APCS2 Final Project Proposal

SEJ Fault — Period 9
Shantanu Jha, Elias Milborn, James Wang
May 21, 2016

Overview

For our AP Computer Science Semester 2 final project, we have decided to make a PC version of the popular mobile multiplayer game Astro Party. We plan to use Processing and the various skills we learned over the course of this year and especially this semester to help us in our endeavor. We have a detailed Road-Map outlining the different stages of our project development. There are many different features in the game Astro Party, including: two button controls, "Pilot Execution" vs "Head Hunters", a variety of Maps, and extensive game physics. Here is a link to the original Astro Party: http://rustymoyher.com/astroparty/.

1 Fundamentals of Astro Party

Astro Party is a two button multi-player game. The basic game objects that the user controls in Astro Party are ships. Ships are able to turn, shoot, automatically move forward, dash, and use a variety of power-ups that can be picked up in a game. One button allows users to turn their ship, while the other allows them to shoot bullets.

There are two main modes of Astro Party. The first is called "Pilot Execution". In "Pilot Execution", a ship hit by an dangerous game object, such as a bullet, lancer, laser, or bomb, will be disintegrated, but will leave a pilot in its place. The pilot will float in the game field, and the two buttons used to shoot and turn the ship can now be used to propel the pilot forward, and turn the pilot. The pilot can be killed by any dangerous game object, or if another ship rams into it. After a brief period of time, if the pilot remains alive, the ship will regenerate around the pilot. The other mode of Astro Party is called "Head Hunters". In this mode, if a ship is attacked by a dangerous game object, then it will automatically disintegrate and the player whose ship was killed will be out for the round.

Some essential game objects in Astro Party include: ships/pilots, grey asteroids (of varying sizes), orange asteroids (containing power-ups), bullets, power-ups (laser, bomb, lancer, reverse), map objects (turrets, anti-gravity wells, breakable walls, solid walls, game boundaries, etc), and more. Grey asteroids are free floating obstacles that can be disintegrated by bullets, lancers and lasers. Grey asteroids can come in several sizes. Orange asteroids are also free floating, and work in similar ways to grey asteroids; however, orange asteroids contain power-ups that can be picked up by ships that pass by close enough to them.

The objective of the game is to get enough points to win. There are short, medium, and long matches. In short matches, it takes only 1 point to win the game; in medium, 3 points; in long, 5 points. The way a player gains points is by killing another player (in "Pilot Execution" this means killing the pilot, and in "Head Hunters" this means killing the ship). If you kill yourself, you will lose points, unless you don't have any.

There are many other nuances to this game, some of which we will explain later on in this proposal. Regardless, through developing this game, not only will we be able to apply the knowledge and skill-set we gained over the course of this semester and year, we will also learn a plethora of new skills and improve our proficiency in Computer Science and Java as a whole.

We plan to integrate many concepts that we've learned over the course of the semester while creating this project. First, we will make heavy use of Processing since this will be a graphical application. Additionally, since responsiveness is very important in a real-time multiplayer game, it is vital that we maintain an efficient algorithm for managing the in-game physics.

2 Power-Ups

- Lasers: one shot power-ups, can sweep and area if the player is turning whilst using them. These power-ups can go through everything, including solid walls, and disintegrate everything in their path except solid walls
- Lancers: two light sabers attached to the side of the ship. These power-ups go away after hitting and disintegrating a certain number of game objects
- Reverse All: reverses the direction that all ships turn (original direction is turn clockwise)
- Bombs: They can be triggered by players (that did not drop them) who wander into a certain radius of the dropped bomb. These power-ups can also be set off if they are hit by a bullet.

3 Maps

- Anti-gravity Wells: Accelerate players away from their centers
- Breakable Walls: Players can disintegrate these walls with lasers, bullets, bombs, or lancers
- Solid Walls: Form the perimeter of any map, and also define unbreakable boundaries inside maps
- Open Walls: Players can loop from one side to the opposite side through open walls
- Beams: Alternate on and off, and these laser beams from across maps and can disintegrate whatever crosses their path
- hidden blocks: translucent blocks that cover all game objects and appear in only one type of map

4 Features Overview

Main Features:

- Basic movement: turning, shooting
- Game Logic: killing, boundaries
- Game Physics
- Power-ups
- Multiple players
- Pilot Execution
- Pretty graphics

Additional Features:

- Slightly different game modes/maps
- Gravity map
- Auto-balancing: giving shields and other power-ups to players that are losing very badly in a game
- Other dynamic maps
- Settings menu, e.g key-configuration
- Other features maybe?

5 Development Tools

Some tools we will use:

- Github
- Java
    - Processing

6 Roadmap - Stages of Development

1. Moving Ship with a turning button
2. Square Map with simple boundary, and boundary logic for moving ship
3. Adding bullets to the ship, and adding free floating grey asteroids of different shapes to the simple map
4. Adding another ship with the same abilities as the first ship. Adding shooting logic, and kill logic. Both of these ships compete in the square map. Ships will not disintegrate into pilots at this point. They will, however, die. (this is the "Head Hunters" mode
5. "Pilot Execution": this includes movement of pilots, kill logic for pilots, and regeneration. This will be a crucial step in the roadmap
6. We will then add some power-ups such as lasers, bombs, lancers, and reverse alls. This is all still happening in our simple square boundary map.
7. Then we will add orange asteroids to hold power-ups, which would be able to be picked up.
8. We will then start to add map objects such as breakable walls, open walls, beams, etc.
9. We will then create separate maps.
10. We will then add up to 4 players in the game. We will figure out the bugs, nuances, and other problems that arise in the game at that point.
11. We will continue to include more elaborate features, if we have time, such as: anti-gravity wells, turrets, auto-balancing, better graphics, better user-interfaces. We will fix any and all bugs that come up, and continue to develop some other cool feature when they come to mind.