# An Experimental Elective Course in Video Game Design for Mechanical Engineers

Joseph M. Mahoney, *Penn State Berks*

***Abstract*—Many engineering students are not motivated to learn or apply computer programming in their courses. Partially, this is due to computer science topics being pushed upon them rather than students learning them as needed. A senior-level video game design class was offered as a technical elective. This class combined a "humanities" viewpoint of video game design (e.g., gaming psychology and theory of fun) with the "technical" side of computer programming. Students compared and contrasted two games and wrote a critical analysis of another game. Most of the course was spent conceptualizing, planning and creating a video game. Groups learned the required programming skills as needed to implement their vision. Students completed a survey about their experience in the course. Students found the course exercised their creative skills, motivated them to learn more programming and provided them with experience in project management.**

***Index Terms*—project based learning, computer programming, creativity, group work**

## I. Introduction & Motivation

Extensive research has been done showing the efficacy of using games, including video games, as instructional resources in the classroom at the K-12 and university levels to improve engagement, motivation, and outcomes [1, 2]. Games have been shown to be valuable for teaching technical and non-technical subjects. A limited selection of examples includes Mathematics, Reading and Spelling [3], Biology [4] and Social Studies [5].

Some research has been performed in utilizing game *creation* as a pedagogical method for computer programming. Games have been employed as a motivational tool for students to learn the course material [6–9]. For first-year computer science (CS) students, some implementations include creating Minesweeper, Asteroids [6] and dice [7] games. According to survey data, students responded positively to having a game-based instruction of programming. These programs have been run as first courses for CS majors that presumably already have an interest in programming.

Game-based curricula have also been implemented for non-CS majors. For non-technical and younger students, Scratch has been used as an introductory programming system. Scratch has attractive features, compared to more "traditional" languages: it is web-based, graphics-based and has a vibrant social component of sharing and remixing other's code [10]. This simplified environment can impart the basic ideas of algorithms, loops and conditional statements to students that have never programmed [11–13].

J. M. Mahoney is with Mechanical Engineering, The Pennsylvania State University, Berks College, Reading, PA, USA (e-mail: joseph.m.mahoney@gmail.com), (web: https://sites.psu.edu/mahoneygroup/).

The use of video game creation as an instructional tool for non-CS engineers is less explored. These students need to use programming as part of their work. They, more than CS majors, often suffer a lack of motivation for learning programming. Students are traditionally presented with a programming method they do not understand to solve a problem that they do not care about in their introduction to programming course [8, 14]. In previous research work, student surveys rated their programming course as the least important in their curriculum. One introductory engineering course gave students a video game shell for a racing game that they had to complete [15]. Here, they applied different numerical methods as needed to get the game to function. The students were found to demonstrate a deeper understanding of the material compared to students in the traditional version of the course.

This previous work in video game creation as a motivating tool for students to learn to program inspired the formation of a new senior-elective course entitled, "Video Game Design and Development" in spring 2017. This course was offered to Mechanical Engineering (ME) and Information Sciences and Technology (IST) majors. All students had at least one formal computer science course beforehand (typically Matlab for ME and Java or Python for IST). This paper details the format, content, outcomes, and recommendations for the course. This paper is an expansion on results and discussion first presented at the 2017 ASEE Mid-Atlantic regional conference [16].

## II. Course Description

### A. Philosophy & Format

The course was designed to approach video game design from both a "technical" view (e.g., coding, scripting, graphics) and "psychological" perspective (e.g., user experience, player types). The course objectives were:

1) Articulate what makes a game "fun" and identify these elements in popular video games
2) Identify various game mechanics utilized in popular video games
3) Apply the game design process to create a playable video game prototype
4) Learn fundamental high-level programming concepts to create a playable video game prototype
5) Enhance presentation skills in conveying both technical and non-technical information
6) Prepare blog postings for a public-facing website

The recommended reference texts were Hiwiller's "Players Making Decisions" [17], Adams' "Fundamentals of Game

Design" [18] and Despain's "100 Principles of Game Design" [19]. These were found to have a good balance of the academic study of design and the practical implementation. Other reviewed texts focused on marketing and distribution aspects of the process which were not covered in this course.

The first half of the course focused on an academic assessment of video games, starting with addressing the question, "What makes a game fun?". Starting with theory and viewpoints such as Koster's *Theory of Fun* [20] and Lazzaro's *4 Keys* [21], students engaged in discussion and cited examples. In other upper-level engineering courses, discussion and difference of opinion do not occur. This course exposed students to ideas that did not come from first principals of physics but from observations. The course then covered topics including player experience, game mechanics, ideation, prototyping, and playtesting.

### B. Guest Presenters

Two guest speakers gave presentations, held discussions and answered questions with the class via video conference. The first presenter was on the *Titanfall* team at Respawn entertainment. He provided insight into their studio's design process and skills they look for in employees. He connected the theoretical process into practical implementation.

The second presenter was a writer, director and game maker who wrote the story for a major-release mobile game. He discussed the interface between the programming and "creative" side of the industry. He offered an artistic perspective that engineers are not often exposed to.

### C. Assignments

The instructor presented interactive lectures during the first three weeks of the course. Then, the presentation of course content was handed over to the students. Groups of four students were responsible for delivering a lecture, moderating a discussion and running activities for a week (2.5 hours) of class. They could utilize the recommended textbook and online resources for their presentations (Objective 5).

Students wrote two compositions for the course blog. During the first third of the course, they compared and contrasted a game made before 2002 with its sequel or spiritual successor made after 2012. They were to assess the technical differences such as graphics, sound and user interface. Moreover, they were to observe more subtle design elements as storytelling. For example, the original 1987 (US) *Legend of Zelda* game told the story mainly through the instruction book and little in the game itself. Whereas, 30 years later, the 2017 *Breath of the Wild* is highly story-driven through character dialogue, full motion video, and in-game artifacts.

In the middle third of the semester, students wrote a critical analysis of a recent AAA game. This assignment focused on applying the theory component of the course upon real-world games (Objectives 1 and 2). For example, assessing how different play styles are catered to and how the flow of the game progresses between "easy fun" and "hard fun."

During the second half of the course, students worked primarily on the overarching course project. The project was to conceive, design, and create a working video game (Objectives 3 and 4). The project had milestones throughout the semester of (1) an initial pitch, (2) midterm update, (3) bug tracking report, (4) final demonstration and (5) final report. Groups were free to choose to make anything from a graphics-free text adventure to a graphics-intensive 3D first-person shooter. No restrictions were placed on the language, engine or platform students could use for their game. The desire for the course was to not focus on the graphics but rather the gameplay and the underlying programming to support it. This, unfortunately, is not the direction the teams chose as discussed in Sec. III-A3.

Students contributed their work to the broader internet community (Objective 6) for reference, criticism, and reuse. Students agreed at the beginning of the course that all work would be submitted under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To that end, all work is publicly available. All work was peer-reviewed by classmates before the final version was posted to the course website and graded by the instructor.

## III. Course Outcomes

### A. Student Work

All student blog posts, project updates and final presentations for the course can be viewed at https://sites.psu.edu/ist446bk/. Downloadable playable final projects are archived in a repository at https://osf.io/uxfr5/ [22].

*1) Student Lectures:* Four student groups delivered one-week lectures on (1) core game mechanics, decision making, and randomness; (2) rules, play balancing, and feedback loops; (3) game theory and design methods; and (4) prototyping, play-testing, and bug tracking. Groups presented their topics using traditional slides, drawing information from the recommended textbooks . The theory was grounded with examples from games the group members were playing and familiar with. Additionally, interactive activities were used to engage the class with the topic.

The first group utilized a *Parsely* live-action text-adventure game to demonstrate that a fun and engaging game requires that player makes meaningful choices. In this small-scale role-playing game (RPG), the group is presented with a description of their dungeon setting (similar to *Zork*) by the "Dungeon Master" (DM). One at a time, players give verbal commands to the DM as one would type into a computer terminal. The group collects items and attempts to escape the dungeon. Choices can result in the death of the player (restarting the game) or victory. The DM must think like a computer in handling commands that are outside the "programming" of the game. For example, how does the DM or game respond when the player attempts to "draw sword" when there is no sword in their inventory? For most of the students, this was their first time playing an RPG that did not rely on graphics or guided interface.

The second group had the class play a simple in-class game as a group. In different iterations of play, the rules were modified to demonstrate the effect of balance on the outcome. In the first rounds, some players had abilities that

let them easily dominate others. In subsequent rounds, players were given different abilities, but now the game remained competitive. This exercise quickly showed the importance of rule balancing but also how playtesting finds flaws in the prototype that may not be apparent during the design phase.

The final group used a class day to run a game prototyping activity. Here, the class was split into groups, given a theoretical list of materials and asked to design a physical game in fifteen minutes. After each group described their rapid prototype, the lecturers then brought in a box for each group that contained all the materials on the list. The groups were given another half hour to build, test, and refine the game that they designed on paper. Finally, each group presented and played their finished game.

*2) Blog Responses:* Students were challenged to make their blog responses not only thorough but interesting for the reader. For the engineering students, this was a departure from the style of written work required in their core courses. In those courses, the technical writing style is expected and documents are formatted as lab or technical reports.

In the peer-review step, each student reviewed – and was reviewed by – five others. In reviewing the reviews, students generally offered constructive criticism to improve the post. Compared to showing a draft to only the instructor, there was additional accountability as peers were going to view and judge the student's effort.

The posts were generally of good quality. Some students were invested in the assignment and clearly enjoyed critically playing the games and reporting on their findings. Possibly due to the relatively low percentage of credit associated with the assignment, some students submitted a perfunctory product.

The course objective of student work contributing to the general community (Objective 6) was achieved. To date, student posts have several thousand views and often appear in Google search results.

*3) Course Project:* The class split into four groups for the course project with each group having at least one student from each of ME and IST. Groups gave brief presentations at the milestones of initial pitch, midterm update and final demonstration. These were opportunities to receive feedback from the other groups, share ideas and best practices. All four groups initially choose to create their games in a modern physics engine (one used Unreal Engine 4 (UE4), two used Unity, and one used GSC, the Call of Duty: Black Ops III Engine). One group switched from Unity to Visual Basic after finding the learning curve too shallow. All groups were able to produce a working interactive demonstration of their game by the end of the course.

One group made the 2D side-scrolling game, *Randy the Running Robot*, in Unity. The team created their own physics rules and scripting as part of the process. They used Creative Commons assets for their graphics and animations. This game matched the closest to the course vision as most of the group's time was spent programming the game and then play-testing and refining it. A screenshot of the game is shown in Fig. 1.

Another group used GSC to make the custom game, *Project Z*, for Call of Duty: Black Ops III. The programming environment they chose had many built-in features (animation, scripts, physics) allowing the group to focus on creating a custom map, weapons, and features for the game. A screenshot is shown in Fig. 2.

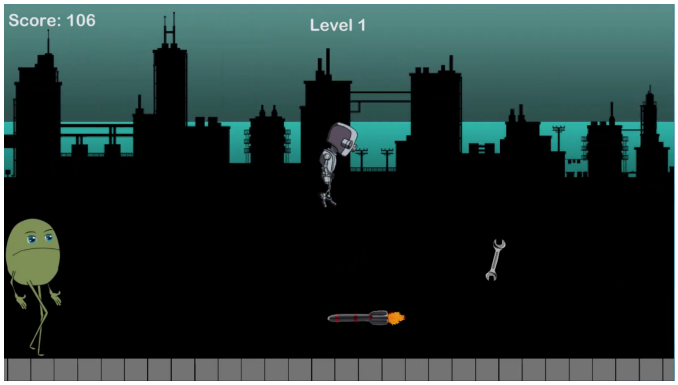


Fig. 1: Screenshot of *Randy the Running Robot*. Students used the Unity Engine with Creative Commons assets to create this side-scrolling game.


Fig. 2: Screenshot of *Project Z*. Students used GSC to create this custom game for Call of Duty: Black Ops III.

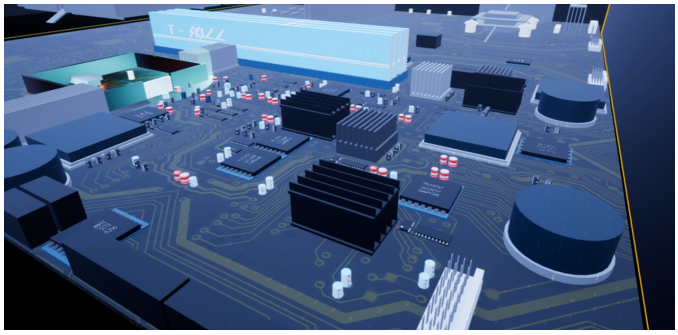

Fig. 3: Screenshot of *Tech Support*. Students used UE4 to create the game including the map, weapons and enemy AI. This image shows an overhead view of their map.

*Tech Support* was created using UE4. The team spent much of their time building and lighting the map, creating weapon systems and programming the enemy AI. A screenshot is shown in Fig. 3.

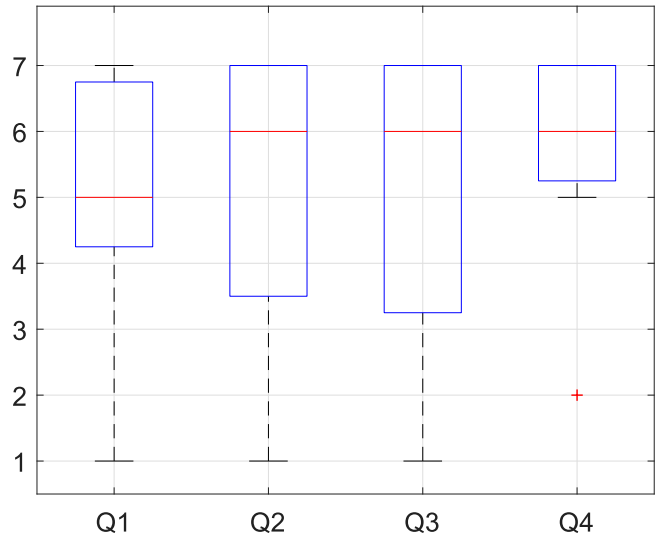


Fig. 4: Boxplot of responses concerning the first programming course from Table I using the scale in Table II. No prompt was found to be generally disagreeable ($p \gg 0.0125$ with a mean less than four. Outliers are marked as "+" but are included in all statistics.

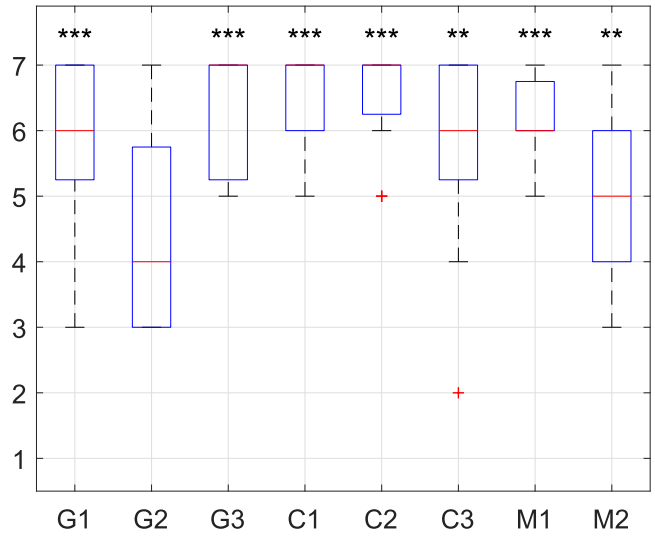


Fig. 5: Boxplot of Outcome Responses from Table I using the scale in Table II. All questions except G2 (group learning) had $p < 0.0063$ for the mean greater than four. One star indicates $p$-value of $p < 0.0063$, two stars indicate $p$-value of $p < 0.0013$ and three stars indicate $p < 0.00013$.

### B. Student Survey Data

To determine the students' perceptions about the course, an online survey was administered during finals week via *Qualtrics* (Provo, UT). A total of $n = 11$ ME students responded to the survey. Participants were asked the Likert-scale questions listed in Table I.

The scale values are listed in Table II. These values were used in the statistical analysis and act as the shorthand reference. Additionally, they were given the option to include open-ended commentary with their answers.

The mean value for each response was individually compared with 4 (the neutral response). The response distributions were generally not symmetric about the mean, not normally distributed and small samples. Additionally, the sample size was "small" ($n < 30$). Therefore, bootstrapping analysis [23] with a resample size of 5 million was used to determine the significance in the difference of means. The significance threshold was set *a priori* to $\alpha = 0.05$. All statistical analysis was performed in Matlab (Natick, MA R2018a).

A boxplot of student responses to questions regarding their first programming course (Q1-Q4) is shown in Fig. 4. Here, bootstrapping was used to test if the mean response was significantly *lower* than 4 (students generally *disagreeing* with the prompt). The significance threshold was corrected for the four responses ($p < 0.0125 = 0.05/4$). No significance ($p \gg 0.0125$) was found for any prompt.

A boxplot of student responses to the outcome questions (G1-M2) is shown in Fig. 5. Here, bootstrapping was used to determine if the mean response was significantly *greater* than 4 (students generally *agreeing* with the prompt). The significance threshold was corrected for the eight responses ($p < 0.0063 = 0.05/8$). Statistical significance was found for all responses except for G2 (group learning).

Finally, the survey asked students to respond to "Overall, my experience in this course left me..." using five Likert items of "extremely dissatisfied" (1), "dissatisfied" (2), "neither satisfied or dissatisfied" (3), "satisfied" (4), and "extremely satisfied" (5). The mean response and [95% CI] was 4.3 [3.82, 4.82].

## IV. Discussion

The mean student opinion was *not* in disagreement with any of the four prompts concerning their introductory computer programming course (Q1-Q4). Furthermore, the mean ratings were all *greater* than 4, though only significant for Q4. This sentiment does not echo previous work investigating student motivation in introductory programming courses [8]. However, it should be noted that this course was an elective so there may be a selection bias of students more interested and adept in computer programming than the average student.

Based on question G1, students generally felt motivated to learn programming methods and techniques to accomplish their end goal of creating a video game. This agrees with previous work finding game creation to be a motivating factor [6–9]. Unlike these previous studies, all students in this course already had a basic foundation in programming prior to the course. There were no specific programming objectives set and each student chose their own path. This experience helps guide students to be lifelong learners and seek out skills and information as they need it.

G2 (team learning) was found not to be a significant help to the students. Some of the groups delegated programming tasks to different members so they may not have been learning the same skills at the same time. Furthermore, students were seeking online resources and individually watching videos.

An unanticipated result of the feedback was students being drawn to the "creative" aspects of the course (C3). Student groups had complete freedom in the design and implementation of their games. Students reported that the course required more creativity than their other core courses. The concept of "creativity" is, admittedly, nebulous. Furthermore, students

TABLE I: Survey questions given to students. Respondents selected their Likert-scale response using the choices in Table II. Prompts marked with * (C2 and M2), used the second column of responses. All others used the first.

| Code | Prompt |
|---|---|
| | *In my first programming course:* |
| Q1 | The way the material was presented in the class (e.g., lecture, video or textbook) helped me learn to program |
| Q2 | The homework and project problems were interesting to solve using programming |
| Q3 | At the end of the course, I was motivated to learn more programming skills |
| Q4 | I remembered what I learned in the course for use in upper-level courses |
| G1 | Having an end goal (i.e., making a video game) motivated me to learn the necessary programming |
| G2 | Learning a new programming language or programming skills in a group was helpful |
| G3 | Already knowing at least one programming language made it easier to learn a new one |
| C1 | This course required me to be creative in order to be successful |
| C2 | Compared to my other **in-major** courses, the creativity required in this course was * |
| C3 | I would prefer that my other **in-major** courses allowed me to be creative or inventive |
| M1 | To complete the project in this course, good organization and project management skills were required |
| M2 | Compared to my other **in-major** courses, the organization and project management required in this course was * |

TABLE II: Likert-Scale choices for survey questions in Table I. The numeric code was used for statistical analysis and shorthand notation.

| Code | Choice | Choice* |
|---|---|---|
| 1 | Strongly Disagree | Much Lower |
| 2 | Disagree | Moderately Lower |
| 3 | Slightly Disagree | Slightly Lower |
| 4 | Neither Agree nor Disagree | About the Same |
| 5 | Slightly Agree | Slightly Higher |
| 6 | Agree | Moderately Higher |
| 7 | Strongly Agree | Much Higher |

generally wanted more creative outlets in their other core courses (C1-C2). Again, there is a sampling bias of students that selected to take this course and this recommendation may not apply to all engineering students. However, these students sought an outlet for their creativity that they felt was missing from the general program.

The need for engineering students to be able to form creative solutions or "think outside the box" has been the subject of considerable research [24–28]. Teaching and promoting creativity in technical majors is important but overlooked (at least in our program). The majority of courses have students learn equations and principals and then apply these new tools to similar problems that they saw in class. It is worth investigating how to integrate concepts from a broader range of courses (technical and non-technical) to create unique solutions to homework and project problems.

Students felt that the course required good project management and planning skills (M1-M2). Having more experience with large-scale projects is beneficial to upper-level engineers as they prepare to apply for industry jobs [29] or graduate school.

Outside of specific course outcomes, students indicated that they were generally "satisfied" with the course. Since this course was an elective, students should have derived some enjoyment from it in addition to the technical skills acquired.

There are some limitations to these analyses and inferences. All outcome measurements were based on *self-reported* views at the end of the course. Though most students in the course did participate in the survey, the statistical power is limited by having only eleven respondents in the sample. However, strong statistical significance (based on $p$-value) was found for most prompts.

## V. Lessons Learned & Future Course Changes

Overall, the course was a success in its first offering on campus. Most of the course objectives were met, students were able to create a working video game and they felt like they received value out of the course. Students felt motivated to learn new skills on their own to finish the course project. Based on observations from the instructor and student survey data, improvements and suggestions follow. The major shortcoming of the course was that students did not learn the intended *high-level* programming concepts. All groups were drawn to the modern game engines for their project. Unfortunately, the learning curve on these software packages is shallow and groups spent most of the semester learning the basics of the specific software rather than designing and refining their games. Though students did learn programming and software skills through the project, they were not using the *fundamental* programming skills that are directly transferable to many engineering problems. The graphics-heavy engines added too much complexity to the project without the benefit of students learning to code.

To ensure that Objective 4 is achieved the next time the course runs, the project will be more constrained. Students will be provided with online resources (e.g., Lynda, Udemy, Couresera, edX) over winter break if they want to use a modern game engine (e.g., Unity, UE4). Otherwise, they will be restricted to high-level programming choices (e.g., C++, Python with PyGame). Students will be encouraged to integrate hardware into their video games. For example, they could use accelerometers and other sensors attached to an Arduino to control their game. The programming project will begin the first week of class rather than waiting until some theory is presented. Finishing the game first prototype earlier will allow more time for iterative refinement and play-testing.

No similar examples of critical analyses of video games

could be found prior to the beginning of the course. Now with several complete examples from this course, future students will have a better idea of the format and expectations for the assignment. The assignment will be modified in consultation with colleagues in English and Communications to more closely mirror close-reading assignments of literature or cinema.

The grand vision for this course is to reach across disciplines and include students with "non-technical" majors. Professional writing and English majors could lead the creation of the narrative and dialogue for a role-playing game. Kinesiology majors could perform motion capture with theatre majors as actors.

This version of the course and the survey represent only a pilot and initial data collection. The next instance of the course will have students complete a pre- and post-survey with both objective and subjective questions. Questions will be more closely tied to the previous literature and more specific.

## Acknowledgements & Notes

The author would like to thank the spring 2017 IST 446 class for their participation in the first offering of this course and for their survey responses. Thanks also to the guest presenters, Thom Woodley and Mohammad Alavi, for taking time out of their schedules to provide real-world insight to the students.

All survey data were collected with approval of the Penn State IRB (STUDY00006329). 

## References

[1] K. Becker, "End game," in *Choosing and Using Digital Games in the Classroom*. Springer, 2017, pp. 335–340.

[2] A. Mitchell and C. Savill-Smith, "The use of computer and video games for learning: A review of the literature," *Learning and Skills Development Agency*, 2004.

[3] R. Rosas, M. Nussbaum, P. Cumsille, V. Marianov, M. Correa, P. Flores, V. Grau, F. Lagos, X. López, V. López *et al.*, "Beyond nintendo: design and assessment of educational video games for first and second grade students," *Computers & Education*, vol. 40, no. 1, pp. 71–94, 2003.

[4] L. A. Annetta, W. M. Frazier, E. Folta, S. Holmes, R. Lamb, and M.-T. Cheng, "Science teacher efficacy and extrinsic factors toward professional development using video games in a design-based research model: The next generation of stem learning," *Journal of Science Education and Technology*, vol. 22, no. 1, pp. 47–61, 2013.

[5] B. M. Maguth, J. S. List, and M. Wunderle, "Teaching social studies with video games," *The Social Studies*, vol. 106, no. 1, pp. 32–36, 2015.

[6] K. Becker, "Teaching with games: the minesweeper and asteroids experience," *Journal of Computing Sciences in Colleges*, vol. 17, no. 2, pp. 23–33, 2001.

[7] J. C. Adams, "Chance-it: an object-oriented capstone project for cs-1," *ACM SIGCSE Bulletin*, vol. 30, no. 1, pp. 10–14, 1998.

[8] R. Moser, "A fantasy adventure game as a learning environment: why learning to program is so difficult and what can be done about it," in *ACM SIGCSE Bulletin*, vol. 29, no. 3. ACM, 1997, pp. 114–116.

[9] S. Leutenegger and J. Edgington, "A games first approach to teaching introductory programming," *ACM SIGCSE Bulletin*, vol. 39, no. 1, pp. 115–118, 2007.

[10] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman *et al.*, "Scratch: programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60–67, 2009.

[11] I. Ouahbi, F. Kaddari, H. Darhmaoui, A. Elachqar, and S. Lahmine, "Learning basic programming concepts by creating games with scratch programming environment," *Procedia-Social and Behavioral Sciences*, vol. 191, pp. 1479–1482, 2015.

[12] F. Ke, "An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing," *Computers & Education*, vol. 73, pp. 26–39, 2014.

[13] J. H. Maloney, K. Peppler, Y. Kafai, M. Resnick, and N. Rusk, *Programming by choice: urban youth learning programming with scratch*. ACM, 2008, vol. 40, no. 1.

[14] B. Coller, "Implementing a video game to teach principles of mechanical engineering," in *114th Annual ASEE Conference and Exposition, 2007*, 2007.

[15] B. D. Coller and M. J. Scott, "Effectiveness of using a video game to teach a course in mechanical engineering," *Computers & Education*, vol. 53, no. 3, pp. 900–912, 2009.

[16] J. M. Mahoney, "Case study of a video game design & development course for mechanical engineers," in *ASEE Mid-Atlantic Section Fall 2017 Conference*. American Society of Engineering Education, 2017.

[17] Z. Hiwiller, *Players making decisions: Game design essentials and the art of understanding your players*. New Riders, 2015.

[18] E. Adams, *Fundamentals of game design*. Pearson Education, 2014.

[19] W. Despain, *100 principles of game design*. New Riders, 2013.

[20] R. Koster, *Theory of fun for game design*. O'Reilly Media, Inc., 2013.

[21] N. Lazzaro, "Why we play games: Four keys to more emotion in player experiences," in *Proceedings of GDC*, vol. 306, 2004.

[22] J. M. Mahoney, "Video game course for engineers," https://osf.io/uxfr5/, November 2018.

[23] P. Hall, "Theoretical comparison of bootstrap confidence intervals," *The Annals of Statistics*, pp. 927–953, 1988.

[24] D. H. Cropley and A. J. Cropley, "Fostering creativity in engineering undergraduates," *High ability studies*, vol. 11, no. 2, pp. 207–219, 2000.

[25] D. H. Cropley, "Promoting creativity and innovation in engineering education." *Psychology of Aesthetics, Creativity, and the Arts*, vol. 9, no. 2, p. 161, 2015.

[26] S. R. Daly, E. A. Mosyjowski, and C. M. Seifert, "Teaching creativity in engineering courses," *Journal of Engineering Education*, vol. 103, no. 3, pp. 417–449, 2014.

[27] C. Charyton and J. A. Merrill, "Assessing general creativity and creative engineering design in first year engineering students," *Journal of engineering education*, vol. 98, no. 2, pp. 145–156, 2009.

[28] C. Charyton, R. J. Jagacinski, J. A. Merrill, W. Clifton, and S. DeDios, "Assessing creativity specific to engineering with the revised creative engineering design assessment," *Journal of Engineering Education*, vol. 100, no. 4, pp. 778–799, 2011.

[29] D. Q. Nguyen, "The essential skills and attributes of an engineer: A comparative study of academics, industry personnel and engineering students," *Global J. of Engng. Educ*, vol. 2, no. 1, pp. 65–75, 1998.

**Joseph M. Mahoney** Dr. Joseph Mahoney is an Assistant Professor of Mechanical Engineering at Penn State Berks. He received both his BS (with Honors) and MS in Mechanical Engineering from Penn State. He received his Ph.D. in Engineering Science and Mechanics also from Penn State. His research is in Biomechanics and his teaching is in Statics, System Dynamics, Vibrations and Video Game Design. He is a member of The American Society of Biomechanics and has reviewed for ASEE, International Journal of Industrial Ergonomics, Ergonomics, Safety and Health at Work, and BMC Musculoskeletal Disorders. He was workshop chair and co-organizer for the ASEE Mid-Atlantic Section Fall 2017 Conference. He was the chair of the 2018 American Society of Biomechanics East Coast Meeting.