Classifying behavior to elucidate elegant problem solving in an educational game

Laura Malkiewich
Teachers College,
Columbia University
525 W 120th St.
New York, NY 10027
Laura.malkiewich@
tc.columbia.edu

Ryan S. Baker
Teachers College,
Columbia University
525 W 120th St.
New York, NY 10027
baker2@
tc.columbia.edu

Valerie Shute
Florida State
University
3205G Stone Building
1114 West Call St.
Tallahassee, FL
32306
vshute@fsu.edu

Shimin Kai
Teachers College,
Columbia University
525 W 120th St.
New York, NY
10027
smk2184@
tc.columbia.edu

Luc Paquette
University of Illinois,
Urbana Champaign
383 Education
Building
1310 S. Sixth St.
Champaign, IL 61820
lpaq@illinois.e

ABSTRACT

Educational games have become hugely popular, and educational data mining has been used to predict student performance in the context of these games. However, models built on student behavior in educational games rarely differentiate between the types of problem solving that students employ and fail to address how efficacious student problem solutions are in game environments. Furthermore, few papers assess how the features selected for classification models inform an understanding of how student behaviors predict student performance. In this paper, we discuss the creation and consideration of two models that predict if a student will develop an elegant problem solution (the Gold model), or a non-optimal but workable solution (the Silver model), in the context of an educational game. A pre-determined set of features were systematically tested and fit into one or both of these models. The two models were then examined to understand how the selected features elucidate our understanding of student problem solving at varying levels of sophistication. Results suggest that while gaming the system and lack of persistence indicate non-optimal completion of a problem, gaining experience with a problem predicts more elegant problem solving. Results also suggest that general student behaviors are better predictors of student performance than level-specific behaviors.

Keywords

Educational games; Problem solving, Classifiers.

1. INTRODUCTION

Educational games can be a great way to enhance learning; in some cases games lead to better learning than standard instructional activities [5, 22]. Yet while understanding how students learn in educational games is important, not much work has been done on modeling student learning in educational games that are open-ended, where students have a lot of freedom to explore. Furthermore, although there has been work on modeling behavior in games and educational learning environments to predict performance in these environments [6, 10, 13, 14, 16, 20] or more generally in school [4], there is not a lot of work that specifically looks at student problem solving strategies in games. Analyzing how students solve complex problems is a key part of understanding student learning in a domain [1, 3, 12], especially in open-ended environments [2]. For this reason, we are investigating student problem solving techniques in order to better understand the nature of student behavior and performance in open-ended educational games.

One key problem solving skill for learning is the ability to produce elegant solutions as well as workable solutions [8, 17], especially as one of the key markers of expertise in a field is the ability to solve problems more elegantly than a novice [11]. Even though there has been research on how to model different student approaches to problem solving [7] there has not yet been sufficient work on modeling the behaviors associated with elegant problem-solving vs. creating workable but less-optimal solutions to problems, especially in game environments. This paper examines how students solve problems to create elegant versus non-optimal, workable, solutions to problems in open-ended educational games. We study this issue in the context of Physics Playground, an open-ended discovery based learning game where students learn about Newtonian physics while trying to solve problems.

2. THE GAME: PHYSICS PLAYGROUND

Physics Playground, formerly called Newton's Playground [19], is an educational game that measures and supports knowledge of conceptual physics for middle and high school students. The game requires students to draw simple machines (consisting of ramps, levers, pendulums, and springboards) that act in accordance with Newton's laws of force and motion. In each level of the game, students are tasked with freehand drawing these machines, which are used to get a green ball to hit a red balloon. In addition to drawing machines, students can draw objects that interact with the ball directly in order to get the ball to reach the balloon. For example, students can draw objects made to fall and hit the ball directly, causing the ball to move. These objects are called "divers" in the context of the game. Students can also draw objects through the ball to move it up slightly. This technique is called "stacking" and is considered a form of "gaming the system" [21]. Similarly, students can click on the ball to "nudge" it forward slightly, if need be, without drawing an object at all. When students finally find a way to hit the red balloon with the green ball, they have completed the level, and are awarded a badge based on their performance.

Students can either receive a gold badge, silver badge, or no badge, depending on their performance in any given level. Badges are awarded according to the efficiency of the student's solution to a problem — determined by the number of objects a student draws in his or her attempt to solve a given problem. For most levels, gold badges are awarded if the student solves the problem by drawing three or fewer objects. Silver badges are awarded if the student solves the problem, but draws more objects. Each level is designed so that one simple machine (a ramp, springboard, pendulum or lever) will optimally solve the given problem. Accordingly, badges for performance are also tied to the type of machine that a student drew in the given level. For example, if a student creates an efficient solution to a level using a ramp, then the student would be awarded a "gold ramp" badge upon completion of the level. Badges are awarded as a means to give students feedback about the efficiency of their solution, so students can reflect on their solution quality. Badges are not necessarily constructed for motivational purposes. Student badges are referred to as "trophies" in the context of the game, and are displayed in the top right hand side of the screen upon level completion.

The game consists of seven "playgrounds", or game worlds, that each contains 10-11 problems. In total there are 74 problems in the entire game. Problems are ordered by difficulty, and problem difficulty is determined by a number of factors including the location of the ball to the target, the magnitude and location of obstacles between the ball and the balloon, the number of agents required to get the ball to the balloon, the novelty of the problem. Students do not have to move through the game in a linear fashion. All levels are unlocked and accessible to students when the game starts (i.e., level access does not depend on a student's performance or progress in the game). Therefore, students can choose to go to any playground and work on any problem that they wish. That being said, there is a logical ordering to the levels, and many students do choose to go through the game in a linear fashion.

3. METHOD

3.1 The Study

This project is based on data collected during a prior study using Physics Playground. A more detailed description of the study population and methods can be found in [9, 18].

3.1.1 Participants

This data is from a study on 137 8th and 9th grade students who attended a diverse K-12 school in the southeastern United States.

3.1.1.1 Procedure

Students played the game in class for about 2.5 hours across four days of the study. Days 1 and 4 of the study consisted of student assessments, including a pretest and isomorphic posttest of students' knowledge of physics concepts. Learning data will not be discussed in the context of this paper [for learning data see 9, 18]. Days 2 and 3 of the study, as well as the first half of Day 4, consisted entirely of gameplay.

3.1.1.2 Measures

Physics Playground captured student log data during gameplay. The final data set consisted of 2,603,827 lines of action codes across the 137 students. Data collected included over seventy variables including information on student progression through the game, time stamps for actions, metrics on student drawings, gameplay actions, and badge awards. Across the 137 students, 919 levels were completed, 203 gold badges were awarded and 500 silver badges were awarded.

3.2 Model Selection

Two models were built for the purpose of distinguishing which features indicate elegant problem solving, and which indicate non-optimal problem solving. The first model was built to classify the award of a gold badge, where problem solutions are optimal (Gold Model). The second model was built to classify the award of a silver badge, where students solve a level, but in a non-optimal way (Silver Model). Levels that a student attempted but did not complete (levels where the student was not awarded a badge) were not used in this analysis.

By building two models, we were able to more effectively differentiate between features that predict elegant problem solving and features that predict non-optimal problem solutions more effectively. For example, creating two models allows for the identification of features that positively load onto one model but negatively load onto another. In turn, understanding these distinctions allows for a deeper understanding of how different levels of various features are indicative of the two types of problem solving. Badges were used as labels because they are the game's proxy for assessing student problem solution quality by marking the efficiency of a student's solution. Although badges in many modern games are used for motivational purposes, for the purpose of this project, we were only interested in what badges indicated about the elegance of a student's problem solution.

Features were created, tested, and iteratively improved upon, across a variety of classification algorithms. During this process, the J48 algorithm, which is Weka's implementation of the C4.5 algorithm [15], consistently provided the strongest predictive power, while protecting against over fitting. For this purpose, when it came to final feature selection and model creation, J48 was the sole algorithm used.

The models were built on less than half of the student data (61 students) so that the remaining test set could later be used to validate and test the final models. In order to validate the models during model creation and feature selection, batch level cross-validation was used. Each student was randomly assigned into 1 of 10 batches, and 10-fold validation was used to assess model goodness. Kappa was used as a measure of model fit.

3.3 Feature Selection

To make the two models, gold and silver labels were made. The gold label had a value of 1 if the student was awarded a gold badge, and a value of 0 if the student was awarded any other kind of badge (or no badge). A label for silver was created in the same way. The original log data tied each badge to the type of machine it awarded a badge for, but for the purpose of this project badge color and machine type were separated into two different features. This was done in part because we wanted to see if machine type affected which type of badge was awarded and in part because making machine type part of the label would result in models predicting what machine the student was building. Instead, we wanted to simply assess how successful students were at solving any given problem, regardless of the nature of the problem given.

Over fifty features were created and assessed for their goodness in predicting badge awards on any given level. The feature engineering process started with a restructuring of the raw student data logs to the problem-level (raw logs came at the action level) because the label of interest categorized student performance at the problem-level grain size. This process was then followed by a descriptive analysis of the variables that came out of this re-structured data, followed by structured brainstorming to elicit ideas about the types of features that could be built out of this data. Features were then created to measure certain constructs (e.g., time on task, gaming the system behavior, etc.) and behaviors of interest. Once a core set of features was created, colleagues and system experts were consulted about the quality, interest, and potential effectiveness of those features. Features were then iterated on. New features were created in an attempt to both measure constructs in more ways (e.g., measuring time on task by looking at time on level, standardized time, or just time spent drawing objects) and to measure different student behaviors and constructs that the first set of features failed to measure. Features were then refined based on colleague and system expert feedback and used in single feature models to assess feature quality. An iterative process of feature creation, peer consulting, and feature refinement then continued for several more cycles until the final set of fifty features had been created.

Once all features had been created, single-feature models were used to choose the seventeen features that were the best predictors of any given construct. For example, Time on Level in minutes was determined to be a better classification of the amount of time that a student spent on a level than standardized time.

The final seventeen features were then ordered in terms of their goodness within a single-feature J48 model, under student-level cross-validation. The best feature was added, and then a recursive process was used where additional features were tested in the same order to determine whether adding that feature improved model goodness, as measured by an increase in kappa. Only features that improved kappa were added. The final gold model contained fourteen features, and the final silver model contained nine features.

3.4 Feature Descriptions

The final seventeen features used for model creation are listed and described below in addition to which model they ended up being included in. Features are listed in the order that they were tested and selected.

Sum Elapsed (silver): The total amount of time that a student spent actively drawing objects up until that point in the game. For example, if a student spends 90 seconds actively drawing objects in Level 1, and then 30 seconds drawing during Level 2, then Sum Elapsed by the end of Level 2 would have a value of 120 seconds.

Time on Level (both): The total amount of time spent playing the level that the student is being awarded the badge for (in seconds).

Nudge Count (gold): The total number of times that the student pressed the ball to nudge it forward a little in the level.

Number of Objects (both): The total number of distinct objects (machines, random lines, weights, etc.) the student drew in the level.

Diver Count (none): The total number of divers that a student created in the level.

Pause Before End (both): Binary indicator of whether or not the student hit the pause button as their last action before the level ended. Usually this happens when students wants to exit out of a level before completing the level. In this case, students would neither be awarded a gold badge nor a silver badge.

Ball Count (both): The number of balls a student uses in a level. If a student knocks a ball off the screen or if the ball provided to the student falls to the bottom of the screen, then it disappears and the student gets a new ball to try again.

Max Velocity Y (both): The maximum velocity that any ball a student used in a level traveled in the y direction (up and down). Velocity values in the Physics Playground system are given in meters-kilogram-second (MKS) units.

Max Velocity X (gold): The maximum velocity that any ball a student used in a level ever traveled in the x direction (left and right).

Erased Object Count (silver): Number of objects that a student drew, and then erased in the level. Students can erase an object that they have drawn by clicking on it.

Stack Count (both): Number of times student drew an object through the ball in order to move the ball up.

Badge Before (gold): Binary indicator of whether or not a student has received a badge (of any color) on this level before.

Played Before (gold): Binary indicator of whether or not a student has played this level before.

Average Free-fall Distance (gold): Free-fall distance is a measure of how far any divers fell before striking a ball. This feature averages across all those distances in the level. Units are percentage of the game screen. So if the diver falls half the distance of the game screen, this would have a value of 0.5.

Restart Count (gold): The number of times a student re-started the level.

Play Count (gold): The number of times that a student has played the current level before. Restarts are not included in this count. A student has to have either completed the level or made an attempt at the level, left the level, and then returned, in order for it to contribute towards this play count.

Machine (both): The type of machine that should be created to optimize movement of the ball to the target. There is one machine per level and they can take the form ramp, lever, pendulum, or springboard.

3.5 Final Models

The final J48 gold classification model with ten-fold student batch cross-validation, which was built on half the data, had a Kappa value of 0.69, and the silver classification model had a Kappa of 0.83. The other half of the data was held out for future analysis comparing the models developed here to other, future models. As is evident from the features mentioned above, seven features fit into both the gold and silver classification models. Those features were Time on Level, Number of Objects, Pause Before End, Ball Count, Max Velocity Y, Stack Number, and Machine. Seven features only fit the gold classification model; those were Nudge Count, Max Velocity X, Badge Before, Played Before, Average Free-fall Distance, Restart Count, and Play Count. Finally, two features only fit the silver classification model. Those were Sum Elapsed and Erased Object Count.

3.6 Qualitative Analysis of Models

The primary goal of this project was to use classification models to help elucidate how student behavior predicts gold and silver badge acquisition differently. For this reason, we take a more qualitative look at which features were included in each model, which were included in both, and which were included in neither.

Table 1 indicates how each of the features loaded onto each of the models when used in a single-feature model (machine type does not have a numeric value, so it is not included in the table). Since both models were built using J48 decision trees, this is simply a proxy for the general loading of each feature on the model outcomes, and not a comprehensive measure of how each feature fits into each model.

Table 1. Feature loadings onto each model

Feature	Gold Model	Silver Model
Sum Elapsed	-	Negative
Time on Level	Negative	***Positive***
Nudge Count	Negative	-
Number of Objects	Negative	***Positive***
Diver Count	-	-
Pause Before End	Negative	Negative
Ball Count	***Positive***	Negative
Max Velocity Y	Negative	***Positive***
Max Velocity X	***Positive***	-
Erased Object Count	-	***Positive***
Stack Count	Negative	***Positive***
Badge Before	Negative	-
Played Before	Negative	-
Average Free-fall Distance	Negative	-
Restart Count	***Positive***	-
Play Count	***Positive***	-

3.6.1 Features included in both models

Features that were included in both models mostly helped indicate whether the student was able to achieve optimal performance or simply workable solutions. For the majority of the features that were in both models, the value was higher for non-gold and higher for silver, indicating that these behaviors were typical of students who developed workable yet non-optimal solutions.

For example, Time on Level was a good indicator of which students produced non-optimal, yet workable solutions. Students who spent a very short time on the level could have entered a level and then immediately quit, so they were likely to not receive a badge. However, longer time in level is associated with a badge but not a gold badge. This loading is likely because students who spend a long time on a level are struggling more or drawing more and those students are therefore less likely to develop the most optimal solution in a single level attempt.

Other features that were higher for non-gold and silver were Number of Objects, Max Velocity Y, and Stack Count. It makes sense that students who drew more objects would get silver, because they are doing more work than students who quit the level (no badge) and students who developed optimal solutions (gold badge). Also, badges are awarded in accordance with the number of objects a student draws in his or her attempt to solve a given problem, so it makes sense that this feature would be a significant indicator of performance. Stack Count could have been a good indicator of whether students solved a problem optimally or non-optimally because students who are stacking a lot could be trying to game the system, likely because they don't know how to solve the problem more effectively using machines. These students are likely to get a silver badge if they complete the problem, because stacking requires drawing many objects.

Only one feature that appeared in both models was higher for both non-gold and non-silver, Pause Before End. This is likely because students who paused before the end of the level were quitting, and therefore did not receive a badge at all. However, that was not always the case.

It is curious that students who had a higher Ball Count per level were more likely to produce optimal solutions; the value for ball count was higher for gold and non-silver indicators. This may be because students who created optimal solutions were experimenting more, and therefore going through more balls, but without spending too much time or drawing too many objects. This behavior could be indicative of students who are quickly iterating on a single idea, or thinking of what to do before drawing objects. (On some levels balls keep dropping down until you draw an object underneath to catch the ball, so the longer you spend without drawing an object, the more balls you use).

3.6.2 Features that only fit the gold model

Features that only fit the gold model are interesting because they specifically separate those who were able to solve problems elegantly as opposed to students who could not find an optimal solution to the problem. The features fit three general categories, relative to whether or not they indicate experience, shallow strategies, or efficiency.

Features that indicate experience include Badge Before, Played Before, Play Count, and Restart Count. It is interesting that Badge Before and Played Before, which are both binaries, indicate non-Gold performance while Play Count and Restart Count indicate gold performance. This indicates that if a student is working on a problem they have completed or played once before, they are not likely to develop an optimal solution, but the more they play a level, the closer they are to get to an optimal solution. Students who have played the level before have some experience with the problem space, even if they did not complete the level previously and that experience could help them determine an optimal problem solution. Play Count and Restart Count tell the model the precise amount of experience the current student has had with a level. Students who re-start or play a level more often might be optimizers, aiming to iterate several times on their problem solution in an attempt to find the best approach to solving the problem. They might be thinking more critically about the choices they are making and choosing to come back to a level or start it again when they've determined that they have acquired the skill or knowledge necessary to now perform more effectively. Resetting also enables students to clear their screens of all objects, and start over, so they can approach the problem afresh. This can be a good strategy for students who want to try going in a different direction instead of iterating on an earlier idea, and it can lead to more efficient problem attempts later.

Nudge Count is a feature that indicates shallow strategies, or even potentially gaming the system. Students who nudge the ball a lot are trying to make the ball move without using a drawn machine to move the ball. This could lead to effectively moving the ball without drawing more objects, which could lead to a problem solution despite a low object count, which would result in a gold badge. Or, it could indicate a student who is nudging because they are struggling a lot with the problem, perhaps because they have already drawn many objects, but are unable to get the ball to move effectively, so they try to nudge it along.

The other features associated with gold badges but not silver badges measure how efficiently students are building machines. These include Average Free-fall Distance and Max Velocity X. Max Velocity X is a predictor of gold badges while Max Velocity Y can predict gold and silver badges, because Max Velocity X is a more effective measure of how well a student has constructed his or her machine. If a ball is dropped from the starting point, then regardless of how effective the student's machine is, the ball will, in many cases, hit the same maximum velocity as it falls because all balls in the Physics Playground interface follow the laws of physics, and therefore accelerate at g. However, how fast a ball moves in the x direction is a direct result of how well a student's designed machine moved the ball in that direction. Likewise, Average Free-fall distance measures student machine efficiency, because students have to carefully choose where to draw divers so that they have a desired effect on ball movement. Divers that are positioned too far away might not hit the desired target, requiring another driver to be drawn for the desired effect. Therefore, both these features are found in this model because they are able to successfully classify effective and efficient student construction choices.

3.6.3 Features that only fit the silver model

Only two features were associated with silver badges but not gold badges. They were Sum Elapsed and Erased Object Count. Both of these features describe the behaviors of students who are tinkering to iterate to a solution. Sum Elapsed negatively loads on the model, suggesting that it indicates ineffective tinkering, while Erased Object count positively loads on the model, suggesting that it indicates effective yet inefficient tinkering. Sum Elapsed is a measure of how much effort a student has put into the game, up until that point in time. A student who has spent a lot of time drawing objects across all prior game levels will have a higher Sum Elapsed value. This is higher for non-silver badges, maybe in part because students who spend a lot of time drawing on levels are less likely to complete the level they are on. This could be because students are making long strokes while doodling, or doing other off task work. On the other hand, students who erase many objects are more likely to get a silver badge. This might be because students who erase a lot are pruning their work if they drew too many objects or made mistakes. These students are more dedicated to completing the current problem, to acquire a badge, but they are not likely to solve the problem in an optimal manner. Therefore Erased Object Count measures an effective problem solving strategy that is not efficient.

3.6.4 Features that fit neither model

It is important to consider not only the features that fit into the models, but also the features that failed to improve either of the models when added. These included Diver Count and a host of other features that were discarded during the feature engineering process, due to the features' low predictive power for behaviors of interest. Interestingly, more specific features involving specific machines or operators were less predictive of student performance than more general variables. Concrete behavior-specific features like Diver Count and Pin Count (pins are small dots that students can add to a drawing to tack an object in place or create a point for an object to rotate around) were less associated with outcomes than were general features like Object Count and Sum Elapsed, which describe student behaviors that span across several actions or several levels. (Note that divers are objects, so when talking about a distinction between these features Object Count is a more general category than Diver Count). It could be that student performance on any particular problem was not as predictive of their problem-solving efficacy as that student's overall behavior. This could suggest that problem solving scaffolding and teaching should focus more on students' overall strategies, rather than level specific strategies. On the other hand, it may simply indicate that none of the more specific features, by themselves, are as predictive as the more general categories that cut across and combine different specific features. It is also important to note that in addition to improving prediction, using more general features also reduces the risk of models over-fitting.

4. DISCUSSION AND CONCLUSION

This analysis of two models built to predict optimal student performance and non-optimal student performance gives us some interesting insights about the kinds of behaviors that predict student performance, and also about the kinds of features that best fit these types of models. Models that describe student performance more generally are more predictive when fed into a J48 decision tree, which can make cutoffs at different values of those feature variables in order to differentiate students who are solving levels optimally, sub-optimally, and not solving levels at all. In turn, features that differentiate optimal performers from all others focus on student experience with the problem space, shallow strategies, and gaming behaviors in addition to measures of student problem solving efficiency. Classifiers of successful but sub-optimal performance tend to describe more exploratory, tinkering behavior while classifiers of elegant problem solving seem to highlight the value of student exposure to a problem and measures of problem-solving efficiency.

These findings give insight into future designs of Physics Playground and other games and open-ended learning environments. To encourage more elegant student problem solving, the learning environment can encourage students to revisit problems, especially after they've created a workable solution, but failed to create an elegant one. Additionally, student feedback about how effective their solution is or what kind of metrics are needed for an optimal solution (e.g., a prompt indicating that for the ball to reach the target it must hit a certain x velocity) could aid students in understanding what more proximal goals they need to fulfill in order to ultimately solve the problem at hand in the most efficient way.

Future work can explore whether similar features are effective for predicting student problem solving in other games. The models discussed here were built on only one game with a unique form of gameplay and specific design constraints, so the study is limited in its generalizability. However, there is the potential for the results of this paper to be used for constructing models for classifying student performance to differentiate between elegant and non-optimal problem solving strategies in other games or open-ended learning environments.

5. ACKNOWLEDGMENTS

We would like to thank the Bill & Melinda Gates Foundation (SOL1071343/APP181499) for their generous support provided for this research, and Ed Dieterle for helpful suggestions and advice. We would also like to thank the students who participated in the study.

6. REFERENCES

[1] Anderson, J. R. (1993). Problem solving and learning. *American Psychologist*, *48*(1), 35.

[2] Blikstein, P. (2011, February). Using learning analytics to assess students' behavior in open-ended programming tasks. In *Proceedings of the 1st international conference on learning analytics and knowledge* (pp. 110-116). ACM.

[3] Chi, M. T., Feltovich, P. J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, *5*(2), 121-152.

[4] Chi, M., Schwartz, D. L., Chin, D. B., & Blair, K. P. (2014, July). Choice-based Assessment: Can Choices Made in Digital Games Predict 6 th-Grade Students' Math Test Scores?. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*(pp. 36-43).

[5] Clark, D. B., Tanner-Smith, E. E., & May, S. K. (2013). *Digital games for learning: A systematic review and meta-analysis*.

[6] Conrad, S., Clarke-Midura, J., & Klopfer, E. (2014). A framework for structuring learning assessment in a massively multiplayer online educational game: experiment centered design, *International Journal of Game Based Learning, 4(1)*, 37-59.

[7] Eagle, M., & Barnes, T. (2014, July). Exploring differences in problem solving with data-driven approach maps. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*(pp. 76-83).

[8] Ertmer, P. A. (2015). *Essential Readings in Problem-based Learning*. Purdue University Press.

[9] Kai, S., Paquette, L., Baker, Bosch, N., D'mello, S., Ocumpaugh, J., Shute, V., & Ventura, M. (2015). A Comparison of face-based and interaction-based affect detectors in physics playground. In *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*(pp. 77-84).

[10] Koedinger, K. R., Stamper, J. C., McLaughlin, E. A., & Nixon, T. (2013, July). Using data-driven discovery of better student models to improve student learning. In *Artificial intelligence in education* (pp. 421-430). Springer Berlin Heidelberg.

[11] Larkin, J., McDermott, J., Simon, D. P., & Simon, H. A. (1980). Expert and novice performance in solving physics problems. *Science*, *208*(4450), 1335-1342.

[12] Larkin‡, J. H., & Reif, F. (1979). Understanding and teaching problem-solving in physics. *European Journal of Science Education*, *1*(2), 191-203.

[13] Martin, J., & VanLehn, K. (1995). Student assessment using Bayesian nets.*International Journal of Human-Computer Studies*, *42*(6), 575-591.

[14] Olsen, J. K., Aleven, V., & Rummel, N. Predicting Student Performance In a Collaborative Learning Environment. In *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*(pp. 211-217).

[15] Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufman: New York.

[16] Rowe, E., Baker, R., Asbell-Clarke, J., Kasman, E., & Hawkins, W. (2014, July). Building automated detectors of gameplay strategies to measure implicit science learning. In *Poster presented at the 7th annual meeting of the international educational data mining society* (pp. 4-8).

[17] Savransky, S. D. (2000). *Engineering of creativity: Introduction to TRIZ methodology of inventive problem solving*. CRC Press.

[18] Shute, V.J., D'Mello, S., Baker, R.S.J., Cho, K., Bosch, N., Ocumpaugh, J., Ventura, M., & Almeda, V. (2015). Modeling how incoming knowledge, persistence, affective states, and in-game progress influence student learning from an educational game. *Computers & Education, 86*, 224-235.

[19] Shute, V.J., Ventura, M., & Kim, Y.J. (2013). Assessment and learning of informal physics in Newton's Playground. *The Journal of Educational Research, 106*, 423-430.

[20] VanLehn, K. (1988). Student modeling. *Foundations of intelligent tutoring systems*, *55*, 78.

[21] Wang, L., Kim, Y. J., & Shute, V. (2013). "Gaming the system" in Newton's Playground. In *AIED 2013 Workshops Proceedings Volume 2 Scaffolding in Open-Ended Learning Environments (OELEs)* (p. 85).

[22] Wouters, P., Van Nimwegen, C., Van Oostendorp, H., & Van Der Spek, E. D. (2013). A meta-analysis of the cognitive and motivational effects of serious games. *Journal of Educational Psychology*, *105*(2), 249.