# Team Java Specification

## Functionality

- The game itself will be space shooter that will have enemies come in from the top of sides of the screen while the player is situated at the bottom (but can move freely left to right and up and down). The game type is called a side scrolling shooter as the background will be moving to give the illusion of movement.
- The game will be played using the arrow keys or the mouse for movement and clicking or space for shooting.
- If enough time is left after the base game has been created the side scrolling can be expanded upon to move in a different direction (be it scrolling down or left) or go for an "asteroids" style gameplay for some levels.
- The graphics for the game will mostly consist of shapes created with the Java API (such as triangles etc)
- The game will incorporate collision detection for when the projectiles hit the player's ship or the enemy ships as well as when ships collide.
- The game will involve a scoring system with the player earning a higher score the more he shoots down the enemy ships.
- If possible, once the base game has been created difficulty levels can be incorporated into the game.
- The enemies will have set spawn points and will spawn in patterns that will vary throughout the time the player plays the game.
- If possible and have enough time at the end Power ups will be included into the game to make it more varied and fun, these power-ups will change things such as weapon types with different damages and spray patterns. Also super weapon type power-ups could be included.
- Again, if there is time left at the end we will add in boss fights to the game to make it more challenging and exciting to play.
- A menu will be made that has buttons for Options, Help, Single player and Multiplayer. The options menu will allow the user to decide if they want to use the mouse or not.

**GUI**
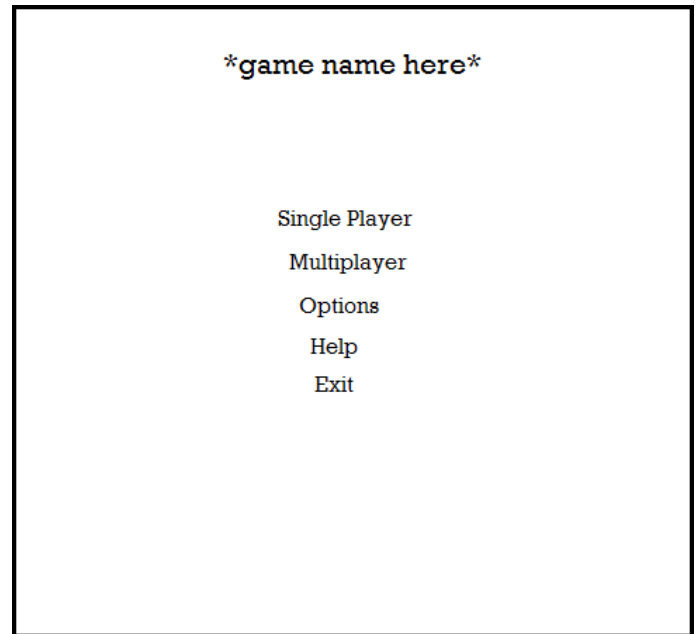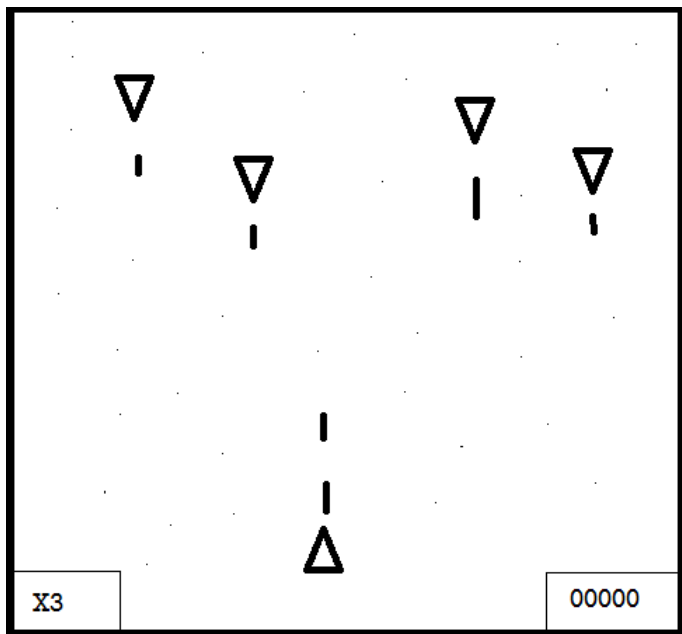


The GUI will be fairly simplistic. Most of the graphics for the project will be made up of shapes. The screen will have a score and life count listed somewhere on it.

We will also create a menu screen that will have buttons that allow you to select different things such as single player or multiplayer.

The screen should scale (including layouts of ships etc) with the screen size for anyone who wants to change from the default screen size.

If we have time we may replace the shapes with dedicated sprites.

Above are some quick mock up ideas for the menu and game screen.

**Release Plan**

**V0.1 – Base Code**

Ship on screen

Movement/Controls


**V0.2**

Shooting from Ship – Simple straight up


**V0.3**

Enemies –Static

Collision detection

Score


**V0.4**

Enemies –Simple movement


**V0.5**

Background

Spawn points


**V0.6 – Base Game**

Networking

## Software Engineering practices

For this project we will be using JUnit for testing as it will help make an automated testing routine which will keep everything fair and standardised when testing. Once the base code (v0.1) is down we can start testing the game properly to check that it's making progress and is going in the direction we want it to.

The main software engineering method we will use along the way for the project will include using the incremental software life cycle with us documenting the progress made each stage and once the goal of the stage is met, move on to the next stage.

We will also branch off the main trunk every so often to make up a quick prototype to see how well it works when it comes to being used by real people (rather than it being all tested by automated tests).

## Networking

For Multiplayer play networking will have to be used to link the games together. We decided upon using a "drop in" method where someone starts a game and someone can then just connect to the game by using the IP of the host though typing it in a popup box that asks for it (after clicking on multiplayer and "connect to".

The networking method we think the game will be use is a server –client type where a player starts a game and hosts it and then other players can then connect to that host.

As we haven't done much game networking in the past we can only guess at what sort of things we will need to send across the network. The sort of things we think will be sent over the network are the enemy positions, the player positions and the projectile positions.

We hope to have a game that can incorporate 2-4 players in multiplayer mode.