

Implementation of Game-Based Programming as a Means to Engage and Excite Students in First-Year Engineering Courses

Kenneth A. Thiemann
Engineering Fundamentals
Michigan Technological
University
Houghton, MI USA
kathiema@mtu.edu

Brett H. Hamlin
Engineering Fundamentals
Michigan Technological
University
Houghton, MI, USA
bhhamlin@mtu.edu

Abstract—Today’s first-year college engineering students are no strangers to spending large amounts of time in the online environment. Social media and gaming have been found to be an arena where multitudes of students enjoy spending hours meeting, developing online social circles, and participating in interactive gaming with others around the world. In our first-year engineering program, we find a student that is very eager to engage in their college career, but one that may not know the many aspects that each engineering discipline has to offer. With roughly 1,000 incoming engineering students at Michigan Technological University, we find that many students can easily become disinterested by the programming heavy nature of our first-year curriculum. Keeping these students challenged, engaged and proud of their accomplishments while looking forward to their next year of college is of paramount importance.

Programming courses in particular pose a distinct challenge to the instructor, as many students may consider them to be dry and uninspiring. With students in this age bracket so closely acquainted with the technology found in the online world, why not leverage this as a means to bring game-oriented programming into their first-year programming classes? In this paper, we assert that by incorporating game design into first-year engineering programming courses, we keep the student excited about their coursework, keep them challenged by their assignments, and give them a programming experience where they can look back on their final product with pride and a sense of accomplishment.

During the first-year engineering experience, it is essential that students exhibit a positive, playful mindset in order to maintain engagement in their classes. With first-year students being very familiar with gaming and the online arena, this paper posits the implementation of game design into their first-year programming courses as a means to elicit excitement and intrigue. In this study, we take a look at three aspects of implementing game-based programming assignments into the engineering curriculum. First, we look at the excitement generated and eagerness to sign up for a class that advertises game programming. Next, we assess to what level the students become and stay engaged when immersed in challenging problems associated with game-based programming. Lastly, we examine the students’ assessment of their programming, and their opinions of how game programming affected their learning in an entry level engineering programming class.

Keywords—*First-Year Engineering, Programming, Game-Based, Gamification*

I. INTRODUCTION

Transition from the high school level to a student’s first year experience in college can be daunting. With the new college experience and all that comes with it, many students find themselves overwhelmed with merely learning to navigate the challenges of this new landscape, let alone keeping up with their first-year mandatory introductory courses. The vast majority of college students that drop out do so in their first semester [1]. For this reason, it is of the utmost importance to keep the incoming engineering student excited, engaged, and challenged with their first-year coursework. This challenge can be especially hard for the instructor when teaching introductory programming.

Programming has provided necessary tools for computer science and engineers for decades. At Michigan Technological University, we have a mandatory two-semester design course sequence for first-year engineering students that focuses heavily on utilizing MATLAB as the primary programming platform. Many of our incoming first-year engineering students are very inquisitive as to why they actually need to learn programming in the first place. Even when the students learn the assets and reasons for programming in their engineering career, many still find it very daunting, difficult, and dry. This paper focuses on the use of game-based programming using MATLAB App Designer to keep the students’ interest high and their curiosity piqued during their first-year coding courses.

Two surveys were conducted with the students in the introductory programming course. The first asked questions of the students’ expectations and levels of excitement before they partook in the game programming portion of the course (semester 2). The second was given after the students had concluded the gaming portion of their second semester programming course and presented their team semester project—a completed gaming application.

II. PREPARATION

Within first-year engineering students’ first two semesters, the students at Michigan Tech University are expected to take

consecutive first and second semester Analysis and Design courses that focus heavily on programming. The first and second semester classes emphasize team-based programming assignments, getting the students used to working with each other to facilitate problem solving. During the first semester the emphasis is on teaching the teams the basics of the MATLAB programming platform. Basic syntax, variables, vectors, algorithm construction and arrays are taught. Once this is learned, the teams are introduced to exercises involving more complex data processing and plotting using conditional statements, data cleaning techniques and loops.

In the second semester course, after learning the basics of programming in MATLAB during their first semester, the teams were introduced to MATLAB Application Designer. Within App Designer, the teams were introduced to game programming concepts such as local and global variables, different ways of assigning randomness to die rolls or card deals, how to assign images of die pips or cards to variables, how to create a discard pile wherein you can only deal from the remaining cards in a diminishing deck, how to create a hold card situation such as in blackjack, and how to use a matrix as a global property variable to simulate games like tic tac toe, connect four, or chess. The basic idea was to gradually step the students up by adding game design tools to their arsenal little by little. The students' second term final project was to use these components that they had learned to create a game within App Designer.

III. RELATED WORK

Abbasi et al. [2] found that the main outcomes incorporating game design into a programming course were the effective and cognitive outcomes such as evaluation and synthesis skills, and the willingness to learn.

Martin et al. [3] found that when implementing game design into a computer science course, 100% of the 30 students surveyed felt that the use of game design kept them more motivated, gave them more personal satisfaction, and taught them more relevant topics for future classes.

The study of Caulfield et al. [4] showed that, as a learning tool, games are becoming more common and students enjoy playing them and feel that they get some value from the experience. Despite sample size being small the authors conclude that enough evidence exists to say that educators could include game design in their courses as a useful and interesting supplement.

Games have been used as a means of learning programming for decades. This type learning has many monikers such as serious games, edutainment, gamification, game-based learning and playful learning [5]. With today's first year students spending a large amount of time per week online, the idea was to get the students excited about programming by taking a platform that they not only were familiar with, but also enjoyed. By integrating game-based programming into their assignments, it was anticipated that the students would exhibit much more enthusiasm and willingness to push themselves, and not become as easily distracted or fatigued with the coursework.

Feldman and Clua [6] found that for their freshman programming students, traditional algorithms, control structures, data types and related concepts were considered

abstract and difficult to explore, and that when issued by themselves, they were not considered stimulating to the students. With implementing game design, they introduced what students regard as their real world: games and game-based programming. They felt that the approach helped to promote an effective learning environment and encouraged class participation.

Another area that was studied was that of serious games, or games that were intended to aid in teaching programming. Miljanovich and Bradbury [7] studied various serious games that focused primarily on problem solving and fundamental programming concepts. Of the 49 serious games that were studied, half were available for free online. Games were assessed with respect to programming fundamentals specified in the ACM 2013 Computer Science Curricula guidelines. The study found that of the 23 students surveyed, the majority of these had positive feedback with respect to whether the games were accessible, that they felt engaged, and that they observed a learning effect from playing the game. Aside from surveys, other metrics used for evaluation were skill tests, game play statistics, and post play formal interviews.

Vahldick, Mendes, and Marcelino [8] also studied these serious games. They studied 40 games. They found that the most popular type of learning games was the action type where the user tries to program the movement of a robot or avatar. The second most popular type were adventure games where the player navigates a hero to explore an open world. The third most popular games were that of the general puzzle format. In Fig. 1, we see the amount of time our first-year engineering students at Michigan Tech University spend online participating in both social media and gaming. With this high percentage of students spending time online, would we not be missing out on a large opportunity to leverage something into their curriculum that they were already comfortable with and passionate about?

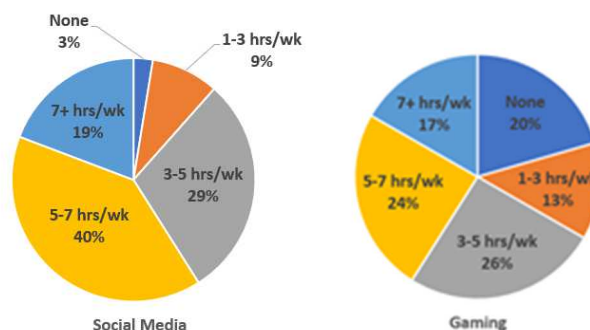


Fig. 1. Amount of Time Spent Online per Week, First-Year Engineering Students S=78.

IV. GOALS

The ultimate hope of utilizing a game-based programming in a first-year introductory programming class is to create an experience for the student that keeps them excited about their learning, challenges them, and creates a sense of pride and accomplishment in their work. The hopes are that this approach might stave off the intimidation, boredom, and burnout that can many times be found in a straight programming class. We can

see in Figure 1 that 67% of the students reported that they were gaming between 3 to 7+ hours a week. With the vast majority of students reporting that they do significant gaming during the week, the hope would be that they would become much more interested and engaged in signing up for a class that used gaming as a platform to learn introductory programming.

V. CURRICULUM/SYLLABUS

Students entered the class having spent one semester learning the basics of MATLAB. During this introductory semester, topics related to variables, vectors, arrays, conditionals, for loops, and while loops were taught.

In the beginning of the following game programming semester, the approach was to show the students how some of these basic concepts could be applied to game programming, and then let teams run with their knowledge and program a game of their choice. Teams' game programming was done using MATLAB App Designer. This allowed for a much easier way to create a game interface utilizing the feature of switching back and forth easily between a design "tabletop" view, and a code view. First, teams were asked to store vector images of various faces of gaming dice, and playing cards. Students were then taught to create random dice rolls (e.g. Yahtzee), or the random deal of a hand of cards with the die faces or card images appropriately displayed. Following this, teams were asked to create two functions: a cumulative score tracker function, and a function that would be used to track the end of a turn or hand. At the end of a turn, the additive scores of the dice or cards were tallied, and the next turn or hand was started. The concept of keeping track of new die rolls and hands via a vector was then added. Small individual concept tools were gradually introduced into the programming, like the use of a hold button under each die or card that could hold your present die or card values when the next hand or roll was executed (e.g. poker). Once the students were used to using vectors as a storage device for hands or rolls, arrays were added as a way to move towards simulating games such as chess, battleship, tic tac toe, or connect four. With these aforementioned programming techniques, the teams were able to then choose or create a game of their own.

VI. METHODS

At the beginning of the second semester game-based app design course, students were surveyed with respect to what their preconceived notions were before taking a programming course that had game-based programming for assignments and the term project. They were then asked to what extent knowing that game design was a large part of the class played in their excitement level and want to sign up for the class. The students were also asked whether or not they considered a first-year programming class that used game design was a valuable tool for learning application design and programming. Lastly, the students were asked whether or not they believed that they would have more pride in turning in a programming assignment that had to do with a game that they designed.

The second survey was given at the end of the 2nd semester following the teams turning in their end of the semester project. The students were asked questions pertaining to their experience in the class and whether or not they considered the programming of games to be a good way to learn programming. As well they

were asked about their enthusiasm towards including more programming courses in their engineering career.

VII. RESULTS

The ultimate hope of utilizing game-based programming in a first-year introductory programming class was to create an experience for the student that keeps them excited about their learning as well as challenging them. When asked if the students would be more inclined to taking a programming class that advertised game design as part of its curriculum, 59% of students either agreed or strongly agreed with this question.

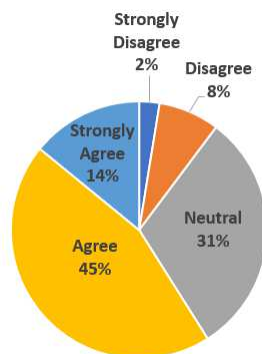


Fig. 2. I would be more inclined to take a programming class that advertised game design, First-Year Engineering Students. S=78.

When asked if the students thought that game design would be a valuable tool for learning programming preceding the class, 66% of the students either agreed or strongly agreed that it would be.

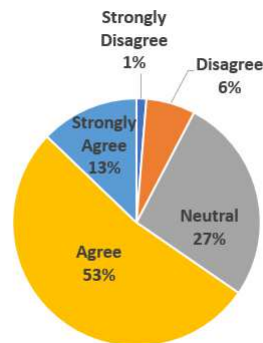


Fig. 3. I feel that game design would be a valuable tool for learning programming S=78.

It is important to state that these survey questions were posed at a point when the students had taken one semester of programming learning the basic coding fundamentals of the MATLAB platform. The students were then taught incremental portions of game design in the second semester utilizing MATLAB Application Designer, having the culmination of the semester end by submitting a full game application of their choice programmed by their 3-4 person teams.

After the second semester final projects had been turned in, the students were given a survey to find out what their opinions were on the class as a whole.

When asked about excitement levels that the students had in knowing that the second semester would contain game design in programming, we see that 39% of the students either agreed or strongly agreed with this. The neutral category made up for the largest percentage of students with a 35% constituency.

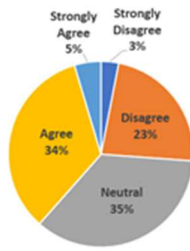


Fig. 4. My excitement level was high coming into this class knowing that game design would be a large part of the programming S=65.

When asked, however, if the game design component was their favorite part of both semesters of the programming course, a total of 51% of the students either agreed or strongly agreed, with a strong majority of students (71% either agreeing or strongly agreeing) stating that working with game design had a large part to play in them staying engaged in the 2nd semester programming class implementing game design.

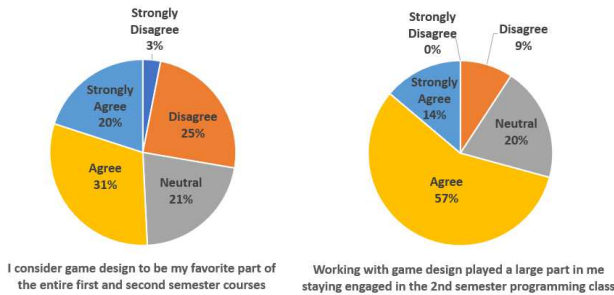


Fig. 5. Students overall rating of both semesters and rating of engagement in game based 2nd semester programming course S=65.

When asked if the game design aspect of the programming sequence made students want to work more with future programming in their engineering careers, 45% of students either agreed or strongly agreed. Lastly, the students were asked whether or not they felt that game design in MATLAB App Designer was a good way to further their programming skills. 81% either agreed or strongly agreed.

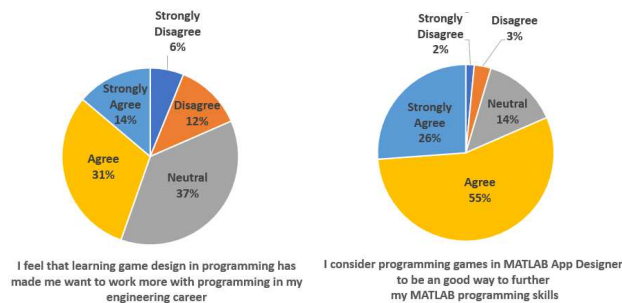


Fig. 6. Students' final assessment of motivation to take further programming based on the game-based course and whether or not they thought that game design was a valuable tool to learn programming S=65.

VIII. DISCUSSION

The purpose of this study was to investigate several things regarding the implementation of game design into a first-year engineering programming class. The class was a two semester contiguous series with the first half getting students acquainted with MATLAB and the second semester getting them used to using MATLAB Application Designer and introducing them to some basic game programming strategies. The thought was that with students being so acquainted with the online environment, it would not only get the students interested in signing up for a course with game design as part of the second semester class, but would also increase the students' engagement and learning in the class. For purposes of this paper, we will consider the term "agreed" to include the survey categories "agreed" or "strongly agreed", and "disagreed" to be inclusive of both "disagree" or "strongly disagree" category ratings.

Looking at Figure 2, which was given at the beginning of the 2nd semester class with game design programming, we see 59% of the students (S=78) agreed that they would be more inclined to take a first-year programming class if it advertised game design as a major constituent of the course. When the other categories were included in this question, there were only 10% of the students that disagreed with this survey question. Prior to taking the second semester, 66% of students also had "agree" or "strongly agree" rating stating agreed that game design would be a valuable tool for programming. These data would lead us to believe that if advertised properly, game-based programming would be a course that students would be more inclined to register for.

After taking the second semester game-based programming class, there were 25% or students (S=65) who disagreed with the statement that game design was their favorite part of the entire two-semester programming class sequence. With this statistic, however, 71% of the students agreed that game design played a large part in helping them stay engaged in the class.

Lastly, 45% of students surveyed in after the second semester course felt that game design in programming made them want to work more with programming in the future in their engineering careers. 81% of all students surveyed did, however, agree that they considered learning to program games in MATLAB Application Designer to be a good way further their MATLAB programming skills.

For further research in this area, it would be advantageous to have a larger sample size with future classes include in this study.

REFERENCES

- [1] Nelson, B, Dunn, R. Griggs, S. A., Primavera, L., Fitzpatrick, M. et al, "Effects of Learning Style Intervention on College Students' Retention and Achievement", Journal of College Student Development, Vol. 34, September 1993. Pp364-369.
- [2] Abbasi, Suhni & Kazi, Hameedullah & Khowaja, Kamran. (2017). A systematic Review of Learning Object Oriented Programming through Serious Games and Programming Approaches. 10: 2017.
- [3] Martins, Valéria & Souza-Concilio, Ilana & Guimaraes, Marcelo. (2018). Problem based learning associated to the development of games for programming teaching. Computer Applications in Engineering Education.2018.

- [4] C. Caulfield, J. C. Xia, D. Veal and S. P. Maj, "A systematic survey of games used for software engineering education," in *Modern Applied Science*, 5(6), 28, 2011.
- [5] Lindberg, R.S.N., Laine, T.H. and Haaranen, L. (2019), Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games. *Br J Educ Technol*, 50: 1979-1995.
- [6] M. Feldgen and O. Clua, "Games as a motivation for freshman students learn programming," 34th Annual *Frontiers in Education*, 2004. FIE 2004., 2004, pp. S1H/11-S1H/16 Vol. 3, doi: 10.1109/FIE.2004.1408712.
- [7] Miljanovic, Michael & Bradbury, Jeremy. (2018). A Review of Serious Games for Programming: 4th Joint International Conference, JCSG 2018, Darmstadt, Germany, November 7-8, 2018, Proceedings. 10.1007/978-3-030-02762-9_21.
- [8] Vahldick, Adilson & Mendes, Antonio & Marcelino, Maria. (2014). A Review of Games Designed to Improve Introductory Computer Programming Competencies. Proceedings - *Frontiers in Education Conference*, FIE. 2015. 10.1109/FIE.2014.7044114. G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955.