes occurred, cluster analysis for that level ceased and the largest number of previously successful clusters was retained. For example, if incorrect actions began to appear in the *standard solution* cluster when a level was run with eight clusters, it was determined that seven was the optimal number of clusters for that level.

This process led to the identification of six different types of valid solution strategies and nine different types of errors in an earlier version of *Save Patch* (Kerr & Chung, 2012). The *standard solution* was the anticipated solution for each level. In the level in Figure 1, as explained earlier, the *standard solution* consisted of placing 1/1 on the first sign, 1/3 on the second sign, and 1/3 on the third sign.

A *fractional solution* differed from the standard solution only in that whole units were represented as their fractional equivalents. As shown in Figure 2, this strategy would result in a student placing3/3 rather than 1/1 on the first sign, but having an otherwise identical response as a student using the *standard solution*. An *alternate solution* was a valid solution using a denominator other than the one represented in the level. As shown in Figure 2, this would result in a student placing 1/1 (or 6/6) on the first sign, 2/6 on the second sign, and 2/6 on the third sign. A *shortcut solution* was a valid solution that skipped one or more signs. As shown in Figure 2, this would result in a student placing 1/1 on the first sign, 2/3 on the second sign, and leaving the third sign empty. An *incomplete solution* occurred when a student placed ropes correctly on one or more signs, but left one or more signs empty. As shown in Figure 2, this might result in a student placing 1/1 on the first sign, and 1/3 on the second sign, but leaving the third sign blank. A *reset solution* occurred when a student placed ropes correctly on all signs, but then hit reset rather than submitting their answer (this strategy is not shown as a separate row of values in Figure 2 because it has the same values placed on each sign as the *standard solution*).

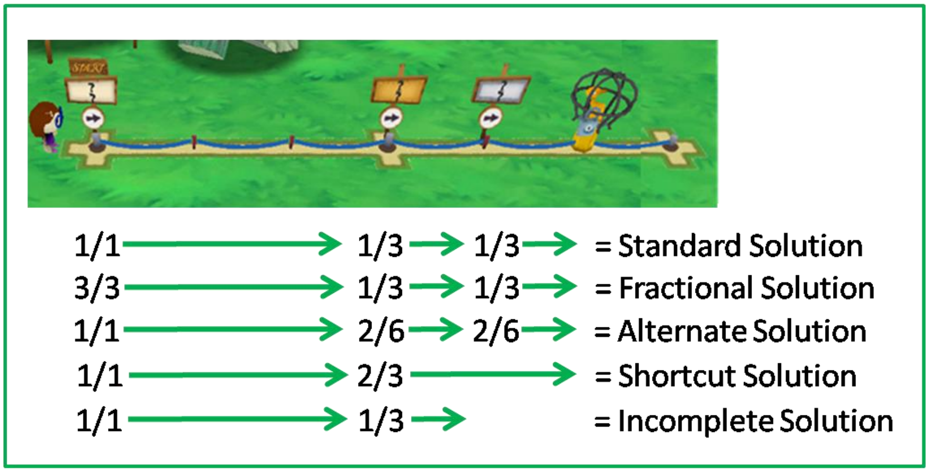

START
1/1 → 1/3 → 1/3 → = Standard Solution
3/3 → 1/3 → 1/3 → = Fractional Solution
1/1 → 2/6 → 2/6 → = Alternate Solution
1/1 → 2/3 → = Shortcut Solution
1/1 → 1/3 → = Incomplete Solution

Figure 2. Solution strategies identified by cluster analysis.

Most of the nine strategies that resulted in errors were mathematical misconceptions involving the three main skills required for adding fractions: unitizing, partitioning, and iterating. Unitizing consists of correctly identifying the number of units represented, partitioning consists of correctly identifying the number of pieces each unit is broken into (e.g., the denominator of the represented fraction), and iterating consists of identifying the number of unit fractions in the representation (e.g., the numerator of the represented fraction) (Olive & Lobato, 2008).

Students who made *unitizing errors* were unable to determine the number of units being represented. These students either saw the entire representation as a single unit, regardless of the number of units represented, or saw each fractional piece as a whole unit. As shown in Figure 3, this would result in a student placing 3/6 on the first sign (rather than 3/3), 1/6 on the second sign (rather than 1/3), and 1/6 on the third sign (rather than 1/3) if he or she saw the entire representation as a single unit. If a student saw each fractional piece as a whole unit, it would result in that student placing 3/1 on the first sign (rather than 3/3), 1/1 on the second sign (rather than 1/3), and 1/1 on the third sign (rather than 1/3).

Students who made *partitioning errors* were unable to determine the number of pieces the unit was broken into. These students either counted dividing marks to determine the number of pieces the unit was broken into (e.g., seeing halves instead of thirds) or counted dividing marks and unit marks (e.g., seeing fourths instead of thirds). As shown in Figure 3, this would result in a student placing 3/2 on the first sign (rather than 3/3), 1/2 on the second sign (rather than 1/3), and 1/2 on the third sign (rather than 1/3) if he or she counted dividing marks to determine the denominator. If a student counted both dividing marks and unit marks to determine the denominator, this would result in that student placing 3/4 on the first sign (rather than 3/3), 1/4 on the second sign (rather than 1/3), and 1/4 on the third sign (rather than 1/3). Some students combined both errors, *unitizing and partitioning*, seeing the entire representation as one unit and counting dividing marks to determine the number of pieces the unit was broken into. As shown in Figure 3, this would result in a student placing 3/8 on the first sign (rather than 3/3), 1/8 on the second sign (rather than 1/3), and 1/8 on the third sign (rather than 1/3).

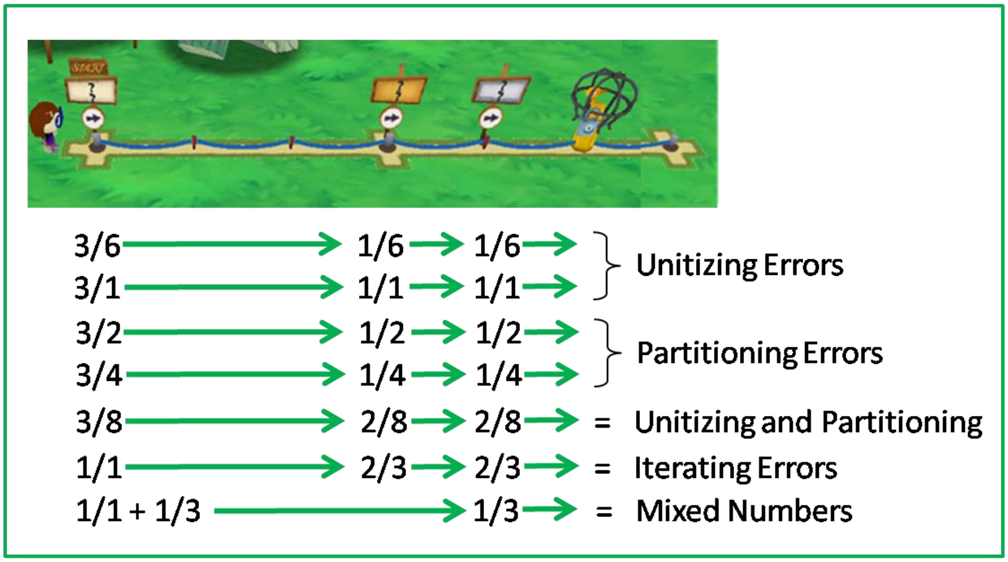

3/6 → 1/6 → 1/6 →
3/1 → 1/1 → 1/1 →
Unitizing Errors
3/2 → 1/2 → 1/2 →
3/4 → 1/4 → 1/4 →
Partitioning Errors
3/8 → 2/8 → 2/8 → = Unitizing and Partitioning
1/1 → 2/3 → 2/3 → = Iterating Errors
1/1 + 1/3 → 1/3 → = Mixed Numbers

Figure 3.Mathematical errors identified by cluster analysis.

Students who made *iterating errors* were unable to determine the numerator of the represented fractions. These students were able to correctly identify the number of units being represented and the number of pieces each unit was broken into, but were unable to determine the number of pieces to put on each sign. As shown in Figure 3, this would result in a student placing 1/1 on the first sign (correctly), but placing 2/3 on the second sign (rather than 1/3) and 2/3 on the third sign (rather than 1/3). Additionally, some students saw improper fractions as *mixed numbers* and attempted to add a fractional rope to a whole unit rope already placed on a sign. As shown in Figure 3, this would result in a student adding 1/1 to the first sign (correctly) and 1/3 on the third sign (correctly), but then trying to add 1/3 to the 1/1 already placed on the first sign (which is not a valid move in the game), leaving the second sign empty because they saw the distance between the first sign and the third sign as being $1\frac{1}{3}$.

Additionally, the cluster analysis identified two game-related errors where students used the correct mathematical strategies but moved in the *wrong direction* or *skipped signs* that were mandatory (indicated in the color of the sign, e.g., the second sign in Figure 3). Other students avoided math entirely by placing all the available resources on the signs in the order that they were given in the resource bin (the *everything in order* strategy). All attempts that did not fall into one of the preceding strategies were identified as *unknown error*.

The game was designed to teach students about adding fractions, so we had anticipated that students might make *unitizing errors* and *iterating errors* and might see fractions as *mixed numbers*, but we did not anticipate students would have trouble determining the denominator that was being represented and were therefore surprised to see the *partitioning error*. Because there was not sufficient knowledge apriori about the effect of the game representation on student behavior, we knew that there would be strategies related to game play rather than mathematics but did not know what those strategies would be. We were also surprised to discover that students made *incomplete solutions* or *reset solutions* without submitting the answer to see what the game character would do. Had the cluster analysis not been run, these strategies would not have been identified and less than half of students' in-game behavior could have been interpreted.

Because the strategies were identified using the same game and students of the same age as the students in the current study, it was deemed unnecessary to run the cluster analysis again. Since the strategies for each level of the game were already identified, the attempts in this study were simply coded with whichever previously identified strategy was being employed. Any attempt not identified as belonging to a known strategy was coded as *unknown error*. This process resulted in a dataset indicating which strategy was used in each attempt each student made to solve each level, resulting in a data set such as the one shown in Table 5.

Table 5

Hypothetical Strategy Assignment for Attempts

ID	Attempt	Strategy
1115	1	Unknown Error
1115	2	Partitioning Error
1115	3	Standard Solution
1116	1	Standard Solution
1117	1	Wrong Direction
1117	2	Standard Solution

Sequence Mining: Identifying Strategy Sequences Within Levels

Frequent patterns of strategy use were identified using the sequence mining algorithm *cspade* (Buchta & Hahsler, 2013) in *R* (R Development Core Team, 2010). Sequence mining looks for frequent patterns of behavior across time. Due to its roots in market research, the data for sequence mining are organized in *baskets*, with each basket corresponding to the set of items purchased by an individual within a given unit of time. In our analysis each basket consisted of a single strategy, and the unit of time used was the attempt number corresponding to that strategy.

Additionally, sequence mining algorithms require the creation of an *alphabet* that shortens the name of each potential basket item into an abbreviated form. The alphabet for this analysis (see Table 6) consisted of the first letter of each strategy, except for cases where more than one strategy shared the same first letter (the *alternate solution*, *skipped key*, *unitizing and partitioning*, *iterating error*, and *unknown erro*r strategies) and cases where the first letter of the second word was deemed more informative (the *wrong direction* strategy). This process resulted in a dataset similar to Table 5, but with the strategy names replaced with their corresponding alphabet letters and with the addition of a column between the attempt number and the strategy indicating the number of items in each basket (in this case, that number was always one because only one strategy was used in any given attempt).

Table 6

Sequence Mining Alphabet

Strategy	Alphabet
Standard Solution	S
Alternate Solution	T
Incomplete Solution	I
Fractional Solution	F
Reset Solution	R
Skipped Key	K
Wrong Direction	D
Unitizing Error	U
Partitioning Error	P
Unitizing and Partitioning	B
Iterating Error	N
Converting to Wholes Error	C
Avoided Math	A
Unknown Error	O

The *cpade* algorithm has six parameters that can be changed to limit the results: *support*, *maxsize*, *maxlen*, *mingap*, *maxgap*, and *maxwin*. The *support* is a required parameter indicating the minimum percentage of the data that must fall in a given sequence for it to be considered frequent. The default value for *support* is .10 (10%), but the *support* was lowered to .02 (2%) due to the exploratory nature of the study and because the sample size was deemed sufficiently large for 2% of the sample to be an interesting subset of students.

The *maxsize* and *maxlen* are optional parameters indicating the maximum number of items a sequence can hold. The *maxsize* parameter is used when there is more than one item in a basket to limit the number of items representing each basket in the sequence of baskets. This parameter was not set because there was only one item in each basket. The *maxlen* parameter limits the number of baskets in a given sequence. This parameter was not set because there was no theory regarding the maximum number of strategies that might fall in a strategy sequence.

The *mingap*, *maxgap*, and *maxwin* are optional parameters indicating the range of time between consecutive items in a given sequence. The *mingap* limits the minimum time difference between consecutive baskets in a sequence and the *maxgap* limits the maximum time difference between consecutive baskets in a sequence. The *maxwin* parameter limits the maximum time difference between both consecutive and nonconsecutive baskets in a sequence. To increase interpretability, *mingap* and *maxgap* were both set to 1.0 so that consecutive strategies in a strategy sequence would indicate strategies used in consecutive attempts rather than strategies used in some later attempt.

This means that for students making a *partitioning error* in their first attempt, a *unitizing error* in their second attempt, and getting the *standard solution* in their third attempt, two sequences were identified: *partitioning error* to *unitizing error*, and *unitizing error* to *standard solution*. However, the sequence *partitioning error* to *standard solution* was not identified as a sequence because those strategies were not consecutive.

The resulting dataset consisted of all frequent sequences of strategies used to complete levels in *Save Patch*. Each row in the dataset corresponded to a single student's behavior in a single level, rather than each attempt made at that level, as shown in Table 7. Columns corresponded to each unique one-step sequence that student used to complete the level. For example, a student who made a *partitioning error*, followed by a *partitioning error*, followed by another *partitioning error*, followed by the *standard solution*, was coded as having completed the sequence *PtoP* because they moved from a *partitioning error* directly to another *partitioning* *error* at least once in the level, and the sequence *PtoS* because they moved from a *partitioning error* to the *standard solution* at least once in the level. A given sequence was coded only once for each student, regardless of the number of times that sequence was completed by that student.[1]

Table 7

Hypothetical Strategy Sequences for Each Level in Stage 4

ID	Level	Sequence1	Sequence2	Sequence3
1115	13	UtoP	PtoP	PtoS
1115	14	UtoU	UtoS	
1115	15	S		
1116	13	DtoS		
1116	14	S		
1116	15	RtoS		

Students who completed a given level on their first attempt did not fall into an identified sequence for that level (since sequences consist of more than one attempt). These students were coded as *S* if they completed the level on their first attempt with the *standard solution* and *F* if they completed the level on their first attempt with a *fractional solution*. Students who did not complete the level on their first attempt and did not perform any of the identified sequences were coded as *O* because they used a strategy sequence other than those identified by the sequence mining process.

Had strategy sequences within a stage been identified using sequence mining (like the identification of strategy sequences within a level), a large percentage of students would not have been identified by the analysis. This is because the sequence mining algorithm looks only for exactly matching sequences. The hypothetical student 1115 in Table 7 moves from the strategy

[1] Sequence mining algorithms identify a sequence only once per student, regardless of the number of times a given student completes that sequence. Therefore, a given sequence was coded only once for each student in the resulting dataset, regardless of the number of times that sequence was completed, to maintain consistency with the sequence mining output.

sequence *UtoP-PtoP-PtoS* in the first level of the stage (Level 13) to the strategy sequence *UtoU-UtoS* in the second level of the stage (Level 14) and to the strategy sequence *S* in the last level of the stage (Level 15).A sequence mining algorithm would only group this student with other students who used exactly the same number of attempts for each level as student 1115 and the exact same strategy sequences in each attempt as student 1115. This would result in the identification of a large number of sparsely populated sequences, particularly in stages with four or five levels where students required more than two or three attempts to complete each level.

This is not just a practical problem, but a substantive issue as well. What is substantively meaningful about student 1115's sequences is that the student began the stage making *partitioning errors*, made no *partitioning errors* in subsequent levels, and ended the stage completing the level on the first attempt. This sequencing demonstrates that student 1115 did not know how to partition correctly at the beginning of the stage, but had stopped partitioning incorrectly by the end of the stage. This student should be coded as having stopped making *partitioning errors* in Stage 4, along with all other students who demonstrated this general pattern. In contrast, student 1116 either completed each level on the first attempt or made a single non-math-related error (e.g., *wrong direction* or *reset solution*) before completing the level correctly. This student should be coded as having completed levels in Stage 4 correctly, along with all other students who demonstrated this general pattern.

Table 8

Targeted Errors for Each Stage

Stage	Intended goal	Targeted error
1	Learn how to play the game	A (Avoided Math) K (Key Error)
2	Learn how to identify the denominator of a fraction	P (Partitioning Error)
3	Learn how to identify the denominator of a fraction	P (Partitioning Error)
4	Learn how to identify the denominator of a fraction	P (Partitioning Error)
5	Learn how to identify the numerator of a fraction	N (Numerator Error)
6	Learn how to identify an improper fraction	N (Numerator Error) C (Converting Error)

In order to group students who displayed substantively similar behavior, the various strategy sequences for each level were recoded based on their movement between targeted errors, other errors, and the correct solution. Targeted errors for each stage are shown in Table 8. In Stages 2, 3, and 4, the targeted error was the *partitioning error*. Therefore, in the levels in these stages, students were coded as either *Correct*, *Partitioning Error to Correct*, *Partitioning Error to Partitioning Error*, *Partitioning Error to Other Error*, *Other Error to Correct*, *Other Error to Partitioning Error*, or *Other Error to Other Error*.

The information used to determine these strategy sequence types came from graphs of the strategy sequences for each level, as shown in Figure 4. These graphs show the percentage of students in each strategy sequence in the blue values next to the lines between strategies. The percentage of students repeating the same error is shown in the orange values inside the dotted loops. The percentage of students using a strategy in the course of the level is shown in the gray values next to each strategy. The targeted error is shown in purple. Graphs for all levels can be found in Appendix B.

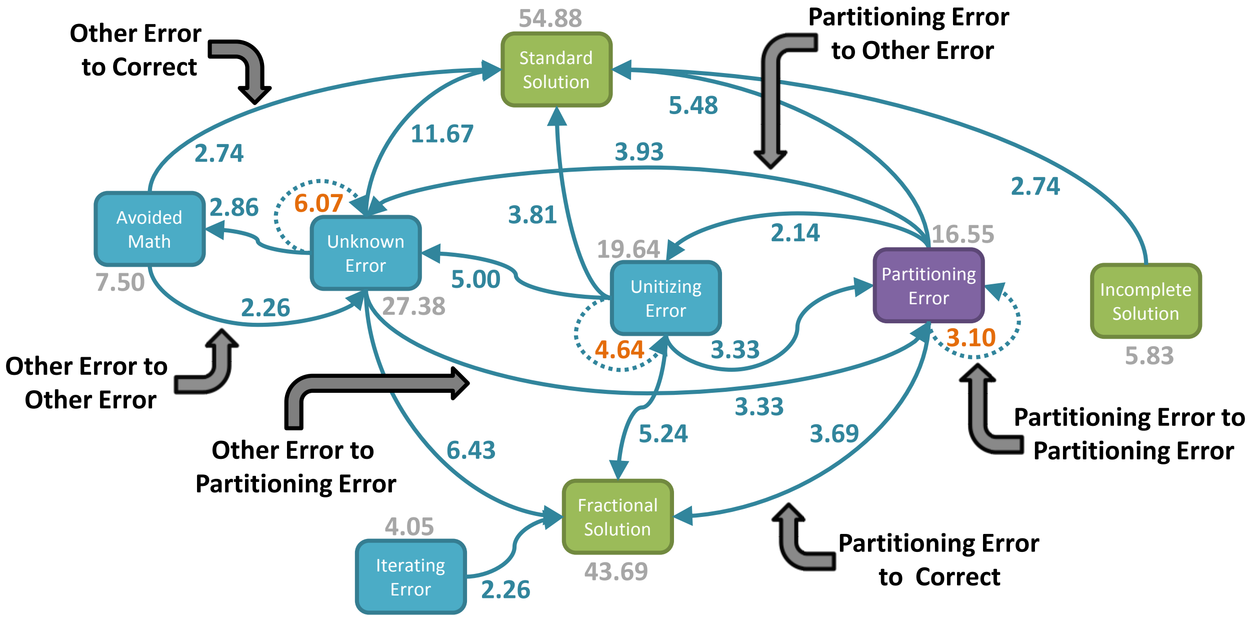


Figure 4.Example of graphed strategy sequences for Level 9.

Students were coded as being in the *Correct* strategy sequence type if they got the level correct on their first attempt. Students were coded as being in the *Partitioning Error to Correct* strategy sequence type if they made *partitioning errors* in early attempts at the level, but quickly moved to the correct solution. Students were coded as being in the *Partitioning Error to Partitioning Error* strategy sequence type if they made *partitioning errors* in early attempts at the level and repeated that error in later attempts at the level. Students were coded as being in the *Partitioning Error to Other Error* strategy sequence type if they made *partitioning errors* in early attempts at the level and made other errors in later attempts at the level.

Students were coded as being in the *Other Error to Correct* strategy sequence type if they made errors other than *partitioning errors* in early attempts the level, but quickly moved to the correct solution. Students were coded as being in the *Other Error to Partitioning Error* strategy sequence type if they made errors other than *partitioning errors* in early attempts at the level and made *partitioning errors* in later attempts at the level. Students were coded as being in the *Other Error to Other Error* strategy sequence type if they made errors other than *partitioning errors* both in early attempts at the level and in later attempts. This process would result in student 1115 being coded as *Partitioning Error to Correct* for Level 13, *Other Error to Correct* for Level 14, and *Correct* for Level 15, and student 1116 being coded as *Correct* in all three levels.

After coding each level in a given stage, the codes for each level were combined into an overall grouping reflecting student behavior across all the levels of the stage. This process mirrored the process for coding each level, using the strategy sequence types for each level to come up with a sequence group type for the stage as a whole. The coding process for determining the *Partitioning Error to Correct* sequence group type in Stage 4 is shown in Table 9. Coding for the sequence group types in Stage 4 can be found in Appendix C.

Table 9

Sequence Groups Coded as Partitioning Error to Correct in Stage 4

Group	Level 13 sequence type	Level 14 sequence type	Level 15 sequence type
1	Correct	Partitioning Error to Correct	Correct
2	Other Error to Correct	Partitioning Error to Correct	Correct
3	Other Error to Other Error	Partitioning Error to Correct	Correct
4	Partitioning Error to Correct	Correct	Correct
5	Partitioning Error to Correct	Other Error to Correct	Correct
6	Partitioning Error to Correct	Other Error to Other Error	Correct
7	Partitioning Error to Correct	Partitioning Error to Correct	Correct

This reduced the information for each student to a single row in the dataset, with a column for each stage, as shown for selected stages in Table 10.

Table 10

Hypothetical Sequence Coding for Stages Targeting Partitioning

ID	Stage 2	Stage 3	Stage 4
1115	Partitioning Error to Correct	Other Error to Correct	Partitioning Error to Correct
1116	Correct	Correct	Partitioning Error to Correct

Classification: Identifying Performance Trajectory Types

The previous section measured performance as changes in strategy use, with positive performance being characterized by either continuing use of either the *standard solution* or *fractional solution* or by moving from the targeted error to the correct solution. Another way of characterizing performance is as changes in the number of attempts made to complete each level in a stage, with positive performance being characterized by either completing each level in only one attempt or by moving from a large number of attempts in early levels in the stage to one attempt in later levels in the stage.

In order to use the number of attempts as a measure, attempts caused by computer glitches had to be separated from meaningful student attempts to complete the level. Computer glitches resulting in meaningless attempts occurred largely either because the student clicked reset twice in a row (either accidentally or due to impatience with the speed of the avatar) or accidentally clicked “Go” immediately after a new level loaded (due to the initial location of the cursor directly above the “Go” button). If left in the dataset, these meaningless attempts would artificially inflate the number of attempts those students required to complete each level and thereby indicate a greater level of difficulty than was actually the case. Therefore, these attempts were not included in the number of times each student tried to complete each level.

The change in the number of attempts required to solve each level in a given stage was plotted to form a performance trajectory. In the performance trajectory shown in Figure 5, the student completed the first level in the stage in nine attempts, the second level in five attempts, the third in two attempts, and the final level in the stage in only one attempt.

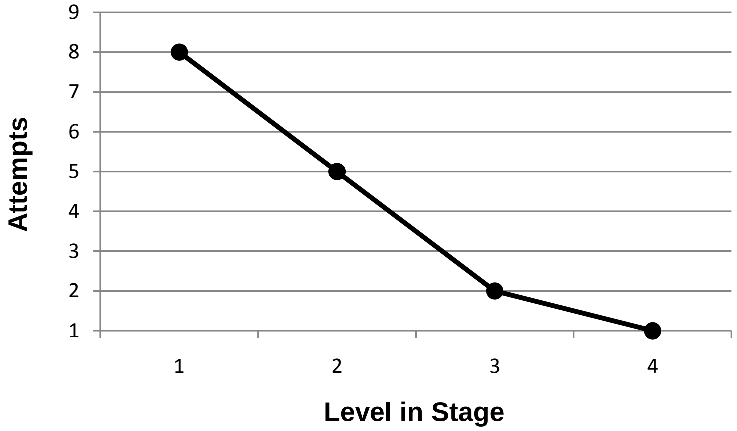


Figure 5.Example performance trajectory.

The first step in analyzing the performance trajectories was to group substantively similar performance trajectories into performance trajectory types similar to the sequence group types in the previous section, but without the ability to discriminate between targeted errors and other errors. In order to identify the different performance trajectory types present in the data, 10% of students in the sample were hand classified.

The hand classification identified six different performance trajectory types (see Figure 6). There were two different performance trajectory types for students who completed the levels in the stage correctly on the first attempt: the *All Correct* performance trajectory type included students who completed every single level in the stage on the first attempt, and the *Only One Mistake* performance trajectory type included students who completed all levels in the stage on their first attempt, except for one level in which they took two attempts.

There were two different performance trajectory types for students whose performance improved throughout the stage: the *Improved to 1* performance trajectory type consisted of students who required fewer attempts to complete each subsequent level in the stage and completed the final level in the first attempt, and the *Partially Improved* performance trajectory type consisted of students who required fewer attempts to complete subsequent levels but who did not complete the final level in the stage in the first attempt.

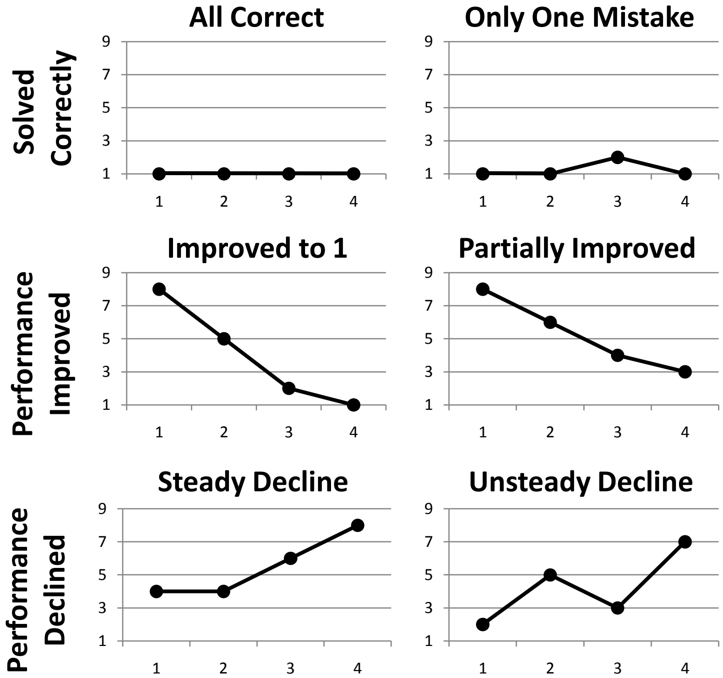


Figure 6. Identified types of performance trajectories.

There were also two different performance trajectory types for students whose performance declined throughout the stage: the *Steady Decline* performance trajectory type consisted of students who required consistently more attempts to complete each subsequent level in the stage, and the *Unsteady Decline* performance trajectory type consisted of students who generally required more attempts over time, but performed better on at least one level in the stage, resulting in a more ragged uphill trajectory. While it was theoretically possible to find additional performance trajectory types (such as *Unsteady Improved*) these were the only performance trajectory types found in any of the stages for the hand-coded students.

After 10% of students' performance trajectory types were hand coded, the remaining students were automatically coded using the *k*-nearest neighbor classification algorithm *knn*

(Venables & Ripley, 2002) in *R* (R Development Core Team, 2010) with *k* equal to one. The *knn* classification algorithm uses a coded subset of the data to determine the codes for the uncoded students. The choice of distinguishing features included in the subset of data used to train the classification algorithm is the most important step in the process, as it determines how the remaining students are classified (Minaei-Bidgoli et al., 2003). This is particularly true in cases where the dataset is too small for machine learning techniques to be applied directly (Kotsiantis et al., 2010), as is often the case with educational data.

The distinguishing features in performance trajectories are the change in performance from one level to the next and the number of attempts in each level. In performance trajectories, the exact value of the change in performance from one level to the next is not as important as the direction of the change, and the exact value of the number of attempts in each level is not as important as whether or not the number indicates mastery. For this reason, each change in performance from one level to the next in a given trajectory was coded as either being *positive* (where the number of attempts in the second level was more than the number of attempts in the first level), *negative* (where the number of attempts in the second level was less than the number of attempts in the first level, or *zero* (where the number of attempts was the same in both levels). The number of attempts in each level in a given trajectory was coded as *one* (indicating mastery), *two* (indicating near-mastery), or *many* (indicating non-mastery).

The list of distinguishing features chosen for classification is listed in Table 11. Each feature for the change in performance between levels is calculated multiple times in each performance trajectory because there is more than one change in each trajectory (equal to the number of levels minus one). The same is true for each feature for the number of attempts, except the number of these features in each performance trajectory is equal to the number of levels in the stage.

Table 11

Features Selected as Input for Classification

Feature	Description
Change_Positive	Second level completed in more attempts than first level
Change_Negative	Second level completed in fewer attempts than first level
Change_Zero	Second level completed in the same number of attempts as first level
One_Attempt	Level completed in one attempt
Two_Attempts	Level completed in two attempts
Many_Attempts	Level completed in more than two attempts

This process (hand coding 10% of the learning trajectories, calculating the discriminating features, running the *knn* algorithm) was repeated for each stage in the game. The resulting dataset consisted of descriptions of the learning behavior of each student in each concept covered by *Save Patch*. This reduced the information for each student to a single row in the dataset, with a column for each stage, as shown in Table 12.

Table 12

Hypothetical Performance Trajectory Coding for Each Stage

ID	Stage 1 Type	Stage 2 Type	Stage 3 Type	Stage 4 Type	Stage 5 Type	Stage 6 Type
1115	Improved to 1	All Correct	Improved to 1	Improved to 1	All Correct	All Correct
1116	Improvedto 1	Improved to 1	Steady Decline	Unsteady Decline	All Correct	All Correct

Developing and Testing Hypotheses Using Data Mining Results

The results of each data mining technique were examined and one or more hypotheses of interest were generated regarding the relationship between the strategies/sequences/classes occurring with sufficient frequency in the game and performance on the pretest and/or posttest.

A rationale for each hypothesis is provided, and the accuracy of the resulting prediction(s) was examined using appropriate analyses. Additionally, information about the number of students in each resulting subsample and the method of determining relevant pretest/posttest items is provided.

Study Design

This study uses data from a larger study[2] examining the effect of four educational video games on students' understanding of rational numbers. In the larger study, 1,746 sixth grade students in 62 math classes in nine urban and suburban school districts were randomly assigned (by class) to either the treatment condition (consisting of four educational video games on rational numbers) or the control condition (consisting of four educational video games on solving equations). Students in both conditions took a paper-and-pencil pretest on the first day of the study, followed by 10 nonconsecutive days of video game play, and completed another paper-and-pencil posttest on the last day of the study. Though students only spent a total of 12 days of class time in the study, the time between pretest and posttest was much longer. This is because teachers were allowed to choose the individual dates of the study and many teachers chose to spend one day a week on study activities, extending the study duration over a number of months. The start date of the study ranged from November 30, 2011, to March 26, 2012, between teachers, and the end date ranged from December 19, 2011, to May 25, 2012. The shortest number of days between pretest and posttest was 19, and the longest was 110.

Students in the treatment condition spent two days playing *Wiki Jones*, a game about the identification of fractions. Then those students spent four days playing *Save Patch*, a game about the addition of fractions, followed by two days playing *Tlaloc's Book*, a game about the

[2] The larger study was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305C080015 to the National Center for Research on Evaluation, Standards, and Student Testing (CRESST).

multiplication and division of fractions. Finally, students spent two days playing *Rosie's Rates*, a game about calculating rates.

This study focuses on *Save Patch*, the longest and most well-researched game in the larger study. There were 855 students in 31 classrooms who played at least one level of *Save Patch*, 724 of whom completed all 27 criterion levels in the game. Of the 855 students who started the game, 373 were male, 401were female, and 81 did not report their gender. 371 were Hispanic, 193 were White, 45 were African American, 35 were Asian, 14 were Native American, 72 were multiracial, and 125 were classified as other or did not report their ethnicity. Additionally, 380 students reported speaking English at home less than half of the time. The log data from the game consisted of 1,288,103 individual actions taken by the students in the course of game play, 17,685 of which were unique.

The paper-and-pencil pretest and posttest were broken by design into four sections matching the content for each of the four games. *Save Patch* focused on the addition of fractions, but also provided remediation for identifying fractions (which was covered previously in the *Wiki Jones* game). Therefore, the pretest and posttest items on both the identification of fractions and addition of fractions were included in the analysis in this study, while the pretest and posttest items on multiplication and division of fractions and calculating rates were not included. A full list of included items can be found in Appendix D. All items on the pretest were repeated on the posttest. Students scored an average of 5.82on the 13 items on the pretest, with a standard deviation of 3.27. Students scored an average of 6.96 on the 13 items on the posttest, with a standard deviation of 3.76. Partial credit was given on pretest and posttest items with multiple parts (e.g., a numerator and a denominator).

CHAPTER 4: RESULTS

Information Extracted Using Cluster Analysis

This section examines the following question: Can cluster analysis be used to extract information leading to testable hypotheses about the relationship between in-game performance and performance on paper-and-pencil posttests? An affirmative answer to this question depends on (1) being able to identify clusters of students who can be characterized according to meaningful strategies or errors that they made during the course of the game, and (2) identifying clusters that are large enough for statistical analysis. In this study, both conditions were satisfied.

Table 13

Strategy Distribution Across Attempts

Strategy	Frequency	Percentage
Standard Solution	18161	33.0%
Fractional Solution	3757	6.8%
Shortcut Solution	161	0.3%
Alternate Solution	521	0.9%
Reset Solution	610	1.1%
Incomplete Solution	1827	3.3%
Unitizing Error	2111	3.8%
Partitioning Error	7916	14.4%
Unitizing and Partitioning	516	0.9%
Iterating Error	5215	9.5%
Mixed Numbers	445	0.8%
Skipped Key	1585	2.9%
Wrong Direction	1223	2.2%
Everything In Order	999	1.8%
Unknown Error	9992	18.2%

As can be seen in Table 13, cluster analysis successfully identified a majority of attempts in *Save Patch* as belonging to a specific solution strategy or error pattern. The most common strategy used in the game was the *standard solution* strategy, accounting for 33.0% of all attempts. The most common error patterns corresponded to *partitioning errors* (14.4% of attempts), *iterating errors* (9.5% of attempts), and *unitizing errors* (3.8% of attempts). Unidentified strategies, coded as *unknown error*, made up 18.2% of the 55,039 attempts in the game.

There were four relatively infrequent strategies, each accounting for less than 1% of attempts. The *alternate solution* and the *unitizing and partitioning* error occurred in 0.9% of attempts, seeing improper fractions as *mixed numbers* occurred in 0.8% of attempts, and the *shortcut solution* occurred in 0.3% of attempts.

Developing and Testing a Hypothesis Using Cluster Analysis Results

This section addresses the following question: Can the information produced by the cluster analysis be used to diagnose student errors related to fractions understanding? To investigate this question, analyses were performed to determine the relationship between the frequency of a specific type of error made during game play and performance on pretest and posttest items measuring proficiency in that area.

The error type selected for this analysis was the *unitizing error*. This error was specifically selected for analysis because *Save Patch* was not designed to remediate unitizing errors. Therefore, a reasonable interpretation of a student making a large number of *unitizing errors* is that the student holds a misconception about unitizing (a misconception that could not be remedied during the course of the game). Similarly, it might be argued that the larger the number of *unitizing errors* a student makes during the course of the game, the stronger the student’s misconception about unitizing. This argument provides the basis for examining the relationship between the frequency of *unitizing errors* performed during the game and performance on the paper-and-pencil test items requiring an understanding of unitizing. If the number of *unitizing errors* made during the game reflects the strength of the unremediated unitizing misconception as hypothesized, then two predictions are possible: (1) the number of *unitizing errors* made during the game should be positively correlated with the number of *unitizing errors* made on the posttest, and (2) there should be no improvement from pretest to posttest in terms of performance on unitizing items (in particular, students making a large number of *unitizing errors* during the game should have low proportions of unitizing items answered correctly on both the pretest and the posttest).

Other major types of errors, including *partitioning errors* and *iterating errors*, could not be used to make such testable hypotheses because their frequency in the game does not necessarily reflect the strength of a misconception held by the end of the game. For example, *Save Patch* was designed to remediate *partitioning errors*. Consequently, a student who made many *partitioning errors* during the course of the game could have held a misconception early in the game that was remediated by the end of the game, or could have held a misconception early in the game that was not remediated during the game. Thus, interpreting the number of *partitioning errors* made by a student is not straightforward. To analyze these types of errors, it is necessary to examine the sequence of behavior during the game and, therefore, analysis of those errors will be deferred to the section on sequence mining.

Student responses to the seven pretest and posttest items in Appendix D that measure the ability to unitize were recoded based on whether the response indicated an understanding of the unit. In the examples in Table 14, an answer of 2/3 demonstrates an understanding of unitizing because it shows that the student could correctly identify where one unit ended in the representation and knew to count the number of spaces in the unit to identify that the fraction was broken into thirds. The answers of 3/4 and 2/4 also demonstrate an understanding of unitizing because they show that the student could correctly identify where one unit ended, even though they incorrectly counted hash marks instead of spaces to determine the denominator of the fraction being represented. The answers of 2/10 and 3/10 demonstrate a lack of understanding of unitizing because they show that the student counted spaces all the way to the end of the representation to determine the denominator of the fraction. The answer of 2/8 also demonstrates a lack of understanding of unitizing because it shows that the student counted hash marks all the way to the end of the representation to determine the denominator of the fraction. Responses that were uninterpretable in the context of unitizing (e.g., numbers far larger or far smaller than represented on the number line in the question) were coded as missing. The percentage of non-missing responses that indicated an understanding of unitizing was then calculated for every student who had non-missing values for at least five of the seven items. Students with missing values on more than two out of seven items were not included in the analyses.

Table 14

Examples of Recoding for Question 6 in Appendix D (correct answer = 2/3)

Demonstrated an understanding of unitizing	Demonstrated a lack of understanding	Uninterpretable in the context of unitizing
2/3	2/10	28
3/4	3/10	-1
2/4	2/8	I Don’t Know

The distribution of *unitizing errors* during the game was highly skewed, with values ranging from 0 to 22, with a mean of 2.47 (27% of students made zero *unitizing errors* in the game, and 75% of students made fewer than four *unitizing errors*). Because a very small percentage of students (2.34%) made more than 10 *unitizing errors* in the game, those students were combined with students who made 10 *unitizing errors*. The final scale then ranged from 0 to 10 (where 10 indicated 10 or more).

Table 15 gives the correlations between *uniting errors* made during the game and pretest and posttest unitizing scores for the 507 students who had non-missing responses to at least five out of seven items on both the pretest and the posttest. The number of *unitizing errors* made during the game was significantly negatively correlated with pretest and posttest scores. This indicates that not only did students with lower pretest scores on unitizing items make more *unitizing errors* in the game, but that the frequency of in-game *unitizing errors* was also negatively related to posttest scores.

Table 15

Correlations Between Unitizing Errors and Test Scores

Test score	Correlation with unitizing errors	Significance
Unitizing Pretest Percentage	-.196	.000
Unitizing Posttest Percentage	-.207	.000

To illustrate the relationship between prior knowledge of unitizing (as indicated by the percentage correct on pretest unitizing items) and *unitizing errors* in the game, the distribution of the unitizing pretest percentage was broken into thirds and the mean number of in-game *unitizing errors* and 95% confidence interval was plotted for each level of pretest unitizing percentage in Figure 7. While quartiles would have been a more standard way to break up the distribution, there were so many students who did not make any unitizing errors on the pretest that breaking the distribution into quartiles would have artificially split those students into two different quartiles. This figure indicates that students who had higher prior knowledge of unitizing made fewer in-game *unitizing errors* than students with lower prior knowledge of unitizing.

Interestingly, even students who had perfect pretest unitizing percentages made on average at least one in-game *unitizing error*. This single in-game *unitizing error* is likely due to students becoming familiar with the in-game number line representation, rather than an indication of lack of understanding of unitizing.

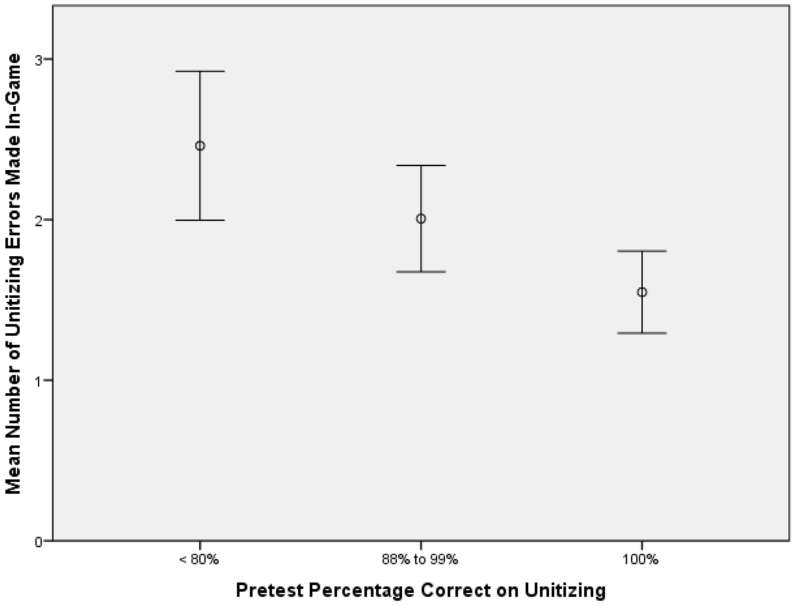


Figure 7. Mean in-game *unitizing errors* and 95% confidence intervals by pretest performance.

Similarly, the relationship between in-game *unitizing errors* and posttest knowledge of unitizing is shown in Figure 8, where the distribution of in-game *unitizing errors* was broken into quartiles and the mean posttest unitizing percentage and 95% confidence interval was plotted for each level of in-game *unitizing errors*. While the negative relationship between in-game *unitizing errors* and posttest performance is visible, there is considerable overlap in the last two categories.

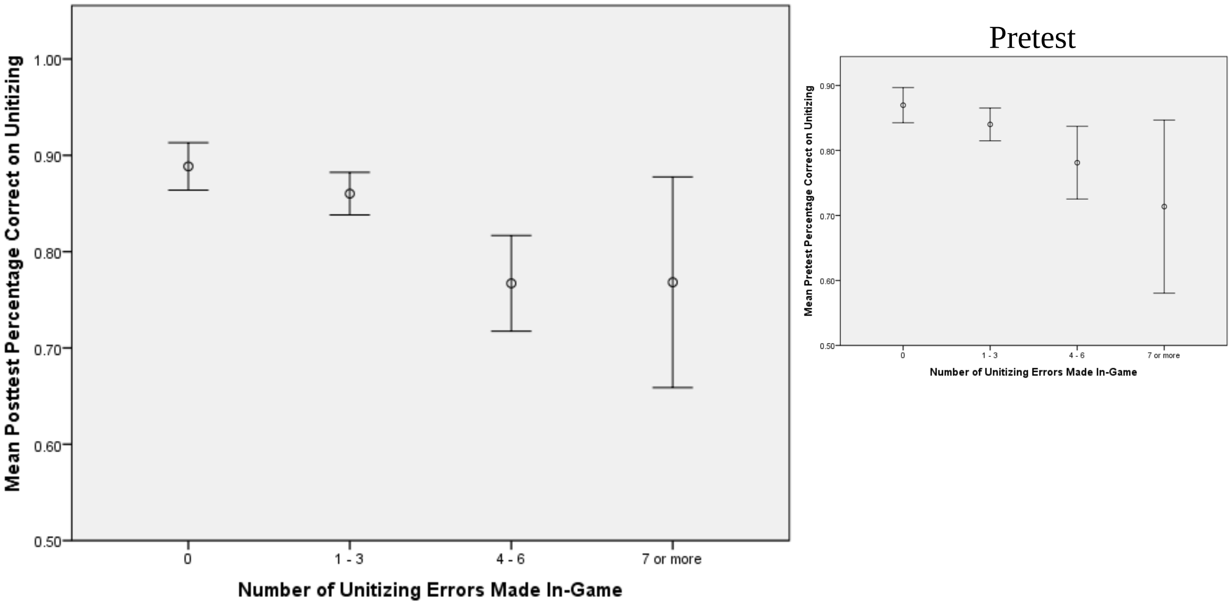


Figure 8.Unitizing posttest percentage means and 95% confidence intervals for *unitizing errors*.

In order to examine the relationship between in-game *unitizing errors* and changes in the understanding of unitizing, the percentage gain on unitizing was calculated by subtracting pretest percentages from posttest percentages. The mean posttest unitizing percentage and 95% confidence interval was plotted for each level of in-game *unitizing errors*. As shown in Figure 9, there appears to be no relationship between the number of unitizing errors made in the game and gains in unitizing transfer to the posttest.

These charts indicate that the number of *unitizing errors* students make while playing *Save Patch* seems to be indicative of both their prior understanding of unitizing and their level of understanding unitizing on the posttest. Also, as expected (since the game does not teach unitizing) the number of *unitizing errors* made in-game does not appear to have any relationship with students' gain in understanding unitizing. Importantly, Figure 9 indicates that, on average, students' understanding of unitizing did not change from pretest to posttest.

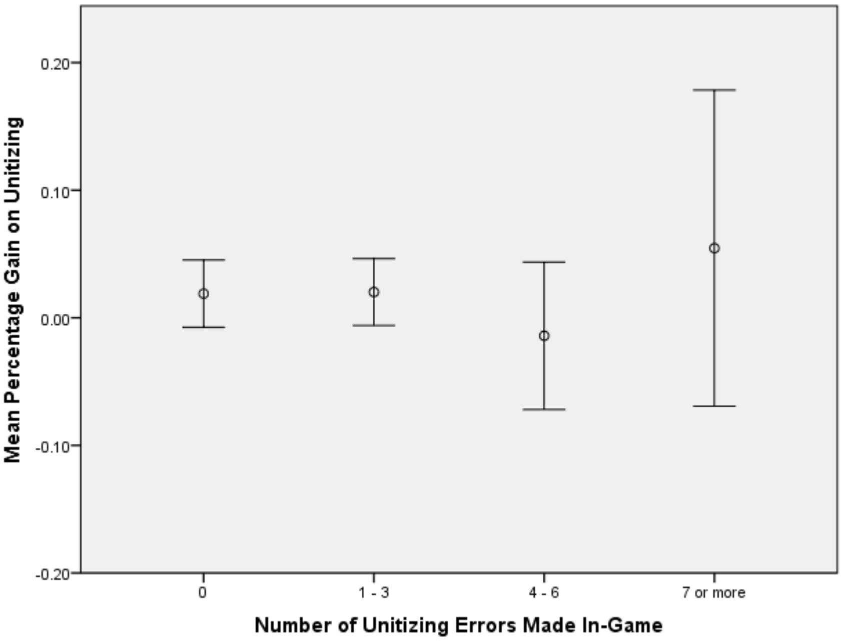


Figure 9.Unitizing percentage gain means and 95% confidence intervals for *unitizing errors*.

Information Extracted Using Sequence Mining

This section examines the following question: Can sequence mining be used to extract information leading to testable hypotheses about the relationship between in-game performance and performance on paper-and-pencil posttests? An affirmative answer to this question depends on (1) being able to identify sequences that can be characterized according to meaningful patterns of strategies made during the course of the game, and (2) identifying sequences that are large enough for statistical analysis. In this study, both conditions were satisfied.

As can be seen in Table 16, sequence mining successfully identified students in each sequence group type in each stage in *Save Patch*. For clarity, only stages with the partitioning error as the target error are shown. The most common sequence group type in Stage 2 was the *Partitioning Error to Correct* sequence group type. The most common sequence group type in Stage 3 was the *Partitioning Error to Other Error* sequence group type, followed by *Partitioning Error to Partitioning Error*. The most common sequence group type in Stage 4 was the *Correct* sequence group type, followed by the *Other Error to Correct*, *Other Error to Other Error*, and *Partitioning Error to Partitioning Error* sequence group types.

Table 16

Number of Students in Each Sequence Group Type for Each Stage

	Stages with partitioning as the target error		
Sequence group type	Stage 2	Stage 3	Stage 4
Correct	51	91	187
Correct to Partitioning Error	2	1	12
Correct to Other Error	6	28	40
Partitioning Error to Correct	510	119	47
Partitioning Error to Partitioning Error	86	154	110
Partitioning Error to Other Error	128	241	97
Other Error to Correct	49	118	127
Other Error to Partitioning Error	5	64	74
Other Error to Other Error	7	22	118

The *Correct to Partitioning Error* sequence group type was uncommon, consisting of fewer than 15 students in all stages. The *Correct to Other Error* sequence group type was also uncommon, with fewer than 50 students in all three stages, and fewer than 10 students in Stage 2. The *Other Error to Other Error* sequence group type was uncommon in Stage 2 and Stage 3, and the *Other Error to Partitioning Error* sequence group type was uncommon in Stage 2.

Developing and Testing a Hypothesis Using Sequence Mining Results

This section addresses the following question: Can the information produced by the sequence mining be used to diagnose student errors related to fractions understanding? To investigate this question, analyses were performed to determine the relationship between specific sequences of strategies made during game play and performance on posttest items measuring proficiency in that area.

Student in-game partitioning behavior for the three stages that were designed to address partitioning was recoded into a single summary of behavior over all three levels for each student. Students were categorized into one of six categories according to their sequences related to partitioning. Students who got all levels in all three stages correct on the first attempt were coded as *All Correct*. Students who were never in a strategy sequence group type related to partitioning were coded as *No Partitioning Errors*. Students who were in strategy sequence group types related to partitioning in early stages, but got all levels of the last stage on partitioning correct on the first attempt were coded as *Corrected Partitioning* (that is, they made partitioning errors in early stages but not in the last stage). Students who were in strategy sequence group types related to partitioning in early stages, but were not in strategy sequence group types involving either the correct solution or partitioning in later stages were coded as *Abandoned Partitioning* (that is, these students made errors in later stages, but those errors were not partitioning errors). Students who were not in a strategy sequence group type involving partitioning in early stages, but were in later stages were coded as *Some Partitioning* (that is, they did not make partitioning errors in early stages, but did exhibit partitioning errors in later stages). Finally, students who were in strategy sequence group types involving partitioning in all three stages addressing partitioning were coded as *Repeated Partitioning*.

Student responses to the seven pretest and posttest items in Appendix D that measure the ability to partition were recoded based on whether the response indicated an understanding of how to determine the denominator of a fraction (even if the answer was not correct), that is, understanding of partitioning. In the examples in Table 17, the answer of 2/3 demonstrates an understanding of partitioning because it shows that the student correctly counted the number of spaces in each unit to determine the denominator of the fraction. The answers of 1/3 and 3/3 also demonstrate an understanding of partitioning because they show that the student correctly counted the number of spaces to determine the denominator of the fraction, even though they were not able to correctly determine the numerator of the fraction. The answer of 2/2 demonstrates a lack of understanding of partitioning because it shows that the student counted the number of hash marks between zero and one to determine the denominator, instead of counting the number of spaces. Similarly, the answers of 2/4 and 3/4 demonstrate a lack of understanding of partitioning because they show that the student counted the number of hash marks, including the marks at zero and at one, to determine the denominator. Responses that were uninterpretable in the context of partitioning (e.g., the answer did not include a denominator) were coded as missing. The percentage of non-missing responses that indicated an understanding of partitioning was then calculated for every student who had non-missing values for at least five of the seven items.

Table 17

Examples of Recoding for Question 6 in Appendix D (Correct Answer = 2/3)

Demonstrated an understanding of partitioning	Demonstrated a lack of understanding	Uninterpretable in the context of partitioning
2/3	2/2	28
1/3	2/4	0
3/3	3/4	I Don’t Know

To examine the relationship between in-game partitioning behavior and student scores on pretest items involving partitioning, the mean pretest partitioning percent and 95% confidence interval around the mean was plotted for each of the in-game partitioning categories in Figure 10. This figure shows that students with lower prior knowledge of partitioning made in-game partitioning errors that they did not correct by the end of the game (*Abandoned Partitioning*, *Some Partitioning*, and *Repeated Partitioning*). Additionally, students with the lowest prior knowledge of partitioning continued to make partitioning errors on all partitioning-relevant stages in the game (*Repeated Partitioning*). In contrast, students with high prior knowledge of partitioning did not make partitioning errors in the game (*All Correct* or *No Partitioning Errors*) or made errors early in the game but corrected those errors by the end of the game (*Corrected Partitioning*).

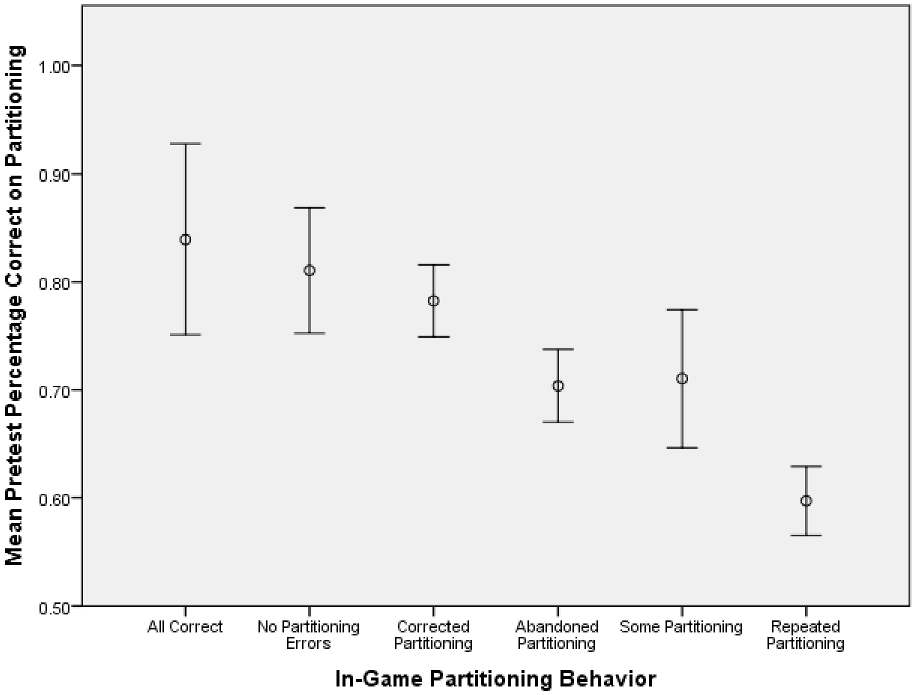


Figure 10. Mean pretest partitioning percentage by in-game partitioning behavior.

The relationship between in-game partitioning behavior and pretest/posttest changes in knowledge of partitioning is shown in Figure 11. Table 18 gives the corresponding means and standard deviations. While all posttest means are approximately 5% higher than their corresponding pretest means, the relative positioning of different in-game partitioning behaviors is largely unchanged. Pretest means ranged from 60% to 84% and posttest means ranged from 66% to 89%. There was on average a 6.4% gain ($p < .001$) in partitioning after playing *Save Patch*, regardless of in-game partitioning behavior.

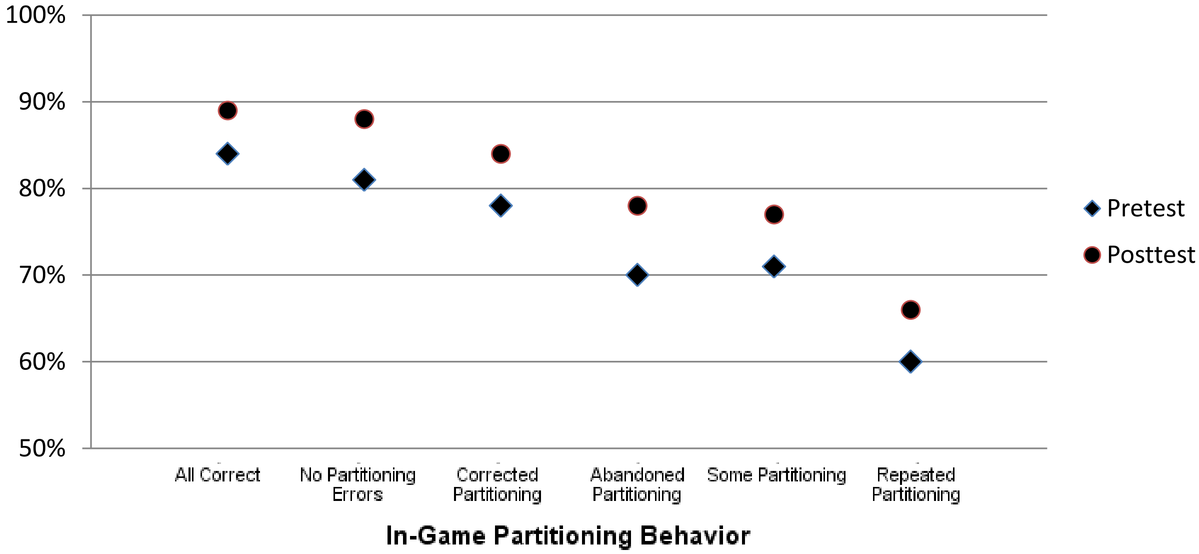


Figure 11.Mean pretest and posttest partitioning percentage by in-game partitioning behavior.

Table 18

Test Means and Standard Deviations by In-Game Partitioning Behavior

		Partitioning pretest		Partitioning posttest		Partitioning gain	
Partitioning behavior	*n*	Mean	*SD*	Mean	*SD*	Mean	*SD*
All Correct	14	.84^{b}	.15	.89^{b}	.12	.05	.09
No Partitioning Errors	47	.81^{a,b}	.20	.88^{a,b}	.16	.07	.15
Corrected Partitioning	120	.78^{a,b}	.18	.84^{a,b}	.17	.06	.18
Abandoned Partitioning	141	.70^{b,c}	.20	.78^{b,c}	.20	.07	.20
Some Partitioning	44	.71^{b}	.21	.77^{b}	.20	.06	.21
Repeated Partitioning	151	.60^{d}	.20	.66^{d}	.20	.06	.21

*Note.*Partitioning behaviors with the same subscript have mean scores that are not significantly different from each other. Partitioning behaviors with different subscripts have mean scores that are significantly different.

An ANCOVA testing for differences in posttest score controlling for pretest score was planned. However, Figure 10 indicates that these data are not appropriate for an ANCOVA because a number of the groups do not have overlapping scores on the covariate (e.g., the *Repeated Partitioning* group has pretest scores far below the other groups). This means, per

Miller and Chapman (2001), that controlling for the covariate is not appropriate as students in the different groups clearly differ substantially on the covariate. Therefore, any significant differences found in an ANCOVA would be due to the largely non-overlapping pretest scores of students in different partitioning behavior categories rather than to actual differences in gains.

ANOVAs were run testing for significant differences between in-game partitioning behavior on pretest and posttest scores, as well as gain scores. The ANOVAs for pretest and posttest were significant at $p < .001$. The Bonferonni-adjusted posthoc tests are shown in Table 18 where lettered superscripts indicate similarities and differences between partitioning behaviors.

The ANOVAs showed that students in *Repeated Partitioning* had significantly lower pretest and posttest scores than all other in-game partitioning behaviors. Students in *All Correct* and *Some Partitioning* had significantly higher pretest and posttest scores than *Repeated Partitioning*, but were not significantly different from any other in-game partitioning behaviors. Students in *Abandoned Partitioning* had significantly higher pretest and posttest scores than *Repeated Partitioning* and significantly lower pretest and posttest scores than *Corrected Partitioning* and *No Partitioning Errors*, which had significantly higher pretest and posttest scores than all other in-game partitioning behaviors (except *All Correct* and *Some Partitioning*). The ANOVA for gain scores was not significant ($p = .996$).

These results indicate that there was an overall increase in partitioning performance between the pretest and posttest, which was expected since the game was designed to teach partitioning. However, the amount of the increase was essentially the same for all in-game partitioning behavior categories. The pattern of results seems to suggest the following possible interpretations. (1) Students in the *All Correct* and *No Partitioning Errors* in-game categories had a pretty good understanding of partitioning on the pretest and the game solidified their understanding so that they performed a little better on the posttest. (2) Students in the *Corrected Partitioning* category showed slightly less understanding of partitioning at the outset than the *All Correct* and *No Partitioning Errors* categories and improved their performance by a small amount through game play. (3) Students who made partitioning errors early in the game and made other errors later in the game (*Abandoned Partitioning*) and students who did not make partitioning errors early in the game but started making partitioning errors later in the game (*Some Partitioning*) had less understanding of partitioning at the outset and increased their scores by a slight amount. (4) Students who made partitioning errors throughout the game (*Repeated Partitioning*) had the least understanding of partitioning at the outset, and improved their performance by a slight amount.

The fact that the posttest means track the pretest means so closely, in conjunction with the nearly equal gains for each in-game category, suggests that posttest scores may reflect pretest understanding of partitioning more than any influence of game play on partitioning understanding. That is, it would be incorrect to conclude, for example, that the game was more effective for remediating partitioning misconceptions for students who exhibited *Corrected Partitioning* than for students who exhibited *Repeated Partitioning*.

However, the results do suggest that it may be possible to interpret patterns of game play as indicative of prior understanding of partitioning, which may have useful implications. For example, additional instruction that focuses on partitioning could be targeted to some groups according to their pattern of game play. Students who continued to make partitioning errors throughout the game (*Repeated Partitioning*) seem to have substantial partitioning misconceptions prior to game play (that were largely unremediated during the game) and therefore seem to be especially good candidates for additional instruction. Students in the *Abandoned Partitioning* and *Some Partitioning* categories may also benefit from additional instruction. On the other hand, students in the *All Correct* and *No Partitioning Errors* categories, and to some extent students in the *Corrected Partitioning* categories, likely need the least additional instruction.

Information Extracted Using Classification

This section examines the following question: Can classification be used to extract information leading to testable hypotheses about the relationship between in-game performance and performance on paper-and-pencil posttests? An affirmative answer to this question depends on (1) being able to accurately classify students based on changes in performance during the course of the game, and (2) identifying classes that are large enough for statistical analysis. In this study, both conditions were satisfied.

The performance trajectory types were: *All Correct*, *Only One Mistake, Improved To 1*, *Partially Improved*, *Steady Decline*, and *Unsteady Decline* (see Figure 6 in the Methods section). Unlike the strategy sequence types, the performance trajectory types do not take into account the type of mistakes students made. That is, for example, students who showed improvement (requiring fewer attempts to solve later levels than earlier levels) did not necessarily make fewer partitioning errors in later levels than earlier levels. The number of attempts contains no information regarding the type of mistakes made and differentiations based on error type such as those between the *Partitioning Error To Correct* and *Other Error To Correct* sequence group types cannot be made using performance trajectory types. However, the number of attempts required to solve a level is still useful as it is a quick and easy approximation of performance. For example solving the level in a single attempt likely indicates mastery of the content, solving the level in two or three attempts likely indicates an error the student could correct, and requiring 10 or 15 attempts to solve a level likely indicates the student got the correct answer by chance or guessing.

In order to determine the accuracy of the classification process, two measures of agreement between the classification results and the performance trajectory types assigned by human raters were calculated. The first measure of agreement was exact agreement, which shows the extent to which the classification processes and the human rater assigned the same individual trajectories to the same performance trajectory type. The second measure of agreement was Cohen's kappa, which shows exact agreement, accounting for the probability of agreeing based on chance.

Table 19

Comparing Classification Results to a Human Rater

	Measures of agreement with a human rater	
Stage	Exact agreement	Cohen's kappa
Stage 1	0.796	0.729
Stage 2	0.819	0.690
Stage 3	0.833	0.788
Stage 4	0.967	0.959
Stage 5	0.902	0.845
Stage 6	0.816	0.769

As seen in Table 19, classification using one nearest neighbor showed a high degree of agreement with the categorization performed by the human rater. Exact agreement between the classification algorithm and a human rater was high, ranging from 0.796 in Stage 1 to 0.967 in Stage 4. Additionally, Cohen's kappa was fairly high, ranging from 0.690 in Stage 2 to 0.959 in

Stage 4. Cohen’s kappa values over 0.75 are generally considered to indicate substantial agreement (Cohen, 1960).

The number of students in each performance trajectory type in each stage can be seen in Table 20. The most common performance trajectory type in Stage 1: Whole Numbers, Stage 2: Unit Fractions, Stage 5: Proper Fractions, and Stage 6: Improper Fractions was the *Improved to 1* performance trajectory type. The most common performance trajectory type in Stage 3: Whole Numbers and Unit Fractions was the *Unsteady Decline* performance trajectory type. The most common performance trajectory type in Stage 4: Wholes Across the Unit Mark was *Steady Decline*.

Table 20
Number of Students in Each Performance Trajectory Type for Each Stage

	Stage					
Trajectory type	1	2	3	4	5	6
Steady Decline	130	103	105	236	10	61
Unsteady Decline	45	0	234	93	51	107
Partially Improved	141	90	156	153	33	93
Improved to 1	379	519	196	112	437	209
Only One Mistake	1	64	56	55	121	117
All Correct	156	66	82	162	138	169

Unsteady Decline was one of the least represented performance trajectory types in each stage (with the notable exception of Stage 3). The *Only One Mistake* performance trajectory type had only a few students in early stages, but much larger numbers of students in later stages. The *Steady Decline*, *Partially Improved*, and *All Correct* performance trajectory types ranged in frequency between stages, with some stages having far more students than others. The *Improved* *to 1* performance trajectory type was the only performance trajectory type to have a large number of students in every stage of the game.

Developing and Testing a Hypothesis Using Classification Results

This section addresses the following question: Can the information produced by classification be used to make interpretations about student understanding of fractions? To investigate this question, analyses were performed to determine the relationship between the frequency of specific performance trajectory types made during game play and performance on the paper-and-pencil pretest and posttest.

Student in-game performance trajectories were recoded into a single summary of behavior over all game levels for each student. Students were categorized into one of six categories according to their performance trajectory types. Students who got all or almost all levels correct on the first attempt (indicated by being in the *All Correct* or *Only One Mistake* performance trajectory types in all stages) were labeled as *All Mastery*. Students who were in the *All Correct* or *Only One Mistake* performance trajectory types in early stages but not in later stages were labeled as *Declined From Mastery*. Conversely, students who were in the *All Correct* or *Only One Mistake* performance trajectory types in later stages but not in earlier stages were labeled as *Improved To Mastery*. Students whose performance trajectory types got consistently better (e.g., *Steady Decline* to *Unsteady Decline* to *Partially Improved*) but were not in the *All Correct* or *Only One Mistake* performance trajectory types were labeled as *Showed Improvement*, and students whose performance trajectory types got consistently worse but did not start in the *All Correct* or *Only One Mistake* performance trajectory types were labeled as *Showed Decline*. Finally, students whose performance trajectory types showed no clear pattern were labeled as *Mixed*.

As opposed to earlier analyses, the percentage of correct pretest and posttest items for this portion of the study were calculated using all pretest and posttest items in Appendix D, rather than only items addressing a specific concept, because performance trajectories only reflect overall performance.

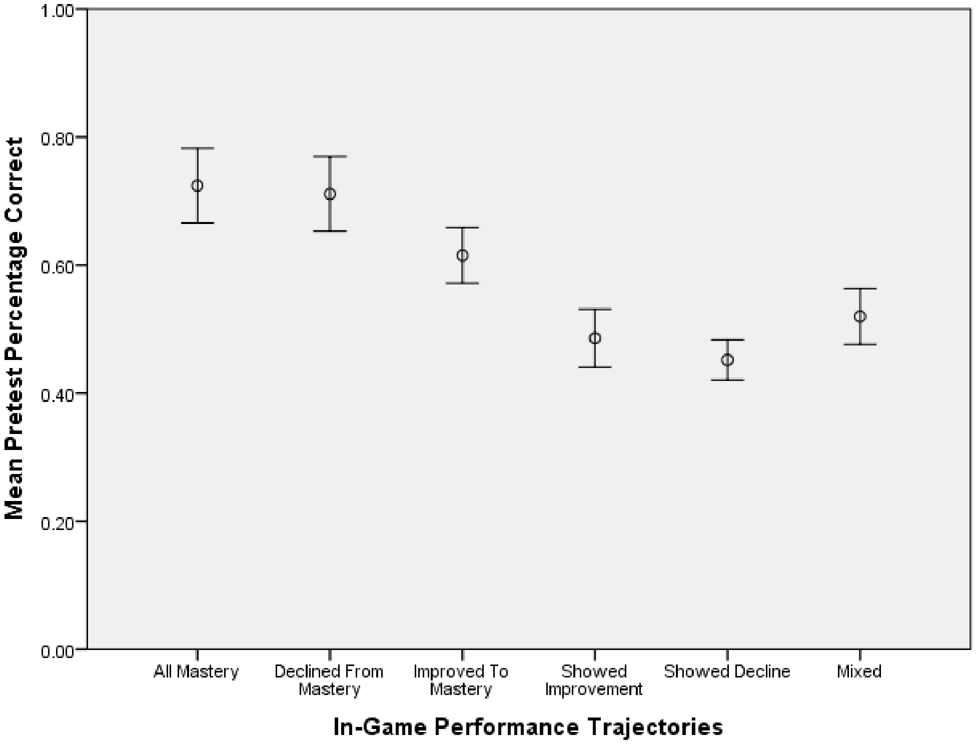


Figure 12. Mean pretest percentage by in-game performance trajectory.

To examine the relationship between in-game performance trajectories and students' pretest scores, the mean pretest percentage and 95% confidence interval around the mean were plotted for each of the in-game performance trajectories in Figure 12. This figure indicates that students with high prior knowledge of fractions exhibited in-game performance trajectories indicating initial mastery (*All Mastery* and *Declined From Mastery*). Students with moderately high prior knowledge of fractions exhibited in-game performance trajectories that did not begin with mastery, but did end with mastery (*Improved To Mastery*). Additionally, students with the lowest prior knowledge of fractions did not achieve mastery at any point in the game (*Showed*

Improvement or *Showed Decline*). The *Mixed* performance trajectory type cannot be substantively interpreted, but is shown for completeness.

The relationship between in-game performance trajectory and pretest/posttest changes in knowledge of fractions is shown in Figure 13. Table 21 gives the corresponding means and standard deviations. While all posttest means are approximately 10% higher than their corresponding pretest means, the relative positioning of different in-game performance trajectories is largely unchanged. Pretest means ranged from 45% to 72% and posttest means ranged from 53% to 89%. There was on average a 10.5% gain (p< .001) in fractions scores after playing *Save Patch*, regardless of in-game performance trajectory.

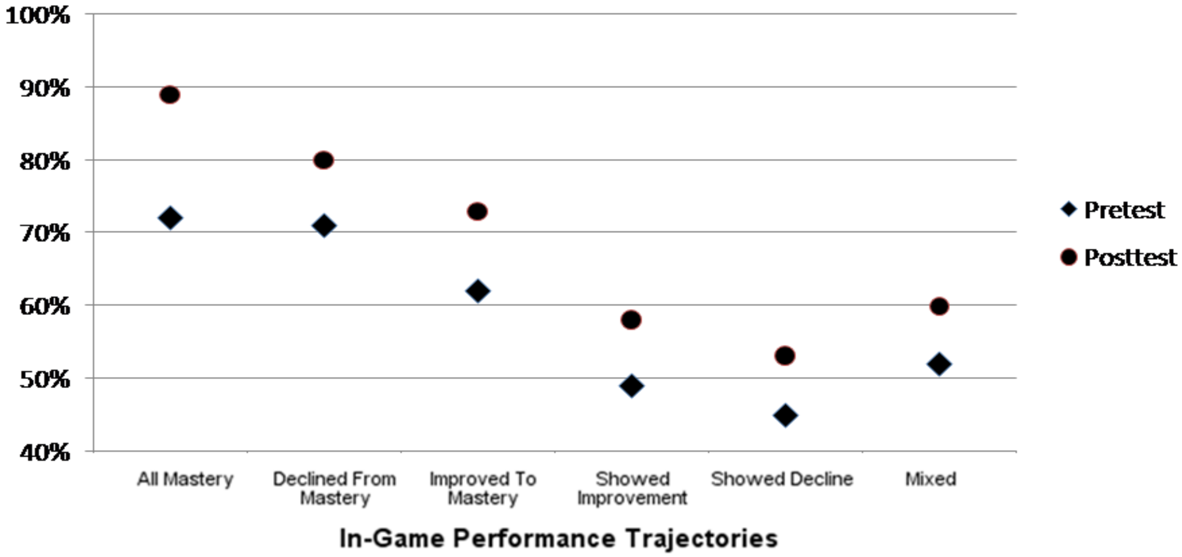


Figure 13. Mean pretest and posttest percentage by in-game performance trajectory.

ANOVAs were run testing for significant differences between in-game partitioning behavior on pretest and posttest scores, as well as gain scores. The ANOVAs for pretest and posttest were significant at p< .001. The Bonferonni-adjusted posthoc tests are shown in Table 21, where lettered subscripts indicate similarities and differences between in-game performance trajectories. The ANOVAs showed that students in the *All Mastery*, *Declined From Mastery*, and

Improved to Mastery performance trajectories had significantly higher pretest and posttest scores than students in the other performance trajectories. However the ANOVA for gain scores was not significant (*p* = .063).

Table 21

Test Means and Standard Deviations by In-Game Performance Trajectory

		Pretest		Posttest		Gain	
Performance trajectory	*n*	Mean	*SD*	Mean	*SD*	Mean	*SD*
All Mastery	47	.72^{a}	.20	.89^{a}	.20	.17	.19
Declined From Mastery	51	.71^{a}	.21	.80^{a}	.22	.09	.16
Improved To Mastery	81	.62^{a}	.20	.73^{a}	.20	.11	.18
Showed Improvement	94	.49^{b}	.22	.58^{b}	.26	.09	.16
Showed Decline	140	.45^{b}	.19	.53^{b}	.23	.08	.16
Mixed	105	.52^{b}	.23	.60^{b}	.24	.08	.17

Note. Performance trajectories with the same subscript have mean scores that are not significantly different from each other. Performance trajectories with different subscripts have mean scores that are significantly different.

These results indicate that there is an overall increase in performance between the pretest and posttest, but the amount of the increase was not significantly different between performance trajectory types. The pattern of results seems to suggest the following possible interpretations. (1) Students in the *All Mastery* and *Declined From Mastery* performance trajectories had pretty good understanding of fractions on the pretest and the game solidified their understanding so that they performed better on the posttest. (2) Students in the *Improved To Mastery* trajectory showed slightly less understanding of fractions at the outset than the *All Mastery* and *Declined From Mastery* categories and improved their performance through game play. (3) Students in the *Showed Improvement*, *Showed Decline,* and *Mixed* trajectories had less understanding of fractions at the outset and increased their scores by a small amount.

As with the sequence mining results, posttest scores may reflect pretest understanding of fractions more than any influence of game play on fractions understanding. That is, it would be incorrect to conclude that in-game improvement in the form of the *Showed Improvement* or *Improved To Mastery* performance trajectories resulted in greater than average improvement between the pretest and the posttest.

However, as with sequence mining, the results do suggest that it may be possible to interpret performance trajectories as indicative of prior understanding of fractions. For example, students in the *Showed Improvement*, *Showed Decline*, and *Mixed* performance trajectories seemed to have a low understanding of fractions prior to game play (as indicated by their low pretest scores) that was largely unremediated during the game (as indicated by their low posttest scores). Although these students seem to be especially good candidates for additional instruction, the performance trajectory results do not indicate the specific area in which students could most benefit from additional instruction.

Comparison of Classification and Sequence Mining Results

Although the different data mining techniques revealed different kinds of insights into students' in-game performance (e.g., sequence mining gives insights into partitioning behavior, while classification gives insights into general performance), an important issue is whether the different data mining techniques categorized students differently. If students exhibiting a certain partitioning behavior (from the sequence mining) all showed a particular trajectory of general performance (from the classification), then the argument could be raised that the two data mining approaches were simply giving rise to different labels for the same group of students. Conversely, the lack of a one-to-one correspondence between the categorization of students using sequence mining and the categorization of students using classification would suggest that the data mining techniques are giving rise to different aspects of in-game performance.

To examine this issue, Table 22 shows the number of students who were categorized into each of the trajectories identified through the use of sequence mining and each of the trajectories identified through the use of classification.

Table 22

Comparison of Classification and Sequence Mining Results

	Classification results					
Sequence mining results	All Mastery	Declined From Mastery	Improved To Mastery	Showed Improvement	Showed Decline	Mixed
All Correct	12	2	0	0	0	0
No Partitioning Errors	8	22	10	5	2	10
Corrected Partitioning	23	10	60	9	3	15
Abandoned Partitioning	0	5	13	39	35	8
Repeated Partitioning	0	2	0	35	74	40
Some Partitioning	4	10	0	8	13	9

The results in Table 22 show that there is not a one-to-one correspondence between trajectories arising from the sequence mining and trajectories arising from the classification. For example, of the 47 students who never made partitioning errors during the game (*No Partitioning Errors* in the sequence mining results), only 8 (17%) were categorized as *All Mastery* using the classification data mining approach. The other 83% of students who never made partitioning errors showed improvement or declined in other respects. Similarly, of the 120 students who corrected partitioning errors over the course of the game (*Corrected Partitioning*), 50% were categorized in the trajectory *Improved to Mastery* in terms of general performance from the classification results. Many of the other students (31%) showed improvement (but not to mastery) or showed decline or showed mixed results. As a third example, of the 151 students who made partitioning errors throughout the game (*Repeated Partitioning*), 35 students (23%) showed improvement in general performance from the classification results (*Showed Improvement*), 74 students (49%) showed a decline in general performance from the classification results (*Showed Decline*), and 40 students (26%) showed mixed general performance (*Mixed*).

CHAPTER5: DISCUSSION

This study examined 1.2 million rows of log data generated by 855 students from 31 classes playing an educational video game about fractions called *Save Patch*. This study successfully used three different educational data mining techniques (cluster analysis, sequence mining, and classification) to extract interpretable information from the game log data that could be used to form testable hypotheses. In this study, the hypotheses generated and tested concerned the relationship between in-game performance and performance on more traditional measures of content understanding (here, paper-and-pencil measures of fractions knowledge). Developing methods of extracting interpretable information from game log data is vital because without the ability to analyze in-game performance it is difficult if not impossible to use games or simulations as an indication of what students may know or understand. Testing the relationship between in-game performance and performance on more traditional measures of content understanding is important as a first step towards validating the information extracted from game log data.

The game in this study was *Save Patch*, an educational video game about the identification of fractions. The game uses a representation similar to a number line and requires students to place fractional pieces (e.g., 1/3, 1/4, etc.) on sign posts located at various positions on the number line (e.g., 3/4) to move the game character successfully from one sign post to another in order to solve each level in the game. The levels in the game are organized in stages. All levels in a given stage address the same fractions content, and the stages progress in complexity throughout the game. Some of the sign posts in the game are required, others are optional, and all signs have a directional indicator that can be changed to make the game character move the indicated distance in the desired direction. There is no limit to the number of times a student can attempt to solve a given level, but each level must be solved before the student can move on to the next level.

Summary and Discussion of Cluster Analysis Results

Using cluster analysis, individual student actions in each level were grouped into nameable action sets representing the different strategies students used to try to solve the game. This process led to the identification of valid solution strategies, strategies representing specific game-related errors, and strategies corresponding to specific mathematical misconceptions.

There were six different types of valid solution strategies identified by the cluster analysis, accounting for 45.5% of attempts to solve levels in the game. The most common valid solution strategy was the standard solution (wherein students solved the level as anticipated, using the correct fractional pieces for fractional distances and whole unit pieces for whole unit distances), followed by the fractional solution (wherein students used fractional pieces for whole unit distances, such as 4/4, instead of using whole unit pieces).

There were three different types of game-related errors, accounting for 6.9% of attempts to solve levels in the game. The most common game-related errors occurred either because students skipped a required stop (e.g., placing 2/3 on the first sign instead of placing 1/3 on the first sign and 1/3 on the second sign as required by the game) or students placed all fractional pieces correctly but moved their game character in the wrong direction (usually because they forgot to change the direction of the sign from its default position).

There were five different types of strategies corresponding to specific mathematical misconceptions, accounting for 29.4% of attempts to solve levels in the game. The most common strategy corresponding to a mathematical misconception was the *partitioning error*, wherein students identified the denominator of the fraction incorrectly because they counted the number of dividing marks to determine the denominator rather than counting the number of pieces the unit was broken into. Other common mathematical misconceptions were the *iterating error* (wherein students determined the denominator correctly but were unable to determine the numerator of the represented fraction) and the *unitizing error* (wherein students were unable to determine the number of units represented because they assumed that the entire representation consisted of a single unit, regardless of the actual number of units represented).

Extracting substantively meaningful information from game log data is a significant challenge (Frezzo et al., 2009; Garcia et al., 2011; Mislevy et al., 2004; Mostow et al., 2011; National Research Council, 2011). Using fuzzy feature cluster analysis to identify in-game strategies rose to this challenge, resulting in substantively meaningful categorizations of 82% of all attempts made to solve problems in the game.

The identification of substantively meaningful in-game strategies accounting for a majority of attempts to solve levels in the game allows for the formulation and testing of hypotheses that were not feasible prior to the extraction of this information. Without access to information about how students solve problems in the game, the game is a black box which cannot be directly assessed. Opening up the black box by the identification of in-game strategies allows not only for the formation of hypotheses about the relationship between in-game performance and pretest and posttest performance (as addressed in this study) but about a variety of other relationships as well. For example, a number of hypotheses can now be formulated about the relationship between background variables and in-game strategy use in order to determine whether the game is a fair measure for all students (e.g., Are students with little gaming experience more likely to make game-related errors?), the relative effect of cluster membership on in-game learning in order to determine areas in need of remediation (e.g., Do students who make *partitioning errors* in the game have lower posttest scores controlling for pretest scores than students who make *unitizing errors*?), or the effects of different classroom instructional techniques on in-game performance in order to make practical recommendations for classroom instruction (e.g., Are students who were taught fractions using circular representations more likely to make in-game *partitioning errors* than students who were taught using linear representations?).

The specific hypotheses generated from the cluster analysis results involved the relationship between errors made in the game (as identified by the cluster analysis) and the same errors made on paper-and-pencil measures to determine whether both behaviors were indicative of the same underlying misunderstanding. The specific relationship examined was between in-game *unitizing errors* and pretest and posttest scores on items requiring an understanding of unitizing. Because the game was not designed to remediate *unitizing errors*, the relationships between pretest, in-game, and posttest performance were expected to be strong and there was expected to be no change in performance on unitizing between the pretest and the posttest.

This expectation was borne out. The number of *unitizing errors* made in-game was significantly correlated with both pretest and posttest unitizing performance, but there was not a significant change in performance between the pretest and the posttest, and the number of *unitizing errors* made in-game was not correlated with gain scores. Students who had high prior knowledge of unitizing made few if any in-game *unitizing errors* and had relatively high posttest scores. On the other hand, students who had low prior knowledge of unitizing made a fairly large number of in-game *unitizing errors* and had relatively low posttest scores.

The finding that strategies as an in-game measure of performance were significantly related to both pretest and posttest scores has implications for the use of educational video games as assessments. This finding indicates that educational technology can potentially produce valid measures of student understanding and makes it possible to speculate that game or simulation components might be able to play a role in large-scale, high-stakes standardized testing environments. The U.S. Department of Education (2010, 2012) has called for research into the use of games and simulations as assessments of the complex skills delineated in state and national standards, and the findings of this study indicate that such research may be fruitful.

Additionally, the identification of in-game strategies allows for the development of a deeper understanding of how students solve different kinds of problems, as Bejar (1984), Rahkila and Karjalainen (1999), Merceron and Yacef (2004), and Quellmalz and Pellegrino (2009) posited would be possible if the black box of in-game performance could be opened. First, the identification of a variety of correct solution strategies provides detail about how successful students reached their answers, rather than just identifying such students as correctly solving the game levels. Second, the identification of a variety of in-game strategies reflecting mathematical misconceptions provides detail about why and how students answer incorrectly that could be used to provide targeted remediation of the content (e.g., providing information about how to identify the denominator to students who make *partitioning errors* and providing information about how to identify the numerator to students who make *iterating errors*) or support diagnostic claims about student understanding (e.g., identifying not just that a given student does not know the correct answer, but determining which specific skills the student lacks). Finally, the identification of in-game strategies reflecting game errors rather than mathematical misconceptions allows for the separation of the impact of the format of the problem from student understanding of the content area. For example, students using the *wrong direction* strategy were mathematically correct but did not solve the level on that attempt because they made a game-related error.

Summary and Discussion of Sequence Mining Results

Using sequence mining, frequent sequences of strategies students used to solve levels in each stage were identified to provide a summary of student behavior in each stage of the game. This process led to the identification of strategy sequence types reflecting continuous use of the valid solution strategies, movement from one erroneous strategy to another erroneous strategy, and movement from erroneous strategies to valid solution strategies. Examination of strategy sequences across levels addressing partitioning (by far the largest portion of the game) led to the identification of different types of partitioning behavior in the game.

There were very few students who used only valid solution strategies in all attempts at all levels of the game addressing partitioning (the *All Correct* sequence strategy, consisting of 3% of students). There were two different types of strategy sequences reflecting movement from one erroneous strategy to another erroneous strategy. Students either repeatedly made partitioning errors (*Repeated Partitioning*, 29% of students) or moved from making partitioning errors to errors that did not involve partitioning (*Abandoned Partitioning*, 27% of students). There were two different types of strategy sequences reflecting movement from erroneous strategies to valid solution strategies. In levels addressing partitioning, students either moved from partitioning errors to valid solution strategies as they moved through the stages (*Corrected Partitioning*, 23% of students) or moved from errors that did not involve partitioning to valid solution strategies as they moved through the stages (*No Partitioning Errors*, 9% of students). Overall, using sequence mining to identify in-game strategy sequences resulted in substantively meaningful categorizations of 91% of in-game behavior.

The identification of substantively meaningful in-game strategy sequences accounting for a majority of attempts to solve levels in the game allows for the formulation and testing of hypotheses that were not feasible prior to the extraction of this information. The identification of in-game strategy sequences allows not only for the formation of hypotheses about the relationship between in-game performance and pretest and posttest performance (as addressed in this study) but about a variety of other relationships as well. For example, remedial instruction could be provided between stages addressing the same targeted concepts and in-game strategy sequences in the stage before the instruction and after the instruction could be examined to determine whether students moved from the strategy sequence representing repetition of the targeted error to the strategy sequence representing correcting the error. This could allow for the testing of the effectiveness of various forms of instruction (e.g., Does text instruction or video instruction result in the most students moving to the strategy sequence representing correcting the error?) or specific wording (e.g., Does using the word “unit” or using the word “whole” in the instruction result in the most students moving to the strategy sequence representing correcting the error?).

The specific hypotheses generated from the sequence mining results in this study involved the relationship between in-game behavior (as identified by the sequence mining) and performance on paper-and-pencil measures to determine whether in-game behavior was indicative of student understanding of the content. The specific relationship examined was between in-game partitioning behavior and pretest and posttest scores on items requiring an understanding of partitioning. In-game partitioning behavior was chosen rather than in-game unitizing or iterating behavior because a large number of stages targeted partitioning, while none targeted unitizing and only one targeted iterating. Therefore there was expected to be a relationship between in-game partitioning behavior and pretest and posttest performance, and certain behaviors (such as the *Corrected Partitioning* behavior) were expected to correspond to gains in scores from pretest to posttest.

A significant relationship between in-game partitioning behavior and pretest and posttest performance was found. Students with the lowest levels of prior knowledge of partitioning continued to make partitioning errors in all three stages of the game addressing partitioning. Students with the next lowest prior knowledge of partitioning made in-game partitioning errors that they did not correct by the end of the game. In contrast, students with high prior knowledge of partitioning either did not make partitioning errors in the game or made partitioning errors early in the game and corrected those errors by the end of the game.

However, there was no significant relationship between in-game partitioning performance and gains in scores from pretest to posttest. There was a significant gain of about 6% from pretest to posttest, but the amount of gain did not differ based on in-game partitioning behavior.

The findings that strategy sequences as an in-game measure of performance were significantly related to both pretest and posttest performance has a number of positive implications for the use of educational video games as assessments. In a game addressing multiple targeted concepts (such as partitioning, unitizing, and iterating), analysis of in-game behavior could allow for the identification of the strengths and weaknesses of individual students in each targeted concept (as posited in Mehrens, 1992). Additionally, examining in-game strategy sequences could allow game designers to provide individualized in-game feedback for students who continue to struggle with a specific concept (Brown et al., 2008) or create instruction in additional stages that adapts to the changing needs of each student (Bejar, 1984; Clark et al., 2009; Radatz, 1979).

Summary and Discussion of Classification Results

Using classification, the change in the number of attempts required to solve each level as students move through each stage was plotted to form performance trajectories for each student in each stage of the game. Examination of performance trajectories across stages led to the identification of performance trajectory types indicating consistently good performance throughout the game, performance that improved over time, and performance that declined over time.

There were few students who demonstrated consistently good performance throughout the game (the *All Mastery* sequence strategy, consisting of 9% of students). There were two different types of performance trajectories reflecting performance that improved over time. Students either required a large number of attempts to solve earlier levels and solved later levels in only one attempt (*Improved To Mastery*, 16% of students) or required a large number of attempts to solve earlier levels and solved later levels in fewer attempts, without solving any levels in only one attempt (*Showed Improvement*, 18% of students). There were also two different types of performance trajectories reflecting performance that declined over time. Students either solved earlier levels in only one attempt but required multiple attempts to solve later levels (*Declined From Mastery*, 10% of students) or required multiple attempts to solve early levels and required more attempts to solve later levels (*Showed Decline*, 27% of students). Overall, using classification to identify in-game performance trajectories resulted in substantively meaningful categorizations of 80% of in-game behavior.

The identification of substantively meaningful in-game performance trajectories accounting for a majority of attempts to solve levels in the game allows for the formulation and testing of hypotheses that were not feasible prior to the extraction of this information. For example, the percentage of students falling in each performance trajectory type for two different concepts could be examined to determine which is more difficult for students to learn (e.g., Are there significantly more students in the *Improved To Mastery* trajectory type in the stage consisting of proper fractions than in the stage consisting of improper fractions?) or the order of stages could be changed and the resulting performance trajectories examined to determine which ordering is more effective (e.g., Are there significantly more students in the *Improved To Mastery* trajectory type if the unitizing stage precedes the partitioning stage than if the partitioning stage precedes the unitizing stage?).

The specific hypotheses generated from the classification results involved the relationship between in-game performance (as identified by the classification) and performance on paper-and-pencil measures to determine whether in-game performance was indicative of student performance on paper-and-pencil measures. In-game performance trajectories on specific concepts (such as partitioning or unitizing) could not be examined because the performance trajectories contain information only on the number of errors made, not the type of error. Therefore, there was expected to be a relationship between in-game performance trajectory and overall pretest and posttest performance, rather than performance on specific concepts, and certain trajectories (such as the *Improved To Mastery* trajectory) were expected to produce larger gains in pretest to posttest scores than other trajectories (such as the *Showed Decline* trajectory).

A significant relationship between in-game performance trajectory and pretest and posttest performance was found. Students with the lowest levels of prior knowledge of fractions do not achieve mastery at any point in the game. Students with moderately high prior knowledge of fractions made in-game errors that they corrected by the end of the game. Students with high prior knowledge of fractions showed mastery in early stages and either continued to show mastery or slipped from mastery in later stages of the game.

However, there was no significant relationship between in-game performance and gains in scores from pretest to posttest. There was a significant gain of about 10% from pretest to posttest, but the amount of gain did not differ based on in-game performance trajectory types. While, as hypothesized, the *All Mastery* trajectory had the highest gain scores (followed by the *Improved To Mastery* trajectory) and the *Showed Decline* trajectory had the smallest gain scores, the average gain scores for the different trajectories were not significantly different.

The finding that performance trajectories as an in-game measure of performance were significantly related to both pretest and posttest performance has a number of positive implications for the use of educational video games as assessments. Though the in-game performance trajectories cannot be used to provide detailed measures of the extent to which players have mastered specific learning goals as called for by the National Science and Technology Council (2011), the findings of this study indicate some promise that educational technology might be able to be used to provide information about the level of student performance related to the complex skills delineated in state and national standards, as called for by the U.S. Department of Education (2010, 2012). That is, educational games or simulations might yield in-game performance trajectories that reflect certain degrees of student understanding.

Additionally, the findings that the performance trajectories only partially correspond to the strategy sequences indicate that the performance trajectories are not simply relabeling the same students as identified in the strategy sequences. Rather, the performance trajectories identified from the classification results appear to provide information about in-game performance that is different from the information provided about in-game partitioning behavior by the sequence mining.

Implications, Limitations, and Future Work

In summary, all three educational data mining techniques were used successfully to extract meaningful information about student understanding of fractions from the game log data. Additionally, the groupings of students from the three data mining techniques were found to be systematically related to student performance on the paper-and-pencil pretest and posttest. Therefore this study successfully addressed the single biggest challenge to embedding assessment in educational games and simulations, extracting meaningful information about student performance from game log data, as put forth by the National Research Council (2011) and Mislevy et al. (2004).

This study offers initial support for the use of educational games and simulations as direct assessments of complex tasks (Linn et al., 1991) that can capture problem-solving strategies and mistakes (Merceron & Yacef, 2004; Quellmalz & Pellegrino, 2009; Rahkila & Karjalainen, 1999) in order to identify the strengths and weaknesses of individual students (Mehrens, 1992). The identification of valid in-game measures of student performance could allow educational video games and simulations to be used to provide detailed measures of the extent to which players have mastered specific learning goals (National Science and Technology Council, 2011) in order to assess complex skills identified in state and national standards (U.S. Department of Education, 2010).

Contrary to expectations, neither strategy sequences nor performance trajectories systematically related to gain scores (all strategy sequences and performance trajectories yielded the same average gain from pretest to posttest), and thus could not be interpreted as corresponding to levels of in-game learning (either high or low). There are three possible reasons for this result: (1) these in-game measures of performance were not adequately sensitive to learning that occurred during the game, (2) no measureable learning occurred during the game, or (3) the paper-and-pencil pretest and posttest were not sensitive to the kind of learning that did occur during the game.

Option 1 (that the data mining results were not sensitive to learning that occurred during the game) is the simplest explanation. However, such an interpretation would indicate that there are issues with the interpretation of the resulting groups of students. For example, the *Improved To Mastery* and *Showed Improvement* performance trajectories clearly lend themselves to the interpretation that in-game learning occurred for students in those trajectories (as opposed to, say, the *Showed Decline* performance trajectory). Before assuming that the data mining results were not sensitive to the learning that occurred during the game, other likely options should be ruled out.

Option 2 (that no learning took place during the game) seems unlikely at first glance, as there were significant pretest/posttest gains for students who played the game. However, there is a distinct possibility that the learning that took place did not occur during the game itself. The administration of the pretest and posttest occurred several weeks apart, during which time the students played other educational games and received regular mathematical instruction. It is possible that the learning that occurred between pretest and posttest was not due to the game, but to the other instruction students received in the interim. If in-game learning was not the primary contributor to the learning that occurred between the pretest and the posttest, that would explain why the in-game measures of learning (such as the *Improved To Mastery* performance trajectory) were not related to the learning gains on the test. If true, this would indicate that the results of the sequence mining and/or classification might actually be valid measures of learning, but that what they measured was not captured on the posttest score. In order to test this possibility, another study would have to be run in which the posttest immediately followed the game to rule out other causes of learning.

Option 3 (that the paper-and-pencil tests were not sensitive to the kind of in-game learning that occurred) is almost certainly a factor. The game is broken into six well-defined stages addressing specific fractions content, almost none of which is directly represented on the pretest or posttest. That is, many if not most of the test items are far transfer for the content taught in *Save Patch*. Therefore, it is not entirely surprising that the in-game measures of learning were not related to learning gains on the paper-and-pencil tests. Had the pretest and posttest consisted of items matched to each stage of the game, it is possible that the groups that showed greater in-game learning would have also shown greater pretest/posttest gains. It order to test this possibility, another study would have to be run in which the pretest and posttest were more directly aligned with the game content.

In addition to questions about the in-game measures of learning, the generalizability of the findings of this study are limited by how representative the *Save Patch* game is of other educational video games. The data mining techniques employed in this study depend on the presence of certain game design mechanics and will not work as intended if those design considerations are not met. First, as used in this study, all three data mining techniques rely on an easily definable delineator between attempts, such as a character death or the press of a Reset button. While these delineating events are almost always present in puzzle games such as *Save Patch*, they are uncommon in other game styles. In more open-world games or games that allow for the recovery from an error without resetting the level, it could be a non-trivial task to identify with certainty where one attempt ended and another began.

Secondly, the sequence mining and classification techniques both rely on the presence of stages in the game which target different concepts, sub-concepts, or difficulty levels and which consist of a minimum of three (preferably at least four) levels of substantively similar content. In games wherein the difficulty progresses linearly throughout the game or the levels addressing similar content are spread out through the game rather than blocked into stages of similar levels, an overall summary of the change in performance over a given stage would be meaningless as the stage itself either would not exist or would not be substantively meaningful.

Finally, the cluster analysis technique employed in this study (and, therefore, the meaningful sequencing of the resulting clusters) requires the in-game actions to have meaning in the context of the subject of the game and depends on log files that record each of those actions. If in-game actions are divorced from the content (e.g., a game such as *MathBlasters* wherein students control a gun that they use to shoot the correct answer and there is nothing in the game that tracks their problem-solving process), clustering actions will result, at best, only in the identification of the correct answer or the incorrect answer and will not result in the identification of misconceptions because the problem-solving actions that would serve as evidence are not valid actions in the game. Even if the game's actions are meaningful, each individual action must be recorded in the log files. Game logs that consist solely of summary information about student in-game performance or simply record the overall state of the game every second of game play are not suitable for this type of analysis.

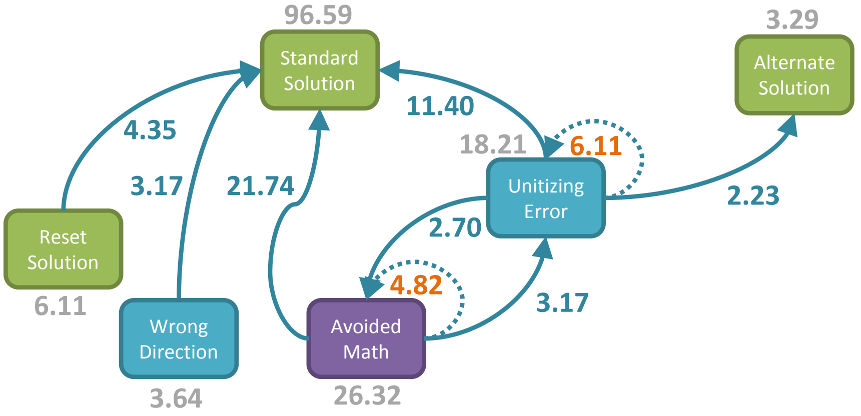
If games or simulations are to be used as stand-alone assessments, the next steps in this research should focus on the measurement property of games. For example, key features impacting the difficulty of individual levels and/or the amount of measurement error introduced by each feature must be identified. When examining in-game performance, it can be difficult to differentiate between an understanding of game mechanics and an understanding of the content being measured. Some features of game play might interfere with students' ability to demonstrate knowledge of the academic content (e.g., solving a level incorrectly due to problems with game mechanics rather than problems with content), while others might allow skilled gamers to demonstrate proficiency in the content area that they do not actually possess (e.g., solving a level correctly due to an understanding of the game mechanic rather than an understanding of the content). If the impact of various game mechanics on student performance can be identified and modeled, then it is possible for games and simulations to be used as assessments of student knowledge of complex academic content.

Appendix A: Stages and Levels in *Save Patch*

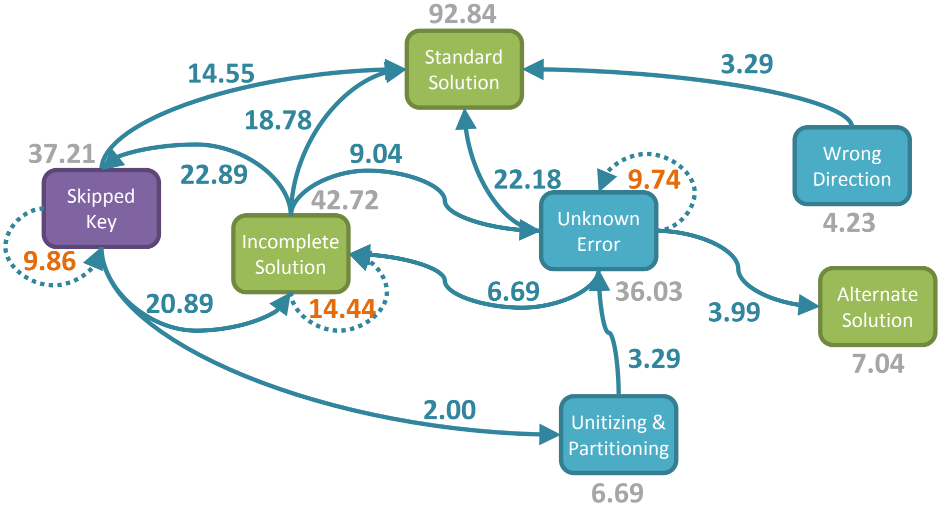
Stage	Content	Level	Denominator	Knowledge specifications
1	Wholes	1-3	1	1.3
2	Unit Fractions	4	2	3.0, 3.1
		5	4	3.0, 3.1, 3.2, 3.3
		6	5	1.0, 1.1, 1.3, 3.0, 3.1, 3.2, 3.3
		7	3	1.0, 1.1, 1.3, 3.0, 3.1
		8	2	3.0, 3.1
3	Whole Numbers and Unit Fractions	9	2	1.0,1.1, 1.3, 3.0, 3.1, 4.0, 4.1, 4.3
		10	3	1.0,1.1, 1.3, 3.0, 3.1, 4.0, 4.1, 4.3
		11	4	1.3, 3.0, 3.1, 4.0, 4.1, 4.3
		12	5	1.0,1.1, 1.3, 3.0, 3.1, 4.0, 4.1, 4.3
4	Wholes Across the Unit Mark	13	2	1.0,1.1, 1.3, 3.0, 3.1, 4.0, 4.1, 4.3
		14	3	1.0,1.1, 1.3, 3.0, 3.1, 4.0, 4.1, 4.3
		15	4	1.0,1.1, 1.3, 3.0, 3.1, 4.0, 4.1, 4.3
5	Proper Fractions	16	4	2.2, 3.0, 3.1, 4.0, 4.1, 4.2
		17	5	1.0, 1.1, 1.3, 2.2, 3.0, 3.1, 4.0, 4.1, 4.2
		18	3	1.0, 1.1, 1.3, 2.2, 3.0, 3.1, 4.0, 4.1, 4.2
		19	4	2.2, 3.0, 3.1, 4.0, 4.1, 4.2
6	Improper Fractions	20	2	1.0, 1.1, 1.3, 2.0, 2.1, 2.2, 2.3, 3.0, 3.1, 4.0, 4.1, 4.4
		21	3	1.0, 1.1, 1.3, 2.0, 2.1, 2.2, 2.3, 3.0, 3.1, 4.0, 4.1, 4.4
		22	4	1.0, 1.1, 1.3, 2.0, 2.1, 2.2, 2.3, 3.0, 3.1, 4.0, 4.1, 4.4
		23	3	1.0, 1.1, 1.3, 2.0, 2.1, 2.2, 2.3, 3.0, 3.1, 4.0, 4.1, 4.4
7	Test Levels	24	4	Same as Level 11 – Whole Numbers and Unit Fractions
		25	3	Same as Level 14 – Wholes Across the Unit Mark
		26	5	Same as Level 17 – Proper Fractions
		27	3	Same as Level 21 – Improper Fractions

Appendix B: Strategy Sequence Graphs

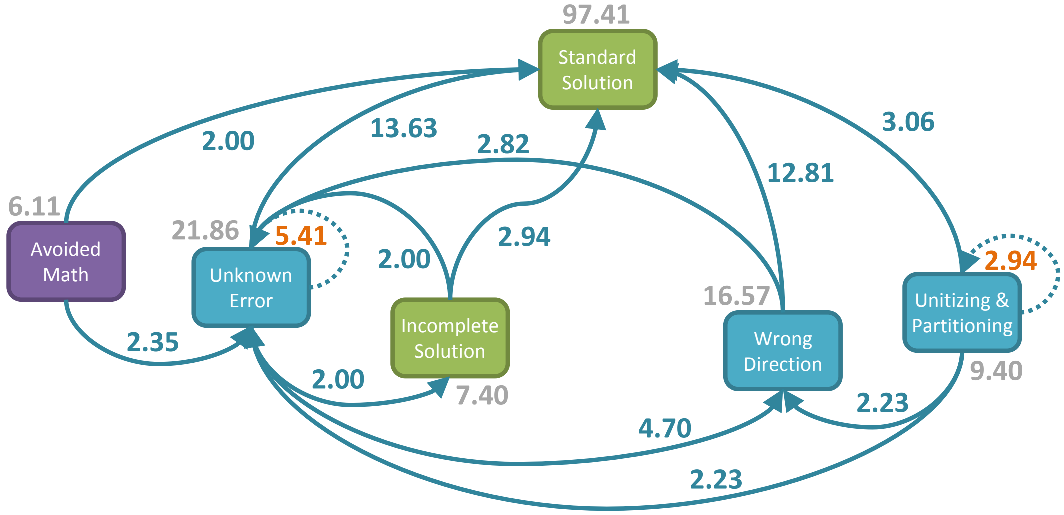
Strategy Sequence Graph for Level 01



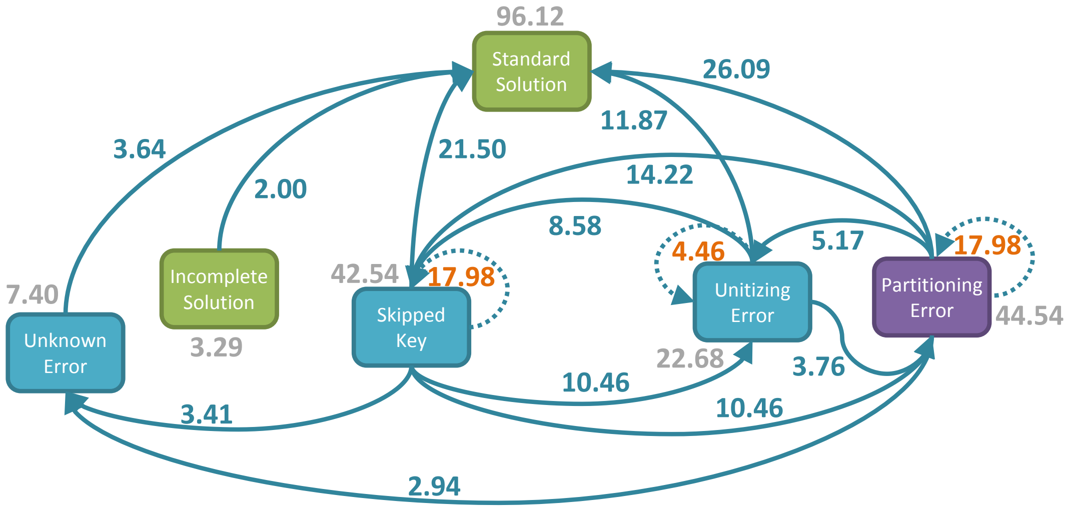
Strategy Sequence Graph for Level 02



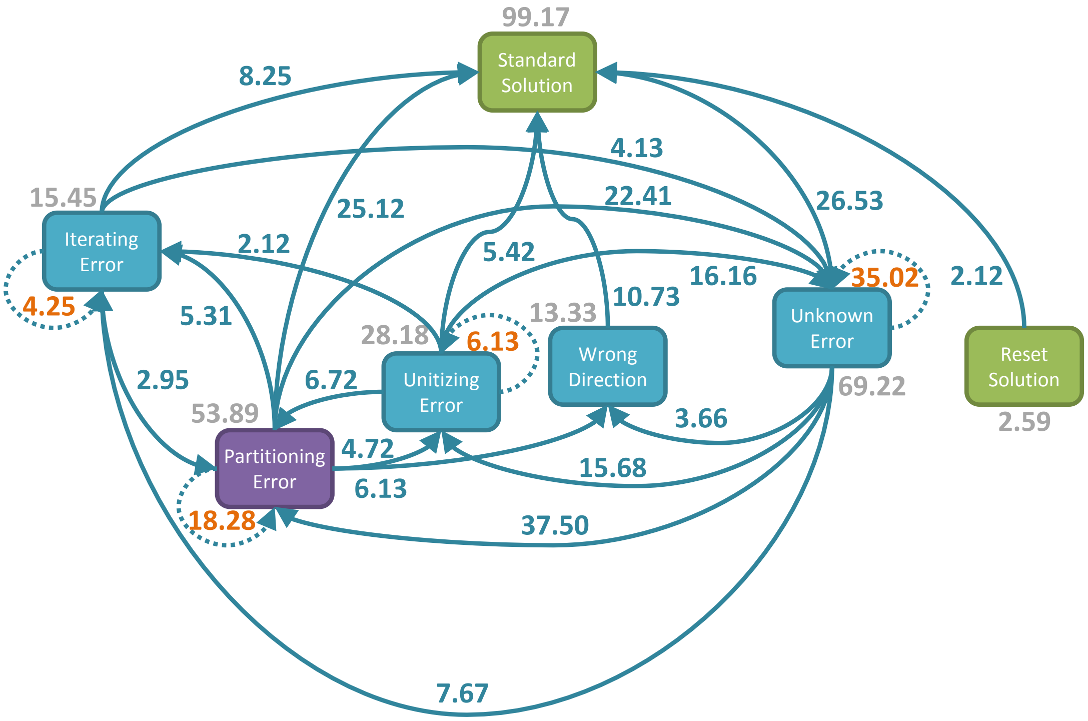
Strategy Sequence Graph for Level 03



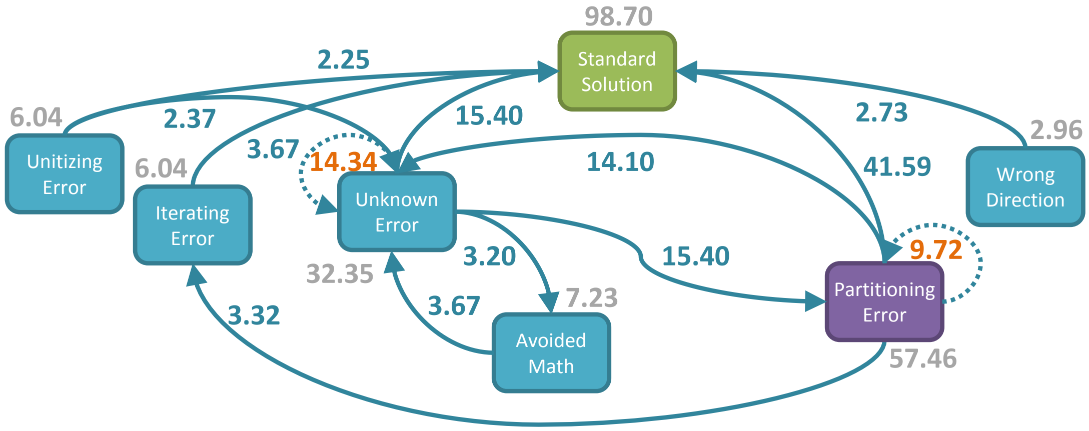
Strategy Sequence Graph for Level 04



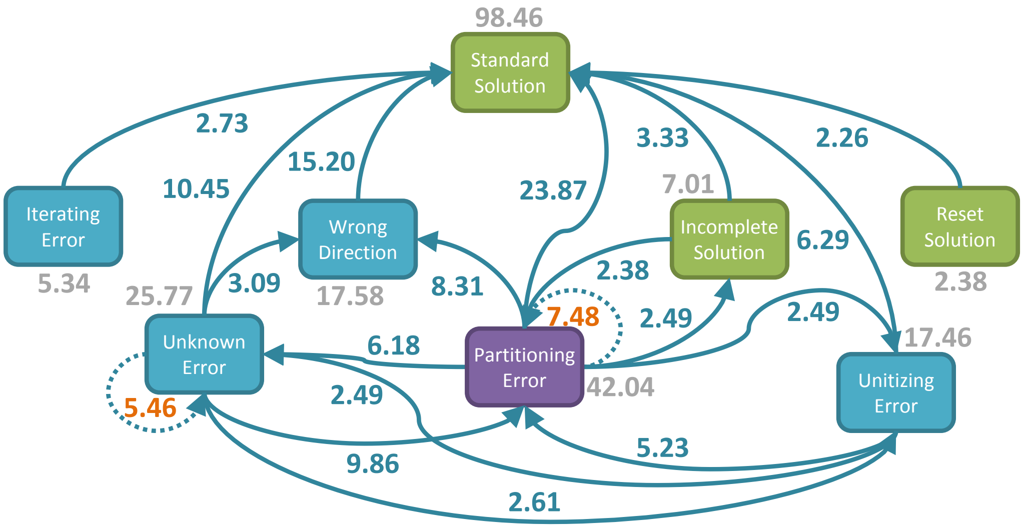
Strategy Sequence Graph for Level 05



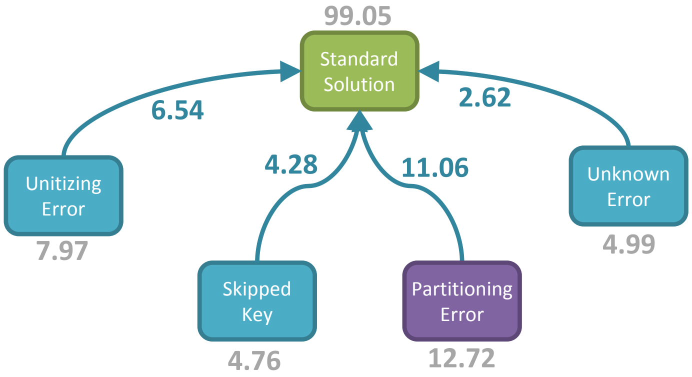
Strategy Sequence Graph for Level 06



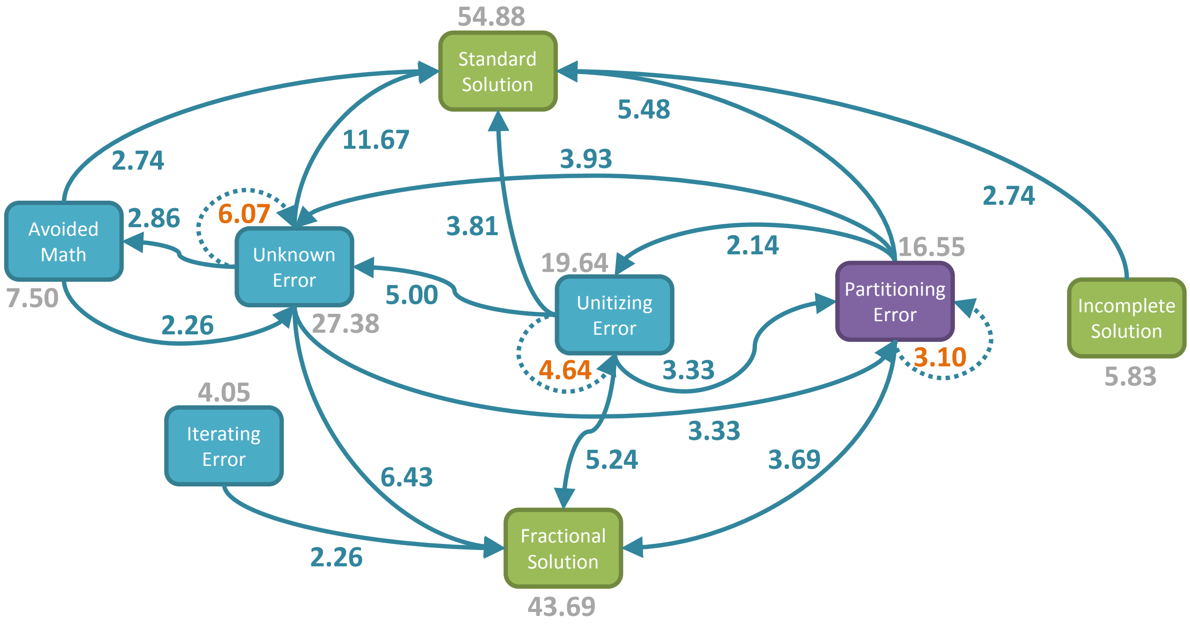
Strategy Sequence Graph for Level 07



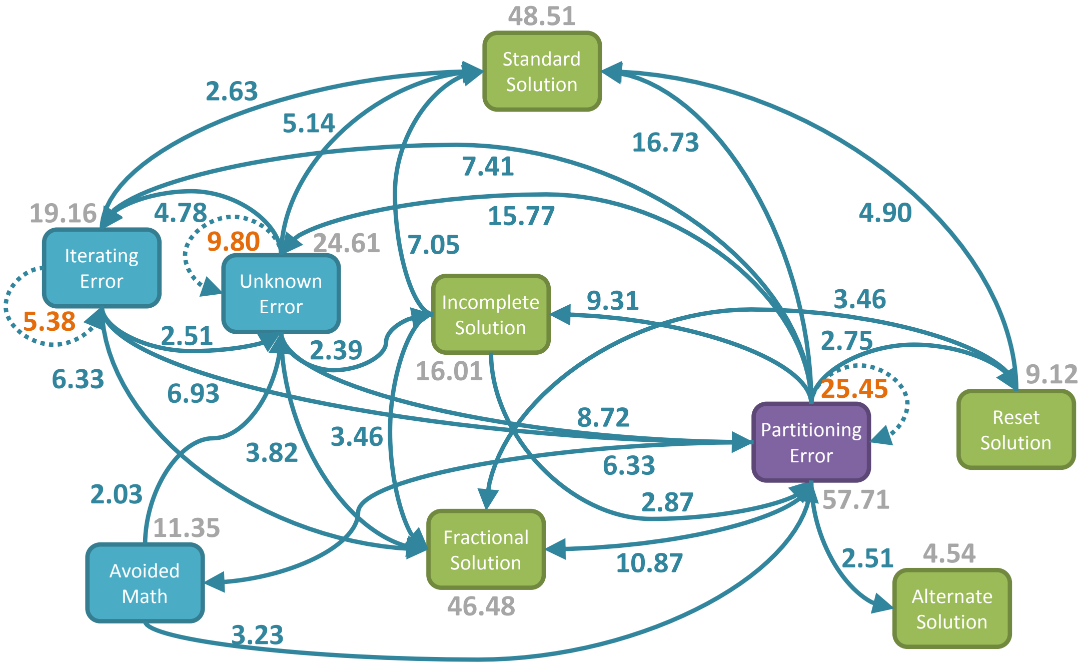
Strategy Sequence Graph for Level 08



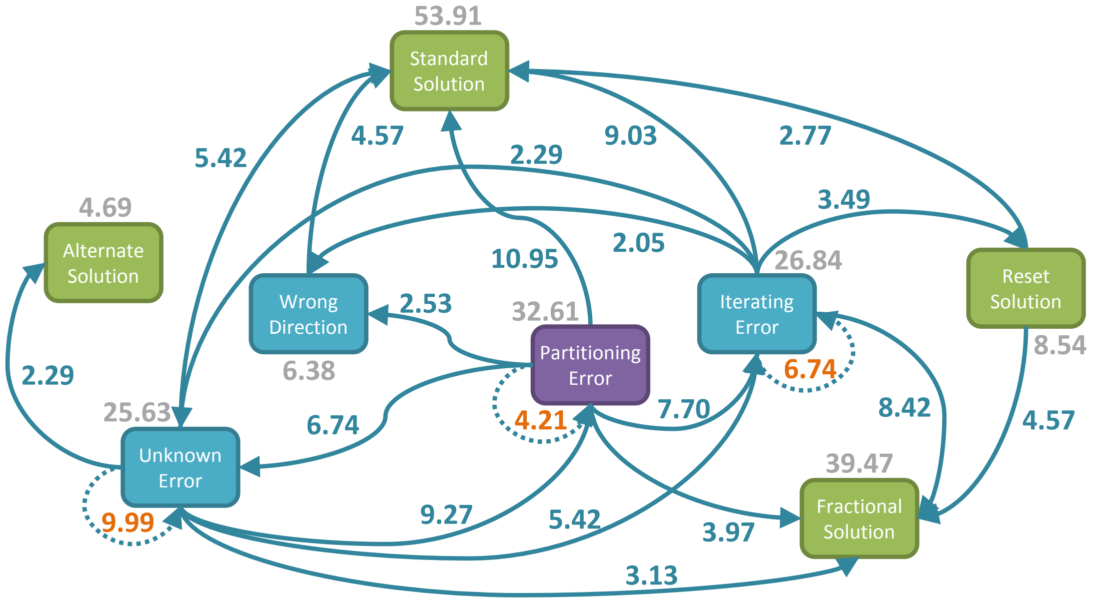
Strategy Sequence Graph for Level 09



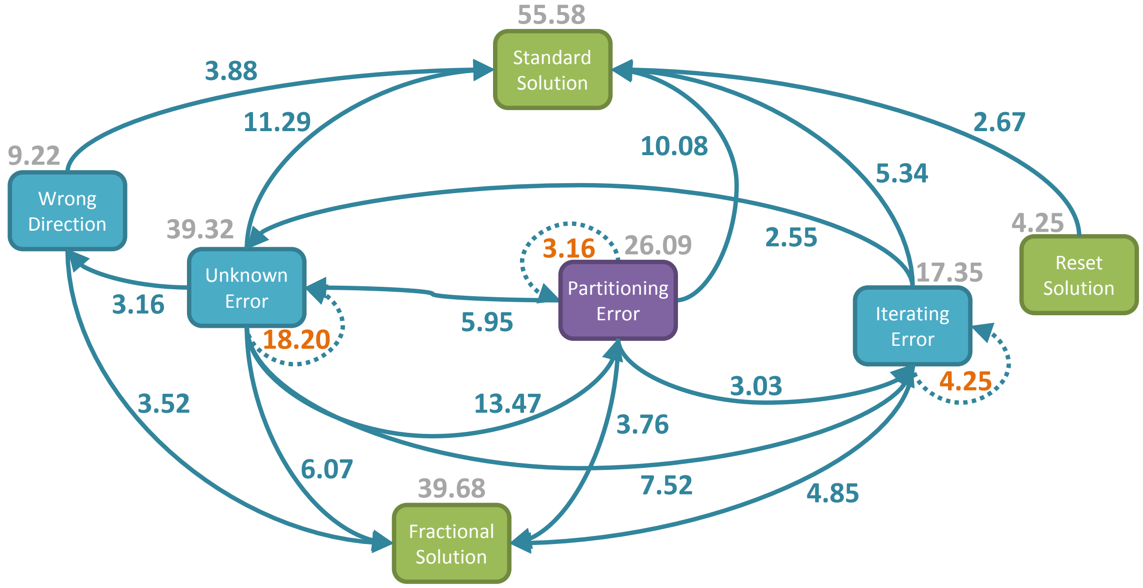
Strategy Sequence Graph for Level 10



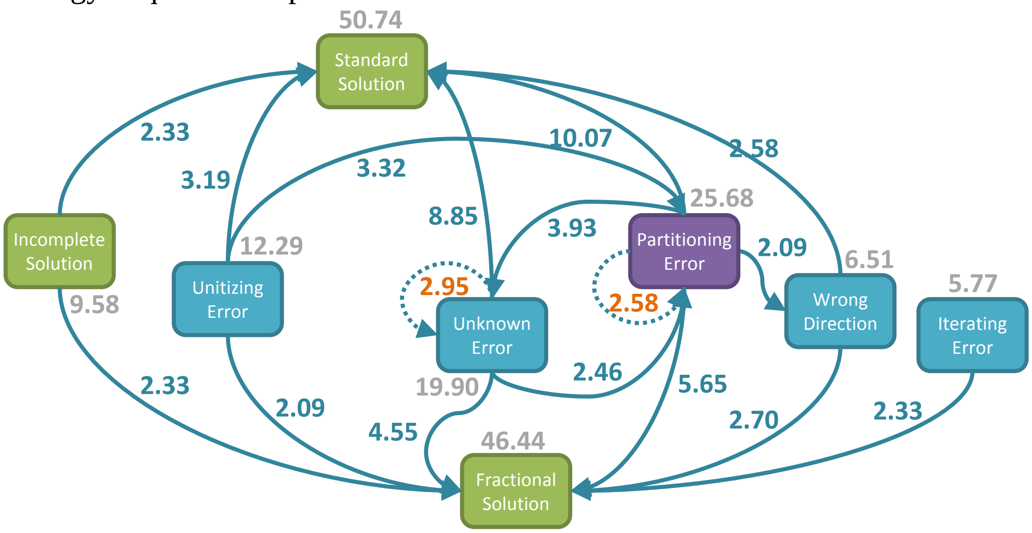
Strategy Sequence Graph for Level 11



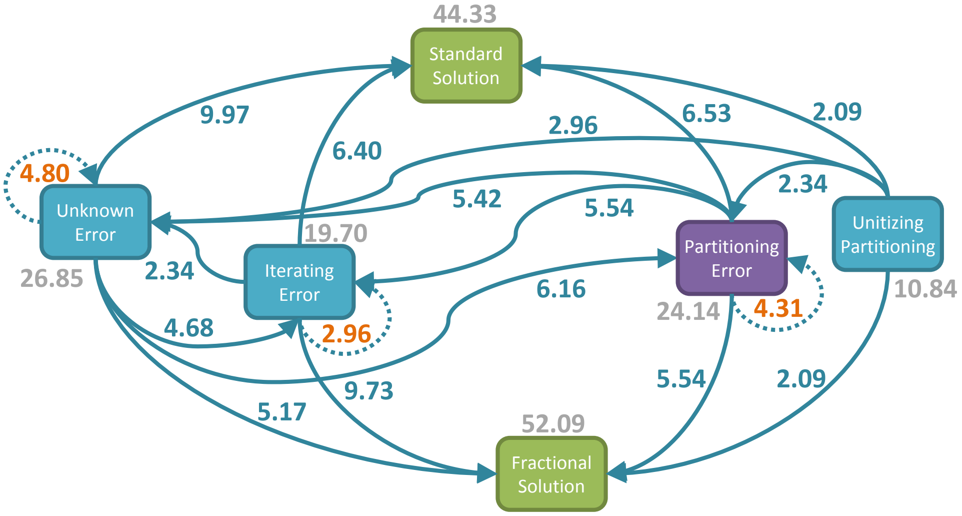
Strategy Sequence Graph for Level 12



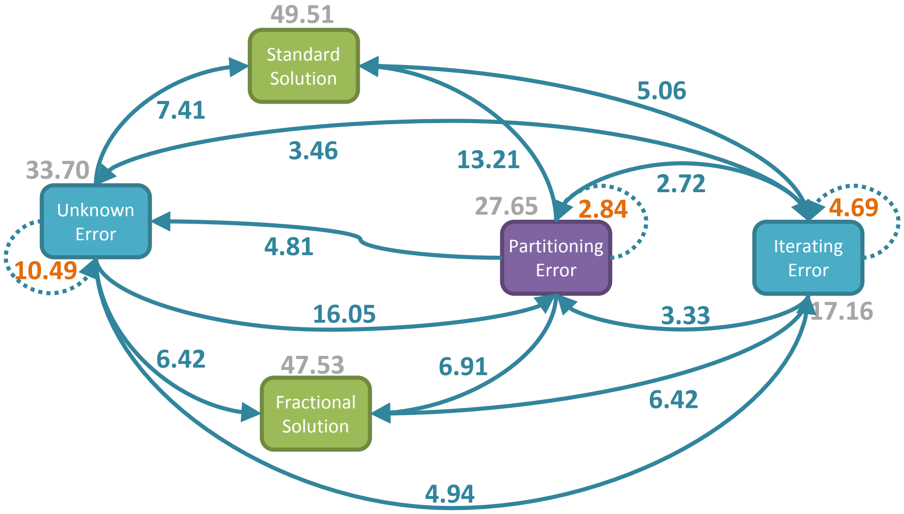
Strategy Sequence Graph for Level 13



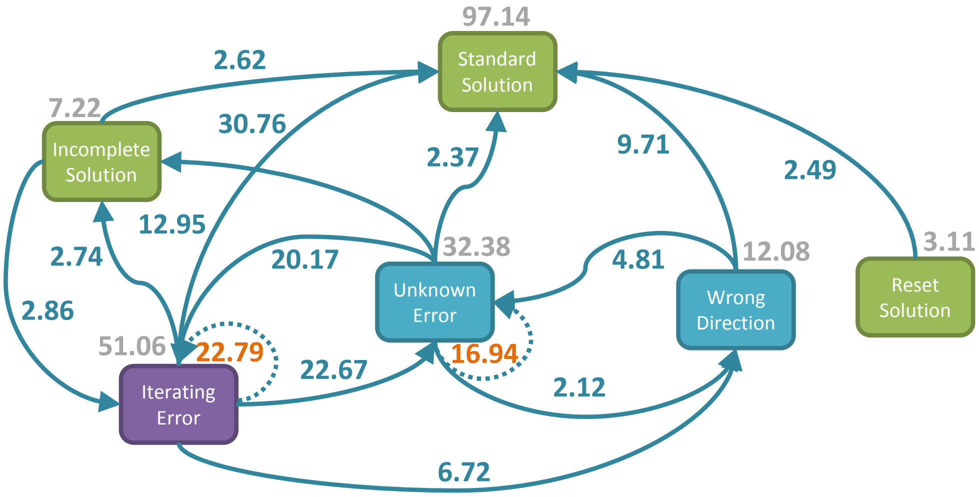
Strategy Sequence Graph for Level 14



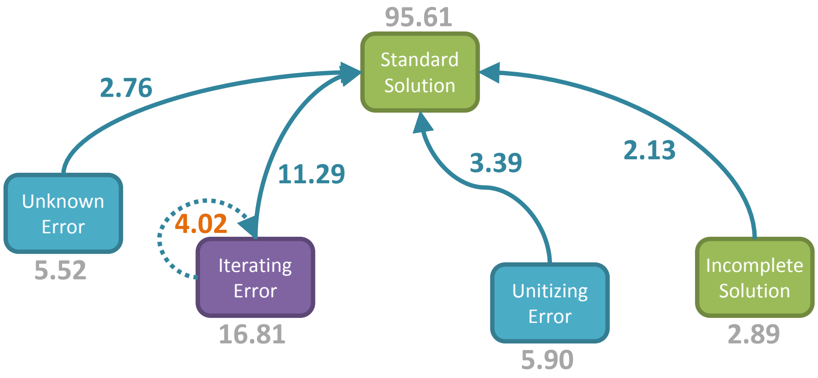
Strategy Sequence Graph for Level 15



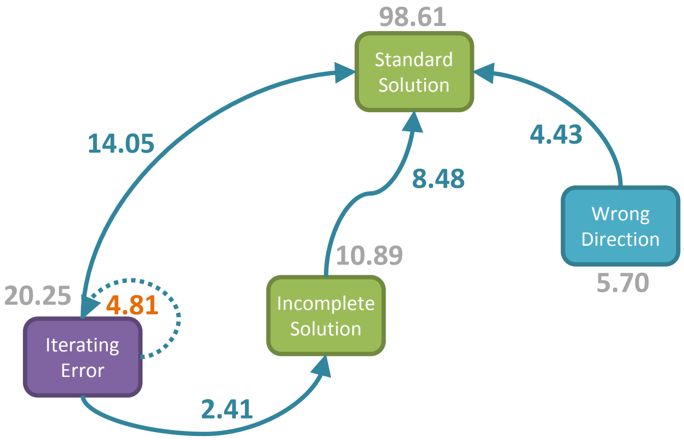
Strategy Sequence Graph for Level 16



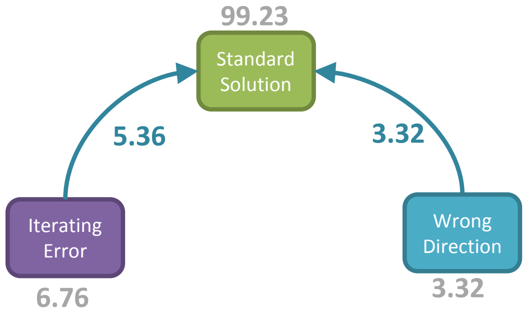
Strategy Sequence Graph for Level 17



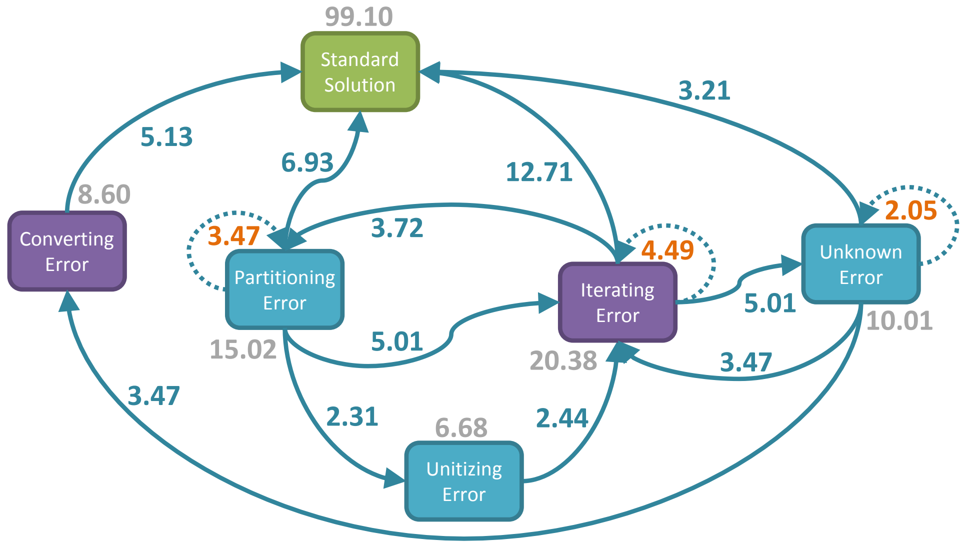
Strategy Sequence Graph for Level 18



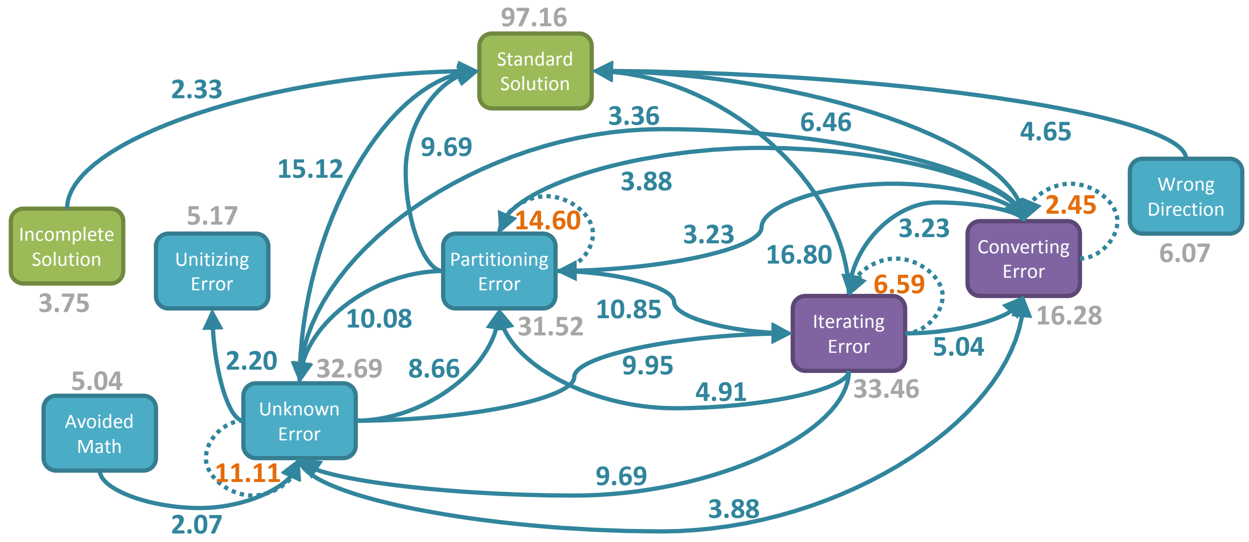
Strategy Sequence Graph for Level 19



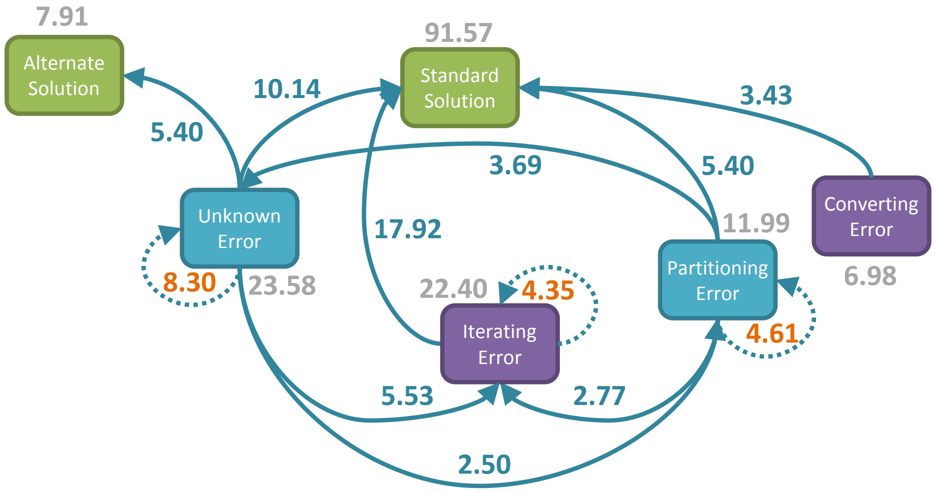
Strategy Sequence Graph for Level 20



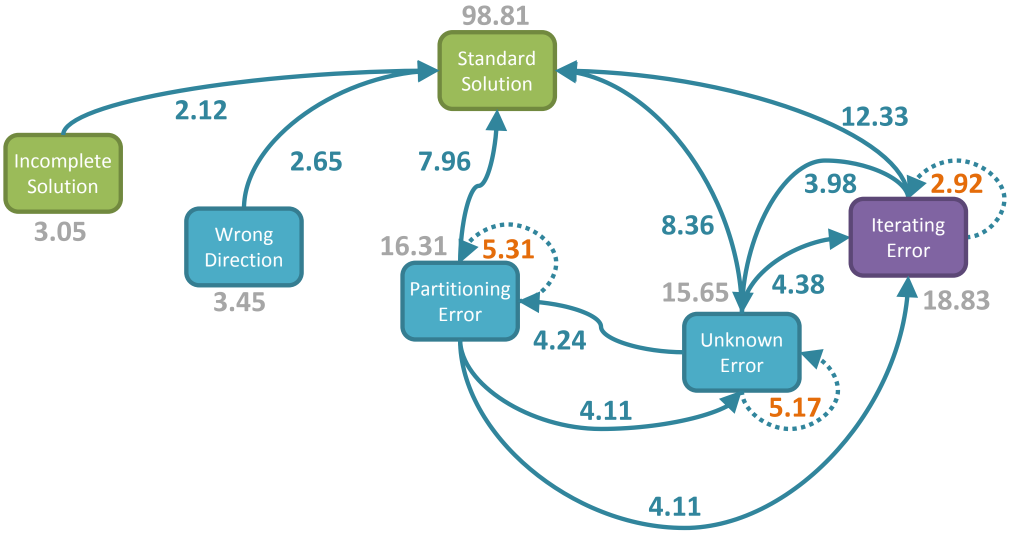
Strategy Sequence Graph for Level 21



Strategy Sequence Graph for Level 22



Strategy Sequence Graph for Level 23



Appendix C: Sequence Group Type Coding for Stage 4

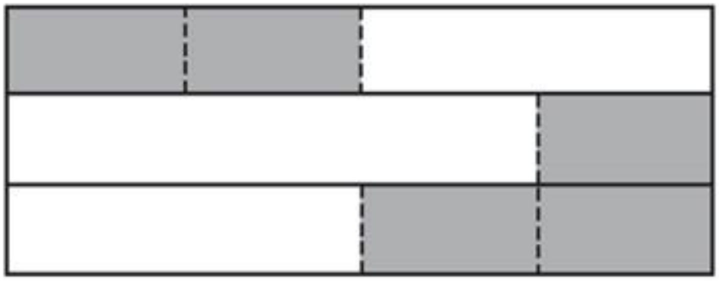
Strategy sequence type	Level 13 sequence type	Level 14 sequence type	Level 15 sequence type
Correct	Correct	Correct	Correct
Correct to Partitioning	Correct	Correct	Other Error to Partitioning
	Correct	Correct	Partitioning to Correct
Correct to Other Error	Correct	Correct	Other Error to Correct
	Correct	Correct	Other Error to Other Error
Partitioning to Correct	Correct	Partitioning to Correct	Correct
	Other Error to Correct	Partitioning to Correct	Correct
	Other Error to Other Error	Partitioning to Correct	Correct
	Partitioning to Correct	Correct	Correct
	Partitioning to Correct	Other Error to Correct	Correct
	Partitioning to Correct	Other Error to Other Error	Correct
	Partitioning to Correct	Partitioning to Correct	Correct
Partitioning to Other Error	Correct	Other Error to Partitioning	Other Error to Correct
	Correct	Partitioning to Correct	Other Error to Correct
	Correct	Partitioning to Correct	Other Error to Other Error
	Other Error to Correct	Partitioning to Correct	Other Error to Correct
	Other Error to Correct	Partitioning to Correct	Other Error to Other Error
	Other Error to Other Error	Partitioning to Correct	Other Error to Correct
	Other Error to Other Error	Partitioning to Correct	Other Error to Other Error
	Other Error to Other Error	Partitioning to Partitioning	Other Error to Other Error
	Partitioning to Correct	Correct	Other Error to Correct
	Partitioning to Correct	Correct	Other Error to Other Error
	Partitioning to Correct	Other Error to Correct	Other Error to Correct
	Partitioning to Correct	Other Error to Correct	Other Error to Other Error
	Partitioning to Correct	Other Error to Partitioning	Other Error to Correct
	Partitioning to Correct	Other Error to Other Error	Other Error to Correct
	Partitioning to Correct	Other Error to Other Error	Other Error to Other Error
	Partitioning to Correct	Partitioning to Correct	Other Error to Correct
	Partitioning to Correct	Partitioning to Correct	Other Error to Other Error
Partitioning to Partitioning	Correct	Partitioning to Correct	Partitioning to Correct
	Correct	Partitioning to Other Error	Partitioning to Correct
	Other Error to Correct	Partitioning to Correct	Other Error to Partitioning
	Other Error to Correct	Partitioning to Correct	Partitioning to Correct
	Other Error to Other Error	Partitioning to Correct	Other Error to Partitioning
	Other Error to Other Error	Partitioning to Correct	Partitioning to Correct
	Other Error to Other Error	Partitioning to Other Error	Other Error to Partitioning
	Other Error to Other Error	Partitioning to Other Error	Partitioning to Correct
	Other Error to Other Error	Partitioning to Partitioning	Partitioning to Correct
	Partitioning to Correct	Correct	Partitioning to Correct
	Partitioning to Correct	Other Error to Correct	Other Error to Partitioning
	Partitioning to Correct	Other Error to Correct	Partitioning to Correct
	Partitioning to Correct	Other Error to Other Error	Partitioning to Correct
	Partitioning to Correct	Other Error to Partitioning	Partitioning to Correct
	Partitioning to Correct	Partitioning to Correct	Other Error to Partitioning
	Partitioning to Correct	Partitioning to Correct	Partitioning to Correct
	Partitioning to Correct	Partitioning to Other Error	Partitioning to Correct

Strategy sequence type	Level 13 sequence type	Level 14 sequence type	Level 15 sequence type
Other Error to Correct	Correct	Other Error to Correct	Correct
	Correct	Other Error to Other Error	Correct
	Other Error to Correct	Correct	Correct
	Other Error to Correct	Other Error to Correct	Correct
	Other Error to Correct	Other Error to Other Error	Correct
	Other Error to Other Error	Correct	Correct
	Other Error to Other Error	Other Error to Correct	Correct
	Other Error to Other Error	Other Error to Other Error	Correct
Other Error to Partitioning	Correct	Other Error to Correct	Partitioning to Correct
	Correct	Other Error to Other Error	Partitioning to Correct
	Other Error to Correct	Correct	Partitioning to Correct
	Other Error to Correct	Other Error to Correct	Partitioning to Correct
	Other Error to Correct	Other Error to Other Error	Partitioning to Correct
	Other Error to Other Error	Correct	Partitioning to Correct
	Other Error to Other Error	Other Error to Correct	Other Error to Partitioning
	Other Error to Other Error	Other Error to Correct	Partitioning to Correct
	Other Error to Other Error	Other Error to Other Error	Other Error to Partitioning
	Other Error to Other Error	Other Error to Other Error	Partitioning to Correct
Other Error to Other Error	Correct	Other Error to Correct	Other Error to Correct
	Correct	Other Error to Correct	Other Error to Other Error
	Correct	Other Error to Other Error	Other Error to Correct
	Correct	Other Error to Other Error	Other Error to Other Error
	Other Error to Correct	Correct	Other Error to Correct
	Other Error to Correct	Other Error to Correct	Other Error to Correct
	Other Error to Correct	Other Error to Correct	Other Error to Other Error
	Other Error to Correct	Other Error to Other Error	Other Error to Correct
	Other Error to Other Error	Correct	Other Error to Correct
	Other Error to Other Error	Correct	Other Error to Other Error
	Other Error to Other Error	Other Error to Correct	Other Error to Correct
	Other Error to Other Error	Other Error to Correct	Other Error to Other Error
	Other Error to Other Error	Other Error to Other Error	Other Error to Correct
	Other Error to Other Error	Other Error to Other Error	Other Error to Other Error

Appendix D: Paper-and-Pencil Pretest and Posttest Items

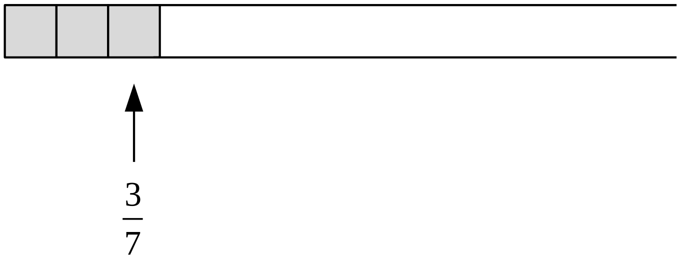
The following are the paper-and-pencil items related to content in *Save Patch*. Specific skills the items are intended to measure are listed in the box to the right. All items were on both the pretest and posttest. Both tests also consisted of additional items not related to *Save Patch*.

Write a fraction that describes the shaded part of the figure below.



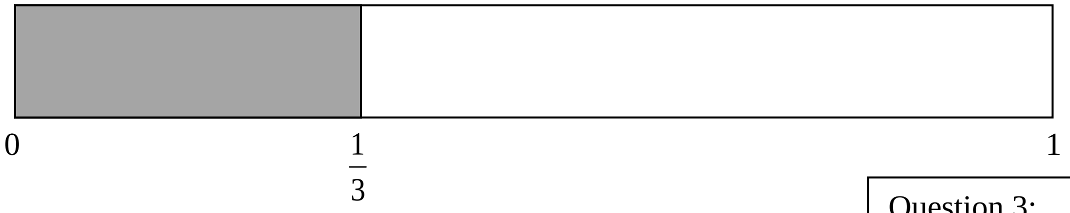
Question 1:
Unitizing &
Partitioning

The figure below shows 3/7 of a whole unit shaded. Complete the figure to show where the whole unit ends. Be sure to draw lines ("|") to show where each piece is.



Question 2:
Unitizing

The shaded part of the block below shows 1/3 of a whole unit. Mark where 1/6 of a whole unit is.



Question 3:
Partitioning

What does the bottom number (4) tell you in $\frac{3}{4}$?

a. It tells you there are four fourths in this fraction

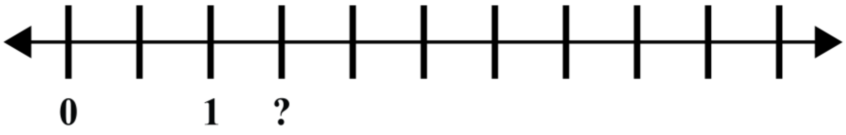
b. It tells you the whole unit is broken into four pieces

c. It tells you there are four whole units in this fraction

d. It tells you to add 3 four times

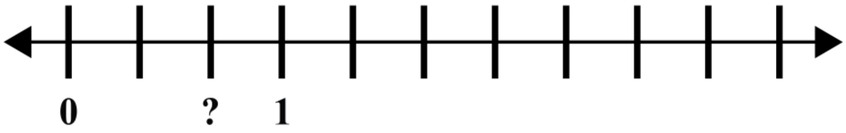
Question 4:
Partitioning

At what number is the "?" located?



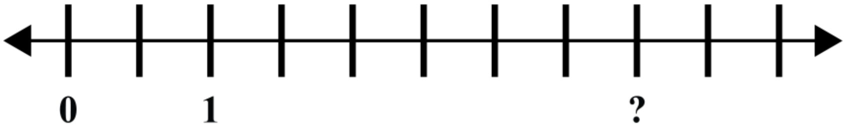
Question 5:
Unitizing &
Partitioning

At what number is the "?" located?



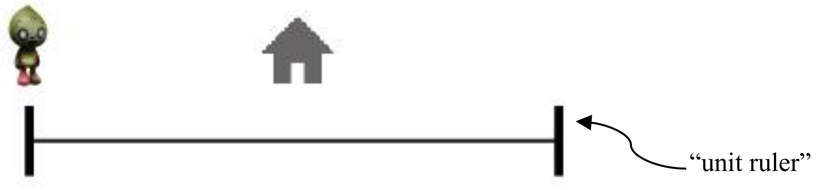
Question 6:
Unitizing &
Partitioning

At what number is the "?" located?



Question 7:
Unitizing &
Partitioning

In the figure below, use the "unit ruler" to measure the distance between the puppet and its home.



Question 8:
Unitizing &
Partitioning

For the questions below, fill in each box with a number that will make the statement true. **The fractions DO NOT need to be simplified**!

$$\frac{2}{7} + \frac{1}{7} = \frac{\square}{\square}$$

Question 9:
Adding

$$\frac{1}{6} + \frac{1}{4} = \frac{\square}{\square}$$

Question 10:
Adding

$$\frac{2}{5} + \frac{3}{10} = \frac{\square}{\square}$$

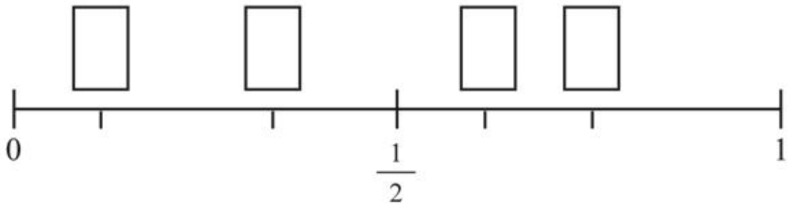
Question 11:
Adding

$$\frac{3}{11} + \frac{\square}{\square} = \frac{7}{22}$$

Question 12:
Adding

Here are four fractions: $\frac{3}{4}$, $\frac{1}{8}$, $\frac{1}{3}$ and $\frac{3}{5}$.

Look at the number line below. Write each fraction in the correct box.



Question 13:
Unitizing &
Partitioning

Appendix E: Percentage of Identified Information Across Stages

			Percentage identified	
Stage	Level	Students	Strategies	Sequences
1: Wholes	1	851	98.1%	88.5%
	2	851	80.9%	94.6%
	3	851	82.4%	87.2%
2: Unit Fractions	4	851	97.1%	95.6%
	5	848	62.6%	98.6%
	6	844	72.8%	90.9%
	7	842	84.0%	92.0%
	8	841	95.3%	89.8%
3: Whole Numbers and Unit Fractions	9	840	81.6%	79.2%
	10	837	86.3%	89.3%
	11	831	77.5%	83.1%
	12	824	66.6%	85.2%
4: Wholes Across the Unit Mark	13	814	87.5%	69.4%
	14	812	83.0%	89.1%
	15	810	76.2%	91.7%
5: Proper Fractions	16	803	69.7%	96.1%
	17	797	95.2%	83.0%
	18	790	98.2%	90.5%
	19	784	96.8%	92.6%
6: Improper Fractions	20	779	92.4%	93.6%
	21	774	79.3%	93.2%
	22	759	80.5%	90.9%
	23	754	85.0%	91.7%

References

Amershi, S., & Conati, C. (2011). Automatic recognition of learner types in exploratory learning environments. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S.J.d. Baker (Eds.), *Handbook of educational data mining* (pp. 389-416). Boca Raton, FL: CRC Press.

Amershi, S., Conati, C., & Maclaren, H. (2006). Using feature selection and unsupervised clustering to identify affective expressions in educational games. In G. Rebolledo-Mendez & E. Martinez-Miron (Eds.), *Proceedings of the Workshop on Motivational and Affective Issues at the 8th International Conference on Intelligent Tutoring Systems* (pp. 21-28). Berlin, Heidelberg: Springer-Verlag.

Anaya, A. R., & Boticario, J. G. (2009). A data mining approach to reveal representative collaboration indicators in open collaboration frameworks. In T. Barnes, M. Desmarais, C. Romero, & S. Ventura (Eds.),*Proceedings of the 2nd International Conference on Educational Data Mining* (pp. 210-219).

Antunes, C. (2008). Acquiring background knowledge for intelligent tutoring systems. In R. S. J. d. Baker, T. Barnes, & J. E. Beck (Eds.) *Proceedings of the 1st International Conference on Educational Data Mining* (pp.18-27).

Araya, R., Jimenez, A., Bahamondez, M., Dartnell, P., Soto-Andrade, J., Gonzalez, P., & Calfulcura, P. (2011). Strategies used by students on a massively multiplayer online mathematics game. *Lecture Notes on Computer Science*, *7048*, 1-10.

Avouris, N., Komis, V., Fiotakis, G., Margaritis, M., & Voyiatzaki, E. (2005). Logging of fingertip actions is not enough for analysis of learning activities. In *Proceedings of the Workshop on Usage Analysis in Learning Systems at the 12th International Conference on Artificial Intelligence in Education*.

Ayala, A. P., Dominguez, R., & Medel, J. d. J. (2009). Educational data mining: A sample of review and study case. *World Journal on Educational Technology*, *2*, 118-139.

Baker, R. S. J. d., Corbett, A. T., & Koedinger, K. R. (2004). Detecting student misuse of intelligent tutoring systems. *Lecture Notes in Computer Science*, *3220*, 531-540.

Baylor, A. L. (2002). Expanding preservice teachers' metacognitive awareness of instructional planning. *Educational Technology Research and Development*, *50*, 5-22.

Bejar, I. I. (1984). Educational diagnostic assessment. *Journal of Educational Measurement*, *21*(2), 175-189.

Berkhin, R. (2006). A survey of clustering data mining techniques. In J. Kogan, C. Nicholas, & M. Teboulle (Eds.), *Grouping multidimensional data* (pp. 25-72). New York, NY: Springer.

Betz, J. A. (1995). Computer games: Increase learning in an interactive multidisciplinary environment. *Journal of Technological Systems, 24,* 195-205.

Blunt, R. (2008, August). *Does game-based learning work? Results from three recent studies.* Paper presented at the Joint ADL/Co Lab Implementation Fest, Orlando, FL.

Bonchi, F., Giannoti, F., Gozzi, C., Manco, G., Nanni, M., Pedreschi, D., ... Ruggieri, S. (2001). Web log data warehouses and mining for intelligent web caching. *Data & Knowledge Engineering, 39,* 165-189.

Brown, J., Hinze, S., & Pellegrino, J. W. (2008). Technology and formative assessment. In T. Good (Ed.), *21st century education*. Thousand Oaks, CA: Sage.

Buchta, C., & Hahsler, M. (2013). *arulesSequences: Mining frequent sequences*. R package version 0.2-4 [Computer Software]. Retrieved from http://cran.r-project.org/web/packages/arulesSequences/index.html

Buckley, B. C., Gobert, J. D., & Horwitz, P. (2006). Using log files to track students' model-based inquiry. In *Proceedings of the 7th International Conference on Learning Sciences* (pp. 57-63). International Society of the Learning Sciences.

Carpenter, T. P., Fennema, E., Franke, M. L., Levi, L. W., & Empson, S. B. (2000). *Cognitively guided instruction: A research-based teacher professional development program for elementary school mathematics*. Madison, WI: National Center for Improving Student Learning and Achievement in Mathematics and Science.

Carroll, J. B. (1961). The nature of data, or how to choose a correlation coefficient. *Psychometrika*, *16*(4), 347-372.

Castro, F., Vellido, A., Nebot, A., & Mugica, F. (2007). Applying data mining techniques toe-learning problems. In L. C. Jain, R. A. Tedman, & D. K. Tedman (Eds.), *Evolution of Teaching and Learning Paradigms in Intelligent Environment* (Vol. 62, pp. 183-221). Berlin, Germany: Springer.

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences, 4*(1), 300-307.

Chen, C. C., & Chen, A. P. (2007). Using data mining technology to provide a recommendation service in the digital library. *The Electronic Library*, *25*(6), 711-724.

Chen, H.-H. & O'Neil, H. F. (2008). A formative evaluation of the training effectiveness of a computer game. In H. F. O'Neil & R. S. Perez (Eds.), *Computer games and team and individual learning* (pp. 39-54). Amsterdam, Netherlands: Elsevier.

Chika, I. E., Azzi, D., Hewitt, A., & Stocker, J. (2009). A holistic approach to assessing students' laboratory performance using Bayesian networks. In *Proceedings of the IEEE Workshop on Computational Intelligence in Virtual Environments (CIVE)* (pp.26-32).

Chiu, D. Y., Pan, Y. C., & Chang, W. C. (2008). Using rough set theory to construct e-learning FAQ retrieval infrastructure. In *Proceedings of the First IEEE International Conference on Ubi-Media Computing* (pp. 547-552).

Christodoulopoulos, C. E., & Papanikolaou, K. A. (2007). A group formation tool in an e-learning context. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence* (pp. 117-123).

Chung, G. K.W.K., Baker, E. L., Vendlinski, T. P., Buschang, R. E., Delacruz, G. C., Michiuye, J. K., & Bittick, S. J. (2010, April). Testing instructional design variations in a prototype math game. In R. Atkinson (Chair), *Current perspectives from three national R&D centers focused on game-based learning: Issues in learning, instruction, assessment, and game design*. Structured poster session at the annual meeting of the American Educational Research Association, Denver, CO.

Chung, G. K.W.K., & Kerr, D. (2012). *A primer on data logging to support extraction of meaningful information from educational games: An example from Save Patch* (CRESST Report 814). Los Angeles, CA, University of California, National Center for Research on Evaluation, Standards, and Student Testing.

Clark, D. B., Nelson, B., Sengupta, P., & D'Angelo, C. (2009). *Rethinking science learning through digital games and simulations: Genres, examples, and evidence*. Paper commissioned for the National Research Council Workshop on Gaming and Simulations.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *21*(1), 37-46.

Coller, B. D., & Shernoff, D. J. (2009). Video game-based education in mechanical engineering: A look at student engagement. *International Journal of Engineering Education, 25*(2), 308-317.

Conati, C., & Merten, C. (2007). Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation. *Knowledge Base Systems*, *20*(6), 557-574.

Cook, D. A., & Beckman, T. J. (2006). Current concepts in validity and reliability for psychometric instruments: Theory and application. *The American Journal of Medicine, 119*(2), 166.e7-166.e16.

Cordova, D. I.,& Lepper, M. R. (1996). Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of Educational Psychology*, *88*(4), 715-730.

Cummins, D., Yacef, K., & Koprinska, I. (2006). A sequence based recommender system for learning resources. *Australian Journal of Intelligent Information Processing Systems*, *9*(2), 49-56.

Delacruz, G. C., Chung, G. K.W.K., & Baker, E. L. (2010).*Validity evidence for games as assessment environments* (CRESST Report 773).Los Angeles, CA: University of California, National Center for Research on Evaluation, Standards, and Student Testing.

Din, F.,& Caleo, J. (2001). The effects of playing educational video games on kindergarten achievement. *Child Study Journal, 31*, 95-102.

Edelson, D. C., Gordin, D. N., & Pea, R. D. (1999). Addressing the challenges of inquiry-based learning through technology and curriculum design. *Journal of the Learning Sciences, 8*(3/4), 391-450.

El Den, A. S., Moustafa, M. A., Harb, H. M., & Emara, A. H. (2013). AdaBoost ensemble with simple genetic algorithm for student prediction model. *International Journal for Computer Science & Information Technology, 5*(2), 73-85.

Ferran, N., Casadesus, J., Krakowska, M., & Minguillon, J. (2007). Enriching e-learning metadata through digital library usage analysis. *The Electronic Library*, *25*(2), 148-165.

Ferry, Y. A., & Ponserre, S. (2001). Enhancing the control of force in putting by video game training. *Ergonomics*, *44*, 1025-1037.

Fisch, S. M. (2005). Making educational computer games “educational.” In *Proceedings of the 4th International Conference for Interaction Design and Children* (pp. 56-61).

Frawley, W. J., Piateski-Shapiro, G., & Matheus, C. J. (1992). Knowledge discovery in databases: An overview. *AI Magazine*, *13*(3), 57-70.

Frezzo, D. C., Behrens, J. T., Mislevy, R. J., West, P., & DiCerbo, K. E. (2009). Psychometric and evidentiary approaches to simulation assessment in packet tracer software. In *Proceedings of the Fifth International Conference on Networking and Services (ICNS 2009)* (pp. 555-560).

Gallagher, A. G., Lederman, A. B., McGlade, K., Satava, R. M., & Smith, C. D. (2004). Discriminative validity of the Minimally Invasive Surgical Trainer in Virtual Reality (MIST-VR) using criteria levels based on expert performance. *Surgical Endoscopy, 18,* 660-665.

Garcia, E., Romero, C., Ventura, S., de Castro, C., & Calders, T. (2011). Association rule mining in learning management systems. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S. J. d. Baker (Eds.), *Handbook of educational data mining* (pp. 93-106). Boca Raton, FL: CRC Press.

Garris, R., Ahlers, R., & Driskell, J. E. (2002). Games, motivation, and learning: A research and practice model. *Simulation and Gaming, 33*(4), 441-467.

Gee, J. P. (2011). Reflections on empirical evidence on games and learning. In S. Tobias & J. D. Fletcher (Eds.), *Computer games and instruction* (pp. 223-232). Charlotte, NC: Information Age Publishers.

Gopher, D., Weil, M., & Bareket, T. (1994). Transfer of skill from a computer game trainer to flight. *Human Factors, 36*, 387-405.

Gremmen, H.,& Potters, H. (1997). Assessing the efficacy of gaming in economics education. *Journal of Economics Education, 28*, 291-303.

Hadwin, A. F., Nesbit, J. C., Code, J., Jamieson-Noel, D., & Winne, P. H. (2007). Examining trace data to explore self-regulated learning. *Metacognition and Learning, 2,* 107-124.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge, MA: MIT Press.

Hart, S. G.,& Battiste, V. (1992).Field test of a video game trainer. In *Proceedings of the Human Factors Society 36th Annual Meeting* (pp. 1291-1295). Santa Monica, CA: Human Factors Society.

Hershkovitz, A., & Nachmias, R. (2011). Log-based assessment of motivation in online learning. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S. J. d. Baker (Eds.), *Handbook of educational data mining* (pp. 287-297). Boca Raton, FL: CRC Press.

Hickey, D. T., Ingram-Goble, A. A., & Jameson, E. M. (2009). Designing assessments and assessing designs in virtual educational environments. *Journal of Science Education and Technology*, *18*, 187-208.

Hislop, S. J., Hsu, J. H., Narins, C. R., Gillespie, B. T., Jain, R. A., Schippert, D. W., … Killig, K. A. (2006). Simulator assessment of innate endovascular aptitude versus empirically correct performance. *Journal of Vascular Surgery*, *43*(1), 47-55.

Hoffman, D., Pack, S., Zhou, Z., & Turkay, S. (2009). Gender differences in a dance-based math game. In I. Gibson, R. Weber, K. McFerren, R. Carlsen, & D. A. Willis (Eds.), *Proceedings of the Society of Information Technology & Teacher Education International Conference 2009* (pp. 2545-2550). Chesapeake, VA: AACE.

Hsu, S. H., Wen, M. H., & Wu, M. C. (2007). Exploring design features for enhancing players' challenge in strategy games. *CyberPsychology & Behavior*, *10*(3), 393-397.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery, 2*, 283-304.

Hunt, E., & Madhyastha, T. (2005). Data mining patterns of thought. In *Proceedings of the AAAI Workshop on Educational Data Mining* (pp. 31-39). Menlo Park, CA: AAAI Press.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys, 31,* 264-323.

James, F., & McCulloch, C. (1990). Multivariate analysis in ecology and systematic: Panacea or Pandora's box? *Annual Review of Ecology and Systematics*, *21*, 129-166.

Johnsen, K., Raij, A., Stevens, A., Lind, D. S., & Lok, B. (2007). The validity of a virtual human experience for interpersonal skills education. In *Proceedings of the SIGCHI Conference on Human Factors in Computer Systems (CHI'07)* (pp.1049-1058). New York, NY: ACM.

Kato, P. M., Cole S. W., Bradlyn, A. S., & Pollock, B. H. (2008). A video game improved behavioral outcomes in adolescents and young adults with cancer: A randomized trial. *Pediatrics, 122*, 305-317.

Kay, J., Koprinska, I., & Yacef, K. (2011). Educational data mining to support group work in software development projects. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S. J. d. Baker (Eds.), *Handbook of educational data mining* (pp. 173-185). Boca Raton, FL: CRC Press.

Kay, J., Maisonneuve, N., Yacef, K., & Zaïane, O. (2006). Mining patterns of events in students' teamwork data. In *Proceedings of the Educational Data Mining Workshop* (pp. 1‑8).

Ke, F. (2008). A case study of computer gaming for math: Engaged learning from gameplay? *Computers & Education*, *51*, 1609-1620.

Kebritchi, M., Hirumi, A., &Bai, H. (2010). The effects of modern mathematics computer games on mathematics achievement and class motivation. *Computers & Education, 55*(2), 427-443.

Kelly, D.,& Tangney, B. (2005). "First Aid for You": Getting to know your learning style using machine learning. In *Proceedings of the IEEE international Conference on Advanced Learning Technologies* (pp. 1-3).

Kerr, D., & Chung, G. K.W.K. (2012). Identifying key features of student performance in educational video games and simulations through cluster analysis. *Journal of Educational Data Mining*, *4,* 144-182.

Kim, J. H., Gunn, D. V., Schuh, E., Phillips, B. C., Pagulayan, R. J., & Wixon, D. (2008). Tracking real-time user experience (TRUE): A comprehensive instrumentation solution for complex systems. In *Proceedings of the 26th annual SIGCHI Conference on Human Factors in Computing Systems* (pp. 443-452).

Koedinger, K. R., Baker, R. S.J.d., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2011). A data repository for the EDM community: The PSLC DataShop. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S.J.d. Baker (Eds.), *Handbook of educational data mining* (pp. 43-55). Boca Raton, FL: CRC Press.

Kopriva, R., Gabel, D., & Bauman, J. (2009). *Building comparable computer-based science items for English learners: Results and insights from the ONPAR project*. Paper presented at the National Conference on Student Assessment (NCSA), Los Angeles, CA.

Kotsiantis, S., Patriarcheas, K., & Xenos, M. (2010). A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems*, *23,* 529-535.

Krier, C., Francois, D., Rossi, F., & Verleysen, M. (2007). Feature clustering and mutual information for the selection of variables in spectral data. In *Proceedings of the 2007 European Symposium on Artificial Neural Networks (ESANN 2007)* (pp. 157-162).

Ksristofic, A. (2005). Recommender system for adaptive hypermedia applications. In M. Bielikova (Ed.), *Proceedings of the 1st Student Research Conference in Informatics and Information Technology (IIT.SCR 2005)* (pp. 229-234). Bratislava, Slovakia: Vazovova 5.

Laffey, J. M., Espinosa, L., Moore, J., & Lodree, A. (2003). Supporting learning and behavior of at-risk young children: Computers in urban education. *Journal of Research on Technology in Education, 35*, 423-440.

Lee, C.-Y.,& Chen, M.-P. (2009). A computer game as a context for non-routine mathematical problem solving: The effects of type of question prompt and level of prior knowledge. *Computers & Education, 52*, 530-542.

Lee, H., Plass, J. L., & Homer, B. D. (2006). Optimizing cognitive load for learning from computer-based science simulations. *Journal of Educational Psychology, 98*, 902-913.

Leighton, J. P., & Gierl, M. J. (2007). Defining and evaluating models of cognition used in educational measurement to make inferences about examinees' thinking processes. *Educational Measurement: Issues and Practice, 26*(2), 3-16.

Lepper, M. R., & Malone, T. W. (1987). Intrinsic motivation and instructional effectiveness in computer-based education. In R. E. Snow & M. J. Farr (Eds.), *Aptitude, Learning, and Instruction, Volume 3: Cognitive and Affective Process Analyses* (pp. 255-286). Hillsdale, NJ: Erlbaum.

Linn, R. L., Baker, E. L., & Dunbar, S. B. (1991). Complex, performance-based assessment: Expectations and validation criteria. *Educational Researcher*, *20*(8), 15-21.

Liu, M., & Bera, S. (2005). An analysis of cognitive tool use patterns in a hypermedia learning environment. *Educational Technology Research and Development*, *53*(1), 5-21.

Maechler, M. (2012). *cluster: Cluster analysis extended Rousseeuw et al*. R package version 1.14.3 [Computer Software]. Retrieved from http://cran.r-project.org/web/packages/cluster/index.html

Madhyastha, T., & Hunt, E. (2009).Mining diagnostic assessment data for concept similarity. *Journal of Educational Data Mining*, *1*, 72-91.

Malcom, S. M., Chubin, D. E., & Jesse, J. K. (2004). *Standing our ground: A guidebook for STEM educators in the Post-Michigan Era.* Washington, DC: American Association for the Advancement of Science.

Masip, D., Minguillon, J., & Mor, E. (2011). Capturing and analyzing student behavior in a virtual learning environment. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S.J.d. Baker (Eds.), *Handbook of educational data mining* (pp. 339-351). Boca Raton, FL: CRC Press.

Mayer, R. E., Mautone, P., & Prothero, W. (2002). Pictorial aids for learning by doing in a multimedia geology simulation game. *Journal of Educational Psychology, 94*, 171-185.

McLaren, B.M., Koedinger, K.R., Schneider, M., Harrer, A., & Bollen, L. (2004). Bootstrapping novice data: Semi-automated tutor authoring using student log files. In *Proceedings of the Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes, Seventh International Conference on Intelligent Tutoring Systems (ITS-2004).*

McNeil, N. M., & Alibali, M. W. (2005). Why won't you change your mind? Knowledge of operational patterns hinders learning and performance on equations. *Child Development, 76*(4), 883-899.

Mehrens, W. A. (1992). Using performance assessment for accountability purposes. *Educational Measurement: Issues and Practice, 11*(1), 3-9.

Merceron, A., & Yacef, K. (2004). Mining student data captured from a web-based tutoring tool: Initial exploration and results. *Journal of Interactive Learning Research, 15,* 319-346.

Miller, G. A., & Chapman, J. P. (2001). Misunderstanding analysis of covariance. *Journal of Abnormal Psychology, 110*(1), 40-48.

Minaei-Bidgoli, B., Kashy, D. A., Kortemeyer, G., & Punch, W. F. (2003). Predicting student performance: An application of data mining methods with the educational web-based system LON-CAPA. In *Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference* (pp. 13-18).

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2004). *A brief introduction to evidence centered design* (CRESST Tech. Rep. No. 632). Los Angeles, CA: University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing.

Mobasher, B., Dai. H.-H., Luo, T., Sun, Y. Q., & Zhu, J. (2000). Integrating web usage and content mining for more effective personalization. In *Proceedings of the First International Conference on Electronic Commerce and Web Technologies* (pp.165-176).

Moreno, R.,& Mayer, R. E. (2000). Engaging students in active learning: The case for personalized multimedia messages. *Journal of Educational Psychology, 96,* 165-173.

Mostow, J., Beck, J. E., Cuneao, A., Gouvea, E., Heiner, C.,& Juarez, O. (2011). Lessons from Project LISTEN's session browser. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S.J.d. Baker (Eds.), *Handbook of educational data mining* (pp. 389-416). Boca Raton, FL: CRC Press.

Muehlenbrock, M. (2005). Automatic action analysis in an interactive learning environment. In C. Choquet, V. Luengo, & K. Yacef (Eds.), *Proceedings of the workshop on Usage Analysis in Learning Systems at AIED-2005.*

National Council of Teachers of Mathematics.(2000). *Principles and standards for school mathematics*. Reston, VA: Author.

National Mathematics Advisory Panel. (2008). *Foundations for success: The final report of the National Mathematics Advisory Panel.* Washington, DC: U.S. Department of Education.

National Research Council. (2011). *Learning science through computer games and simulations*. Washington, DC: The National Academies Press.

National Science and Technology Council. (2011). *The federal science, technology, engineering, and mathematics (STEM) education portfolio*. Washington, DC: Executive Office of the President.

Olive, J., & Lobato, J. (2008). The learning of rational number concepts using technology. In K. Heid & G. Blume (Eds.), *Research on technology in the learning and teaching of mathematics* (pp. 1-53). Greenwich, CT: Information Age Publishing.

Pahl, C., & Donnellan, C. (2003). Data mining technology for the evaluation of web-based teaching and learning systems. In *Proceedings of the 7th International Conference on E-learning in Business, Government and Higher Education* (pp. 1-7). AACE.

Papastergiou, M. (2009). Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Computers & Education*, *52*, 1-12.

Parchman, S. W., Ellis, J. A., Christinaz, D., & Vogel, M. (2000). An evaluation of three computer-based instructional strategies in basic electricity and electronics training. *Military Psychology*, *12*, 73-87.

Pavlik, P. I., Cen, H., Wu, L., & Koedinger, K. R. (2008). Using item-type performance covariance to improve the skill model of an existing tutor. In R. S. J. d. Baker, T. Barnes, & J. E. Beck (Eds.), *Proceedings of the 1st International Conference on Educational Data Mining* (pp. 77-86).

Perera, D., Kay, J., Koprinska, I., Yacef, K., & Zaïane, O. R. (2009). Clustering and sequential pattern mining of online collaborative learning data. *IEEE Transactions on Knowledge and Data Engineering, 21*(6), 759-772.

Quellmalz, E. S., & Haertel, G. (2004). *Technology supports for state science assessment systems.* Paper commissioned by the National Research Council Committee on Test Design for K-12 Science Achievement. Washington, DC: The National Academies Press.

Quellmalz, E. S., & Pellegrino, J. W. (2009). Technology and testing. *Science*, *323*, 75-79.

R Development Core Team. (2010). *R: A language and Environment for Statistical Computing* [Computer Software]. Retrieved from http://www.R-project.org

Radatz, H. (1979). Error analysis in Mathematics education. *Journal for Research in Mathematics Education*, *10*(3), 163-172.

Rahkila, M., & Karjalainen, M. (1999). Evaluation of learning in computer based education using log systems. In *Proceedings of 29th ASEE/IEEE Frontiers in Education Conference (FIE '99)* (pp. 16-22).

Ramnarayan, S., Strohschneider, S., & Schaub, H. (1997). Trappings of expertise and the pursuit of failure. *Simulation & Gaming*, *28*(1), 28-43.

Ravenscroft, A.,& Matheson, M. P. (2002). Developing and evaluating dialogue games for collaborative e-learning. *Journal for Computer Assisted Learning, 18,* 93-111.

Ricci, K. E., Salas, E., & Cannon-Bowers, J. A. (1996). Do computer based games facilitate knowledge acquisition and retention? *Military Psychology*, *8,* 295-307.

Robinet, V., Bisson, G., Gordon, M., &Lemaire, B. (2007). Searching for student intermediate mental steps. In *Proceedings of the Workshop for Data Mining for User Modeling at the 11th International Conference on User Modeling* (pp. 101-105).

Rodrigo, M. M. T., Anglo, E. A., Sugay, J. O., & Baker, R. S. J. d. (2008). Use of unsupervised clustering to characterize learner behaviors and affective states while using an intelligent tutoring system. In *Proceedings of the International Conference on Computers in Education* (pp.49-56).

Rodrigo, M. M. T., Baker, R. S. J. d., D'Mello, S., Gonzalez, M. C. T., Lagud, M. C., Lim, S. A. L.,… Viehland, N. J. B. (2008). Comparing learners' affect while using an intelligent tutoring system and a simulation problem solving game. In B. P. Woolf, E. Aïmeur, R. Nkambou, & S. Lajoie (Eds.), *Proceedings of the 9th International Conference on Intelligent Tutoring Systems (ITS 2008)* (pp. 40-49).

Romero, C., Gonzalez, P., Ventura, S., del Jesus, M. J., & Herrera, F. (2009). Evolutionary algorithms for subgroup discovery in e-learning: A practical application using Moodle data. *Expert Systems with Applications, 39,* 1632-1644.

Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications, 35,* 135-146.

Romero, C., Ventura, S., Espejo, P. G., & Hervas, C. (2008). Data mining algorithms to classify students. In R. S. J. d. Baker, T. Barnes, & J. E. Beck (Eds.), *Proceedings of the 1st International Conference on Educational Data Mining* (pp. 8-17).

Romero, C. Ventura, S., Pechenizkiy, M., & Baker, R. S. J. d. (2011). *Handbook of educational data mining*. Boca Raton, FL: CRC Press.

Romero, C., Ventura, S., Zafra, A., & de Bra, P. (2009). Applying web usage mining for personalizing hyperlinks in web-based adaptive educational systems. *Computer Education, 53,* 828-840.

Ronen, M.,& Eliahu, M. (1999). Simulation as a home learning environment: Students' views. *Journal of Computer Assisted Learning, 15,* 258-268.

Roussos, L., Stout, W., & Marden, J. (1998). Using new proximity measures with hierarchical cluster analysis to detect multidimensionality. *Journal of Educational Measurement*, *35*, 1-30.

Rowley, S. (2000). Profiles of African American college students' educational utility and performance: A cluster analysis. *Journal of Black Psychology*, *26*(1), 3-26.

Rupp, A. A., Gushta, M., Mislevy, R. J., & Shaffer, D. W. (2010). Evidence centered design of epistemic games: Measurement principles for complex learning environments. *The Journal of Technology, Learning, and Assessment*, *8*(4). Retrieved from http://ejournals.bc.edu/ojs/index.php/jtla/article/view/1623/1467

Ruspini, E. H. (1969). A new approach to clustering. *Information and Control*, *15*, 22-32.

Scheuer, O., Muhlenbrock, M., & Melis, A. (2007). Results from action analysis in an interactive learning environment. *Journal of Interactive Learning Research*, *18*(2), 185-205.

Segers, E.,& Verhoeven, L. (2005). Long-term effects of computer training of phonological awareness in kindergarten. *Journal of Computer Assisted Learning*, *21*, 17-27.

Serrano, E. L.,& Anderson, J. E. (2004). The evaluation of food pyramid games, a bilingual computer nutrition education program for Latino youth. *Journal of Family and Consumer Sciences Education*, *22*, 1-16.

Shen, L.P.,& Shen, R.M. (2004). Learning content recommendation service based-on simple sequencing specification. In W. Liu, Y. Shi, & Q. Li (Eds.), *Proceedings of the 3rd International Conference on Advances in Web-Based Learning (ICWL 2004)* (pp. 363-370).

Sherif, A., & Mekkawi, H. (2010). Excavation Game: Computer-aided-learning tool for teaching construction engineering decision making. *Journal of Professional Issues in Engineering Education and Practice*, *136(*4), 188-196.

Shih, B.-Y., & Lee, W.-I. (2001). The application of nearest neighbor algorithm on creating an adaptive on-line learning system. In *Proceedings of the 31st ASEE/IEEE Frontiers in Education Conference* (pp. 10-13).

Shute, V. J. (2011). Stealth assessment in computer-based games to support learning. In S. Tobias & J. D. Fletcher (Eds.), *Computer games and instruction* (pp. 503-524). Charlotte, NC: Information Age Publishers.

Siebert, D.,& Gaskin, N. (2006). Creating, naming, and justifying fractions. *Teaching Children Mathematics*, *12*(8), 394-400.

Sison, R., Numao, M., & Shimura, M. (2000). Multistrategy discovery and detection of novice programmer errors. *Machine Learning*, *38*, 157-180.

Spector, J. M., & Ross, S. M. (2008). Special thematic issue on game based learning. *Educational Technology Research and Development, 56,* 509-510.

Srikant, R.,& Agrawal, R. (1995). Mining sequential patterns: Generalizations and performance improvement. In *Proceedings of the Fifth International Conference Extending Database Technology (EDBT),* Avignon, France.

Su, I.-H., Hung, P.-H., Hwang, G.-J., & Lin, Y.-F. (2010). *An automatic scoring system for PDA integrated ecology observation worksheet.* Paper presented at the 6th IEEE International Conference on Wireless, Mobile, and Ubiquitous Technologies in Education.

Su, J.-M., Tseng, S.-S., Wang, W., Weng, J. F., Yang, J. T. D., & Tsai, W.-N. (2006). Learning portfolio analysis and mining in SCORM compliant environment. *Educational Technology and Society, 9*(1), 262-275.

Sung, Y.-T., Chang, K.-E., & Lee, M.-D. (2008). Designing multimedia games for young children's taxonomic concept development. *Computers & Education, 50*(3), 1037-1051.

Talavera, L., & Gaudioso, E. (2004). Mining student data to characterize similar behavior groups in unstructured collaboration spaces. In *Proceedings of the Workshop on Artificial Intelligence in Computer Supported Collaborate Learning* (pp. 17-22).

Tanner, T., & Toivonen, H. (2010). Predicting and preventing student failure – Using the k nearest neighbor method to predict student performance in an online course environment. *International Journal of Learning Technology, 5*(4), 356-377.

Thompson, M.,& Irvine, C. (2011). Active learning with the CyberCIEGE video game. In *Proceedings of the 4th Conference on Cyber Security Experimentation and Test (CSET'11)*. Berkeley, CA: USENIX Association. Retrieved from https://www.usenix.org/legacy/events/cset11/tech/final_files/Thompson.pdf?CFID=282938153&CFTOKEN=16832848

Tian, F., Wang, S., Zheng, C., & Zheng, Q. (2008). Research on e-learner personality grouping based on fuzzy clustering analysis. In *Proceedings of the 12th International Conference on Computer Supported Cooperative Work in Design (CSWD'2008)* (pp.1035-1040).

Tobias, S., Fletcher, J. D., Dai, D. Y., & Wind, A. P. (2011). Review of research on computer games. In S. Tobias & J. D. Fletcher (Eds.), *Computer games and instruction* (pp. 127-222). Charlotte, NC: Information Age Publishing.

Tompson, G. H.,& Dass, P. (2000). Improving students' self-efficacy in strategic management: The relative impact of cases and simulations. *Simulation & Gaming, 31,* 22-41.

U.S. Department of Education. (2010). *Transforming American education: Learning powered by technology*. Washington, DC: Office of Educational Technology.

U.S. Department of Education. (2012). *Enhancing teaching and learning through educational data mining and learning analytics: A policy brief.* Washington, DC: Office of Educational Technology.

Ueno, M.,& Nagaoka, K. (2002). Learning log database and data mining system for e-learning: On line statistical outlier detection of irregular learning processes. In *Proceedings of the International Conference on Advanced Learning Technologies* (pp. 436-438).

Van Eck, R. (2006). The effect of contextual pedagogical advisement and competition on middle-school students' attitude toward mathematics and mathematics instruction using a computer-based simulation game. *Journal of Computers in Mathematics and Science Teaching*, *25*, 165-195.

Vee, M. N., Meyer, B., & Mannock, M. L. (2006). Understanding novice errors and error paths in object-oriented programming through log analysis. In *Proceedings of the Workshop on Educational Data Mining* (pp. 13-20).

Vellido, A., Castro, F., & Nebot, A. (2011). Clustering educational data. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S. J. d. Baker (Eds.), *Handbook of educational data mining* (pp. 75-92). Boca Raton, FL: CRC Press.

Venables, W. N., & Ripley, B. D. (2002). *Modern Applied Statistics with S: Fourth Edition*. New York, NY: Springer. R package version 7.3-8 [Computer Software]. Retrieved from http://cran.r-project.org/web/packages/class/index.html

Vendlinski, T. P., Delacruz, G. C., Buschang, R. E., Chung, G. K. W. K., & Baker, E. L. (2010). *Developing high-quality assessments that align with instructional video games* (CRESST Report 774). Los Angeles, CA: University of California, National Center for Research on Evaluation, Standards, and Student Testing.

Vogt, W., & Nagel, D. (1992). Cluster analysis in diagnosis. *Clinical Chemistry*, *38*, 182-198.

Warren, S. J., Dondlinger, M. J., & Barab, S. J. (2008). A MUVE towards PBL writing: Effects of a digital learning environment designed to improve elementary student writing. *Journal of Research on Technology and Learning*, *41*(1), 113-140.

Wekesa, E., Kiboss, J., & Ndirangu, M. (2006). Improving students' understanding and perception of cell theory in school biology using a computer-based instruction simulation program. *Journal of Educational Multimedia and Hypermedia*, *15*(4), 397-410.

Whitehill, B. V., & McDonald, B. A. (1993). Improving learning persistence of military personnel by enhancing motivation in a technical training program. *Simulation and Gaming*, *24*, 10-30.

Wiebe, J. H., & Martin, N. J. (1994). The impact of a computer-based adventure game on achievement and attitudes in geography. *Journal of Computing in Childhood Education*, *5*, 61-71.

Wilson, A. J., Revki, S. K., Cohen, D., Cohen, L., & Dehaenel, S. (2006). An open trial assessment of "The Number Race," an adaptive computer game for remediation of dyscalculia. *Behavioral and Brain Functions*, *2*(20), http://www.behavioralandbrainfunctions.com/content/2/1/20.

Yudelson, M. V., Medvedeva, O., Legowski, E., Castine, M., Jukic, D., & Rebecca, C. (2006). Mining student learning data to develop high level pedagogic strategy in a medical ITS. In *Proceedings of the AAAI Workshop on Educational Data Mining* (pp. 1-8).

Zhou, M., Xu, Y., Nesbit, J. C., & Winne, P. H. (2011). Sequential pattern analysis of learning logs: Methodology and applications. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S. J. d. Baker (Eds.), *Handbook of educational data mining* (pp. 107-121). Boca Raton, FL: CRC Press.