Struggling to Detect Struggle in Students Playing a Science Exploration Game

Xiner Liu
University of Pennsylvania
xiner@upenn.edu

Stefan Slater
University of Pennsylvania
slater.research@gmail.com

Juliana Ma. Alexandra L. Andres
University of Pennsylvania,
University of Pennsylvania
aandres@upenn.edu

Luke Swanson
University of Wisconsin-Madison
swansonl@cs.wisc.edu

Jennifer Scianna
University of Wisconsin-Madison
jscianna@wisc.edu

David Gagnon
University of Wisconsin-Madison
djgagnon@wisc.edu

Ryan s. Baker
University of Pennsylvania
ryanshaunbaker@gmail.com

ABSTRACT

The real-time detection of when a player is struggling presents an opportunity for game designers to design timely and meaningful interventions, as well as to provide targeted support that improves student learning and engagement. In this paper, we present a struggle detector in the context of students playing the learning game, Wake: Tales from the Aqualab. Using the interaction log data of the game, we engineered four sets of features that captured distinct aspects of gameplay and trained prediction models to identify human-coded cases of students struggling, cross-validating at the student level. Our best-performing detectors have shown some capability in identifying student struggles with modest performance, at an AUC (Area Under the Curve) value of 0.635. We discuss current limitations of this approach, as well as next steps towards providing real-time support within the game.

CCS CONCEPTS

• **Applied computing**; • **Education;**; • **Computing methodologies**; • **Machine learning;**; • **Human-centered computing**; • **Human computer interaction (HCI)**;

KEYWORDS

Education Data Mining, Automated Detector, Struggle Detection, Educational Game

ACM Reference Format:
Xiner Liu, Stefan Slater, Juliana Ma. Alexandra L. Andres, Luke Swanson, Jennifer Scianna, David Gagnon, and Ryan s. Baker. 2023. Struggling to Detect Struggle in Students Playing a Science Exploration Game. In *Companion Proceedings of the Annual Symposium on Computer-Human Interaction in Play (CHI PLAY Companion '23), October 10–13, 2023, Stratford, ON, Canada.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3573382.3616080

CHI PLAY Companion '23, October 10–13, 2023, Stratford, ON, Canada

ACM ISBN 979-8-4007-0029-3/23/10…$15.00
https://doi.org/10.1145/3573382.3616080

1 INTRODUCTION

In recent decades, educational games have gained popularity as a means of engaging students in learning activities that are both entertaining and educational [9, 15, 32, 35], and educational game design generally aims to promote both learning and engagement [28]. To achieve this, it is important to design games that strike a delicate balance between challenging players at their current competence while maintaining a playable level [3]. **[**33**]** recommend that the game's difficulty level should match the player's current ability to achieve this optimal level of engagement.

Finding the exact point where a game's difficulty is high enough to challenge the player, but not so difficult as to discourage them, requires careful attention to the game's design [11]. To prevent difficulty from becoming too high, educational games frequently incorporate interventions such as hints [17] and feedback [1, 30]. However, there are numerous challenges associated with developing effective hints and feedback for game players, such as what the content of the messages should be, or when the messages should be delivered [18]. In order to select the right time to provide a hint, it is useful to identify when a player is struggling during the game, so that scaffolding can be provided at that point, and withheld otherwise [29]. This approach ensures that interventions are only used when they are likely to be effective, promoting optimal learning and engagement outcomes.

The overarching aim of this study is to build automated detectors that identify instances of struggle that students encounter while playing. Our investigation focuses on detecting struggle within *Wake: Tales from the Aqualab* (referred to as *Wake* within this paper for brevity), a game where students take on the role of a scientist working at an ocean-floor research station to learn about scientific research practices.

2 RELATED WORK

In recent years, there has been considerable interest in detecting learner struggle. Much of this work has involved a behavior termed "wheel-spinning", which refers to the phenomenon where a player or learner spends a significant amount of time working without making progress in the learning environment [14]. In the case of games, [25] detected player wheel-spinning within the context of the game *Physics Playground* using a logistic regression model by incorporating features such as player's prior attempts and past performances. [24] utilized Classification and Regression Tree (CART) to detect wheel-spinning based on player's progression and performance in the adaptive game-based learning system *Mastering Math* (MM).

There has been considerable interest in detecting wheel-spinning in other types of learning environments as well, particularly intelligent tutoring systems. For example, [14] proposed a logistic regression model to predict wheel-spinning using features such as students' performances, seriousness, and the number of problems practiced in two intelligent tutors. [26] predicted wheel-spinning in adaptive online courseware, leveraging problem-level features such as student performance, hint usage, response time, and problem difficulty.

There has also been considerable attention focused on predicting if a student will quit a game level or learning activity, a behavior that is often indicative of struggle [2]. In one such study, [20] developed both level-specific and level-agnostic models to identify student quitting in the game *Physics Playground*, using a combination of data on a student's gameplay and overall performance on the level. Another study evaluated the accuracy of common machine learning algorithms with simple, generic features in predicting quitting and performance on assessment tests within two science learning games, *the Crystal Cave* and *Wave Combinator* [13]. Beyond games, [34] predicted student dropout based on discussion forum participation in a Massive Open Online Courses (MOOC) class. [8] predicted both whether students would quit an online mathematics assignment without achieving mastery and whether they would wheel-spin by applying deep learning to interaction data.

Extensive research has also been conducted to detect affective states, such as confusion and frustration, which are commonly associated with the experience of struggle. A significant body of literature in this area has focused on games. [6] proposed sensor-free detection models for identifying affective states in the virtual environment *EcoMUVE*, utilizing features derived from interaction with the game. [19] compared the performance and effectiveness of video-based and interaction-based affect detectors in *Physics Playground*. [16] employed a deep neural network to detect learner frustration in the game-based medical learning environment *TC3Sim*. Within the adventure game *Danger Island*, [7] constructed a deep-learning model based on Electroencephalography (EEG) data to recognize player confusion.

3 CONTEXT

3.1 Overall Game Design

This project takes place within the larger context of developing and studying a middle-school science practices game, *Wake: Tales from the Aqualab*. In *Wake*, players take on the role of a young scientist named Olivia who grew up at an aquatic research station in a Kelp Forest and has just begun practicing science herself. Over the course of the game, players discover new research sites and acquire new scientific tools, learning about complex phenomena and scientific practices alongside Olivia. *Wake* is designed to be played during one or more 45-minute class sessions, and takes approximately 10 sessions to complete.

Wake contains a total of 14 research sites distributed across 4 major ecosystems: Kelp Forest, Coral Reef, Bayou, and Arctic. At each of these major areas, a research station has been established to study and sometimes manage the local ecosystems. Upon arriving at each station, a lead researcher presents the player with different "jobs" that they need help completing. Jobs are often aligned with a fundamental question, such as "What is causing the urchins to multiply so rapidly?" Each job requires the player to discover new information about the ecosystem, present a claim, and provide evidence to support their claim. Players make these discoveries by making direct observations at the research sites, conducting various experiments, and generating various forms of scientific models. Each new insight is catalogued in a tablet-like device as a "species" or a "fact" and can then be presented as evidence for a claim. In this way, the discovery of a new species or fact is a primary form of progression in the game. These jobs gradually expand on the amount of information available to players and the complexity of relationships and ecosystems that they are required to model to maintain an overall level of challenge as players gain proficiency with both the game tools and the learning content.

To scaffold players, *Wake* breaks down the potentially large investigations required by a job into a series of "tasks." Tasks are displayed on the screen as a short prompt, such as "Count the Loggerhead Turtles at the Oil Rig," and are completed when the player navigates to a particular location or makes a specific set of discoveries. While some jobs have a specific linear ordering of tasks, many others assign several tasks a player may work toward in parallel, and still others feature one large task that contains a number of implied subtasks. Therefore, at any moment, there are many ways for a player to make progress.

A second way *Wake* provides scaffolding is thorough dialog with the various characters in the game, most often through a guide character known as "V1ct0r." V1ct0r appears whenever the player acquires a new tool to instruct them in how the tool is used, acting as training for the different interfaces and high-level concepts in the game. V1ct0r and other characters also appear during some conceptually challenging moments to help explain some part of the system. V1ct0r can also be summoned at any time during gameplay for help, and they will attempt to respond with contextually relevant guidance informed by the current job, currently available tools, and current discoveries.

3.2 Identifying When Students Struggle

In the context of studying player interactions with *Wake*, this research attempts to provide a useful mechanism for differentiating productive and unproductive behaviors within the game environment. Due to its open-ended nature and science-based exploration context, players in *Wake* may approach the game's challenges in different ways or even spend time exploring and experimenting the

world of *Wake* outside of any specific job, simply out of curiosity. These manifold approaches to gameplay make it challenging to differentiate between students who are productively experimenting and playing with game parameters from students who are stuck and require help from some external agent. For example, a student may run experiments which do not align to their current job because they just observed a novel species and are trying to identify all relevant information; this behavior would look similar to a student who is unsure which experiments are needed to provide evidence for their current job's task. A detector which is able to identify these moments of struggle could therefore be used to deliver targeted feedback to stuck players, or players who are simply off-task, while allowing curious players to continue their explorations.

4 METHOD

4.1 Data Collection

Wake leverages the Open Game Data system for player interaction data logging [12]. At a technical level, this involves the integration of the opengamedata-unity package into the game, which communicates with cloud-based server infrastructure to capture and record meaningful events that take place during a play session. We refer to these events, collectively, as telemetry data.

Structurally, *Wake* sends telemetry events using a game-agnostic schema. Drawing inspiration from the framework propose by [23], *Wake* records events that can be placed in three general categories: Player Actions, System Events, and Progression Events. In total, there are 33 distinct player actions, 12 system events and 6 progression events that are sent as telemetry data for storage on the Open Game Data server as they occur. Player events include general navigation (e.g., the player piloting to a specific research site), as well as specific actions that can be taken by the player when interacting with each of the game's mechanics (e.g., adding a species data in the modeling tool, selecting specific evidence during an argument). System events include both formative feedback (e.g., the evidence selected by the player is rejected in an argument) and scripted events (e.g., a guide character appears when the player enters a new location). Progression events describe when the player discovers a new fact or species, and when the player completes a single task or entire job. Each of these events is packaged with metadata including the time and sequence in which the event occurred, a player identifier, and details about the event and game state. Combined, these events form a time-sequenced description of all the significant interactions that take place within a play session, in the game's own language.

For this study, anonymous game telemetry data were collected during the month of June 2022 as part of iterative game development and testing in Wisconsin, Maine, and Massachusetts. A total of 3859 gameplay sessions were captured from 501 students during this period, with an average session duration of approximately 40 minutes. The majority of these sessions were part of a classroom study that included four teachers and 336 of their students from a single middle school in Massachusetts. The remainder of the data was generated from smaller and less structured testing implementations at the other locations. 1,009,026 individual telemetry events were collected from these players.

4.2 Text Replay of Interaction Logs

To make it easier to examine and explore the data, we utilized a technique called text replays [5]. This involves presenting human-readable segments of interaction data, known as clips, to facilitate both the initial exploration and the final coding process. This method has been used in previous studies to label student affect, disengagement, and learning strategies, such as gaming the system [4], confrustion [21], player goals [10] and self-regulated learning strategies, such as whether a student is using a table to plan their analyses [27]. This approach achieves a level of reliability similar to classroom observations and is 2-6 times faster compared to other methods of generating labels, such as classroom observations, screen replay, and retrospective think/emote-aloud protocols [5].

The length and granularity of the text replay clips can vary depending on the researcher's intended predictions. For this study, analysis was focused on struggle behaviors that occurred within each task that a student was required to solve. A single job can be comprised of multiple tasks, each of which have a separate goal state that advances the game. The text replays were set up at the grain size of entire tasks, with each clip containing a student's actions as they worked through a single task. Switching jobs, starting new jobs, or completing a job were all treated as the end of an existing task, and the beginning of a new one. Coding at the level of task required a comprehensive examination across questions and game settings, as students were free to move between different areas, menus, and options of the game within these tasks. The text replays were presented using a Python window, which showed a subset of actions and allowed the coder to page to later or earlier actions within the clip. A pair of human coders labeled these clips as "Struggle", "Not Struggle", or "Bad Clip" if the clip was fragmented in some way (such as if a student had just completed a job, but not yet accepted a new one). Cases of disagreement were then presented to members of the *Wake* team and consensus coded via group discussion. Clips were selected with interval sampling – coders selected line numbers incrementing by 10,000, and then coded starting from the first clip that began after that point. For example, in the first round of coding, clips that began at lines 10037, 20038, and 30015 were coded. Subsequent coding rounds staggered the starting point for these intervals. In the second round of coding, we instead started at 6500 and incremented by 10000 from there.

4.3 Data Preparation

4.3.1 Feature Engineering. A total of 58 features were engineered for each data sample across four categories: General, System Use, Patterns & Habits, and Contextual. The General features provide insight into players' overall interactions with different game mechanics and functionalities, while System Use features capture how players interact with in-game systems and panels. The Patterns & Habits features aim to identify recurring patterns and habits in students' gameplay, such as the frequency and duration of pauses between actions. Finally, the Contextual features offer supplementary information, including players' locations during specific actions. All features are engineered based only on past and present actions and do not incorporate future data since the models will be used to investigate the relationship between current struggle and later performance and for real-time interventions.

4.3.2 Feature Aggregation. Given that struggle is a complex outcome that arises from a sequence of actions and is therefore likely to be predicted by interrelated action patterns, we aggregate the features at the task level. In the case of contextual features, which track moment-to-moment details such as the player's current location for every action, we averaged these features across all of a student's actions. For example, if a student spent 20 of their 40 total actions in the Experiment Room, the feature in_exp_room was assigned a value of 0.50. Following feature aggregation, the resulting dataset comprised 16,704 data points, with a feature space of 58 dimensions.

4.4 Machine Learning Algorithms

To evaluate the efficacy of the struggle detector, we employed five-fold student-level cross-validation. This involved randomly dividing the students into five groups, with each group serving as a test set while the remaining four groups were used to develop the detector. By cross-validating at this level, we aimed to assess the degree to which our detector can generalize to new students.

We applied nine popular classification algorithms commonly used in educational data mining, including step regression, logistic regression, XGboost, Naïve Bayes, J48 decision trees, KNN, support vector machine, random forest, and artificial neural networks, to develop the sensor-free struggle detectors. Feature selection was performed using forward selection within each fold of cross-validation, which repeatedly added the feature that contributed the most to the model's goodness on the training set until no further improvement was possible.

We used AUC ROC as metrics to measure the effectiveness of the detector. The AUC gauges the likelihood that the model assigns a higher predicted probability to a randomly selected positive case than a randomly selected negative case. A model with an AUC of 0.5 performs at the chance level, while a model with a score of 1.0 is perfect. The AUC is often used in conjunction with the Receiver Operating Characteristic (ROC) curve, which is a graphical representation of the performance of a binary classification model that plots the precision (or positive predictive value) on the y-axis and recall (or sensitivity or true positive rate) on the x-axis for various thresholds used by the model to make predictions.

To gain a deeper understanding of the engineered features and their relationship to the constructs and detectors, we calculated the SHapley Additive exPlanations (SHAP) values for each feature within each test set [22]. These values were averaged across 5 testing sets and ranked based on their absolute values. A positive average SHAP value for a feature indicated that the feature was a positive predictor of struggle. In other words, the model was more likely to detect struggle when the value of that feature was higher. Conversely, features denoted with a negative sign had negative average SHAP values. This indicated that these features predicted an absence of struggle. Features with a SHAP value of 0 had no direct impact on the model's prediction, but they may still be useful for providing context or aiding in model interpretability.

4.5 Attempts to Improve Model Performance

Among the 288 clips examined, only 10.76% of them were identified as instances of struggle, resulting in a highly imbalanced dataset. This poses a significant challenge in machine learning, as most algorithms tend to perform poorly on minority class prediction in these cases. In an effort to improve our model performance, we resampled the minority cases through both duplicating existing examples and synthesizing new ones using the Synthetic Minority Oversampling Technique (SMOTE).

Additionally, we adopted an iterative feature engineering approach as proposed by [31]. The approach employs misclassifications from previous models to uncover patterns of behaviors that current features fail to capture. Based on these patterns, we engineered 4 new contextual features in an attempt to enhance model performance. The 4 additional features, in addition to the original set of 58 features, were forwarded to the feature selection process to assess their impact on the models' performances.

5 RESULTS

5.1 Model Performance

The Naïve Bayes algorithm demonstrated the best performance for struggle detection when trained and evaluated on non-resampled dataset, using features selected through forward feature selection. For that algorithm, the AUC ROC obtained from 5-fold student-level cross-validation for the struggle detectors was 0.635, with a standard deviation across folds of approximately 0.05. These results indicate that the detector was capable of detecting struggling students more accurately than chance levels. However, the results also highlight the fact that there is still considerable room for improvement.

Table 1 presents the features selected through the forward feature engineering approach, organized into four categories and ranked based on their absolute SHAP values. The directionality of each feature, indicating its impact on the model's predictions, is indicated in the last column using positive and negative signs. Given that Naïve Bayes is the best performing model, we can assume that the presence or absence of one feature has no impact on the presence or absence of any other feature.

Out of the 22 features listed, 6 features were drawn from the General category, 9 from the System Use category, 5 from the Patterns & Habits category, and 2 from the Contextual category. The System Use category accounts for the highest number of features among all categories, indicating the significant influence of players' system-using behaviors on the predictive models. For example, players' interactions with the game's built-in functionalities, or their requests for assistance from the guide V1ct0r.

6 DISCUSSION AND CONCLUSION

In this paper, we have presented automated detectors that aim to identify student struggle solely from log files in *Wake: Tales from the Aqualab*. We have engineered multiple sets of features to capture the players' behavior patterns and contextual information over time, and our best-performing detectors are better than chance at identifying struggle. However, our initial attempts to further improve model performances did not yield significant results. In particular, employing resampling techniques on the minority classes led to a decrease in AUC, while the features generated through iterative feature engineering failed to pass the forward feature selection stage in any fold. This observation leads us to suggest that the struggle cases are of high complexity or poor quality, which limits

Table 1: The features in the final detectors and their respective directionality according to SHAP values.

Category	Feature Definition	Directionality
General (6)	How many times has the player dived?	-
	How many times has the student changed rooms?	-
	How many seconds has the student spent during the current task?	-
	How many times has the player opened the world map?	-
	How many times has the player visited the modeling room?	+
	How many times has the player visited the argumentation room?	-
System Use (9)	How many times has the player opened the job summary tab?	-
	How many times has the player opened the model tab?	-
	How many times has the player opened the job board?	+
	How many times has the player opened the bestiary?	+
	How many times has the player opened the environment tab?	+
	How many times has the player asked for help?	+
	How many system-generated "fact rejected" messages (submitted fact is incorrect and/or rejected during argumentation) has the player received?	+
	How many times has the player opened the species tab?	+
	How many facts have the player obtained?	+
Patterns & Habits (5)	How many experiments has the player conducted?	-
	How many argumentations has the player conducted?	-
	How many times has the player run an experiment and not received a fact?	-
	How many times has the player run a model and not achieved synchronization?	-
	How many times has the player changed the parameters of the experimentation tank?	+
Contextual (2)	How many seconds has the player spent in the current station?	+
	Is the player currently in the kelp forest, bayou, arctic, or coral station?	-

the potential of SMOTE to generate new examples that can address the class imbalance and enhance the model's performance. Moreover, further exploration and experimentation are necessary to gain a deeper understanding of the underlying factors contributing to whether the struggle can be detected and for identifying effective strategies for improving model performances in this context.

The current performance of this model is not sufficient for use in a targeted intervention for most situations, where a prediction is being made for an individual student. However, it is sufficient to use in aggregate analyses that take model confidence levels into account, such as calculating the average rate of struggle in different levels and comparing them. Future research could explore alternative definitions for struggle, further feature engineering, and alternative methods for aggregating data.

6.1 Limitation

The study presented here has some potential limitations that need to be acknowledged. One significant limitation is the lack of a clear and standardized definition of the term "struggle." This ambiguity may have restricted the accuracy of our labels, as it is possible that some players who were labeled as "struggling" may only have been experiencing confusion or frustration due to the game's user interface or design, rather than their own skill level.

Moreover, the generalizability of the model we have developed may also be limited by the lack of demographic information for the players. It is important to test models on a diverse range of students to ensure its effectiveness across different backgrounds and experiences. In this case, based on how the game is used, we were unable to obtain this information – without it, we cannot be confident that the model will perform well on students from different populations. In other words, this lack of demographic information makes it impossible to check for algorithmic biases which can occur when a model's performance varies significantly across mutually exclusive groups that are separated by factors that cannot be easily changed, such as gender and race/ethnicity.

6.2 Future Direction

This paper is part of a broader project that aims to leverage insights gained from game log data for the purpose of understanding and identifying players' behaviors in order to provide real-time support and foster data-driven iterative design. Despite its modest performance, our struggle detection model is the first step towards identifying factors that may lead to potential disengagement or frustration among students during their gameplay experiences. Our current detector's performance is insufficient to support real-time intervention, but it is sufficient to identify situations where students are struggling in aggregate, and better understand why. Through doing so, we can improve our game's design to help scaffold students past these difficulties. Ultimately, it is our hope that by identifying when students are struggling, we will be able to provide players

with more enriching gameplay experiences and enhance knowledge and interest in science among the tens of thousands of K-12 learners learning from *Wake*.

REFERENCES

[1] We would like to thank NSF #DRL-1907437 for support of this project.
[2] REFERENCES
[3] Deanne M. Adams and Douglas B. Clark. 2014. Integrating self-explanation functionality into a complex game environment: Keeping gaming in motion. Computers & Education, 73, 149– 159. DOI:https://doi.org/10.1016/j.compedu.2014.01.002
[4] Maryam Abdulsalam Ali, Noraidah Sahari Ashaari, Siti Fadzilah Mat Noor, and Suhaila Zainudin. 2022. Identifying students' learning patterns in online learning environments: A literature review. International Journal of Emerging Technologies in Learning (Online), 17(8), 189. DOI:https://doi.org/10.3991/ijet.v17i08.29811
[5] Maria-Virginia Aponte, Guillaume Levieux, and Stephane Natkin. 2011. Measuring the level of difficulty in single player video games. Entertainment Computing, 2(4), 205-213. DOI:https://doi.org/10.1016/j.entcom.2011.04.001
[6] Ryan S. Baker and Adriana M.J.A. de Carvalho. 2008. Labeling student behavior faster and more precisely with text replays. In Proceedings of the 1st International Conference on Educational Data Mining, 2008, 38–47.
[7] Ryan S. Baker, Albert T. Corbett, and Angela Z.Wagner. 2006. Human classification of low-fidelity replays of student actions. In Proceedings of the Educational Data mining Workshop at the 8th International Conference on Intelligent Tutoring Systems, 2002, 29-36.
[8] Ryan S. Baker, Jaclyn Ocumpaugh, Sujith M. Gowda, Amy M. Kamarainen, and Shari J. Metcalf..2014. Extending Log-Based Affect Detection to a Multi-User Virtual Environment for Science. In Lecture Notes in Computer Science, 290–300. DOI:https://doi.org/10.1007/978-3-319-08786-3_25
[9] Mohamed Sahbi Benlamine and Claude Frasson. 2021. Confusion detection within a 3D adventure game. In Intelligent Tutoring Systems: 17th International Conference, ITS 2021, 17, 387-397. DOI:https://doi.org/10.1007/978-3-030-80421-3_43
[10] Anthony F. Botelho, Ashvini Varatharaj, Thanaporn Patikorn, Diana Doherty, Seth A. Adjei, and Joseph E. Beck. 2019. Developing early detectors of student attrition and wheel spinning using deep learning. In IEEE Transactions on Learning Technologies, 12(2), 158-170. DOI:https://doi.org/10.1109/TLT.2019.2912162
[11] Thomas M. Connolly, Elizabeth A. Boyle, Ewan MacArthur, Thomas Hainey, and James M. Boyle. 2012. A systematic literature review of empirical evidence on computer games and serious games. Computers & education, 59(2), 661-686. DOI:https://doi.org/10.1016/j.compedu.2012.03.004
[12] Kristen E. DiCerbo and Khusro Kidwai. 2013. Detecting player goals from game log files. In Proceedings of the 6th International Conference on Educational Data Mining, 2013, 314–316.
[13] Julian Frommel, Fabian Fischbach, Katja Rogers, and Michael Weber. 2018. Emotion-based dynamic difficulty adjustment using parameterized difficulty and self-reports of emotion. In Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play, 163-171. DOI:https://doi.org/10.1145/3242671.3242682
[14] David Gagnon and Luke Swanson. In Press. Open Game Data: A Technical Infrastructure for Open Science with Educational Games. Submitted to Joint Conference on Serious Games.
[15] David J. Gagnon, Erik Harpstead, and Stefan Slater. 2019. Comparison of off the shelf data mining methodologies in educational game analytics. In Proceedings of the 12th International Conference on Educational Data Mining, 2019, 38–43. DOI:https://doi.org/10.1002/9781118956588.ch16
[16] Yue Gong and Joseph E. Beck. 2015. Towards detecting wheel-spinning. In Proceedings of the second ACM conference on learning @ scale, 67-74. DOI:https://doi.org/10.1145/2724660.2724673
[17] Robert T. Hays. 2005. The effectiveness of instructional games: A literature review and discussion (Report No. NAWCTSD-TR-2005-004). Orlando FL: Naval Air Warfare Center Training Systems Division. DOI:https://doi.org/10.21236/ada441935
[18] Nathan Henderson, Jonathan Rowe, Luc Paquette, Ryan S. Baker, and James Lester. 2020. Improving affect detection in game-based learning with multimodal data fusion. In Lecture Notes in Computer Science, 228–239. DOI:https://doi.org/10.1007/978-3-030-52237-7_19.
[19] Andrew Hicks, Barry Peddycord, and Tiffany Barnes. 2014. Building games to learn from their players: Generating hints in a serious game. In Lecture Notes in Computer Science, 312–317. DOI:https://doi.org/10.1007/978-3-319-07221-0_39
[20] Cheryl I. Johnson, Shannon K. T. Bailey, and Wendi L. Van Buskirk. 2017. Designing effective feedback messages in serious games and simulations: A research review. Instructional Techniques to Facilitate Learning and Motivation of Serious Games, 119-140. DOI:https://doi.org/10.1007/978-3-319-39298-1_7~
[21] Shiming Kai, Luc Paquette, Ryan S. Baker, Nigel Bosch, Sidney D'Mello, Jaclyn Ocumpaugh, Valerie Shute, and Matthew Ventura. 2015. A comparison of video-based and interaction-based affect detectors in physics playground. In Proceedings of the 8th International Conference on Educational Data Mining, 77-84.
[22] Shamya Karumbaiah, Ryan S. Baker, and Valerie Shute. 2018. Predicting quitting in students playing a learning game. In Proceedings of the 11th International Conference on Educational Data Mining, 167–176.
[23] Diane Marie C. Lee, Ma. Mercedes T. Rodrigo, Ryan S. J. D. Baker, Jessica O. Sugay, and Andrei Coronel. 2011. Exploring the relationship between novice programmer confusion and achievement. In Lecture Notes in Computer Science, 175–184. DOI:https://doi.org/10.1007/978-3-642-24600-5_21
[24] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In Proceedings of the Advances in Neural Information Processing Systems (2017), 4765–4774. DOI:https://doi.org/10.1021/acs.jcim.1c01467.s001
[25] V. Elizabeth Owen and Ryan S. Baker. 2020. Fueling prediction of player decisions: Foundations of feature engineering for optimized behavior modeling in serious games. Technology, Knowledge and Learning 25, 2 (2020), 225–250. DOI:https://doi.org/10.1007/s10758-018-9393-9
[26] V. Elizabeth Owen, Marie-Helene Roy, K. P. Thai, Vesper Burnett, Daniel Jacobs, Eric Keylor, and Ryan S. Baker. 2019. Detecting wheel-spinning and productive persistence in educational games. In Proceedings of the 12th International Conference on Educational Data Mining, 378–383.
[27] Thelma D. Palaoag, Ma. Mercedes T. Rodrigo, Juan Miguel L. Andres, Juliana Ma. Alexandra L. Andres, and Joseph E. Beck. 2016. Wheel-spinning in a game-based learning environment for physics. In Lecture Notes in Computer Science, 234–239. DOI:https://doi.org/10.1007/978-3-319-39583-8_23
[28] Seoyeon Park and Noboru Matsuda. 2018. Predicting students' unproductive failure on intelligent tutors inadaptive online courseware. In Proceedings of the Sixth Annual GIFT Users Symposium, Vol. 6. USArmy Research Laboratory, 131–138.
[29] Michael A. Sao Pedro, Ryan S Baker, Janice D. Gobert, Orlando Montalvo, and Adam Nakama. 2013. Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. User Modeling and User-Adapted Interaction 23, 1 (2013), 1–39. DOI:https://doi.org/10.1007/s11257-011-9101-0
[30] Jonathan P. Rowe, Lucy R. Shores, Bradford W. Mott, and James C. Lester. 2010. Integrating learning and engagement in narrative-centered learning environments. In Lecture Notes in Computer Science, 166–177. DOI:https://doi.org/10.1007/978-3-642-13437-1_17
[31] Asmalina Saleh, Chen Yuxin, Cindy E. Hmelo-Silver, Krista D. Glazewski, Bradford W. Mott, and James C. Lester. 2020. Coordinating scaffolds for collaborative inquiry in a game-based learning environment. Journal of Research in Science Teaching 57, 9 (2020), 1490–1518. DOI:https://doi.org/10.1002/tea.21656
[32] Valerie J. Shute, Ginny Smith, Renata Kuba, Chih-Pu Dai, Seyedahmad Rahimi, Zhichun Liu, and Russell Almond. 2021. The design, development, and testing of learning supports for the physics playground game. International Journal of Artificial Intelligence in Education 31, 3 (2021), 357–379. DOI:https://doi.org/10.1007/s40593-020-00196-1
[33] Stefan Slater, Ryan S. Baker, and Yeyu Wang. 2020. Iterative feature engineering through text replays of model errors. In Proceedings of the 13th International Conference on Educational Data Mining, 503–508.
[34] Kurt Squire. 2005. Changing the game: What happens when video games enter the classroom?. Innovate: Journal of Online Education, 1(6).
[35] Penelope Sweetser and Peta Wyeth. 2005. GameFlow. Computers in Entertainment 3, 3 (2005), 3–3. DOI:https://doi.org/10.1145/1077246.1077253
[36] Diyi Yang, Tanmay Sinha, David Adamson, and Carolyn Penstein Rose. 2013. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In Proceedings of the 2013 NIPS Data-Driven Education Workshop, 1-8.
[37] Michael F. Young, Stephen Slota, Andrew B. Cutter, Gerard Jalette, Greg Mullin, Benedict Lai, Zeus Simeoni, Matthew Tran, and Mariya Yukhymenko. 2012. Our princess is in another castle: A review of trends in serious gaming for education. Review of Educational Research, 82(1), 61-89. DOI: https://doi.org/10.3102/0034654312436980