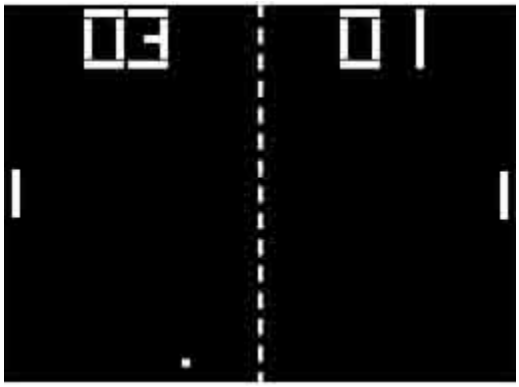
Mark Walker
Senior Project Proposal
Physics Engine in Java
October 3, 2005

1. For my senior project I propose to design and write a physics engine in Java from scratch. A physics engine is basically a part of a program (usually a game) that handles and simulates physics-related tasks. It is what calculates the movements of all the objects in a simulated world. For instance, in a simple game like Pong, a very simple physics engine is employed to determine the speed and direction of the ball, when the ball collides with a paddle or wall, and more advanced versions may track other variables such as the spin of the ball.



Pong

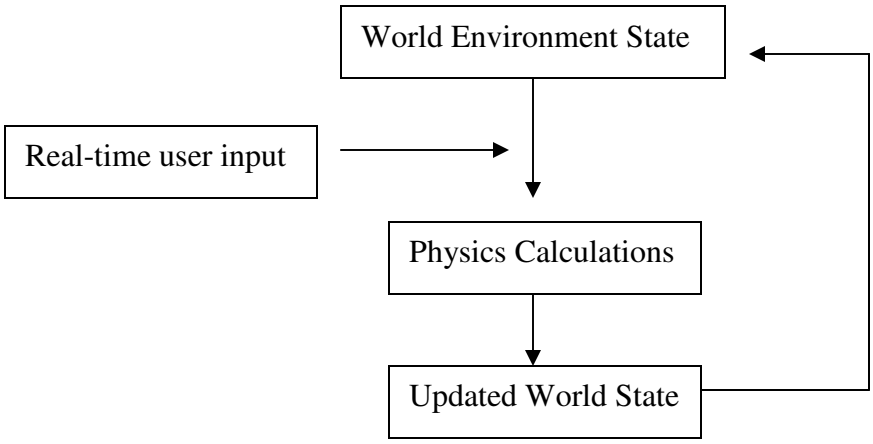
In more complicated games today, in particular 3D games, the number of parameters and functions that handle worlds are vastly more complicated. Instead of just speed and direction there are factors such as 3D rotation, advanced collision detection for irregularly-shaped objects, air resistance, constraints, ragdoll (human body) physics, explosion forces, etc.

Half-Life 2 physics demonstration

Because 3-dimensional programming would involve a working knowledge of both a 3D graphics engine and linear algebra, my own engine will only be in 2 dimensions. Having completed AP computer science last year, I will be writing it in Java because that is the language which I am most comfortable with right now. I want to avoid getting hung up on technicalities so that I can concentrate more on the important

aspects of the project.

The engine will be capable of handling a number of fundamental physical events, as well as providing an environment for the simulated world. Here is a rough overview of the workflow of the engine:



Elements that should be implemented into the engine include: basic kinematics and trajectories, rotational motion, collision detection between spherical and polygon shapes, elasticity, springs, rope constraints, air resistance, and frictional forces. One thing I would like to include that is not commonly done in 2D engines is collision between irregularly-shaped polygons. Generally simple shapes, limited to lines, rectangles, and triangles have been the limit of colliding shapes. This poses a unique challenge, especially when there are more than two objects in the environment.

As a final product I plan to develop a program specifically designed to demonstrate the capabilities of the engine. It will allow the user to play with various aspects of the engine by being able to draw in elements such as shapes, balls, ropes, etc. into the window. This will also be a program I will use to test and debug the development of the engine itself. I do not want to spend a lot of time on anything other than the physics engine, so I may use someone else's simple graphics and interface engines/programs instead of writing my own from scratch to save time. If I feel compelled to, I may design a game that would also demonstrate the capabilities of the engine, though it is likely that I may not have time.

2. The reason I chose this to be my senior project is because I wanted to be able to implement systems from outside the realm of computer science into a program. There has obviously been a lot of work done on physics engines, and since I have taken accelerated physics and am in advanced physics now, I have ample knowledge to apply these systems into an engine with the help of a physics consultant.

3. In college I hope to narrow in on an interdisciplinary field, which will probably involve computer programming. This project is an ideal way to introduce myself to the challenges of translating one area of knowledge (in this case physics) into

computer science.

4. Bill Adams has agreed to be both my primary advisor for the project as well as computer science consultant. Brian Giannino-Racine said he would be willing to be my consultant for physics.

5. Weeks 1-2: Research and plan program architecture; find graphics engine
   Weeks 2-3: Develop necessary supporting classes; complete graphics engine and interface.
   Weeks 4-5: Develop basic point mass physics simulations.
   Weeks 6-7: Solid body simulations.
   Weeks 8-11: Solid body collision detection and handling.
   Weeks 12-13: Rope constraints, springs.
   Weeks 14-15: Elasticity, (non-solid objects?).
   Weeks 16-17: Complete any unfinished components.
   Weeks 18-20: Finish demonstration program, (write a game?).

6. I will be using three textbooks among various internet sources: *Series in Interactive 3D Technology: Game Physics*, by David H. Eberly; *Series in Interactive 3D Technology: Real Time Collision Detection*, by Christer Ericson; *Physics for Game Programmers*, by Grant Palmer. The first two books are fairly advanced and include a lot of linear algebra, but they also include a lot of important information about applications of simpler vector math as well as overall concepts. The third book is easier to understand and not as complicated but still contains a lot of very useful material.

7. The biggest risk of course is running into some sort of fundamental problem in my coding and having to redo a lot of work. I am going to have to spend a lot of time on this project because of the ambition of my goals, but with the help of Bill and Brian I don't think there is a problem we will run into we can't solve.

8. The only costs associated with the project were the textbooks, which I have already purchased.

9. I will drop American Pop Culture for the project. That will leave me with the project, Ship of Fools, AP Harmony, Advanced Physics, and Advanced Statistics (and a music lesson).