
AC 2012-3659: PRELIMINARY RESULTS ON USING A VIDEO GAME IN TEACHING DYNAMICS

Dr. Brianno D. Collier, Northern Illinois University

Brianno Collier is an Associate Professor of mechanical engineering. He started his research career applying fairly deep mathematical ideas to gain insight into how complex physical and engineering systems work. His work was theoretical and somewhat abstract. Since then, his research has evolved toward studying a different type of complex system: how students learn and become excited about engineering. In this endeavor, Collier is mostly a "nuts and bolts" practitioner, an engineer, and an experimentalist.

Preliminary Results on Using a Video Game in Teaching Dynamics

Introduction

Malcolm Gladwell¹ famously argued that the way one becomes an expert at something, almost anything, is to practice for 10,000 hours. Given that 10,000 hours is equivalent to practicing 20 hours per week for nearly ten years, it seems unlikely that an undergraduate student can develop anything close to expertise over the course of a 15-week semester. Nonetheless, in a sophomore-level engineering dynamics course, it is hard to argue against giving students as much homework as practical. In dynamics, particularly, there is an incredibly rich variety of permutations of mass configurations, force elements, and boundary conditions that one can consider. By making a small modification, one can transform a straightforward $F = ma$ problem to a work-energy problem. It takes practice to recognize the subtle differences. The good news is that our textbooks are filled with problems to solve.

The unfortunate news is that the vast majority of these problems are highly structured, narrowly focused questions that have single “right” answers. These types of problems seem antithetical to what attracts students to engineering. Our students often tell us that they chose to pursue engineering because they want to build things, to create things, to make things work. Solving dynamics problems with the goal of matching the answer in the back of the book is very different from solving problems with the objective of making something work. In the two different perspectives, knowledge is valued quite differently. The latter is more aligned with how engineers think and know.

Recognizing the disconnect between these two different epistemologies, and arguing for the need to introduce design thinking and inductive thinking earlier into the curriculum, Sheppard et al.² have made a powerful argument, firmly grounded in cognitive science, for balancing textbook-type problems with less structured, more open-ended assignments. Of course, the challenge of designing effective open-ended learning experiences is far from trivial. Sheppard et al.² observe that novice students, who are capable of solving highly structured problems, are often “at sea” when confronted with open-ended challenges. Sheppard and coauthors call for easing students through a gradual process in which the task, or series of tasks, starts with a considerable amount of structure and scaffolding. As the student develops skills, layers of structure and scaffolding are removed and complexity is added.

It is interesting to note that modern video games are designed in much the same way, with levels. In the initial levels of a game, challenges are relatively simple. Many of the game’s characteristics are removed so that a player may focus on developing specific elementary skills. As a player achieves a degree of proficiency, he/she advances to the next level where more of the game mechanics and more complexity are revealed. Education scholars³⁻⁷ who study video games have found that the most successful games often incorporate mechanisms that “teach” their players to solve complex problems by leveraging learning pedagogies such as constructionism, inquiry-based learning and anchored instruction. Players are motivated to learn within video games because it is clear that knowledge is powerful. Learning is situated, and occurs through a process of hypothesizing, probing, and reflecting upon the simulated world within the game. The goals are clear. Games provide players immediate and unambiguous

feedback on how well they are progressing. Information becomes available to players at just the time they will be able to make sense of it and use it. Much of the emerging literature^{8,9} on video game design is explicitly grounded in scholarship on cognition, including concepts such as Vygotsky's *zone of proximal development*.

Recognizing the potential of video games as potential platforms for learning engineering, I began experimenting with a car/motorcycle driving game in a junior-level numerical methods course and a junior-level dynamic systems & control course. In these experiments, we found that students learning with the game scored better on concept tests¹⁰, show deeper levels of learning¹¹, exhibited higher levels of engagement¹², and signed up for advanced engineering courses at a higher rate¹².

Recently, we began creating another video game called *Spumone*, this time covering core content in a sophomore-level engineering dynamics course. We have given an overview of the game previously¹³. Now that we have collected two semesters worth of data on two main challenges in the game, we are ready to present an initial analysis on the effectiveness of the game.

The Video Game

We built *Spumone* from scratch. Currently, it has two main dynamics challenges for students to solve, and we are play-testing two more. The general premise is that the student/player pilots a vehicle called the *spuCraft* as he/she explores a subterranean, two-dimensional, dynamic world. In order to make progress in the *spuWorld*, the student/player must put on one's engineering hat and figure out how to make something work, dynamically. Screenshots are shown in Figures 1 through 3.

The *spuCraft* is a collection of masses, each with negligible rotational inertia, connected by nominally rigid rods in a truss-like manner, almost like Tinker Toys. The masses have different characteristics. Some masses disintegrate as soon as they touch other solid objects, while other masses are more durable, permitting the *spuCraft* to land on suitable surfaces. The connecting rods have finite strength, so, if a landing is too hard, one or more of the rods might break, causing the craft to no longer move as a single rigid body.

The astute dynamicist will observe that many, if not most, commercial video games have flaws in their physics. In contrast, the physics engine in *Spumone*, is a high fidelity engineering-quality simulation that runs in real time. It conserves energy and momentum when it should. Holonomic constraints are maintained through a Lagrange multiplier approach.

Spumone Free Fall Challenge

Within the first two weeks of the semester, students encounter their first game-based challenge. It is called Free Fall. The *spuCraft* is dropped from a height of several hundred meters above a landing pad at the bottom of a cavern. The goal is for students to bring the *spuCraft* to a soft, safe landing on the pad. The student can slow down the *spuCraft* by firing its thrusters. Notice the green flames in Figure 1. With a little practice, students can quickly learn to manually land the

spuCraft softly on a landing pad. There is another part of the game in which students/players can gain such practice.

The tricky part of the challenge can be seen at the top and left side of Figure 1. As the craft is descending, a roof is closing over the landing pad. Therefore, if the student slows down too soon, the roof crosses the spuCraft's path and prevents it from reaching the landing pad. Conversely, if the student slows down too late, it will hit the landing pad with a lethal amount of speed.

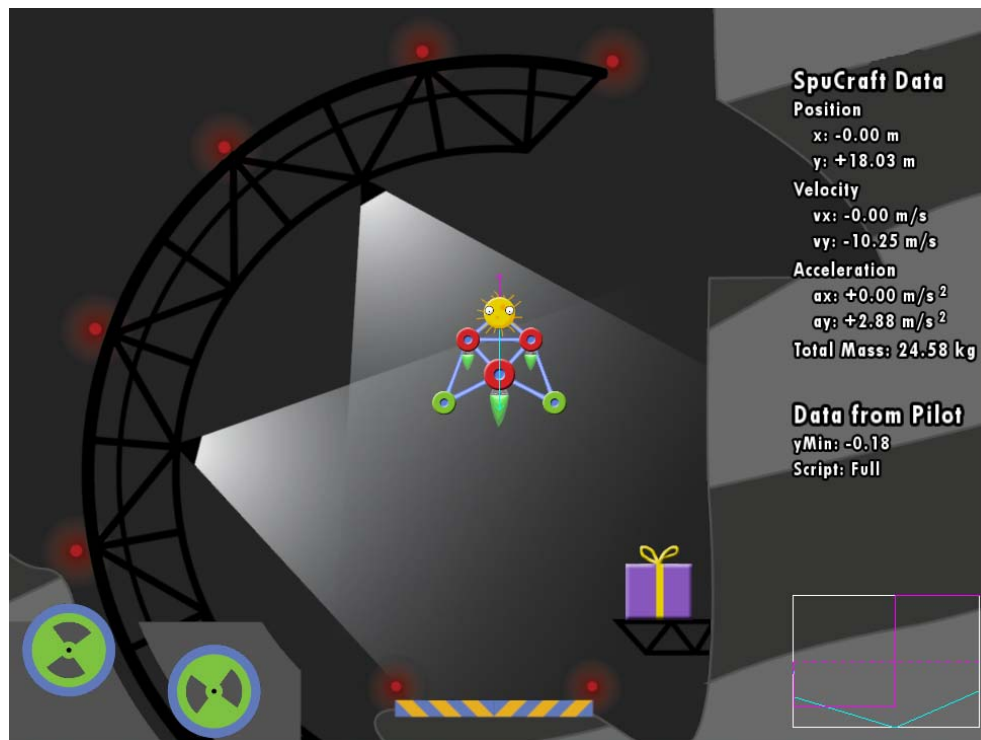


Figure 1: Screenshot of the *Spumone* Free Fall Challenge.

Game controller in hand, students attempt to time the thrust perfectly, but it is a nearly hopeless exercise. The game is carefully designed so that it is almost impossible complete by hand. The time window in which players have to safely turn on the thrust is about a tenth of a second in duration. Furthermore, since the mass of the spuCraft is different each time one runs the game, the time (or height) at which one lights the thrusters is always different.

In addition to the gamepad controller, however, students can interact with *Spumone* with a simple programming interface. One of the windows of the interface is shown in Figure 2. With the programming interface, one can write mathematical rules which tell the craft to turn on thrusters as a function of quantities given: *i.e.* current height, current velocity, and mass of the spuCraft, as well as gravitational constant.

To successfully complete the challenge, students have to derive mathematical rules which make it work. This is the whole point of the course! But it is not a typical textbook-like problem. Since they are not given any information about the closing roof, students have to recognize that this, at its core, is a minimum time problem: bring the spuCraft to a stop on the landing pad in the least amount of time. Next, students have to recognize that the solution to this minimum time problem

amounts to letting the spuCraft fall for as long as possible, then turning on full thrust at the last possible moment. We have provided students a fair amount of scaffolding to help them think through these first two hurdles.

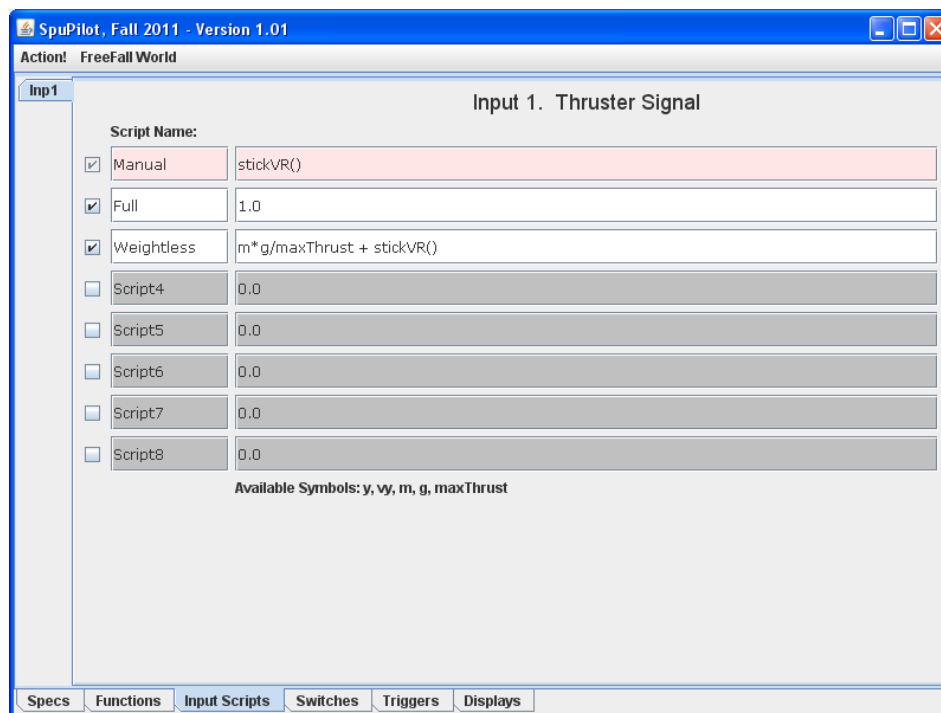


Figure 2: Screenshot of the student programming interface.

Having discovered the structure of the problem, finding the conditions for turning on the thrusters is still nontrivial because the initial height from which the craft was dropped is not a given quantity. However, based on the *current* height and speed of the spuCraft, one can derive a mathematical expression which indicates whether it is too early or too late to turn on thrusters. Students can derive the condition by integrating Newton's second law of motion directly, or (more elegantly) by employing the work-energy principle.

Spumone Swingline Challenge

A screenshot of the final Spumone challenge is shown in Figure 3. In this event, the spuCraft has three masses. We call the red mass the “snare.” Under normal conditions, the snare is constrained to slide without friction along the green horizontal line (the swingline) shown in Figure 3. Meanwhile the rest of the body is free to rotate about the snare.

By pushing a button on the gamepad controller, students can make the snare “grab” onto the swingline, impulsively bringing the velocity of the snare immediately to zero. The rest of the spuCraft is still free to rotate about the snare. By pushing the button again, one can release the grab constraint so that the snare is free to slide along the swingline again.

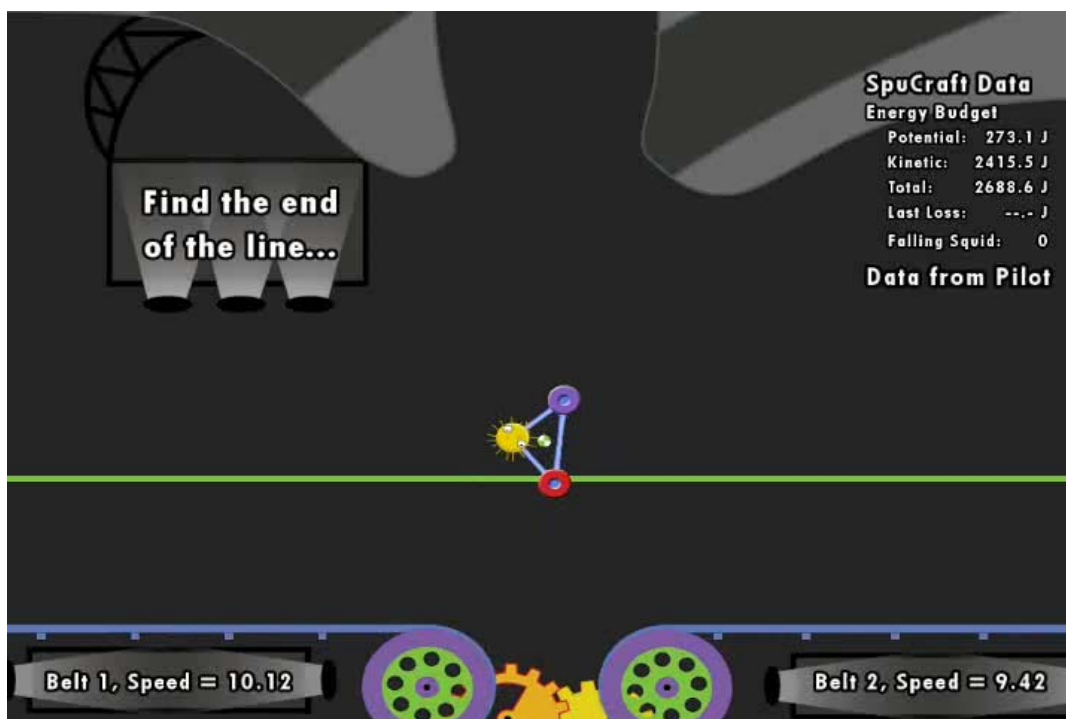


Figure 3: Screenshot of the *Spumone Swingline Challenge*.

In the original “state,” the spuCraft may slide along the swingline and simultaneously rotate. All external forces are vertical, so the horizontal acceleration of the center of mass is zero. In the other “state” the spuCraft only rotates about the fixed snare. External forces are vertical *and* horizontal. By switching between these states, one can generate locomotion.

To successfully complete the Swingline Challenge, one must elegantly coordinate switching between the grab and release states. The spuCraft must traverse across the simulated world at exactly the right speed, switching speeds at the right times, and do so without losing more than just a few percent of its original total (kinetic + potential) energy. Like the Free Fall Challenge, it is essentially impossible to complete by hand. Students must perform quantitative analyses and express strategies mathematically. The task requires students to understand planar kinematics dynamics of rigid bodies.

Impact on Learning: Design of Experiment

One of the primary goals of the video game intervention is to improve learning outcomes. At the time we were first imagining how we would create such a video game, we thought that, if it did have an impact on learning, the impact would be most apparent in students’ conceptual understanding of the course material. Therefore, to evaluate the impact of the game-based teaching intervention, we focused on comparing students’ scores on established concept inventories.

Control Group

The control group consisted of ($N_N = 42$) students who took the engineering dynamics course in the Spring of 2010. At that time, *Spumone* did not exist; not a single line of computer code had been written. Our idea of a dynamics video game was something we had tentatively called *DynaMonkey*. It was similar to a three dimensional version of *Pong*, very different from *Spumone*.

Although the control group did not have a video game to use for learning, students were required to complete two semi-structured, project-based assignments. The first project was a dynamic analysis of the Vancouver luge track where a 21 year old Olympic athlete had recently lost his life. In the second assignment, students had to derive equations of motion for an electric cart (four rigid bodies connected by axle bearings and a chain/sprocket) and integrate the equations with Matlab. Through intuition-guided iteration, students selected cart parameters they thought would give them the best chances of winning a class-wide race.

Experimental Group

Our experimental group consisted of students who took engineering dynamics in Spring 2011, and Fall 2011. For these two classes, the project-based assignments consisted of the Free Fall and Swingline challenges described in the previous section.

In these classes, there were a few students who were using *Spumone* in a separate context, in addition to their engineering dynamics work. Because their experience with the game was considerably different, we did not include their data in this study. Our experimental group consisted of $N_G = 62$ students.

Pretest

On the first day of the semester, students from each group took a pretest consisting of 19 questions from the Force Concept Inventory (FCI)¹⁴ and the Mechanics Baseline Test (MBT)¹⁵. The instruments assess understanding of concepts covered in an elementary calculus-based physics course which serves as an important prerequisite to engineering dynamics. They are reliable and valid^{14, 15}.

Posttest

On the final day of the semester, we measured conceptual understanding of topics discussed in the course, through an instrument called the Dynamics Concept Inventory (DCI)¹⁶. Five of the 29 items on the DCI are the identical to items on the FCI. The remaining items are unique.

When we report results in the next section, we will split the DCI items into two categories. One category consists of questions 2, 3, 4, 6, 11, 12, 17, 21, 22, 23, 24, 27, and 28. These are the 13 items covering rigid body kinematics and dynamics that are directly related to the *Spumone* Swingline challenge. The remaining 16 items correspond to concepts cover in the course, but not directly covered by either of the *Spumone* exercises.

Other Data

In addition to the concept test data, we collected a considerable amount of demographic and survey data for these students. Also, we logged all their actions as they played Spumone. These data are not analyzed in this article.

Impact on Learning: Results

Pretest and posttest results are presented in Figure 4 in terms of Cohen effect size:

$$\frac{\bar{x}_G - \bar{x}_N}{S}$$

Here \bar{x}_G and \bar{x}_N represent the average scores of students in the experimental (Game) and control (Non-game) groups respectively. Symbol S represents the pooled standard deviation. Therefore, if the quantity is positive, students in the game-based experimental group performed better. If negative, students in the control (non-game) group performed better.

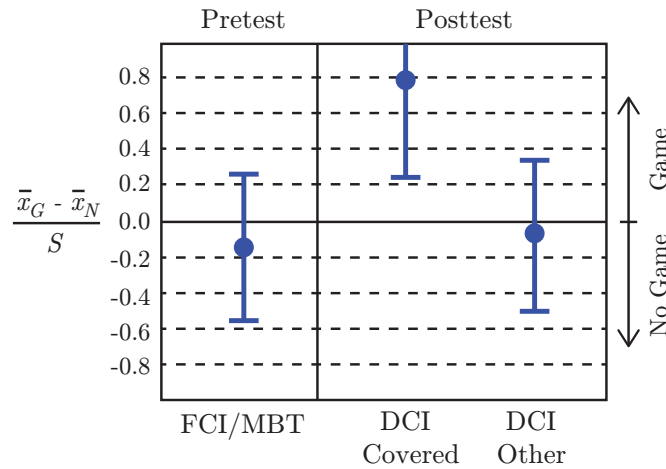


Figure 4: Pretest and posttest results.

The first result in Figure 4, shows that for the FCI/MBT pretest conducted at the beginning of the semester, students in the control group scored slightly better ($T = -0.69$). However, the error bars denote the 95% confidence interval for the effect size. It is clear that the difference is not statistically significant.

As discussed in the previous section, we split the results of the posttest into two categories. For the 13 rigid body kinematics and dynamics items on the DCI, directly covered by the *Spumone* assignments, students taking the game-based course scored significantly better than students in the control group ($T = 2.89$, $p < 0.001$). An effect size of 0.8 is generally considered large in the behavioral sciences^{17, 18}.

The final result of Figure 4 shows that both groups of students scored similarly ($T = -0.37$) on the “other” items on the DCI that were not directly covered in Spumone. Therefore the gains in the Spumone-related concepts did not come at the expense of other topics.

Discussion

These preliminary results are both encouraging and consistent with our previous studies of using video games in upper level undergraduate courses^{11, 12}. As a first step toward understand why and how these learning gains occurred, we present two scatter plots in Figure 5. The horizontal axis of each plot represents students’ normalized pretest scores. The vertical axis measures students’ normalized posttest scores for those items directly covered in the *Spumone* Swingline assignment.

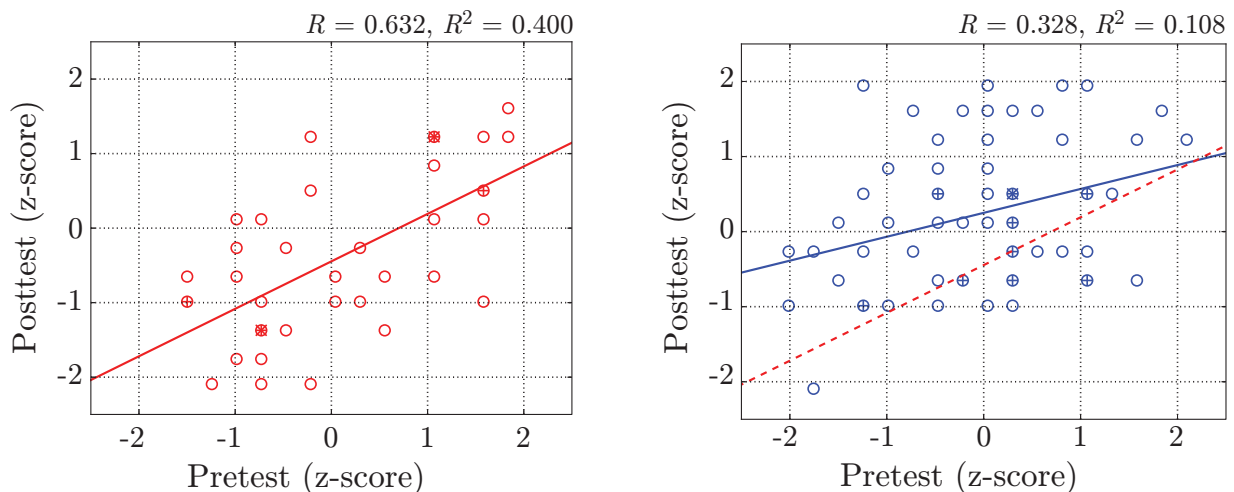


Figure 5: Scatter plots showing normalized pretest and posttest scores. The plot on the left is for students in the control group (no game). The plot on the right is for students in the experimental group (with game).

The scatter plot on the left, for students in the control group who learned without the benefit of the video game, depicts a strong correlation between pretest and posttest scores. With a correlation coefficient of $R = 0.632$, we can say that roughly 40% of the posttest variance can be attributed to pretest performance. The data suggest that students who enter engineering dynamics with poor understanding of basic mechanics concepts tend to have a poor understanding of more advanced dynamics concepts at the end of the semester. Students who score low on the pretest seem to be “doomed” to perform poorly in the class.

However the story appears to be rather different in the game-based engineering dynamics course depicted on the right side of Figure 5. For the experimental group, the connection between pretest and posttest is weaker. Only 10% of the posttest variance can be attributed to performance on the pretest. Introduction of the video game appears to have “leveled the playing field” to some degree. Students with high pretest scores are still scoring high at the end of the

semester. But now, many more of the students who scored average and below average initially are able to perform well by the end of 15 weeks.

One possible interpretation of the scatter plot data in Figure 5 is that the video game framework makes the course content easier to understand. Perhaps it provides a new perspective or learning modality that resonates better with college-age students. But if that were the case, one might expect students to learn the material and finish their assignments more quickly.

In fact, the opposite appears to be happening. On average students “played” Spumone more than 700 times over the semester. Therefore, an alternative hypothesis is that students with low pretest scores are spending more time on their coursework because they are motivated to get their projects to work. With the extra time on task, these students learn more. The hypothesis is consistent with our previous results demonstrating that students working on video game-based assignments in a junior-level control course are more engaged than their counterparts in an equivalent non-game course¹².

As stated earlier in the paper, Spumone collects detailed information about how students play the game. The logs contain information on how frequently students “play”, how long it takes them to achieve certain milestones within each challenge, and copies of each iteration of their mathematical rules. Additionally, we attempted to measure student engagement by embedding experience sampling surveys¹⁹ into the game. In the near future, we plan to analyze this data to look for patterns that might support or refute our working hypothesis.

References

1. M. Gladwell, *Outliers: The Story of Success*. Back Bay Books, 2011.
2. S. D. Sheppard, K. Macatangay, A. Colby and W. Sullivan, *Educating Engineers: Designing for the Future of the Field*. Jossey-Bass, 2009.
3. T. W. Malone, Toward a theory of intrinsically motivating instruction, *Cognitive Science* 5 (4), pp. 333-369, 1981.
4. J. P. Gee, *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave Macmillan, 2003.
5. C. Aldrich, *Learning by Doing*. Pfeiffer, 2005.
6. Federation of American Scientists, *Harnessing the Power of Video Games for Learning*, 2006.
7. D. W. Shaffer, *How Computer Games Help Children Learn*. Palgrave Macmillan, 2006.
8. T. Fullerton, *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. Morgan Kaufmann, 2008.
9. J. Schell, *The Art of Game Design: A Book of Lenses*. Morgan Kaufmann, 2009.
10. B. D. Collier, in *Proceedings of the American Control Conference* (Baltimore, MD, 2010).
11. B. D. Collier and M. J. Scott, Effectiveness of using a video game to teach a course in mechanical engineering, *Computers & Education* 53 (3), pp. 900-912, 2009.
12. B. D. Collier, D. J. Shernoff and A. D. Strati, Measuring Engagement as Students Learn Dynamic Systems and Control with a Video Game, *Advances in Engineering Education* 2 (3), 2011.
13. B. D. Collier, in *ASEE Annual Conference* (Vancouver, BC, 2011).
14. D. Hestenes, M. Wells and G. Swackhamer, Force concept inventory, *The Physics Teacher* 30, pp. 141-158, 1992.

15. D. Hestenes and M. Wells, A mechanics baseline test, *The Physics Teacher* 30, pp. 159-165, 1992.
16. G. L. Gray, F. Costanzo, D. Evans, P. Cornwell, B. Self and J. L. Lane, presented at the Proceedings of the 2005 American Society for Engineering Education Annual Conference, 2005.
17. J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Academic Press, 1988.
18. J. Cohen, A power primer, *Psychological Bulletin* 112 (1), pp. 155-159, 1992.
19. J. Hektner, J. A. Schmidt and M. Csikszentmihalyi, *Experience Sampling Method: Measuring the Quality of Everyday Life*. Sage, Thousand Oaks, CA, 2007.