AC 2011-1384: FIRST LOOK AT A VIDEO GAME FOR TEACHING DYNAMICS

Brianno Coller, Northern Illinois University

Brianno Coller 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, Dr. Coller is mostly a "nuts & bolts" practitioner, an engineer, and an experimentalist.

First Look at a Video Game for Teaching Dynamics

Introduction

For decades, education scholars have been studying video games[1-5]. What they have found is that the most successful games often "teach" their players how to solve complex problems. The problems within a game typically start off rather easy and then progressively get more difficult as players' skills develop. Players are motivated to learn within video games because it is clear that knowledge is powerful. The 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.

Within the highly engaging techniques that game designers employ to get players to "learn" the game, one finds echoes of modern learning pedagogies such as constructionism, inquiry-based learning, and anchored instruction. Much of the emerging scholarship on video game design[6-8] is explicitly grounded in scholarship on cognition, including concepts such as Vygotsky's *zone of proximal development*.

Over the past half decade, I have experimented with integrating video games into core undergraduate mechanical engineering courses. In particular, I developed an automobile/bicycle driving game called *EduTorcs* for teaching Numerical Methods and for teaching Dynamic Systems & Control. In the former case, we found that students learning numerical methods with a video game learned the material more deeply, as measured by a concept map assessment[9]. In the dynamic systems & control class, we found that students who learned with video game-based homework and laboratory exercises scored significantly better on concept tests[10]. Furthermore, using a technique known as the experience sampling method, we found students learning dynamic systems & control with a video game are significantly more engaged[11]. Furthermore, these students were much more likely to take the more advanced dynamical systems & control course as a technical elective[11].

Recently, I have begun developing a new video game for teaching/learning engineering dynamics. The game is still very much a work in progress. Nonetheless, we are getting some preliminary results that indicate that it is having a positive impact on learning. In this paper, I outline some of the progress we have made and discuss where we are headed.

A Different Approach to Developing and Implementing the Dynamics Game

My previous game, *EduTorcs*, was derived from an existing, open-source game called *Torcs* (www.torcs.org). It has look, feel, and 3D graphics similar to what one finds in commercial video games. We built a programming interface so that students could write algorithms for driving virtual cars and bicycles/motorcycles around a track. We added new vehicles to the game. We created new challenges that required students to apply their knowledge of numerical methods and control system design in order to achieve goals within the game. The challenges we created spanned most of the topics covered in the courses.

In developing the dynamics game, we deliberately chose a different approach. The goal was to create a lightweight, flexible research platform that would allow us to study how different game features impact student learning and student engagement. For example, we could investigate effects of different goal structures and scoring mechanisms embedded in the game. We could investigate effects of cooperative versus competitive play. The intent was to design the game to target specific, limited, learning outcomes so that we can measure its effect.

The Game Environment

Our game is called *Spumone*. We have built it from scratch. It is still in its early stages of development, so elements of the game are changing all the time. The general premise is that the student/player controls a vehicle which we call the spuCraft as it explores a labrynthian, subterranean world. A screenshot is shown in Figure 1.

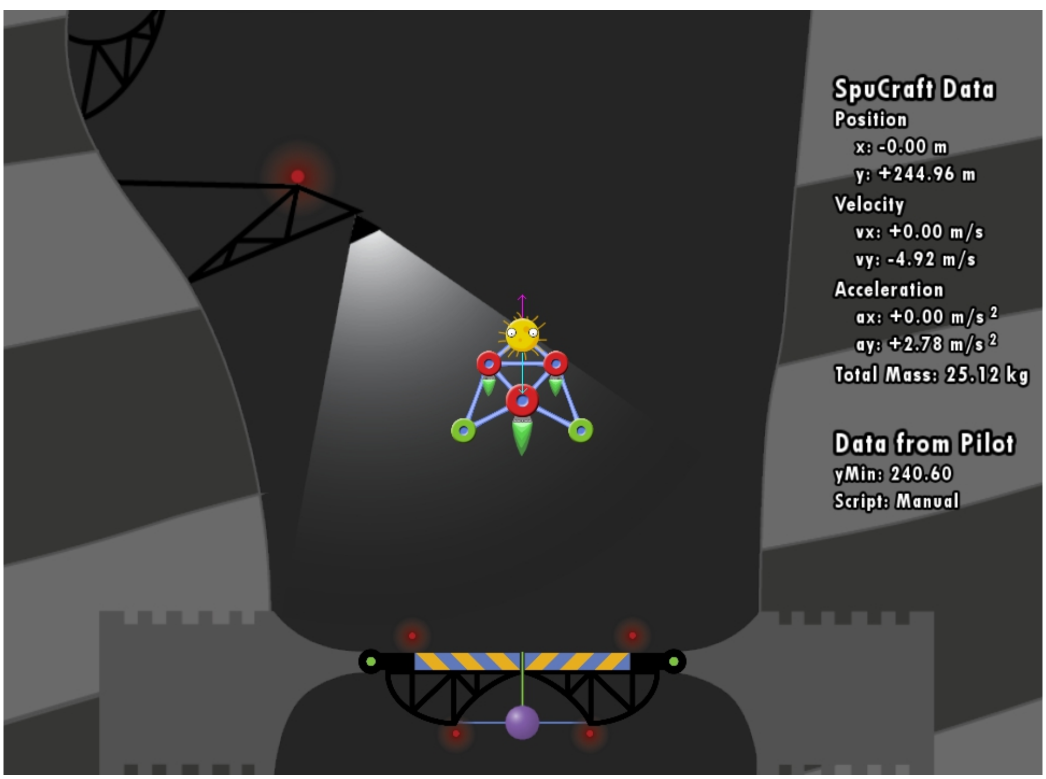

Figure 1. Screenshot of the Free Fall world within the video game *Spumone*.

The spuCraft is a collection of masses connected by nominally rigid rods in a truss-like manner, almost like Tinker Toys. The different masses have different characteristics. Some masses disintegrate as soon as they touch other solid objects, while other masses are more durable, permitting the craft 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 spuCraft to no longer move as a single rigid body. Thrusters that are attached to certain masses may be used to propel and steer the spuCraft through the maze.

In different parts of the "game," the masses of the spuCraft are configured differently. However, the different configurations always include one special mass called Pokey. Pokey looks like a cartoon sunshine with eyes that occasionally blink and wink. Pokey seems to give the spuCraft personality. When a student/player executes a good maneuver, Pokey sometimes responds approvingly with a gleeful sound effect. When a player makes a poor move, Pokey often responds with a disapproving mumble.

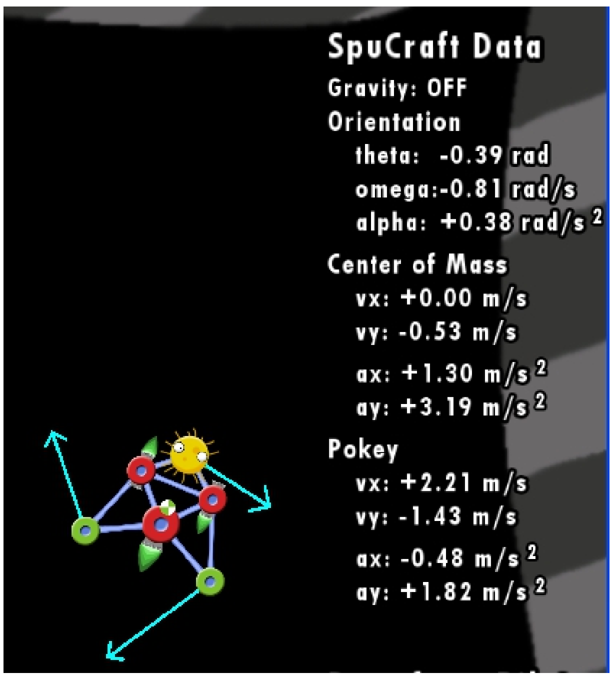

Figure 2: An event within *Spumone* that allows students to observe and experiment with rigid body kinematics. The light blue arrows represent velocities of various parts of the spuCraft.

The computer simulation of the planar spuCraft dynamics has the degree of accuracy that one would want in a video game for learning dynamics. Energy is (essentially) conserved under the right circumstances, as are linear and angular momenta. In the world of *Spumone*, we have chosen a gravitational acceleration of 2.0 m/s^2. The more customary value of 9.81 m/s^2 produces dynamics that are rather fast and difficult for players to control with a joystick.

What we have described thus far is something professional video game designers would call a "toy" rather than a "game."[6,7] Having a compelling "toy," though, is often regarded as an important initial step in developing the rules and challenges that would turn it into a video "game."[6,7] Over the first half of the Fall 2010 semester, I had students play with the spuCraft toy to explore relationships between position, velocity, and acceleration; to experience the difference between mass and weight; to observe how different points on a rigid body have different velocities and accelerations (See Figure 2); and to examine how the body's center of mass responds to external forces and moments.

At the end of the semester, students played a "game" within Spumone that served as their final project.

The Game

For the final game in *Spumone*, players/students entered a world called "Sling." I show a screenshot in Figure 3.

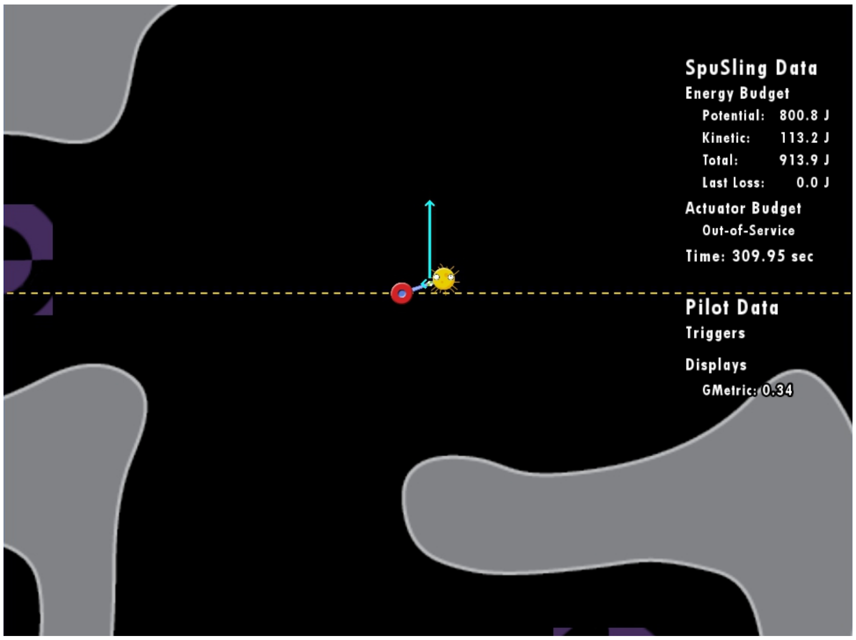

Figure 3. Screenshot of the Sling world within *Spumone*. The light blue arrows represent the horizontal and vertical components of the velocity of the center of mass of the spuCraft.

In this world, the spuCraft consists of two masses: Pokey, and another mass that we call the "Snare." The spuCraft has no thrusters or other internal sources of energy. Instead, the spuCraft gains kinetic energy via the work done by gravity.

As the spuCraft is falling the player/student may maneuver the craft by activating and deactivating constraints. In particular, when the player pushes one button on the joystick, the Snare "grabs" onto the background, and the Snare mass is suddenly constrained to move (without friction) along a horizontal line. The horizontal constraint is shown as a dashed line in Figures 3 and 4. By pressing the same button again, the Snare releases its grip and horizontal constraint is removed.

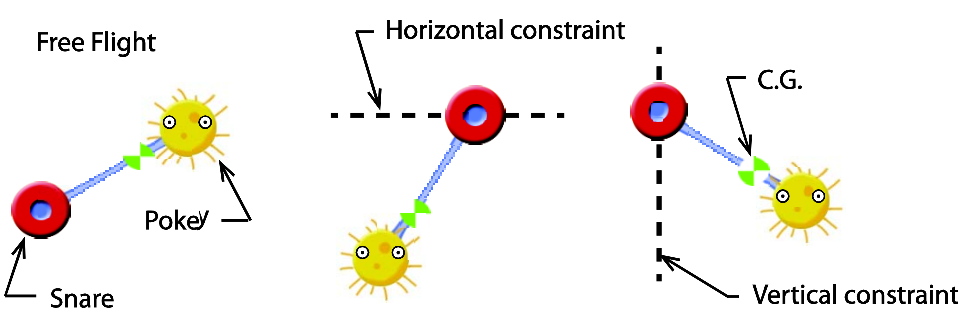

Figure 4. Pokey, the Snare, and constraints.

Similarly, there is another button on the joystick with which the player may activate/deactivate a vertical frictionless constraint. If the player chooses to activate the horizontal and vertical constraints simultaneously, the Snare is forced to rotate around a fixed point and Pokey is constrained to move on a circular path.

By activating and deactivating the constraints at opportune moments, the player can change the direction of the spuCraft's momentum, and hence she can navigate it through the maze, without touching the walls. (Touching a wall causes instant death of Pokey.)

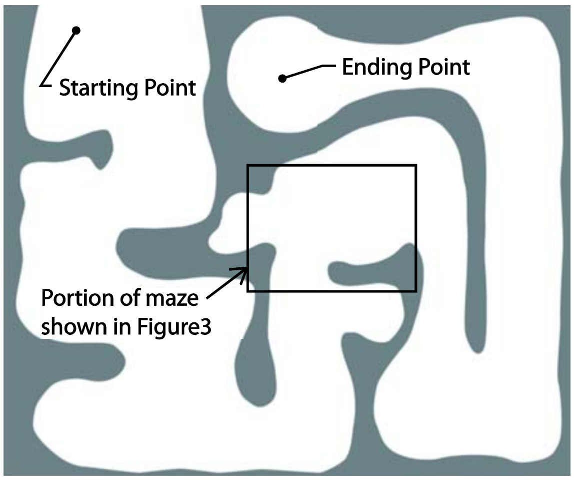

Figure 5. Maze in the Sling world.

The challenge comes from the shape of the maze, shown in Figure 5. The first part of the maze requires some side-to-side horizontal motion, but vertically, it's all downward. Gravity does positive work on the spuCraft which increases its kinetic energy.

Eventually, though, the spuCraft must move back upward. Generally, this is accomplished by releasing the horizontal constraint when the velocity of the center of mass has a positive vertical component. In this process, kinetic energy of the craft is exchanged for potential energy in the form of a higher elevation. Since the ending point of the maze is almost at the same elevation as the starting point, almost all the positive work performed by gravity as the spuCraft fell must be recovered as the spuCraft is "slung" back upward!

The problem is that each time the Snare grabs (i.e. a horizontal or vertical constraint gets activated), the spuCraft loses some of its kinetic energy. If one is not careful, it is not uncommon to lose more than 80% of the kinetic energy. Therefore, students must figure out how to "grab" very efficiently. *In order to make it to the ending point, the "grabs" must lose almost no energy*.

To figure out how to employ the constraints efficiently and navigate through the entire maze, students had to think about the problem from multiple perspectives: angular momentum, energy, and rigid body kinematics/kinetics. The problem was hard!

Even when one did figure out how to best engage and release the constraints, it was almost impossible for a human with normal dexterity to push the buttons at exactly the right times, especially at the bottom of the maze where kinetic energy must be quite large. Therefore, we created a set of software-based programmable triggers for the game. There was a user interface in which students could type equations into the game. Dynamic variables that students had access to included the position, velocity, and acceleration of Pokey, and the position and velocity of the center of mass. As the game was running, *Spumone* would automatically engage or release selected constraints at the instant the students' equations were satisfied. In the end, achieving goals within the game required brains rather than super-human reflexes.

Preliminary Results

In the Fall of 2010, all 39 students who took the engineering dynamics course at NIU described the final game-based project, as "very challenging." Nonetheless, all but two of the students were able to successfully complete the challenge *and* write a report providing sufficient technical detail to give me confidence that they understood the necessary dynamics to complete the game's task.

As a more objective measure of student learning, I had students take a series of standardized tests. On the first day of the semester, students took a test consisting of selected questions from the Force Concept Inventory[12] and the Mechanics Baseline Test[13]. In the final week of the classes, students took the Dynamics Concept Inventory (DCI)[14]. Since the final game-based project was not due until the week after classes ended, the DCI was administered while students were working on their projects. Most had not completed their projects at the time the DCI was administered.

For comparison, I had the students who took engineering dynamics in Spring 2010, without the game, complete the same sequence of tests at the same times during the semester. Both the control group and the experimental group had the same instructor, same textbook, and nearly the same instruction. The Spring 2010 class had a non-game final project in which they had to derive equations of motion for a small motorized cart depicted in Figure 6, simulate the dynamics in Matlab, and choose parameters which would reduce the time it takes the cart to travel a fixed distance. Torque vs. RPM characteristics of the motor were given, along with mass properties, friction coefficients, and other parameters and constraints.

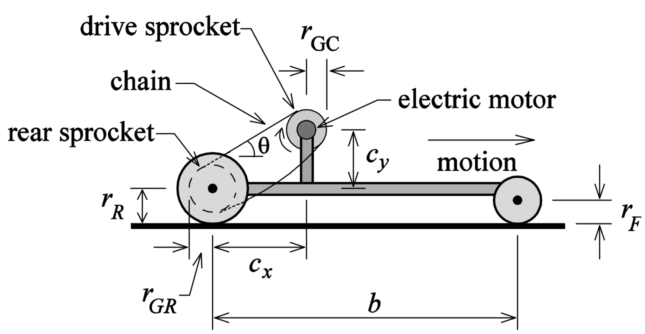

Figure 6. The non-game control group of students in Spring 2010 had to model and simulate a motorized cart for their final project.

As with the experimental group, the DCI was administered to the Spring 2010 students the week before their final project was due.

To compare test scores from the two different groups of students, I formed quantities

$$d = \frac{\overline{x_G} - \overline{x_N}}{S}.$$

Here, $\overline{x_G}$ is the average item score(s) from students taking the course in Fall 2010 with the game, and $\overline{x_N}$ is the same average over students in the non-game course in Spring 2010. The S is a pooled standard deviation. Therefore, d is a normalized difference between game and non-game groups; it is a Cohen effect size. A positive value of d would indicate that the game group performed better, on average than the non-game group.

For the pre-test administered at the beginning of the two semesters, we found $d = -0.24$ ($p = 0.30$, $N_G = 34$, $N_N = 46$). Therefore students who took the non-game class in Spring performed somewhat better on average, but given the p-value in the two-tailed t-test, the difference should probably not be considered significant. Furthermore, none of the individual items on the pre-test had differences that exceeded a $p = 0.20$ level of significance.

For the 29 items on the DCI post-test, I organized them into subject categories: Newton's 3rd law, particle dynamics, polar coordinates, rigid body kinematics, impact, friction, energy, momentum, rigid body kinematics, and rigid body dynamics. Here, we saw significant differences in two categories. For Newton's 3rd law (questions 1,14), I found $d = 0.56$ ($p < 0.01$, $N_G = 35$, $N_N = 42$). For rigid body dynamics (questions 11, 24, 27, 28), I found $d = 1.25$ ($p < 0.01$, $N_G = 35$, $N_N = 42$). In other words, students who learned rigid body dynamics with the aid of the game scored more than one standard deviation better. In social science research, a Cohen effect size of $d = 0.5$ is considered moderate, while an effect size of $d > 0.8$ is considered large[15,16].

In the other categories on the DCI, the differences were rather small. None of the others exceeded a $p = 0.10$ level of significance.

Closing Remarks

Results of our first experiment with the engineering dynamics video game are encouraging. In the primary outcome targeted by the game, rigid body dynamics, students learning with the game scored significantly better. The results are consistent with our previous studies of video game-based engineering education[9-11], and they encourage us to continue developing and refining *Spumone*. Given that *Spumone* appears to correlate with better learning outcomes, we also plan to start investigating research questions that focus on video game features that impact or hinder learning and motivation.

Acknowledgement

The author gratefully acknowledges support from the National Science Foundation under grant 0935225. Any opinions, findings, and conclusions are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

[1] Malone, T. W. Toward a theory of intrinsically motivating instruction, *Cognitive Science* 4, pp. 333 – 369, 1981.

[2] Gee, J. P. *What Video Games Have to Teach Us About Learning and Literacy*, Palgrave Macmillan, 2003.

[3] Aldrich, C. *Learning by Doing*, Pfeiffer, 2005.

[4] Federation of American Scientists, *Harnessing the Power of Video Games for Learning*, Federation of American Scientists, 2006.

[5] Shaffer, D. W. *How Computer Games Help Children Learn*, Palgrave Macmillan, 2006.

[6] Schell, J. *The Art of Game Design: A Book of Lenses,* Morgan Kaufmann, 2008.

[7] Fullerton, T. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*, 2nd Edition, Morgan Kaufmann, 2008.

[8] Brathwaite, B. & Schreiber, I. Challenges *for Game Designers: Non-Digital Exercises for Video Game Designers*, Course Technology, 2009.

[9] Coller, B. D. & Scott, M. J. Effectiveness of using a video game to teach a course in mechanical engineering, *Computers & Education*, 53, pp. 900 – 912, 2009.

[10] Coller, B.D. A video game for teaching dynamic systems and control to mechanical engineering undergraduates, *Proceedings of the American Control Conference*, 2010.

[11] Coller, B.D., Shernoff, D.J. and Strati, A.D., Measuring Engagement as Students Learn Dynamic Systems & Control with a Video Game, *Advances in Engineering Education* (in press).

[12] Hestenes, D., Wells, M., & Swackhamer, G., Force concept inventory, *The Physics Teacher,* 30, pp. 141 – 158, 1992.

[13] Hestenes, D. & Wells, M. A mechanics baseline test. *The Physics Teacher*, 30, pp. 159 – 165, 1992.

[14] G. L. Gray, F. Costanzo, D. Evans, P. Cornwell, B. Self, J. L. Lane, The dynamics concept inventory assessment test: A progress report and some results, *Proceedings of the 2005 American Society for Engineering Education Annual Conference*, 2005.

[15] Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*, New York: Academic Press, 1988.

[16] Cohen, J. "A power primer." *Psychological Bulletin*, 112(1): 155 – 159, 1992.