National College of Ireland
BSc in Computing
Specialisation - Gaming
2016/2017

James McConville
X13114441
James.mcconville@student.ncirl.ie

# Army Ant's

Technical Report

Supervisor: Dr Paul Stynes

National College of Ireland

## Declaration Cover Sheet for Project Submission

**SECTION 1** *Student to complete*

| |
|---|
| **Name: James McConville** |
| **Student Id: X13114441** |
| **Supervisor: Dr Paul Stynes** |

**SECTION 2 Confirmation of Authorship**

*The acceptance of your work is subject to your signature on the following declaration:*

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature:___________________________________________ Date:___________

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

# Table of Contents

## Executive Summary

Army Ant's is a third person shooter built in the Unity game engine, which is one of the most popular game engines in the world with over 5 billion downloads in 2016. Built from the ground up in a fantasy world as a fast paced 3rd person fantasy shooter.

Army Ant's is a fully engaging 3rd person shooter were you are the hero of the story.

The idea for the game was a third person shooter with the characters being ants the vehicles being other insects as in helicopters are dragonflies and the tanks will be something like beetles.

- Advanced AI developed and implemented to engage the player's skills.
- Dynamic resolutions where the game can meet the requirements of different displays.
- The environment is a 3D open world designed to engage the user.
- Uniquely designed main character, along with uniquely designed vehicles which the player can use.
- Gameplay and story designed around main character and the world in which it has been created.

You the player will take control of the uniquely designed character which is Anthony Anterson, you take control of Anthony directly after the queen has been kidnapped by your mortal enemy.

From here you must fight your way through the open worlds designed to free the queen and save the future of your people.

*"The reason why this type of game was chosen to be made is because of the fact that gaming and also building games is an interesting thing, the thought that this would be an interesting spin on the typical games out there. This idea has the ability to grow into a fully-fledged game, in the genre that most gamers seem to enjoy. It needed to be done to give people a new experience in an already existing genre."*

# 1 Introduction

### 1.1.1 Purpose

The purpose of this document is to set out the requirements for the development of the game which is a third person shooter called Army Ant's, this will give an in-depth view of the game both the idea and characters to the structure of the game with its classes and the technologies which it will use.

The intended customers will be anybody who is interested in gaming in general and people who enjoy animation based from a different point of view.

### 1.1.2 Project Scope

The scope of the project is to develop a 3rd person shooter set in a miniature world where we don't usually see, using ants as the main characters rather than humans which gives me other capabilities.

- The world will be created in Unity 3d i.e. the terrain
- The character's will be created in Maya and Blender along with vehicles
- Enemies will be made in Maya
- The weapons will be either made in Maya or Unity 3d
  This will be fully decided when I get to know the constraints better

Luckily 3d animation products both Maya and Unity are free, but Maya is only free for students. The constraints on graphics with Unity and the fact that the character made in Maya is identified as a student made element might be something that could affect the final game, but this can probably be overcome by using Blender, also unities graphic limitation are not a major deal as the likes of unreal might be more difficult with the lack of tutorials.

Army ants the third person shooter which will be set in a world we never see. This will give the game more capabilities to play with, as the player will also be able to build their base by picking up blocks as an ant for defensive situations.

### 1.1.3 Definitions, Acronyms, and Abbreviations

| | |
|---|---|
| AD | Autodesk Maya |
| 3ps | Third person shooter |

### 1.1.1 Background

Gaming has had continuous growth since roughly the 80's when having a gaming console in your house became more prevalent and as like many kids learned of this through the Sega mega-drive, this made many kids wonder how these games where made and what approaches people made to come up with these ideas.

So the idea evolved from an initial first person shooter of miniature people, but since being told this idea was not unique enough. It was decided to change things and hopefully for the better. It was then decided to change it to third person shooter and rather than having miniature people fighting a war, it was decided to go with an army ants style game.

These ants will have items like assault rifle and the likes. They will be fighting in a world that none of us ever see, as part of their military they will have mechanized dragonfly's as part of aerial support and some sort of beetle will take the place of a heavy infantry vehicle or a tank. The reason why It was chosen to make a multiplayer 3rd person genre game as it allows a large majority of people to enjoy the game.

With it being insects rather than human you don't need an age rating as most parents wouldn't see it as gory or that their impressionable kids would follow this by watching it.

## *1.2 Aims*

The objective of the game is to provide a fun and interesting bit of entertainment, it's ultimately for people of all gaming talents to ultimately to relax, for this the objective is to learn all the elements required to create a viable game from design of the environment to coding elements which you interact with. It would be to create a fully interactive environment which people will enjoy.

The objective is to try and create a game with slightly new elements, this is where it can be quite difficult as games have been around for so long now and many things have been done before. New elements would need to be learned and more elements in which that can add to the project. To bring new elements to shooting style game this will be done by it being ant's, as with humans you're quite limited in terrain but ant's will be able walk up vertical walls.

This will create more elements of attack scenario and will hopefully make the game more enjoyable and have a longer shelf life than the usual games of this type. So, as you can see in the stats below the vast majority of games are a shooter of some kind which tend to be the most popular.

## *1.3 Game Concept*

So, the initial game concept was a typical first person shooter, in which you were a typical human character just shrunken down into a world where everything normally which is standard size to you, is giant, but this idea to the panel decided it was not unique enough to garner a large score so I had to come up with a new concept which would be more beneficial.

So, the way it was altered was fairly minute in the grand scheme of things, this was changed to a third person and changed the hero to an ant, as It hasn't been done before, seeing a character which was an insect in a third person shooter and this means that your character's ability also changes as ant's have very different characteristics to humans with the ability to climb walls and carry things 10 times their own weight it gives more endless possibilities.

The single player storyline would be something like, the red ant queen has been stolen by the enemy black ant's and your job is try and rescue her as one of the elite army ant's. In the game, you will have battle against enemy AI to get through to save the queen and on the way, you will have the ability to use other insects which have been mechanized as vehicles to make your job that little bit easier.

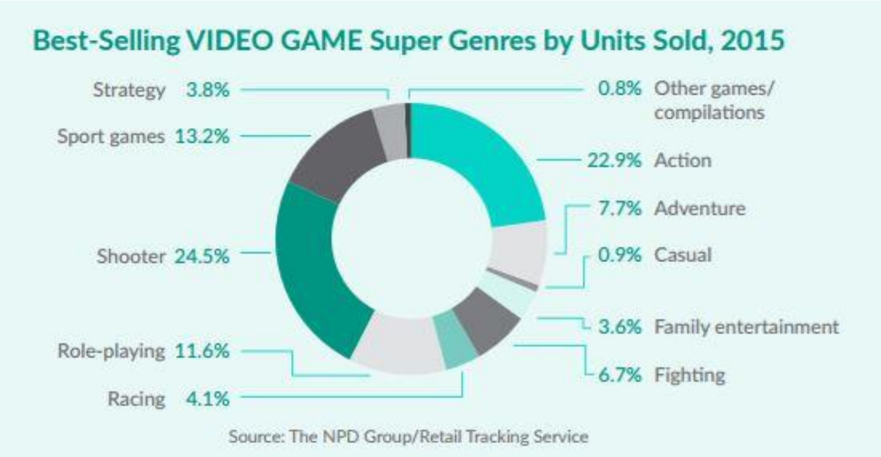
## *1.4 Research*

Although first person shooters are plentiful, the third person shooter is not so popular. This is a game where your point of view is from directly behind the character in which you are playing which doesn't seem to be one where most people don't get drawn to as much.

One of the most popular in the genre would be Gears of War which is for Xbox and PC, it is the typical human third person shooter based on an earth type planet which was colonised by humans, this is based in a fictional future and the weapons are based on things that don't exist, so nothing new here just your typical third person game.

The initial genres for research where based off the usual typical games which most people buy, this was based off research done by comparing PC, PS4 and Xbox one game popularities to give me a guide as to what would be a beneficial area to consider further.

As you can see stats from 2015 one of the most popular genre's is shooter with a quarter of the market, combine that with fantasy and action and that probably equates to 50% of the gaming market, people don't play games to do regular stuff, just like movies they like to see something different from games and the something which ignites their imagination.



Source: The NPD Group/Retail Tracking Service

**Figure 1**

The 10 grossing games of 2015 once again point to the need for escapism along with a social drive in gaming as being one of the main catalysts. (Lofgren, 2017)

- 1. Call of Duty: Black Ops III (Xbox One, PS4, 360, PS3, PC)
- 2. Madden NFL 16 (PS4, Xbox One, 360, PS3)
- 3. Fallout 4 (PS4, Xbox One, PC)
- 4. Star Wars: Battlefront (Xbox One, PS4, PC)
- 5. Grand Theft Auto V (PS4, Xbox One) 360, PS3, PC)
- 6. NBA 2K16 (PS4, Xbox One, 360, PS3)
- 7. Minecraft (360, Xbox One, PS3, PS4)
- 8. FIFA 16 (PS4, Xbox One, 360, PS3)
- 9. Mortal Kombat X (PS4, Xbox One)
- 10.Call of Duty: Advanced Warfare (Xbox One, PS4, 360, PS3, PC)

Now 5 of these top 10 games are shooter style games and the one which has lasted the longest is Grand Theft Auto V which is the only 3rd person shooter on the list, this was released as far back as September 2013 and this shows the longevity that some games can have.

The research also covered who the average gamer was and this point's that the typical gamer's average age would be 35 years old so people roughly born in the 80's would be a large part of that market. This is broken down into 27% under 18, 29% 18 – 35, 18% 36 – 49 and finally 26% 50+. These stat's kind of surprised me as you may believe older generation would have little time for gaming. (Anon, 2017)

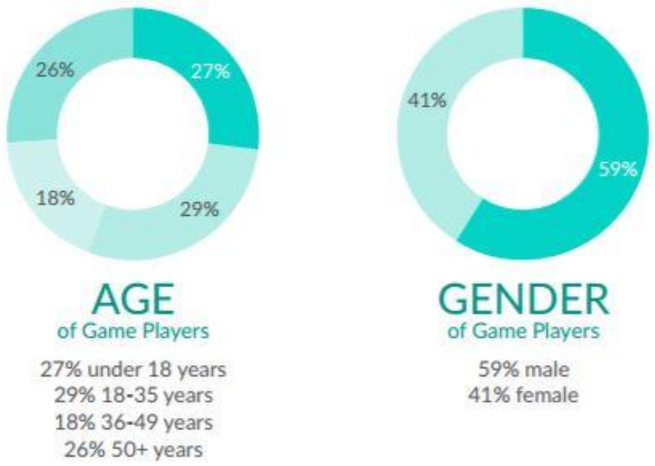


**Figure 2**

All these stats are based on American specific which states that 63% of American homes have at least one gamer living and the gender stats have become a lot more equal in the last couple of decades mainly being seen as a male pass time, the split up as 59% are male and 41% are female. This also points to females being on average older game's with an age of 44 while males are average at 35.

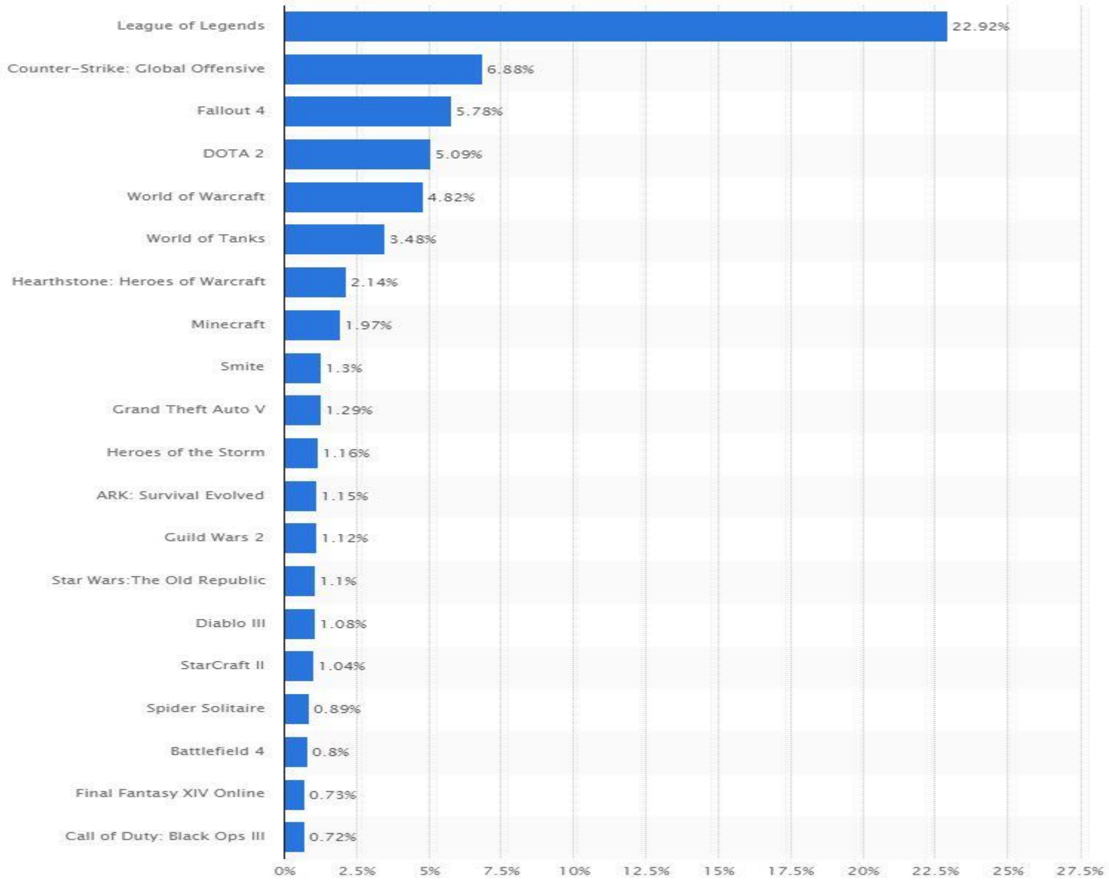


**Figure 3**

Watchdogs is another popular game which promised so much but ultimately failed to deliver. This game is based around hacking your environment to gain the upper hand, but this became too repetitive for most people and although there was the shooting elements in the game it wasn't enough to keep most people's attention for too long, but there is a sequel out this year.

From many years ago their was games like resident evil which was a sci-fi zombie killer which you had to solve puzzles to continue through the game and one of the most popular is Metal Gear Solid which was a game based around stealth and this game came from a top down shooter from many years before.

## 1.5 *Full Game Concept Breakdown*

**Overview of Genre Setting:**

- The main genre of the game will be shooter
- The game will be set in a 3rd person view from directly behind the character which will let you see more detail of the main character as otherwise things could have been faked.
- The genres that will help create the game are both fantasy and also shooter.

**Environment (World) Setting:**

- The world setting is an open world set on a planet called Arazmus
- The Year 2010.
- World is set in a fantasy universe one where humans don't exist.
- The queen of the ant colony has been kidnapped by the evil alien insects.

**Story Characters:**

- **Anthony Anterson:** is the hero of the game and it is his job to save the queen from the kidnappers, he is the only military ant left as they have lived in peace for many years on this planet by themselves.
  - **Weaponry** the only rifle left behind as a relic in a war museum, fires single shot's as fast as you can pull the trigger.
  - **Alternate weapons** is the BeetleTank which is a mechanised beetle which the ants have utilised to destroy larger enemies.

**Enemies Types:**

- **AlienInsect: these are the controllers of all that is evil on this planet actively patrol region looking for character.**
  - **Can Kill Player**
  - **Only detects character directly in front of them.**

- **DragonBoar: these are like the attack dogs of this planet can see much further than their masters and don't need to be looking in your direction.**
  - **Can Kill Player easier by doing more damage**
  - **Moves much faster and can detect you much easier.**

- **Enemy BeetleTank: enemy has utilised the ant's own tech against them and can even up things.**
  - **Can Kill Player easier by doing more damage**
  - **Moves much faster and can detect you much easier.**

**Setting History**

The Year 2010, the planet Arazmus is a peaceful planet which is home to a Ant humanoid type creature. This has all changed when the Ant's queen is kidnapped by Alien insects.

- Antopians
- The Zeirgeist

The planet was a peaceful farming planet ran by ant type humanoids who have prospered for many decades, but all that changed when the evil Zeirgeist crash landed. Things changed when the Zeirgeist kidnapped the Antopians queen. This meant that somebody needed to rescue the day or else society would be ruined and that person is you.

## *1.6 Technologies*

The technologies that will be used is the standard mouse and keyboard, this will be used in conjunction with Maya to model the 3D character and the environment. These technologies where not part of college in any aspect, Maya we have used on a little basis just constructing very limited shapes, but in this regard we had to learn it all by ourselves.

Unity 3D is another technology which is to be used and this technology has also not been touched and the coding is done in C# which is a language we have briefly touched when doing Advanced Internet Technology.

Blender

Blender is also a 3d modelling suite which is completely free to use, it is an open source software which has been provided by many developers and is a lot easier for indie developers to create games and animations, but is also an easy way to get rid of limitation's built in Maya 3d modelling suite.

C#

Is an object orientated computing language which is the primary language used through Unity, there are other languages that are also used in Unity and that is JavaScript which is known as a scripting language which means unlike other languages it does not need to be compiled to be used.

Photon Unity Networking(PUN)

This is a technology which allows you to create a multiplayer element to your game and as that's what was set out to do from the start and this is designed by the game engine creators with which will be used. So, because they use a globally supported cloud, the latency is very low and the game should run smoothly. This means it can create a 20 multiplayer game for free and if people want to stretch that to 100 it's only $95 for 60 months.

Photon is able to scale quite rapidly according to the information provided it can scale from 10's to 1,000's. It is also cross platform available as with many games now are playable against other platforms, more recently people who play on pc's have been playing against people who play on a normal ps4 which was once not really thought of.

This also allows you to determine the matchmaking, whether it's done by people ranking or whether you allow people to create rooms and make their own matches and as the game probably won't have a ranking system, this is the option which will probably be gone for. It also supports voice and messaging which is definitely something which people enjoy about multiplayer games.

AI

Although at the minute I don't have a fully thought out strategy about implementing an AI algorithm like min-max or alpha beta pruning as I'm not sure whether they would work properly for a third person shooter, I do know Unity has a path finding mesh where you can make an AI character navigate around a game level, that means manoeuvring around obstacles and opening doors.

I have however found a tutorial on Unity to implement a finite state machine which may be viable for my game and maybe with a few alterations could be the ideal solution to the problem.

## *1.7 Structure*

The first chapter is outlining the technologies used and with a brief overview of the project, this will include how I intend to use those technologies to implement the making of the game. This includes all the coding type languages which I will use and a brief overview of the gaming industry and its growth.

The second chapter outlines the system requirements and the functional and non-functional requirements used to outline how I think the system should work, this includes how the system needs to react to certain inputs like what section of the game in which you choose whether it's multiplayer and so on.

## *1.8 System*

## *1.9 Requirements*

As this game will be based on Unity and will be mainly based around multiplayer elements, the user will be required to have an internet connection.

- Supported OS: Windows 7 SP1, Windows 8/8.1
- Processor: Intel Core i5-2500K @ 3.3 GHz or AMD FX-8350 @ 4.0 GHz or AMD Phenom II x4 940 @ 3.0 GHz.
- RAM: 3 GB(Recommended)
- Video Card: NVIDIA GeForce GTX 680 or AMD Radeon HD 7970 (2 GB VRAM)
- DirectX: Version 11.

## *1.10 Functional requirements*

- A start menu where people can choose what they want to do
- A terrain to explore
- A character to interact with the environment through
- AI intelligence so that they don't just walk into a wall the whole time
- Realistic physics along with damage
- Other insects which will act as modes of transport like a helicopter
- Character must have weapon available and must be able to re-stock ammo
- The terrain should be attractive but the environment on in which the player can explore should be limited.
- The player can choose either single or multiplayer

## *1.11 Use Case Diagram*

The Use Case Diagram provides an overview of all functional requirements.

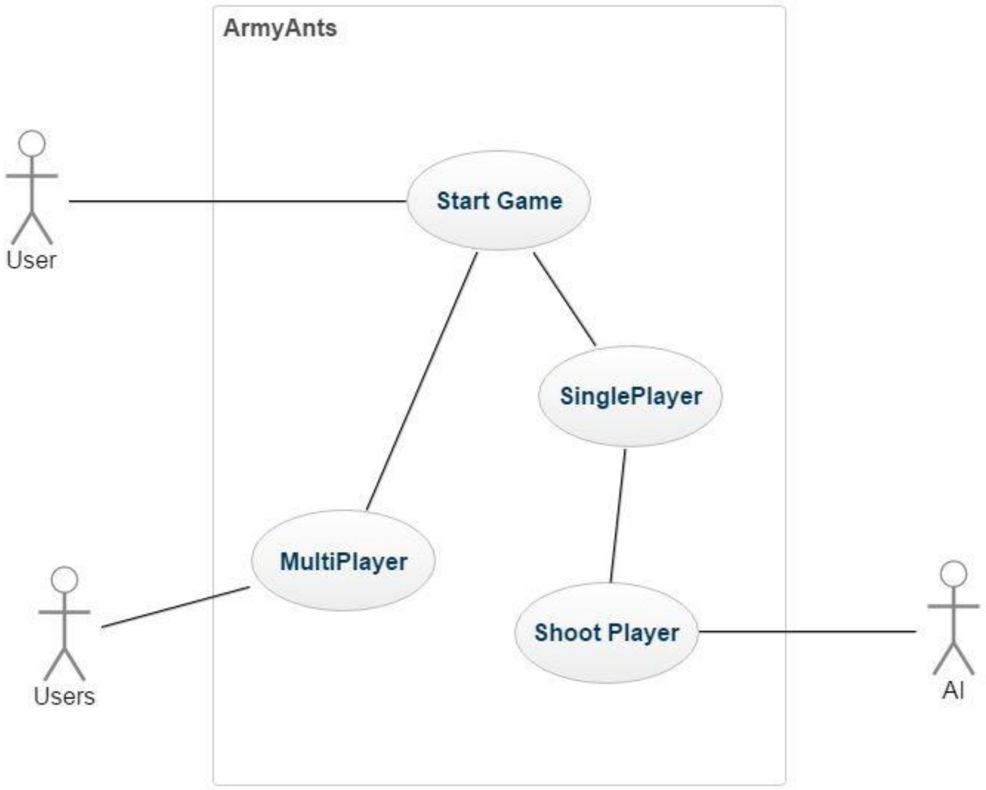


**Figure 4**

## *1.12 Requirement 1 <Single Player>*

### 1.12.1 Description & Priority

The user shall be able to select single player game. The user will be able to customise their character.

### 1.12.2 Use Case

**ID**

UC01

**Scope**

The scope of this use case is to allow the player to play vs AI.

**Description**

This use case describes the how the user selects single player and launches the game.

**Use Case Diagram**

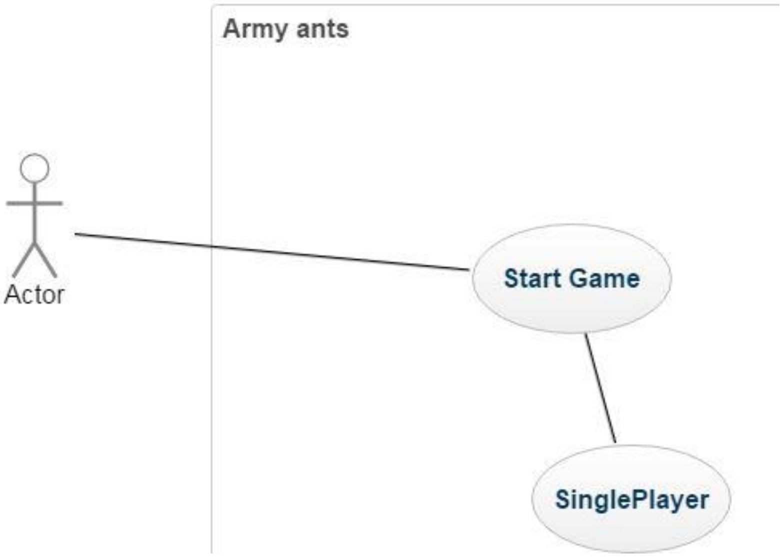


**Figure 5**

### Flow Description

### Precondition

The game must be launched on the player's computer

### Activation

This use case starts when an <Actor>clicks single player

### Main flow

1. The user launches the game
2. The <Actor>selects single player
3. The <Actor> customises it's character
4. The game launches
5. The <Actor> quits game

### Alternate flow

A1 : <title of A1>

1. The user launches game
2. The <Actor> single player
3. The <Actor> quits game

### Exceptional flow

E1 : <title of E1>

4. The user launches the game
5. The <Actor> quits the game

### Termination

The system presents the next your desktop

### Post condition

On success the game launches in single player mode and loads AI.

### 1.1.4 Requirement 2 <Multiplayer>

**ID**

UC02

### 1.12.3 Description & Priority

Many users should be able to select multiplayer

### 1.12.4 Use Case

**Scope**

The scope of this use case is to allow many people play against each other

**Description**

This use case describes how many people are allowed play the game at the same time

**Use Case Diagram**

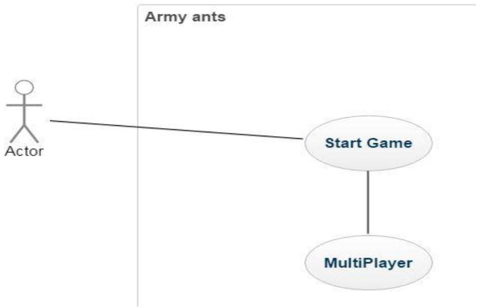


**Figure 6**

**Flow Description**

**Precondition**

The game must be launched on the players computer

**Activation**

This use case starts when an <Actor> clicks multiplayer

**Main flow**

1. The user launches the game
2. The <Actor> selects multiplayer
3. The <Actor> customises their character
4. The <Actor> starts the game

**Alternate flow**

A1 : <title of A1>

1. The user launches game
2. The <Actor> single player
3. The <Actor> goes back to main menu
4. The <Actor> selects multiplayer

**Exceptional flow**

E1 : <title of E1>

5. The user launches the game
6. The <Actor> quits the game

**Termination**

The system presents your desktop screen

**Post condition**

On success the system launches a multiplayer

## *1.13 Requirement 3 <Character Customise>*

**ID**

UC03

### 1.13.1 Description & Priority

User should be allowed customise his character

### 1.13.2 Use Case

**Scope**

The scope of this use case is to allow a single or multiplayer customise their character

### Description

This use case describes how many people are allowed to customise their character

### Use Case Diagram

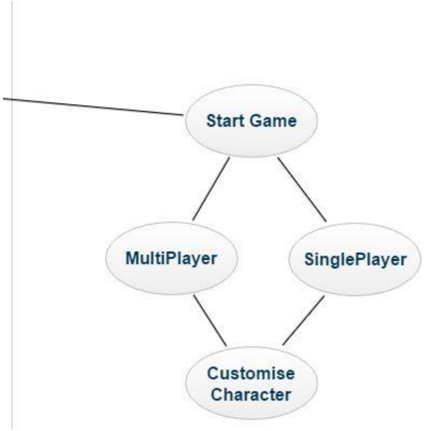


Figure 7

### Flow Description

### Precondition

The user must have selected either single/multiplayer

### Activation

This use case starts after an <Actor> clicks multiplayer/single

### Main flow

1. The user launches the game
2. The <Actor> selects multiplayer/single
3. The <Actor> customises their character
4. The <Actor> starts the game

### Alternate flow

A1 : <title of A1>

1. The user launches game
2. The <Actor> multiplayer
3. The <Actor> quits game

**Exceptional flow**

E1 : <title of E1>

1. The user launches the game
2. The <Actor> quits the game

**Termination**

The system presents the main menu for single/multiplayer

**Post condition**

On success the system launches a level

## *1.14 Requirement 2 <Tutorial>*

**ID**

UC04

### 1.14.1 Description & Priority

The user should be allowed select tutorial

### 1.14.2 Use Case

**Scope**

The scope of this use case is to allow the user to select tutorial

**Description**

This use case describes how the user can select tutorial and be guided through controls

### Use Case Diagram

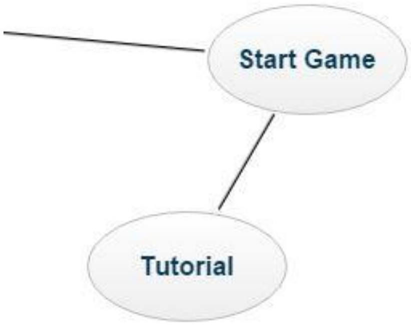


**Figure 8**

### Flow Description

### Precondition

The game must be launched on the players computer

### Activation

This use case starts when an <Actor> clicks tutorial

### Main flow

1. The user launches the game
2. The <Actor> selects tutorial
3. The <Actor> starts the game

### Alternate flow

A1 : <title of A1>

1. The user launches game
2. The <Actor> single player
3. The <Actor> goes back to main menu
4. The <Actor> selects tutorial

### Exceptional flow

E1 : <title of E1>

7. The user launches the game
8. The <Actor> quits the game

### Termination

The system presents your desktop screen

**Post condition**

On success the system launches a tutorial

## *1.15 Requirement 2 <ShootPlayer>*

**ID**

UC05

### 1.15.1 Description & Priority

The AI must be able to shoot the player

### 1.15.2 Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

**Scope**

The scope of this use case is to allow the AI to shoot the main player

**Description**

This use case describes how the AI player is to try and kill the single player

**Use Case Diagram**

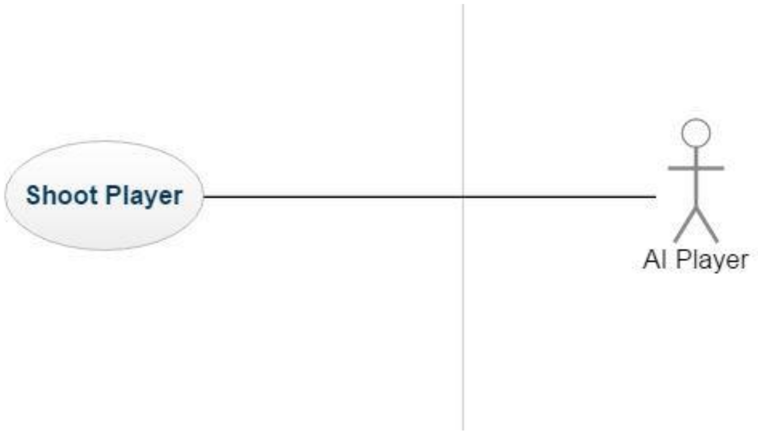


**Figure 9**

**Flow Description**

**Precondition**

The game must be launched on the players computer

**Activation**

This use case starts when a Human<Actor> clicks multiplayer/singleplayer

**Main flow**

5. The user launches the game
6. The <Actor> selects Shoot

**Alternate flow**

A1 : <title of A1>
9. <None>

**Exceptional flow**

E1 : <title of E1>
10. <None>

**Termination**

The system presents your desktop screen

**Post condition**

On success the system launches a multiplayer

## *1.16 Requirement 2 <Quit Game>*

**ID**

UC06

### 1.16.1 Description & Priority

Many users should be able to select Quit Game

### 1.16.2 Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

**Scope**

The scope of this use case is to allow a player to quit game

**Description**

This use case describes how user can quit game

**Use Case Diagram**

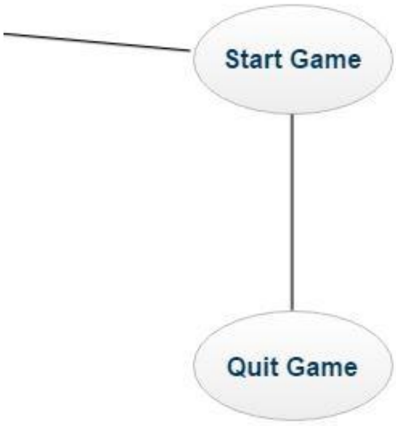


**Figure 10**

**Flow Description**

**Precondition**

The game must be launched on the players computer

**Activation**

This use case starts when an <Actor> launches the game

**Main flow**

1. The user launches the game
2. The <Actor> selects quit game

**Alternate flow**

A1 : <title of A1>

1. The user launches game
2. The <Actor> multiplayer
3. The <Actor> quits game

**Exceptional flow**

E1 : <title of E1>

1. The user launches the game
2. The <Actor> quits the game
3. The game doesn't respond
4. Task manager has to be launched

**Termination**

The <Actor> continues play

**Post condition**

On success the system closes the game

## *1.17 Non-Functional Requirements*

## *1.18 Performance/Response time requirement*

The game should respond fast and with no glitches, the game should not freeze and should respond promptly to the actions of the user.

## *1.19 Availability requirement*

The game shall be available 24/7 to users as long as they have a desktop that can meet the requirements needed to launch the game.

- Using a digital distributor, is the norm for independent game developers there are a number of platforms but the one that is most used is Steam and their greenlight system.
- Steam greenlight is a system where users get to pick the games which is available on digital download which is a system where rather than having a physical copy, you just get the software.
- Greenlight from a developer view initially charges $100 fee which is donated to a children's charity which stops people submitting any game they knocked together.

- Greenlight is where users vote for the game to be launched, this will make your game available and then your project will be greenlit, you will also need to prove its your own idea.
- As for marketing this would allow customers to directly interact with you and you can tell if it will be successful or not, especially with a low cost initially, your marketing is basically done.
- They help you figure out what the price of your game should be to best attract custom.

## *1.20 Security requirement*

Once installed the game doesn't require any connection to the internet unless they are playing the internet. All that is required is the user to have the required privileges on the computer they wish to install.

## *1.21 Reliability requirement*

The game should launch in 10 seconds when the exe file is clicked.

## *1.22 Maintainability requirement*

The game should be easy to extend as new features should be able to be added. If steam is the option of distribution patches and other things like DLC should be made available to keep the game growing and interesting.

## *1.23 Portability requirement*

As long as you have a connection to the internet you should be able to launch the game or a laptop with the game already installed

## *1.24 Extendibility requirement*

New levels and features should easily be added to the game including new characters.

## *1.25 Reusability requirement*

The game levels and characters can be used in new games and even animated movies of different kinds.

## *1.26 Resource utilization requirement*

The game should use resources common to most games i.e. mouse, keyboard.

## *1.27 Interface requirements*

## *1.28 User interfaces*

- Keyboard
- Mouse

## *1.29 Controls*

Up-arrow forward movement

Down-arrow backward movement

Left-arrow turn left

Right-arrow turn right

## *1.30 Design and Architecture*

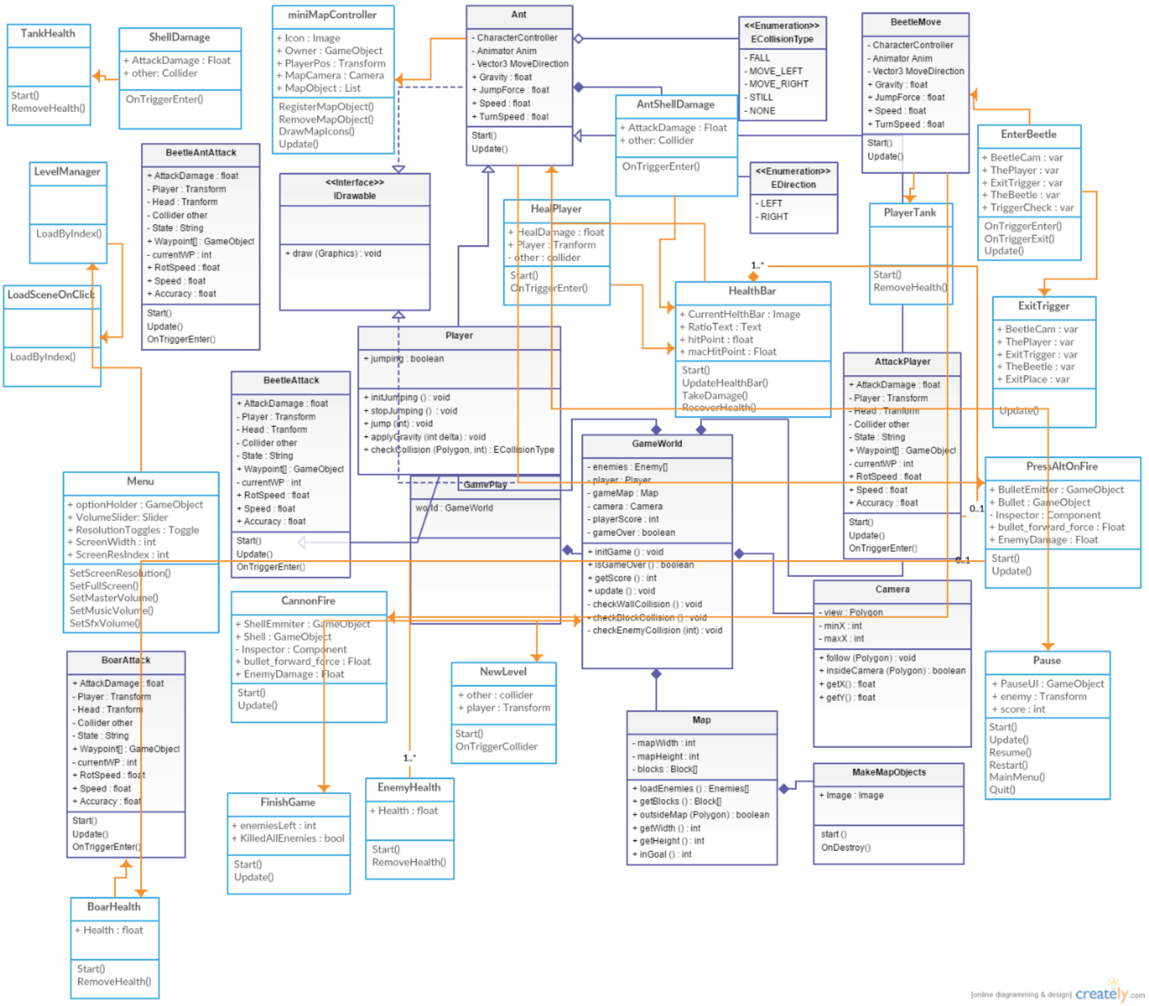


Figure 11 https://creately.com/diagram/j1w7k4kc1

# Implementation

Each implementation will include a breakdown including screenshots from the code implementation for these aspects of the game.

## *Game Systems Implementation*

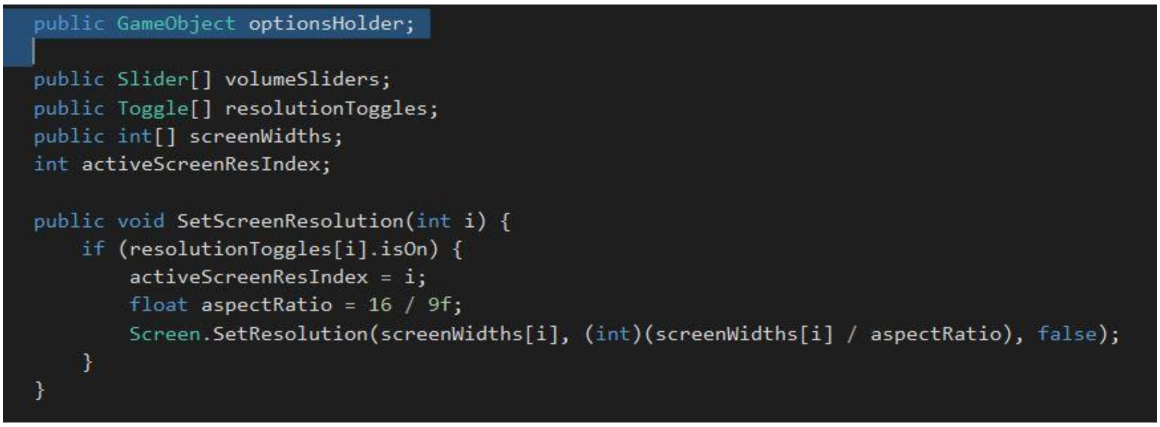
This will cover the implementation of main systems which are crucial for games. These are linked to many functional requirements which have already been discussed previously.

### *Main Menu Implementation*

These will be set out in screenshots of code as unities implementation is done mainly by coding in C# as then attaching that code to a UI.

**Controls, Maintains and Implements:**

- **Setting:** Settings are the default settings which will remain the same if the user doesn't alter anything.
- **Setting screen resolution:** can be altered as you see fit at the start of the game

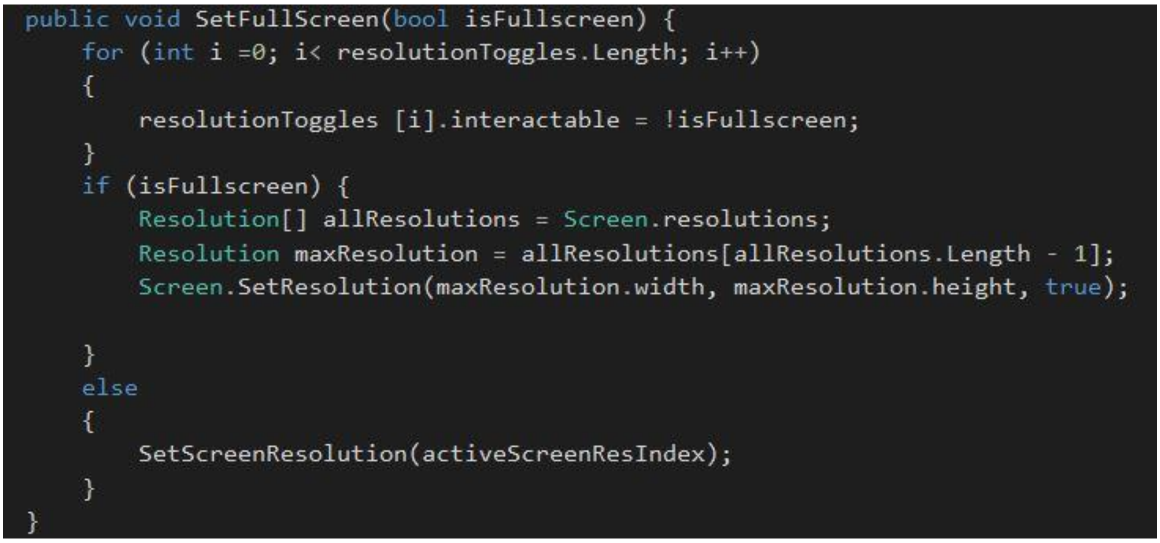


```
public GameObject optionsHolder;

public Slider[] volumeSliders;
public Toggle[] resolutionToggles;
public int[] screenWidths;
int activeScreenResIndex;

public void SetScreenResolution(int i) {
    if (resolutionToggles[i].isOn) {
        activeScreenResIndex = i;
        float aspectRatio = 16 / 9f;
        Screen.SetResolution(screenWidths[i], (int)(screenWidths[i] / aspectRatio), false);
    }
}
```

**Figure 12**

This script is attached to the player UI main screen at the start, these functions are attached to both toggles and sliders to make the use as user friendly as possible.

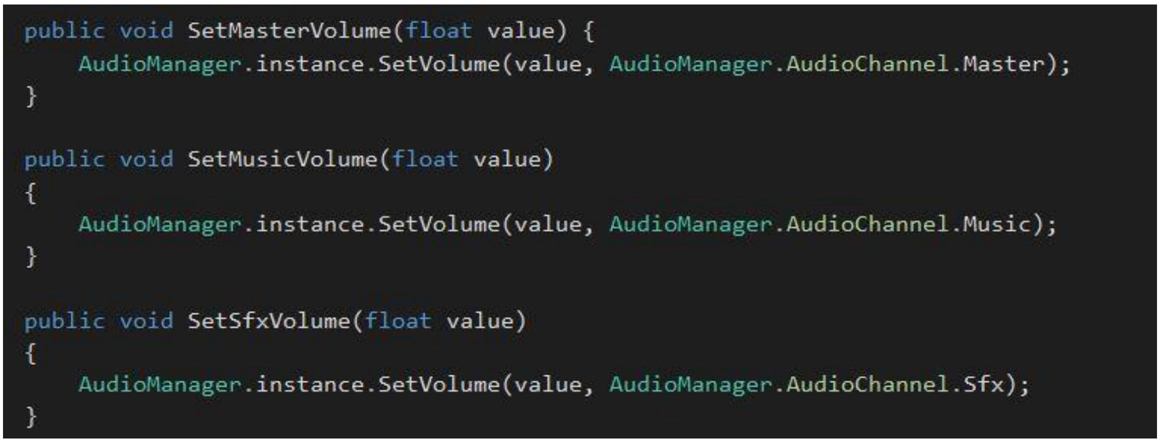
The C# code to implement the full screen resolution which checks the max possible resolution and then implements it.



```
public void SetFullScreen(bool isFullscreen) {
    for (int i =0; i< resolutionToggles.Length; i++)
    {
        resolutionToggles [i].interactable = !isFullscreen;
    }
    if (isFullscreen) {
        Resolution[] allResolutions = Screen.resolutions;
        Resolution maxResolution = allResolutions[allResolutions.Length - 1];
        Screen.SetResolution(maxResolution.width, maxResolution.height, true);

    }
    else
    {
        SetScreenResolution(activeScreenResIndex);
    }
}
```

**Figure 13**

The code snippet which is used to implement the audio in the game separated into music, sound effects and also the master volume which is designed to implement on the sliders in the menu.

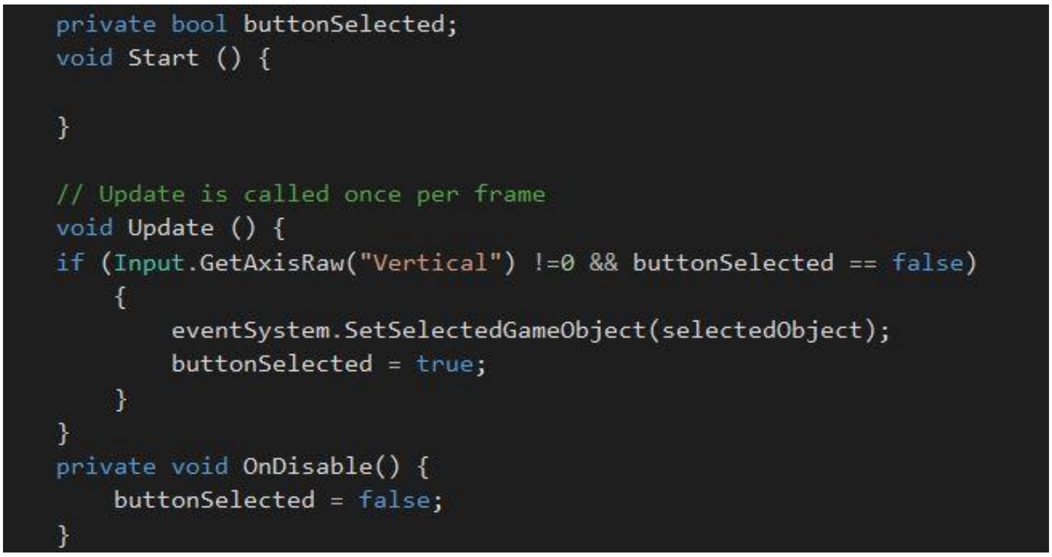


```
public void SetMasterVolume(float value) {
    AudioManager.instance.SetVolume(value, AudioManager.AudioChannel.Master);
}

public void SetMusicVolume(float value)
{
    AudioManager.instance.SetVolume(value, AudioManager.AudioChannel.Music);
}

public void SetSfxVolume(float value)
{
    AudioManager.instance.SetVolume(value, AudioManager.AudioChannel.Sfx);
}
```

**Figure 14**

### *Menu Button Implementation*

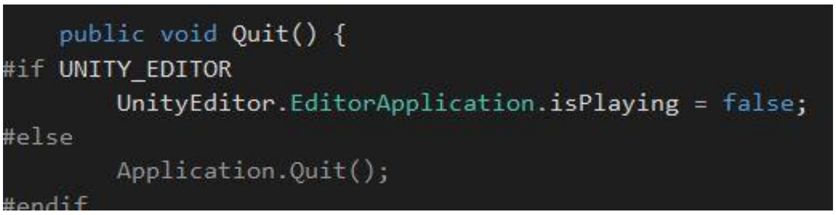
This is the select bit of code used to implement the button clicks on selection to pick one of the many options of the game menu like the previous options menu.



```
    private bool buttonSelected;
    void Start () {

    }

    // Update is called once per frame
    void Update () {
    if (Input.GetAxisRaw("Vertical") !=0 && buttonSelected == false)
        {
            eventSystem.SetSelectedGameObject(selectedObject);
            buttonSelected = true;
        }
    }
    private void OnDisable() {
        buttonSelected = false;
    }
```

**Figure 15**

The implementation to quit the game is quite simple and is set to check initially are you playing in Unity editor as the implementation is different for both and makes development easier.



```
    public void Quit() {
#if UNITY_EDITOR
        UnityEditor.EditorApplication.isPlaying = false;
#else
        Application.Quit();
#endif
```

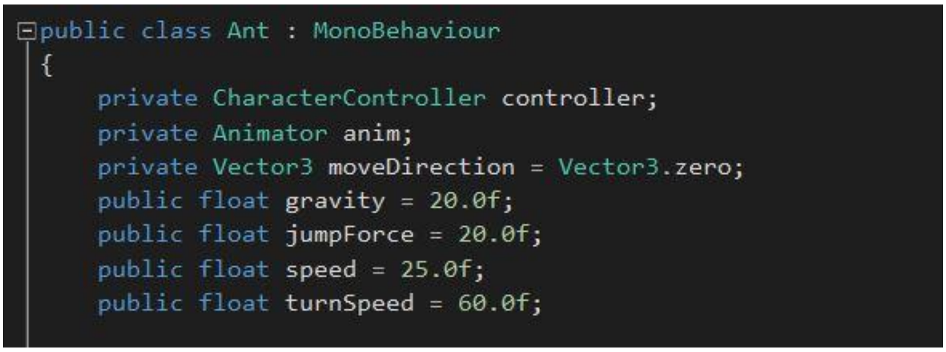
**Figure 16**

### *Main Character Implementation*

With the main character there are certain things which need to be implemented:

- Movement
- Shooting
- Health

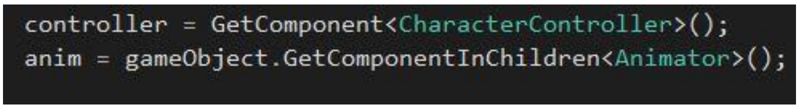
The first thing we need to look at is the code for the character controller.



```
public class Ant : MonoBehaviour
{
    private CharacterController controller;
    private Animator anim;
    private Vector3 moveDirection = Vector3.zero;
    public float gravity = 20.0f;
    public float jumpForce = 20.0f;
    public float speed = 25.0f;
    public float turnSpeed = 60.0f;
```

**Figure 17**

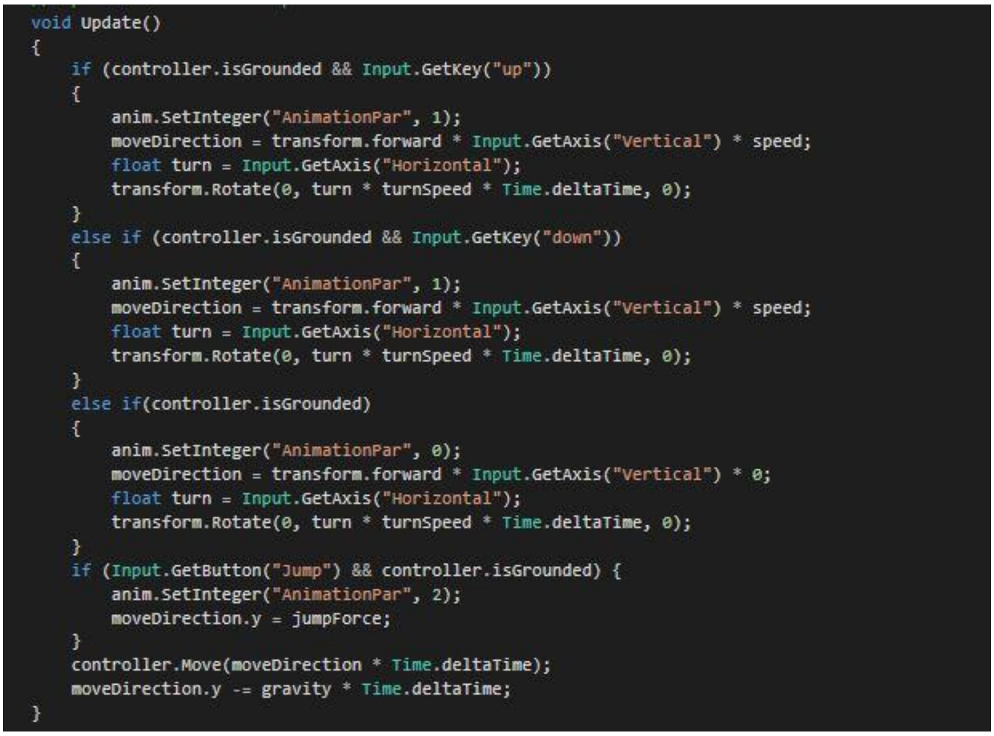
As you can see I have made all the characteristics public, this is so in the Unity editor you can alter these numbers without the need to go back to visual studio and alter them, this means you can develop at a faster rate.



```
controller = GetComponent<CharacterController>();
anim = gameObject.GetComponentInChildren<Animator>();
```

**Figure 18**

As you can see this is connecting to the character controller object used in Unity designed to allow the user to control the player character.

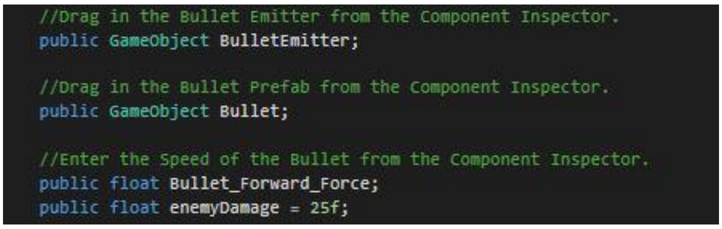


```
void Update()
{
    if (controller.isGrounded && Input.GetKey("up"))
    {
        anim.SetInteger("AnimationPar", 1);
        moveDirection = transform.forward * Input.GetAxis("Vertical") * speed;
        float turn = Input.GetAxis("Horizontal");
        transform.Rotate(0, turn * turnSpeed * Time.deltaTime, 0);
    }
    else if (controller.isGrounded && Input.GetKey("down"))
    {
        anim.SetInteger("AnimationPar", 1);
        moveDirection = transform.forward * Input.GetAxis("Vertical") * speed;
        float turn = Input.GetAxis("Horizontal");
        transform.Rotate(0, turn * turnSpeed * Time.deltaTime, 0);
    }
    else if(controller.isGrounded)
    {
        anim.SetInteger("AnimationPar", 0);
        moveDirection = transform.forward * Input.GetAxis("Vertical") * 0;
        float turn = Input.GetAxis("Horizontal");
        transform.Rotate(0, turn * turnSpeed * Time.deltaTime, 0);
    }
    if (Input.GetButton("Jump") && controller.isGrounded) {
        anim.SetInteger("AnimationPar", 2);
        moveDirection.y = jumpForce;
    }
    controller.Move(moveDirection * Time.deltaTime);
    moveDirection.y -= gravity * Time.deltaTime;
}
```

**Figure 19**

In the code above is partly the control function and also to call the animations on pressing a key, if the player is idle he play's the idle animation, but on moving walk animation runs. This is the same with the jump animation if you press the spacebar.

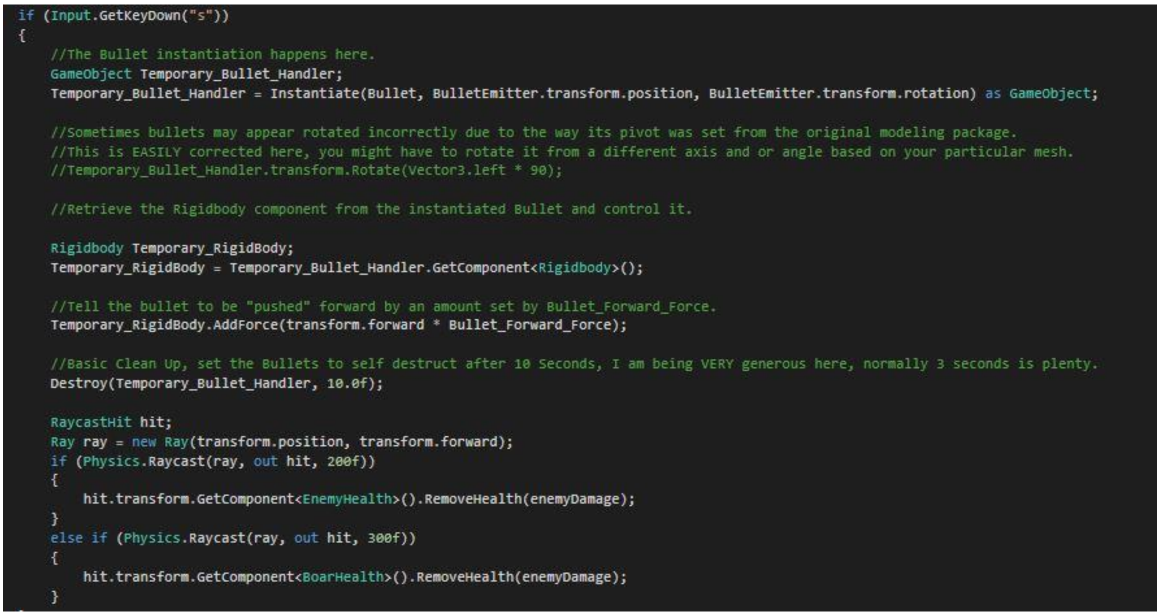
## Main Character Weapon Implementation



```
//Drag in the Bullet Emitter from the Component Inspector.
public GameObject BulletEmitter;

//Drag in the Bullet Prefab from the Component Inspector.
public GameObject Bullet;

//Enter the Speed of the Bullet from the Component Inspector.
public float Bullet_Forward_Force;
public float enemyDamage = 25f;
```

**Figure 20**

In the code above is setting up the players weapon by using an empty game object which was named BulletEmitter, this then fired an object from this point as a bullet. Both the force and the damage that would be done was set up here, but this was linked to a raycast.



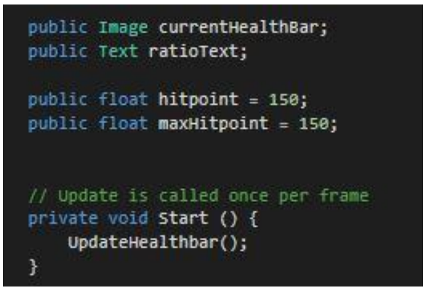
```
if (Input.GetKeyDown("s"))
{
    //The Bullet instantiation happens here.
    GameObject Temporary_Bullet_Handler;
    Temporary_Bullet_Handler = Instantiate(Bullet, BulletEmitter.transform.position, BulletEmitter.transform.rotation) as GameObject;

    //Sometimes bullets may appear rotated incorrectly due to the way its pivot was set from the original modeling package.
    //This is EASILY corrected here, you might have to rotate it from a different axis and or angle based on your particular mesh.
    //Temporary_Bullet_Handler.transform.Rotate(Vector3.left * 90);

    //Retrieve the Rigidbody component from the instantiated Bullet and control it.

    Rigidbody Temporary_RigidBody;
    Temporary_RigidBody = Temporary_Bullet_Handler.GetComponent<Rigidbody>();

    //Tell the bullet to be "pushed" forward by an amount set by Bullet_Forward_Force.
    Temporary_RigidBody.AddForce(transform.forward * Bullet_Forward_Force);

    //Basic Clean Up, set the Bullets to self destruct after 10 Seconds, I am being VERY generous here, normally 3 seconds is plenty.
    Destroy(Temporary_Bullet_Handler, 10.0f);

    RaycastHit hit;
    Ray ray = new Ray(transform.position, transform.forward);
    if (Physics.Raycast(ray, out hit, 200f))
    {
        hit.transform.GetComponent<EnemyHealth>().RemoveHealth(enemyDamage);
    }
    else if (Physics.Raycast(ray, out hit, 300f))
    {
        hit.transform.GetComponent<BoarHealth>().RemoveHealth(enemyDamage);
    }
```

**Figure 21**

In the code above we set the button used to fire the projectile which was set in the previous lines of code when pushing the (S) button as you can see the projectile is set to be temporary and disappear after 10 seconds in this instance but this has been changed as to not clutter the games memory.

We also set the raycast which is actually the way the damage is dealt to the enemy character, this basically like a laser which hit's the enemy and deducts health from their associated health class.

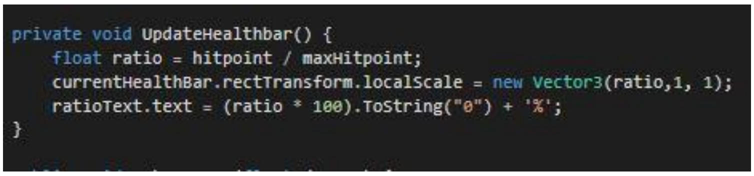
## Main Character Health Implementation



```
public Image currentHealthBar;
public Text ratioText;

public float hitpoint = 150;
public float maxHitpoint = 150;


// Update is called once per frame
private void Start () {
    UpdateHealthbar();
}
```

**Figure 22**

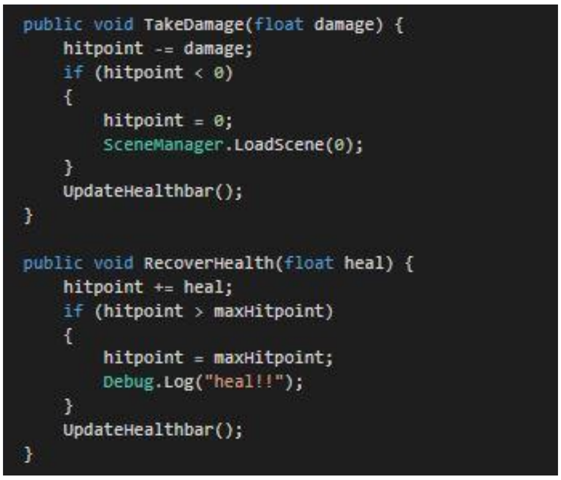
In the code above we set visual healthbar to an image which the player can see and the ratioText gives you a percentage value of the healthbar, we set the maximum possible health value for character as 150 and start it with this.



```
private void UpdateHealthbar() {
    float ratio = hitpoint / maxHitpoint;
    currentHealthBar.rectTransform.localScale = new Vector3(ratio,1, 1);
    ratioText.text = (ratio * 100).ToString("0") + '%';
}
```

**Figure 23**

These lines of code are designed to alter the visualisation of the healthbar along with changing the percentage values of the text.

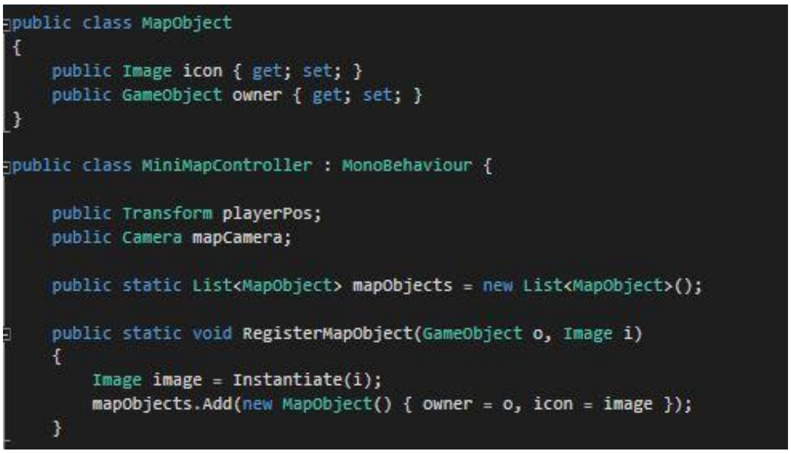


```
public void TakeDamage(float damage) {
    hitpoint -= damage;
    if (hitpoint < 0)
    {
        hitpoint = 0;
        SceneManager.LoadScene(0);
    }
    UpdateHealthbar();
}

public void RecoverHealth(float heal) {
    hitpoint += heal;
    if (hitpoint > maxHitpoint)
    {
        hitpoint = maxHitpoint;
        Debug.Log("heal!!");
    }
    UpdateHealthbar();
}
```

**Figure 24**

In these lines we set out that if the character's life is extinguished we then go back to the replay menu, it also makes sure the healthbar is updated throughout. We also have the ability to recover health and this then adds to the bar.

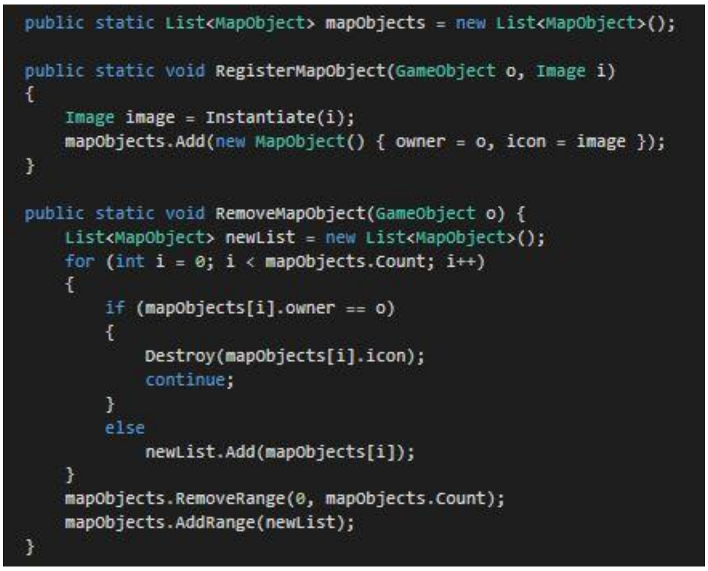
## MiniMap Implementation



```
public class MapObject
{
    public Image icon { get; set; }
    public GameObject owner { get; set; }
}

public class MiniMapController : MonoBehaviour {

    public Transform playerPos;
    public Camera mapCamera;

    public static List<MapObject> mapObjects = new List<MapObject>();

    public static void RegisterMapObject(GameObject o, Image i)
    {
        Image image = Instantiate(i);
        mapObjects.Add(new MapObject() { owner = o, icon = image });
    }
```

**Figure 25**

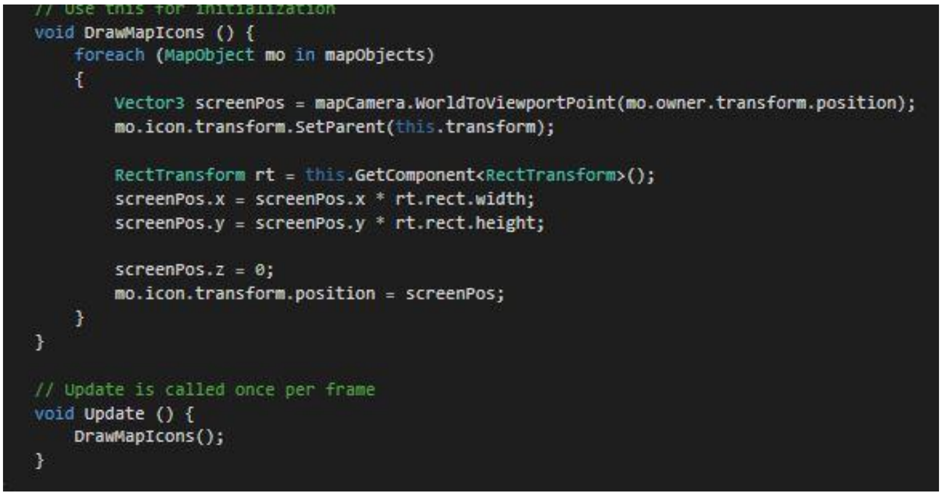
Inside this class we create a class called map object this holds the image which shows on the map which is attached to the enemy character and this moves on the map as the enemy move, we also attach a camera to this map and attach it to the player so things change as they move.



```
public static List<MapObject> mapObjects = new List<MapObject>();

public static void RegisterMapObject(GameObject o, Image i)
{
    Image image = Instantiate(i);
    mapObjects.Add(new MapObject() { owner = o, icon = image });
}

public static void RemoveMapObject(GameObject o) {
    List<MapObject> newList = new List<MapObject>();
    for (int i = 0; i < mapObjects.Count; i++)
    {
        if (mapObjects[i].owner == o)
        {
            Destroy(mapObjects[i].icon);
            continue;
        }
        else
            newList.Add(mapObjects[i]);
    }
    mapObjects.RemoveRange(0, mapObjects.Count);
    mapObjects.AddRange(newList);
}
```

**Figure 26**

This is the bit of code which puts the objects into a list and attaches them to a specific enemy character, this also deletes the image once that enemy character has been deleted.



```
// Use this for initialization
void DrawMapIcons () {
    foreach (MapObject mo in mapObjects)
    {
        Vector3 screenPos = mapCamera.WorldToViewportPoint(mo.owner.transform.position);
        mo.icon.transform.SetParent(this.transform);

        RectTransform rt = this.GetComponent<RectTransform>();
        screenPos.x = screenPos.x * rt.rect.width;
        screenPos.y = screenPos.y * rt.rect.height;

        screenPos.z = 0;
        mo.icon.transform.position = screenPos;
    }
}

// Update is called once per frame
void Update () {
    DrawMapIcons();
}
```

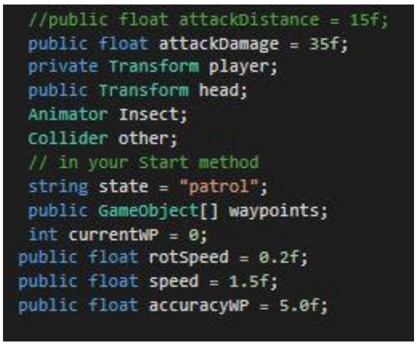
**Figure 27**

In these lines of code it actually projects the images to the map and allows you see where the enemy character are in comparison to the player. These are set on a 3 dimensional figure in the xyz axis.

## *Main Enemy Implementation*

In this section I will be outlining the code used to create the main enemy character which you meet in the first level.

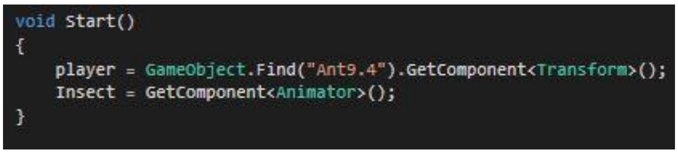
In the code below we initially set out the amount of damage the enemy can do to the player character with each attack. These are made public again to make altering these in the Unity editor easier. We set our target as the player and we initialise the animator. We set the collider so as to do damage and initialise move for the AI through waypoints.



```
  //public float attackDistance = 15f;
  public float attackDamage = 35f;
  private Transform player;
  public Transform head;
  Animator Insect;
  Collider other;
  // in your Start method
  string state = "patrol";
  public GameObject[] waypoints;
  int currentWP = 0;
public float rotSpeed = 0.2f;
public float speed = 1.5f;
public float accuracyWP = 5.0f;
```

**Figure 28**

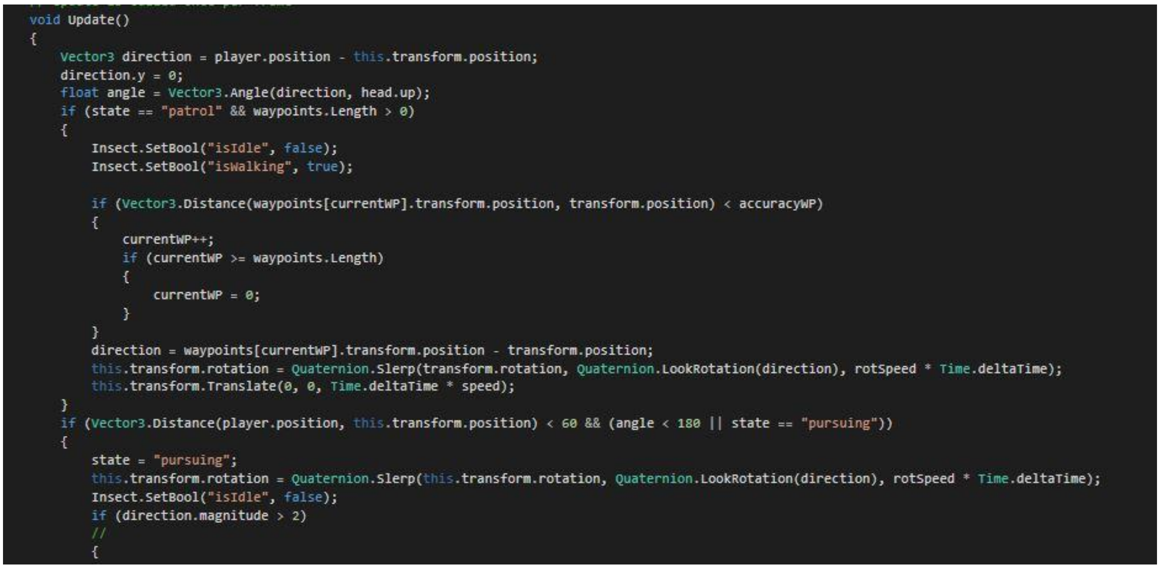
In these 2 lines of code we set our target as (ant9.4) this is the player character and initialise the animator.



```
void Start()
{
    player = GameObject.Find("Ant9.4").GetComponent<Transform>();
    Insect = GetComponent<Animator>();
}
```

**Figure 29**

The code below sets out the patrolling and the animation which should run while the enemy character patrols around the waypoints.

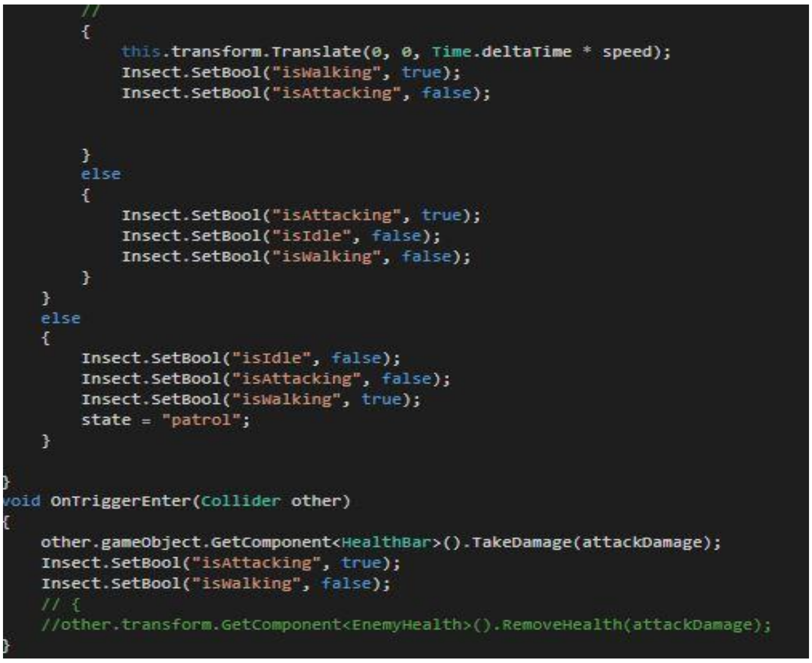


```
    void Update()
    {
        Vector3 direction = player.position - this.transform.position;
        direction.y = 0;
        float angle = Vector3.Angle(direction, head.up);
        if (state == "patrol" && waypoints.Length > 0)
        {
            Insect.SetBool("isIdle", false);
            Insect.SetBool("isWalking", true);

            if (Vector3.Distance(waypoints[currentWP].transform.position, transform.position) < accuracyWP)
            {
                currentWP++;
                if (currentWP >= waypoints.Length)
                {
                    currentWP = 0;
                }
            }
            direction = waypoints[currentWP].transform.position - transform.position;
            this.transform.rotation = Quaternion.Slerp(transform.rotation, Quaternion.LookRotation(direction), rotSpeed * Time.deltaTime);
            this.transform.Translate(0, 0, Time.deltaTime * speed);
        }
        if (Vector3.Distance(player.position, this.transform.position) < 60 && (angle < 180 || state == "pursuing"))
        {
            state = "pursuing";
            this.transform.rotation = Quaternion.Slerp(this.transform.rotation, Quaternion.LookRotation(direction), rotSpeed * Time.deltaTime);
            Insect.SetBool("isIdle", false);
            if (direction.magnitude > 2)
            //
            {
```

**Figure 30**

The bottom lines of this code initiate the pursuit of the friendly character, if the player character is within 180 degrees of the front of the enemy and within a distance of 60f it will initiate the pursuit. The attack animation is triggered if the player is within a certain distance of the enemy.

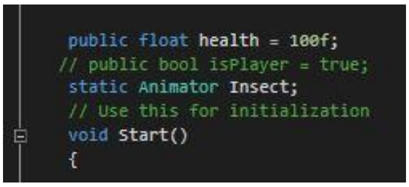
In the image below we set a collider which if it makes contact with the player character, it makes a call to the players healthbar script and applies the subsequent damage.



```
        //
        {
            this.transform.Translate(0, 0, Time.deltaTime * speed);
            Insect.SetBool("isWalking", true);
            Insect.SetBool("isAttacking", false);


        }
        else
        {
            Insect.SetBool("isAttacking", true);
            Insect.SetBool("isIdle", false);
            Insect.SetBool("isWalking", false);
        }
    }
    else
    {
        Insect.SetBool("isIdle", false);
        Insect.SetBool("isAttacking", false);
        Insect.SetBool("isWalking", true);
        state = "patrol";
    }

}
void OnTriggerEnter(Collider other)
{
    other.gameObject.GetComponent<HealthBar>().TakeDamage(attackDamage);
    Insect.SetBool("isAttacking", true);
    Insect.SetBool("isWalking", false);
    // {
    //other.transform.GetComponent<EnemyHealth>().RemoveHealth(attackDamage);
}
```

**Figure 31**

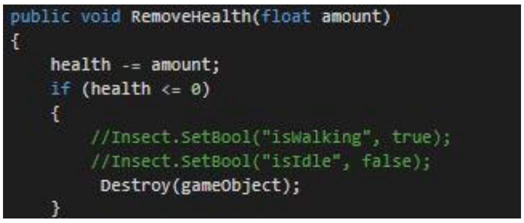
We set the enemies health to 100, but make it public if we want to change it in Unity.



```
    public float health = 100f;
   // public bool isPlayer = true;
    static Animator Insect;
    // Use this for initialization
    void Start()
    {
```

**Figure 32**

When the enemy character dies, we just destroy the object rather than taking up system memory.

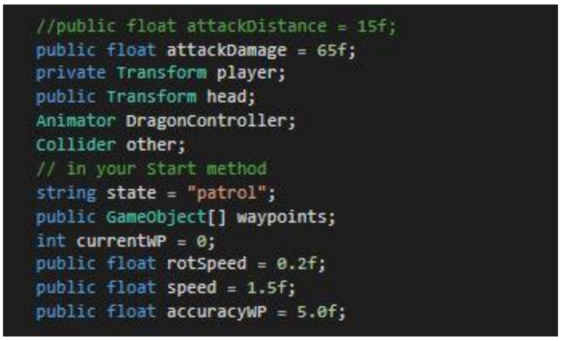


```
public void RemoveHealth(float amount)
{
    health -= amount;
    if (health <= 0)
    {
        //Insect.SetBool("isWalking", true);
        //Insect.SetBool("isIdle", false);
         Destroy(gameObject);
    }
```

**Figure 33**

### *Boar Attack Implementation*

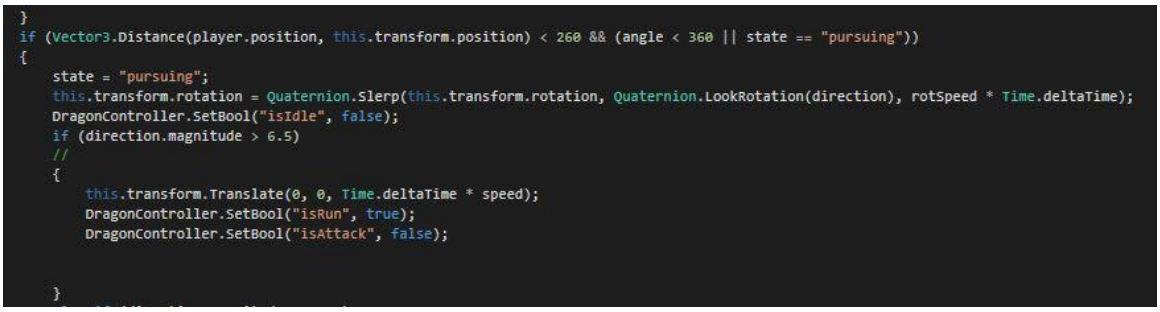
The difference between the implementation of this AI is the animator is called something different altogether.



```
   //public float attackDistance = 15f;
   public float attackDamage = 65f;
   private Transform player;
   public Transform head;
   Animator DragonController;
   Collider other;
   // in your Start method
   string state = "patrol";
   public GameObject[] waypoints;
   int currentWP = 0;
   public float rotSpeed = 0.2f;
   public float speed = 1.5f;
   public float accuracyWP = 5.0f;
```

**Figure 34**

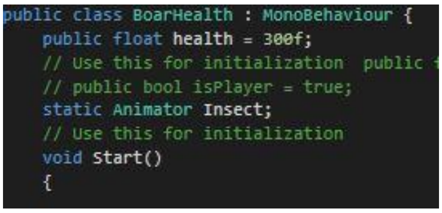
As you can see the enemy boar can detect players in a 360 degree radius and also a further distance of 260f compared to main AI which only has 180 degree with vision of only 60f.



```
}
if (Vector3.Distance(player.position, this.transform.position) < 260 && (angle < 360 || state == "pursuing"))
{
    state = "pursuing";
    this.transform.rotation = Quaternion.Slerp(this.transform.rotation, Quaternion.LookRotation(direction), rotSpeed * Time.deltaTime);
    DragonController.SetBool("isIdle", false);
    if (direction.magnitude > 6.5)
    //
    {
        this.transform.Translate(0, 0, Time.deltaTime * speed);
        DragonController.SetBool("isRun", true);
        DragonController.SetBool("isAttack", false);


    }
```

**Figure 35**

The boar's health is set higher to 3 times higher than the main enemy character and his attack is also set higher and will do more damage to the player character.

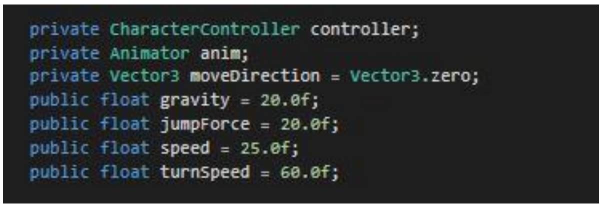


```
public class BoarHealth : MonoBehaviour {
    public float health = 300f;
    // Use this for initialization  public
    // public bool isPlayer = true;
    static Animator Insect;
    // Use this for initialization
    void Start()
    {
```

**Figure 36**

## *BeetleTank Implementation*

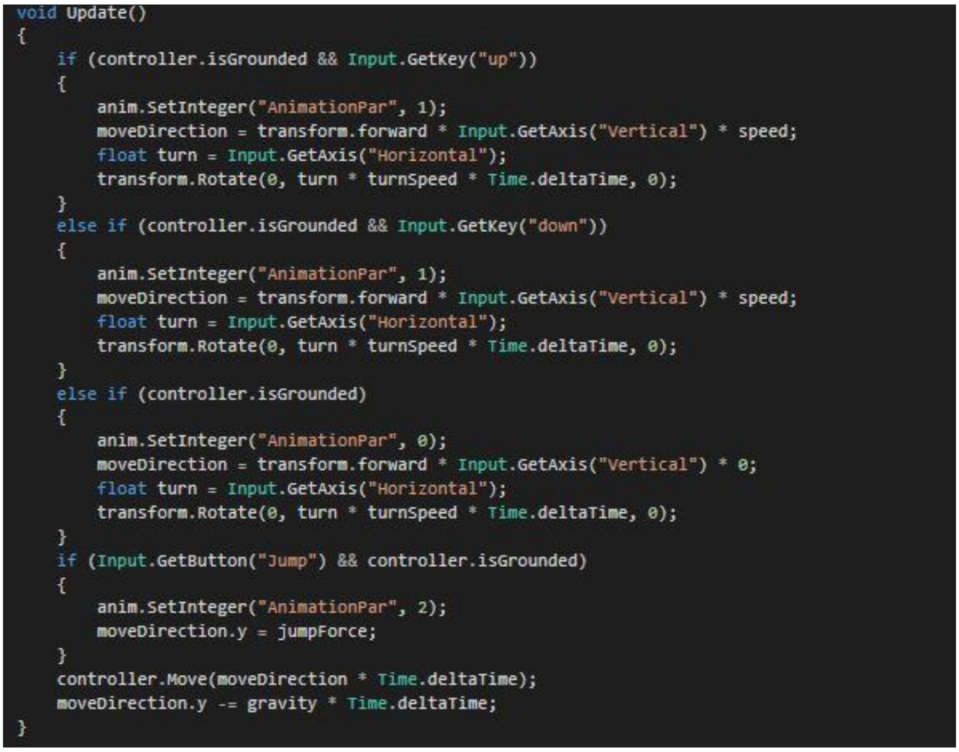
This is a look at the code used to implement the BeetleTank which is the vehicle used in the game. As you can see in the code below the code is open to alteration with Unity developer.



```
private CharacterController controller;
private Animator anim;
private Vector3 moveDirection = Vector3.zero;
public float gravity = 20.0f;
public float jumpForce = 20.0f;
public float speed = 25.0f;
public float turnSpeed = 60.0f;
```

**Figure 37**

The implementation of the controls is the same as the main character

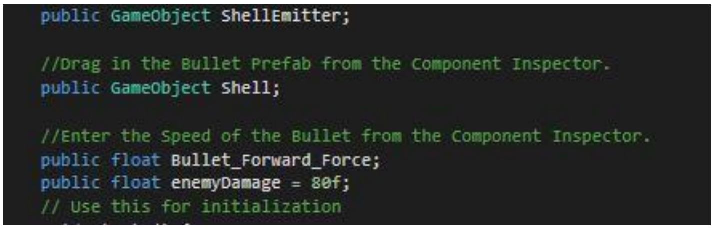


```
void Update()
{
    if (controller.isGrounded && Input.GetKey("up"))
    {
        anim.SetInteger("AnimationPar", 1);
        moveDirection = transform.forward * Input.GetAxis("Vertical") * speed;
        float turn = Input.GetAxis("Horizontal");
        transform.Rotate(0, turn * turnSpeed * Time.deltaTime, 0);
    }
    else if (controller.isGrounded && Input.GetKey("down"))
    {
        anim.SetInteger("AnimationPar", 1);
        moveDirection = transform.forward * Input.GetAxis("Vertical") * speed;
        float turn = Input.GetAxis("Horizontal");
        transform.Rotate(0, turn * turnSpeed * Time.deltaTime, 0);
    }
    else if (controller.isGrounded)
    {
        anim.SetInteger("AnimationPar", 0);
        moveDirection = transform.forward * Input.GetAxis("Vertical") * 0;
        float turn = Input.GetAxis("Horizontal");
        transform.Rotate(0, turn * turnSpeed * Time.deltaTime, 0);
    }
    if (Input.GetButton("Jump") && controller.isGrounded)
    {
        anim.SetInteger("AnimationPar", 2);
        moveDirection.y = jumpForce;
    }
    controller.Move(moveDirection * Time.deltaTime);
    moveDirection.y -= gravity * Time.deltaTime;
}
```

**Figure 38**

### Implementing BeetleTank Weapon

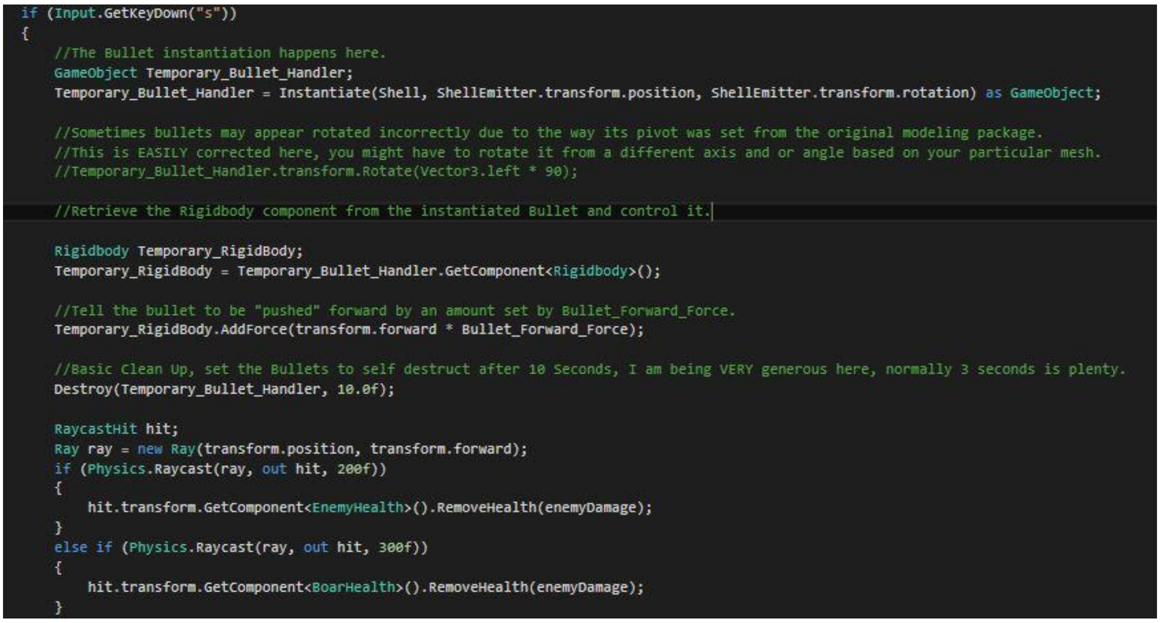
Instead of firing a bullet obviously, we fire a shell from the shell Emitter.



```
    public GameObject ShellEmitter;

    //Drag in the Bullet Prefab from the Component Inspector.
    public GameObject Shell;

    //Enter the Speed of the Bullet from the Component Inspector.
    public float Bullet_Forward_Force;
    public float enemyDamage = 80f;
    // Use this for initialization
```

**Figure 39**

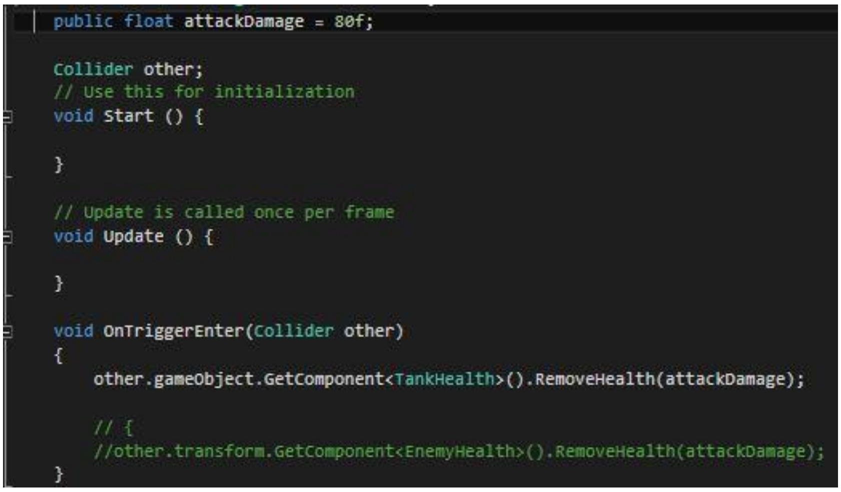
We use a raycast to do damage to the enemy insect and boar but we implement a different thing to attack enemy BeetleTank.



```
if (Input.GetKeyDown("s"))
{
    //The Bullet instantiation happens here.
    GameObject Temporary_Bullet_Handler;
    Temporary_Bullet_Handler = Instantiate(Shell, ShellEmitter.transform.position, ShellEmitter.transform.rotation) as GameObject;

    //Sometimes bullets may appear rotated incorrectly due to the way its pivot was set from the original modeling package.
    //This is EASILY corrected here, you might have to rotate it from a different axis and or angle based on your particular mesh.
    //Temporary_Bullet_Handler.transform.Rotate(Vector3.left * 90);

    //Retrieve the Rigidbody component from the instantiated Bullet and control it.

    Rigidbody Temporary_RigidBody;
    Temporary_RigidBody = Temporary_Bullet_Handler.GetComponent<Rigidbody>();

    //Tell the bullet to be "pushed" forward by an amount set by Bullet_Forward_Force.
    Temporary_RigidBody.AddForce(transform.forward * Bullet_Forward_Force);

    //Basic Clean Up, set the Bullets to self destruct after 10 Seconds, I am being VERY generous here, normally 3 seconds is plenty.
    Destroy(Temporary_Bullet_Handler, 10.0f);

    RaycastHit hit;
    Ray ray = new Ray(transform.position, transform.forward);
    if (Physics.Raycast(ray, out hit, 200f))
    {
        hit.transform.GetComponent<EnemyHealth>().RemoveHealth(enemyDamage);
    }
    else if (Physics.Raycast(ray, out hit, 300f))
    {
        hit.transform.GetComponent<BoarHealth>().RemoveHealth(enemyDamage);
    }
```

**Figure 40**

We cause damage to enemy tanks by adding this script to the object emitted by the cannon and this effects the enemy when the object collides with the enemy object.

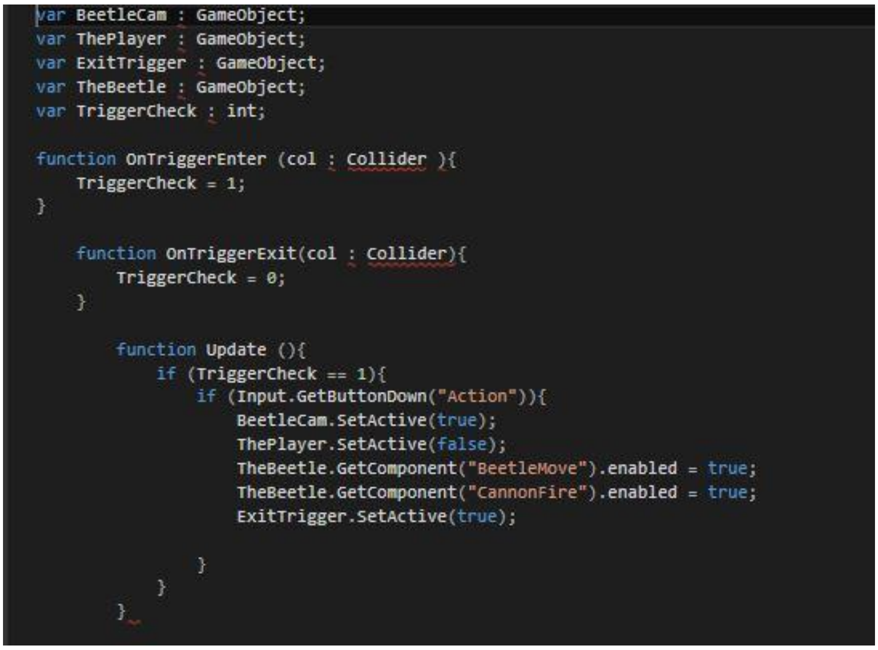


```
    public float attackDamage = 80f;

    Collider other;
    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    void OnTriggerEnter(Collider other)
    {
        other.gameObject.GetComponent<TankHealth>().RemoveHealth(attackDamage);

        // {
        //other.transform.GetComponent<EnemyHealth>().RemoveHealth(attackDamage);
    }
```

**Figure 41**

## Implementing BeetleTank Enter/Exit

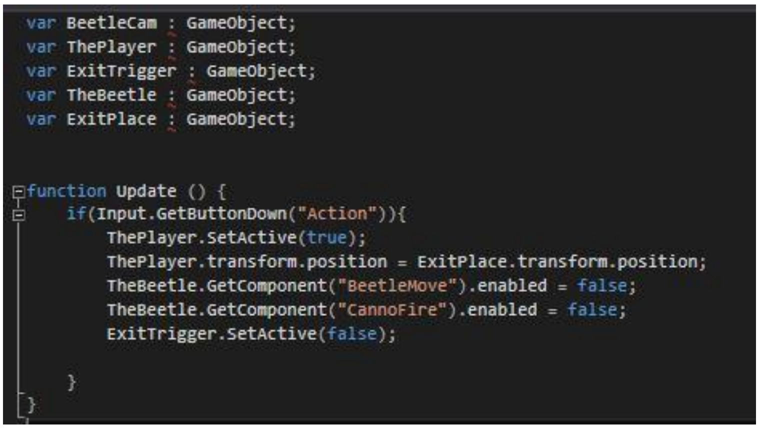
JavaScript was used to implement entering the BeetleTank, this is set to bring to life the beetlecam and initialise the movement of the BeetleTank. The collider is set to the side of the tank which initialises the ability to enter the tank.



```
var BeetleCam : GameObject;
var ThePlayer : GameObject;
var ExitTrigger : GameObject;
var TheBeetle : GameObject;
var TriggerCheck : int;

function OnTriggerEnter (col : Collider ){
    TriggerCheck = 1;
}

    function OnTriggerExit(col : Collider){
        TriggerCheck = 0;
    }

        function Update (){
            if (TriggerCheck == 1){
                if (Input.GetButtonDown("Action")){
                    BeetleCam.SetActive(true);
                    ThePlayer.SetActive(false);
                    TheBeetle.GetComponent("BeetleMove").enabled = true;
                    TheBeetle.GetComponent("CannonFire").enabled = true;
                    ExitTrigger.SetActive(true);


                }
            }
        }
```

**Figure 42**

In this bit of code we set everything to false after we exit the tank, so when the player is moving the character the tank will stay in one spot.



```
var BeetleCam : GameObject;
var ThePlayer : GameObject;
var ExitTrigger : GameObject;
var TheBeetle : GameObject;
var ExitPlace : GameObject;


function Update () {
    if(Input.GetButtonDown("Action")){
        ThePlayer.SetActive(true);
        ThePlayer.transform.position = ExitPlace.transform.position;
        TheBeetle.GetComponent("BeetleMove").enabled = false;
        TheBeetle.GetComponent("CannoFire").enabled = false;
        ExitTrigger.SetActive(false);

    }
}
```

**Figure 43**

### Animation Implementation

I used Blender to create the animations of my main character as their seems to be an issue with animations created in Maya, the animations where initially created by moving the character in a motion for every keyframe and anything in between would be filled in by the program. Similar to the way stop motion animation like Wallace and Gromit are created, but the software is smart enough to apply movement to the other frames.

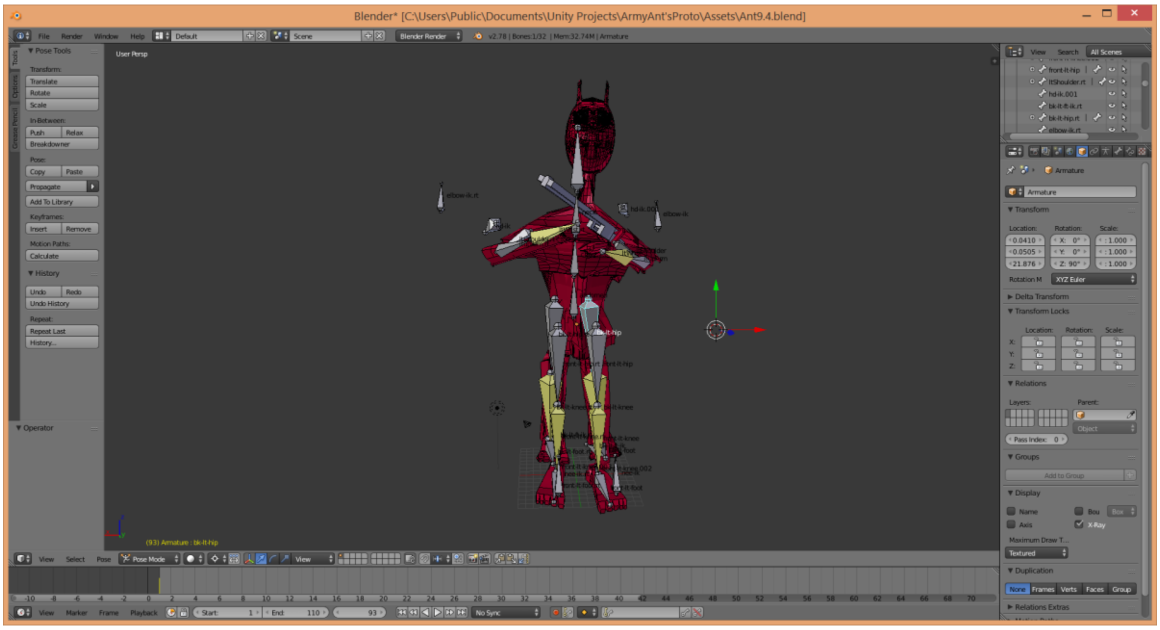


**Figure 44**

The creation of the animation transition is done in Unity itself by separating the 100+ frames into the individual animations and then we set the animation to start off in the idle state this changes when we move the character and of course when we jump the character, it runs his jump animation and this always reverts back to the idle state.

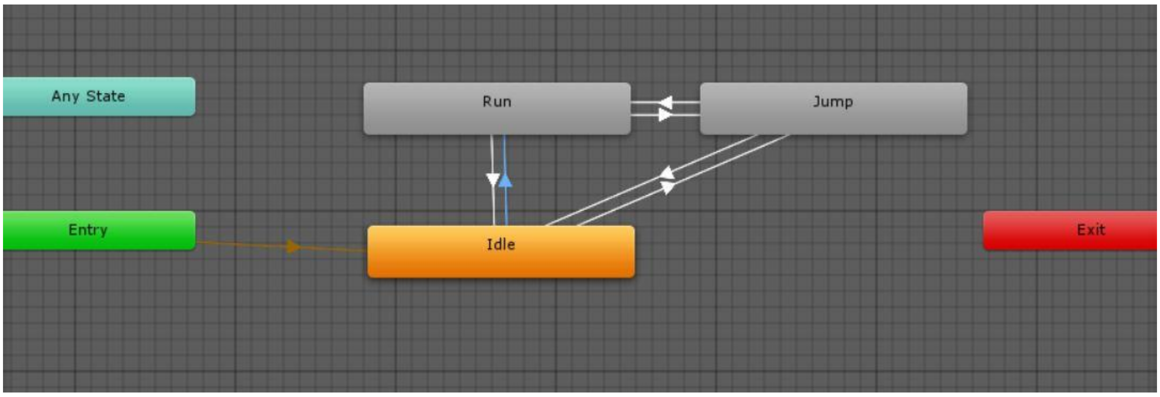
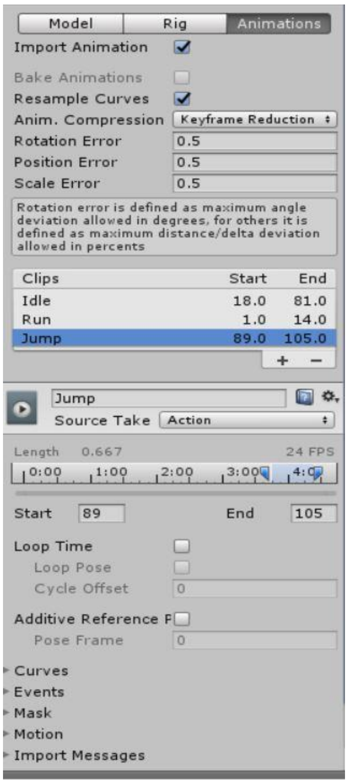


**Figure 45**



Import Animation ☑

Bake Animations □

Resample Curves ☑

Anim. Compression: Keyframe Reduction

Rotation Error: 0.5

Position Error: 0.5

Scale Error: 0.5

Rotation error is defined as maximum angle deviation allowed in degrees, for others it is defined as maximum distance/delta deviation allowed in percents

| Clips | Start | End |
|---|---|---|
| Idle | 18.0 | 81.0 |
| Run | 1.0 | 14.0 |
| Jump | 89.0 | 105.0 |

Jump

Source Take: Action

Length 0.667 24 FPS

Start 89 End 105

Loop Time □

Loop Pose □

Cycle Offset 0

Additive Reference P □

Pose Frame 0

**Figure 46**

As you can see we have all the animations for the main character set on different frames, as you may notice the jump animation is not a set loop, but the other animations are so the character keeps running through animation even if we haven't animated for that length of time.

### *Enemy Implementation*

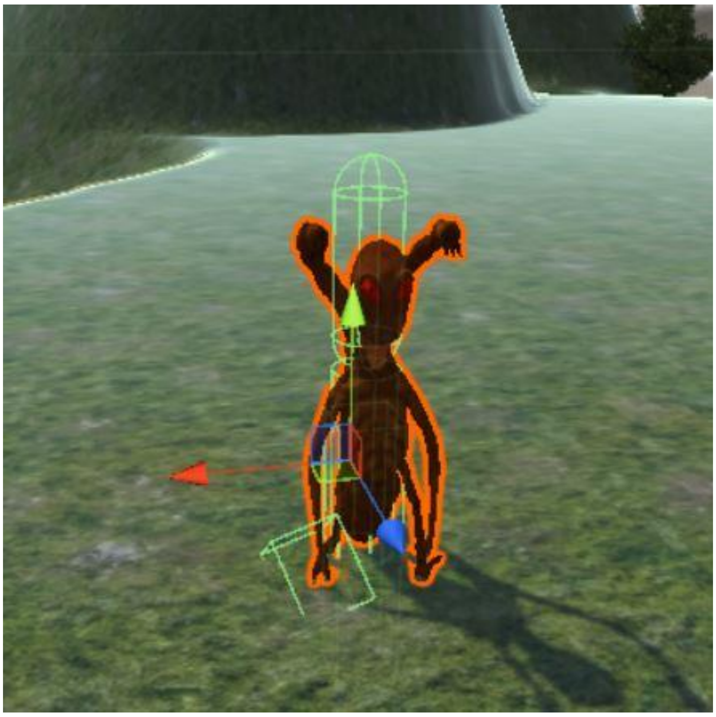


**Figure 47**

As you can see above is the main enemy insect you will notice a box collider on its right hand. If this make's contact with the player this is how it causes damage. This enemy is fairly slow and will only notice you if you are directly in front of him and only at a fairly short distance, if you move far enough away he will return to his patrolling space.

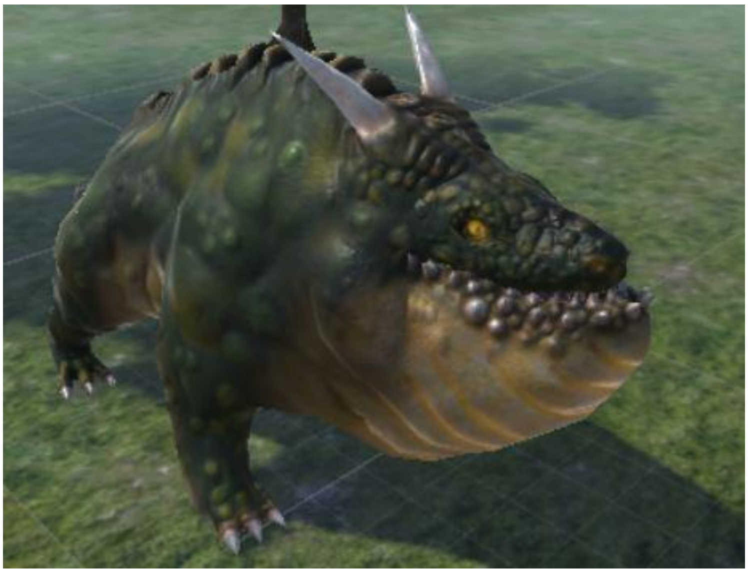


**Figure 48**

As you can see the DragonBoar is the enemies attack dog, this beast moves faster and actively hunts the player over a lot further distance if you get struck by this enemy character then it won't take much to kill you.

## *1.31 Graphical User Interface (GUI) Layout*

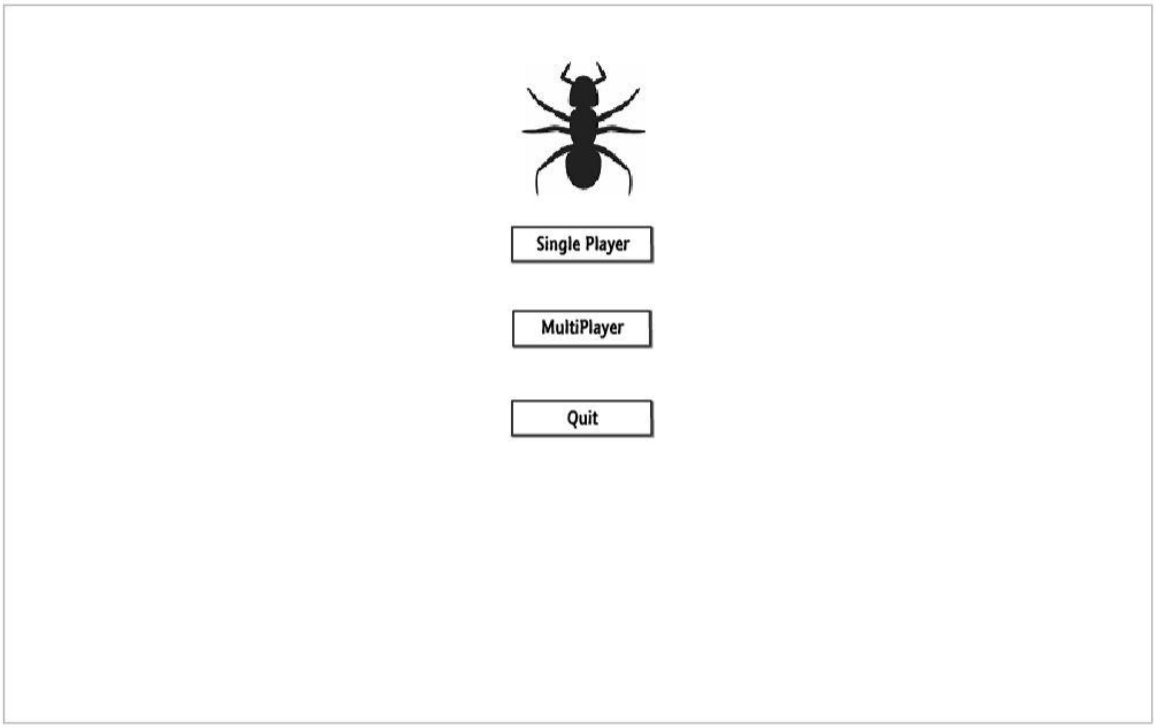


**Figure 49**

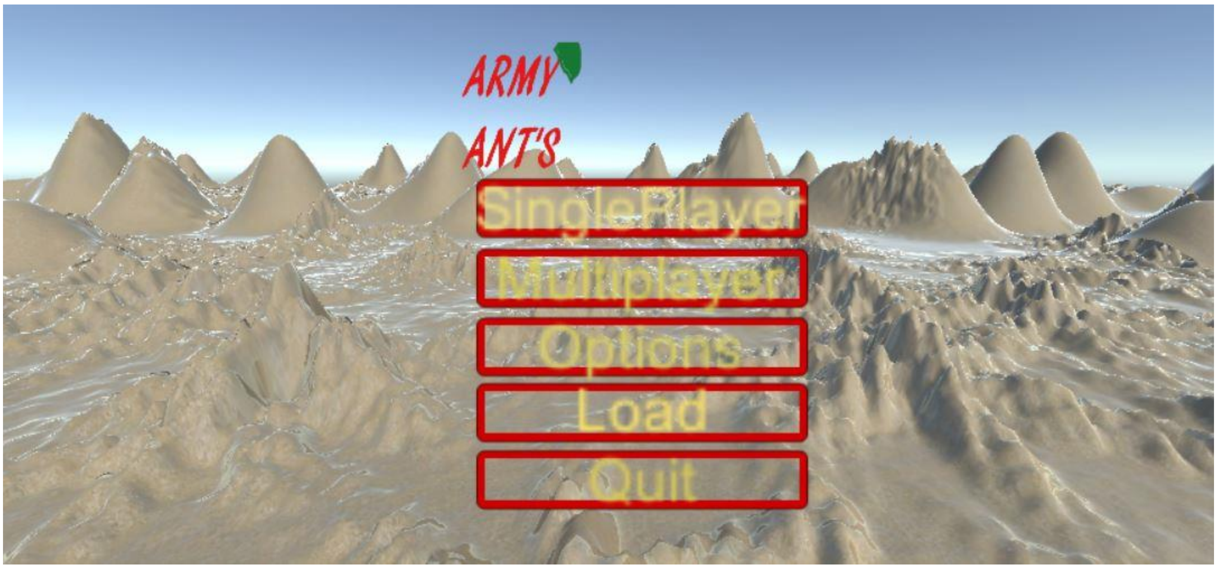


**Figure 50**

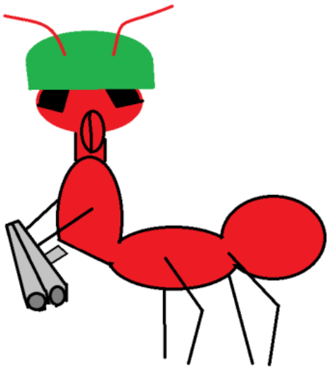


Figure 51(Concept Ant)

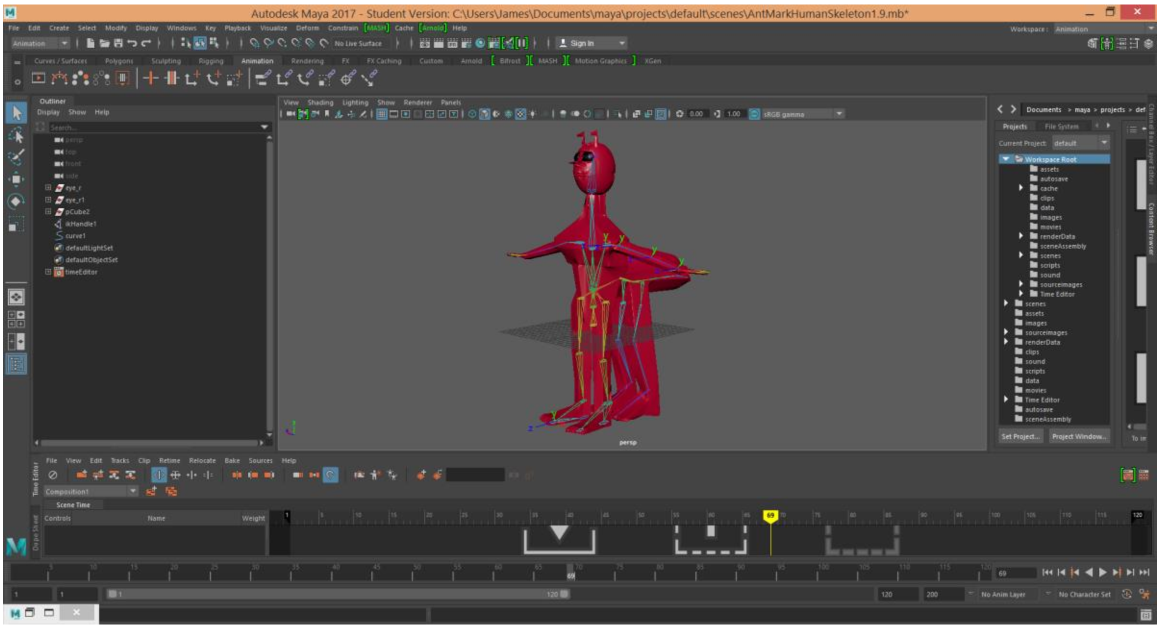

Autodesk Maya 2017 - Student Version: C:\Users\James\Documents\maya\projects\default\scenes\AntMarkHumanSkeleton1.9.mb*
File Edit Create Select Modify Display Windows Key Playback Visualize Deform Constrain MASH Cache Arnold Help
Curves / Surfaces Polygons Sculpting Rigging Animation Rendering FX FX Caching Custom Arnold Bifrost MASH Motion Graphics XGen
Outliner
persp
top
front
side
eye_r
eye_r1
pCube2
ikHandle1
curve1
defaultLightSet
defaultObjectSet
timeEditor
View Shading Lighting Show Renderer Panels
sRGB gamma
persp
Documents > maya > projects > def
Projects File System
Current Project: default
Workspace Root
assets
autosave
cache
clips
data
images
movies
renderData
sceneAssembly
scenes
scripts
sound
sourceimages
Time Editor
Set Project... Project Window...
File View Edit Tracks Clip Retime Relocate Bake Sources Help
Composition1
Scene Time
Controls Name Weight
No Anim Layer
No Character Set

Figure 52

As you can see this is the main character which was developed in Autodesk Maya, this animation and modelling suite was chosen because it is the industry standard and are forerunners in this field, but were initially only really used for movies like Wreck It Ralph and brave. But has branched into games with the Call of Duty series and with 24 the game.

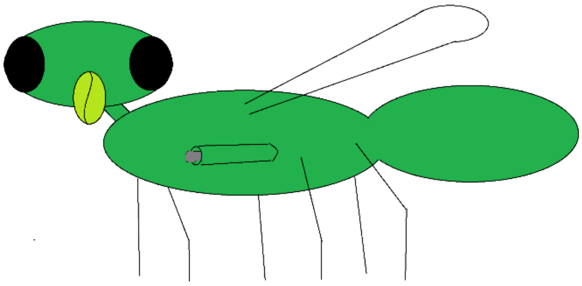


**Figure 53(Concept Helicopter)**

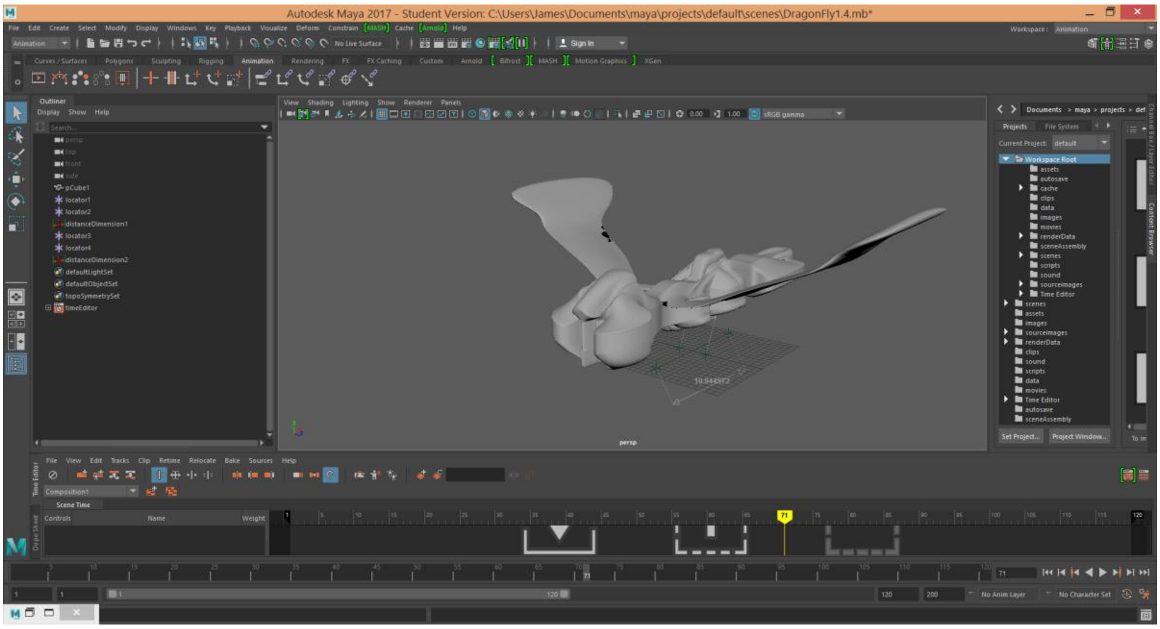


**Figure 54**

So far the development of the helicopter hasn't been completed as you can see, the main body of the character has been created but it has no armament yet and will probably not make the game in the end.

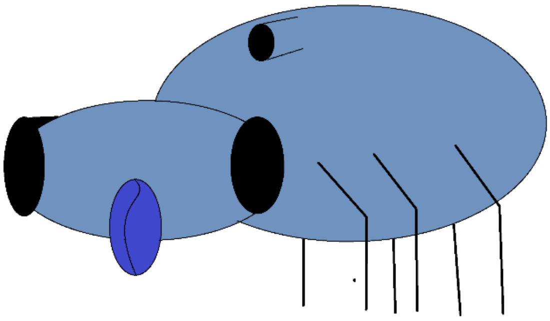


**Figure 55(Concept Tank)**

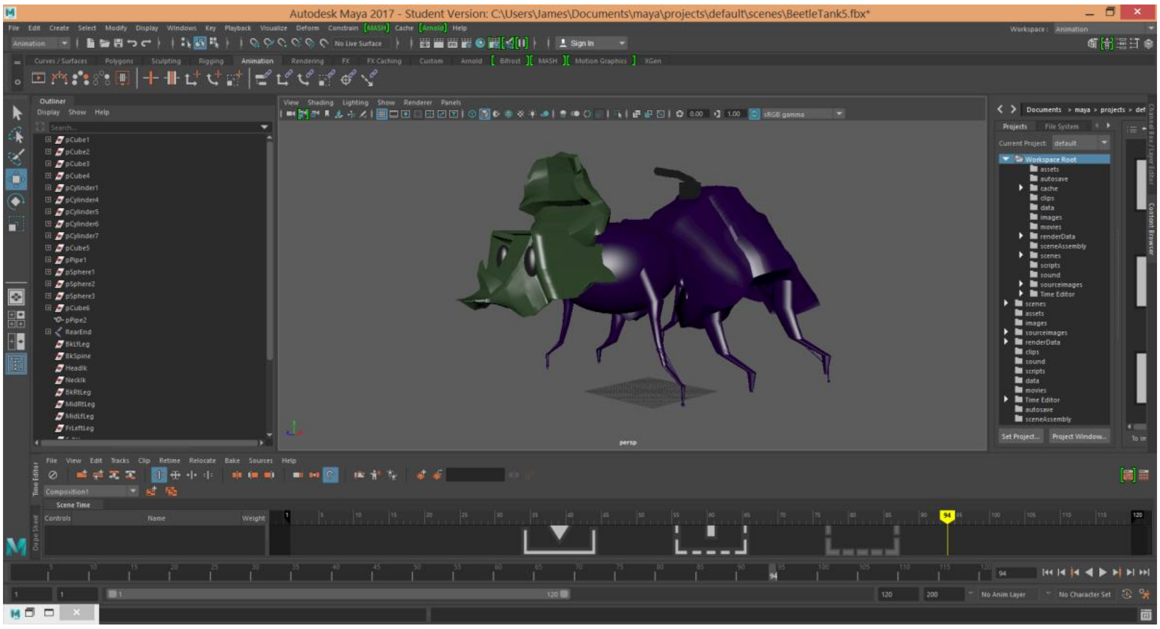

Autodesk Maya 2017 - Student Version: C:\Users\James\Documents\maya\projects\default\scenes\BeetleTank5.fbx*
File Edit Create Select Modify Display Windows Key Playback Visualize Deform Constrain MASH Cache Arnold Help
Outliner
pCube1
pCube2
pCube3
pCube4
pCylinder1
pCylinder4
pCylinder5
pCylinder6
pCylinder7
pCube5
pPipe1
pSphere1
pSphere2
pSphere3
pCube6
pPipe2
RearEnd
BkLfLeg
BkSpine
HeadIk
NeckIk
BkRtLeg
MidRtLeg
MidLfLeg
FrLeftLeg
persp
Workspace Root
Set Project... Project Window...

**Figure 56**

The BeetleTank was designed in Maya again and the animations were created in Maya as well but this will have to be redone in Blender as there was an error when importing it to Unity. This will be the main vehicle in the game and as you can see it has a cannon on the back to launch projectiles.

# 2 Game Design & Analysis

**Figure 57 (The main logo for the game.)**

## 2.1 Introduction

The game is a 3rd person shooter based in sci – fi adventure world, where you are an ant set out to save the queen who has been kidnapped by the enemy.

## 2.2 Game Analysis

This is a general overview of the game.

| Game Description | |
|---|---|
| Genre: | ▪ Adventure<br>▪ Strategy |
| Game Elements: | ▪ Shooting<br>▪ Chase<br>▪ Combat<br>▪ Dodging |

| | |
|---|---|
| Game Content: | ▪ Fantasy<br>▪ Third-Person-Shooter |
| Theme: | ▪ Sci-Fi<br>▪ Fantasy |
| Style: | ▪ Surreal |
| Game Sequence: | ▪ Linear- Storylines |
| Player: | • The Number players that can play the game at once |
| **Game Reference** | |
| Game Taxonomy: | • Game Taxonomy is here as reminder what the design direction is.<br>• The Taxonomy of the is a sci-fi based game<br>• Which is based on fiction |
| Player Immersion: | ▪ Tactical<br>▪ Strategy<br>▪ Narrative<br>▪ Emotional |
| Reference: | • References can come from anywhere.<br>• The idea is to describe your game's story, play and style with references.<br>• Using can be have negative and positive affects |
| | • |
| Game Technical | |
| Technical From: | • 3D graphics |
| View: | • Camera view the player will experience the game from behind Character<br>• More Examples: Appendix A |
| Platform: | • JavaScript, C#, |
| Device: | • PC, Console |
| Game Sales | |
| Consumer Group: | • The Consumer Group would be anybody who enjoys online gaming and fantasy games. |
| Payment: | • Through PayPal/Credit Card |
| Estimated Price: | • 15 |
| Device Support List | • PC |

### 2.3 Game Atmosphere

- Atmosphere Mood Board – the atmosphere of the game is a fast paced third person shooter designed to test your skill against multiple kinds of enemy.
- Character/ Units Sketch & Description – all these can be found further above in the document. (Fig51-56)
- A Level(Locations) Sketch & Description – levels are an open world environment designed to allow the player explore the world. (Fig70-71)
- Audio Description – a continuous play of music and when you fire your weapon you will hear a gunshot sound.

### 2.4 Game Play

- Opening the game application - this launches onto the main menu where you get to make a choice
- Game Options – allows you to change graphics and sound output of the game, also single or multiplayer
- Story Synopsis – lone army ant destined to rescue the queen who has been kidnapped
- Modes – single and multiplayer
- Game Elements – apples, enemy
- Game Levels – first level on foot taking on units that can only hit you, second level you have a tank and will battle other tanks to win.
- Player's Controls – these are the typical arrow keys for movement, S to shoot, spacebar to jump and E to enter the tank
- Winning – is when you defeat all the enemy tanks
- Losing – is dying
- End – saving the queen
- This is fun as it allows you to challenge yourself against the enemy

## 2.5 Key Features

- Number of Levels is 2
- Number of Enemies – 3 the primary enemy insect, the 2nd is the dragonboar who does more damage and the final is the enemy tank which fire's projectiles at you.
- Time of Game Play 30mins to 1 hour
- Replay ability – to improve skills and multiplayer implementation
- Audio Specifications – continuous track, Gunshot sound when fired
- Graphic Specifications
- Device Compatibility – PC and potentially console
- Number of Players – 1+
- Online Activities (high scores)
- Number/Type Modes – 1/Deathmatch

## 2.6 Selling Features

- Anthony Anterson main character
- BeetleTank
- Use of insects as vehicles

## 3 Design Document

This document describes how game objects behave, controlled and properties they have.

### 3.1 Design Version

Windows

### 3.2 Design Guidelines

Game to be developed in both C# and JavaScript as the Unity developer engine only develops in this.

### 3.3 Game Design Definitions

Issues that should be addressed here are:

- Menu – option to pick either single or multiplayer, they can also go into options to alter screen and sound effects.
- Game Play – Third person fantasy adventure based on gun and run to survive.
- You control the main character and the BeetleTank, this is with keyboard arrows, S to shoot and E to enter the BeetleTank, spacebar to jump.
- Game Over (Winning by saving the queen & Losing by dying)

### 3.4 Game Flow Chart

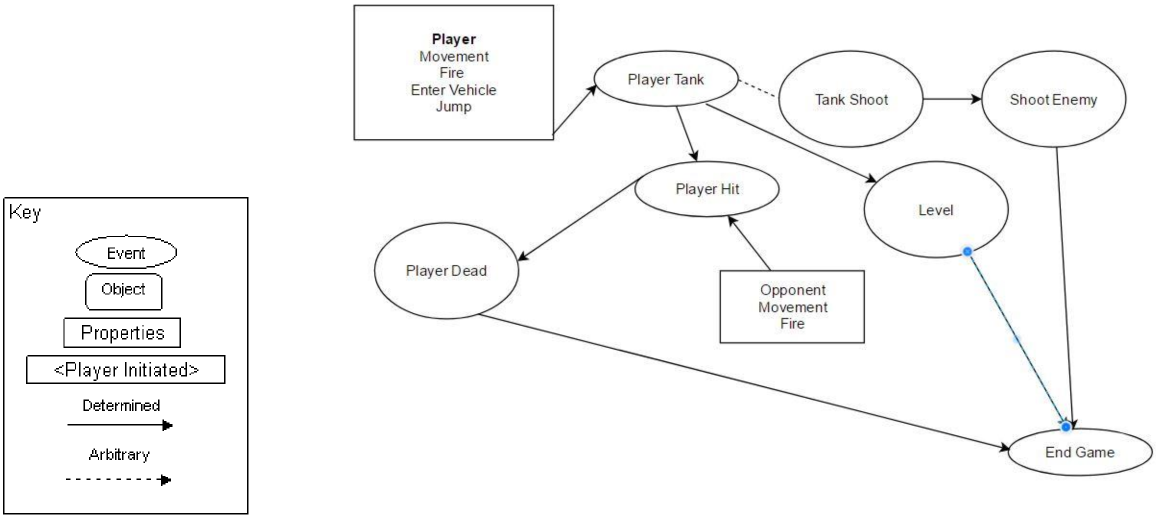


**Figure 58**

## 3.5 Player Elements

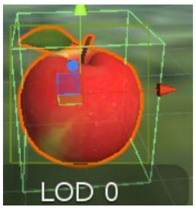


**Figure 59**

Apples directly increase player’s health by 10%

**Figure 60**

Signs are used to help players learn throughout the game.

### 3.5.1 Player Definition

A suggested list of player definitions:

- Actions: The player can run jump and shoot, they can also take advantage of the vehicles.
- Information (Status): Information is available throughout the game through signs and notifications
- Default Properties: The player  start’s off in his village with 150 hitpoints.
- Winning: The player win’s when he defeats the enemy in the second level.
- Loosing: The player loses when he dies and wishes to no longer continue.

### 3.5.2 Player Properties

A suggested list of player definitions:

- Health – health is effected by apple’s, by increasing it and it is decreased by taking damage from enemies, this can be seen by a healthbar
- Weapons – player weapons do damage to enemy characters and this can be seen after they die, you will also see an object being ejected from the weapon.
- Actions BeetleTank fires projectiles to do damage and also you can jump.

### 3.5.3 Player Rewards (Power-ups & Pick-ups)

Apple (health replenished)

When you eat this apple it is instantly destroyed and it replenishes 10% of the character's health

### 3.5.4 User Interface (UI)

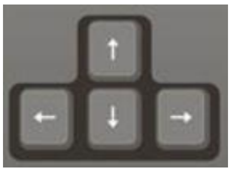


**Figure 61**

Keys to move your character.

**Figure 62**

How to jump player and tank, use (spacebar).

**Figure 63**

Use this (S) to shoot and kill enemies.

**Figure 64**

Use this (E) to enter and exit vehicles.

### 3.5.5 Heads up Display (HUD)

This is where a description of the graphics and displays will be shown.

### 3.5.6 Player View

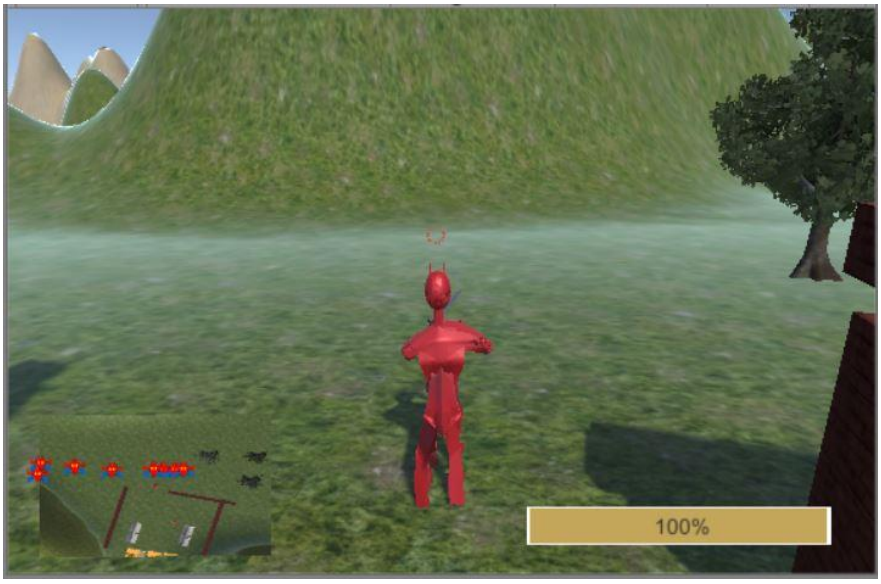


Figure 65

Camera follows behind the player rigidly.

## 3.6 Antagonistic Elements

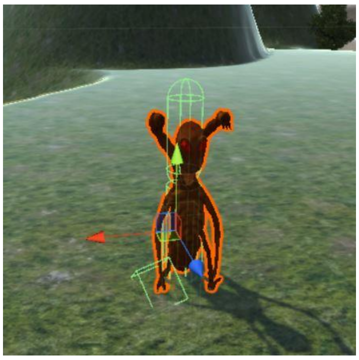


Figure 66

### 3.6.1 Antagonistic Definitions

Enemy insect is slowest of the enemies but more abundant, uses its claw to do damage, will only notice player character when directly in front of them within a 180 degrees radius.

### 3.6.2 Antagonistic Properties

Main Enemy does damage with its right claw.

### 3.6.3 Antagonistic List

Right Claw

### 3.6.4 Artificial Intelligence (AI)

This is how an AI action could be broken down:

- Normal State: Patrols specified area
- Detection State: Once player directly in front, it pursues the player
- Reaction State: once close enough it swings it's right claw to do damage
- End State: if player the destroys/kills object, it destroys it from the Gameworld

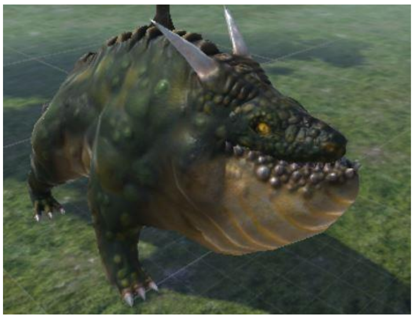


**Figure 67**

### 3.6.5 Antagonistic Definitions

DragonBoar is the second enemy you will encounter and he moves faster and also does more damage to the player character by charging you. He can also hunt you down in a larger radius of 360 degrees at a much further distance away.

### 3.6.6 Antagonistic Properties

Main Enemy does damage with its right claw.

### 3.6.7 Antagonistic List

Horn

### 3.6.8 Artificial Intelligence (AI)

This is how an AI action could be broken down:

- Normal State: Patrols specified area
- Detection State: Once player within a certain, it pursues the player
- Reaction State: once close enough it charges the player to do damage
- End State: if player the destroys/kills object, it destroys it from the Gameworld

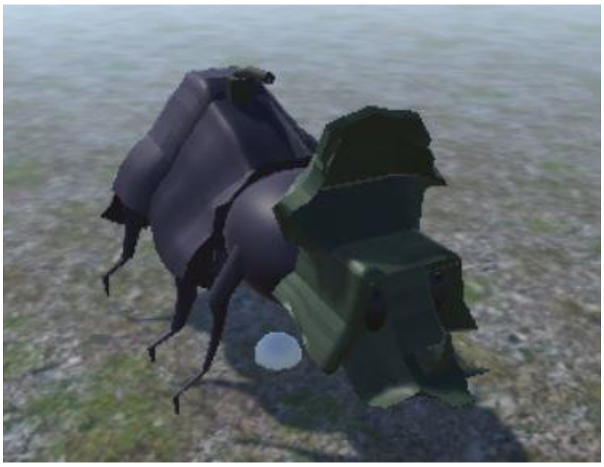


**Figure 68**

### 3.6.9 Antagonistic Definitions

The third enemy character is their own BeetleTank which is smaller, this does damage by launching a projectile in the range of the human character to do damage.

### 3.6.10 Antagonistic Properties

Cannon

### 3.6.11 Antagonistic List

Canon

Canon Shell

### 3.6.12 Artificial Intelligence (AI)

This is how an AI action could be broken down:

- Normal State: Patrols specified area
- Detection State: Once player within a certain, it pursues the player
- Reaction State: once close enough it fires a projectile at the player
- End State: if player the destroys/kills object, it destroys it from the Gameworld

## 3.7 Global Game Elements

In this game, there is only one camera which is directly behind the player, however another camera is utilized to provide the character a minimap reference of where he is in comparison to other game object's. The are 3 enemy units which try and foil your plans. You have the ability to regain health from apples, but you can't store these items and they instantly disappear on use.

## 3.8 The Story

The story is not to complex, you are colony of ant's living peacefully on your planet, when all of a sudden the evil Zeirgeist kidnap your queen for ransom.

**Figure 69**

As the hero (Anthony Anterson) you set out to rescue the queen, by taking on the characters in the world you start in, later you utilize the might of your arsenal against the enemy with your BeetleTank, which the enemy also have a smaller version of these tanks. Once you complete level 2, the beta has finished. Further vehicles will be added and utilized in later iterations.

### 3.9 Level Design

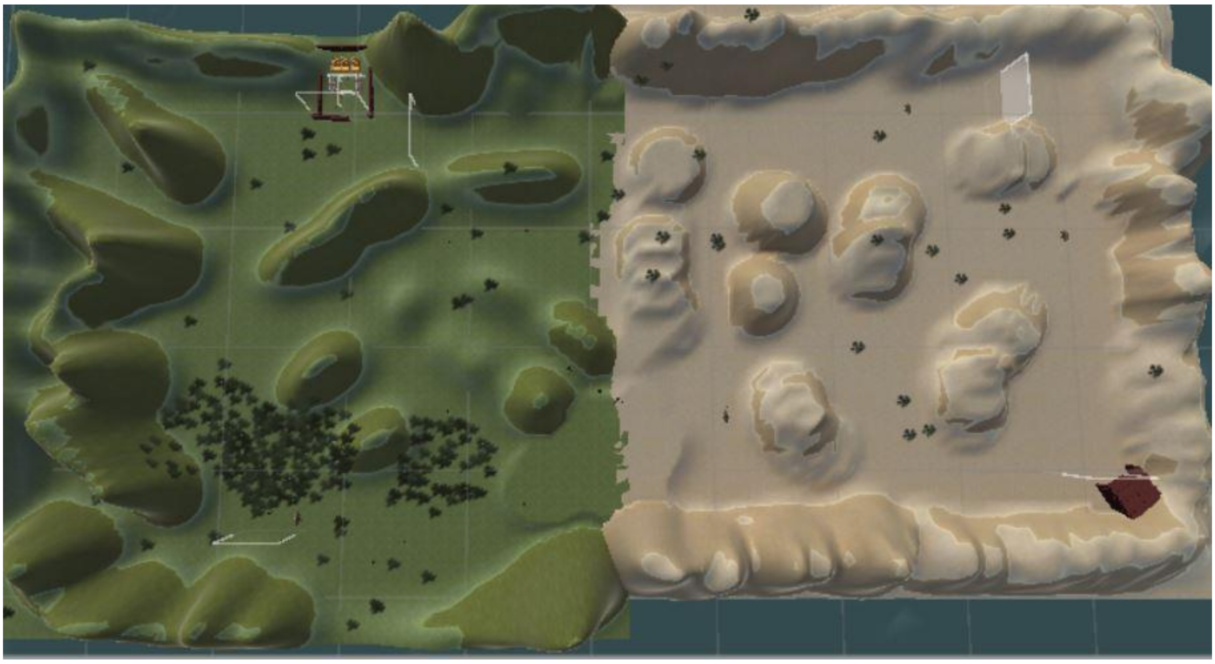


**Figure 70 (Level one)**

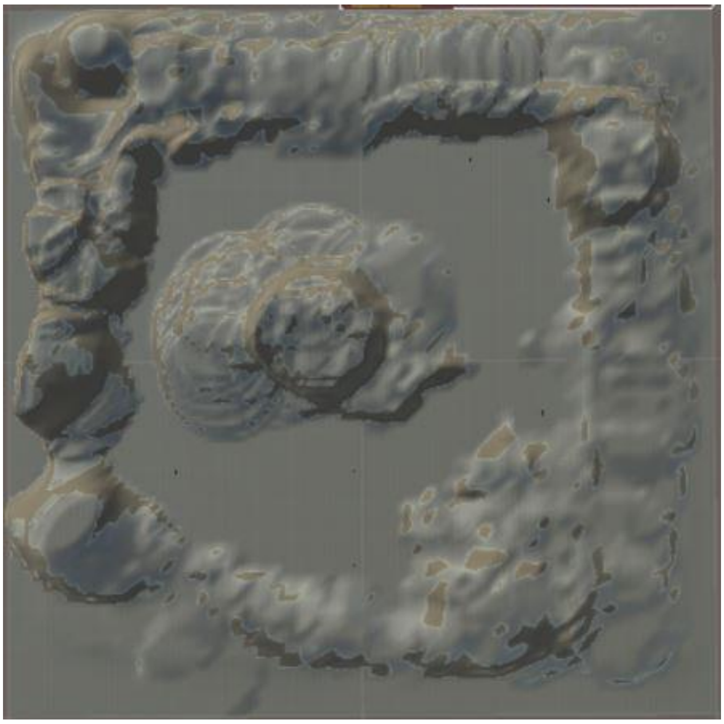


**Figure 71(Level two)**

### 3.10 Game Architecture

- Title Screen

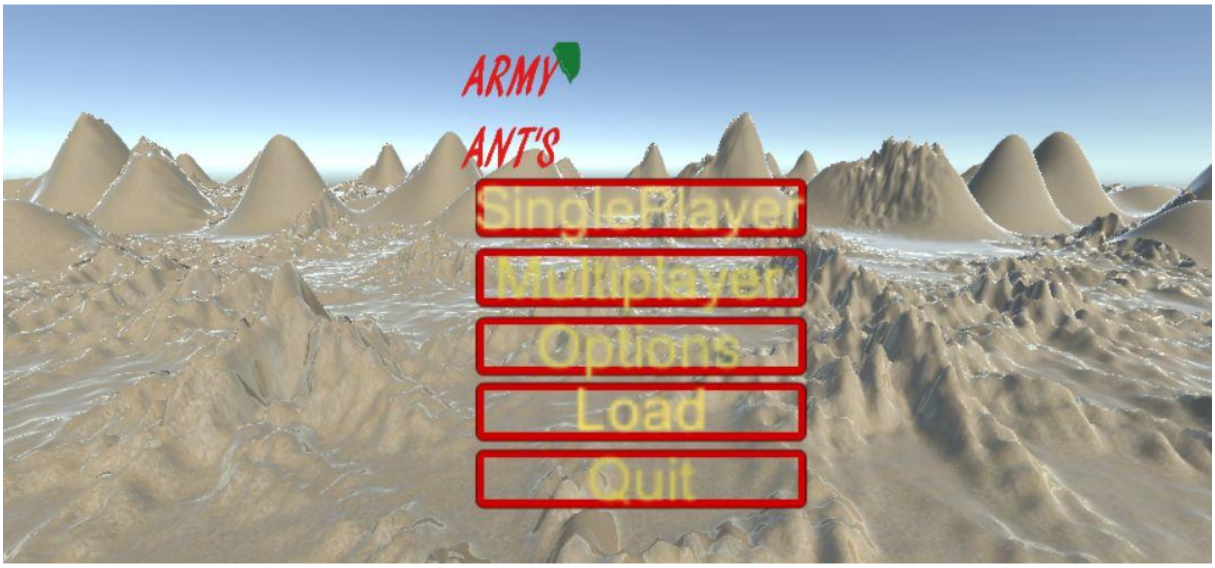


**Figure 72**

- Option Screens

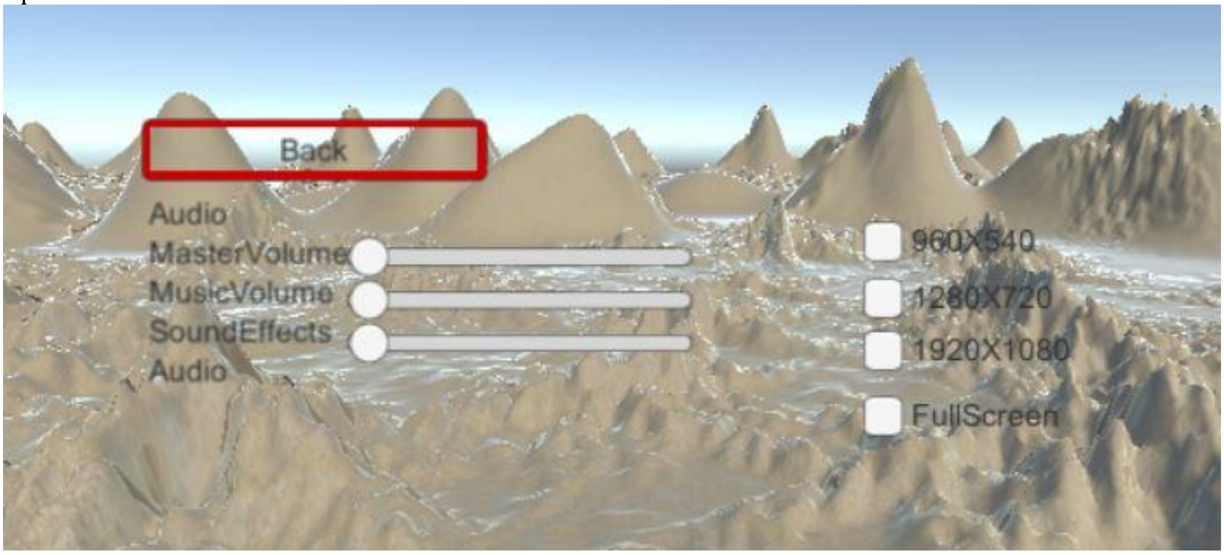


**Figure 73**

- End Screens

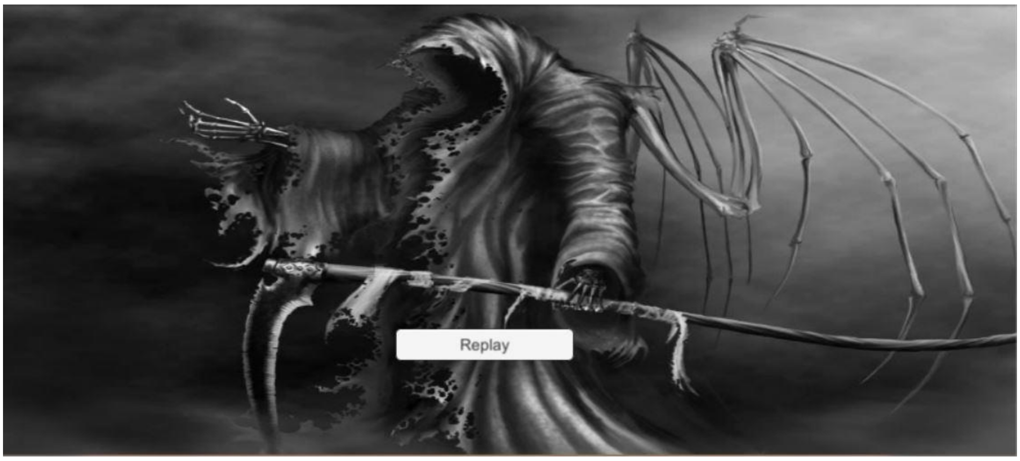


**Figure 74**

Death scene which gives you the option to replay that level

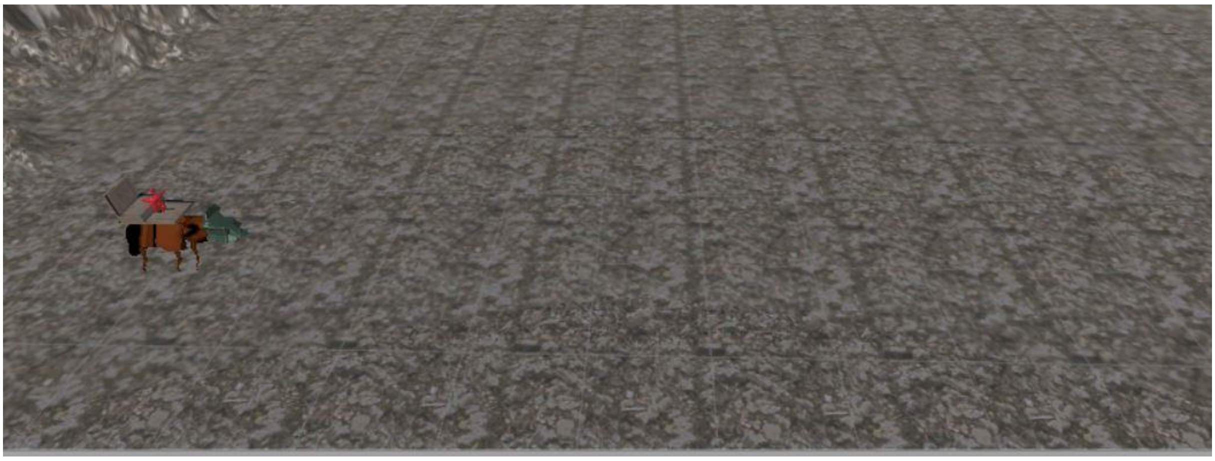


**Figure 75**

If victorious, the win scene launches and rolls through the titles as the queen rolls down in a BeetleTank.

## *3.11 Testing*

### *3.11.1.1 Overview*

Testing is an important part of any potential marketable project and no less so with video games, as they are one of the first industries to implement alpha and beta testing on actual future users of products. Testing has been done on the basis that when somebody creates something they tend to be blind to the flaws.

Tools used in the testing process of Army Ant's:

- **Visual Studio:** unit test C#.
- **Army Ant's Beta:** An open world created with most functions already implemented
- **Online Surveys:** Survey's which will be used are ones which will be created using survey monkey and this will provide a visual graphic for us to analyse.

Methods of testing used in Army Ant's:

- **Usability Testing**
- **Unit & Integration Testing**
- **Customer Testing**

Each type of test will be given a better definition under it's own heading and will explain what we wish to gain from this type of test.

### *3.11.1.2 Usability Testing*

The usability study used for army ant's is to find flaws within the control system of the game and also the design of the targeting system for the players.

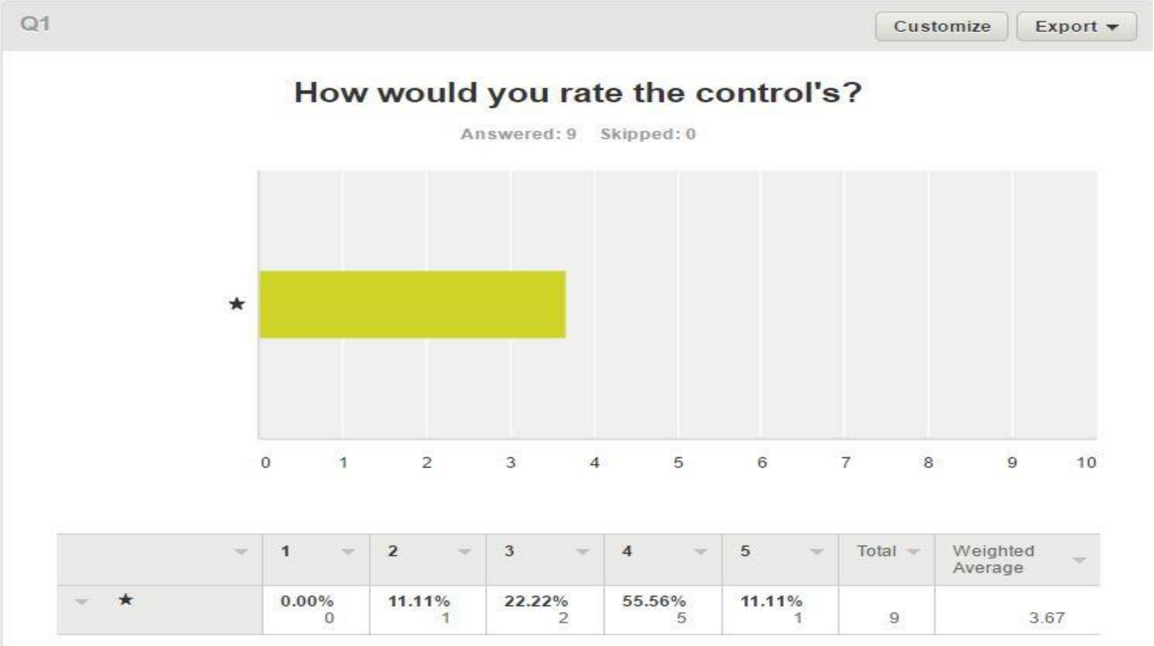
Number of participants: **9** *(between the ages of 10 - 30+)*

Tools used in Usability study:

- **Online Survey:** the survey was created to help find if the current control system was adequate and where it could be improved.
- **Army Ant's Beta:** The initial creation of the game

Usability Study:

*The rating system is based on a star 1 – 5, with 1 being a terrible grade and 5 being a flawless one.*



How would you rate the control's?

Answered: 9 Skipped: 0

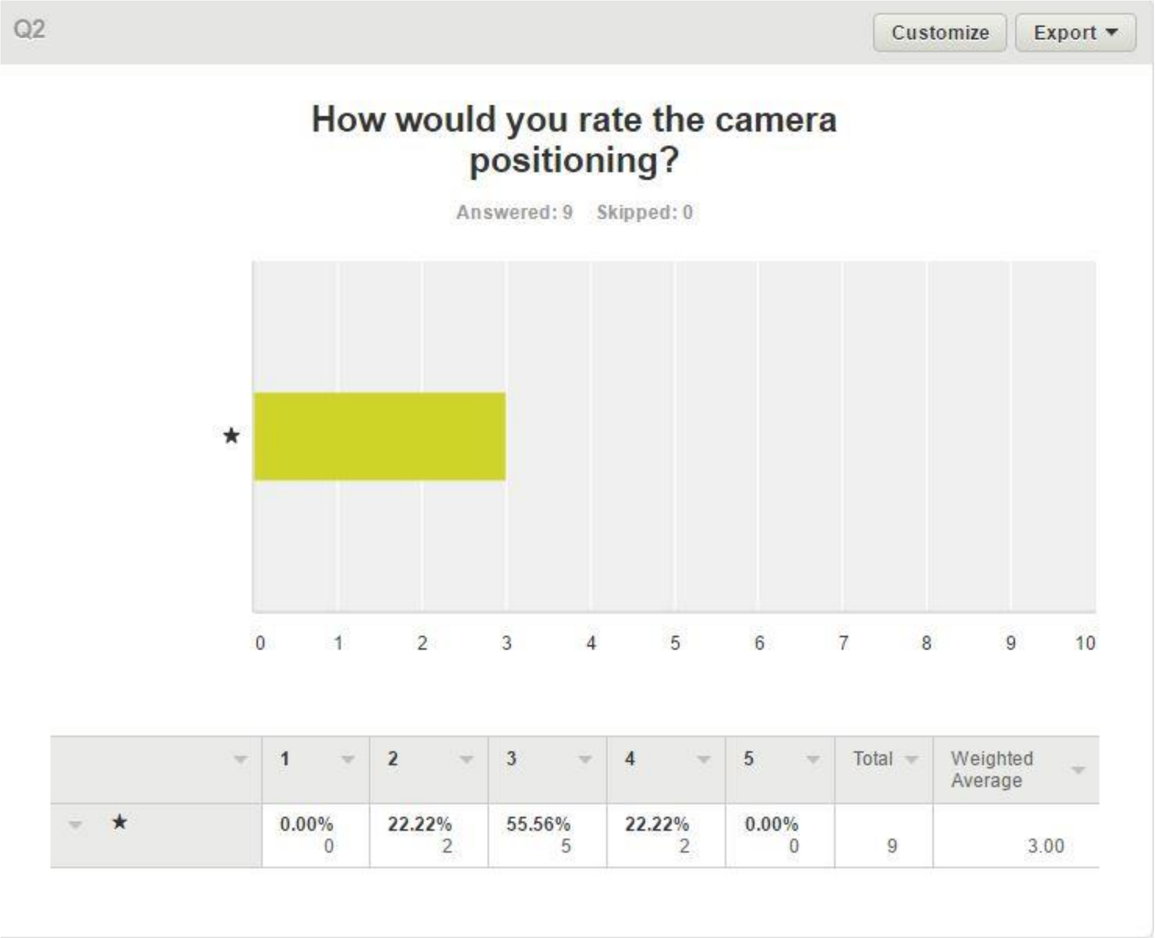
| | 1 | 2 | 3 | 4 | 5 | Total | Weighted Average |
|---|---|---|---|---|---|---|---|
| ★ | 0.00%<br>0 | 11.11%<br>1 | 22.22%<br>2 | 55.56%<br>5 | 11.11%<br>1 | 9 | 3.67 |

**Figure 76**

## Analysis & Solution

From the graph the ratings show that although the scoring isn't flawless, it is pretty solid, the lower rating may be down to the fact that we have not implemented the strafe left or right movement and this may make things difficult when approaching the enemy,

Solution at this moment is not fully needed but in the future will need to implement a much larger range of motion for the character.



How would you rate the camera positioning?

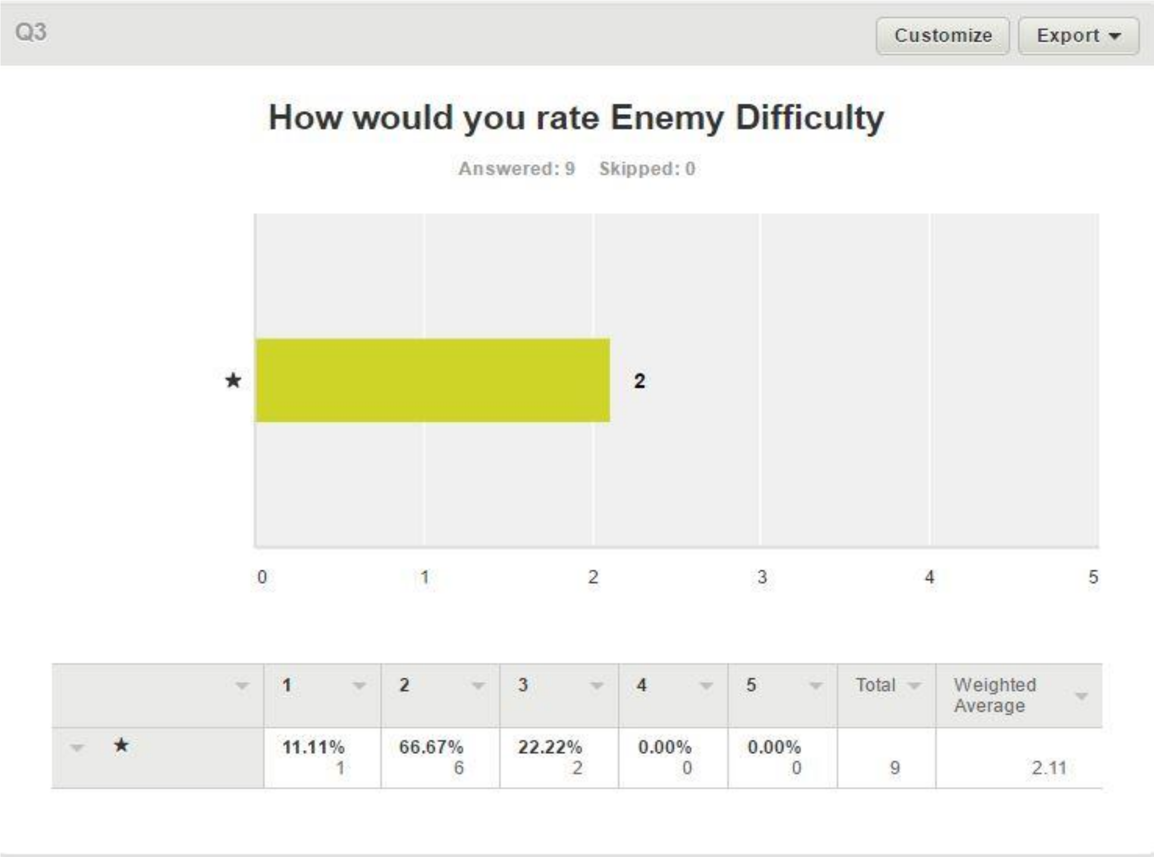
Answered: 9 Skipped: 0

| | 1 | 2 | 3 | 4 | 5 | Total | Weighted Average |
|---|---|---|---|---|---|---|---|
| ★ | 0.00%<br>0 | 22.22%<br>2 | 55.56%<br>5 | 22.22%<br>2 | 0.00%<br>0 | 9 | 3.00 |

**Figure 77**

With an average rating of 3 once again it's a fairly solid score and the vast majority of participants seem to be in this category.

Solution at the moment would maybe allow the camera rigged like more of a bungee cord system rather than a locked in linear system.



How would you rate Enemy Difficulty

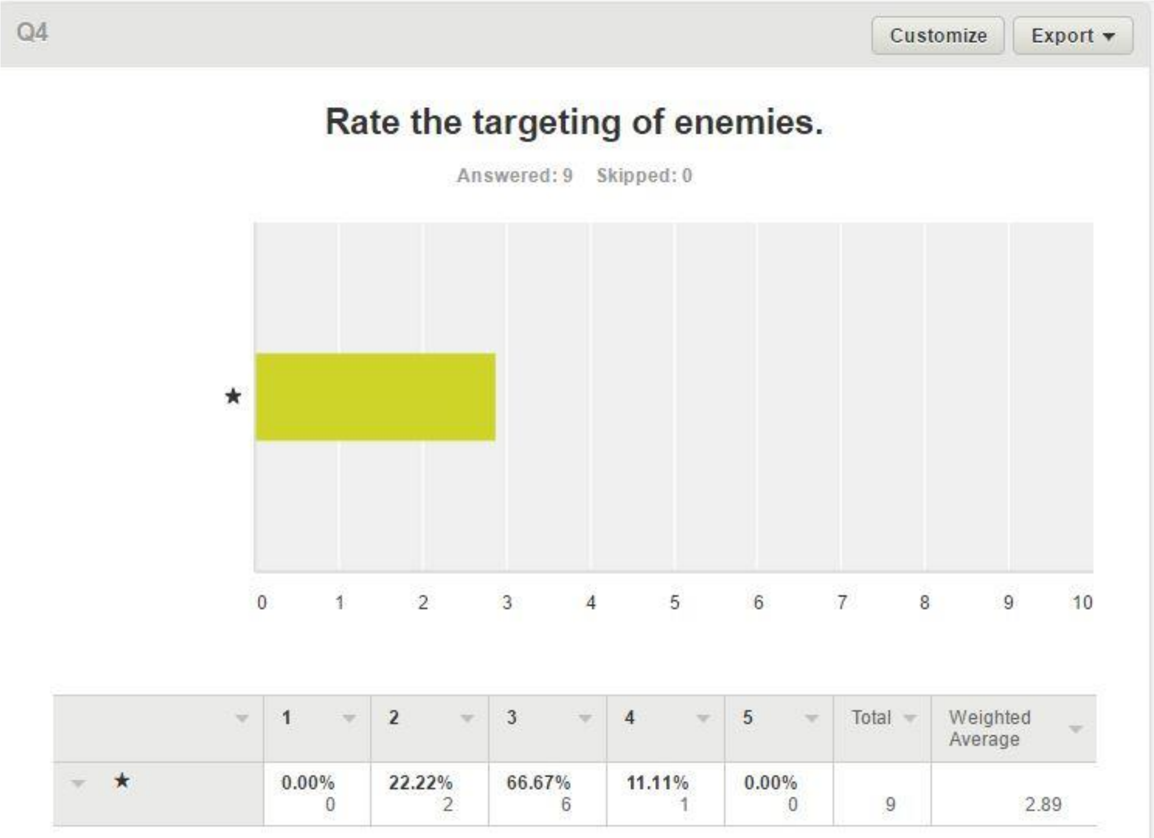
Answered: 9 Skipped: 0

| | 1 | 2 | 3 | 4 | 5 | Total | Weighted Average |
|---|---|---|---|---|---|---|---|
| ★ | 11.11%<br>1 | 66.67%<br>6 | 22.22%<br>2 | 0.00%<br>0 | 0.00%<br>0 | 9 | 2.11 |

**Figure 78**

With an average rating of 2.11 things would need to improved greatly as many people seemed to find it quite easy to defeat the enemy characters.

Solution would be to apply more enemy characters to the level and make them attack at a much greater rate as this means the player will have to react faster to the enemy which approach him.



Rate the targeting of enemies.

Answered: 9 Skipped: 0

| | 1 | 2 | 3 | 4 | 5 | Total | Weighted Average |
|---|---|---|---|---|---|---|---|
| ★ | 0.00%<br>0 | 22.22%<br>2 | 66.67%<br>6 | 11.11%<br>1 | 0.00%<br>0 | 9 | 2.89 |

**Figure 79**

With an average rating of 2.89, things could obviously be improved, the alignment of the raycast with the enemy character obviously needs improving.

Solution would be to either make a more identifiable response when you hit an enemy target or make the enemy collider larger but this would negate solutions created for the previous question.

## *3.12 Customer testing*

The customer user testing was carried out on a large group of people to find what they disliked about the game and also the genres which are most popular in the locality of Ireland.

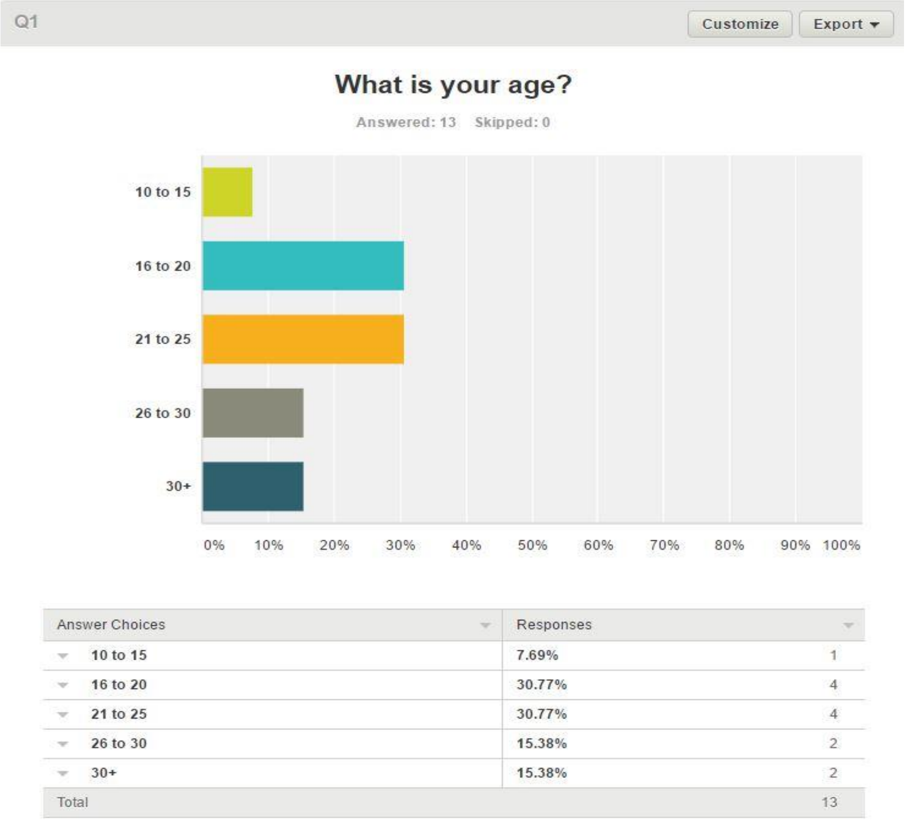
This testing will be helpful with future iterations of the game and will help us focus on issues which may exist.

Number of participants: **13** *(between the ages of 10 - 30+)*

Tools used in Usability study:

- **Online Survey:** the survey was created to help find if the current control system was adequate and where it could be improved.
- **Army Ant's Beta**

Customer User Study:



What is your age?

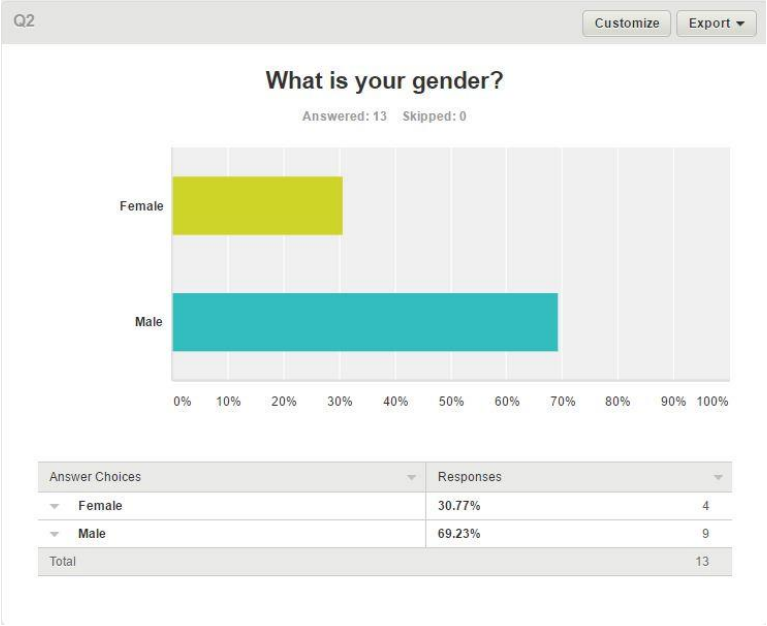
Answered: 13 Skipped: 0

| Answer Choices | Responses | |
|---|---|---|
| 10 to 15 | 7.69% | 1 |
| 16 to 20 | 30.77% | 4 |
| 21 to 25 | 30.77% | 4 |
| 26 to 30 | 15.38% | 2 |
| 30+ | 15.38% | 2 |
| Total | | 13 |

**Figure 80**

### Analysis & Thoughts

The first question was one which is more generalised to get a basis of the people who were actually testing this product and as you can see 60% of the test group lie in the age bracket of 16 – 25, this would be typical the age group which would have the most time and money to socially game.



What is your gender?

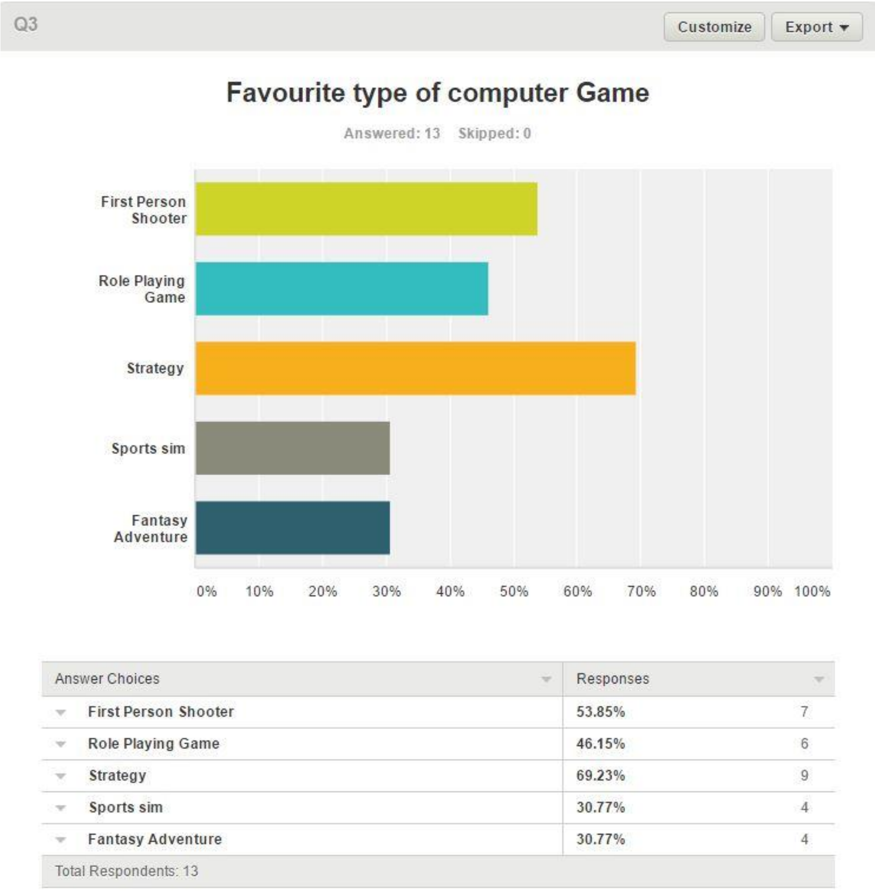
Answered: 13 Skipped: 0

| Answer Choices | Responses | |
|---|---|---|
| Female | 30.77% | 4 |
| Male | 69.23% | 9 |
| Total | | 13 |

**Figure 81**

### Analysis & Thoughts

The majority of base group with 69% are male which is typical higher than the world wide typical ratio as females are getting more involved in playing games especially at older ages. This will give me a clue as to the games to a typical market and who potentially prefers this kind of game.



Favourite type of computer Game

Answered: 13 Skipped: 0

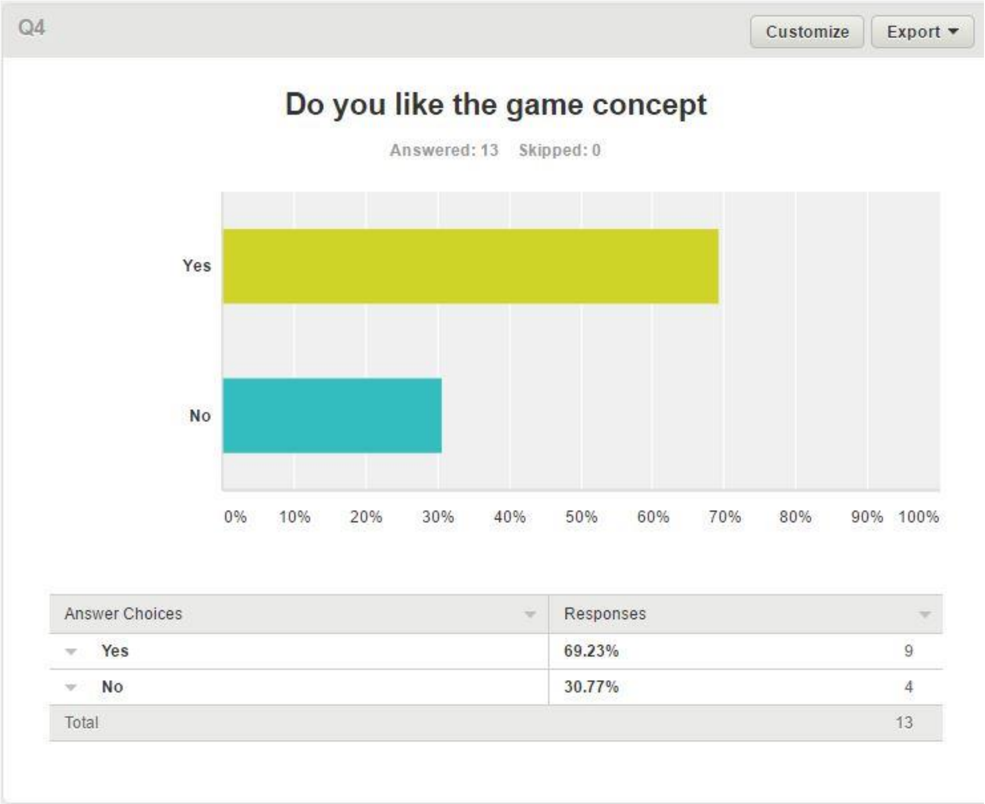
| Answer Choices | Responses | |
|---|---|---|
| First Person Shooter | 53.85% | 7 |
| Role Playing Game | 46.15% | 6 |
| Strategy | 69.23% | 9 |
| Sports sim | 30.77% | 4 |
| Fantasy Adventure | 30.77% | 4 |
| Total Respondents: 13 | | |

Figure 82

## Analysis & Thoughts

This question was designed to see what my group typical prefers as a gaming release to see whether this group would have a good basis and knowledge of the game genre, as you see can strategy and first person shooter come out top, now people were allowed make multiple choice's as many games would fall into more than one category.

My game would fall into fantasy, strategy and first person shooter even though it is a 3rd person shooter but many of the elements are the same.



Do you like the game concept

Answered: 13 Skipped: 0

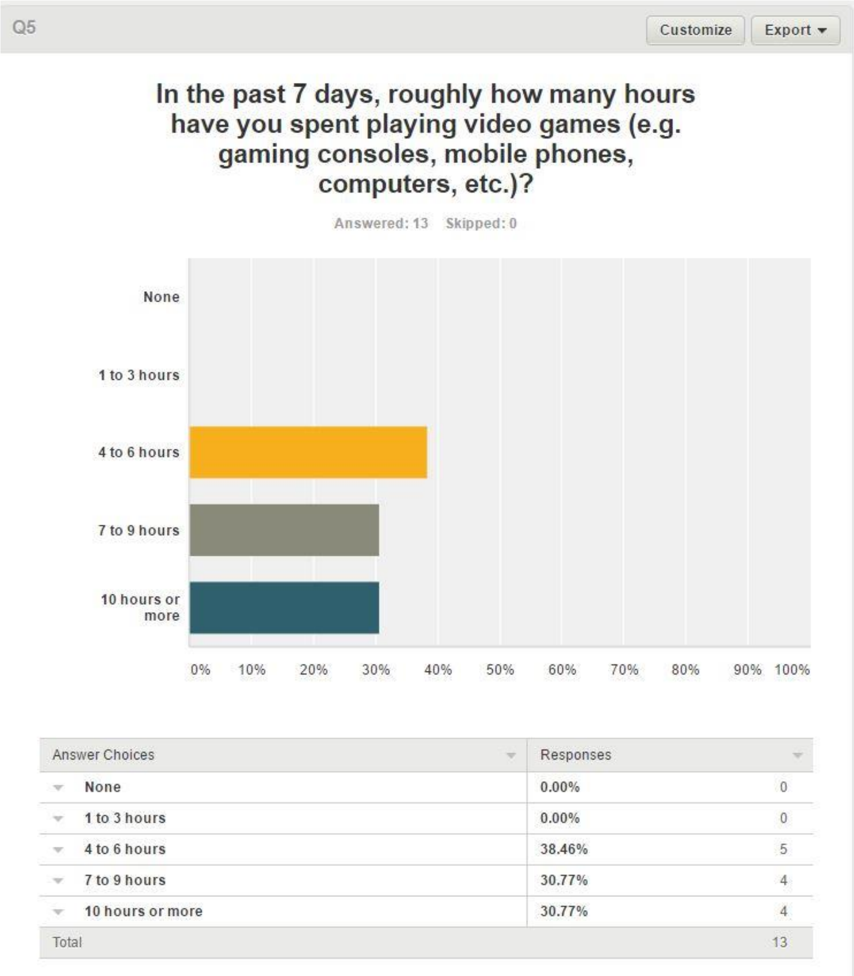
| Answer Choices | Responses | |
|---|---|---|
| Yes | 69.23% | 9 |
| No | 30.77% | 4 |
| Total | | 13 |

Figure 83

## Analysis & Thoughts

This question was simple enough to just get a basis of do people even like the concept of the game and with a 70% positive reaction it seems as if the concept is at least something which people are happy with.

The thoughts on this result is obviously that it is a very positive one, as I chose to use people who I don't know to get honest feedback, so at least the idea itself shows promise.



In the past 7 days, roughly how many hours have you spent playing video games (e.g. gaming consoles, mobile phones, computers, etc.)?

Answered: 13 Skipped: 0

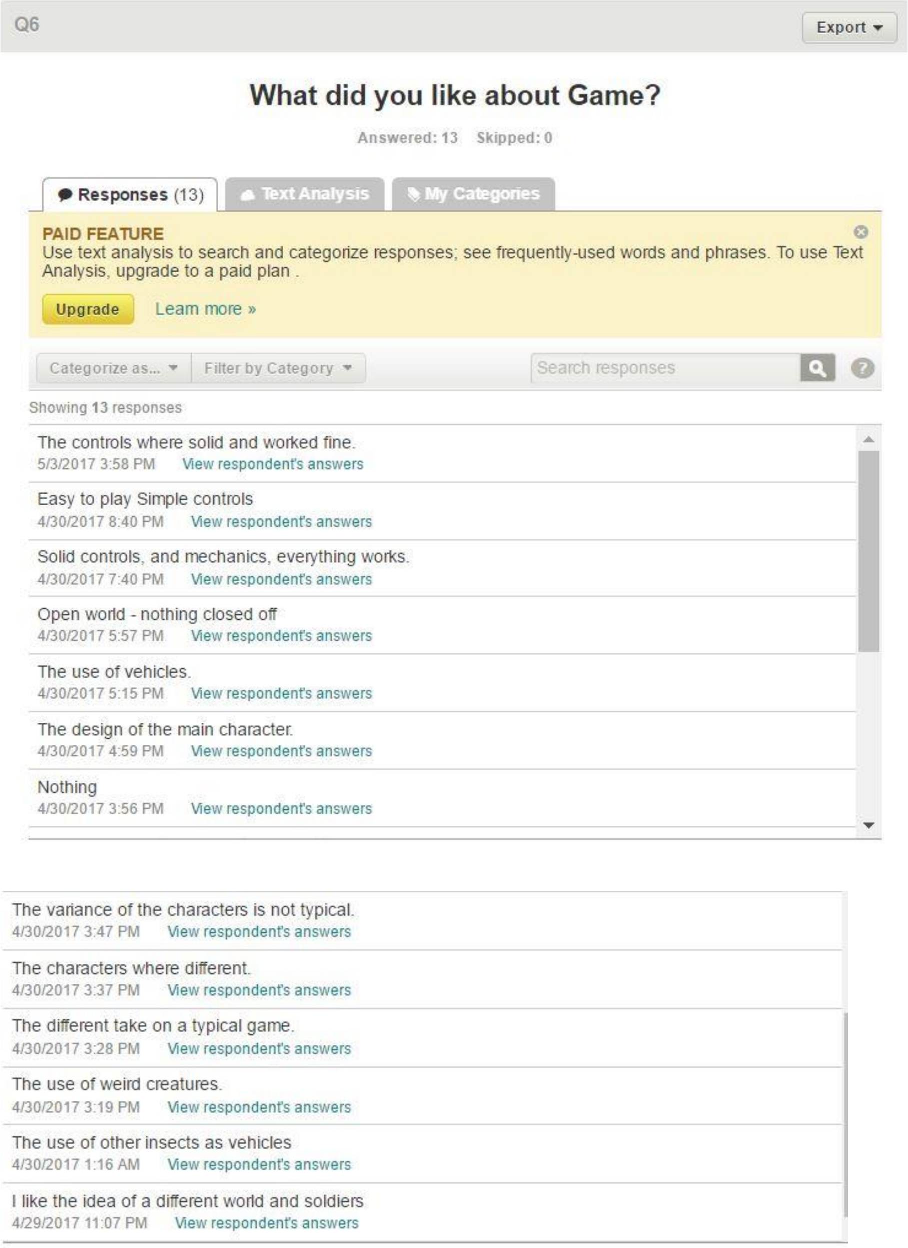
| Answer Choices | Responses | |
|---|---|---|
| None | 0.00% | 0 |
| 1 to 3 hours | 0.00% | 0 |
| 4 to 6 hours | 38.46% | 5 |
| 7 to 9 hours | 30.77% | 4 |
| 10 hours or more | 30.77% | 4 |
| Total | | 13 |

**Figure 84**

## Analysis & Thoughts

This question was designed so I could gauge my group against what would be classed as a typical gamer group, as you can see 61% of them play 7+ hours of games.

My thoughts on this are any game to be fully successful would have to be very engaging and provide a lot of hours of entertainment to be fully successful.



What did you like about Game?

Answered: 13 Skipped: 0

PAID FEATURE
Use text analysis to search and categorize responses; see frequently-used words and phrases. To use Text Analysis, upgrade to a paid plan .

Showing **13** responses

The controls where solid and worked fine.
5/3/2017 3:58 PM View respondent's answers

Easy to play Simple controls
4/30/2017 8:40 PM View respondent's answers

Solid controls, and mechanics, everything works.
4/30/2017 7:40 PM View respondent's answers

Open world - nothing closed off
4/30/2017 5:57 PM View respondent's answers

The use of vehicles.
4/30/2017 5:15 PM View respondent's answers

The design of the main character.
4/30/2017 4:59 PM View respondent's answers

Nothing
4/30/2017 3:56 PM View respondent's answers

The variance of the characters is not typical.
4/30/2017 3:47 PM View respondent's answers

The characters where different.
4/30/2017 3:37 PM View respondent's answers

The different take on a typical game.
4/30/2017 3:28 PM View respondent's answers

The use of weird creatures.
4/30/2017 3:19 PM View respondent's answers

The use of other insects as vehicles
4/30/2017 1:16 AM View respondent's answers

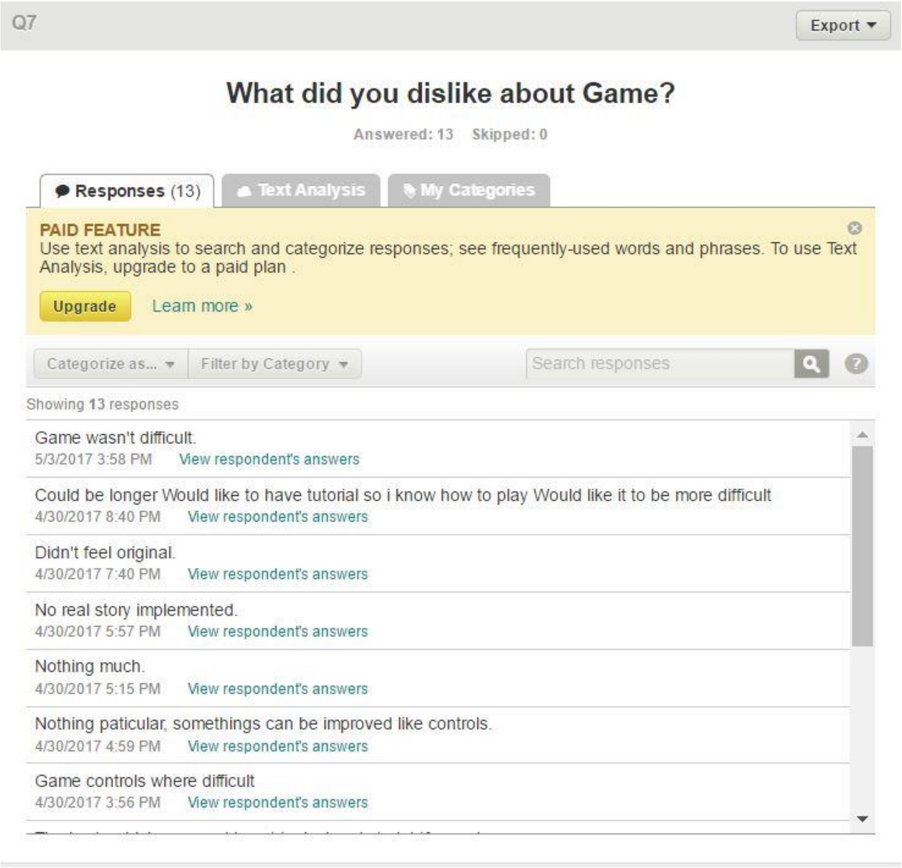
I like the idea of a different world and soldiers
4/29/2017 11:07 PM View respondent's answers

**Figure 85**

## Analysis & Thoughts

In this question it was decided to leave it open rather than leading people to an answer, I thought it was best to let them provide what they deemed appropriate.

Other than the one person who provided nothing, the controls got a positive feedback and so did the character's use of insects as vehicles.



What did you dislike about Game?

Answered: 13 Skipped: 0

PAID FEATURE
Use text analysis to search and categorize responses; see frequently-used words and phrases. To use Text Analysis, upgrade to a paid plan .

Showing 13 responses

Game wasn't difficult.
5/3/2017 3:58 PM View respondent's answers

Could be longer Would like to have tutorial so i know how to play Would like it to be more difficult
4/30/2017 8:40 PM View respondent's answers

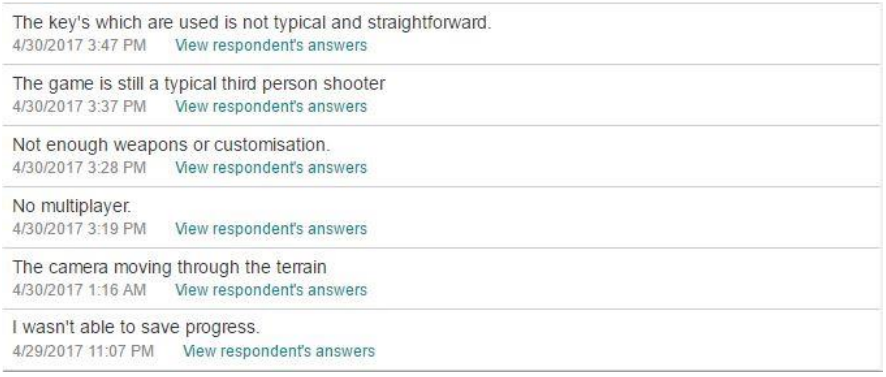
Didn't feel original.
4/30/2017 7:40 PM View respondent's answers

No real story implemented.
4/30/2017 5:57 PM View respondent's answers

Nothing much.
4/30/2017 5:15 PM View respondent's answers

Nothing paticular, somethings can be improved like controls.
4/30/2017 4:59 PM View respondent's answers

Game controls where difficult
4/30/2017 3:56 PM View respondent's answers



The key's which are used is not typical and straightforward.
4/30/2017 3:47 PM View respondent's answers

The game is still a typical third person shooter
4/30/2017 3:37 PM View respondent's answers

Not enough weapons or customisation.
4/30/2017 3:28 PM View respondent's answers

No multiplayer.
4/30/2017 3:19 PM View respondent's answers

The camera moving through the terrain
4/30/2017 1:16 AM View respondent's answers

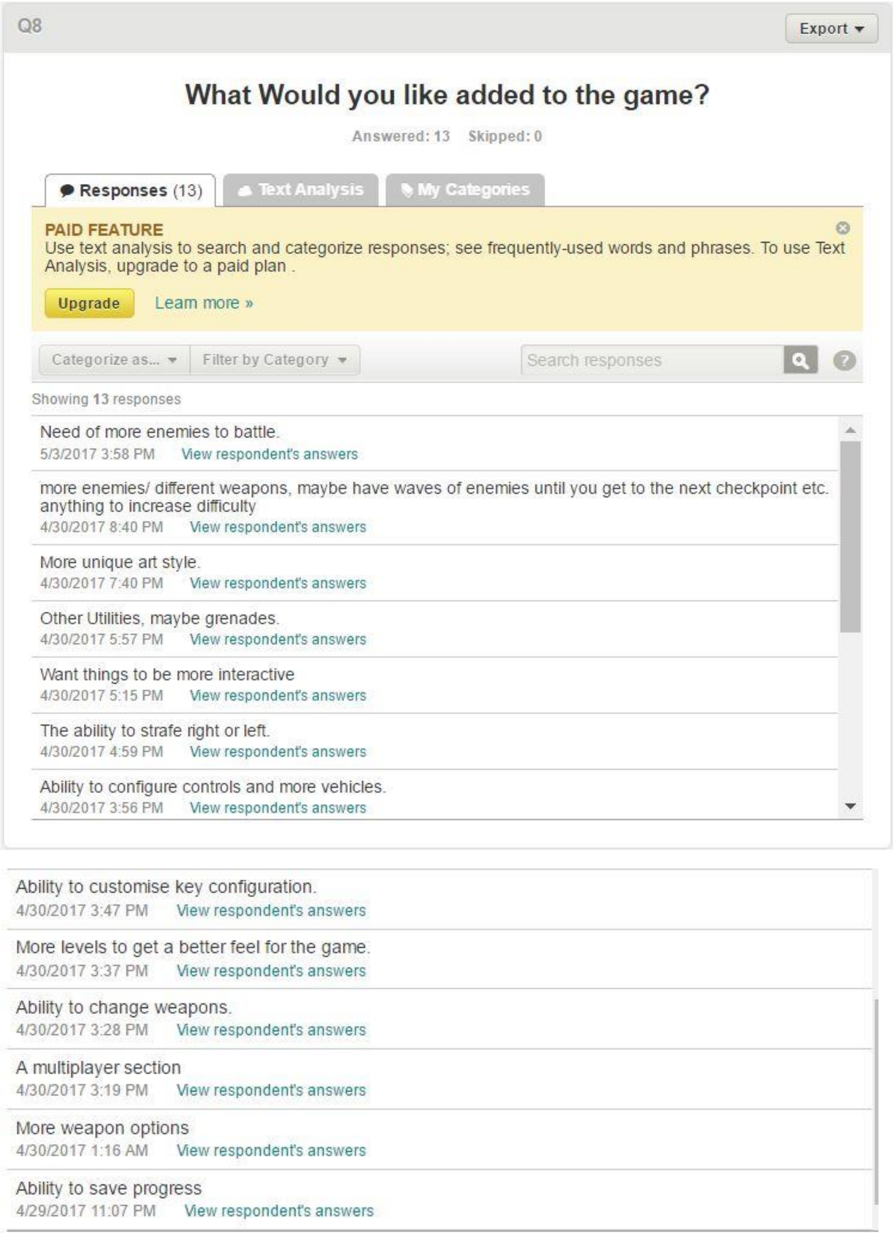
I wasn't able to save progress.
4/29/2017 11:07 PM View respondent's answers

**Figure 86**

## Analysis & Thoughts

In this question it was decided to leave it open rather than leading people to an answer, I thought it was best to let them provide what they deemed appropriate.

Other than the one person who said the game didn't feel original, it seemed like the game needed to be developed further and more objects added. They also want the game to be made more difficult.



What Would you like added to the game?

Answered: 13 Skipped: 0

PAID FEATURE
Use text analysis to search and categorize responses; see frequently-used words and phrases. To use Text Analysis, upgrade to a paid plan .

Showing 13 responses

Need of more enemies to battle.
5/3/2017 3:58 PM View respondent's answers

more enemies/ different weapons, maybe have waves of enemies until you get to the next checkpoint etc. anything to increase difficulty
4/30/2017 8:40 PM View respondent's answers

More unique art style.
4/30/2017 7:40 PM View respondent's answers

Other Utilities, maybe grenades.
4/30/2017 5:57 PM View respondent's answers

Want things to be more interactive
4/30/2017 5:15 PM View respondent's answers

The ability to strafe right or left.
4/30/2017 4:59 PM View respondent's answers

Ability to configure controls and more vehicles.
4/30/2017 3:56 PM View respondent's answers

Ability to customise key configuration.
4/30/2017 3:47 PM View respondent's answers

More levels to get a better feel for the game.
4/30/2017 3:37 PM View respondent's answers

Ability to change weapons.
4/30/2017 3:28 PM View respondent's answers

A multiplayer section
4/30/2017 3:19 PM View respondent's answers

More weapon options
4/30/2017 1:16 AM View respondent's answers

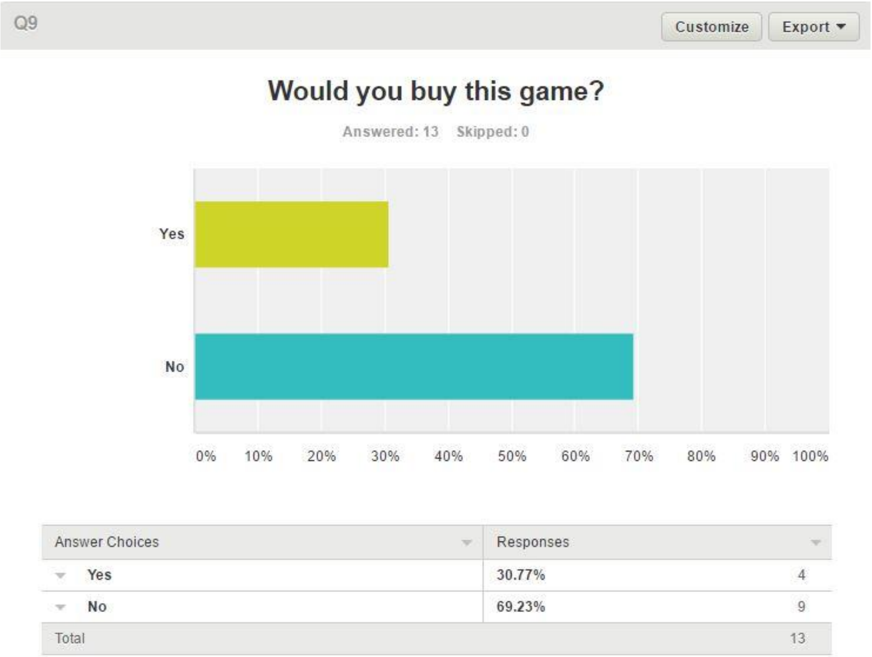
Ability to save progress
4/29/2017 11:07 PM View respondent's answers

**Figure 87**

## Analysis & Thoughts

This question needed to be left open so that anything which has not been thought of could be put forward for future iterations of the game and to improve the overall concept.

So things like more enemies to battle or provide the user with a varied selection of weapon's are all things in which would be implemented in the future.



Would you buy this game?

Answered: 13 Skipped: 0

| Answer Choices | Responses | |
|---|---|---|
| Yes | 30.77% | 4 |
| No | 69.23% | 9 |
| Total | | 13 |

**Figure 88**

### Analysis & Thoughts

This question is simple to see whether the game has any marketplace.

With a 31% to a 69% negative it doesn't seem as being complete lost cause and is potentially more positive than was initially anticipated.

## *3.13 Unit Testing, assertion + integration*

### 3.13.1 Tools used

- Unity Army Ant's Beta
- Unity Test Tools

First test (Health decrease on player)

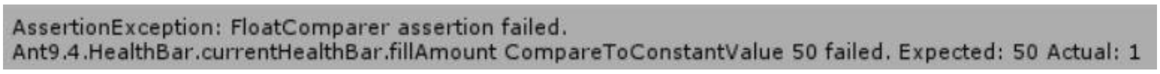
This test is necessary as with all games players need some way in which they lose and in this game it's when you lose your only life.

Steps

- Download Unity test tools
- Apply assertion script to player
- Set Assertion to Players healthbar script
- Trigger when player's health is equal to 50
- Set Game to pause on error

## Assertion testing

Assertion testing is something which is used in game testing to provide exact points when things happen in a game, i.e. player loses a certain amount of health etc.



```
AssertionException: FloatComparer assertion failed.
Ant9.4.HealthBar.currentHealthBar.fillAmount CompareToConstantValue 50 failed. Expected: 50 Actual: 1
```

**Figure 89**

As you can see this will notify you when your health reaches below the amount you have specified, this can also be set to pause the game when this error fires up.

This was also applied to triggers which open the next scene to understand how and when this item is triggered.

## Integration testing (Character Gravity)

Integration testing is done in gaming not in the pre-built levels which have already been populated, but in a more isolated environment to test specifics.

- Unity Army Ant’s Beta “Level”
- Unity Test Tools

First test is gravity test, as all 3d games need a gravitational function, it was decided to see how fast the object would fall and this means it could be customised better.

**Process:**

- Create plane and attach trigger to it and also the call testing script, this would say whether the test passed or failed
- Put main character into scene.

**Problems:** Main character drops at faster rate than required.

**Solution:** Decrease gravity character script.

## Integration testing (Destroy Enemy)

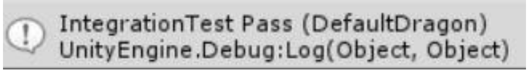
This test is just simply setting up the scene with an enemy character in it.

- Unity Army Ant’s Beta “Level”
- Unity Test Tools
- DragonBoar

In this integration, we are calling the enemy on destroy method, to make sure the object disappears.

**Problems:** No Problems (Enemy destroyed on method called)

**Solution:** No Solution Needed.



```
IntegrationTest Pass (DefaultDragon)
UnityEngine.Debug:Log(Object, Object)
```

**Figure 90**

## *3.14 Unit testing*

Unit testing is the testing of a specific action of code for an expected outcome. Unit testing is designed to find flaws before they become too big to handle.

- Unity Army Ant's Beta
- Unity Test Tools
- Microsoft visual studio

As many of the game elements required game objects to collide with them to be triggered unit test was carried on a more manual basis and iterative as new elements where implanted.

**Test1(Apple)**

Set - Up

- Player Character
- Damaged Health
- Apple

Expectation

- Increase health by 10

Outcome

- Success health increase from 54% to 60% with max health of 150/10 which is required increase of 6.66.

**Test2(NextScene)**

Set - Up

- Player Character
- Invisible Object
- Collider

Expectation

- Trigger Next Scene

Outcome

- Success on Ant colliding with object next scene launches.

**Test3(DamageEnemy)**

Set - Up

- Player Character
- Raycast
- Collider

Expectation

- Enemy life decrease

Outcome

- Success on Raycast hit, enemy loses health and eventually destroys(GameObject)

**Test4(DamageEnemyTank)**

Set - Up

- Player Character
- Raycast
- Collider
- EnemyTank
- Shell

Expectation

- EnemyTank life decrease on colliding with shell.

Outcome

- Success on colliding with EnemyTank it loses health and eventually destroys(GameObject).

## *3.15 Evaluation*

Just some of the things in which I wished to get from testing:

From an end user perspective, I will see if they find the game enjoyable, whether they find the setting enjoyable and the graphics appealing and engaging as environment should be immersive from a gamers perspective as when people enjoy and get lost in a game they tend to enjoy the experience more, the popularity tends to spread and you gain advertisement by word of mouth.

The world environment should be interactive and highly engaging and have a unique factor which will make them not want to put it away and always imagine what if I try this. Also, that everything runs smoothly and doesn't have many annoying bugs which take away from the gaming elements which people enjoy to make a more inclusive environment.

In a technical perspective, I would check for any network lag and pixel distortion and of course break up in what should be solid environment and people being able to pass through what they shouldn't be able to along with other glitches. I will also try to implement unit test on the code but this not usually a big practice in game development.

As their will, be multiplayer elements in the game its hoped that no connection is prioritised over others, so that some people have an advantage of others which would ultimately take away from others enjoyability. I would also compare my game to the industry standards out their and wouldn't expect it to have all the features it has, as they have many developers but would want it to hit key points.

## *3.16 Conclusions*

### *Milestones and Hurdles*

Milestones are what are the things I wish to achieve with the game.

These milestones didn't have deadlines as such but where things I wanted to implement as I progressed in the game development.

- Design the game with new unique characters and build a world which is fully open.
- Implement the ability for the hero to destroy enemy characters and also a health system for the hero so he can take damage.
- Create characters with varying abilities to attack the hero.

None of the these milestones had a particular timeline in mind, but they were corner stones in which the game relied on being implemented

Required: Milestones that where required where ones which had deadlines set by the college themselves, these needed deadlines as they would need to be graded throughout the year:

- Making sure I did the monthly reports done on time so as to allow myself to evaluate things.
- Create an initial prototype for the mid-point presentation to get across what I am trying to do.
- Complete the single player levels before final submission.

These milestones where important as to show that I was making some sort of progress.

### Hurdles

**Unity engine 5.3 – 5.5**

As I've never created a game before getting use to the layout of Unity was the first hurdle which had to be encountered. Unity is a game engine which is constantly changing and like many systems things become depreciated in future iterations. I also was required to get a better understanding of C# which is main programming language used in Unity.

Another issue I found with Unity is you could create a character in your modelling suite which is perfectly fine, but when you import it to Unity he can become deformed or lose the animation which was baked into the character.

**Autodesk Maya**

Maya is something which needed to be used to create the hero and his vehicles, this is something I have previously never used and to create objects of any great detail needs a lot of time. This was not only used to create characters but animate them and rig them up with skeleton for the animation but this did not work well at times so I chose to use a second suite.

**Blender**

Blender is a completely free modelling suite which is probably more beneficial for students who wish to create items which can have a marketable value. Blender was used for rigging the main character and animating, I think Unity and Blender have things in common as it is quite easy import characters into a Unity project from Blender, another benefit is character that may have been initially made in Maya and then put through Blender become the full property of the creator again.

I suppose one the hurdles was really understanding what could be realistically achieved in such a relatively short timeframe and with multiple different issues throughout the year.

**Unit Testing**

Unit testing is one of the biggest hurdles and one which I could not get over even though I tried to create a legitimate test many times it just would not work for me. This is partially down to the aspect that unit testing in gaming is a more difficult thing to do and is not really set up in the same way elements which may expect a string back.

## *3.17 Further development or research*

**Further development for:** Army Ant's

- **New levels**: Create new levels with new character controllers as you can see I already had started to develop a new user vehicle and I wished to create a level where flight would be a new element and you could have aerial combat.
- **Main Character Improvements:** Allow the main character to be more customisable by the player, also add additional weapons and player classes which may change the dynamics of how you engage with the enemy.
- **Improve Game Overall:** This will be done by further beta testing to make sure we are on the right track
- **Improvement in Graphics:** Remodel the main character so that he is made to higher standard than he is currently and allow customisation by providing different kinds of clothing.
- **Continue Creating:** Make a release grade projects with more things fully integrated and a storyline fully scripted over the game which will reveal why's and when's things happened.

**Research opportunities:**

- **Surveys:** Use the feedback from previous surveys and future surveys to implement and evolve the project more.
- **Continue Learning:** Create new better systems through the use of Unity and better understanding modelling of characters in Maya. Much cleaner animations would be of use, as if I add more weapons a changing weapon animation would need to be created.

**Closing Statement:** On reflection the creation of a game was something which I definitely enjoyed and wish I could have had more time in the end to create more to the game. This is definitely a project which allows you more creative freedom than a basic coding of a website which would have been a lot easier to create but I would have learned very little in the end.

I think game development doesn’t only give people skills in the creation of games, but many other careers need these skills, the likes of Maya is used by many animation studios and has been at the heart of many Oscar nominated movies.

# 4 References

Unity.2017. *Unity-GameEngine*.[ONLINE] Available at: https://www.Unity3d.com. [Accessed 08 May 2017].

Autodesk | 3D Design, Engineering & Entertainment Software. 2017. *Autodesk | 3D Design, Engineering & Entertainment Software*. [ONLINE] Available at: https://www.autodesk.com. [Accessed 08 May 2017].

Anon, (2017). [online] Available at: http://essentialfacts.theesa.com/Essential-Facts-2016.pdf [Accessed 25 Apr. 2017].

Lofgren, K. (2017). 2016 Video Game Statistics & Trends | Big Fish Blog. [online] Big Fish Games. Available at: http://www.bigfishgames.com/blog/2016-video-game-statistics-and-trends/ [Accessed 25 Apr. 2017].

Saylor, J. (2017). Top 3rd Person Shooter Games of All-Time. [online] Gameranx. Available at: http://gameranx.com/features/id/1932/article/3rd-person-shooter-games/ [Accessed 25 Apr. 2017].

"Simple Waypoint Pathfinding With The Line Of Sight AI". YouTube. N.p., 2017. Web. 3 May 2017.
https://www.youtube.com/watch?v=OmoKw1ikAi8&t=986s

Patreon. 2017. Sebastian Lague is creating Game Development Tutorials | Patreon. [ONLINE] Available at: https://www.patreon.com/SebastianLague. [Accessed 04 May 2017].

Lile, Darrin. "Blender101.Com". Blender101.com. N.p., 2017.Web.
[ONLINE] Available at: https://Blender101.com/ [Accessed 18 Dec. 2016.]
Patreon. 2017. Bracer Jack is creating Fun Stuff! | Patreon
[ONLINE] Available at: ttps://www.patreon.com/BracerJack [Accessed 02 Jan 2017]

Geig, Mike. "Unity - Unity Test Tools". *Unity*. N.p., 2017. Web. [ONLINE] Available at:
https://Unity3d.com/learn/tutorials/topics/production/Unity-test-tools [Accessed6 May 2017.]

### *4.1 Assets used*

DragonBoar -http://alexkim0415.wixsite.com/dungeonmason

Apples-
https://www.assetstore.Unity3d.com/en/#!/search/page=1/sortby=popularity/query=publisher:26225

AlienInsect - http://jayanam.com/

Signs – City Props

Blocks – Block Creator lite

# 5 Appendix

## *5.1 Project Proposal*

1. Objectives

The objective of my game is to provide a fun and interesting bit of entertainment, it's ultimately for people of all gaming talents and ultimately to relax, for me the objective is to learn all the elements required to create a viable game from design of the environment to coding elements which you interact with. I would like to create a fully interactive environment which people will enjoy.

My objective is to try and create a game with slightly new elements this is where it can be quite difficult as games have been around for so long now and many things have been done before. I would like to learn more elements in which I can add to my project. To bring new elements to shooting style game this will be done by it being ant's, as with humans you're quite limited in terrain but ant's will be able walk up vertical walls.

This will create more element's of attack scenario and will hopefully make the game more enjoyable and have a longer shelf life than the usual games of this type. So as you can see in the stats below the vast majority of games are a shooter of some kind which tend to be the most popular.

The vast majority of these games also tend to be multiplayer as most people play online and enjoy trying to beat opposing player's or co-operate with them to achieve a common goal.

2. Background

Gaming has had continuous growth since roughly the 80's when having a gaming console in your house became more prevalent and as like many kids i learned of this through the Sega mega-drive, this like many kids made me wonder how these games where made and what approaches people made to come up with these idea's.

So my idea evolved from an initial first person shooter of miniature people, but since being told this idea was not unique enough I decided to change things and hopefully for the better. I have decided to change it to third person and rather than having miniature people fighting a war, I have decided to go with an army ants style game.

These ants will have items like assault rifle and the likes. They will be fighting in a world that none of us ever see, as part of their military they will have mechanized dragonfly's as part of aerial support and some sort of beetle will take the place of a heavy infantry vehicle or a tank. The reason why I chose a multiplayer 3rd person genre game as it allows a large majority of people to enjoy the game.

With it being insects rather than human you don't really need an age rating as most parents wouldn't see it as gory or that their impressionable kids would follow this by watching it.

## 3. Technical Approach

My research will initially be on Unity's game engine and how to implement the game mechanics that is required for the outcome I require. Unity has quite a good commUnity forum with tutorials so I will be availing of them to learn how to use the game engine properly and how to implement certain elements and actions.

This will mean trying to implement real world physics in flight and weaponry and this will also include the destruction of objects and the ability to build your own defence by picking up items just like an ant can which will hopefully be my unique element. Further research will be done into graphical design and how to implement the movement of an ant.

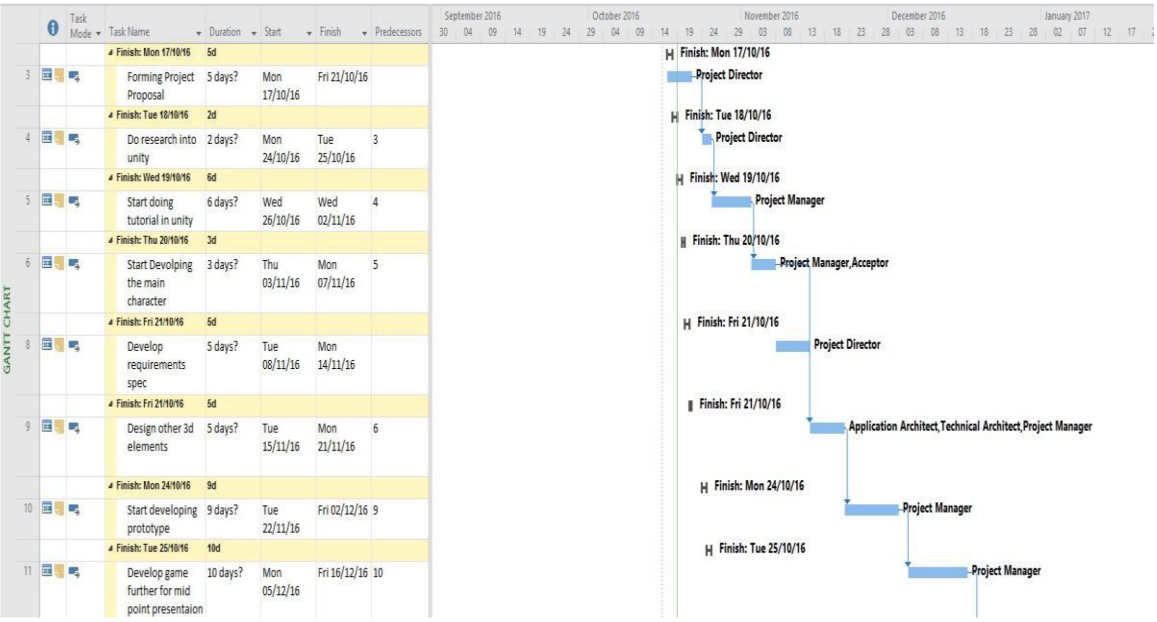
Further research will be done to see what elements I can add to make the project more enjoyable for the gamer, I will need to use a 3d software to create the environment and haven't decided yet between Maya and Blender as I really need to do more research into these things as Blender seems to have changed recently.

4. Special resources required

Unity Game Engine

Maya 3d designer

5. Project Plan



| | Task Mode | Task Name | Duration | Start | Finish | Predecessors |
|---|---|---|---|---|---|---|
| | | Finish: Mon 17/10/16 | 5d | | | |
| 3 | | Forming Project Proposal | 5 days? | Mon 17/10/16 | Fri 21/10/16 | |
| | | Finish: Tue 18/10/16 | 2d | | | |
| 4 | | Do research into unity | 2 days? | Mon 24/10/16 | Tue 25/10/16 | 3 |
| | | Finish: Wed 19/10/16 | 6d | | | |
| 5 | | Start doing tutorial in unity | 6 days? | Wed 26/10/16 | Wed 02/11/16 | 4 |
| | | Finish: Thu 20/10/16 | 3d | | | |
| 6 | | Start Devolping the main character | 3 days? | Thu 03/11/16 | Mon 07/11/16 | 5 |
| | | Finish: Fri 21/10/16 | 5d | | | |
| 8 | | Develop requirements spec | 5 days? | Tue 08/11/16 | Mon 14/11/16 | |
| | | Finish: Fri 21/10/16 | 5d | | | |
| 9 | | Design other 3d elements | 5 days? | Tue 15/11/16 | Mon 21/11/16 | 6 |
| | | Finish: Mon 24/10/16 | 9d | | | |
| 10 | | Start developing prototype | 9 days? | Tue 22/11/16 | Fri 02/12/16 | 9 |
| | | Finish: Tue 25/10/16 | 10d | | | |
| 11 | | Develop game further for mid point presentaion | 10 days? | Mon 05/12/16 | Fri 16/12/16 | 10 |

6. Technical Details

The principle languages being used are more than likely going to be C# and javascript along with the libraries they are associated with. I will use Unity game engine to develop the game and probably use Maya or 3dsmax to develop the characters and all the 3d elements.

7. Evaluation

From an end user perspective I will see if they find the game enjoyable, whether they find the setting enjoyable and the graphics appealing and engaging as environment should be immersive from a gamers perspective as when people enjoy and get lost in a game they tend to enjoy the experience more, the popularity tends to spread and you gain advertisement by word of mouth.

The world environment should be interactive and highly engaging and have a unique factor which will make them not want to put it away and always imagine what if I try this. Also that everything runs smoothly and doesn't have many annoying bugs which take away from the gaming elements which people enjoy to make a more inclusive environment.

In a technical perspective, I would check for any network lag and pixel distortion and of course break up in what should be solid environment and people being able to pass through what they shouldn't be able to along with other glitches. I will also try to implement unit test on the code but this not usually a big practice in game development.

As their will be multiplayer elements in my game I hope to make sure that no connection is prioritised over others, so that some people have an advantage of others which would ultimately take away from others enjoyability. I would also compare my game to the industry standards out their and wouldn't expect it to have all the features it has, as they have many developers but would want it to hit key points.

## *5.2 Project Plan*

## *5.3 Monthly Journals*

Reflective Journal:September

Student name: James McConville(x13114441)

Programme (BSc in Computing/Gaming):

Month: September

My Achievements

So far this month has just been doing a bit of research on technology to use and like Unity and getting feedback on my initial idea for my game which I was told to add a unique element to maximise marks which I think I have. Learning the code which is required to make the game and what I can use to maximise the fun element of the game. Doing more research on what is required on the tech that I'm required to use to implement my idea.

Trying to figure out what 3d modelling software I can use to make my designs and bring them to life.

My Reflection

As this month was getting the initial feedback on my idea, I haven't really gone massively in depth in what I would like to do as I've been waiting on the green light, with 2 weeks now to go for the next part of the submission it's something I need to get cracking on. I also need to request a supervisor who needs to tell what elements I'm missing to maximise my marks. So I think I really need to do a bit more research and finalise my unique element which I think I have which is very difficult for gaming.

Intended Changes

Right now I don't have any changes set out as I really need to do a lot more research for what is required to do.

Supervisor Meetings

Date of Meeting: No supervisor yet

Items discussed:

Action Items:

Reflective Journal:October

Student name: James McConville(x13114441)

Programme (BSc in Computing/Gaming):

Month: October

My Achievements

So far what I've achieved is doing reports and doing more research into how things should be implemented, this includes first meeting with my academic supervisor which was a group meeting just so he could get a broad perspective of our ideas. A bit of research on Unity and 3d modelling.

My Reflection

On reflection I don't think I did as much as I might have wanted to do especially with how much work we need to do this year separate from the software project. What I really need to do is implement one of my characters and try and create some animation. Also I must complete my requirements spec as it's the next important upload.

Intended Changes

Right now I don't have any changes set out as I really need to do a lot more research for what is required to do.

Supervisor Meetings

Date of Meeting: (Paul Stynes)24/10/2016

Items discussed: My idea, Start requirements spec and do story board

Action Items:

Student name: James McConville(x13114441)

Programme (BSc in Computing/Gaming):

Month: November

My Achievements

The achievements for this month have been to finish off requirements document and continue modelling my main character.

My Reflection

On reflection I don't think I did as much as I might have wanted to do especially with how much work we need to do this year separate from the software project. Continued heavy workload means I'm not where I would like to be at this point. This means during Christmas most will have to be done so next semesters workload doesn't get in the way.

Intended Changes

Right now I don't have any changes set out, this may change after my mid-point presentations which is taking place soon

Supervisor Meetings

Date of Meeting:

Items discussed:

Action Items:

My Reflection

On reflection my presentation did obviously not go well with such a low score, but with no help from my supervisor and 0 feedback I don't know where to go from here, feeling very underwhelmed and feel like there is little hope for good mark. I've only missed 1 class of software project and 1 meeting with the supervisor and was basically told by said supervisor that the mid-point presentation doesn't really matter, it's just get a feel of where you are, so with such a low score I feel he isn't taking me seriously at all and I see little point in having one if their no help.

I have tried to complete my main character while studying for these exams, also the mid-point presentation falling in between when we have things due did not help at all, both classes had AI due but then the opposite class got an extra 2 weeks which is after mid-point presentations which is a massive unfair advantage, I don't see how we can all be judged on the same criteria if we are all not always set the same challenges.

Reflective Journal

Student name: James McConville(x13114441)

Programme (BSc in Computing/Gaming):

Month: December

My Achievements

The achievements for this month have been to finish off requirements document, also with this my main character is nearly completely remodelled and looks completely different to the previous one. I gave my presentation also.

My Reflection

On reflection my presentation did obviously not go well with such a low score, but with no help from my supervisor and 0 feedback I don't know where to go from here, feeling very underwhelmed and feel like there is little hope for good mark. I've only missed 1 class of software project and 1 meeting with the supervisor and was basically told by said supervisor that the mid-point presentation doesn't really matter, it's just get a feel of where you are, so with such a low score I feel he isn't taking me seriously at all and I see little point in having one if their no help.

I have tried to complete my main character while studying for these exams, also the mid-point presentation falling in between when we have things due did not help at all, both classes had AI due but then the opposite class got an extra 2 weeks which is after mid-point presentations which is a massive unfair advantage, I don't see how we can all be judged on the same criteria if we are all not always set the same challenges.

Intended Changes

No intended changes as I don't know where I'm going wrong.

Supervisor Meetings

Date of Meeting:

Items discussed:

Action Items:

Reflective Journal

Student name: James McConville(x13114441)

Programme (BSc in Computing/Gaming):

Month: January

My Achievements

Continued remodelling characters and add some animation. Also started coding things like the games option menu, things like the sound effects and resolution.

My Reflection

On reflection I have added things to the best of my ability and tried to find out what game's need specifically and have requested a meeting with my supervisor twice to get a meeting and have provided him with my phone number but he is yet to provide me with a date other then a group meeting which is little use to me now 4 weeks later. I have not got what I wanted done yet but I believe I can make rapid progress very soon.

Intended Changes

No intended changes as I don't know where I'm going wrong.

Supervisor Meetings

Date of Meeting:

Items discussed:

Action Items:

Reflective Journal

Student name: James McConville(x13114441)

Programme (BSc in Computing/Gaming):

Month: February

My Achievements

Modelled all characters and implemented the animation is both the main character and the BeetleTank. I have added some of the character controllers. I have identified some of the assets to use for my game and wrote my 40 word for the booklet.

My Reflection

On reflection I have gotten a little further but probably not as far as I wish to be, I think there is a lot more to implement and I wish I had more time. I think with certain issues which caused a bit of a limbo situation it may have interfered with time scheduling issues and I wish I had used the time better.

Intended Changes

No changes to be made at the minute, I believe the current game can be made.

Supervisor Meetings

Date of Meeting: 20 + 27 February

Items discussed: Items discussed where what to implement next

Action Items: To have BeetleTank implemented by the next meeting

Reflective Journal

Student name: James McConville(x13114441)

Programme (BSc in Computing/Gaming):

Month: March

My Achievements

I have started to try to integrate a save and load function, this does not seem easy. I have integrated a second enemy attack strategy.

My Reflection

On reflection I have moved forward with game but still feel I have a lot more to do, I think because of certain other subjects being due in the last couple of weeks meant I may have let the games progress fall by the wayside. I hope to add a multiplayer section to the game.

Intended Changes

No changes to be made at the minute, I believe the current game can be made.

Supervisor Meetings

Date of Meeting: 3 march 2017

Items discussed: Items discussed where what to implement a new level and a story

Action Items: To have BeetleTank implemented by the next meeting https://www.awseducate.com/student/s/login/?language=en_US

x13114441@student.ncirl.ie

Password: G1asn3v1n

Project contained in s3 bucket