

National College of Ireland  
BSc in Computing  
2016/2017

Adam Butterly  
X13308011  
X13308011@student.ncirl.ie

## Software Project (Gaming and Multimedia Stream)

“Gemini”

Technical Report



## Table of Contents

### Contents

Introduction to Gemini .....	5
Executive Summary.....	5
Research.....	5
Brief Description .....	5
Game Analysis .....	5
Software .....	7
Game Atmosphere .....	8
Level Description.....	11
Game Play .....	14
Key Features.....	14
Selling Features .....	15
Project Scope.....	15
Definitions, Acronyms, and Abbreviations.....	15
User Requirements Definition.....	15
Project Design Report .....	17
Design Version.....	17
Design Guidelines .....	17
Game Design Definitions.....	17
Player Definition .....	17
Player Properties.....	18
Player Rewards (Power-ups & Pick-ups).....	19
Heads up Display .....	22
Implementation.....	23
Character Implementation .....	23
Character Animations.....	23
Player Properties.....	25
Weapons Implementation.....	29
AI Implementation .....	31
Game System Implementation .....	34
Project Technical Report .....	36
Requirements Specification.....	36

Functional requirements .....	36
Requirement: Start New Game.....	36
Requirement: Save Game .....	38
Requirement: Load Game .....	39
Requirement: Exit Game .....	40
Requirement: Change Game Settings .....	42
Non-Functional Requirements .....	43
Performance/Response time requirement.....	43
Availability requirement.....	44
Reliability requirement .....	44
Maintainability requirement.....	44
Extendibility requirement .....	44
Reusability requirement.....	44
GUI Implementation .....	45
Main Menu.....	45
Settings Menu .....	46
Pause Menu .....	47
Upgrade Menu.....	48
Game Over Menu .....	49
End Game Menu .....	50
System Evolution.....	50
Testing Overview .....	51
Appendix .....	52
Project Proposal.....	52
Project Plan.....	53
Monthly Journals .....	54
Month: September .....	54
Month: October.....	55
Month: November .....	55
Month: December.....	56
Month: January.....	56
Month: February .....	57
Month: March .....	57

**Declaration Cover Sheet for Project Submission**

**SECTION 1 Student to complete**

<b>Name: Adam Butterly</b>
<b>Student ID: x13308011</b>
<b>Supervisor: Frances Sheridan</b>

**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: Adam Butterly Date: 10/05/2017

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.

**Complete the sections above and attach it to the front of one of the copies of your assignment,**

## **Introduction to Gemini**

### **Executive Summary**

Gemini has been created using Unreal engine alongside Autodesk Maya, Blender and Adobe Photoshop.

Gemini is a first-person, action-orientated shooter. It's in depth sci-fi setting and story will be its attraction to players. The level design and graphics provided by Unreal engine will be what sets this game apart. Its combination of software has created visually striking level design and innovation 3D characters.

The player will take control of Lynn, a doctor on board a space station orbiting a foreign planet that is over run by an unheard-of alien race. The game will be set in the future where the limits of the worlds technology far exceeds what it would be now. After losing the rest of her crew and escaping the station on a ship, she is left stranded on the surface of an alien planet, this is the setting for the tutorial level. With no experience in fighting, she must return to the space station to uncover the truth behind what has happened and discover the fate of her crew. Consumed by the guilt of leaving them behind she desperately struggles to find and help them. Can she beat back the aliens that have taken everything from her.

### **Research**

After doing some research into other games of this category it is safe to say the target audience for the game would be between the age of 18 and 35, this works well as it incorporates the average age of a gamer today which is 34. The game can also be catered towards an audience younger than 18 which would open the target audience up to the majority of people as in the future the game could be moved from PC/Steam onto any console. The sci-fi genre is growing ever more popular in gaming as it offers infinite possibilities in regard to storytelling and Gemini is a perfect example of this.

Another aspect that is planned to set the game apart from games within the same genre is, Lynn. The role of a female protagonist is something the gaming world is still trying to become accustomed to and this almost controversial design choice is something that will grab potential consumers' attention and really set it apart. Although research into the subject does indicate that games with female lead characters don't sell as well as games with male lead characters

Last year in 2016 one of the biggest games of the year was "Doom Remastered", a sci-fi shooter not far off what was planned for Gemini. The genre continues to thrive as more franchises are investing in it, including Call of Duty and will new releases set for 2017, including triple A titles such as "Prey".

### **Brief Description**

The game will have a functioning tutorial level which will show the player everything they need to know about its mechanics and will showcase the individuality the game has to offer. The tutorial level will come with widgets and pop ups at certain intervals in the level to guide the player through what they must do and how to use some of the features in the game. The tutorial level, along with the rest of the game, will have a save and load feature so player can save their progress and pick up from where they left off. This function is vital to any game users would lose interest quickly if they were forced to repeatedly replay actions that they had previously lost. The game will have a Sci theme as this genre spans across many diverse types of games which have proven to be a massively popular area in the games market.

### **Game Analysis**

This is a general overview of all the games aspects.

<b>Game Description</b>	
Genre:	<ul style="list-style-type: none"> <li>• First Person</li> <li>• Adventure</li> <li>• Action</li> </ul>
Game Elements:	<ul style="list-style-type: none"> <li>• Shooting</li> <li>• Collecting</li> <li>• Combat</li> <li>• Dodging</li> </ul>

Game Content:	<ul style="list-style-type: none"> <li>• Drama</li> </ul>
Theme:	<ul style="list-style-type: none"> <li>• Sci-Fi</li> </ul>
Style:	<ul style="list-style-type: none"> <li>• Realistic</li> </ul>
Game Sequence:	<ul style="list-style-type: none"> <li>• Linear- Storylines</li> </ul>
Player:	<ul style="list-style-type: none"> <li>• Campaign – One Player</li> </ul>
<b>Game Reference</b>	
Game Taxonomy:	<ul style="list-style-type: none"> <li>• <i>Gemini is a Fictional Game/Narrative.</i></li> </ul>
Player Immersion:	<ul style="list-style-type: none"> <li>• Narrative</li> <li>• Destructive</li> <li>• Collection</li> <li>• Emotional</li> </ul>
Reference:	<ul style="list-style-type: none"> <li>• Gemini Weapons and weapon advancing system will be a lot like that of the Doom 2016 Remaster.</li> </ul>
<b>Game Technical</b>	
Technical From:	<ul style="list-style-type: none"> <li>• The game will be made with 3D graphics.</li> </ul>
View:	<ul style="list-style-type: none"> <li>• The player will experience the game from a first person perspective.</li> </ul>
Platform:	<ul style="list-style-type: none"> <li>• C++</li> </ul>
Device:	<ul style="list-style-type: none"> <li>• PC</li> </ul>

## Software

Below is a brief description of how the technologies mentioned in the executive summary were used in the project.

- Unreal engine 4
- Autodesk Maya
- Mixamo
- Blender
- Photoshop

### Unreal Engine 4

The majority of the game will be created using unreal engine as this is where the level will be created and all of the coding will be done. Unreal engine was the best choice of software to create the project in as it offers the best graphics from any available game creation software. This is evident in the project as it allows users to create dynamic environments with its sculpt tools and bring them to life using its foliage painter which will fill the map with trees and bushes.

Tasks to be completed in unreal engine:

- Level design
- Character/enemy implementation
- Game mechanics and overall coding of in-game items and functions

### Autodesk Maya

Maya was used to create the initial character model. It offers exceptional 3D modelling options and allows for the most accurate creation of the model for the game. As all of the characters in the game are based on a humanoid structure e.g. two arm and two legs, only one mesh was needed to be created using Maya and it could then be manipulated using other software.

Tasks to be completed in Maya:

- Creating 3D character
- Sculpting the mesh for the model

### Mixamo

Mixamo is an online 3D character creation tool. It also offers the service to auto-rig character meshes uploaded to the website so long as they meet their requirements.

Tasks to be completed in Mixamo:

- Create smoothing group for 3D model
- Auto-rigging 3D mesh with a skeletal mesh to allow for animation

### Blender

Although Mixamo offers its own preset animations to be used with any model rigged using its software, Blender offers the chance to create custom animations that users can create themselves. Once a mesh has been rigged with a skeletal mesh it is capable of being manipulated into making animations.

Models can be changed in Blender with the same options as those in Maya which is how each individual character was modeling from the same mesh provided by Maya.

Finally, Blender was used to create a UV map for the characters which allows for textures to be applied to their surface for customizable appearances.

Tasks to be completed in Blender:

- Modelling meshes further to create individual characters
- Animating skeletal meshes for custom character animations
- Creating UV mapping for character meshes
- Applying textures to the surface appearance of meshes

## **Photoshop**

Photoshop was used to create the custom skins for the character meshes. Once a UV map is created in Blender this can be converted into an image file which can be edited in Photoshop.

Tasks to be completed in Photoshop:

- Edit the colour of UV images to create individual textures.

## **Game Atmosphere**

### ***Character/ Units Concept & Description***

Below are the screenshots from the Unreal engine 4 character editor of the enemies in the game and the player character. There are two types of enemies in the tutorial, Zeniths and the boss, Angmar. All the character meshes were created from nothing using Autodesk Maya. They were then uploaded to Mixamo. Mixamo is an online character creating tool used to model 3D characters and provided a skeletal mesh. Once the mesh had been uploaded and passed the requirements needed a skeletal mesh was generated for the for the mesh. A skeletal mesh is a component that is added to any type of static mesh, the most common meshes being guns and human characters, that gives it a “bone structure”. This allows the components of the mesh to be manipulated by the armatures, “bones”, assigned to the limbs and other areas. Essentially this allows the character to be animated.

The character is then converted and added to blender. Using UV mapping to break down the surface of the mesh an image can be created for the character which depicts is entire surface in a flattened version. This image can be exported to Photoshop as a PNG and edited from there, allowing textures to be added to the image. With the textured image completed the UV can now be re-applied to the character mesh which will result in the images below and the entire creation is ready to be imported to Unreal Engine.



*Lynn*

The image above shows the character the player will be in control of for the duration of the game, Lynn. Also in the picture is the player's starting weapon, the "Blaster".



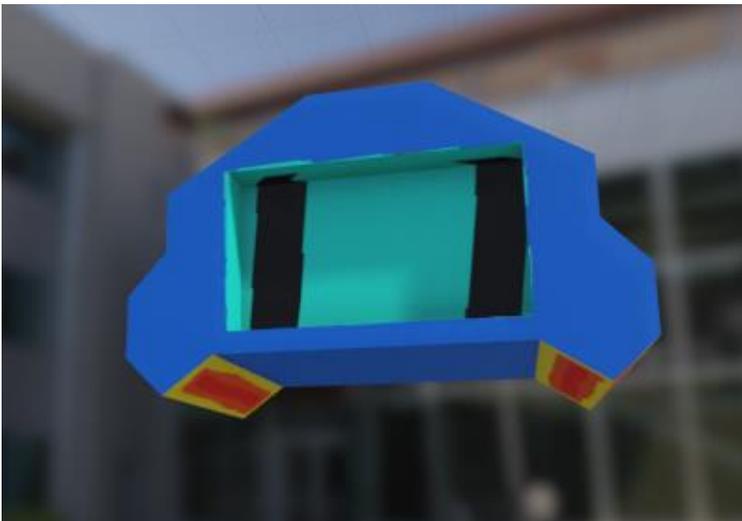
*Zenith*

The image above shows the standard enemy in the game, a Zenith. This is the first enemy the player will come across.



*Angmar*

The image above shows the mini boss of the tutorial level, Angmar. This enemy will be significantly stronger than the standard Zenith enemy.



*Finn*

The above image shows the friendly robot character Finn. Although not an actual playable character, Finn is the character that will act as the players guide and ally for the duration of the game, providing a tutorial on how the game works and setting the players objectives for the tutorial level.

## Level Description

Below are the screenshots of the map that are semi-completed in the game. The purpose of the screenshots being to give an understanding of the order the player is expected to complete the level and the general size and content of each section.

### Map Aerial View



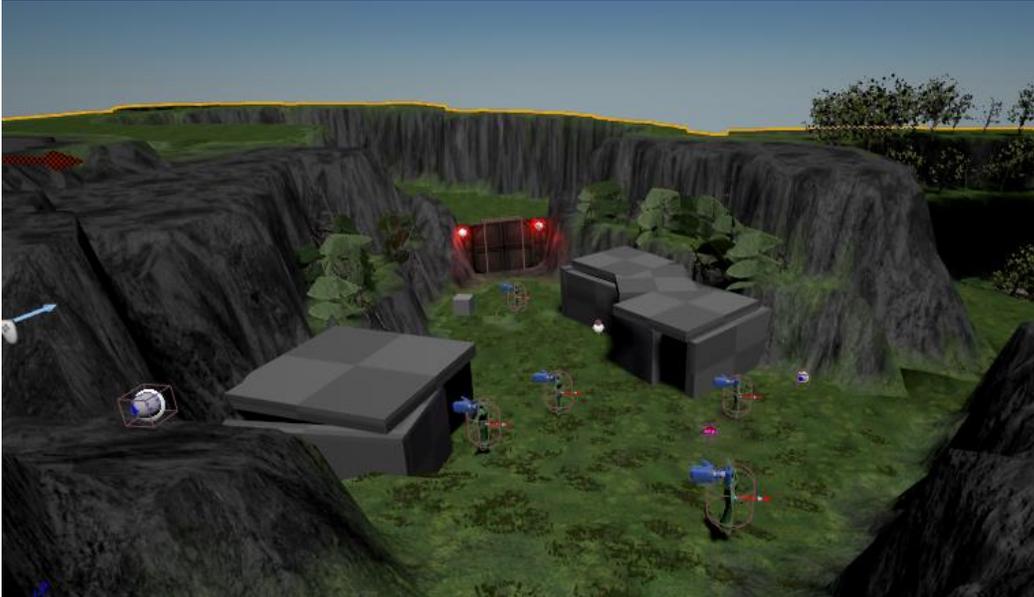
The aerial shot gives a scope of the entire area the player must complete. From this angle, not all details of each section are visible and as such there will be a description of each area provided as well as a closer screenshot. Each of the images provided have been brightened for the purpose of clarity as the lighting in the game is darker.

### Map Area One



This is the area where the player will start the game. They will first spawn into the black container positioned in the bottom right corner of the image. The area will have no enemies present as it allows the player to get their bearings. The area is dotted with flames to depict the crash of your space ship (black container). The player must make their way up the hill and through the crevasse.

### Map Area Two



In this area, the player will come across the enemy for the first time. There will be five enemies in area two. The area is essentially the center of the map, the player will emerge from the crevasse from the bottom central section of the image. They will have the option to travel to any of the other areas from here. The entrances are located to the left, in the center and to the right if the player wanted to go to area three, four or five respectively. The gate in this section (visible with red lights) leads to the final area, area 5, but is locked. This section also contains two health packs, two ammo packs and a power charge located in the building closest to the red gate.

### Map Area Three



Area three consists of a pit the player will have to jump into. There will be six enemies hidden in the forest area as well as two ammo packs, two health packs and a relic which the player will need to use to progress to area four. The purple pads that fill the area will deal 5 damage to the player.

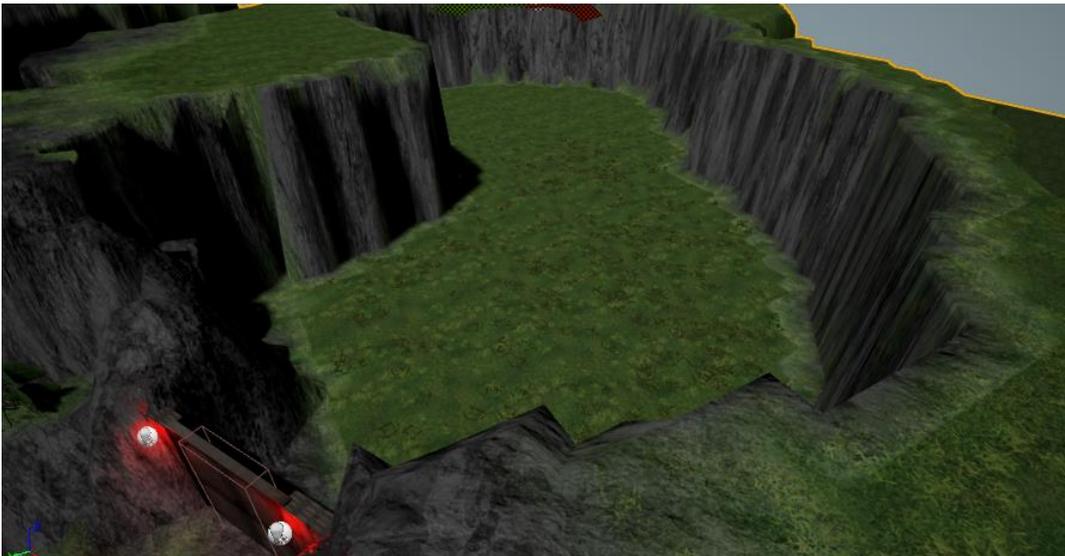
should they touch them. The exit to the section is in the bottom left corner of the image, a small path leads back up the cliff the player first jumped off.

#### *Map Area Four*



The entrance to area four is located through a cave entrance that the player will need the double jump ability to be able to reach the boulder that leads to it. After following the path downwards, the player will enter the pool of water. This is where the mini boss battle will take place. The only enemy in the area is Angmar, located on the left of the image. Also in the area are three health packs, one ammo pack, a haste rune and the key to the gate located in area two. After defeating the boss, the player will be free to collect the key and leave the area. The key is positioned under the boss in the image.

#### *Map Area Five*



Area five is the final area in the tutorial level, it is accessed by using the key collected in area four on the red gate in area two. It will consist of two more enemies and launch pad that will be located at the back of the area (center background of the image). There are no items located in this area. After defeating the enemies, the player must walk onto the launch pad and the game will end automatically. The end game widget will appear and the player can return to the main menu.

## Game Play

When the player first opens the game application they will see the main menu screen which will allow them to enter either the campaign or options screens. The player will play through a first-person camera perspective. Offset to the right-hand side of the screen will be the players equipped weapon, when they first start the tutorial this weapon will be the "Blaster". Throughout the tutorial and the rest of the game the player will need to jump over platforms and obstacles while defeating enemies, gathering weapons and leveling up their character. There will be at least two different enemies in the tutorial section, "Zeniths" which will be the weakest of enemies in terms of health points and the amount of damage they can deal out and "Angmar", which will have twice the amount of health points and can deal significantly more damage. The player will complete the tutorial level when they have defeated the last of the enemies and stepped into the designated area at the end of the map. The player will lose the game if their health is depleted to zero from either the enemies or the environment. There will be hidden collectibles around the map for them to upgrade their character.

The game options will allow players to change the screen resolution for what best suits them, giving users the choice between 1920 x 1080, 1280 x 720, 640 x 480 and fit to screen. The game will have a tutorial level as part of the campaign. The campaign will have a single difficulty to begin with but a harder difficulty is something that could later be added and more levels and a chapter select would need to be added to give users a choice in what level to play and to keep them interested.

Below is a list of the player controls:

W – Forwards

A – Right

S – Backwards

D – Left

P – Pause

B – Upgrade Menu

R – Reload

E – Action Button (Used for interacting with objects in game, e.g. opening doors etc)

Q - Melee

Spacebar – Jump

Left mouse – Fire weapon

## Key Features

The following section will list all the key features prominent in the game and outline why a potential user would be attracted to playing the game.

The game will feature a tutorial level filled with tips to get the player started on how to play. The tutorial level will contain 15 enemies, 14 "Zeniths" and 1 "Angmar" and it should take approximately 15-20 mins to complete.

It is filled with collectibles for the player to retrieve and use to upgrade their character. The "Power Charge" is used to upgrade the players' health or stamina, the user must choose between which one they would prefer, like the rest of the objects in the game it is then destroyed after its used. The "Relic" is used to upgrade the players' special abilities which can be accessed through the upgrade menu. The upgrade menu will allow the player to select between a list of options to

upgrade their character, allowing for a double jump, faster movement speed etc. Each Relic you collect will allow you to upgrade one attribute from the list provided on the menu.

It will also contain tutorials on how to use the menu system. There are two menus that can be brought up when playing the game, the pause menu and the upgrade menu. The pause menu will allow the player to access the options sub-menu and to exit back to the main menu.

The game would be attractive to users as it would be something they are familiar with already as first person shooters take up such a large section of the market. It is for this reason the game will feature controls and concepts users will know, a player should be accustomed to the concept of a first-person shooter

## **Selling Features**

The main selling feature of the game will be the concept that users will already be familiar with the type of game that is being created. Users would be willing to buy into something that they c=have seen before and that they know they have enjoyed before. Across most console platforms there has been successful titles launched that have now made an impression on consumers so much that they will be willing to continuously buy into that type of game. This is where Gemini will fit into the market, a first-person shooter consumers will be familiar with as this is the largest part of the gaming market.

## **Project Scope**

The scope of the project is to develop a first-person shooter game. The game will have the following:

- A game map and terrain made in unreal engine.
- At least two different enemy AI characters.
- A player controlled character, modelled in Maya.
- Collectible Items

Restrictions:

It will be necessary for the project that all the characters in Autodesk Maya as well as creating weapons and sound effects.

## **Definitions, Acronyms, and Abbreviations**

AI – Artificial Intelligence

HUD – Heads Up Display

## **User Requirements Definition**

The users' machine can run the minimal requirements of Unreal Engine 4, listed below:

- Minimum of 4GB of RAM
- Dual core intel or AMD processor
- NVIDIA GeForce 470 GTX graphics card or any card higher
- Windows 7, 8 or 10

Although for optimal settings recommended requirements are listed below:

- Ran on a desktop PC
- Quad-Core Intel or AMD processor
- NVIDIA GeForce 470 GTX graphics card or any card higher
- Windows 7 64 bit
- 8GB of RAM

The user will also need input devices such as a keyboard and mouse (a controller is not essential to play the game but would enhance a player's experience).

## **Project Design Report**

This section of the document describes how game objects behave, how they are controlled and what properties they have.

### **Design Version**

D1 – The base version of the games design. In it, the user interface is at its lowest level of design and it is only useful for navigating the game menus. The button images used have been created using the default settings in unreal engine and all the backgrounds are a simple black image. There are no introduction images or special effects.

D2 – This version of the project is after all the functions have been added to the game it is clear what each button will be doing. The button images are still designed as they were in the first version as it allows for a clean simplistic design that users will be able to easily navigate and understand immediately from looking at it.

### **Design Guidelines**

The overall design of the project is to be kept simple with a bright landscape and vibrant enemies. In regard to restrictions, the character and weapon designs will be kept in a simplistic design due to the time-consuming nature of photoshopping UV layouts to apply to these meshes.

### **Game Design Definitions**

#### Campaign

The user will control the character through a first-person perspective with their weapon offset to the right-hand side of the screen. There is a red cursor positioned in the center of the screen to indicate where the players' projectiles will go when they fire their weapon. The main focus of the gameplay is to destroy the enemies standing between you and the end platform. A secondary focus for the player is to collect all the upgrades hidden around the map. The player will win the tutorial level when they stand on the blue light located on the platform in the final section. The player will lose the game if at any point their health points are depleted to 0, this can be through enemies attacking them or through environmental damage e.g. Poison Grass. The player only has one life and will be forced to retry the tutorial level should they die.

This form of gameplay would be the same throughout each level in the game, the player must defeat enemies and reach a goal at the end of the level. Although each level will be different in the sense that there would be more difficult enemies to overcome and the design of each level would be different. The story for the game allows for a diverse change in scenery between the surface of the alien world where the tutorial takes place and the interior of the space station where the next chapter would bring the player.

### **Player Definition**

#### **Default Player Settings**

The details of the characters' settings when they first spawn into the level are as follows.

The player will spawn into the game inside their crashed space ship in the first area in the game map. Their only option from this point is to exit the ship and follow the path towards the next area, there will be further instructions as the player gets accustomed to the controls of the game.

By default, none of the players' special abilities will be active at the beginning of the game. The player will start the game with 100 health points and 100 sprint points. The 100 sprint points would translate to approximately 3 seconds of sprint time. The players sprint ability will be stopped when their points reach 0 but the attribute will regenerate automatically. The players' health will not regenerate over time and will need to be replenished using health packs. They will start the game

with 20 bullets for their blaster. The maximum amount bullets the Blaster can hold is 5, the player will then have to reload to continue firing.

The player can win the tutorial level by reaching the end platform in the final section of the level and walking onto the blue orb located in its center. Defeating all the enemies in the level is not actually a requirement to finishing the game and the player can technically complete the level without having to defeat any enemies or even firing their weapon, although this is not advised. If at any time the players' health points drop to 0 they will lose and be forced to try again.

At the beginning of the level a widget will pop up to explain vital parts of the story so the player can grasp what it is exactly that they will be doing. This information will include how the events leading up to the beginning of the game unfolded, the situation the character is currently in and a vague description of what they must do. Shortly after this the player will be given more exact objectives on how they are expected to complete the level, and instructions on how to control various aspects of the game.

## Player Properties

- Health

The player will have a health bar that will hold 100 health at its maximum. The health bar governs whether the character is alive or not. It can be affected by the player taking damage from either enemies or the environment and can also be recovered using health packs found in the game.

- Weapons

The main source of damage in the game. The player will use their weapon to destroy enemies. The standard damage that is dealt by the players "Blaster" weapon is 30, meaning if a standard unit e.g. a Zenith alien, was to have 100 health points it would take 4 direct hits to destroy the unit. If another level was to be added to the game more weapons could be introduced, e.g. the alien rifle, featured in the "Character/ Units Concept & Description" section of the document.

- Actions

- Keys and Gates

The player will need to collect keys to open certain gates, the keys and gates will be colour coded to avoid confusion. The tutorial features a red gate with a corresponding red key. The player cannot enter the red gate until the red key is in their inventory. To collect the key the player must walk over the key and it will be added automatically. Without the key the gate will remain locked. It will change the players state by changing the variable for the amount of keys the player is currently holding.

- Power Charge

A power charge can change the players state through either their stamina or their health. It will do this by increasing either of them by 25 points.

- Upgrades

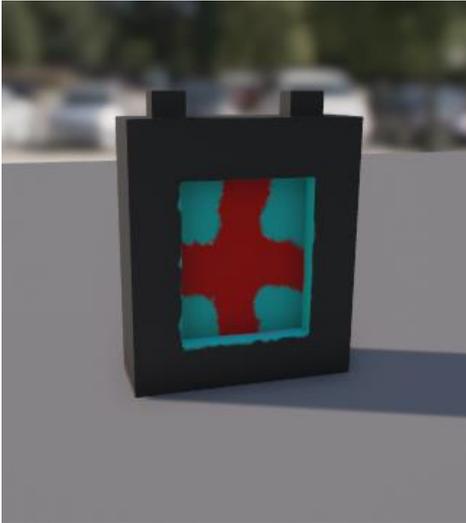
Upgrades can be applied from the character upgrade menu and can come in different forms. For the tutorial level I will use the example of the double jump ability. Once activated it will change the player state by increasing the max jumps counter by 1.

- **Runes**  
Runes apply temporary changes to the players' state. These changes usually push the character to an extremely high statistic in a certain field but the affects only last for up to 10 seconds. In the case of the tutorial level, a haste rune is present. A haste run will change the players' maximum movement statistic from the base speed of 600 to 1500 which it 2.5 times faster.

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

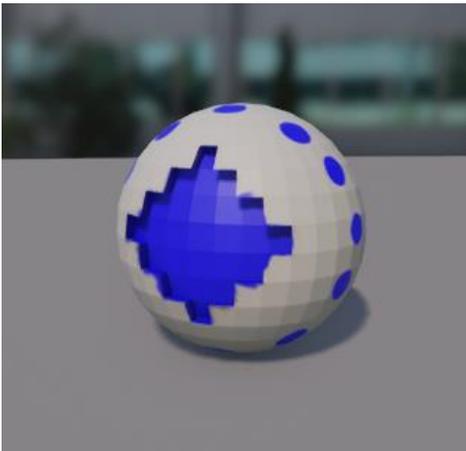
Below is a list of the power-ups, items and collectibles available for the player.

### **Health Pack**



There is one type of health pack in the game which it pictured below. A player simply has to walk over the box and it will instantly be consumed. This will heal 20 health points back to the player unless they are already on full health, in which case the box will be consumed and no health will be returned.

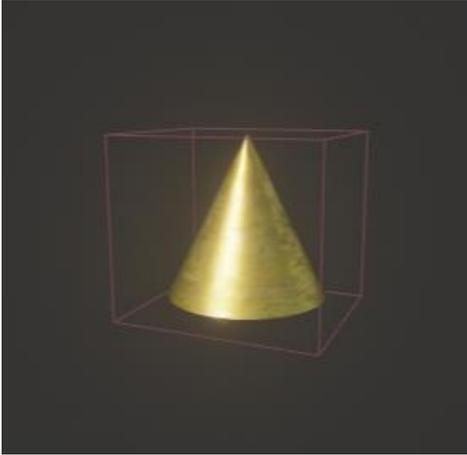
### **Ammo Pack**



There is only one type of weapon in the tutorial level that requires the collection of this ammo pack and that is the Blaster. A player must walk over the object for it to be collected. Upon collection, it

will fully restore all of the players' ammo for the blaster which is 20 bullets. If there is no ammo to be replenished then the ammo pack will still be consumed.

### **Relic**



The Relic is used to upgrade the special abilities of the player. There are limited amounts of relics hidden throughout the game that users will need to search for. Upon collecting a relic, the player upgrade menu will open automatically, prompting the player to expend the point they received. When an upgrade is purchased the menu will close and the player must find another to purchase further upgrades. In-game the relic will have bright aura surrounding it to make it more visible to players.

### **Power Charge**



The power charge is another special collection item. The charge itself is positioned on top of an altar style platform. The player must be within range of the glowing blue orb and press the action button in the game to interact with the item. Upon interacting with the object a new menu will be opened. This simple menu will ask the player whether they would like to upgrade their health or the stamina. When either button is selected, the menu will close the blue orb will be destroyed to prevent further use. The players max health/stamina should be upgraded by 25 point instantly.

### **Haste Rune**



The haste is a consumable item within the game that will be activated instantly when a player walks over it. Were the game to have more levels, more runes could be introduced such as Power and Healing runes which would provide the character with increased damage and health regeneration respectively. The haste rune provides an extreme increase to the players maximum speed variable but the effects only last for up to ten seconds before it is disabled.

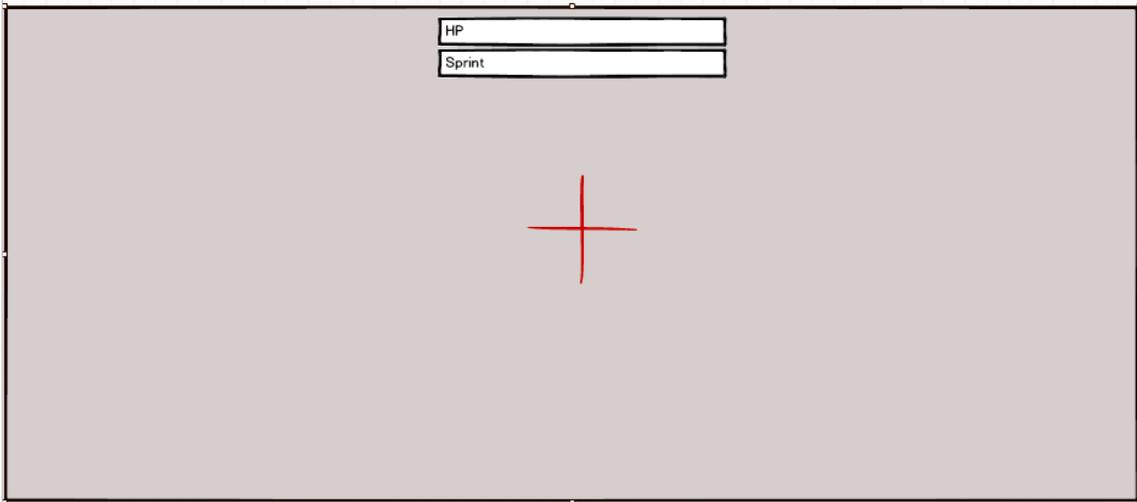
#### Red Key



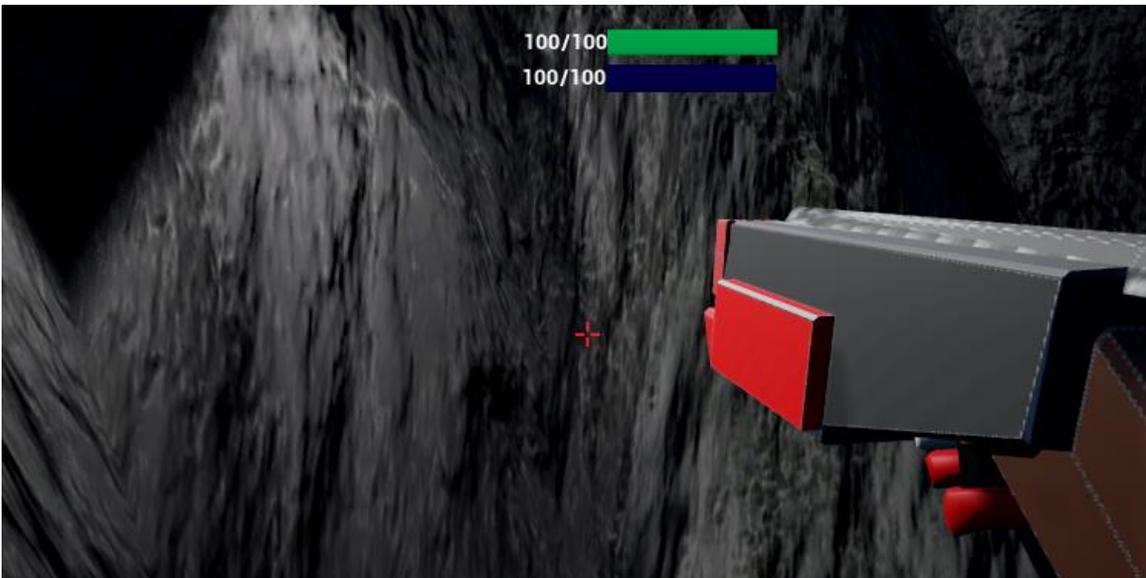
The tutorial level feature a red gate that is only able to be opened once the player has a red key within their possession. The red key doesn't provide any changes to the players' state statistically but allows for the gate to be opened.

## Heads up Display

For clarity, below is a mockup of the players Heads up Display in the game and a screenshot of how it looks when the player is in the game.



The image above is a mock-up of the player view. It includes a health bar and stamina bar in the top of the image and provides the player with a red cursor to track where they are shooting. Below is a screenshot in game of how the final player view turned out.



The health and stamina bar remained in the same position as well as the cursor but the players weapon (Blaster) has now been added which takes up a large portion of the players view.

## Implementation

The implementation section will describe how each of the functions within the game are processed and will also contain screenshots of the code from unreal engine to highlight this.

## Character Implementation

The section of implementation will break down the components that create the main character and explain the functions behind it.

## Character Animations

The character has been set up with a list of functions that were tied to input actions to play when certain buttons were pressed or events took place. All of the animation have been created using Blender and none of the default animations provided by unreal engine have been used in the making of the game.

### Idle

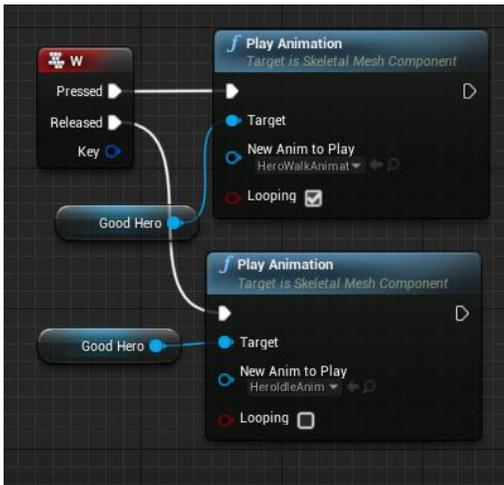
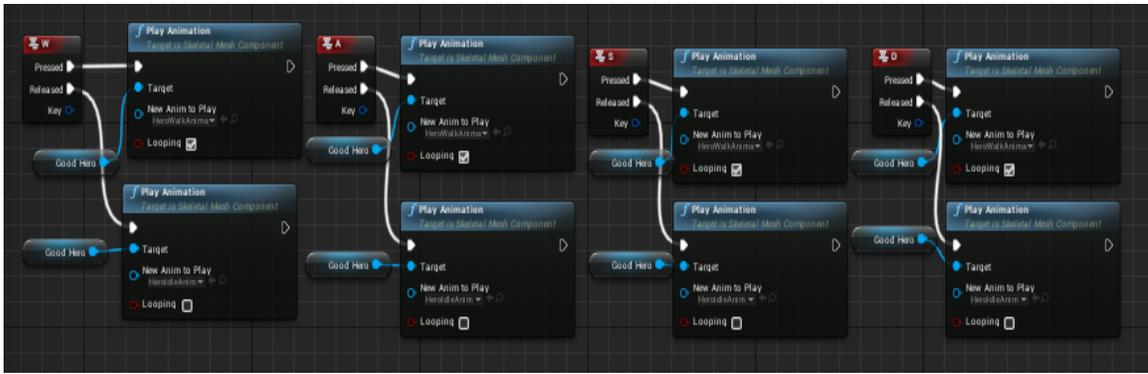
When the game is launched, the idle animation is activated immediately. The event "EventBeginPlay" is the first event to be called when the game starts, this is why it is being used in this instance. The idle animation it used to break the character model out of its default pose. This default pose or "T Pose" is used in the modelling of the character in Blender and Maya and is effective for accurately creating the look you desire but is not required once in game. It is also critical in creating a skeleton for the mesh.



EventBeginplay will activate the idle animation but as there is no specific action to stop the animation from running the function will always be called again. This means that even if other animations are called during the game, whenever the animation has played itself all the way through the idle animation will take effect again so when the character is not performing an action he will be in an idle state. This can be achieved by selecting the looping checkbox on the Play Animation function.

### Walk/Sprint

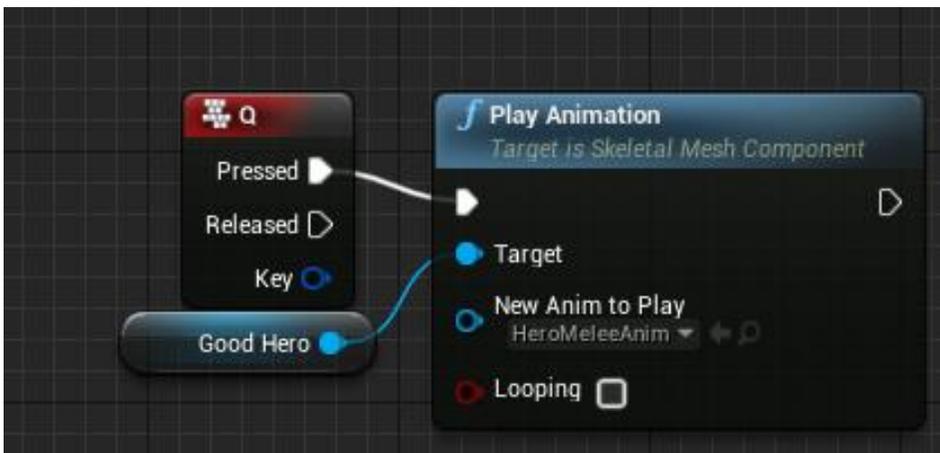
The walk animation is called when the W, A, S or D key is selected as seen in the code snippet below. This will play a loop of the animation for the player to walk in any direction. As a catch method, when the button is released a new event is called which will play the "HeroldleAnim", tis will return the character model back to their original idle state.



The image above shows the four buttons that are connected to run the walk animation so it will play no matter the direction the character is moving. On a closer look in the image to the left it is easier to see the pressed and released commands in the W key button function and also that the target in this instance is the "GoodHero" which represents Lynn, the player character.

**Melee**

The melee animation is simply called when the Q key is selected and plays itself once. It will remove the players weapon from the viewport and thrust his left are forward. The tutorial for this is not covered in the tutorial and the action does not deal any damage to an enemy until the player starts the campaign.



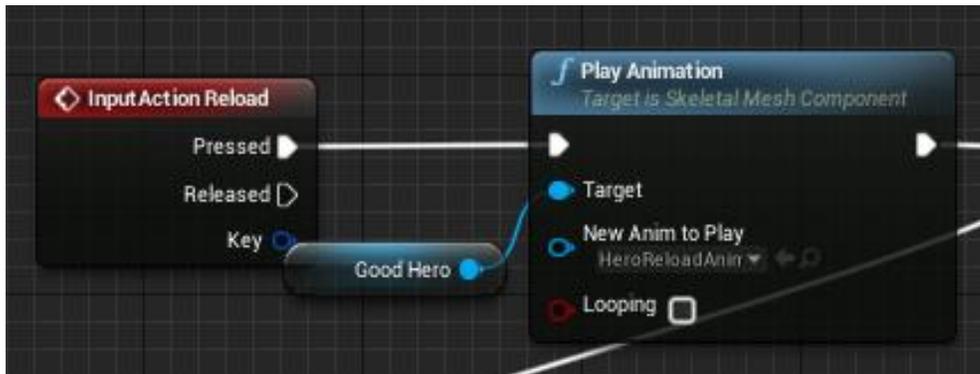
### **Jump**

Much like the melee animation the jump animation uses the same code structure and will be played whenever spacebar is pressed. An upgrade for the player in the game allows for a double jump action to be taken, this simply repeats the animation when the character takes its second jump while midair.



### **Reload**

The final animation for the player character. The animation works as part of a larger code network which will be explained under the weapons section of the implementation. The animation plays once and has the character grab ammo from his waist and put it in the gun. The code structure for reloading follows the same as the reload and jump animations.

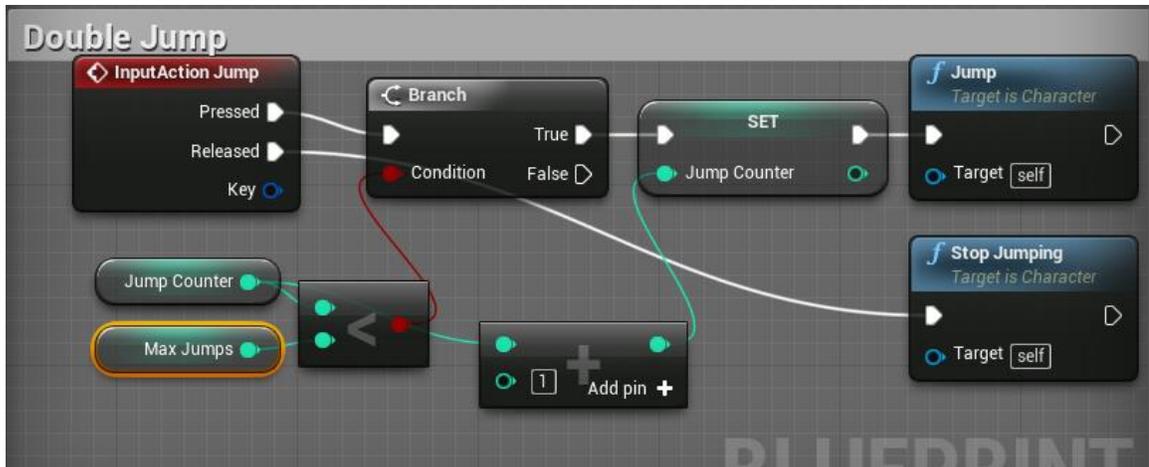


## **Player Properties**

In this section, the basic player properties and some of the actions that change them will be explained. Most of the actions described will be revolving around the players' movement, including jumping, sprinting and the players' stamina system.

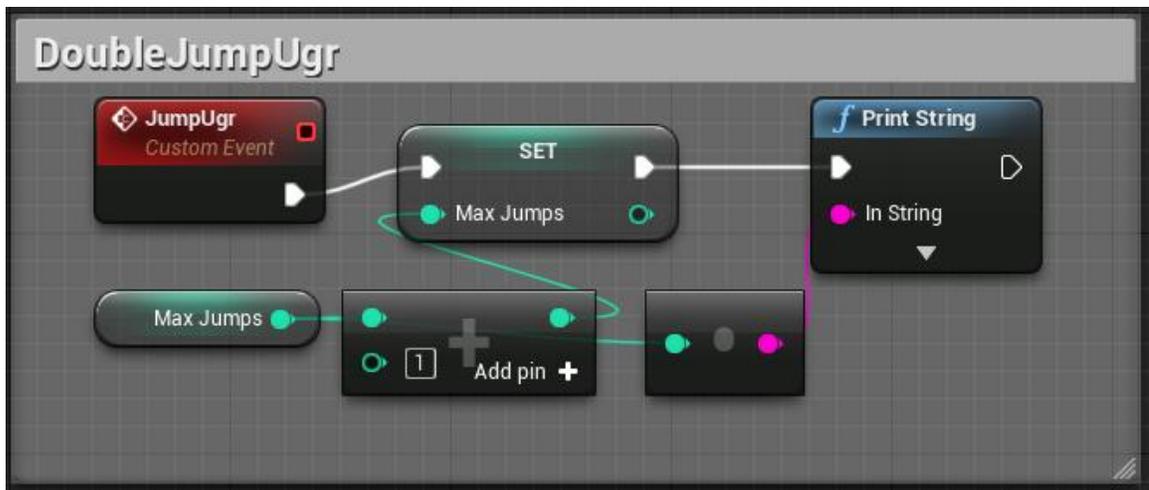
### **Double Jump**

The double jump feature is, as expected, an extension on the jump feature in the default unreal engine package. Jumping is an automatic function in the game already applied to the character. The code shown in the sample below simply adds a counter to the number of jumps on a character.

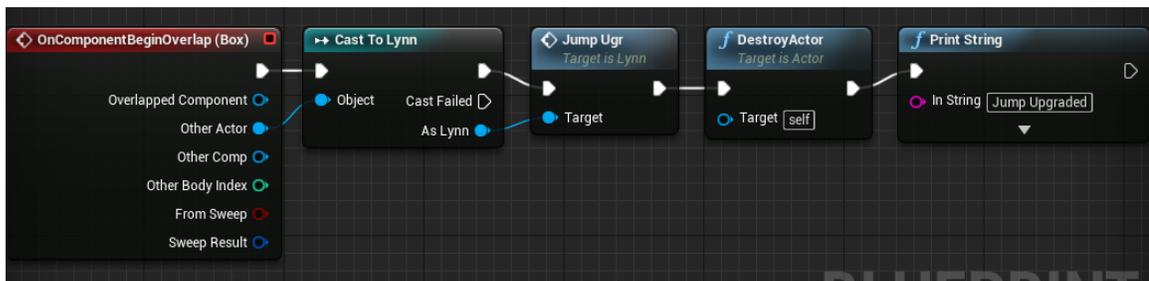


The "Jump Counter" is an integer variable that will be increased by one when the player makes a jump action and sets this as the number of jumps taken. The variable is reset when the button is released and the "Stop Jumping" function is activated.

The second variable in the image "Max Jumps" is also an integer and the oval shaped Max Jumps that is highlighted is simple a get method to get a reference to the object. In this instance, the default setting for the Max Jump variable is set to one. This means that the Jump Counter must always be less than the Max Jumps so a player can only make one jump by default.



The double jump upgrade is how the max jump variable is changed for the player. It is a custom event played from the jump upgrade collectible in the game. When it is activated it simple get a reference to the Max Jumps variable and permanently increases it by one for the player and prints this to the screen.

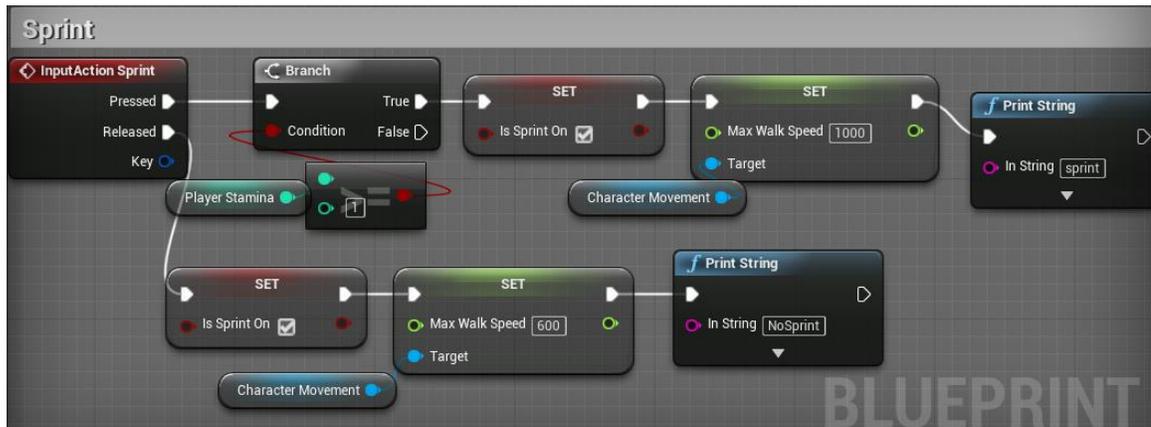


The image above shows the jump upgrade item. Much like a lot of the items in game when the red node, OnComponentBeginOverlap, is activated the event will play. In this case, the component is

the box collision that has been set with the object. When the player comes into contact with the box it will target the player "Cast to Lynn". This means we can now run events that are located within the Lynn Character Class. It will now run the custom event JumpUgr in the character class and upgrade the players jump, as stated before. Finally the Upgrade Item will destroy itself and print to the screen that the users jump have been upgraded.

### **Sprinting**

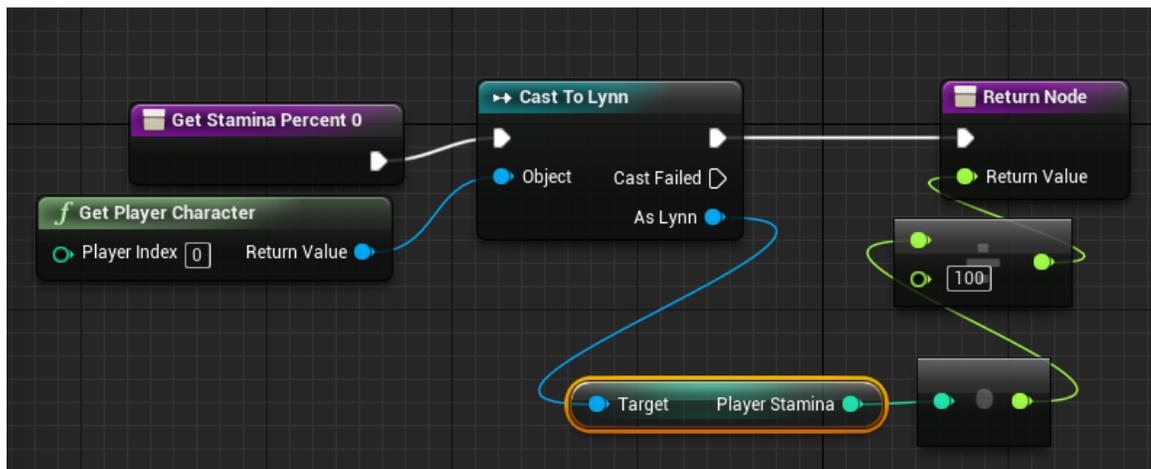
The input action Sprint node comes with two parts that are best explained separately. When the button is pressed, the branch node check whether a statement is true or not, in this case it checks if the players' stamina is greater than or equal to one e.g. does the player have any stamina left. It then sets the IsSprintOn variable to true and changes the variable for the max walk speed to 1000 by getting a reference to the Character Movement, this is an increase to the standard movement speed. It then prints this information to the player HUD.



When the button is released or the player has no stamina remaining, it will get a reference to the Character Movement variable again and set the max walk speed back to 600, the base movement speed, and print to the screen that the player has no stamina remaining.

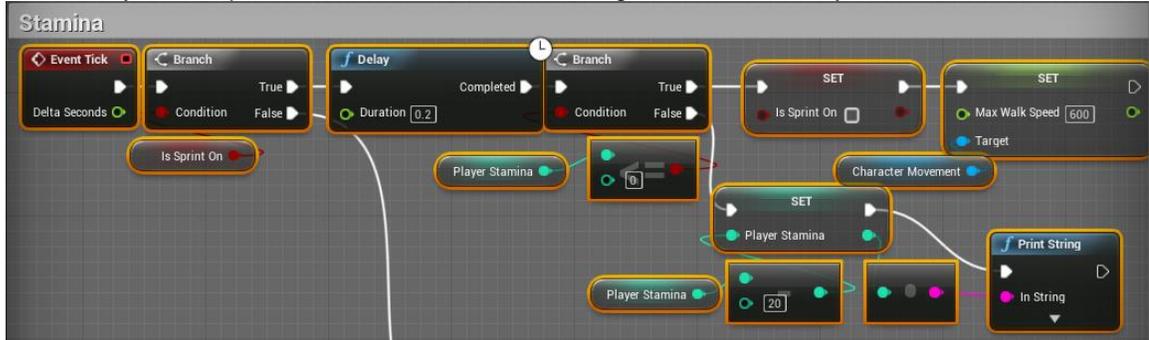
### **Stamina**

The stamina system on the players HUD work the same as the health system, by using a percentage bar, but how it is depleted is the difference.

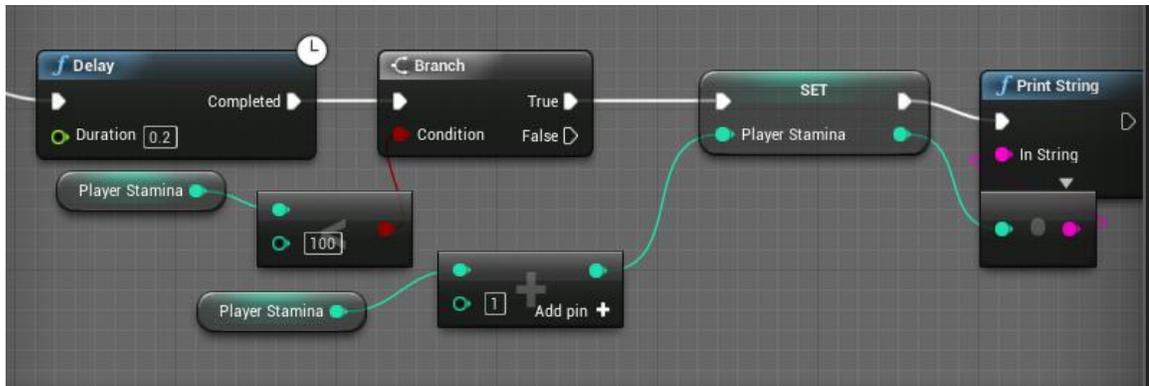


The event activating this function is the Event Tick node which is called every frame per second. The function can be easily described in two parts. Firstly, if the IsSprintOn Boolean variable set to

true and if the player stamina is less than or equal to zero then the players IsSprintOn variable will be switched back off and the max walk speed set to the standard. However, if the players' stamina being less than or equal to zero is set to false then the player is in fact sprinting. This means the player stamina needs to be set, in this case the stamina is being reduced by 20 and printed to the screen. This is being call after the delay of 0.2 seconds each time.

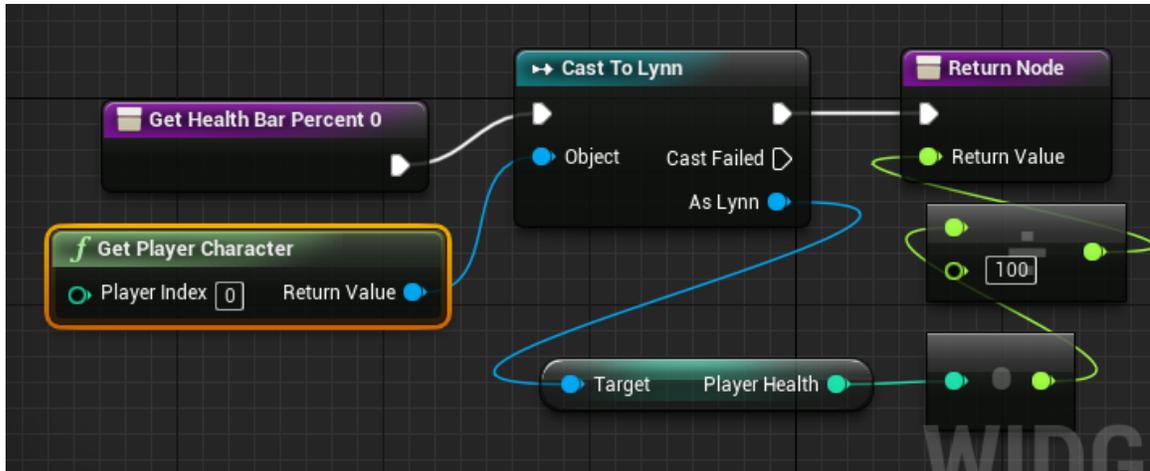


Now back to second option of the player IsSprintOn being set to false. If the player is not sprinting then his stamina is set to be regenerating and this is seen in the code below. After a Delay of 0.2 seconds the stamina will start regenerating. If the Player Stamina is less than 100 then the if statement is set to true and the Player Stamina will be set to increase by 1 every tick and print the result to the screen.



## Health

The image below is taken from the player HUD and represents how the players' health is displayed on screen. The health bar percent is a percentage bar located on the HUD that is running as a function. It works by getting a reference to the Player Character and casting to the character Lynn so it can also get a reference to the Player Health variable. As the player health is a round number and the percentage bar calculates in decimals the player health is divided by 100 before it is returned to the node. All this does is simply bind the two values together so when a percentage of the players' health is lost it will be deducted from the percentage bar.

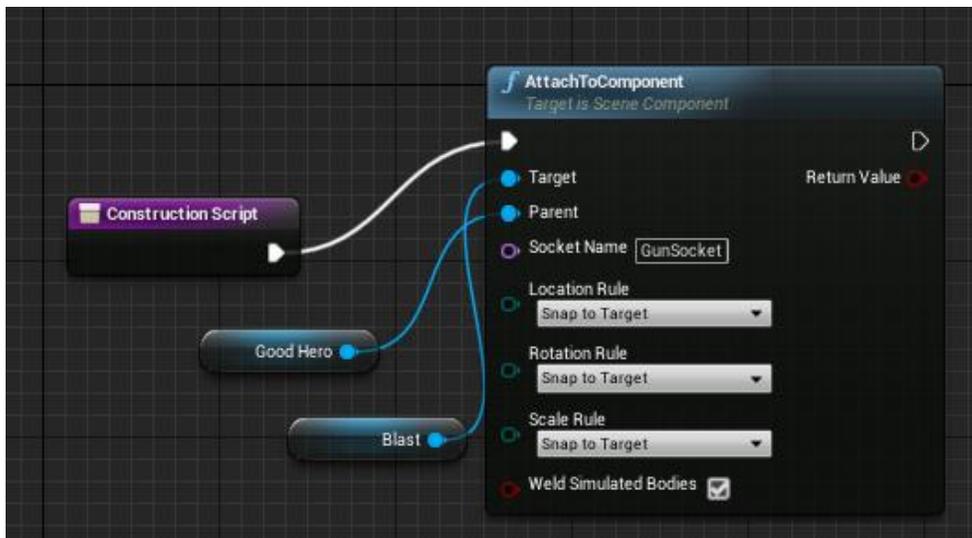


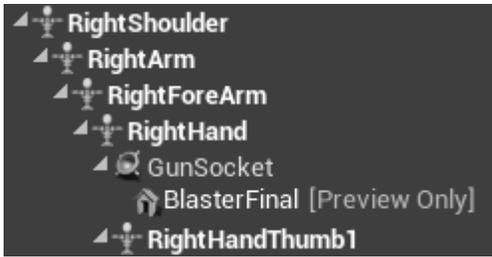
## Weapons Implementation

The following section will cover the fundamentals of how the weapon in the game functions including shooting and reloading as well as explaining how the character and skeletal mesh interact with the weapon.

### Hold Weapon

The weapon in Gemini was created as a separate static mesh to the player character, meaning it is not a part of the character skeleton and the two needed to be bound together for the purpose of the game. The code below shows a function in the Player character class that is being activated upon construction of the character. The `AttachToComponent` function will connect the weapon to the character as the Parent class is the character and the target is the weapon itself. The weapon is being attached to the socket "GunSocket" which is part of the skeletal mesh structure. The location, rotation and scale are all set to snap to target and weld simulated bodies will allow the weapon to snap to the position it's been set to in the skeleton editor and stay there.





The image on the left shows the skeleton structure for the character's skeleton. To create the desired effect a socket must be added to the right hand, before the RightHandThumb1 joint. This is the socket that is being referenced in the construction code. A preview mesh can then be added to the socket, this can be any static mesh within the unreal engine project. The BlasterFinal represents the players weapon which has now been added to the character skeletal mesh. Note,

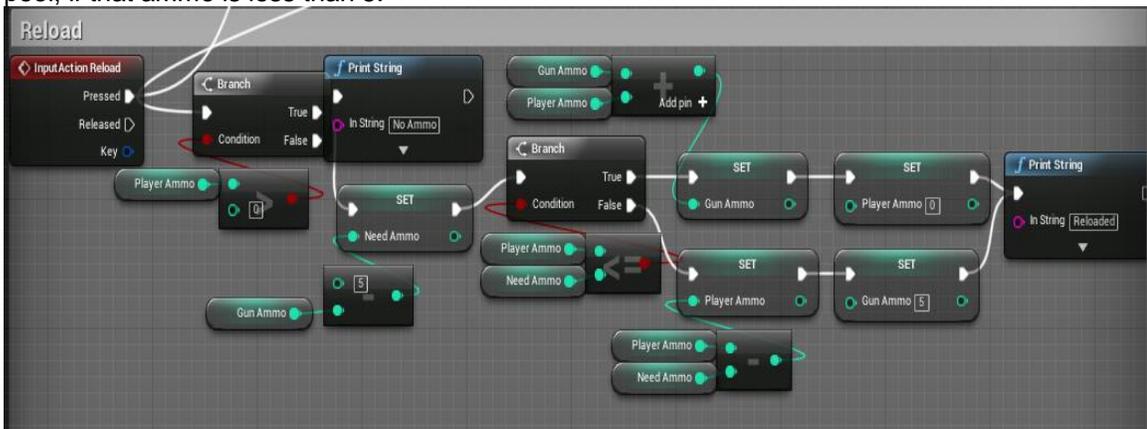
in some cases the static mesh may not be in the right scale, rotation or location. In this situation, it is important to keep trying different angles and scales in the editor viewport so that the weapon can fit normally into the player's hand and will look natural to the user when they are playing the game.

### Reloading

The load action is called when the player presses the R key. The first branch states that if the integer variable Player Ammo is not greater than zero then a message will be printed to the screen informing the player they have no ammo. After this there are three integer variables brought into the equation:

- **Player Ammo**  
Represents how much total ammo for this weapon the character can hold at one time. This value is set to 20.
- **Gun Ammo**  
Represents the maximum amount of ammo that can be held in in the gun. This value is set to 5.
- **Need Ammo**  
Represents the amount of ammo that is needed to reload the gun. This value is also set to 0.

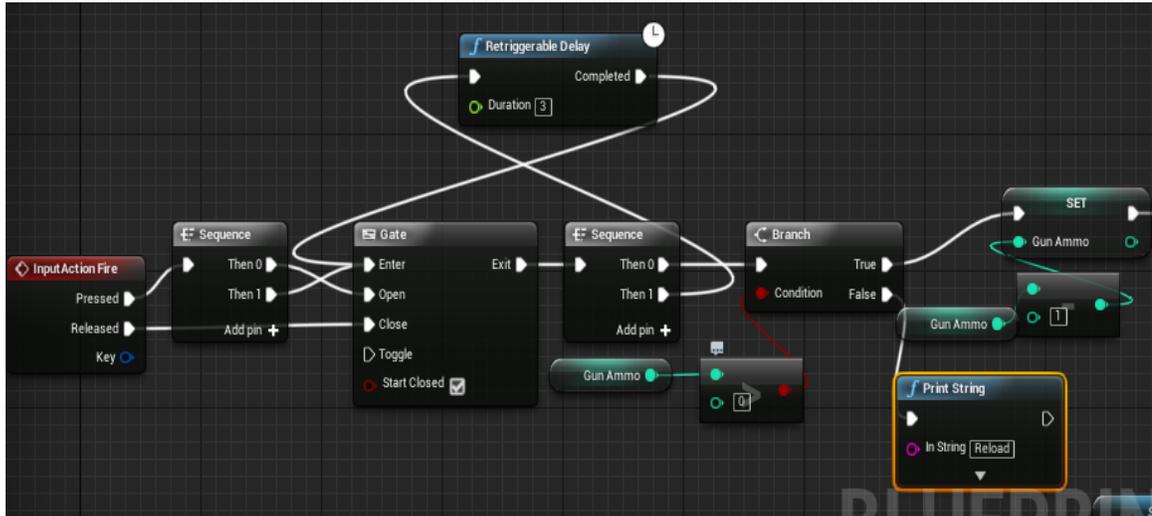
Next the set Need Ammo variable is set as 5 minus the Gun Ammo, leaving us with 0, meaning you need ammo when you have 0 bullets. Moving forward to the Branch statement in the middle of the image. The branch is set to true when the Need Ammo is less than or equal to the amount of Player Ammo. Regardless of whether this statement is true or not the gun will be reloaded with the ammo remaining to it in the Player Ammo. If it is true it will be reloaded with a full 5 bullets if it is false it will be reloaded with the whatever number of bullets is currently remaining in the ammo pool, if that ammo is less than 5.



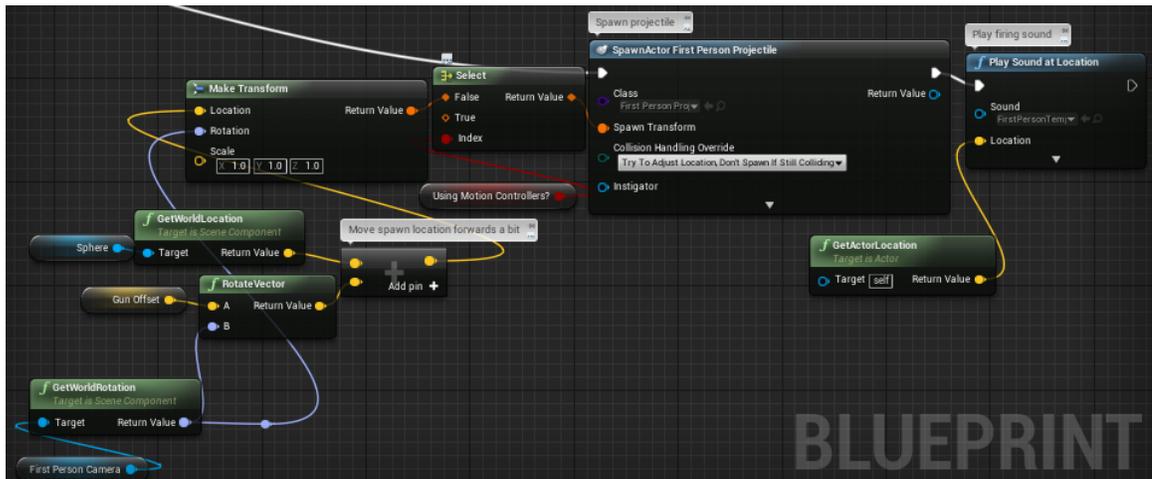
## Shooting

The shooting mechanic is a default function part of unreal engine but has been modified for the purpose of Gemini. The code sample below has added a loop sequence to the firing of the weapon. In this loop sequence is a delay of 3 second which effectively slows down the rate of fire on the gun.

The Branch statement is checking whether the player needs to reload. If the gun ammo being greater 0 is set to false then a message telling the player to reload is printed to the screen. If it is set to true then it will reduce the gun ammo by one until it needs to reload.



The second section of the code relates to the projectile itself. It is simply getting the location and rotation of the gun offset, as this is where the projectile will spawn from and getting the world location so it knows where the character is. The "Spawn Actor" function will get a reference to the class First Person Projectile and a reference to the ActorLocation. It will then spawn the projectile and play the specified sound using this information.



## AI Implementation

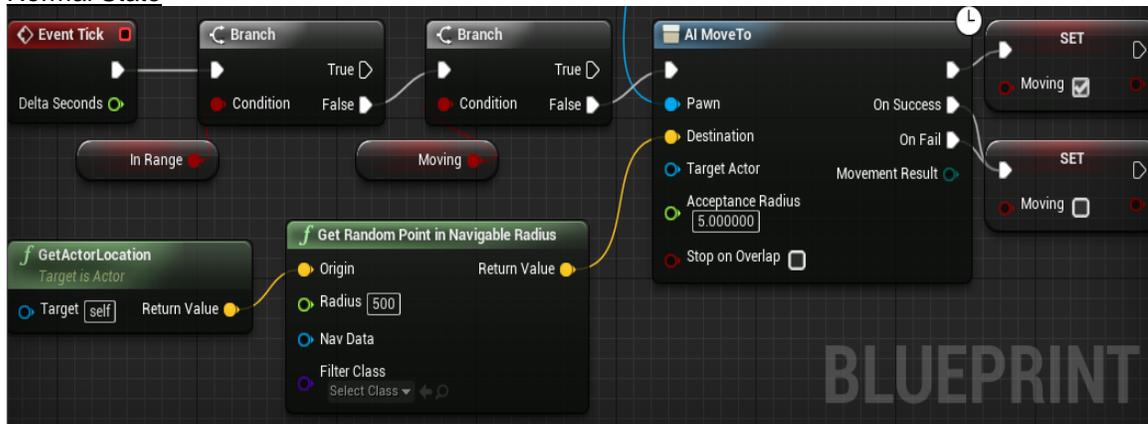
The following section explain how the enemy artificial intelligence operates in the game. This will include how the AI will sense and attack the layer and how the player will destroy the enemy.

## Pawn Sensing

The Pawn Sensing function can be broken down into two separate phases for a clearer way of explaining how the process work, the normal and the detection state. To set up the ability for the AI to move a navigation box must be set up on the map, no coding is needed behind this, it will just dictate where the AI can move to as it will not leave the confines of the box. The image below shows the navigation box for area 2 in the tutorial, this highlight area is as far as the AI can move.



## Normal State

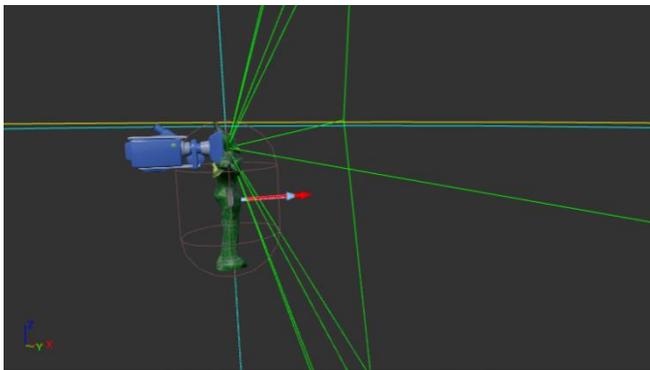
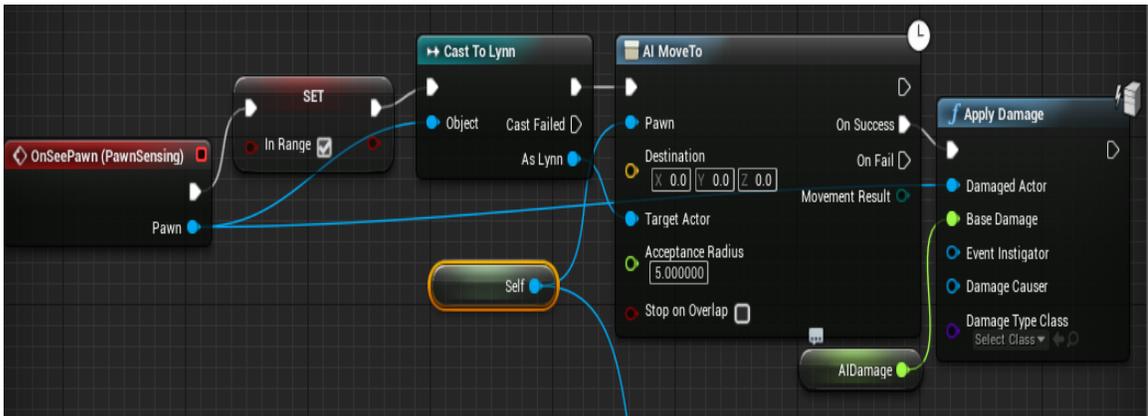


The AI is set to roam whenever the player is not within the vicinity of its navigation area. This is the AI's normal state and as such they will never be standing still in the same area. The InRange Boolean variable is used to check if the player within the navigation area and also within the AI Pawn Sensing radius and the moving Boolean variable checks if the AI has a set destination to move to. If both variables are then set to false the AI is given a command to move. This is generated by the Get Random Point in Navigable Radius node, which essential does exactly what the name of the node indicates. It will get the origin of where the actor is by referencing the ActorLocation and give it another random location in the navigation area to move to. To ensure the AI keeps moving, on both success and fail the moving variable will be set to false so the AI will continue to receive new destinations.

## Detection / Attacking State

The second phase of the AI movement is when it detects a player using its Pawn Sensing. The event is activated as soon as a player enters the navigation area, is caught in the Pawn Sensing radius and the InRange variable it set to true. Lynn is set as the target for the pawn, as such in the AI MoveTo function the pawn is set as itself and the target is cast to Lynn. This will give the world location of Lynn to the AI allowing it to continuously chase the character.

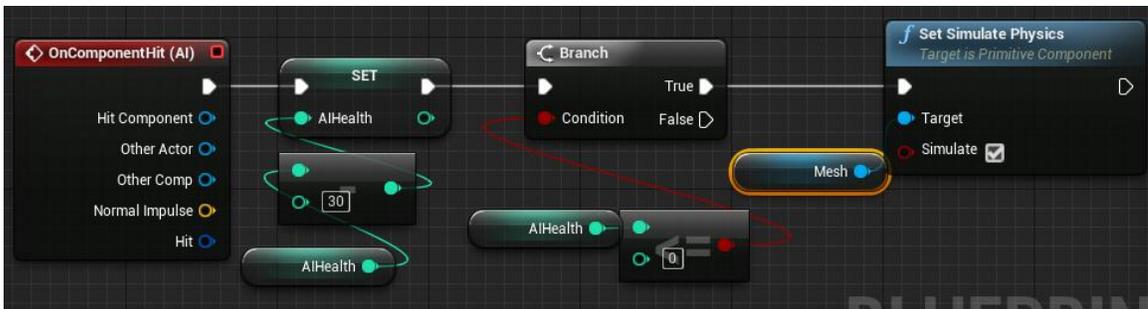
The new variable brought in at the end of the function is the AI Damage node which it set to 5. The Apply Damage function allows the AI to deal 5 damage to the player on successfully hitting them.



The image to the left shows the pawn sensing component as part of the character editor. The line branching out from the character's head indicates its Pawn Sensing, this can also be described as its field of vision. Its field of vision will move as the AI moves from each random destination, as it is connected to and pointing out of the front of its head. This makes the AI almost impossible to avoid.

### ***Destroy AI / Ragdoll***

The character has the ability to destroy the AI before they chase them down and start dealing damage. As described in the Character Collision section, the AI also has a collision box that will react to the First Person Projectile class encountering it. When the projectile overlaps with the box the AI, health will be reduced by 30 point, The maximum AI health is set to 100 so 4 direct hits will destroy the AI. The branch statement is the same as the Player Health branch checking if the AI health is less than or equal to 0, if true then the AI will be destroyed. The Simulate Physics function will have the AI fall to the ground in a ragdoll style effect and then disabled instead of just being outright destroyed and removed from the game world.



## Game System Implementation In-game System Interaction

### Character Collision

Before describing how the player interacts with the objects in the game, this section first explain how collision boxes work, as this is the basis for most player interaction.



The image to the left represent a collectible object in the game, this could be any object like a health pack, haste rune or environmental damage object like the Dmg\_Floor. The object will always be surrounded by a collision box as highlighted in the image to the left. These collision boxes come with their own set of functions that will activate an event. The most used function being the OnComponentBeginOverlap, this allows the event to start when any actor overlaps with the box. An example of this can be seen in the image to the right.

### Floor Damage

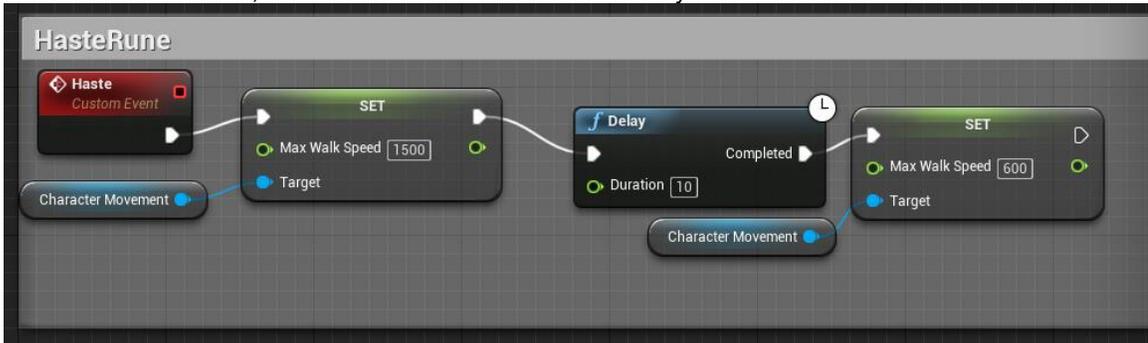
The following code explains the process that happens when the player stands on the purple grass located in area three of the tutorial level.

The code itself is located in the Lynn Character Class. The event is activated when the player overlaps with the collision box for the Dmg\_Floor, as described earlier in the Character Collision section. The event casts to the Dmg-Floor class and gets a reference to the player health variable. It then uses the set method to reduce the player's health by 5 each time it overlaps the box. After it has reduced the health, the branch statement checks if the player's health is less than or equal to zero. If this is true, the player character is destroyed.



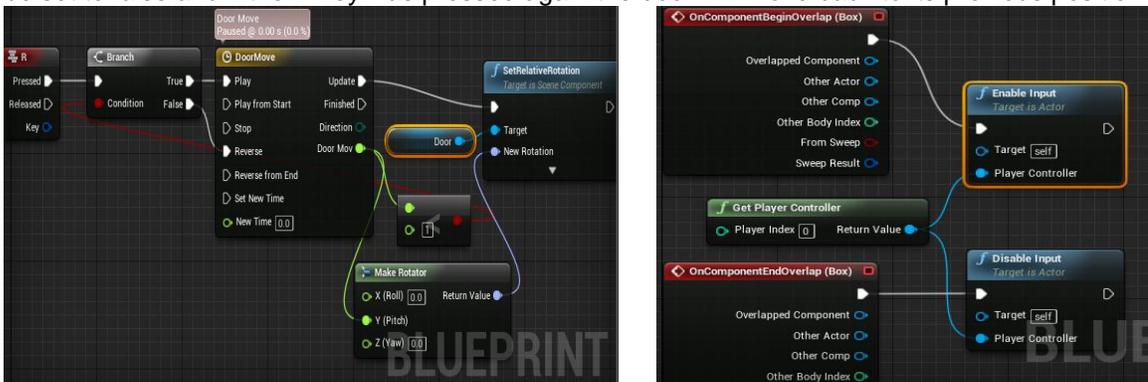
## Haste Rune

The haste rune is activated when the player overlaps with the haste rune item. It is a custom event that is activated in the haste rune class and is run in the Lynn character class. This will get the Character Movement, increase it to 1500 and after a delay of 10 seconds it will reset it to 600.



## Gate

The works on the same principle as the other overlap object but requires the player to press the R Key whenever input has been enabled. Once these requirements have been met the door will set a need rotation location and move to that position. Once the door is in that position the branch will be set to false and if the R key was pressed again the door will move back to its previous position.



## **Project Technical Report**

### **Requirements Specification**

This requirements specification section covers some of the areas the user will learn about and what they will achieve from playing the game:

- A user should be able to competently control the game after completing the tutorial level, which in estimation will take up to 10-15 minutes.
- They will have a firm understanding of what will be taking place in the story and the background of the world.
- Will understand all the game rules and functions.

### **Functional requirements**

Requirements that will be covered in this section:

1. Start New Game
2. Save Game
3. Load Game
4. Exit Game
5. Change Game Settings

### **Requirement: Start New Game**

#### **Description & Priority**

This functionality is essential as a user will need to be able to begin a new game and start playing the tutorial level. It is important a user is guided through how to play so this requirement is of the highest priority.

#### **Use Case**

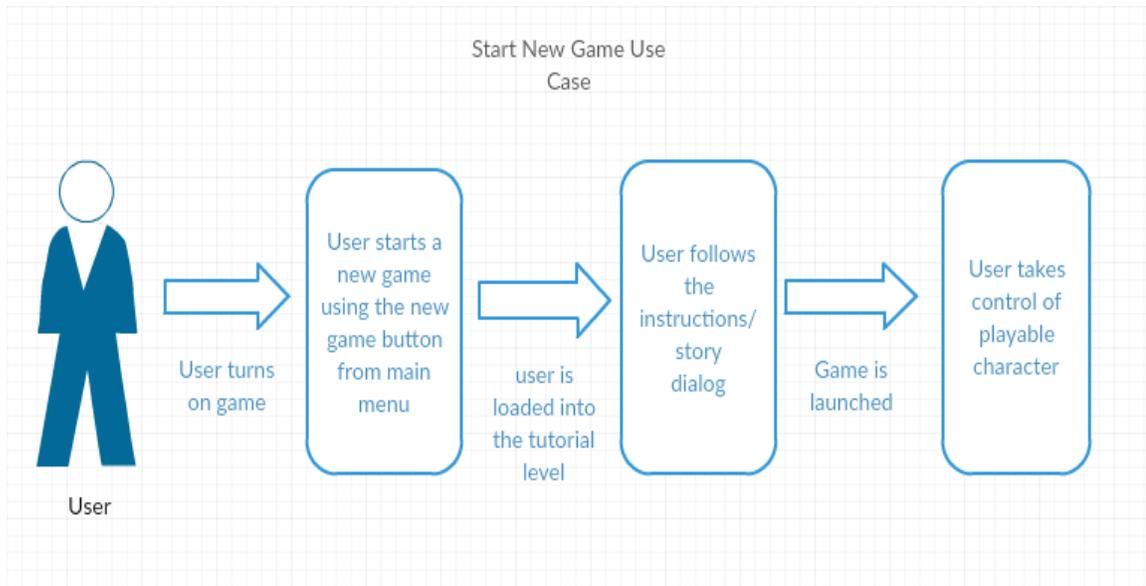
#### **Scope**

The scope of this use case is to allow the player to start a new game and play the tutorial level.

#### **Description**

This use case describes the actions a player will take in order to start playing the first level of the game.

#### **Use Case Diagram**



**Flow Description**

**Precondition**

The game is installed on the users' device and they have a way of activating the game, IE desktop shortcut, game file etc.

The users' device must at least reach the minimum requirements needed to run the game, as stated earlier in the document.

**Activation**

This use case starts when a user turns on the game and selects the new game button from the main menu.

**Main flow**

1. The user turns on the game.
2. The system loads the user into the main menu.
3. The user selects the start new game button.
4. The system loads the tutorial level and instructions dialog.
5. User accepts Instructions.
6. System Launches game level and allows user to begin controlling character.
7. Player uses input device to start playing.

**Alternate flow**

Player skips tutorial

1. The user turns on the game.
2. The system loads the user into the main menu.
3. The user selects the start new game button.
4. The system loads the tutorial level and instructions dialog.
5. The user chooses to skip the instructions dialog.
6. The Use Case continues from position 6 of the main flow.

**Exceptional flow**

Unexpected exit of game

1. The user turns on the game
2. The system loads the user into the main menu.
3. The user selects exit game.
4. The system shuts game down.

**Termination**

The process is terminated when the controls are given to the player to start moving units.

**Post condition**

The system goes into a wait state until the user starts a new game.

**Requirement: Save Game**

**Description & Priority**

This functionality is essential as a user will need to be able to save their progress in a game and pick up from where they left off. Without this a player would need to complete the game in a single play through without turning the game off, which is an impractical and frustrating experience.

**Use Case**

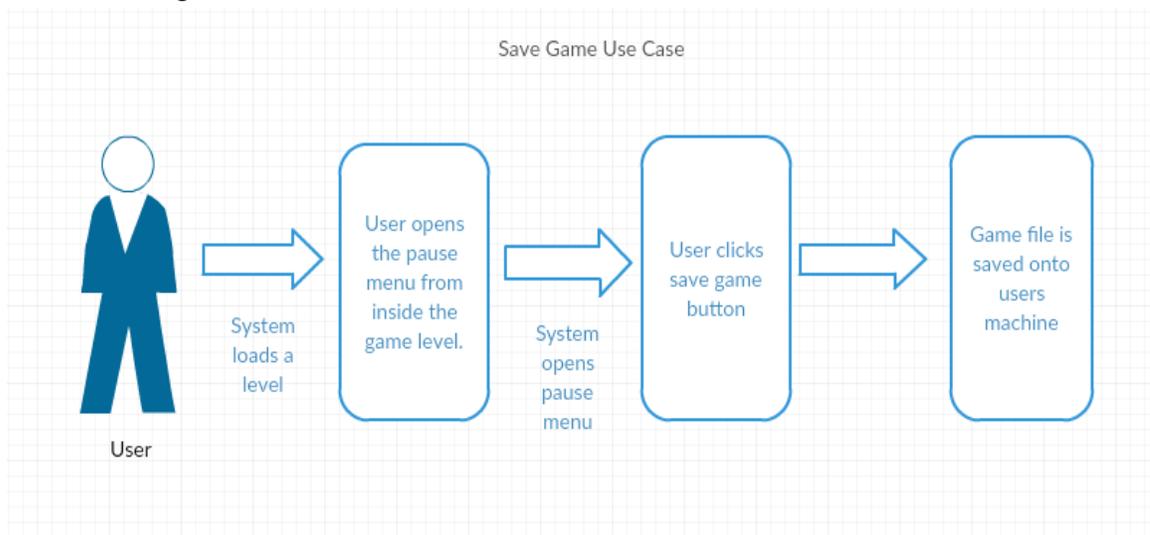
**Scope**

The scope of this use case is to allow the player to save their game progress.

**Description**

This use case describes the actions a player will take in order to save their game from the pause menu.

**Use Case Diagram**



**Flow Description**

**Precondition**

The game is installed on the users' device and they have a way of activating the game, IE desktop shortcut, game file etc.

The users' device must at least reach the minimum requirements needed to run the game, as stated earlier in the document.

The user will have to have started a new game to have progress to save and to currently be in a game level.

**Activation**

This use case starts when a user opens the pause menu while they are loaded into a level.

**Main flow**

1. The user opens starts the game.
2. The system loads the user into a game level
3. The user progresses in the game and opens the pause menu.
4. The system loads the pause menu.
5. User clicks to save game.
6. System saves the game file to the users' machine.

**Alternate flow**

(Not applicable)

**Exceptional flow**

(Not applicable)

**Termination**

The process is terminated when the game file has been saved to the users' device.

**Post condition**

The system goes into a wait state until the user needs to save their game again.

**Requirement: Load Game**

**Description & Priority**

This functionality adds a load feature to the main menu and allows the user to load their save game (from previous step) and continue from where they had left off in their last play-through.

**Use Case**

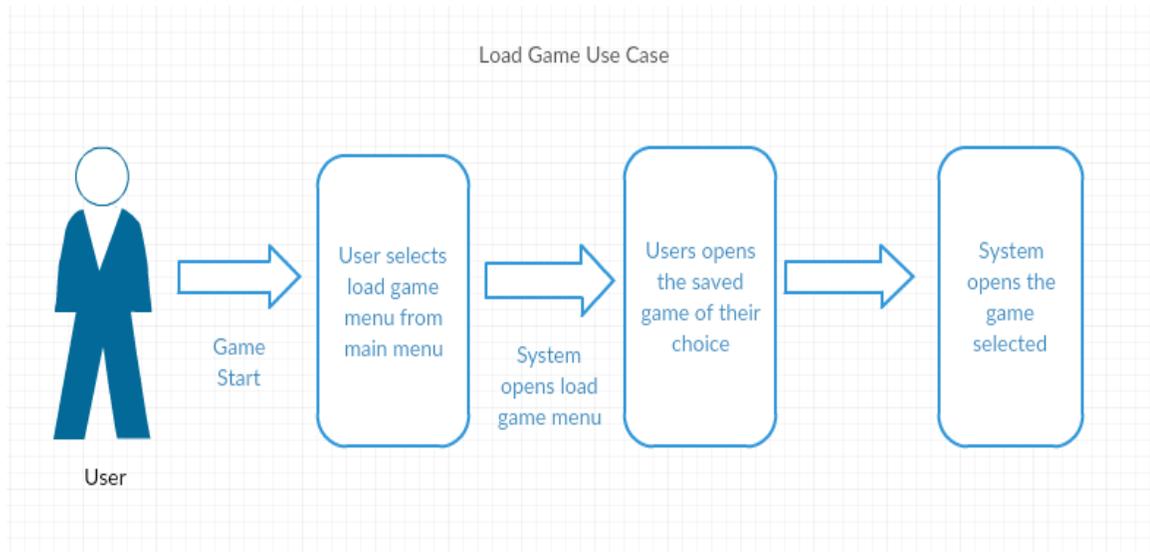
**Scope**

The scope of this use case is to allow the player to load their previous saved games.

**Description**

This use case describes the actions a player will take in order to start playing the game from where they had last saved it.

**Use Case Diagram**



### **Flow Description**

#### **Precondition**

The game is installed on the users' device and they have a way of activating the game, IE desktop shortcut, game file etc.

The users' device must at least reach the minimum requirements needed to run the game, as stated earlier in the document.

The user must have already started a game and have a save file on their device.

#### **Activation**

This use case starts when a user turns on the game selects their saved game from the load game menu.

#### **Main flow**

1. The user starts the game.
2. The system loads the user into the main menu.
3. The user selects the load game menu.
4. The system opens the load game menu.
5. User opens the saved game of their choice.
6. System launches the game back to the state when it was saved.
7. Player continues playing the game.

#### **Alternate flow**

Player loads the game back from in the game level.

1. The user is already playing the game and pauses the game.
2. The system loads the pause menu.
3. The user selects the load game menu from the pause menu.
4. The use case continues as of step 4 in the main flow.

#### **Exceptional flow**

User deletes their saved game.

1. The user turns on the game.
2. The system loads the user into the main menu.
3. The user selects the load menu from the main menu.
4. The system opens all saved games.
5. User selects the delete game option.

#### **Termination**

The process is terminated when the player has been successfully loaded into their game.

#### **Post condition**

The system goes into a wait state until the user needs to load another game.

## **Requirement: Exit Game**

### **Description & Priority**

This functionality allows the user to exit out of the game from any point, be that from the main menu or from the pause menu inside the game level which is an essential function for the game.

### **Use Case**

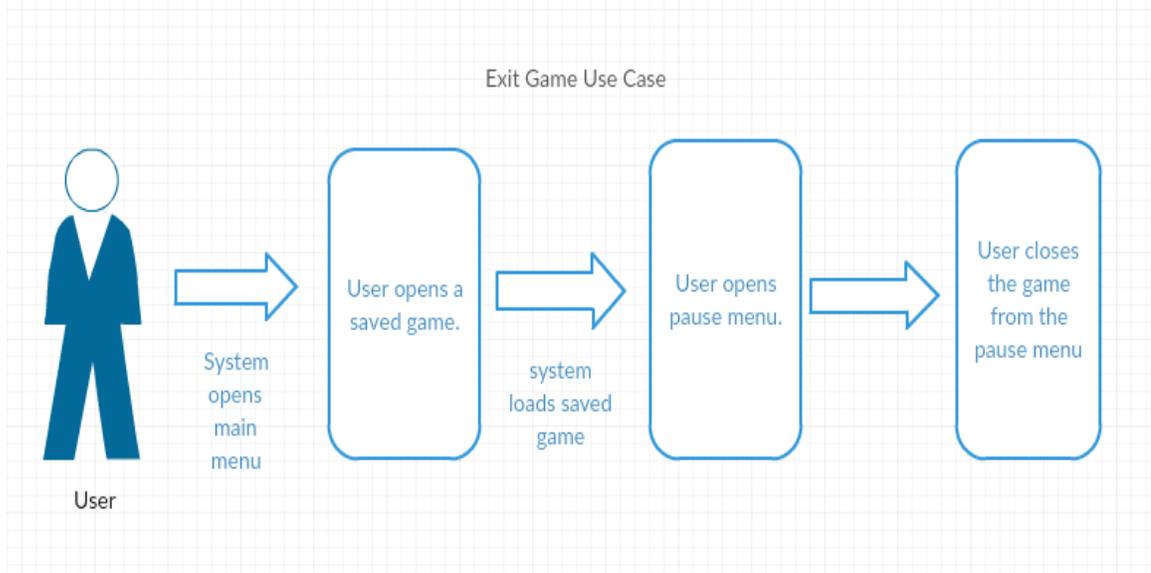
#### **Scope**

The scope of this use case is to allow the player to exit the game from any position.

### **Description**

This use case describes the actions a player will take in order to exit the game while they are actually loaded into a level.

### **Use Case Diagram**



### **Flow Description**

#### **Precondition**

The game is installed on the users' device and they have a way of activating the game, IE desktop shortcut, game file etc.

The users' device must at least reach the minimum requirements needed to run the game, as stated earlier in the document.

#### **Activation**

This use case starts when a user loads into a game and opens the pause menu to exit.

#### **Main flow**

1. The user opens a game, be that a saved or new game.
2. The system loads the user into the level.
3. The user opens the pause menu.
4. The system loads the pause menu for the user.
5. User selects the exit button.
6. System opens a dialog box to receive confirmation of user exit.
7. Player selects okay.
8. System closes the game down.

#### **Alternate flow**

Player exits from the main menu.

1. The user turns on the game.
2. The system loads the user into the main menu.
3. The use case picks up from step 5 in the main flow.

#### **Exceptional flow**

(Not Applicable)

#### **Termination**

The process is terminated when the game is closed down.

**Post condition**

The system has been closed down.

**Requirement: Change Game Settings**

**Description & Priority**

This functionality is essential as a user will need to be able to change the game settings for the machine that they are using, otherwise a user may not be able to play the game at its optimal potential.

**Use Case**

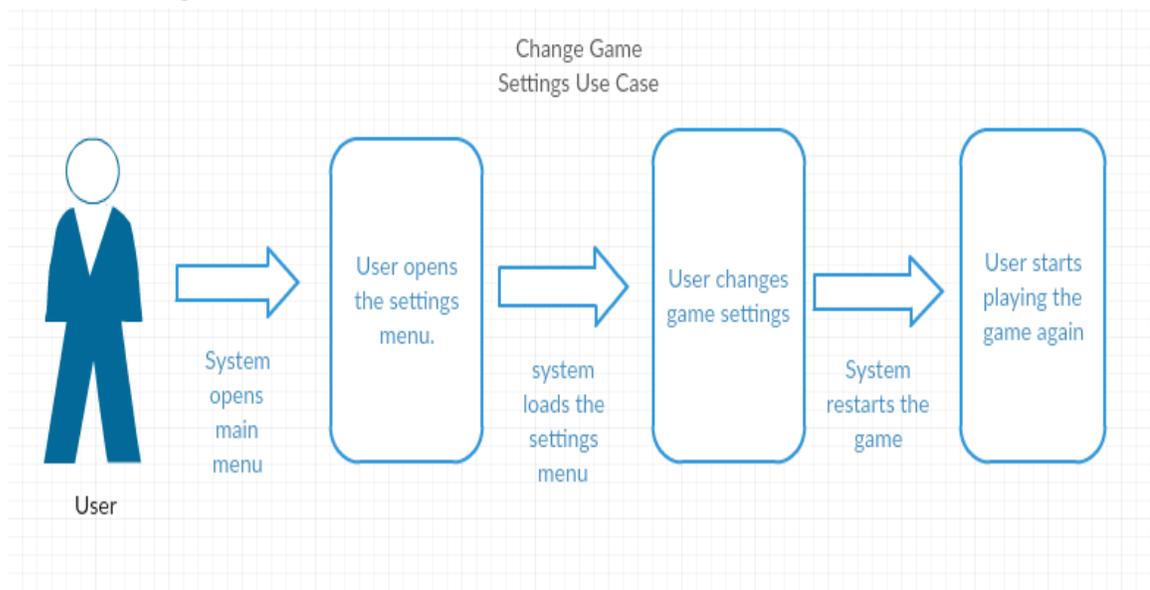
**Scope**

The scope of this use case is to allow the player to change certain settings in the game.

**Description**

This use case describes the actions a player will take in order to change the game settings and in this instance it will be the difficulty setting.

**Use Case Diagram**



**Flow Description**

**Precondition**

The game is installed on the users' device and they have a way of activating the game, IE desktop shortcut, game file etc.

The users' device must at least reach the minimum requirements needed to run the game, as stated earlier in the document.

## Activation

This use case starts when a user turns on the game and opens the change game settings menu.

## Main flow

1. The user turns on the game.
2. The system loads the user into the main menu.
3. The user selects the settings menu button.
4. The system loads the settings menu.
5. User selects to change difficulty between easy or hard.
6. System must restart to implement these changes and shuts down.
7. Player restarts game.

## Alternate flow

Player changes difficulty from in-game level.

1. The user turns on the game.
2. The system loads the user into the main menu.
3. The user selects a saved game from the load game menu.
4. The system loads the saved game.
5. The user opens the pause menu from in the game.
6. The system opens the pause menu.
7. The use case picks up from step 3 in the main flow.

## Exceptional flow

(Not Applicable)

## Termination

The process is terminated when the game is restarted with the new settings changes having been applied.

## Post condition

The system goes into a wait state until the user changes the settings.

## **Non-Functional Requirements**

### **Performance/Response time requirement**

The performance of a user's device depends on what game settings they meet. These settings will also be a factor on the response time for opening the game and loading in-game levels and actions.

### **Minimal Game Settings:**

- Minimum of 4GB of RAM
- Dual core intel or AMD processor
- NVIDIA GeForce 470 GTX graphics card or any card higher
- Windows 7, 8 or 10

### **Recommended Game Settings:**

- Ran on a desktop PC
- Quad-Core Intel or AMD processor
- NVIDIA GeForce 470 GTX graphics card or any card higher
- Windows 7 64 bit
- 8GB of RAM

### ***Availability requirement***

The game will be available through a digital distributor and there are a lot of options available to developers now. The most beneficial option that would suit the project would be steam. Although the service that Steam offers as a rating system for potential customers to leave feedback on projects that are uploaded to the platform called Steam Greenlight has recently been pulled. Steam will soon be debuting a new service to offer small developers the chance to get their games on the platform and tested. If a project receives enough feedback and recognition it can be added to the steam marketplace for sale. As of now the game is being stored on Amazon Web Services S3 service. The link to download the website is currently on the website:

<http://gamehost.gear.host/>

Which is specifically set up to distribute the game.

### ***Reliability requirement***

The reliability of the game is the simplest of its requirements as once the game file is downloaded all the user needs to do is launch the game and unreal engine takes over running the file.

### ***Maintainability requirement***

The steam community offers enough support to be able to maintain the game from the platform alone. The community includes a forum page where users can view FAQ's and post about bugs. The projects page on steam also allows you to inform users about up-coming updates and foreseeable downtime. Any updates needed for the game are added immediately to a user's account meaning they need to update before they can play again.

### ***Extendibility requirement***

The game offers enough reusability that it will be quite easy to extend. I will cover these aspects further in the reusability requirement but once the mechanics of the game are set in place the game can be extended by adding new a new story and chapters through DLC content.

### ***Reusability requirement***

Through using the Animation and Rigging Toolkit available for Maya and Unreal Engine it will be possible to re-use the skeletal structure of the models I create, meaning they can all cleanly share the same animations and properties without having to change any of the code for them. The skeleton being used to model my character is the same one being used with the default character in Unreal engine so you can drastically change the appearance of the character and it will still be able to use the default animations stored in Unreal engine.

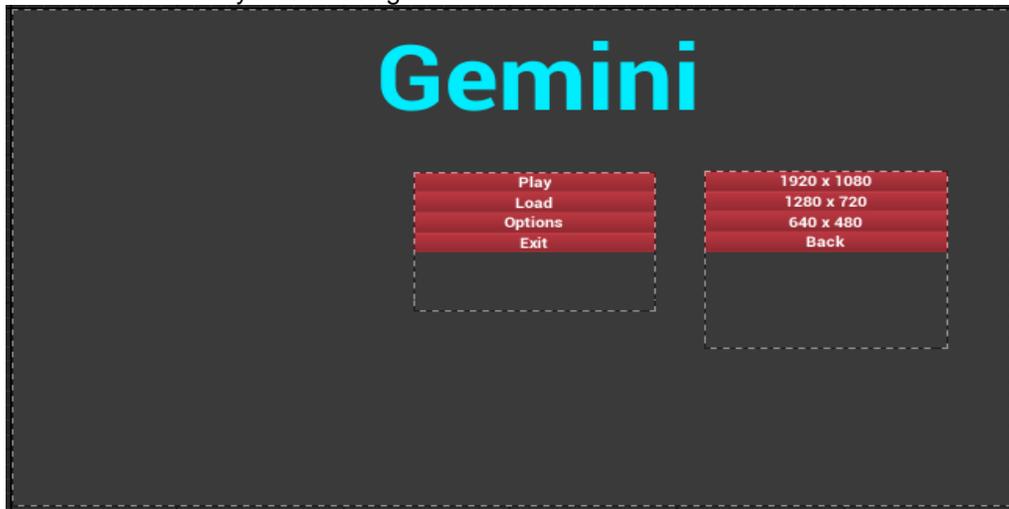
## GUI Implementation

This section of the report shows all the menus for the game, how they connect to one another and how a user would navigate through the game.

Some sections have been moved to clearly show each button for the purpose of giving a better description and explain how the user can interact with it. In game, a lot of the sections will be overlapping with each other and work on a system of changing the visibility of each sub menu.

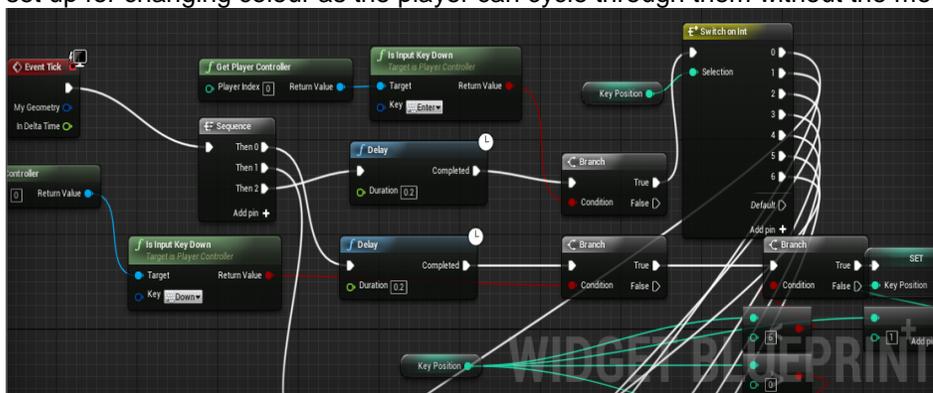
### Main Menu

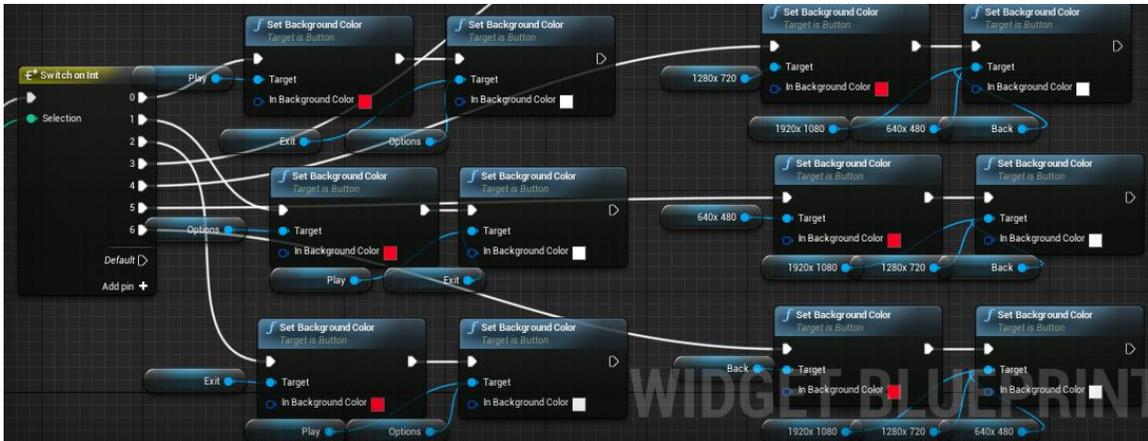
This menu contains the buttons used to navigate to all areas of the game and is the first thing a user sees when they launch the game.



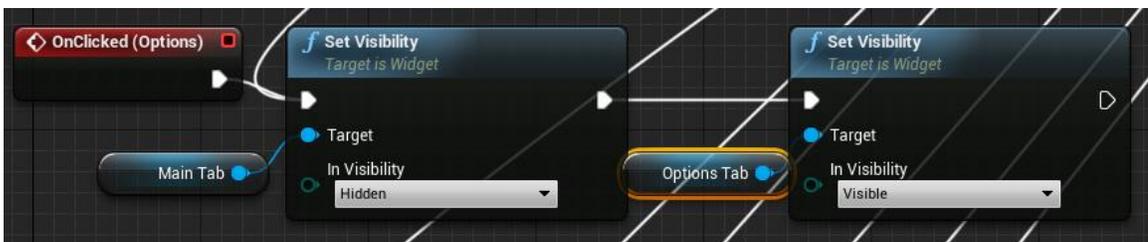
- Play allows users to start playing the game and will launch the tutorial level immediately.
- Load will allow users to start playing the game from where they last saved.
- Settings will give the player the opportunity to change how the game operates.
- Exit will allow users to close the game.
- All of the backgrounds will be of the same colour throughout all of the GUI's.

The images below show some of the functionality behind the main menu. The main menu is compatible with a controller. They will change colour as if they are being hovered over and navigate between each button just using buttons on the keyboard. The second image shows each button set up for changing colour as the player can cycle through them without the mouse.





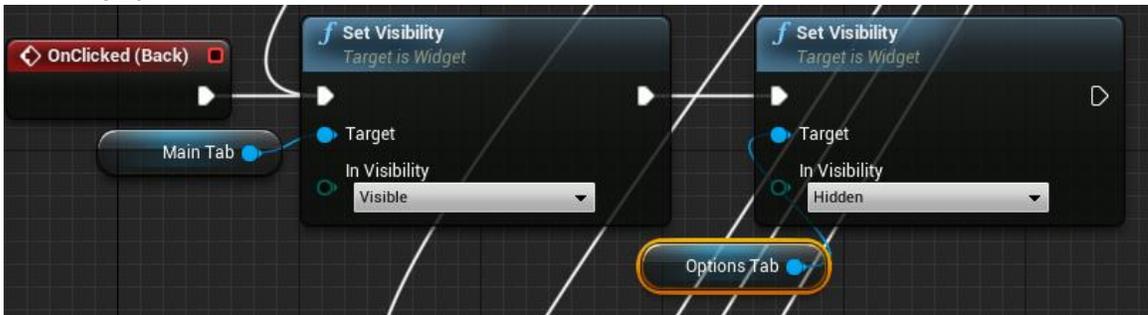
The image shows how the player would get onto the settings menu from the main menu and how some of the GUI's function. When the option button is selected, it will set the visibility for the Main Tab to hidden and the visibility for the Options Tab to visible. This allows for an easy to use design so users can switch effortlessly between each menu. The default visibility for the Options Tab is hidden so it must be selected to be viewed and it will take the place of where the Main Tab was on the menu.

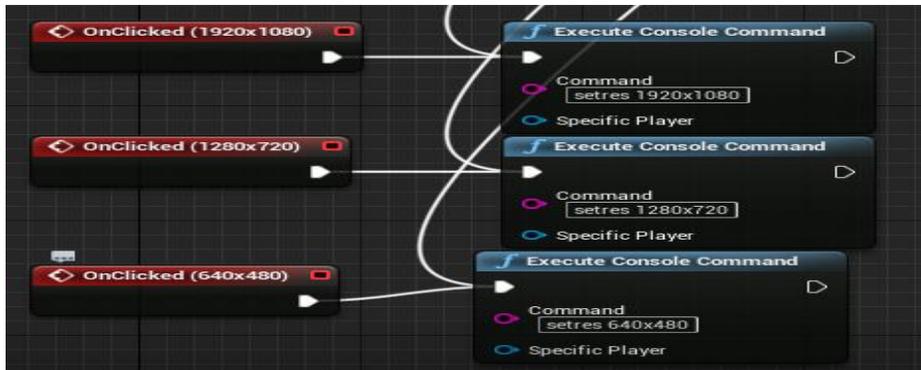


### Settings Menu

This section is a sub-menu connected to the main and pause menu and explain all the button options contained in it.

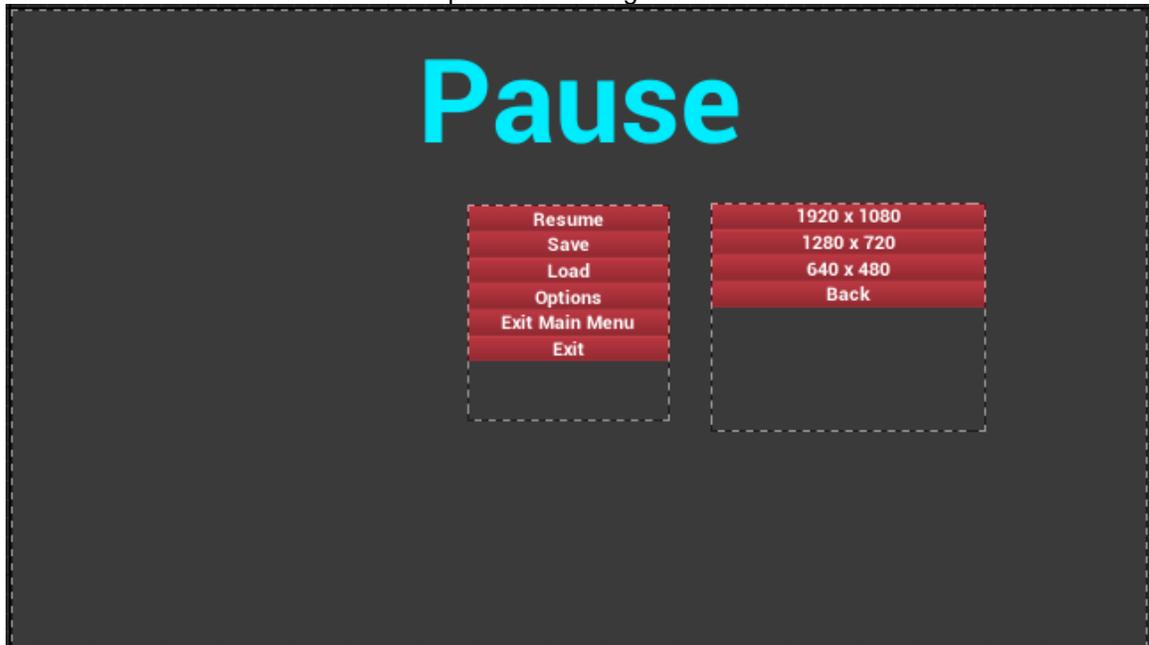
- The image below shows how to open the main menu from the settings sub-menu and works using the same visibility functions as the main menu but reverses the order so the Options Tab is set to hidden.
- The resolution options will execute a console command to set the screen to the specified size.



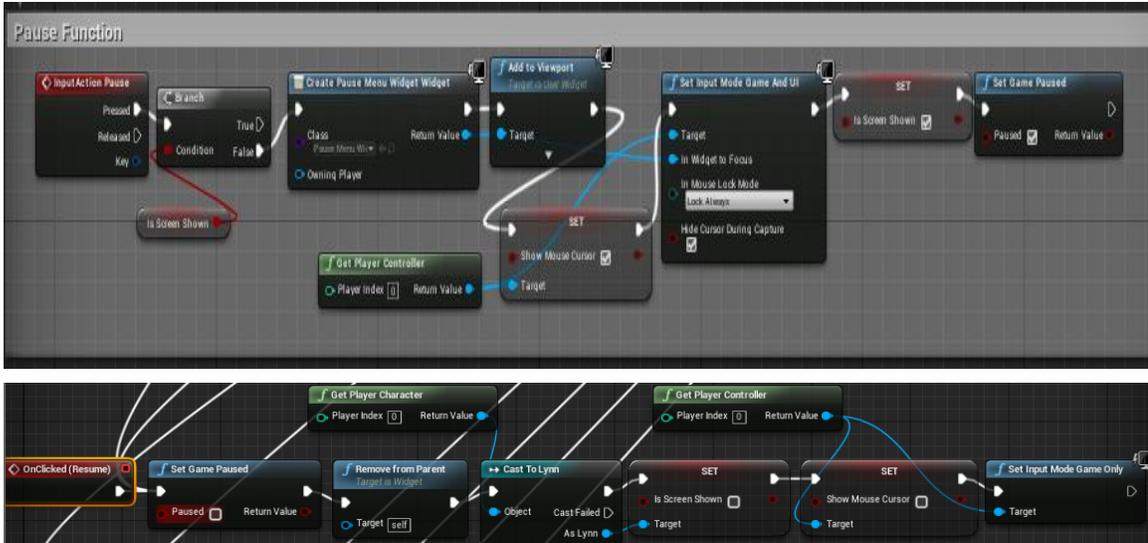


### **Pause Menu**

This section will display when if the user pauses the game. Its offers almost all of the same functionalities as the main menu except for the save game button.



- The user can select to resume the game which will remove the pause menu and re-join the game.
- Save, will allow the user to save their file to the machine.
- Load, will allow the user to play the game from where they last saved.
- Main menu will navigate the user back to the main menu.
- Exit will close the game but also prompt the user to save their game first.



When the user activates the InputAction Pause the system will check if the screen is being shown through the IsScreenShown Boolean variable. It will then create the pause menu widget and add it to the viewport as well as setting the mouse cursor to be able to select the buttons on screen. It will also set the game to pause mode so all of the background process are frozen.

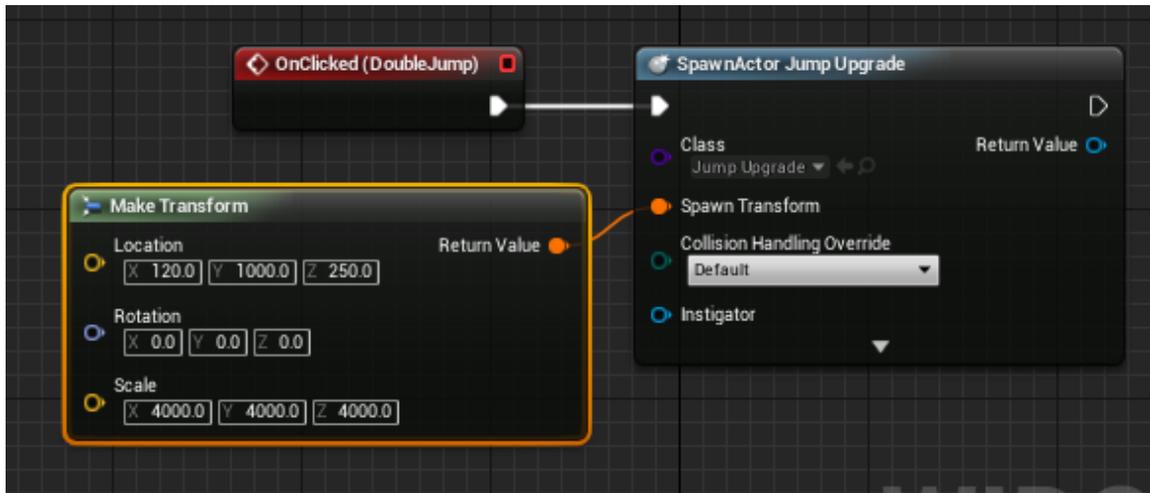
The resume function will essentially reverse the functions of the pause event. It will set pause to false, resuming the background functions, remove the menu from the viewport and give control back over to the player controller so the player can move again.

### Upgrade Menu

The section below shows the Character Upgrades menu. In this menu, the player can exchange relic points for upgrades under any of the three headings, conditioning, survival or assault. For the purpose of the tutorial level the only button that the user can select is the double jump ability. It also has the same pause game properties as the pause game menu. The upgrade menu is an input action called by pressing the B key.



When the double jump button is selected, it will spawn the upgrade double jump item into the world. The scale of the object has been increased to a size where it will automatically collide with the player and the double jump upgrade custom event will be activated and the item destroyed in the process.



### **Game Over Menu**

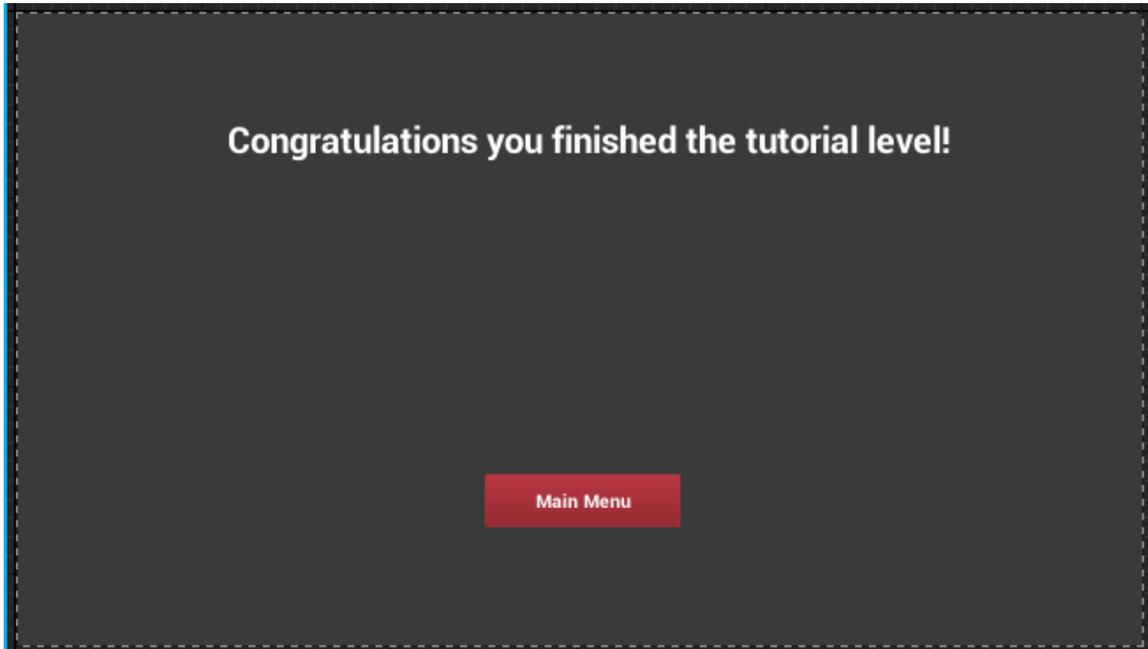
This section will display when the user's health points drop to 0 and they are killed.



- Exit to main menu will end the game and return the user to the main menu.
- Restart will start the entire level again although the user can reload to a previous save file through the pause menu once they have loaded back into the level.

### ***End Game Menu***

This section will display when the player makes it to the end of the tutorial level and stands on the blue light to finish the section. It offers only one function and that is to return the user back to the main menu.



### ***System Evolution***

The plan is to build a fully functioning game on top of the foundation I have made for the project. The obvious addition being to add more levels onto the game after the initial tutorial/prologue level. This should be an easy task to achieve as once the mechanics of the game have been established in the first level all there is left to do is create more content. The goal is to tell the story to its fullest extent while fitting it into 20-25 chapters/levels. Depending on the success of the project, it would be easy to add more DLC content to the game, this would include addition characters and chapters to add to the story and give the game more depth. An interesting area to develop would be to introduce a campaign allowing you to play as the antagonist of the story and control the enemy units with new abilities and mechanics. The highlight of evolving the game would be to add a multilayer mode, allowing users to pick the units of their choice and play against one another which will add a whole new level of re-playability.

## **Testing Overview**

The following section will analyse the testing results sent out for the game. The game was tested by sending out the link to download the game from:

<http://gamehost.gear.host/>

And also a link to a survey on Survey Monkey so player could leave feedback on how they felt about the game. The link is as follows:

<https://www.surveymonkey.com/r/XJZPW5G>

**1. Was it possible to complete the game by making it all the way to the end platform? If not please specify what point you stopped.**

All users made it to the end of the level although some found it difficult when they had to upgrade to the double jump to be able to proceed further and some found the jump itself difficult to reach.

**2. Was the user interface easy to use/user friendly? If no please specify how you would want it improved.**

Most users found the user interface very easy to use but some stated this was because the design was very limited. Many would have found the game better if the game had some type of style to fit its genre and not just grey backgrounds.

**3. What difficulty did you find the tutorial level?**

Most users found the tutorial easy to complete once they had clearly read through the instructions widgets.

**4. Would you buy a game with a female lead character?**

Although result would conclude that games with a female lead character don't sell as well as their male counterparts many users stated that it was the content of the game that is important.

**5. What age group do you fall under?**

All the users who tested the game fell between the age groups of 18-22 and 23-27.

**6. Most enjoyable aspect of the game?**

All users agreed in that the combat was the best aspect of the game.

**7. Was there any part of the game you would like to change?**

The design of the user interface was the main concern that users wanted changed and many of them also stated that more content e.g. another level would have improved their experience.

**8. What aspect of a shooting game do you think is most important?**

All users agreed that the story is important in a shooting game, otherwise it gets boring shooting enemies for no reason but the most important aspect was the combat itself.

**9. While playing the tutorial level did you encounter any glitches in the map? If yes please specify which section.**

In the final testing of the game there were no glitches reported in the map and users could complete the level to the end. Although the double jump section to get to area four was reported as a glitch by one user as it was too difficult to make.

**10. How did you find using the controls of the game?**

The users found the controls of the game easy to use as they are essentially the same as any shooting game currently on the market. This was the desired result as it was important to create a game users would be accustomed to.

## **Appendix**

### **Project Proposal**

#### **Objectives**

The aim of my project is to create a fun, addictive and visually striking first person shooter. The game will have a multiplayer function in it and it is important that it is fun, competitive and to have a unique aspect that sets it apart from other first person shooter multiplayer games. The visuals and having unique character models are an important part of creating the game for me and because of this I will create a separate list below for just the design components that I plan to use.

#### **The Functionality Objectives of my project are:**

- To create a first person shooter game.
- To create a tutorial level to introduce new players to the basic game concepts.
- The functionality for the game to accept either controller or keyboard/mouse input.
- Create steam achievements and/or mission objectives so levels will have an element of re-playability.
- Include a scoreboard to track points collected and time needed to complete each level.
- Create a complete user interface system including pause, death and main menu screens for players to navigate the game and a HUD to display information to the player during gameplay.
- Create Collectible objects and hide them in the game map, players can then use them to upgrade their character.
- Create an experience system for players to level up their weapons by destroying enemies.
- Create a settings menu for players to change certain aspects to personalize their game experience.
- Create a life system which will have the player restart the section when depleted.
- Include a save function so players don't lose their progress when they turn the game off.
- Include a load function to resume where they left off.

#### **The Design Objectives of my project are:**

- Creating a storyline that will capture players' interest.
- Create 3d models for my main characters using Maya.
- Create 3d models for my enemies using Maya
- Create a vibrant visual style and unique theme for the game.

#### **Background**

In this section I will give a brief description of the background behind the project and explain a lot of the general information needed to understand what I am planning on doing. For my project I have chosen the genre of a first person shooter for the simple reason that these are incredibly popular games in today's market and going in with a blank template gave me the chance to create something innovative and different. I feel like it would be the best approach to telling the story I want to get across and the games themselves are a memorable experience for anyone who plays them. The game will be more like an adventure game than anything else, platform jumping, quirky enemies and different gameplay features as opposed to just your everyday shooter. Games like this combined with a good story have proven to be popular on the market and sell to gamers of all ages and backgrounds.

My goal is to make the games multiplayer as fun as possible to give players a great experience while also introducing a competitive nature which will have players wanting to play more and more. On the other side, in the games campaign it will be more difficult, making players believe that their progress actually means something. This will be possible by implementing a life system that will have players restart the level, section or game when depleted The basic objective of the game is

to make it to the end of the level, having to clear obstacles and destroy enemies along the way. A game like this will thrive off giving users more abilities\powers to play around with, this gives the game depth and without it the game is just a simple shooter like the hundreds of other ones they have most likely played. It is important for people to feel like the game is not so easy that they can complete it without fear of being beaten and I feel a life system is the perfect way to enforce this. This system I hope will add tension as players will be under pressure to complete

The game will be set with a vibrant theme, much like platformer and my target audience will also be aimed at the type of people who would also buy these games. This is mainly aimed at a younger audience but has also proven to be bought by a lot of an older generation as well. The first person shooter genre spans across many different types of game and has proven to be a massively popular area in the games market. It is important to get a multiplayer system working in the game and to get as many different game modes as possible so a player has a choice between what to play. A competitive edge and fun, unique game play is what will have players coming back for more which downloadable content now a huge part of games, in the future it would be possible to add additional characters, game modes and maps to the games multiplayer.

Players will have the ability to level up their character over the course of the campaign, growing with them and giving them more powers as they proceed from challenge to challenge. They could also have the option to level up abilities, weapons and other modifications through the action of finding collectibles making it worthwhile to explore each level.

#### ***Technical Approach and Details***

I plan on creating a solid foundation for the game to be built on, doing thorough research on Pluralsight and watching other tutorials on creating a game on unreal engine. Research and the correct approach to the project will be vital to its success as I have never created a project of this size or done a lot of work on Unreal engine. I will use a blueprint structure for programming in unreal engine and this is the standard practice used for it.

#### ***Special resources required***

I will be using unreal engine to create my game and from the research I have conducted there is an Animation and Rigging Toolkit on the marketplace for free which can be used in conjunction with Autodesk Maya to create character models and animations that use the same skeletal structure as unreal engine models, meaning they can share the same assets and animations.

#### ***Evaluation***

I will of course be continuously testing the game on my own laptop as I work through the project, correcting the flaws as I progress. Although I would like to have the game tested by other gamers.

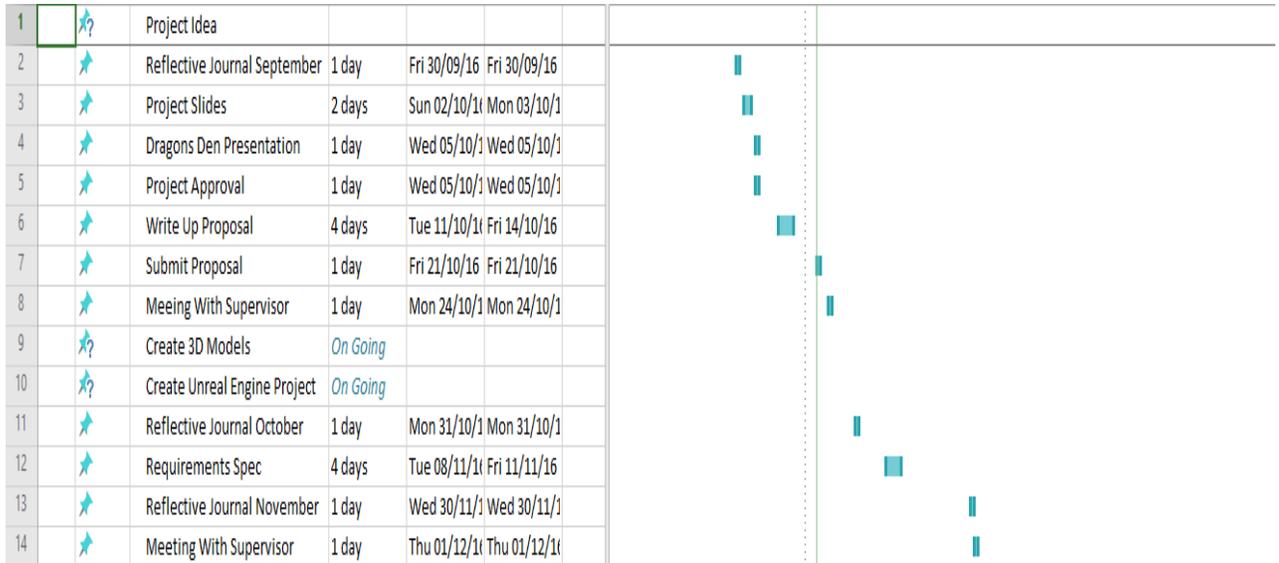
I also plan to use the following testing techniques:

**Test Environment:** To test the project so I can see if minor functionalities work before implementing them into the main project itself.

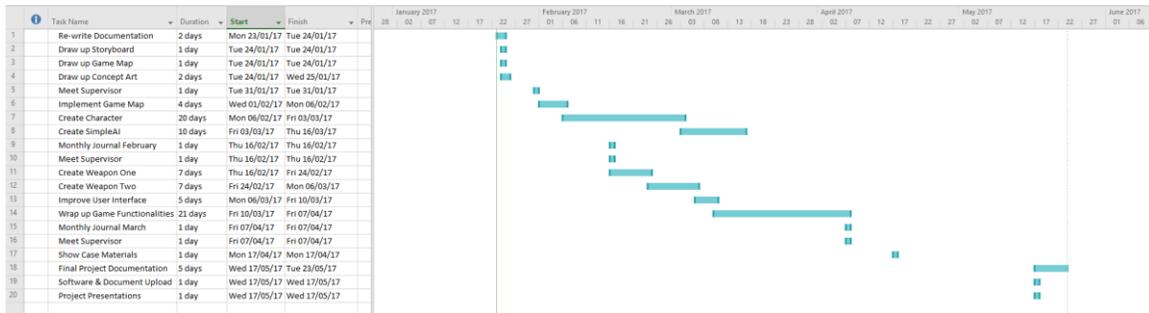
**Simulation Environment:** From what I have gathered unreal engine 4 also has a simulation mode that can be used to check changes as if the game were to actually run.

#### ***Project Plan***

As seen below is a Gantt chart depicting my project plan over the coming months:



Since changing the project after the mid-point presentation, below is the revised project plan from the months of January to the end of the project in May:



## Monthly Journals

Below are the monthly reflections for the first semester:

### Month: September

#### My Achievements

This month, I was able to utilize the time given to us for independent research to complete the documents that we would need for next month. It was also a great opportunity to study the previous year's final projects to see what was expected of us, the amount of work that would be required and extent of what we would be able to make in the time given to us.

My contributions to the projects included the PowerPoint slides needed for the dragon's den pitch and a lot of research into other projects and what game engine we could be using to run our projects.

#### My Reflection

At the beginning of this month I felt it difficult to get a full understanding of the project. Although I had come up with an idea for my game that I was planning to pitch next month, at this stage without having in-depth knowledge of our other modules I found it hard to realize what we would be capable of creating. This was helped by looking over projects from the year before us.

#### Intended Changes

Next month, I will try to get a realistic scope of what I will be doing for the project. Based on whether my idea is approved I will write up a project proposal and hopefully begin working on a basic foundation that I will build my project idea on.

I realized that I need to use all of the time given to us for independent research and set goals to manage my progress throughout the project. The project will require more time and effort than any task that we have been set before but having more of an interest in the stream I have chosen I feel will be a crucial difference then in my previous three years' work.

#### Supervisor Meetings

(Supervisor not assigned)

### **Month: October**

#### My Achievements

This month, I was able to get my original project idea approved and a head start on creating some of the 3d models I will be using later in the project. This tied in well with my computer graphics module as for our first CA we were to specialize in a chosen area (3d printing) and print off a model of our own creation.

On top of this I began taking beginners tutorials in unreal engine to get an understanding how to build an environment and how to move characters around the game area.

My contributions to the projects included the 3d model on the characters I will be using, a basic project created on unreal engine and the project proposal document.

#### My Reflection

I felt, it worked well to do my own research into what I think would work well for the project and get a good foundation of understanding in the systems that I plan on using. I can't rely on being taught everything I need to know in class and will need to use my free time fill in the blanks in what I don't understand in my project.

#### Intended Changes

Next month, I will try to research and gain an understanding of how animation works so I can start moving the models I am designing.

I realized that I need to put aside a lot more time into using unreal engine as for my computer graphics module we have chosen to focus our work on Unity. This means that anything I need to know about unreal engine I will have to research on my own.

#### Supervisor Meetings

I was unable to attend the first of our group supervisor meetings but have sent my project proposal to my supervisor, Frances Sheridan.

### **Month: November**

#### My Achievements

This month, I was able to go further with my research into unreal engine. A CA in computer graphics asks us to create a mini game on a game engine of our choosing, this was the perfect opportunity to get an understanding of unreal engine. Although not all of the aspects of the mini game are directly related to my software project idea it allowed me to experiment and test functionalities that I could implement into it later.

My contributions to the projects included completing the Technical report in advance for next month.

#### My Reflection

I felt it worked well to complete small parts of the project whenever I got a chance. Tutorials on Pluralsight have given me a great start on what I can put into the basic foundation of my game and with more advanced lessons all I need is more time to create something really impressive.

#### Intended Changes

Next month, I intend on getting as much work done for the project as possible in whatever time I can get so as to meet the deadlines in December. With the Technical report completed I will focus on getting a prototype prepared for the midpoint presentation but other than this I can see work on the project slowing down until next semester. This is unavoidable as the month of December has four other deadlines for various other projects and we are to sit our exams in January.

#### Supervisor Meetings

I have yet to set up a meeting with my supervisor but have been in email contact to get feedback on the reports needed to be uploaded. I plan to set up a meeting for December to set out a plan for the project over the holidays.

### **Month: December**

#### My Achievements

This month, progress on the project was incredibly slow due to deadlines and projects in my other modules which has taken almost all of my attention. Achievements were minimal for the entire month and were mainly aimed at how I am going to increase work over the Christmas break although our deadlines lead right up until the 23<sup>rd</sup>. The most outstanding contribution of the month was the mid-point presentation, a great opportunity to receive feedback on the work I have created so far and how I am to improve it.

#### My Reflection

December has been a difficult month to complete work for the project with the workload for the semester piling up towards the end of the semester. After my mid-point presentation, I am now having serious reservations about the direction my project is going in and I will have to revise what I am going to do about this over the holidays and possibly change my project idea. This would mean having to re-make all of my documentation but that is only a drop in the pond when compared to the difficulty of the original project I had proposed.

#### Intended Changes

Next month, I intend on getting a lot of work done for the project in the time before my exams although much of my time will then be taken up by studying for these exams. These changes may even include changing my entire project idea.

#### Supervisor Meetings

The mid-point presentation has been the first contact I have made with my project supervisor this semester other than emailing Frances for assistance and feedback on documentation. Although we have tried meeting in the last week before my presentation we could not get a date where we were both free.

### **Month: January**

#### My Achievements

This month, my progress on the project has improved substantially over the course of our time off. Since the start of the year I have decided to change my project idea entirely and I have achieved more in the month of January than I did in the previous semester. I have re-wrote my project proposal and all other documentation. My first plan was to create an entire user interface which I was created in the first week and have now started created a model as well as animations on Maya using the unreal engine plugin, Animation and Rigging Toolkit.

#### My Reflection

January has been the most productive month of the entire project and now that I have changed my project idea to something more achievable I have made huge steps to obtaining that goal. There is a lot of work to catch up on now but with the progress I have made in this short time since starting again I feel it will be possible to create something even better than my original project proposal.

With the documentation out of the way I can now proceed with coding on the project and add some real functionality to the game.

#### Intended Changes

Next month, I plan to carry on with the progress I have made and create my own character model complete with animations and adding my own weapons instead of the Default ones. Unreal engine provides all of its own models but I feel it is an important aspect, and in my eyes one of the most difficult, to add my own to the project.

#### Supervisor Meetings

First meeting with Frances went well. We covered, Changing the project, creating a storyboard, re-writing my documents and how it is essential to meet more often this semester.

### **Month: February**

#### My Achievements

This month, my progress on the project has slowed down since I restarted it last month due to being back in college and having to keep up with other course work. For the month, I have made slight changes to the project including, a leader board, creating working enemy AI and started texturing the game map. I also spent much of the month writing out the analysis report and this has given me insight into what direction I want to go with the game.

#### My Reflection

I have slowed down in my progress in the project but have spent the month completing other course work. I am now in a position where I am ahead of my current course work and plan on using this to catch up on work next month in March. I did not manage to start modelling my character's weapons and this may be something I will need to commit more time too, possibly after I finish my exams in April. Since returning to college I have had more guidance in the project and have set out a lot more goals for the month of march.

#### Intended Changes

Next month, for my cloud gaming module we must set up an Amazon web services account and start the process of launching an online game feature, this is something I plan to use with my project to launch an online leader board. I plan to complete the game map and have it fully texture and create a more complex AI enemy that will have the ability to attack/shoot at the player.

#### Supervisor Meetings

I've started meeting with Frances more often and am now setting goals based on how she advises me. In our meeting this month she advised me to continue with some work on the analysis report which is now nearly complete and make the leader board and map.

### **Month: March**

#### My Achievements

This month, my progress on the project has been going well. Coming to the end of the project it is clear what will be possible to include and what won't, this made it easy to set out my goals for the month of March. My achievements include creating my own character that is compatible with Unreal engine and even creating my own animations for the character which is an aspect I never planned on implementing into the game, this means not using the default animations provided by UE4 but instead creating something more satisfying.

### My Reflection

Regarding my intended changes from last month I have managed to get the game on Amazon Web Services and hosted on Gearhost where people can download the game from. As for creating the AI and finishing the map this still needs to be completed and will be included into the goals for this month. As for modelling a weapon for my character this has been completed earlier than I expected and I plan to create more.

### Intended Changes

Next month is essentially the last full month of the project so by the end of it I plan to have my project completed. The following list is what still needs to be completed in the project:

- My character with clothes and weapons.
- Both enemy AI and creating their models and animations.
- Texturing the game map as well as populating it with enemies and props.
- The menu and upgrade system.
- The project documentation.

### Supervisor Meetings

Meetings with Frances have been keeping the project on track and we have been setting out goals for the project every 2 weeks. The main aspect she has emphasized completing is to come up with a story.

## ***Bibliography***

Below are the references for the project. The UE4 Answer Hub was used for many queries throughout the course of the project and the YouTube account Virtus Learning Hub provided more tutorial series than the one provided below.

UE4 Answer Hub. **(2014, October 11<sup>th</sup>)**. Blueprint Scripting. Forum Post. Retrieved from <https://answers.unrealengine.com/questions/111935/how-to-make-an-actor-deal-damage-and-have-another.html>

Youtube. **(2014, April 5<sup>th</sup>)** Series Introduction - #1 Unreal Engine 4 Beginner Tutorial Series. [Tutorial Series]. Retrieved from <https://www.youtube.com/watch?v=1M3S3eiJK5I&list=PLL0cLF8gjBpgDdMoeid6VI5roMI6xJQGC>