Paper ID #19418

# Examining Software Design Projects in a First-Year Engineering Course: How Assigning an Open-Ended Game Project Impacts Student Experience

**Dr. Krista M Kecskemety, Ohio State University**

Krista Kecskemety is a Senior Lecturer in the Department of Engineering Education at The Ohio State University. Krista received her B.S. in Aerospace Engineering at The Ohio State University in 2006 and received her M.S. from Ohio State in 2007. In 2012, Krista completed her Ph.D. in Aerospace Engineering at Ohio State. Her engineering education research interests include investigating first-year engineering student experiences, faculty experiences, and the connection between the two.

**Mr. Allen Benjamin Drown, Ohio State University**

Allen is a third year Industrial and Systems Engineering Undergraduate student at The Ohio State University who is an Undergraduate Teaching Assistant for the Fundamentals of Engineering for Honors (FEH) Program. His interests include Engineering Education, Lean Manufacturing, and Humanitarian Engineering. He will graduate with his B.S.I.S.E in May, 2018.

**Lauren Corrigan, Ohio State University**

Lauren Corrigan is a lecturer in the Department of Engineering Education at The Ohio State University. She earned both her Bachelor's and Master's in Civil and Environmental Engineering from Ohio State. She has two years of industry experience as an environmental engineering consultant. Her responsibilities included solid waste design, construction quality assurance, and computer aided design in support of various environmental projects. Lauren currently engages in teaching and curriculum development within the First-Year Engineering Program. Her research interests include the retention and success of students in STEM fields, with a particular focus on under-represented populations.

# Examining Software Design Projects in a First-Year Engineering Course: How Assigning an Open-Ended Game Project Impacts Student Experience

**Abstract**

Courses that teach programming often include large software design projects intended to synthesize the elements learned in the class. A software design project has been used at a large public university in the first-year engineering honors course to practice using programming elements and prepare students for the second semester course. Prior to Autumn 2014 all sections of the course created a program to detect and compute the frequency of an infrared (IR) signal. In Autumn 2015, six sections of the course changed the design project to be designing and programming a game, four sections assigned the traditional IR project, and two sections gave students the choice. Based on survey results, students who completed the game project indicated greater enjoyment, greater sense of creativity, greater teamwork skill development, greater preparation to their future as an engineering, and preparation for the spring semester project compared to those who completed the IR project.

## Introduction

The use of games as an instrument of educational instruction is not new. Games have often been used throughout history as a tool to teach students important skills such as reading, logic, and mathematics; therefore, it logically follows that computer games can be utilized as an effective tool for teaching students programming skills[1]. There is evidence that programming computer games as a method to teach programming skills has an integral place within the curriculum. The use of game development in programming courses in both upper level courses[2] and in first-year engineering courses[3,4] has increased in popularity. Additionally, game development naturally lends itself to more open-ended problems where students have the freedom of choice. There has been research in the area of choice in student projects and its impact on student motivation and interest[5]. By incorporating student choice into a project, students are more intrinsically motivated to succeed. Student interest and motivation varies from student to student, thus providing students with the opportunity to choose their own project allows the instructor to accommodate the varying interest across the student population[4].

Universities around the country have been developing open-ended projects for their classes that allow their students to use their creativity. At one large public university, students were given the task to control a robot arm using MATLAB to play tic-tac-toe. Students felt motivated to work on and perform well in the project as it was deemed “fun” [6]. At a small private university, a professor used game design to teach students different algorithms[1]. Students felt that having a visual representation of their program when completing the algorithms helped them understand the logic of both the algorithms and of coding. Students found that these projects allowed them to make use of the concepts learned in class. Having students use what they have learned in class in a creative software design project, specifically game development, allows them to achieve higher levels of Bloom’s taxonomy[7].

At The Ohio State University, the first-year engineering honors course concludes the first semester with a week-long software design project. Prior to the start of the project, the curriculum focused on problem solving using computer tools such as Excel, MATLAB, and C/C++ programming. For many years, the software design project was to develop a code to detect an IR signal and compute its frequency. The IR project provided students with real time programming and gave the students experience working with the controller that they would use in the second semester. For most of these students, the second semester course involves designing, building and programming an autonomous robot.

Recently, some faculty have piloted a computer game software design project. This project has allowed students the freedom to propose any game of their choosing, work with the instructor to decide the difficulty of the project, and then most use the controller, the same one used in the autonomous robot design project during the second semester, to implement and create the game. This study seeks to address the following research questions: *In what ways do software design projects prepare students for semester-long design projects?* and *What are students perceptions and engagement in these projects?*

## Methods

### *First-Year Program Description*

The first-year engineering program has two major tracks for students: honors and standard. The pilot project was conducted in the first-year engineering honors track. The first-year engineering honors program is for university honors designated students and consists of a mandatory 2-semester sequence. In both the honors and standard tracks, the first semester focuses on problem solving using computational tools and computer programming and the second semester focuses on engineering design and graphics.. In the honors program, the first semester includes instruction in MATLAB, C/C++, and Excel.

For the second semester, students have the option to complete a design and build project of an autonomous robot, a theoretical design, build and research nanotechnology project, or an integrated business and engineering project. While intended major sometimes drives the choice between these projects, students are ultimately able to choose the project that interests them the most. Typically, between 70-80% of the students choose to complete the robot project, 10-20% choose to complete the nanotechnology project, and 5-10% choose the business project.

### *Software Design Project Description*

At the end of the first semester C/C++ unit, the students engage in a weeklong design project to test the programming skills in real-time conditions. Prior to Autumn 2014, the sole software design project (SDP) was to have students use C/C++ code to detect the frequency of an IR Beacon. The IR beacon randomly transmits frequencies between 20 Hz and 150Hz. The students use the robotics controller and an IR receiver to detect the transmitted frequency and print the frequency to the screen of the in-house developed controller, the Proteus[8].

During the Autumn 2014 semester, the honors program piloted the game software design project. The project has less standard requirements as the students choose what their product will be. The primary requirement for the game project is to create a game with a graphical display either using the Proteus controller and C/C++, MATLAB, or Linux and C/C++. Other requirements include the use of different loop structures, logical and relational structures, and functions. Students must also create a title screen that lets them choose between playing the game, seeing the instructions, quitting the game, or viewing the stats tracked by the game.

Using the baseline requirements for the game project, each team of two students drafts a project proposal that will detail the requirements of the project. The students specify which base features of their game which comprise a majority of the credit, and additional features that allow a team obtain the maximum points. The instructor can approve or ask for changes to the game based on quality of the features and the difficulty of the game. Well-known games that are popular for students to use as inspiration for their game design include Tic-Tac-Toe, Flappy Bird, Pokémon/Street Fighter Battle Simulators, and Snake. Students are also allowed to create a new game concept for the project.

It is important to note that the final programmed product is only worth 10 out of the 55 points associated with the project. The majority of the project points are from the project documentation. The initial pilot was expanded from two sections in Autumn 2014 to six in Autumn 2015.  Most sections also required the students to pitch their game to the class through an in-person demonstration of their game or in the form of a video. It is during this Autumn 2015 implementation that the students were surveyed to assess the projects.

***Software Design Project Learning Objectives***

The learning objectives of the software design project are given below.

Students will be able to:

- program in a real-time environment
- respond to an external physical signal or input
- synthesize programming elements learned throughout the course into a cohesive program
- work effectively on a team to solve a problem

The way these learning objectives are achieved vary between the two projects (game vs. IR), as the implementations are different. Both projects deal with programming utilizing physical input, coding in real time, and modular programming. The IR project focuses heavily on data acquisition, cleansing, and analysis from a physical input. Students can achieve these objectives by learning how to create a system that can know which acquired data should be processed, and which data should be discarded. They can complete this task with programming fundamentals learned throughout the course.

The game project has physical inputs mostly from a button board, accelerometer and the touchscreen on the controller.  Sometimes students use additional switches or LEDs depending on the game.  The focus of the game project is primarily real-time and modular programming. Similar to the IR project, students can complete the game project with programming fundamentals learned from the course.  In both projects, students often learn new ways to use these concepts in order to develop an effective program.

Both projects require students to further develop skills such as teamwork, time management, and project planning. As students work in teams of two, they learn how to divide programming between the team members, and to do so within time constraints. As part of the required documentation, teams create algorithms and flow charts to plan the code for their project.

## *Theoretical Framework and Survey Description*

The theoretical framework for this study is based on Engagement Theory[9]. In Engagement Theory, students are to work on collaborative teams to complete meaningful, authentic project. The main aspects of Engagement Theory, relate, create, and donate are shown in Figure 1.

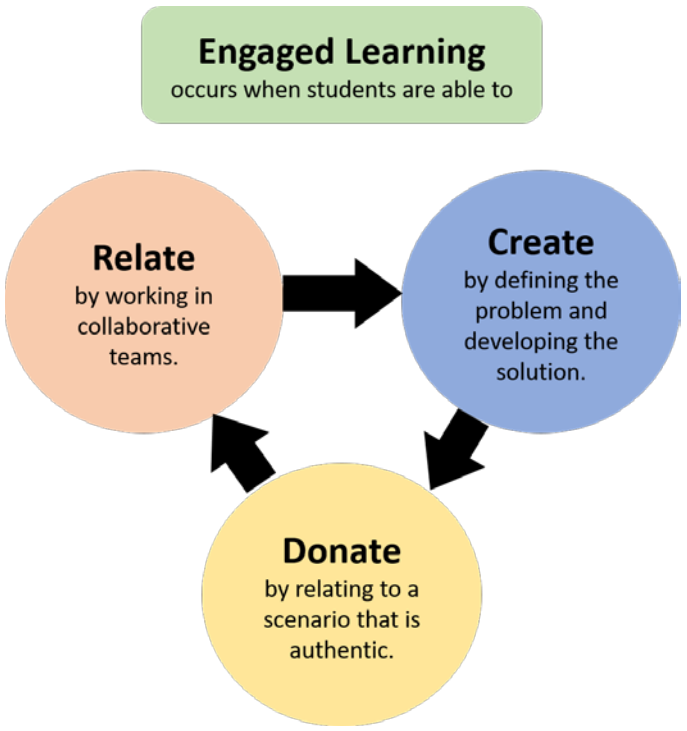
Engaged Learning
occurs when students are able to
Relate
by working in collaborative teams.
Create
by defining the problem and developing the solution.
Donate
by relating to a scenario that is authentic.

**Figure 1: Engagement Theory [9], adapted from [4]**

To examine the impact of the software design project, a survey was used to ask the students 8 Likert scale questions and an open-ended question about their experience with the project. These Likert scale questions are a subset of those used in Ref [4] which was used as the foundation for this work. The Likert scale questions were designed to study aspects of the "relate-create-donate" model of Engagement Theory. The Likert scale questions are listed below:

1. I enjoyed the SDP.
2. The SDP enhanced my understanding of the design process.
3. The SDP operated under a manageable timeframe.
4. The SDP was a good example of applied engineering.
5. The SDP helped develop my engineering teamwork skills.
6. The SDP helped develop my creativity in engineering.
7. The SDP helped prepare me for my future as an engineer.
8. The SDP helped prepare me for my design-build project in [second semester course].

The survey concluded with open-ended questions; *please provide any additional comments you have on SDP. What was your favorite part of the SDP? Any changes you would make to the SDP?*

Since one of the goals of this work was to look at how students felt the project prepared them for the larger second semester project, the survey was given to students at the end of the second semester. During Autumn 2015, six sections of 36 students assigned the game project, four sections assigned the IR project and two sections gave students the choice.  In the following spring survey, 358 students participated: 229 students who completed the game project and 129 who completed the IR project.

## Results and Discussion

### *Likert Scale Question Analysis*

The Likert scale questions were converted to a 1 to 5 scale where 1 corresponded to strongly disagree and 5 corresponds to strongly agree. The average and standard deviation of each question was calculated. The results of the survey are shown below in Table 1.

**Table 1: Likert Scale Questions results.**
**Statistically significant values (p<0.05) designated with a *.**

| Question | | IR N=129 | | Game N=229 | | Difference (Game-IR) |
|---|---|---|---|---|---|---|
| | | Average | Std | Average | Std | Average |
| 1 | I enjoyed the SDP | 3.7 | 0.92 | 4.32 | 0.79 | 0.62 * |
| 2 | The SDP enhanced my understanding of the design process | 3.88 | 0.96 | 4.12 | 0.77 | 0.24 |
| 3 | The SDP operated under a manageable timeframe | 4.15 | 0.82 | 3.93 | 1.03 | -0.21 |
| 4 | The SDP was a good example of applied engineering | 4.09 | 0.8 | 4.23 | 0.72 | 0.14 |
| 5 | The SDP helped develop my engineering teamwork skills | 3.9 | 0.96 | 4.16 | 0.87 | 0.26 * |
| 6 | The SDP helped develop my creativity in engineering | 3.57 | 1.12 | 4.36 | 0.71 | 0.79 * |
| 7 | The SDP helped prepare me for my future as an engineer | 3.64 | 1.02 | 3.98 | 0.91 | 0.34 * |
| 8 | The SDP helped prepare me for my design-build project in [second semester course] | 3.47 | 1.21 | 3.93 | 1.01 | 0.46 * |

Questions 1, 5, 6, 7, and 8 were found to be statistically significant ($p<0.05$) using a Mann Whitney U test.  The students rated each of the statistically significant questions higher for the game project. Figures 2-6 show the distribution of the Likert Responses for the questions that were statistically significant.  Question 6, the question pertaining to creativity of the project, resulted in the largest difference in both means and standard deviations, as students felt that the game project allowed more creative freedom. Students found the game project to be more enjoyable than the IR project, as shown by Question 1.

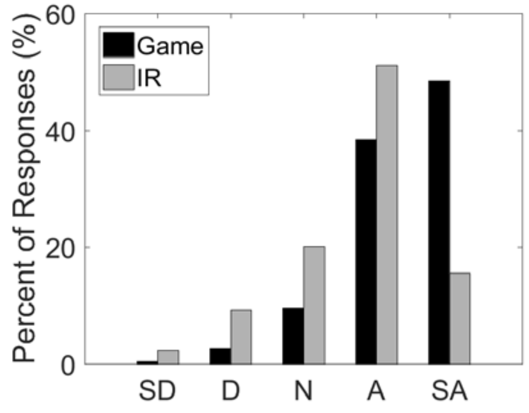

Figure 2: I enjoyed the SDP

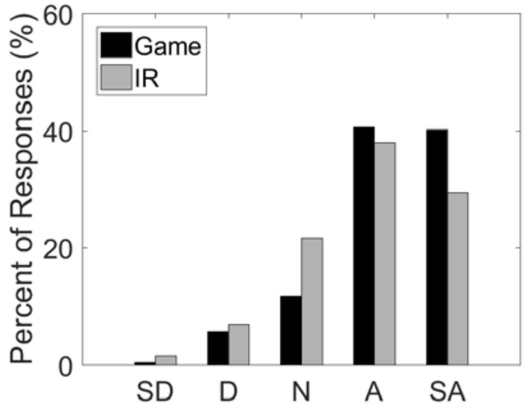

Figure 3: The SDP helped develop my engineering teamwork skills

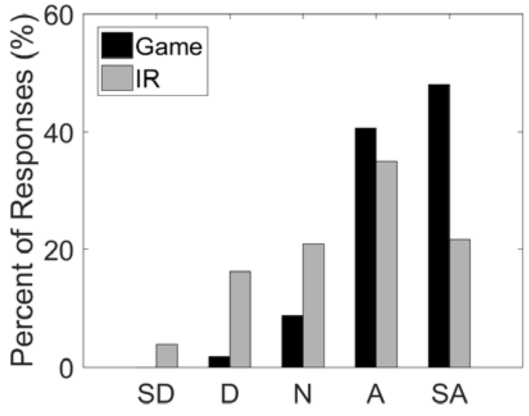

Figure 4: The SDP helped develop my creativity in engineering.

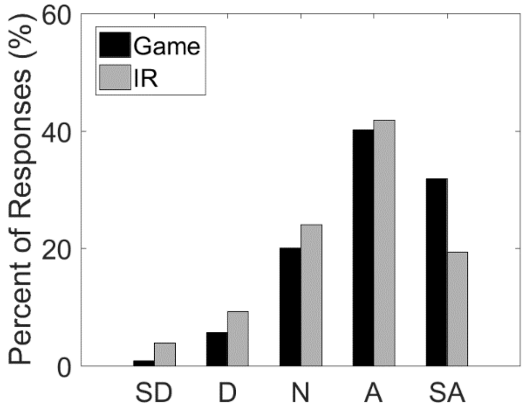

Figure 5: The SDP helped prepare me for my future as an engineer.

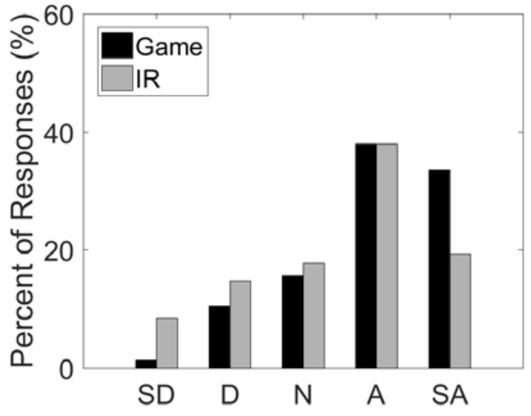

Figure 6: The SDP helped prepare me for my design-build project in [second semester course].

## *Open-ended Question Trends*

The open-ended questions were analyzed for trends and coded into categories. The questions may have fit into multiple categories or into none, and the results are shown in Figure 7. The data was normalized by dividing the responses by the number who completed the survey for that population and scaling it to 100. Many of the categories aligned with the Likert scale questions. One category that was not included in the Likert scale responses was students who wished they would have done the other software design project. There were more comments about students who wished they could have completed the game project than there were students who wished they could have completed the IR project.

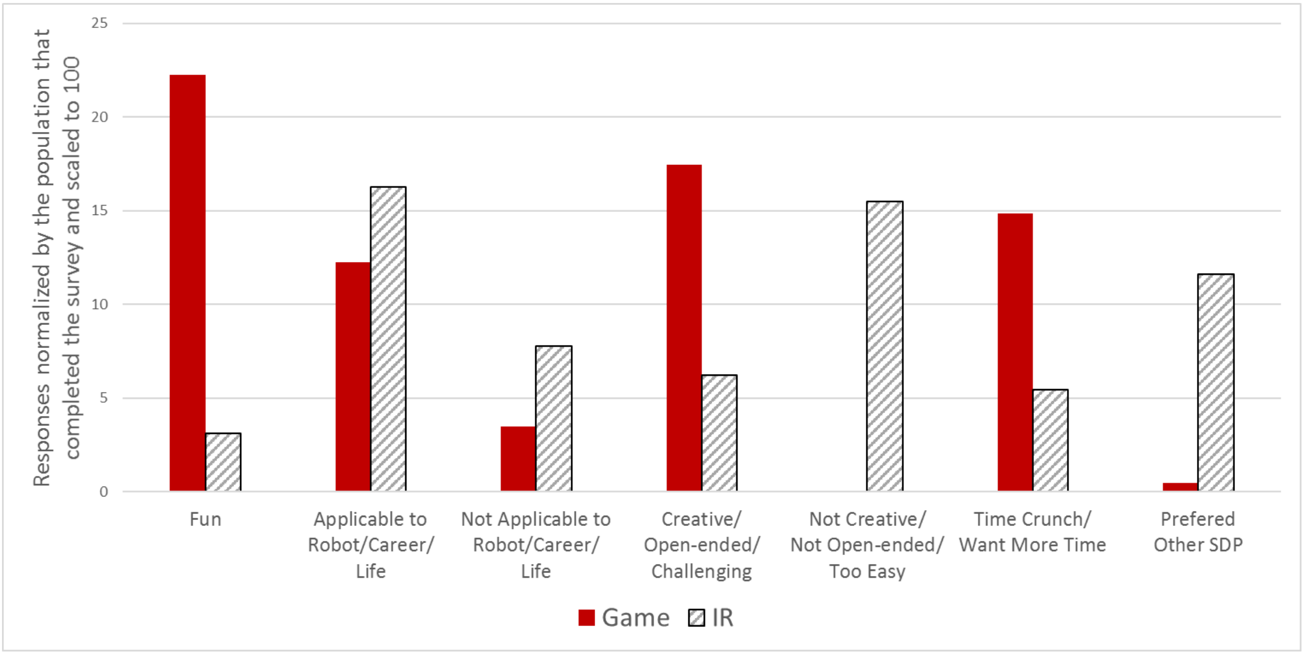

**Figure 7: Student Responses to Open Ended Questions**

Students viewed the IR project as a realistic, but constrained, experience that can be encountered in industry. The greatest number of positive comments focused on its applicability to the robot project, their career, or their life. Students also indicated they liked that the coding project used something "physical" and "practical" as opposed to the calculation heavy problems found in the programming assignments earlier in the semester. Additionally, many of these students mentioned that this was the first experience that they had to debug a code that received input from an outside system, without any set guidelines on how to complete the task. This experience is shown in a comment from a student who completed the IR project:

> *"I thought the SDP was the first engineering application with no direction. I really liked this aspect of it because the previous classes consisted of giving directions and completing an assignment. This was an open ended problem that required teamwork to solve. I would not make any changes for I think the time limit was good as well as the grading. I would do more projects like this for lab instead of following a lab procedure."*

As seen from Figure 7, students commented more frequently on how "fun" the game project was compared to students who completed the IR project. This is supported by the Likert scale analysis as shown by Figure 2. Students found that the game project allowed more creativity, also seen in the Likert response analysis. Students felt immense pride with what they done with they created as one game project student said:

> *"As a millennial child, video games carry a special place in my heart, so this project was a great way to allow for some creativity to flow and some more advanced Proteus coding to be built. It's fun because it's easy to make a simple game to get a good grade, but everyone felt obligated to make as cool of a game as they could possibly manage."*

These same sentiments were not found as often in the responses of the students who completed the IR project as those topics tended to elicit negative comments. This can be seen in the following comment from a student who completed the IR project:

> *"The IR SDP didn't involve much creativity; there was essentially one or two correct answers, which wasn't at all the case for the robot project and isn't true in industry."*

A significant portion of the game project students mentioned in the open-ended response that they viewed the project to be fun even though a significant portion of those students also said that the project caused a lot of stress as they were under a time crunch and had limited time for testing. Many students mentioned that the reason they felt rushed in the project was that they did not feel prepared for the debugging portion of the project. Some student felt that they did not have the proper experience to learn debugging from the assignments beforehand. As one game project student said:

> *"The fact that I was able to participate in the game SDP was incredible, and I cannot express how much I enjoyed that project. However, my only complaint regarding this project would be the timeframe. It was clear that everyone participating had a clear understanding of what they wanted to do, and everyone was able to prototype and near finish their ideas, but most people ran out of time when they reached the troubleshooting phase. Since I believe that this is one of the most important phases in the programming process, I was disappointed to see that adequate time wasn't given for this phase."*

Many students took these projects as learning experiences in time management, creative, thinking, and problem solving. Having these lessons before the second semester project greatly affected student's attitudes during the semester long project. One IR project student who went on complete the robot project explained:

> *"My favorite part of the SDP was how it was tricky at first, but it was not impossible. It showed me that hard work and creative thinking can be used to solve some really complex engineering problems. It also introduced me to problem solving through the QT creator. I would somehow require that each team member spend more time working with the QT creator. I believe this experience would increase confidence when doing the robot project."*

As shown from the previous quotes, students found creativity and practicality to be important factors when learning the engineering design process. Students want to be challenged in this regard as they feel that problems like these are the problems that they will encounter in their future classes, internships, and careers, and that they feel that these types of projects are the most enjoyable to design and solve for.

Many students in the IR project sections found that their project prepared them for dealing with signals and the noise encountered in the coding for a physical system. Many students in the game project sections found that their project helped them with creating and debugging the complex code. As the robot project is focused on learning how to design and debug code that obtains signals and noise from a physical system, both projects are very applicable to these areas. Student teams who master both of these concepts normally are able to complete the robot project effectively and more efficiently.

## Conclusions and Future Work

The results of this study show that students perceive more benefits from the game project compared to the traditional IR project. Students who completed the game noted that the project allowed for more creativity than the students who completed the IR project. Additionally, students who completed the game indicated that it prepared them more for the second semester project compared to the other project. From this result, it would be recommended to have a more open-ended, creative project to prepare students for the second semester design project. While student enjoyment is not the sole indicator of a good project, the ability to incorporate more student choice and cultivate a fun learning environment during the game project may result in increased intrinsic motivation, thus encouraging students to learn more and apply what they learned in new, creative ways. Anecdotally, the instructional staff observed that teams who found the game software project to be very enjoyable, more often than not, chose harder and more complicated game projects. More complicated game projects allowed these teams to be more creative, not only with the game but also with how they coded the project and how they chose to document the project.

The implementation of the game project served as a pilot in the honors sequence of the first-year engineering program. A full-scale implementation in honors may occur soon based on the results of this study. In addition to the game project piloted in the honors program, a similar game project has been developed and piloted with the standard first-year engineering sequence. This project grew out of the honors pilot and another known game project for first-year students[10]. While similar in its intent to the honors project, the standard pilot project required different criteria and scope due to the differing course content. One of the important keys to this standard pilot was the inclusion of a business and customer element that was missing from this initial honors pilot. In order to address the value of customer input, students were required to conduct interviews based on end-users and design and receive feedback from users.

Additionally, as part of their thinking of the end users, students had to create the pitch advertisement video for their game. These additional elements could be included in the honors implementation in subsequent years. The second semester projects for both the robot and nanotechnology tracks use an online portfolio[11] for documentation, so a simplified version of online portfolio has been introduced used in both the standard and honors sequences of the game software design project. Additional studies need to be done to look at the impact of all these elements.

One additional study that could occur includes the analysis of the code used to complete the two different projects. By analyzing the structure of the code, different levels of coding elements used and differences in complexities between the two projects may be discovered. This may help demonstrate if one project lends itself to more practice and experience with specific programing fundamentals. This would help inform future project decisions.

## References

1. Baibak, T, and Agrawal, R., "Programming Games To Learn Algorithms", *Proceedings of the 2007 American Society of Engineering Education Annual Conference,* Honolulu, HI, June 2007.
2. Maxim, B., "Serious Games as Software Engineering Capstone Projects," *Proceedings of the 2008 American Society of Engineering Education Annual Conference,* Pittsburgh, Pennsylvania, June 2008.
3. Estell, J.K., "Writing Card Games: An Early Excursion into Software Engineering Principles", *Proceedings of the 2005 American Society of Engineering Education Annual Conference,* Portland, Oregon, June 2005.
4. Helber, E., Brockman, M., and Kajfez, R., "Gaming with LabVIEW: An Attempt at a Novel Software Design Project for First-Year Engineers", *First Year Engineering Experience Conference,* August 7-8, College Station, TX, 2014.
5. Shepard, T., "Implementing First-Year Design Projects with the Power of Choice", *Proceedings of the 2013 American Society of Engineering Education Annual Conference*, Atlanta, Georgia, June 2013.
6. Hamrick, T.R., and Hensel, R.A., "Putting the Fun in Programming Fundamentals - Robots Make Programs Tangible", *Proceedings of the 2013 American Society of Engineering Education Annual Conference*, Atlanta, Georgia, June 2013.
7. L.W. Anderson, et al., *A Taxonomy for Learning, Teaching, and Assessing.* Addison Wesley Longman, Inc., Illinois, 2001.
8. Vernier, M. A., & Wensing, P. M., & Morin, C. E., & Phillips, A., & Rice, B., & Wegman, K. R., & Hartle, C., & Clingan, P. A., & Kecskemety, K. M., & Freuler, R. J., "Design of a Full-Featured Robot Controller for Use in a First-Year Robotics Design Project," *Computers in Education Journal*, vol. 6, no. 1, pp. 55–72, January-March 2015.
9. Kearsley, G. & Shneiderman, B., "Engagement Theory: A Framework for Technology-Based Teaching and Learning", *Educational Technology*, Vol. 38, No. 5, September 1998, pp. 20-23.
10. Ossman, K., and Bucks, G., "First Year Student Team Projects Using MATLAB", *First Year Engineering Experience Conference,* August 8-9, Pittsburgh, PA, 2013.
11. Kajfez, R. L., & Kecskemety, K. M., & Kross, M., "Electronic Notebooks to Document the Engineering Design Process: From Platform to Impact Paper," *Computers in Education Journal*, vol. 7, no. 1, pp. 37–46, January-March 2016.