Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

OVERCOMING CHALLENGES IN EDUCATIONAL STEM GAME DESIGN AND DEVELOPMENT

Katherine Smith

Department of Mathematics
and Statistics
Old Dominion University
5115 Hampton Boulevard
Norfolk, VA 23528, USA

John Shull
Yuzhong Shen

Department of Modeling, Simulation and
Visualization Engineering
Old Dominion University
5115 Hampton Boulevard
Norfolk, VA 23528, USA

Anthony Dean

Engineering Fundamentals Division
Old Dominion University
5115 Hampton Boulevard
Norfolk, VA 23528, USA

Jennifer Michaeli

Engineering Technology Department
Old Dominion University
5115 Hampton Boulevard
Norfolk, VA 23528, USA

ABSTRACT

Captivate is a mobile game for STEM in higher education. In the development of Captivate, the employment of popular game mechanics and development of a set of mathematical tools have allowed developers to overcome some of the problems commonly facing developers of educational STEM games for higher education. Game mechanics from a variety of popular games are incorporated which supports player engagement by providing a familiar narrative to each game. Additionally, mathematical tools have been developed which provide symbolic and numeric manipulation capabilities within the game. These tools allow for graphing, problem generation and display, multiple types of player input, and evaluation of player answers all to be conducted at run time. This paper details how challenges were overcome and addressed in Captivate with examples, namely display and manipulation of complex expressions and the complexity and length of problems.

1 INTRODUCTION

A set of educational games for Science, Technology, Engineering, and Mathematics (STEM) education is currently being developed. Recently the development of Maven, a desktop game that helps students learn and review pre-calculus at the college level, was completed. Now, work is nearing completion on Captivate which is a mobile game that will include topics from calculus and physics. Captivate reuses the modular framework that was developed for Maven which has allowed for many improvements without starting from scratch. Captivate combines this framework with game mechanics from popular games to overcome a number of challenges in developing educational STEM games for higher education.

While there are many games available for STEM education at the primary and secondary school levels, there have been very few games developed to target STEM content in higher education including pre-calculus, calculus, physics, and chemistry. These courses are important in STEM education, particularly in engineering, as they provide the foundation for upper level engineering courses. In this work, we attempted

978-1-5386-3428-8/17/$31.00 ©2017 IEEE

to solve challenges that stand in the way of developing these games, namely display and manipulation of complex expressions and the complexity and length of problems.

Since Captivate is a mobile game developed for all Android and iOS devices, player data is stored in the cloud so that players can maintain their progress across devices. Also, two function classes have been developed to handle symbolic and numeric manipulation of mathematical functions. These classes facilitate graphing, question generation, multiple types of player input, and symbolic evaluation of the player's answer.

The remainder of the paper is organized as follows: Section 2 summarizes challenges in developing educational games for STEM and overviews how they were handled in the present work, Section 3 provides an overview of Captivate highlighting the features that address the challenges from Section 2, and Section 4 provides conclusions and summarizes future work.

2 CHALLENGES IN DESINGING AND DEVELOPING STEM GAMES FOR HIGHER EDUCATION

The challenges facing educational STEM game design and development include successfully incorporating game design and instructional design as well as addressing the complexity and length of problems from calculus and physics. Here each of these challenges will be summarized along with previous approaches and overview of how each challenge was solved in Captivate will be provided.

2.1 Incorporating Game Design with Instructional Design

One of the major challenges in developing educational games is maintaining a focus on content and learning while simultaneously adhering to good game design principles in order to create a game that is engaging and instructive (Rubin 1999; Williams et al. 2004). While there are numerous resources detailing game design considerations (Hunicke et al. 2004; Fullerton 2014; Schell 2014) as well as many well-developed instructional design methodologies (Peterson 2003; Allen and Sites 2012), balancing these considerations can be difficult. In fact, it has been pointed out that both educators and game designers need to be better informed about considerations for educational games and experts from game and instructional design need to be key players on game development teams (Felicia 2011).

Using a modular design has been identified as the best approach to overcome challenges in the design of educational games (Aslan and Balci 2015). In order to address this, we have developed a modular framework and process that provides a template for educational game development (Smith et al. 2018). The main iterative development process defined in this framework involves a focus on the player, content and instructional design considerations and game design considerations. An additional component that makes this framework unique is that it focuses on educational games as a component of a larger educational program or learning environment. This has several benefits including using stakeholders in that learning environment as members of the development team (similar to the game idea generation team from (Aslan and Balci 2015)) and having a well-defined target audience consisting of the students in the education program. In our case, our target audience consists of adult students who are leaving military service and entering higher education in pursuit of engineering degrees.

The framework that accompanies the process discussed above emphasizes modularity in every stage of the design process. While each component should focus on each of the three aspects highlighted in the process, the components themselves should have clean interfaces and connections in order to facilitate modification of individual components with few or no changes to the rest of the application. While there are many benefits to modular design of software in general, the benefits are amplified in the development of educational STEM games by providing flexibility to cover a broad range of topics within a single development model.

In Captivate, this framework has been fully implemented. This allowed the development team to build off previous successes while improving in areas that were not as successful. For example, the entire online player data storage system has been replaced, but there have been very few changes to the game manager that connects this system to the individual games. Additionally, a problem generation and checking system has been developed that has very clean connections to the individual scripts that control each game. This allows each game to request the player be presented with a question upon some game event and then receive back only whether the question was answered correctly or incorrectly. This separation makes it easy to use the same question system in different games or to re-implement a single game to cover additional content areas.

2.2 Display and Manipulation of Complex Expressions

As difficulty in mathematics and science increases, the complexity of the equations and expressions to be manipulated increases as well. Examples of complex expressions that need to be displayed and manipulated are shown in Figure 1. Previously, a framework was developed to allow use of Python libraries to manipulate expressions and generate images of expressions and their corresponding graphs (Smith et al. 2016). As part of that work, an extensive literature review was completed to determine that there were no existing examples of manipulation and display of complex mathematical expressions and graphs in educational games. This search also showed that while there were many games for mathematics education in elementary and early secondary education, there was a lack of games for later secondary and higher education that allowed players to interact with, solve, and graph complex expressions in topic such as pre-calculus and calculus. The framework developed as part of that work was used to develop Maven, an educational game for pre-calculus. Some examples of the equations and graphs generated by this framework are shown in Figure 2.

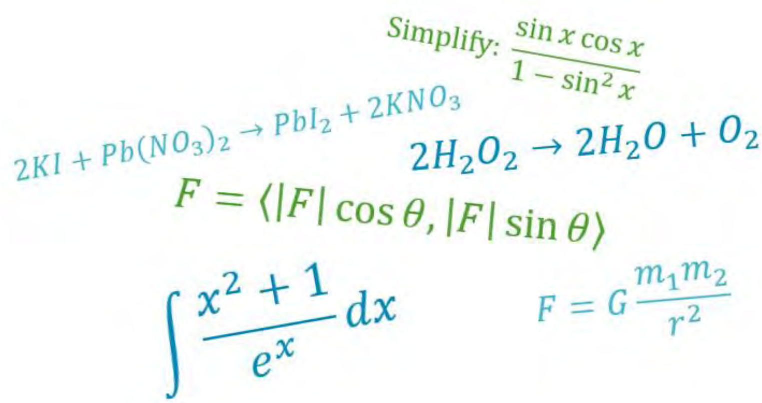
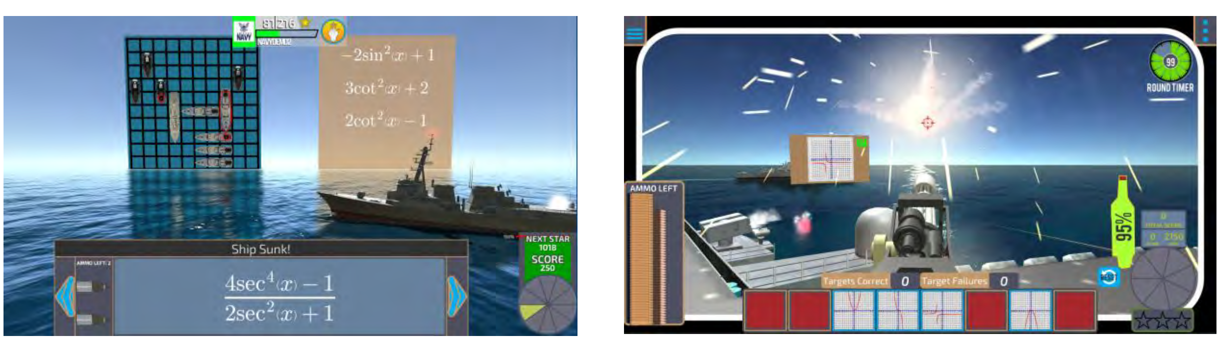


Figure 1: Example equations and expressions from college-level STEM courses.



(a) (b)

Figure 2: Screen captures showing equations and graphs from the previously developed framework.

While the framework allowed for rapid development and incorporation of a variety of problem types by employing the capabilities of Python's symbolic mathematics libraries, generating the images during run-time caused a noticeable delay and would have required an installation of Python was required to be distributed with the application. For these reasons, it was more practical to use the framework on the developer side to generate a large variety of problems with answers that would be distributed with the game. Algorithmically generating a large, random problem set approximates a user experience where problems are generated on-the-fly during gameplay.

Generating the problem and solution images ahead of time results in a trade-off in flexibility. While the player can be presented with a variety of complex problems of varying difficulty, the ability to present the user with a variety of input methods that resemble free response methods is limited. This issue has been addressed in Captivate with the development of a set of classes that handle symbolic manipulation and graphing for a variety of mathematical functions. Additionally, a graphing tool employing numerical methods has been developed that can be used to display the graphs of functions and classify their behavior at each point.

2.3 Problem Complexity and Length

Problems in pre-calculus, calculus, physics and chemistry can require the application of complex rules and principles over many steps. This is necessary to in order to assess the student's understanding of the different concepts involved in the problem solution, but is also unfortunate because it tends to require that the student be provided with each step required and does not reflect their ability to solve a problem independently, from beginning to end (Greenhow 2015). In Maven, this was addressed best in the destroyer game shown in Figure 2a. In this game, the user was presented with a trigonometric expression to simplify and all the steps in the simplification process were hidden on the ships on the game board. Players had to search through the ships to find each of the expressions that were mathematically equivalent to the given expression. There is also a similar game that requires the player to simplify logarithmic expressions. The key to the player's ability to match the equivalent expressions is that there is no time limit. The player can take their time manipulating the expression, using scratch paper if necessary, and then locating the equivalent expressions. Having no time constraint and the player potentially needing to write down the steps can result in less engagement as longer tasks in games can lead to boredom and distractions from the game itself should be avoided (Murphy et al. 2011).

The length and complexity issue is especially difficult to address in STEM problems as there may be a number of paths or approaches that can lead to a correct answer. While there are available tools that will generate answers to a multistep problem, symbolically checking each step of a player's answer is difficult.

In Captivate, considerations between allowing the player to solve the problem independently and keeping them engaged in the game have been balanced. The ability to generate single or multi-step problems has been improved. After each step of a problem the player returns to the game and receives feedback. By reengaging players in game play between problem steps and providing step-by-step feedback, the players will be more engaged. Additionally, players are asked to complete questions with a variety of options for input. On more complex problems, multiple choice is used to prevent the user from becoming frustrated when entering longer, more complex answers on a mobile device. Other questions provide players with the opportunity to enter a more free-form response. In either case, all questions are generated dynamically during run-time and answers are checked symbolically.

3 DESIGN AND DEVELOPMENT OF CAPTIVATE

Captivate is a mobile game that allows players to learn calculus and physics content. Players can choose from a variety of topics based on what they are interested in practicing. Each topic leads to an individual game for that content area. As they successfully complete the levels within each game, the difficulty increases and the problems become more complex. Player data is stored using in the cloud which allows player progress to be maintained even across different devices. Progress information is displayed to players in multiple places, so that they see how they are doing within a single topic or across the entire game. Figure 3 shows images from menus in Captivate. Each of the menu buttons shows the player progress on the associated content area using a radial fill. This is shown in yellow for the button in Figure 3(a) and light blue for Figure 3(b). Since the button in Figure 3(b) will take the player to the game FunctionHero, there is also a darker blue radial fill that allows the player to select the level they want to play. Players can also see their progress on all games along with some summary information in the stats menu shown in Figure 3(c).

(a) (b) (c)

Figure 3: Screen captures from the Captivate menu. (a) and (b) main menu and the function hero game menu, respectively. (c) shows the user statistics menu.

One of the key elements of game design is to incorporate a story or narrative that engages the player and connects them to the game. This presents a challenge for educational games that players may be using to support their study as students may use the game to only review topics that they struggle to remember or find particularly challenging. It is not feasible to have a continuous narrative that flows throughout a game when there are innumerable paths that individual players can take through the game.

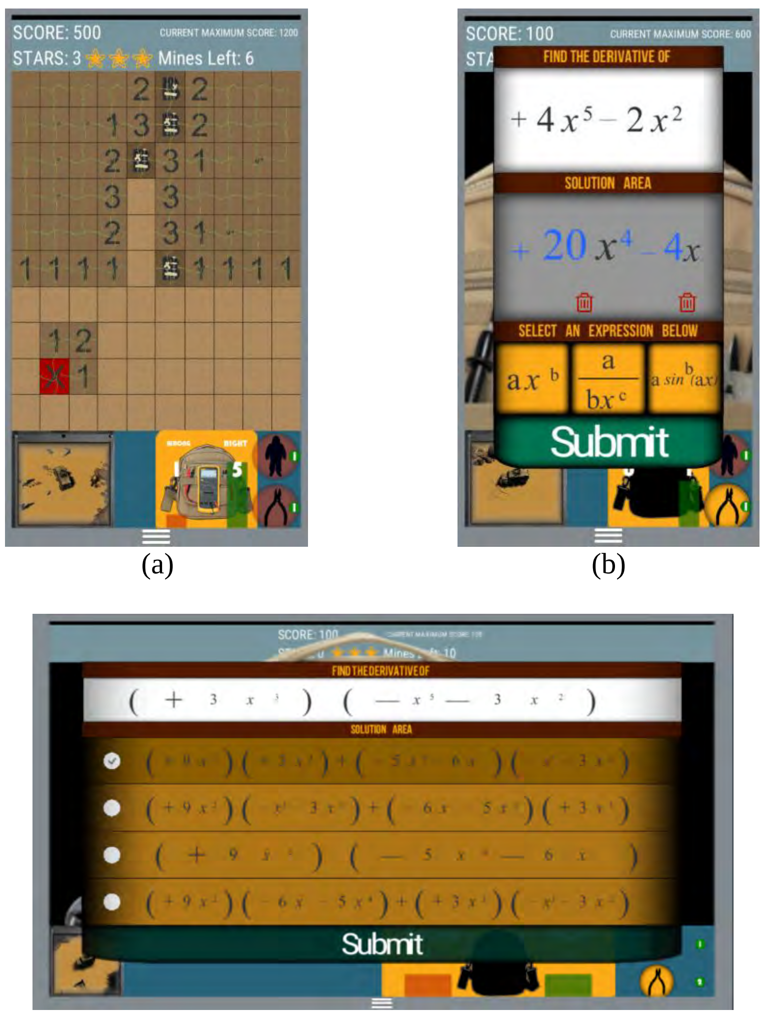
During the development of Maven, one game that received the most enthusiasm at demonstrations was a variation on Battleship (Hasbro 2017). By using familiar game mechanics, players were connecting with a narrative they knew well which alleviated the necessity to have a continuous narrative. Also since players were already familiar with the game mechanics, the challenge of learning to play was minimal allowing the main challenge to be the content. Building upon this success, all the games developed for Captivate use familiar game mechanics. The recommendation to use familiar game mechanics can be perceived as a practical recommendation for the development of educational games. Practical recommendations are identified as important and a gap in existing research into educational games by Felicia (Felicia 2011).

3.1 MathSweeper

MineSweeper has been reported by some in the media as the most successful game ever (Cobbett 2009). While the designation of most successful may be debatable, there is no doubt that this is a well-known game that has enjoyed popularity for more than two and a half decades. In the classic version, players are presented with a grid of buttons. Some of these buttons are hiding mines and will end the game if clicked.

The player is tasked with correctly uncovering all squares that do not contain mines and consequently identifying the locations of the mines.

In MathSweeper, shown in Figure 4, players are presented with the familiar gridded game board and task of finding explosives. The mines have been replaced with improvised explosive devices (IEDs) and players have a bomb suit, which protects them if they accidentally uncover an IED, and a set of wire cutters, which they can use to skip a question. Each time a player taps a square that does not contain a mine, they are presented with a mathematical expression that they need to differentiate as shown in Figure 4(b) and 4 (c). If they answer the question correctly, they are rewarded with information showing the number of mines in surrounding squares. There is also a small chance for the reward to include an additional bomb suit or wire cutters. If they answer incorrectly they are penalized as the square is turned red and they receive an "X" instead of additional information. This provides the player with immediate feedback on their answer. As in the classic game, players can mark squares they determine contain IEDs. The game ends when the player has uncovered all squares that do not contain IEDs. Figure 4(a) shows the game board mid-game where the player has answered five questions correctly and on question incorrectly. The player has also marked four potential IEDs.



(a) (b) (c)

Figure 4: Screen captures from MathSweeper. (a) shows the game in play mode while (b) and (c) show two different questions presented to the player.

The question shown in Figure 4(b) is a single-step problem where players enter their answers by combining polynomial and rational terms and selecting the correct coefficients and exponents. The player has the ability to use the buttons on the right to add as many terms as they need and then tap to set each coefficient or exponent. They can also change the operators by tapping them or remove a term by clicking the trash icon. This provides the player with an experience that is close to freely entering their answer. Once they tap submit, their answer is symbolically compared to the correct answer. The player is provided feedback that indicates whether they were correct or incorrect and also provided with the correct answer.

The question shown in Figure 4(c) is a multi-step problem where the player is required to use the product rule in the first step and then simplify in the second step. In this case, the player selects their answer from a set of four possibilities. The question and all potential answers are generated dynamically when the player taps the square. Though this particular question has two steps, questions can vary in length and combine free response input steps with multiple choice steps. By letting the player know whether or not they are correct after each step, they are receiving immediate feedback. Additionally after each step the player returns to the game board. This keeps the player engaged in learning and play as they work through each problem.

(a) (b)

Figure 5: Screen captures showing SineRun. (a) shows the game in play mode while (b) shows an example of a question generated for the player to answer where they are asked to take the derivative of a trigonometric expression.

3.2 SineRun

Endless runner games are a popular variation of platform games. In these games, the character runs through an endless, procedurally generated world while the player has control of a limited set of actions such as jumping, shooting, or changing lanes. The goal is usually to cover as much distance as possible before the game ends. Power-ups can be included that give the character special abilities such as a burst of speed or increased jumping ability. Obstacles are used to slow the character down or reduce their abilities.

In the Captivate endless runner game, SineRun, shown in Figure 5, the player controls the tank by making it jump as it drives automatically. Power-ups are available that increase the tank's speed while obstacles decrease its speed. In SineRun, the power-ups are gas cans and the obstacles are fire and enemy planes. Additionally, there are separate check points occurring at set time intervals which generate a question involving derivatives of trigonometric functions for the player to answer. The game is paused while the player answers the question. Once the player answers a question, they receive immediate feedback and return to the game. If the question is answered correctly, they are rewarded with a speed boost. Incorrect answers are penalized by slowing the tank down.

The question panel in the endless runner game is the same one that is used in MathSweeper. In SineRun, players are asked to differentiate trigonometric expressions instead of sums of polynomial and rational terms but the freeform input panel is the same from the previous game. This is beneficial for both the player and the developer. The player benefits because they are presented with a familiar tool. They do not have to learn to how to interact with a new tool for each game. In addition, development of games is accelerated because there is no need to develop a question panel for each game. Reusability is enhanced as the game is only required to request a question and then receive a simple right or wrong response once the player has answered.

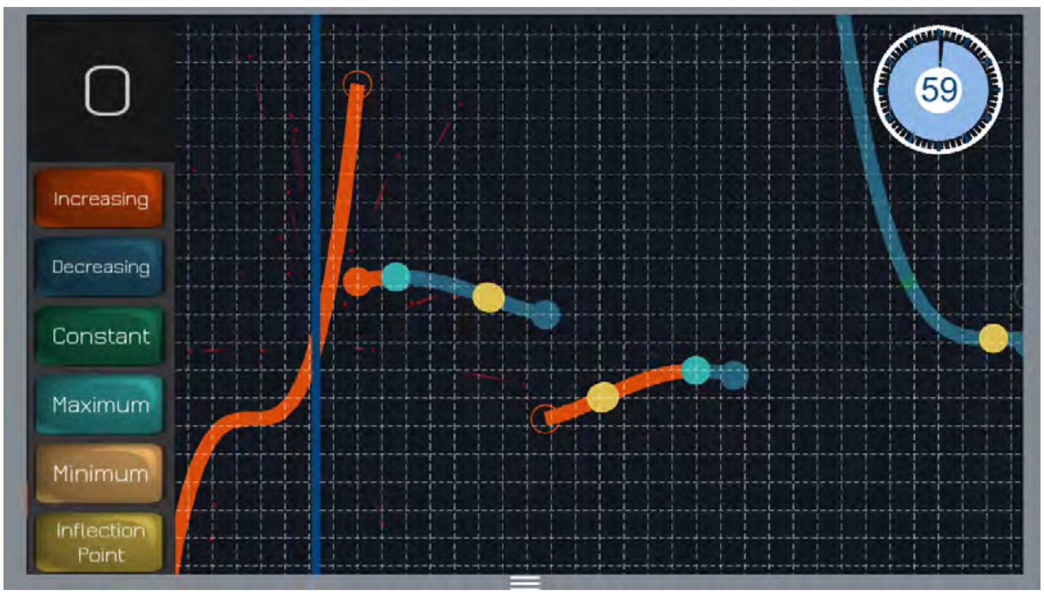


Figure 6: Screen capture from FunctionHero.

3.3 FunctionHero

In the popular console game Guitar Hero (Activision 2017), players are challenged to play along with familiar songs by pressing the correct buttons as the notes scroll toward them on the screen. In FunctionHero, a dynamically generated, piece-wise function scrolls across the screen from right to left as shown in Figure 6. At the point where the function intersects the blue target line, the player is challenged to classify the function behavior as increasing, decreasing or constant and identify critical points as maximum values, minimum values, and inflection points. Since function behavior occurs over a range, players need to hold down the correct button over the entire range to accumulate the maximum score. In order to classify the critical points, the player taps the correct button. In Figure 6, the game is shown in an early level. In early levels, the sections of the function and critical points are color coded to match the button that corresponds to the correct behavior. This is designed to help players learn about function behavior as they learn to play the game. In later levels, the function is shown in solid colors so that the player has to determine the correct behavior at each point.

For this game, a function graphing tool previously developed for Maven was expanded and enhanced to allow for the generation of multiple functions. In addition, numerical methods are used to determine the correct function behavior at each point. The correct function behavior is compared to the player's input and correct classification results in a score increase. Players also receive visual feedback in the form of green or red sparks at the intersection of the function and the target line indicating if their choice is correct or incorrect, respectively.

4 CONCLUSIONS AND FUTURE WORK

Educational game design and development for STEM in higher education requires developers to overcome a number of challenges. Many of these challenges have been overcome in the development of Captivate by employing popular game mechanics to keep players engaged. Modularity in the design has facilitated reuse of components between games and allowed the player data system to be upgraded from previous work. Additionally, classes that perform mathematical computations using both symbolic and numerical methods have been developed.

With the existing tools and menu framework, work will continue to expand Captivate by adding additional games and content areas in calculus and physics. A game covering chemistry is scheduled to be developed next. Additionally, work is underway to plan and carryout an efficacy study for these games. Another interesting area for future work is to modify the game mechanics of the existing games and assess the impact. For example, the SineRun game could be modified to give the player the ability to shoot obstacles in addition to jumping or MathSweeper could be modified to include a starting point and ending point so that the player only has to clear squares along a path. It would be interesting to see if adding additional game elements or tasks effected learning efficacy in the games.

ACKNOWLEDGMENTS

This work was made possible through the Office of Naval Research STEM under ONR GRANT11899718.

REFERENCES

Activision. 2017. "Guitar Hero Live Home". https://www.guitarhero.com/.

Allen, M., and R. Sites. 2012. *Leaving Addie for Sam: An Agile Model for Developing the Best Learning Experiences*. Alexandria, VA: American Society for Training and Development

Aslan, S., and O. Balci. 2015. "Gamed: Digital Educational Game Development Methodology". *Simulation* 91 (4):307-319.

Cobbett, R. 2009. "The Most Successful Game Ever: A History of Minesweeper." *TechRadar*. http://www.techradar.com/news/gaming/the-most-successful-game-ever-a-history-of-minesweeper-596504.

Felicia, P. 2011. *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches: Multidisciplinary Approaches*: IGI Global

Fullerton, T. 2014. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. Boca Raton, FL: CRC press

Greenhow, M. 2015. "Effective Computer-Aided Assessment of Mathematics; Principles, Practice and Results". *Teaching Mathematics and its Applications: An International Journal of the IMA* 34 (3):117-137.

Hasbro. 2017. "Battleship". http://www.hasbro.com/en-us/product/battleship:2560F81B-5056-9047-F55A-F26A61C519C3.

Hunicke, R., M. LeBlanc, and R. Zubek. 2004. "MDA: A Formal Approach to Game Design and Game Research". In *Proceedings of the AAAI Workshop on Challenges in Game AI*.

Murphy, C., D. Chertoff, M. Guerrero, and K. Moffitt. 2011. "Creating Flow, Motivation, & Fun in Learning Games". *The design of learning games. Springer-Verlag*.

Peterson, C. 2003. "Bringing Addie to Life: Instructional Design at Its Best". *Journal of Educational Multimedia and Hypermedia* 12 (3):227-242.

Rubin, A. 1999. "Technology Meets Math Education: Envisioning a Practical Future Forum on the Future of Technology in Education". Office of Education Technology, Washington, D.C.

Schell, J. 2014. *The Art of Game Design: A Book of Lenses*: CRC Press

Smith, K., J. Shull, A. Dean, Y. Shen, and J. Michaeli. 2016. "Sigma: A Software Framework for Integrating Advanced Mathematical Capabilities in Serious Game Development". *Advances in Engineering Software* 100:319-325.

Smith, K., J. Shull, Y. Shen, A. Dean, and P. Heaney. 2018. "A Framework for Designing Smarter Serious Games". In *Smart Universities*, edited by V. Uskov, J. Bakken, R. Howlett, and L. Jain, 263-294. Springer, Cham.

Williams, D. L., R. Boone, and K. V. Kingsley. 2004. "Teacher Beliefs About Educational Software: A Delphi Study". *Journal of Research on Technology in Education* 36 (3):213-229.

AUTHOR BIOGRAPHIES

KATHERINE SMITH is a senior lecturer in the Department of Mathematics & Statistics. She holds an M.S. in Applied and Computational Mathematics from Old Dominion University as well as Bachelor's degrees in Applied Mathematics and Mechanical Engineering. She is currently pursuing her PhD in Modeling, Simulation, Visualization Engineering. Her research interests are in scientific and data visualization; augmented and virtual reality; educational game development for STEM education; and instructional technologies that promote student understanding of mathematics in higher education. Her e-mail address is k3smith@odu.edu.

JOHN SHULL is a PhD student within the Modeling, Simulation, Visualization Engineering program at ODU. He holds an M.E. in Modeling, Simulation, Visualization Engineering and a B.S.B.A. in Marketing and Decision Sciences both from Old Dominion University. His research interests are in the use of virtual environments, augmented and virtual reality engineering, the use of educational games for advancements in education, data visualization techniques with AR/VR technologies, and Agent Based Modeling applications involving complex systems His e-mail address is jshull@odu.edu.

YUZHONG SHEN has expertise and experience in visualization and computer graphics, game-based learning, and modeling and simulation. He is currently an Associate Professor at Old Dominion University within the Modeling, Simulation & Visualization Engineering Department. He received his PhD in Electrical Engineering from the University of Delaware. He holds an M.S. in Computer Engineering from Mississippi State University and a B.S. in Engineering Electrical Engineering from Fudan University. His email address is yshen@odu.edu.

ANTHONY DEAN is an Associate Professor at Old Dominion University within the Engineering Fundamentals Division. He received his PhD in Engineering Management Systems from Old Dominion University. He earned an M.B.A. from the College of William and Mary and a B.S. in Engineering Technology from Old Dominion University. His email address is adean@odu.edu.

JENNIFER MICHAELI has 13 years of experience as a Naval Architect and Program Manager, she currently leads the Marine Engineering Specialty as an Assistant Professor at Old Dominion University within the Engineering Technology Department. She received her PhD in Mechanical Engineering from Old Dominion University. She holds an M.Sc. in Ocean Systems Management from the Massachusetts Institute of Technology and a B.S. in Naval Architecture and Marine Engineering, Webb Institute. Her email address is jgmichae@odu.edu.