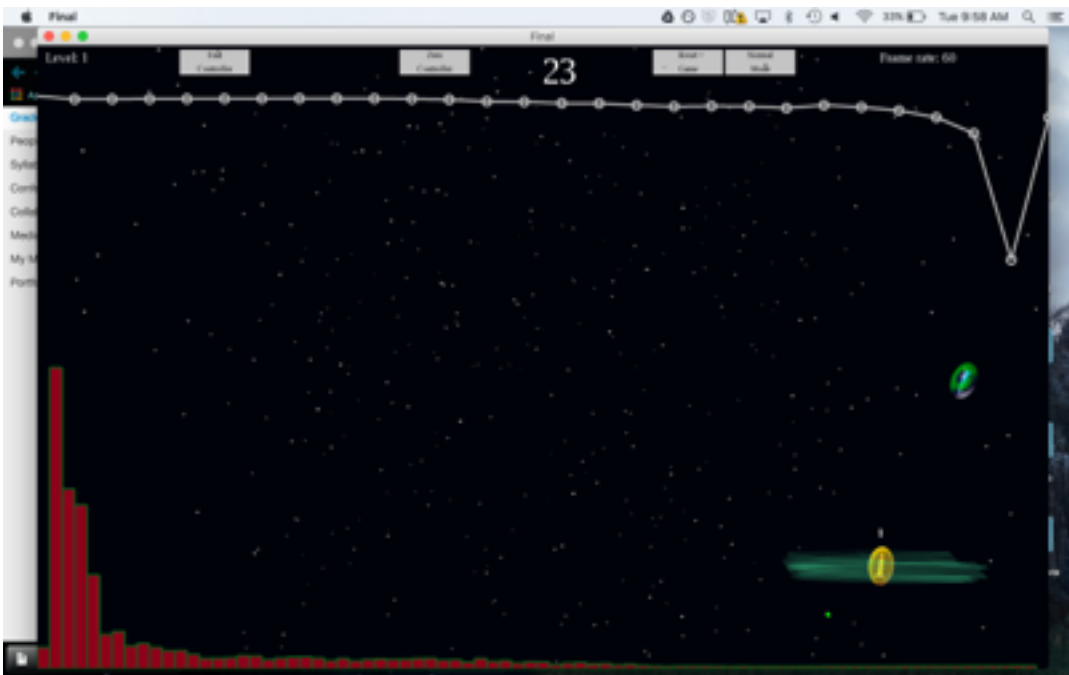
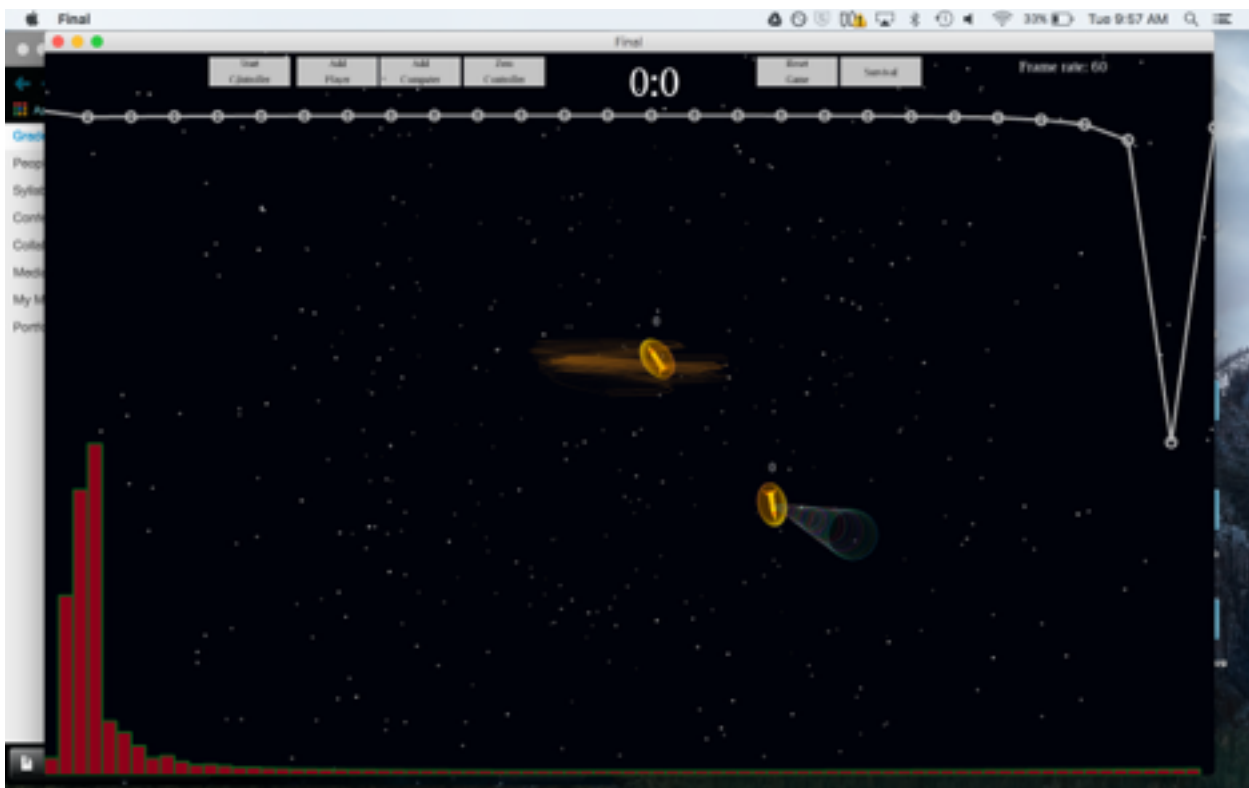


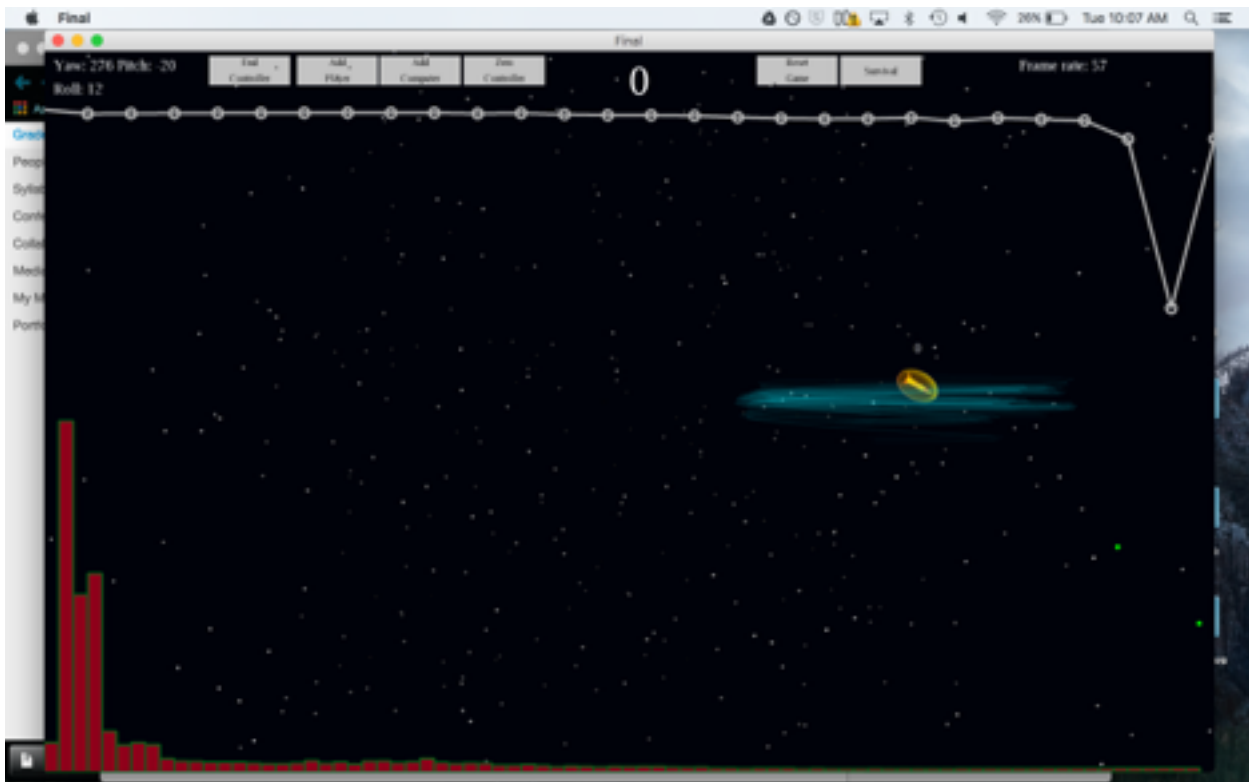
The rocket in the middle is controlled by the arrow keys, and it has shot twice, which are the two green dots. The bars on the top and bottom are visualizers of a song playing in the background.



This is a demo of survival mode, in which you try to survive as long as possible with increasing number of AI enemies.



This is a demonstration of the multiplayer mode. The second player is controlled by WASD and E to shoot.



The rocket is now controlled by a combination of accelerometer and gyroscope data by using the controller. It points in the direction of tilt and shoots and accelerates using the two buttons.

### Conceptual Note

I was inspired by the games Rockets, Rockets, Rockets. It is an extremely strange game in which half the challenge is just trying not to crash because of the crazy controls. My game has two modes: Survival and Normal. In Normal, three extra buttons appear at the top, so you can add as many ai rockets and up to two player rockets. In Survival, it is a single player game with an increasing number of enemies. Each player has a character identifier following it. I put these to prevent confusion between players, and I think they add to the overall trippy-ness of the game, so I decided to keep them in permanently. The two visualizers on the top and bottom are mirrors of each other. They are both adjusted for the logarithmic hearing scale of the human ear. The rockets can collide into each other and the visualizers. If they go too far right or left, they appear on the other side. There is also a controller that comes with this game. It is a laser-cut piece with two buttons and an IMU(Inertial Measurement Unit). The program running on the Arduino inside of it takes accelerometer, gyroscopic, and magnetic data and sends understandable output via serial.

### Project Reflection

I came at this project with too many ideas and too little time. I removed a couple of ideas entirely, and used other people's implementation(Crediting them) for some others. I also ran into the problem of Arduino's real barebone-ness. I had to optimize an example sketch to fit onto the board. Then I had to rewrite its output format, as it was completely corrupted. I also had trouble with using ellipses, as the math gets difficult really quickly, but it helped when you and Thomas said that asking whether two overlap is different from finding intersection points. Overall, I ran into too many problems really to count. I'm sure I'm forgetting some major ones. All in all, though, this project was a great cumulation of everything I've learned in this class.

## Help from

Arduino help. Especially of code hanging

<http://www.i2cdevlib.com/forums/topic/113-mpu6050-dmp-sketch-hangs-sometimes/>

<http://cdn.akamai.steamstatic.com/steam/apps/289760/header.jpg?t=1447361012>

Stars code and example

[https://www.processing.org/discourse/beta/num\\_1209965886.html](https://www.processing.org/discourse/beta/num_1209965886.html)

Reading file names from folder

<https://forum.processing.org/two/discussion/1747/reading-filenames-from-a-folder>

Wave/circles code and example

<https://forum.processing.org/one/topic/new-to-processing-and-need-help-with-ripple-effect-urgent.html>

Explosion animation

<https://sebastiencourtois.files.wordpress.com/2011/08/explosion.png>

Rockets pictures

<http://cdn.akamai.steamstatic.com/steam/apps/289760/header.jpg?t=1447361012>

Background audio

Songs from <http://www.bensound.com/royalty-free-music/corporate-pop>

Also Thomas Marshall and Buzzy Barrow for playing and suggestions.