Teaching Physics by Designing Games

Peter M. Border*
School of Physics and Astronomy
University of Minnesota, Twin Cities

1 Introduction

We report on an experimental physics class which taught elementary mechanics by having students design computer games. Students made games with vPython classes and by writing “mods” for UnrealEd. They learned physics by programming agents to move, roll and collide in a physically corrrect manner, which required them to understand the physics behind the motion.

Animation and physics are natural partners, and physics education benefits immensely from being joined with animation. Thinking in terms of animation makes the physics much clearer, as it focuses attention on the differential equation rather than the integrated solution. Integrating numerically with a computer vastly increases the scope of solvable problems as well, opening up all sorts of effects that were previously ignored. Students are much more motivated to make a game or a movie than to do abstract physics problems, and they will work much harder on creative assignments than traditional ones.

Simulation is a much more robust and dependable method of problem-solving than traditional exact calculation. Real world problems are solved almost completely by simulation nowadays, unlike freshman physics where everything is done by exact integration. Visualizing results is also much easier with a computer than with pencil and paper. Looking at a graph, or seeing a simulated experiment happen, is much clearer than looking at an algebraic equation. This course uses computer visualizations as much as possible, and keeps the algebraic, pencil-and-paper manipulation to an absolute minimum.

2 The Course

The University of Minnesota offers several “freshman seminars” every year. These are small, two credit courses where faculty are encouraged to try out new ideas. Enrollment is limited to 15 freshmen. The schedule is two hours per week, with no laboratories. We tried three different technologies in the class: spreadsheets, vPython and UnrealEd (a product of Epic Games).

VPython is a product of Bruce Sherwood and Ruth Chabay at Carnegie-Mellon University (see the website at www.vpython.org). It is a set of classes for the python programming language that allow one to make a 3D world with moving objects very simply and easily. Unfortunately, a great deal was left out of vPython, such as textures, lighting, and so forth, and the students tired of it rapidly, so we also tried UnrealEd (a product of Epic Games). Objects in an Unreal world can have attached scripts written in UnrealScript, which is a near-clone of Java. Students wrote scripts that made objects move and interact realistically as the game progressed. Unreal is supplied with Karma, a built-in physics engine, but we instructed the students to ignore it.

The first assignment was to play a game of “Racetrack”, and to make a spreadsheet of the result. Players draw a racetrack on graph paper, then start their “cars” at the starting line. In each turn the cars are allowed to change their velocities by one square/turn horizontally and vertically. All students made successful graphs, and gained experience with frame-based motion and the basic dynamic loop.

The next four homeworks were done in vPython. The assignments were to “make something cool with vpython”, “make a game involving trajectories”, “spacewar!”, and “make a game involving collisions”. Spacewar in particular was a very popular assignment, as was the collision game (most people opted for some form of mini-golf or shuffleboard). The real standout projects were a version of Pong and an actual pinball machine (with flashing lights). Students learned about modelling physical situations with these projects, and eventually were quite good at it.

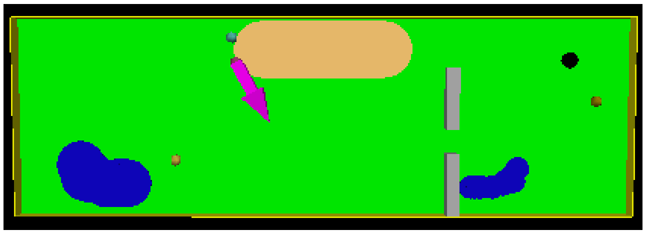

Fig. 1. A mini-golf game in vPython. The arrow shows the force in the next stroke, the blue and brown areas are water and sand traps, and balls bounce realistically when they collide.

Eventually we shifted away from vPython and began programming in UnrealEd. Mastering the interface took several weeks of class time. The physics taught with UnrealEd was 3D rotations, which is normally a subject for Junior level mechanics, but making pool balls roll around, collide, and bounce off walls was concrete enough for the students to understand. We simplified the physics by treating all moments of inertia as spherical, which is reasonable for a game environment.

3 Conclusion and Future Work

Our future efforts will be to offer the course again, and to continue refining it. Next year we will make better use of UnrealEd, and have our students start writing UnrealScript programs earlier. We are also collaborating with people at the Center for Creative Research at the Minneapolis College of Art and Design, and hope to produce some usable course materials with their assistance. There is a possibility of redesigning the Honors Physics labs to use simulation, and we will work on that as well.

*email: border@physics.umn.edu