

Team Java Project Specification, Group B3

Michal Staniaszek

January 26, 2011

1 Functionality

The project we intend to work on is a scrolling space shooter. The player controls a ship which can move around the screen in any direction and shoot. The player's ship cannot leave the screen, but can move anywhere otherwise. Enemies are spawned at the top of the screen and move towards the bottom. The player must destroy the enemy ships without getting destroyed by enemy fire or by colliding with enemy ships. Destroying ships grants some points. There will be some sort of way of adding a multiplier to the points that the player gets, for example, destroying a certain number of ships without any leaving the screen will increment the multiplier. At the end of each level, or after a certain time, a boss enemy with a significant amount of health will appear. Defeating the boss will grant a large number of points, and allow the player to continue, but will be harder to kill than the standard enemies. The game will allow for a second player to connect over the network and control a similar ship to the same ends. In the multiplayer mode, enemy health will be scaled in such a way that it is still challenging — if all enemies were to have the same amount of health as they did for one player, they would likely die twice as fast with two players playing.

1.1 Features

- Simple shapes to represent enemies and player
- Scoring (with multipliers?)
- Difficulty ratings
- Endless mode
- Various weapon types for player and enemies
- Multiple enemy spawn points
- Boss enemies
- Power ups

2 GUI Design

The GUI will start with a menu which will allow the player to check high scores, select different game modes, edit options and controls, and connect to a game that is already in place. The main game screen will contain information about the number of lives the player has remaining, and their current score.

3 Release Plan

The release plan that we have currently is up to the first release of the game that could be a potentially completed product. Releases will be numbered in steps of 0.1, starting from v0.1. In each release stage, we will attempt to complete a certain part of the project, such as movement,

shooting, enemies, menus, etc. Once we have reached a version that could be submitted as a final version, we will mark this as the alpha version. We will then build upon this version, adding more complex parts to the GUI and gameplay. If we finish all the simpler parts that we have planned, we will move on to adding some kind of extensions.

4 Software Engineering and Testing

4.1 Software Engineering

The approach that we will be using will be similar to the agile process. Each release, we will have some specific things that we want to implement. Once we have implemented and tested these, we will move on to the next release, with things to add to the previous release.

4.2 Testing

Our testing will be based on JUnit tests. This will allow our tests to be automated, which will be good once we have a lot of code to work with. We will not be enforcing any specific kind of testing strategy, but we will expect that each member's code is tested in ways that check that the code actually does what it is expected to do.

5 Networking

To connect to another game, a player will be required to type in the ip address of the other game into a text box. This ip will be processed, and then attempts will be made to connect. The network will most likely use a client server method, where the player being connected to is the server, and all others are clients. We are unsure of how many additional players to allow, but the final number will most likely be a maximum of 4 players per game. Probably the best way to work with the networking would be to send the locations of all objects on the screen to the client, which will respond with its location on the screen, and the locations of any weapons it has fired. The server would be in control of how enemies move, and updating the locations of all objects once it has received the locations from clients. This is still to be decided, though.