

Team Java - Specifications document (rough draft).

Introduction

The aim of our project is to create working game with multiplayer capabilities. The game itself will be a sidescrolling spaceshooter, much like the old games from Amiga/DOS times. We will work in a group of 4 and will have 11 weeks to complete the project.

Functionality

We have decided to make the game sidescrolling rather than static, for it should make it seem more lively and vibrant and be easier to do in the end, as we might be able to avoid any hard trigonometry. The game will be a top-down scroller rather than a left-right one. The player's ship will be able to move freely along the x- and y-axes, the exception being the boundaries of the game screen that the ship will not be able to pass. It will point upwards always shooting in the same direction, along the y-axis.

The opponents will spawn at the top and move down towards the bottom. If they reach the bottom they vanish. They will shoot in random or set patterns and the aim of the game is to avoid these bullets/lasers/etc and the ships of the opponents themselves, for collision with them will cause loss of life aswell - while shooting down as many of them as possible to gather score. The player's ship will be destroyed from a single collision, where as some of the enemies will need multiple hits before they are blown to smithereens, which results in the game ending. To win the game, you just need to get through the set amount of spawns, so basically survive with the highest score possible.

The controlling of the ship will be with a keyboard, most likely with the arrows or WASD setup, shooting with spacebar. We will normalize the movement so that the ship does not move quicker if going diagonally. The feeling of movement will be implemented by creating a camera that moves constantly at a certain pace. There will be a title screen where you can choose your gamemode and if you see the high scores or quit.

We have come up with a list of extra things we might add when the basegame is working with networking implemented:

- Powerups
- Boss fights
- Levels
- Different weapons
- More advanced movement for the enemy
- Etc.

GUI Design

The GUI will be really simplistic. For the graphics of the ships and ammunition we will just use shapes. There will be a score counter placed at the bottom/top and the amount of your lives/tries left. The other GUI part of the game will be the title screen, where we will just have buttons for singleplayer, multiplayer, highscore and quit. The game will be made so that we can scale everything, so that the user can choose the size of the window he wants to play in or if he wants to do it in fullscreen.

If we have time and gain access to any sort of sprites, we might change the shapes into them.

Release Plan

V0.1 – Base Code

Ship rendering properly and movement working smoothly.

V0.2

Basic shooting from the player's ship.

V0.3

Static enemies rendering properly, collision detection and score counter working.

V0.4

Simple movement for the enemies.

V0.5

Background rendering nicely and giving the illusion of movement, ie. camera movement. Opponents now spawn.

V0.6 – Base Game

Networking working smoothly and properly.

Software Engineering and Testing

For testing purposes we have decided to use JUnit and we will try to write some tests before we write the actual code for it, so that we would not engineer the tests after the code. Testing the game will be done at a regular quick interval, to ensure everything is working as it should and that we are making progress.

We are going to use software engineering in the manner that we will document everything carefully and plan out our class hierarchies and draw diagrams where necessary. It is hard

for us to utilize some of the SE tools however, for we have no idea how to assess how long something takes to code for any of us, so making any kind of critical path/arrow-on-activity is hard in the normal sense. We will plan out when things need to be done and the proper order of doing them. What is dependent on what and so forth.

Requirements is an important part of any project and we have decided to use time on trying to think up every possible problem we might have with the project and how to solve them so that we would minimize the surprises and changes of the project.

Networking

As for networking, we have had some trouble defining what needs to be in technical detail, as we do not really know how to code it or how it works. The game will work on a client-to-server basis over TCP/IP most likely. Choosing multiplayer option, you will have a box where to input the IP of the host player and the connection will be made. At first we will allow only one other player to play, but we might up the number later on, however it might make the game get too crowded for it to be enjoyable in the first place.