**Faculty of Science and Technology**

**2017/2018**

**Level 4**

**Object Oriented Games Programming**

**2D Video Game Program**

**Analysis, Design, and Implementation**

**Report**

**1. Game Concept (Marks: 5%)**

The game I have made is a space invader inspired space shooter, the player has to move from left to right firing at the ships while dodging them as they fall. Destroying all the enemies allows the player to win the game.

The genre of my game would be “Shooter” as the player primarily has to shoot enemies to win the game.

The purpose of my game is to allow a player to be able to pick up and play my game without any practise and still have a good chance at winning, all while having fun trying to kill the enemies as fast as possible.

The game is played using the AD and Space keys which allow the player to move left and right and fire, the Escape key is used to exit the game and the mouse can be used to control the on-screen buttons.

One of the main game mechanics is the players controls, by pressing AD and space, the player can manoeuvre themselves around to shoot the closest enemy to them.

Another mechanic is the enemies, they begin to fall when one dies and their speed gradually increases as more and more die, this makes the game harder for the player as they have to change focus to enemies closer to the bottom of the screen, enemies that touch the player or the bottom of the screen end the game.

**2. Analysis (Marks: 20%)**

# Module – Window Manager

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| N/A. | * Set the window title, width, height and load all textures/fonts/objects related to the window * Create the window * Handle the game loop * Communicate with the State Manager * Communicate with the Player class * Communicate with the Enemy class * Communicate with the Audio Manager * Communicate with the Bullet class * Communicate with the Event Manager * Handle destruction of the window | * Returns when either an error occurs or it successfully executes. |

# Module – Entity Manager

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| * Entity texture location * Entity positions * Entity scale * Window object | * Initialises entities with a texture * Set the entities position * Get the entities position * Set the entities scale * Draw the entity * Check the entities collision | * Entities position   Whether the entity is colliding with the window border |

# Module – State Manager

|  |  |  |
| --- | --- | --- |
| Inputs | Processing | Outputs |
| * State integer | * Create a GAMESTATE enum * Set the initial state to “Menu” * Set the state to the inputted state * Get the current state | * The current state |

# Module – Event Manager

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| * Window object * Player object | * Check for window closure * Check for “Space” key pressed | * Whether the player is firing or window is closing |

# Module – Audio Manager

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| * Background music file * Gunshot sound file * Death sound file * Is the player firing * Is the enemy dead | * Play the background music when the game begins * Play the gun shot when the player fires * Play the death sound when the player or enemy dies | * Background Audio * Gunshot audio * Death audio |

# Module – Player

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| * Location of player texture file * Player position * Player scale * Whether the player has won * Whether the player has lost | * Pass the texture file to Entity Manager to set the texture * Pass the position to Entity Manager to set the position * Pass the scale to Entity Manager to set the scale * Check whether the player has pressed either the A or D key * Check whether the player collides with the window border * If the player wins then change state * If the player loses then change state | * Player object on screen * Player movement * Win or Loss screen |

# Module – Enemy

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| * Location of Enemy texture file * Enemy position * Enemy scale | * Pass the texture file to Entity Manager to set the texture * Pass the position to Entity Manager to set the position * Pass the scale to Entity Manager to set the scale * Check whether the enemy collides with the window border * Spawn 10 enemies and add them to a vector * Set the texture of each enemy within the vector * Move the enemy down the screen slowly * Handle enemy collision with the player * Handle enemy collision with the bullet | * Enemy object on screen * Enemy movement * Death of the player of Enemy |

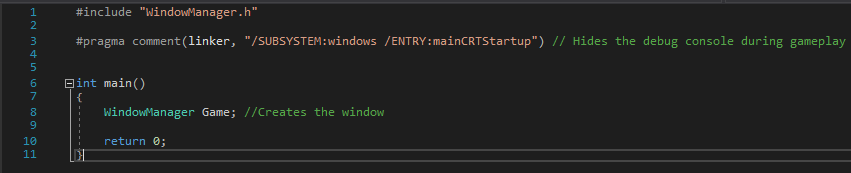
# Module - Bullet

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| * Location of Bullet texture file * Bullet position * Bullet scale | * Pass the texture file to Entity Manager to set the texture * Pass the position to Entity Manager to set the position * Pass the scale to Entity Manager to set the scale * Check whether the bullet collides with the window border * Move the bullet slowly up the screen | * Bullet object on screen * Bullet movement * Destruction of the bullet |

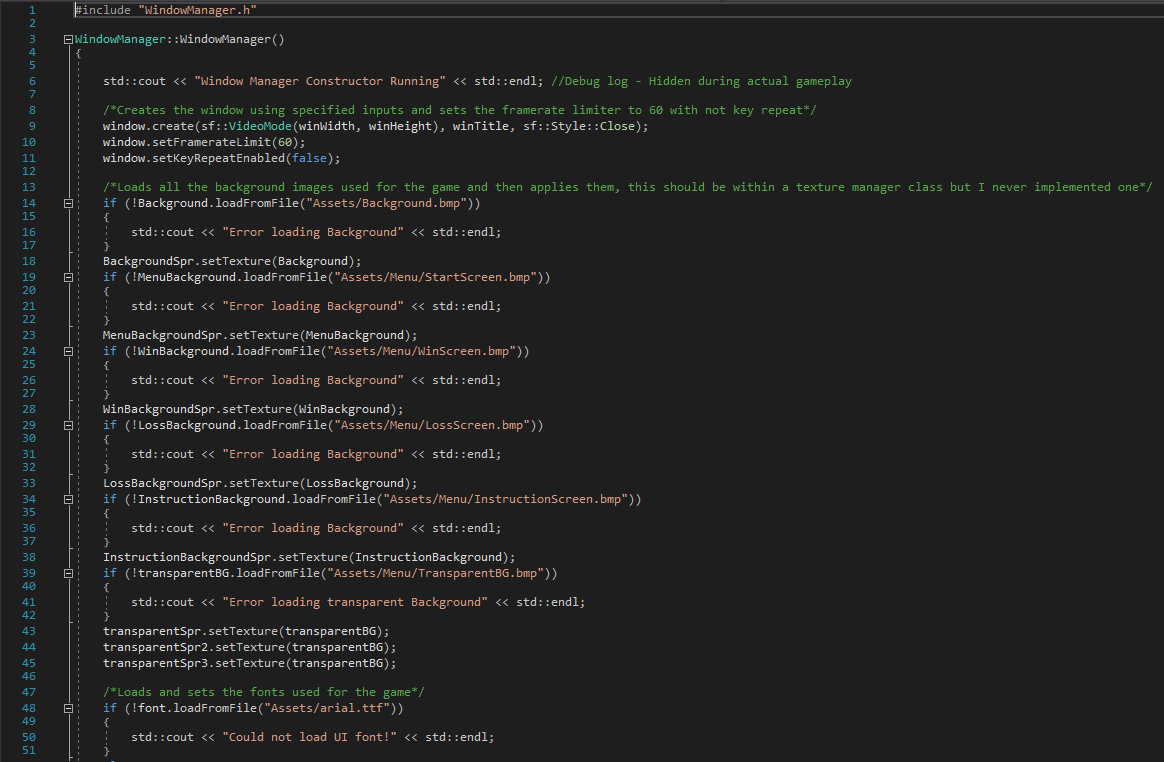
**3. Design (Marks: 25%)**

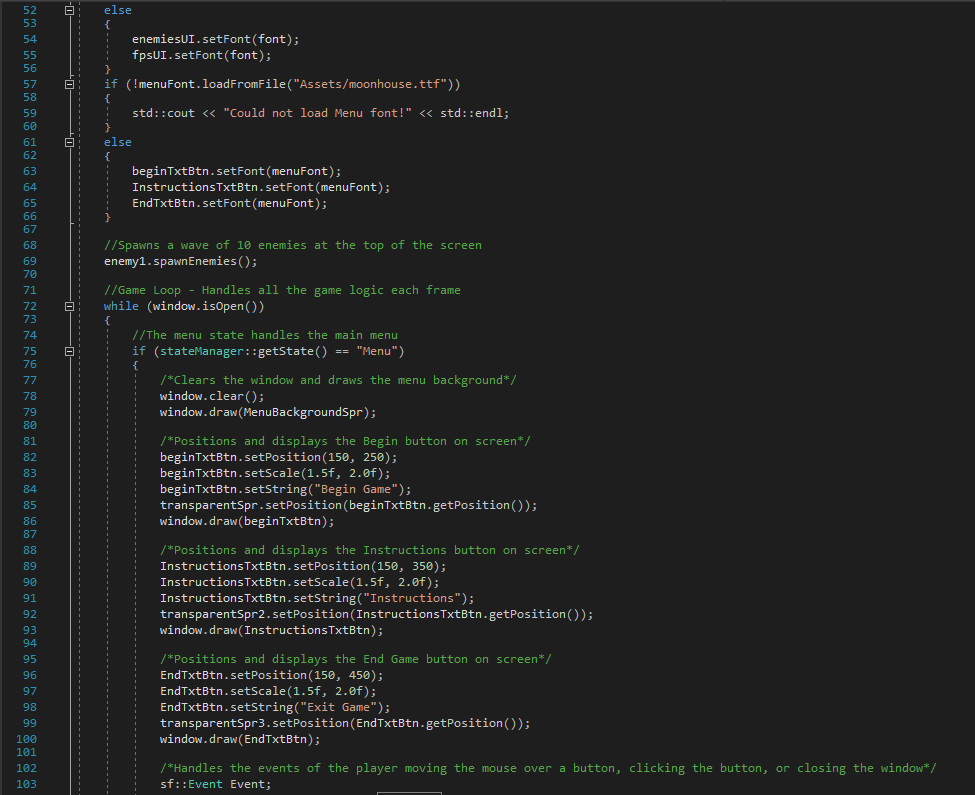
**4. Commented Source Code**

# Main

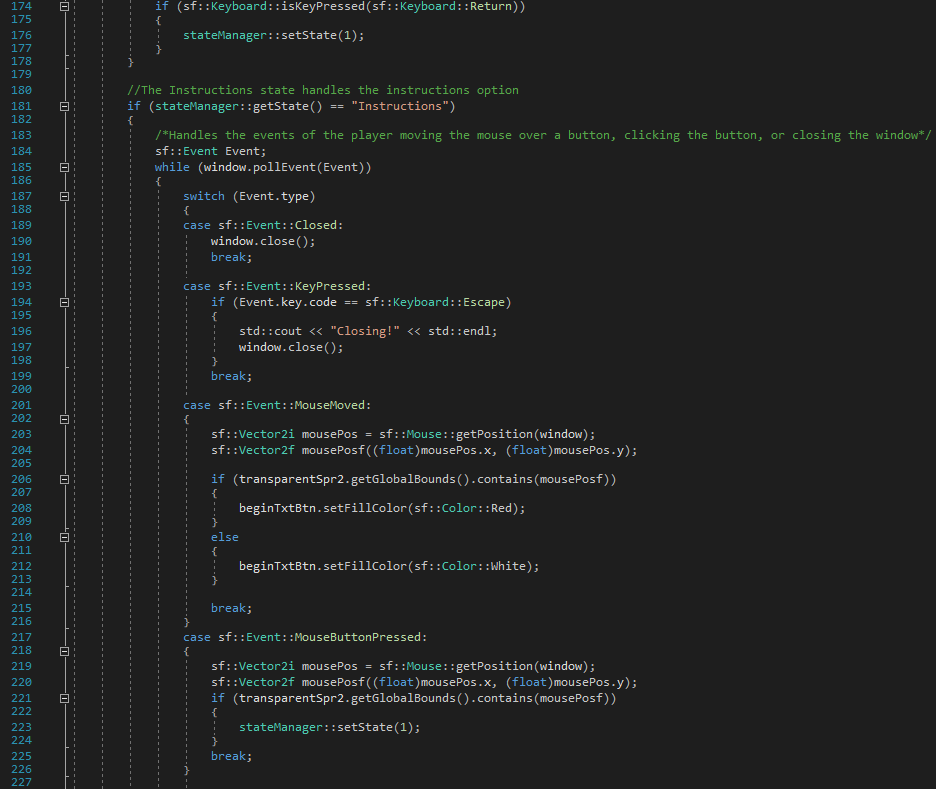


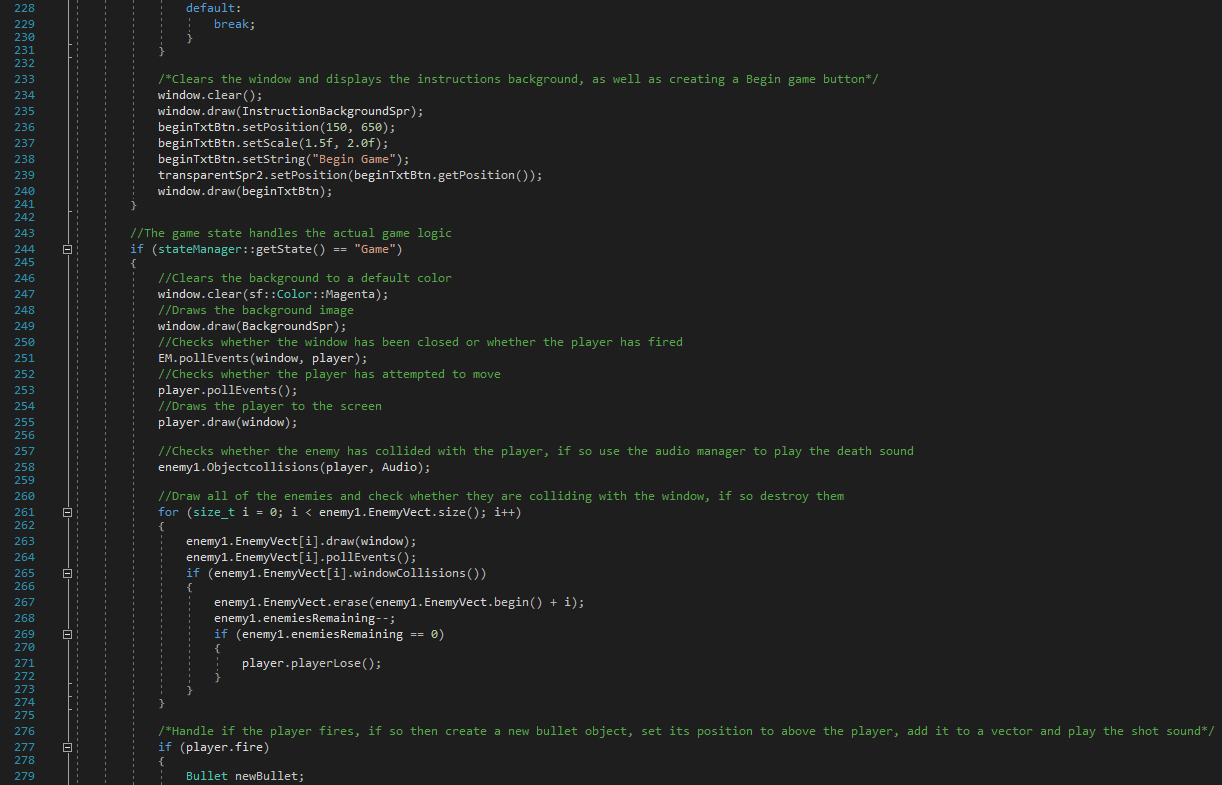
# Window Manager.cpp

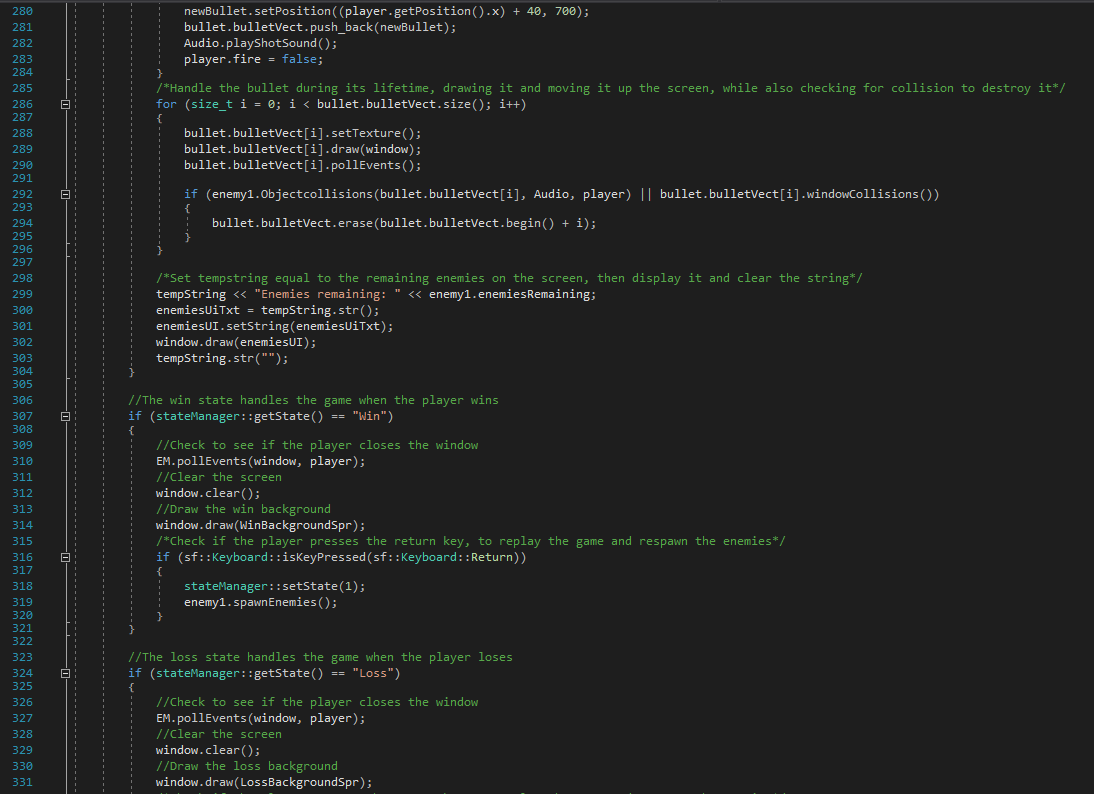


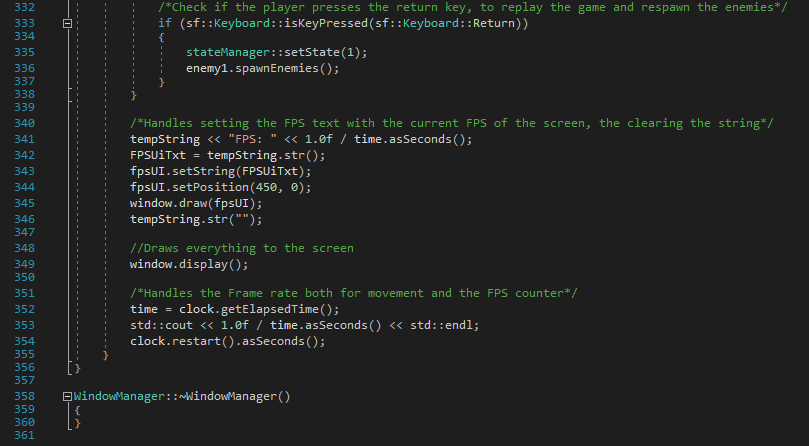




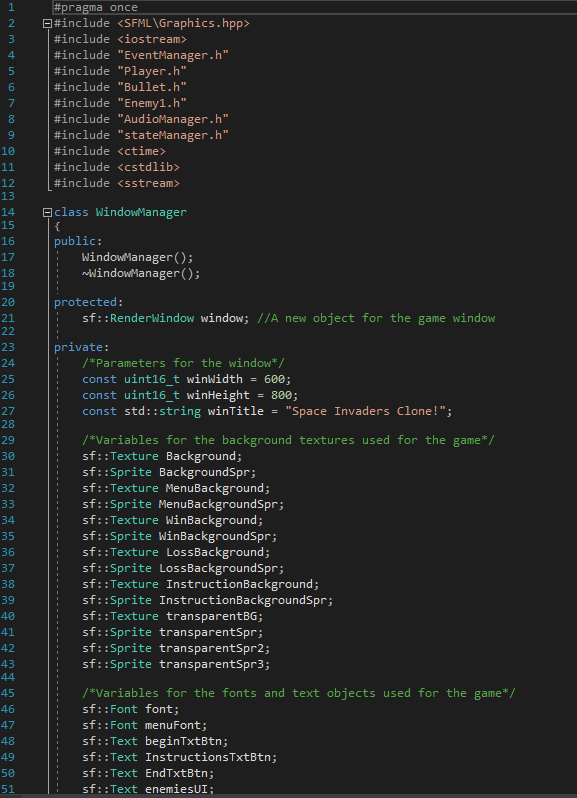


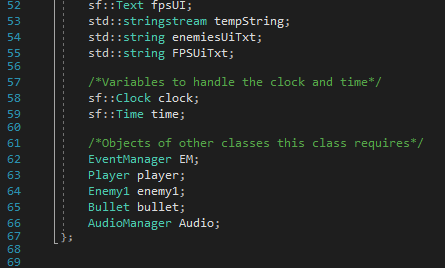




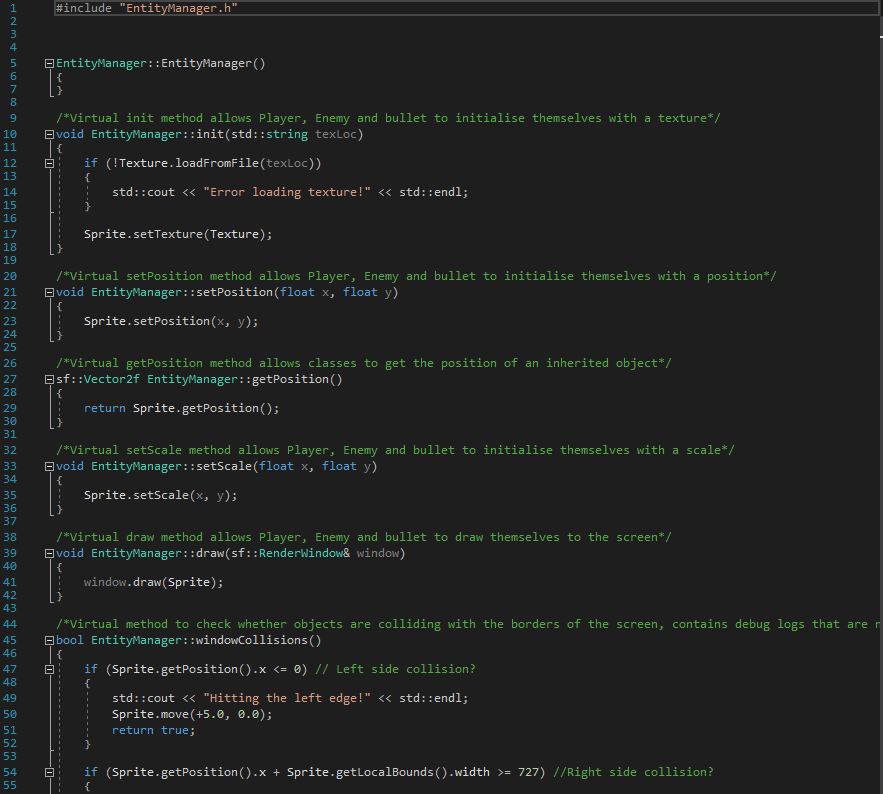


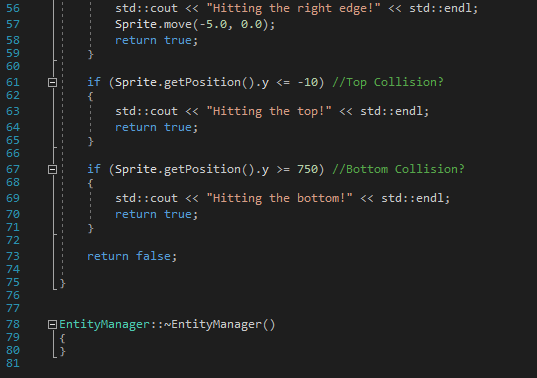
# Window Manager.h



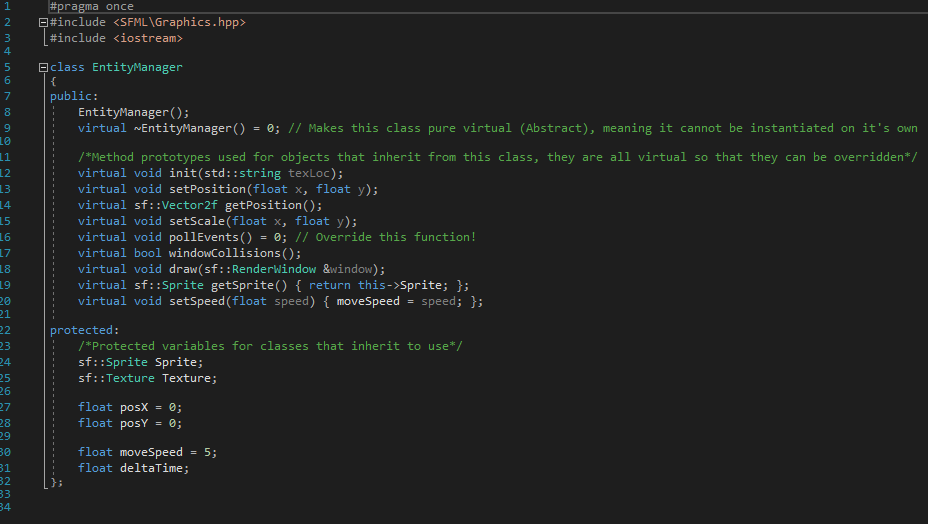


# EntityManager.cpp

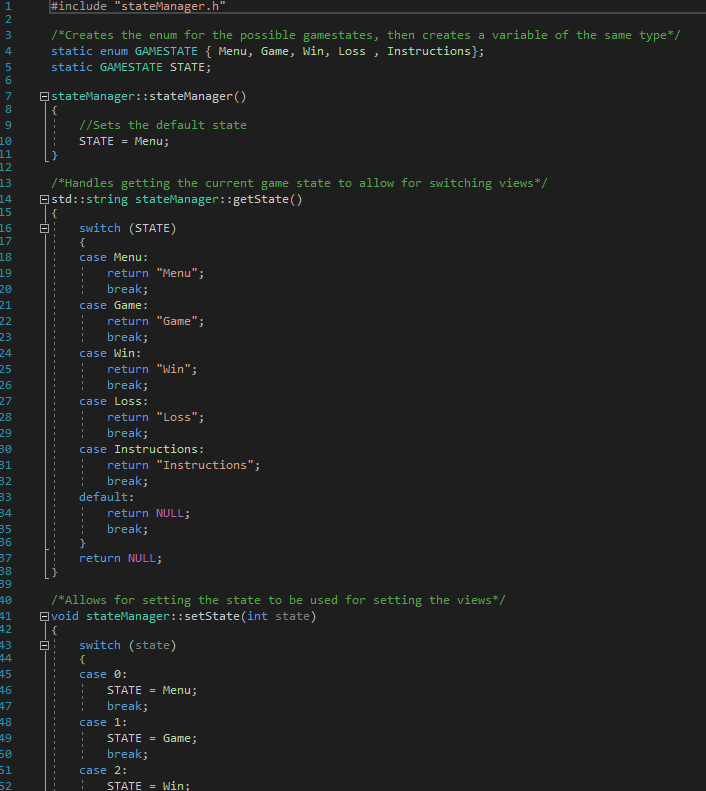


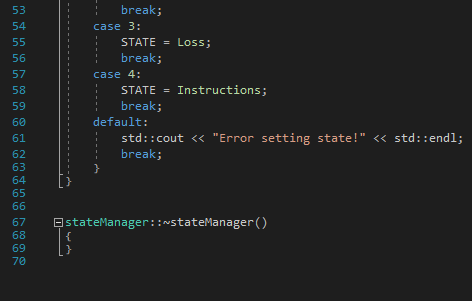


# EntityManager.h

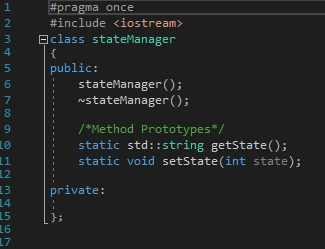


# StateManager.cpp

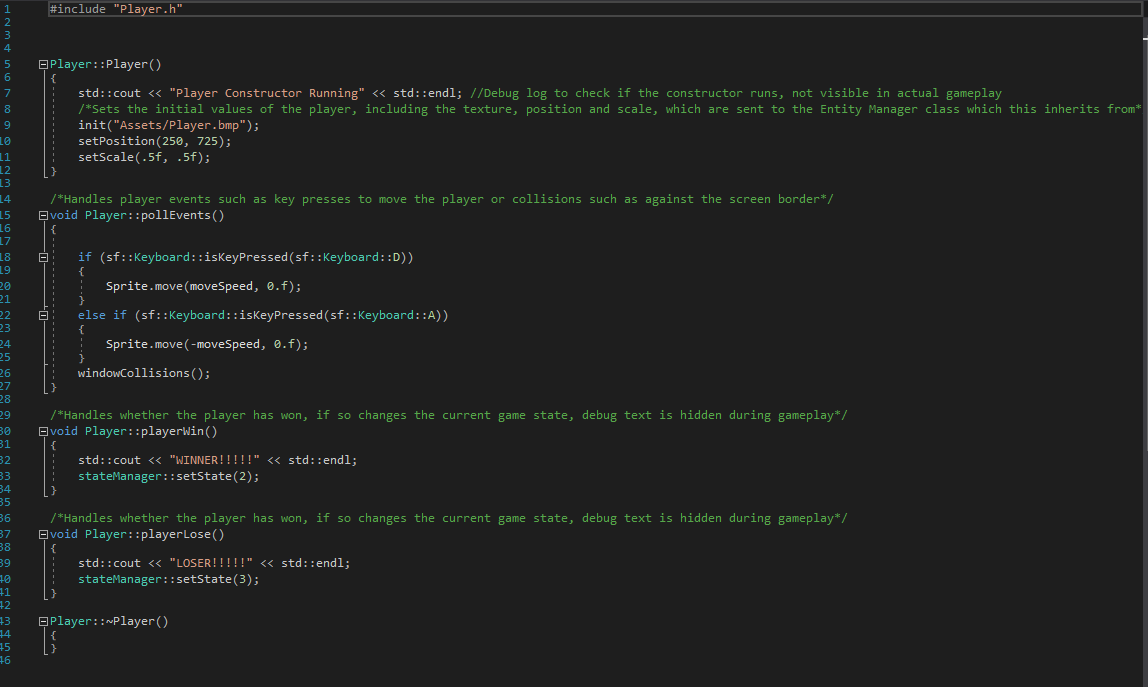




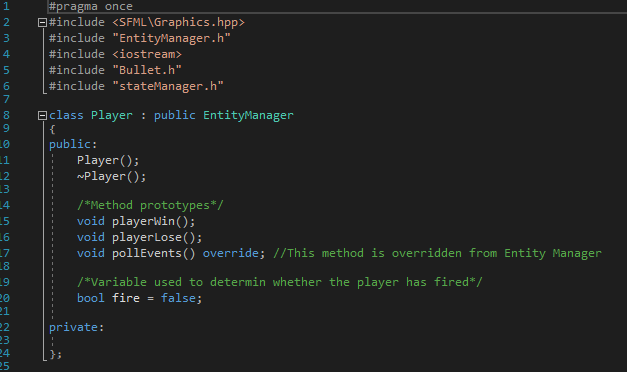
# StateManager.h



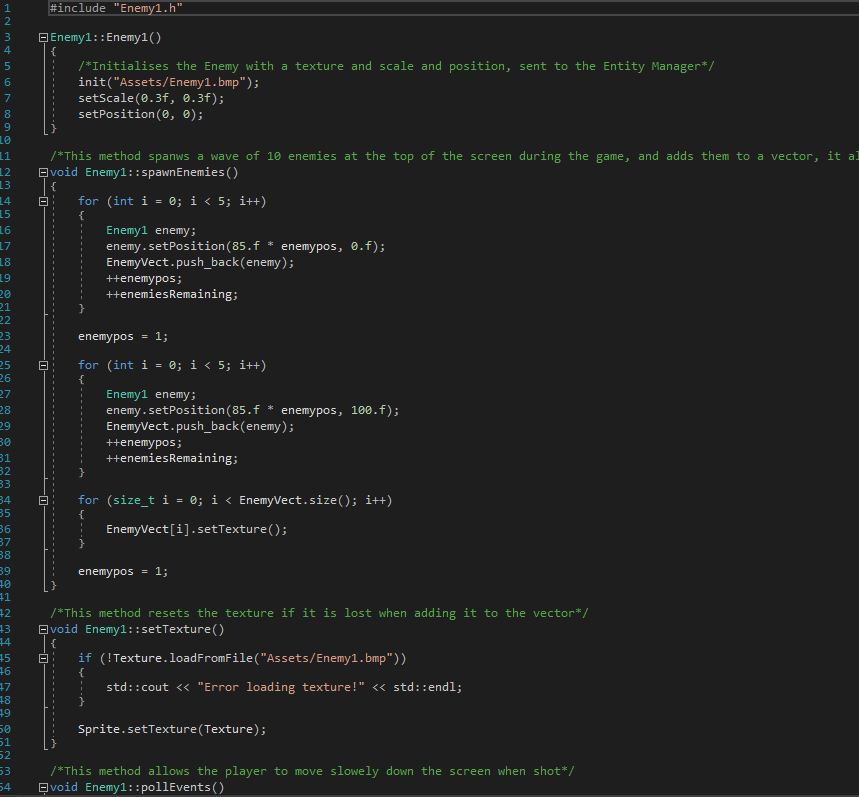
# Player.cpp

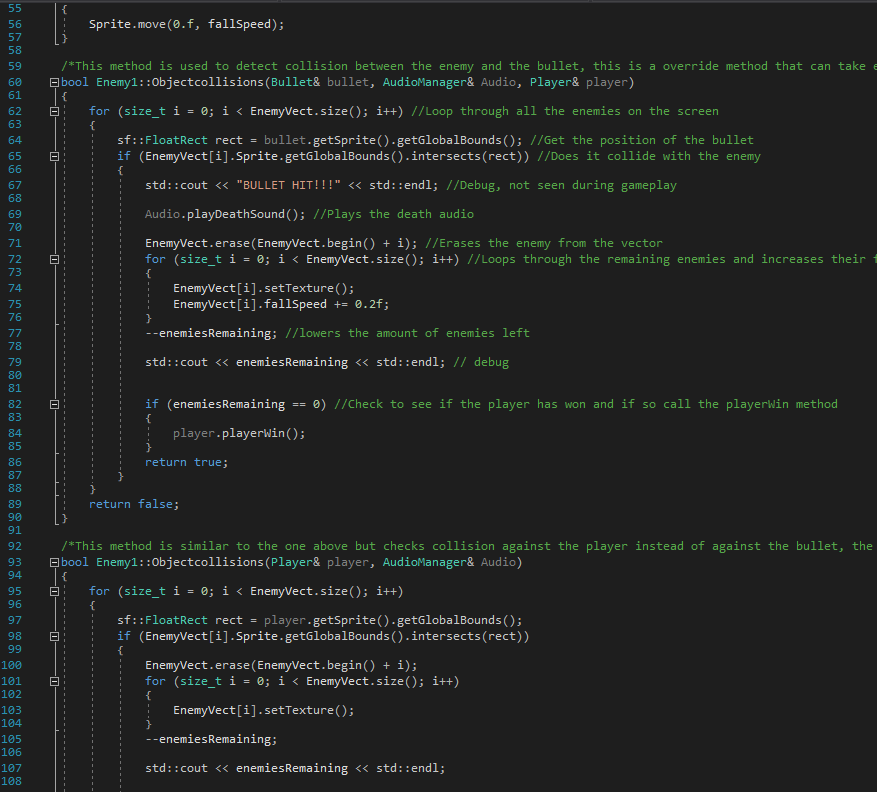


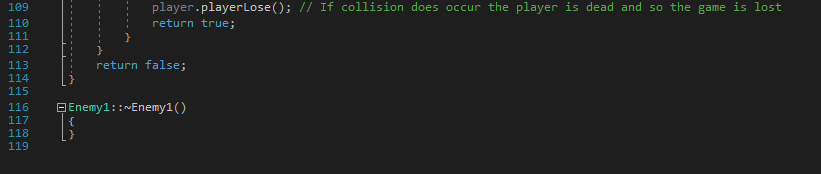
# Player.h



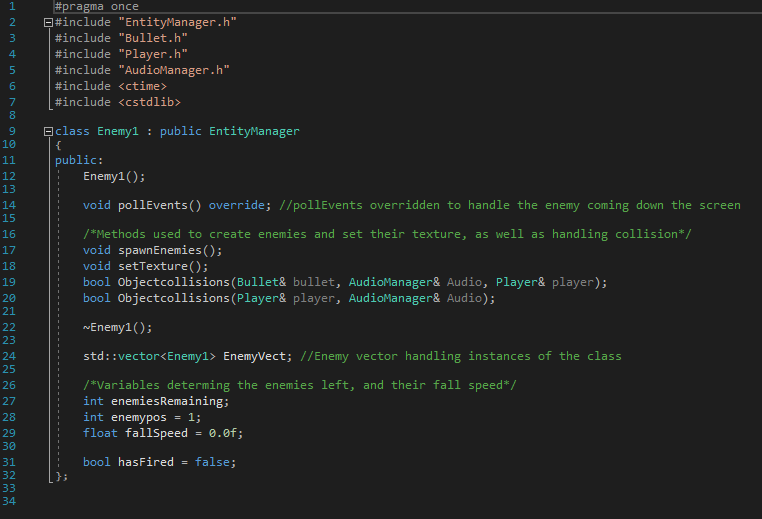
# Enemy1.cpp



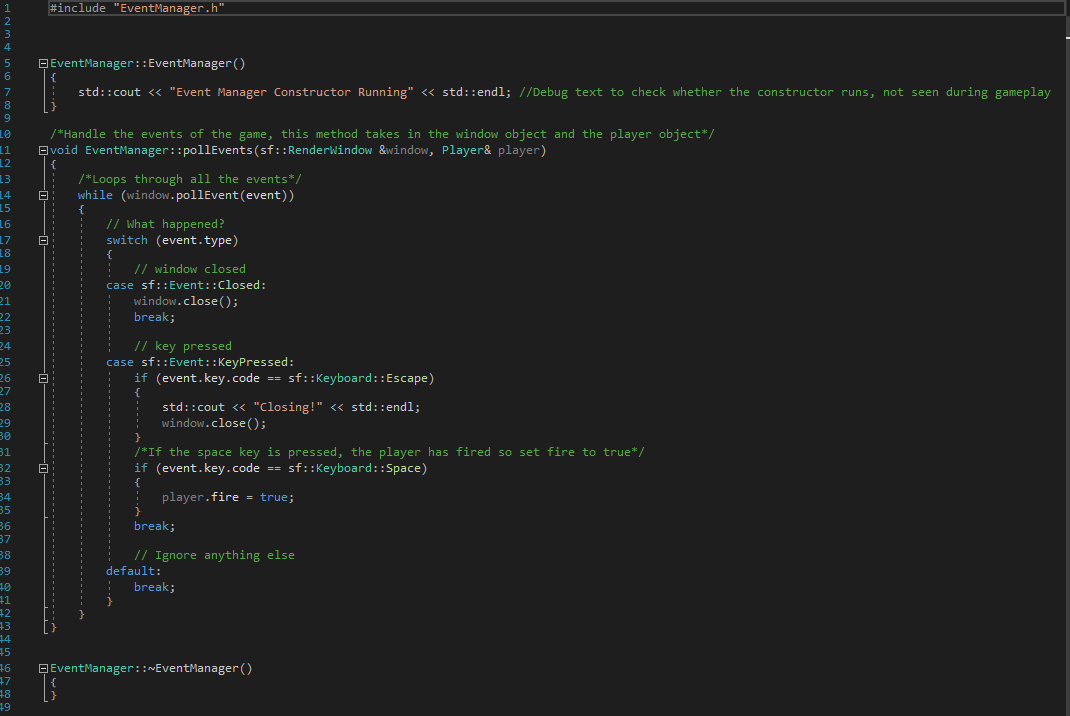




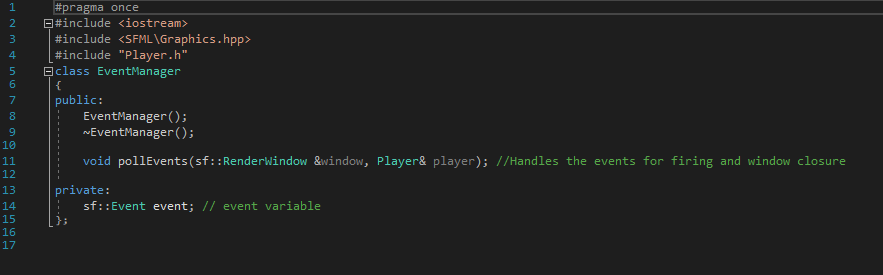
# Enemy1.h



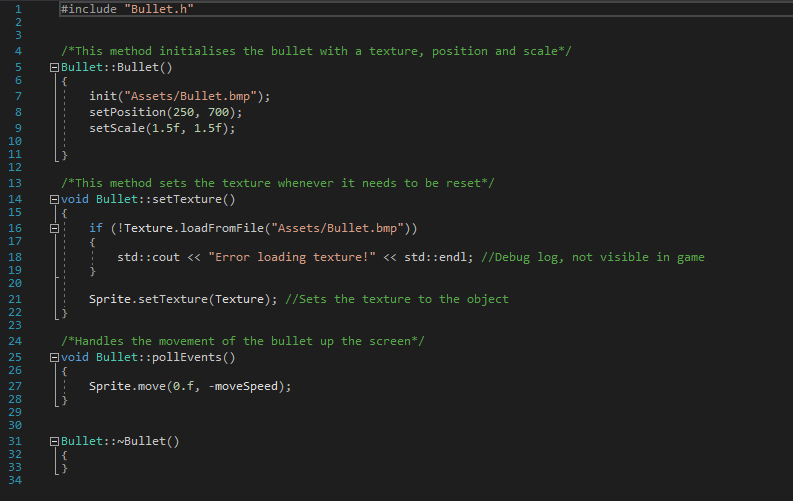
# EventManager.cpp



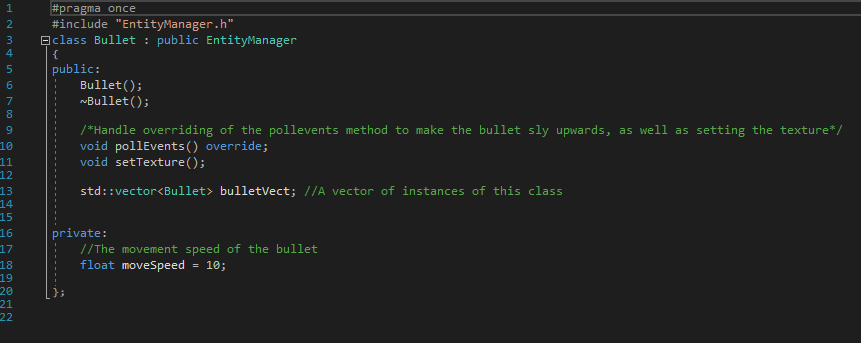
# EventManager.h



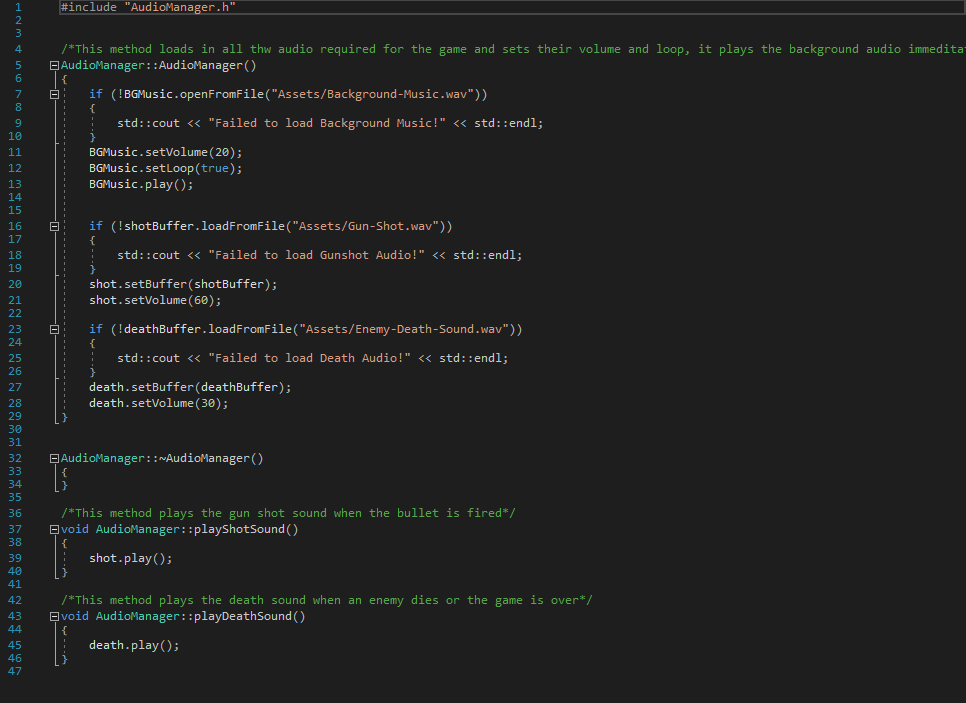
# Bullet.cpp



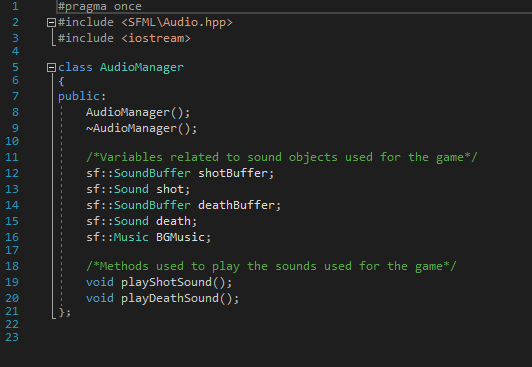
# Bullet.h



# AudioManager.cpp



# AudioManager.h



**5. Testing and Conclusions (Marks: 5%)**

Over the course of making “Space Destroyers” as I came to name it, I made sure to debug and test as much as I could, this allowed me to not only spot any blatant issues but also change things I thought were not fun or did not fit.

My initial idea was to make a game much more similar to space invaders (Almost a 1 to 1 rip off), however early on I found myself not competent enough to allow the enemies to shoot back at the player and still handle all the collisions/updates, this lead me to implementing the mechanic where the enemies drop faster the more they are killed.

Another issue I had which stopped me from creating a game identical was with my object vectors, because I store objects in the vector and not references, when they are added they lose their textures and position, this means I have to keep resetting them whenever I update the vector.

After a while of trying and failing I managed to get window border and bullet collision working which formed the core gameplay mechanics of my game and so I was extremely happy when these worked.

All in all I believe my game is enjoyable, although very short, and fulfils my original goal of creating a game that can be picked up and played very quickly and easily.

**6. Self-Assessment of Performance (Marks: 5%)**

Tutor: Andrew Watson

Student ID: 5067868

Indicate the appropriate response:

|  |  |
| --- | --- |
| Did I submit the assignment on time? | Yes |
| Did I complete the assignment? | Yes |
| If no, approx. how much did I complete | 100% |
| How happy am I with what I submitted? | Disappointed (I wanted the enemies to be able to shoot back at you) |
| What mark do I expect | 65% |
| Did I spend enough time on the assignment | Yes |
| Did I get it proof-read by someone else | No |
| Have I properly referenced it? | Yes |
| Could I improve the presentation? | Yes |

|  |  |
| --- | --- |
| The best part of my performance was: | Getting collisions to work so that the bullets could destroy enemies as well as themselves. |
| The worst part of my performance was: | The fact I could not get enemies to fire back at the player. I feel like I let myself down with this. |
| One way in which I could improve the content of my assignment is: | To of spent more time on it so that I could have implemented more features into the game. |
| One way in which I could improve the presentation of my assignment is: | Spent time creating more than just the backgrounds myself, so that I could control how they felt within the game. |
| One thing I will do to improve my performance in my next assignment is: | Spend more time on it and make sure I have a clear game plan in my head ready to create. |
| Another thing I will do to improve my performance in my next assignment is: | Ask for help and try to implement more advanced features that push my knowledge and allow me to learn. |

**7. References**

# Audio

## Background Music

<https://freesound.org/people/Mrthenoronha/sounds/371516/>

## Bullet Sound

<https://freesound.org/people/Soundholder/sounds/425347/>

## Death Sound

<https://freesound.org/people/ProjectsU012/sounds/333691/>

# Graphics

## Space Background

<https://www.pinterest.com/pin/494621971560637919/>

## Player Ship

<https://thenounproject.com/term/player/215686/>

## Enemy ship

<https://img14.androidappsapk.co/300/0/1/4/uk.co.coolalien.alieninvaders.png>

## Bullet

Unknown, it was found on google with a now dead link, I did not create it.

All other images were created by me using Photoshop