Mobile Games Assignment 2 : Game Implementation.

Ryan Hood.

P1617694X.

Design Document.

Introduction.

In this design document I will talk about the aim of the project alongside the target platforms and audience for the game. I will continue onto a brief outline of the game mechanics with a section for a storyboard which will display the general flow of the game. I will conclude this document with a some more detailed diagrams in the form of a flow diagram which will detail how the game will move between the different stated and will show how the player will navigate as well as some UML diagrams to show the different objects within my game; how they work, what they do and how they interact with other game objects.

Aim.

The aim for this project is to create a 2D clone of the popular game Galaga using Android Studio. I aim to use this project to gain a better understanding of both Java and how Android Studio works.

Target Area.

After conducting some research I have decided to make the game for phones of version 4.1 (Jelly Bean) or higher, this is because an average of 2% of people use phones with a version lower than Jelly Bean.

(src:- <http://support.visiolink.com/hc/en-us/articles/207516149-Android-Minimum-supported-version-is-now-Android-4-1-Jelly-Bean> )

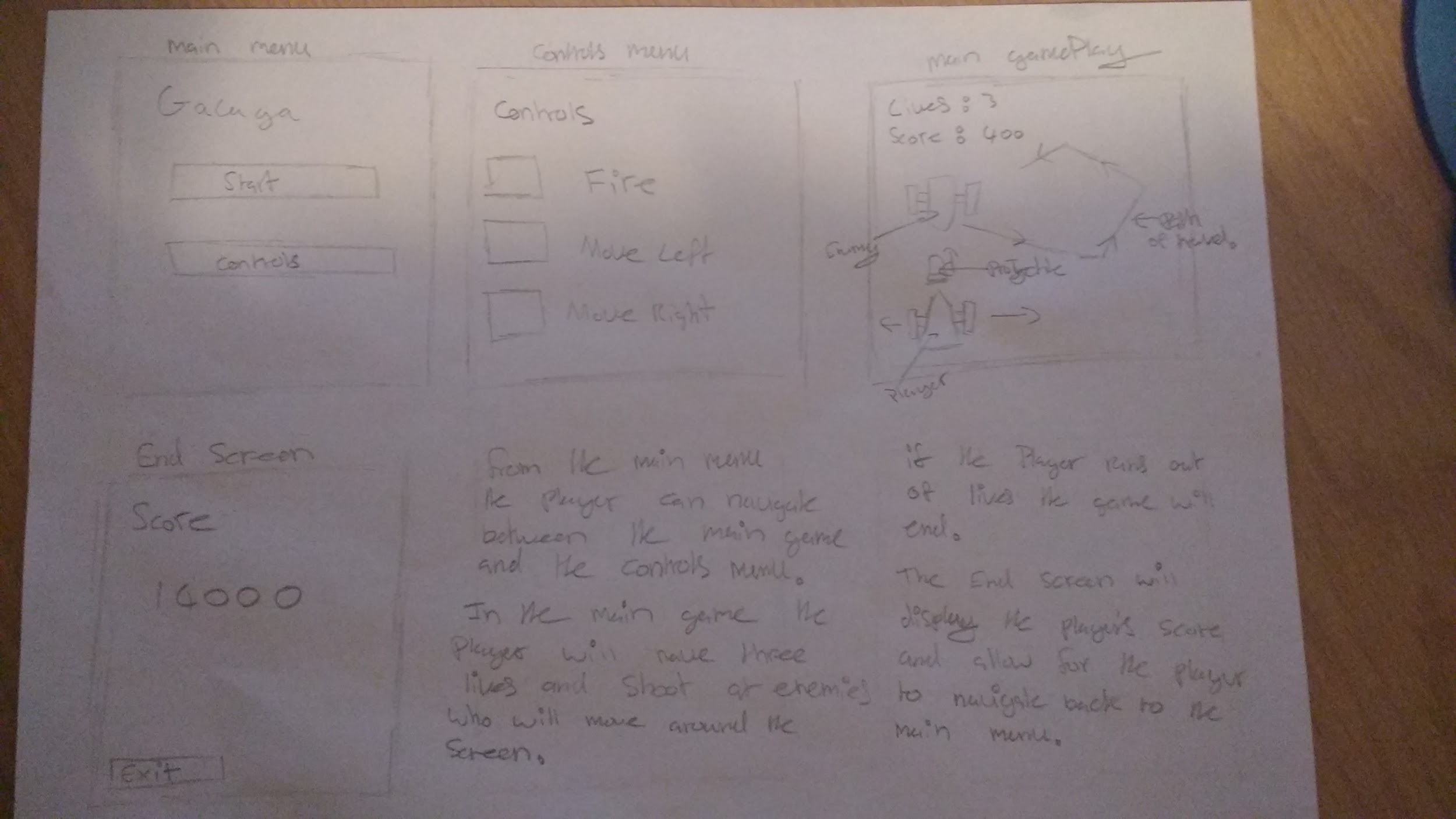
As for a target audience the game should be aimed at the older demographic of players, people who might have played the original and will feel nostalgia for the recreation; thus making it more likely for them to continue playing. However because if this it would be beneficial to make the UI and gameplay similar to the original. An issue with this is that it may cut off the younger demographic so making the game more modern in terms of gameplay and style might be a thing to consider. In conclusion I aim to make the game have modern, yet retro style graphics and a faithful recreation of gameplay and mechanics and a complete overhaul of the UI to make it more intuitive and responsive for use on a phone.

Game Mechanics.

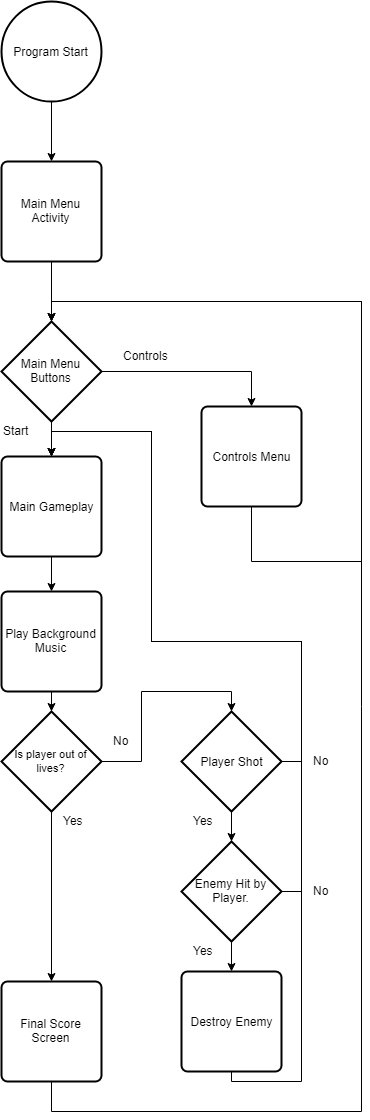
Player : The player will be confined to the bottom of the screen and will be able to move either left or right on the screen to avoid both enemies and their projectiles. The player will have a maximum of three lives at one given time and will lose one of them if they are hit by anything. The player will earn points every time they shoot and destroy an enemy and this will be added to their high score.

Enemies : Enemies will fly around the screen attempting to avoid the player shooting at them, they will sometimes stop at the top of the screen in a similar fashion to that if Space Invaders and will fire at the player before flying around the screen again. The enemies will occasionally shoot either from a stopped or flying position. There will be a couple of differing enemies with slightly differing sprites and colours, and matching this some of them will be faster than others and some might be able to take more than a single hit.

Storyboard.



Flow Diagram.



Once the game is launched the it will load the main menu activity which will allow for the player to navigate to navigate between the controls menu and the main game. From the controls menu the player will only be able to go back to the main menu.

Once the main game starts it will loop as long as the player has more than a single life; it will also wait for the player’s input to see weather the player needs to shoot. Also if an enemy is hit by a projectile and they have less than their max health they will be destroyed and removed from the game. If at any point the player reaches zero lives then the game will end. At the game’s end a final end screen will be displayed, it will prevent the player from going back into the game and will allow for the player to get back to the main menu; the end screen will display the player’s current score.

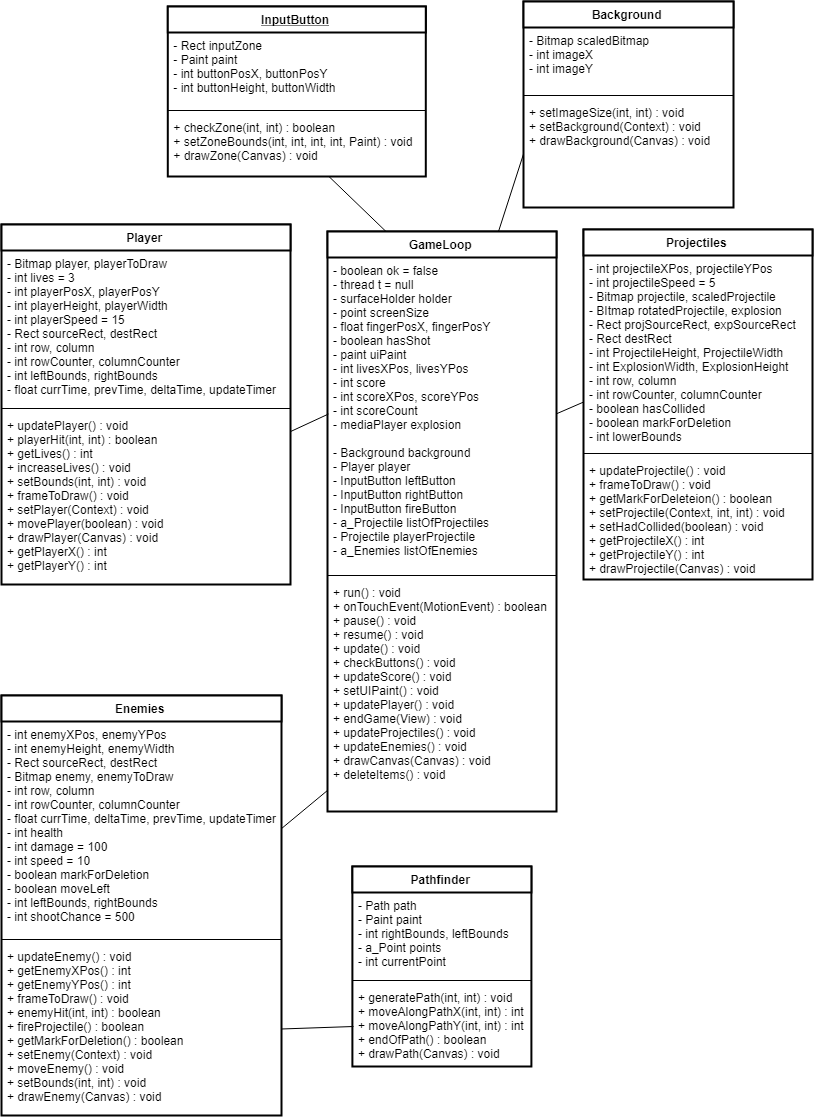
UML Diagrams.

In this game there will be several different classes each will do different things in the game; the most important class in the game will be the game loop class, it will be used to both run the game loop and handle all of the drawing to the game’s canvas. The next two are quite similar the player and the enemy. They will contain similar functions one to fire a projectile and another to draw the game object; both are critical to the game.

The projectile class will be used to fire from both the player and the enemies with minimal changes, the main way I aim to implement the projectiles is by having an array of projectiles.

Input button will be used to allow for input to be detected by passing in a set of coordinates and checking if they are within a rect; the input button will be able to take a size and a position on the screen for where it will be placed allowing for multiple different instances to be created and thus allowing for more than one working button.

The path finding class will be used to allow for a gameobject to randomly move along a set of points.



Conclusion.

In this design document I have outlined the aim of this game as well as its target audience. I have outlined the main game mechanics and provided a basic storyboard for the game showing a brief description of how the game will look. I have also designed and created a set of UML diagrams explaining what classes will be in the game and outlined what they will do. I have also made a simple flowchart showing the flow between the different activities in the game and what choices will need to be made during the gameplay.