

Declaration Cover Sheet for Project Submission

SECTION 1

Name: Kyle Mc Galey
Student ID: X12431598
Supervisor: Padraig DeBurca

SECTION 2 Confirmation of Authorship

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

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

Signature: _____ Date: _____

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

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

What constitutes plagiarism or cheating?

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

Penalties for Plagiarism

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. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- that a student's marks shall be reduced
- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and
- the requirement that a student to attend additional or other lectures or courses or undertake additional academic work.

National College of Ireland
BSc in Computing
2015/2016

Kyle Mc Galey
x12431598

x12431598@student.ncirl.ie

kylem5624@gmail.com

abvss

Technical Report



National
College *of*
Ireland

Contents

1	Introduction.....	7
1.1	Background.....	7
1.2	Aims.....	9
1.3	Technologies	9
1.4	Game Analysis.....	11
2	System.....	14
2.1	Requirements	14
2.1.1	Functional requirements.....	14
2.1.2	User Requirements	14
2.1.3	Usability Requirements	15
2.2	Design and Architecture.....	16
2.2.1	Use Case Diagram.....	16
2.2.2	Requirement 1 <Main Menu>.....	17
2.2.3	Requirement 2 <Play Game>.....	19
2.2.4	Requirement 3 <Collect Items>.....	21
2.2.5	Requirement 4 <Launch Game>.....	23
2.2.6	Requirement 5 <Pause Game>.....	25
2.3	Graphical User Interface (GUI) Layout.....	27
2.4	Testing.....	29
2.4.1	Usability Testing.....	29
2.4.2	Unit testing	29
2.4.3	Customer testing	30
3	Non-Functional Requirements	32
3.1	Performance/Response time requirement	32
3.2	Availability requirement.....	32
3.3	Security requirement.....	32
3.4	Portability requirement.....	32
3.5	Data requirements	33
3.6	User requirements	33
3.7	Environmental requirements	33
3.8	Usability requirements	33

3.9	Extendibility requirement.....	33
4	<i>Implementation</i>	34
4.1	Main menu	34
4.2	Input Recognition.....	35
4.3	Player Movement.....	36
4.4	Jumping	37
4.5	Characters Flashlight.....	38
4.6	Health HUD and player Health.....	40
4.7	Pause Screen	42
4.8	Gem Pickups	42
4.9	NPC (Non Playable Character) Enemy.....	44
4.10	AI Aspect of the game.....	46
	4.10.1 <i>How it works?</i>	46
	4.10.2 <i>Unreal Engine Behaviour Trees.</i>	50
	4.10.3 <i>The Technical Architecture</i>	51
5	<i>User Manual</i>	52
5.1	Control scheme.....	52
6	<i>Appendix</i>	53
6.1	Project Proposal	53
	6.1.1 <i>Objectives</i>	55
	6.1.2 <i>Background</i>	56
	6.1.3 <i>Technical Approach</i>	57
	6.1.4 <i>Project Plan</i>	57
	6.1.5 <i>Technical Details</i>	57
	6.1.6 <i>Evaluation</i>	58
6.2	Monthly Journals.....	59

Executive Summary

Abyss is an action adventure game in its essence. It has graphical and lighting qualities that can rival most games available on the market today. It inspires to combine the concepts of an indie game with that of a large scale title such as the Elder Scrolls franchise. Although the core gameplay is more simplistic than a lot of games out there now, which can be considered to be a risk or a strength depending on how it's viewed I feel that if a game is immersive it can achieve great things. Visually this game should look amazing, that alone is reason enough to obtain the interest of gamers.

Due to this fact Abyss is a game that originally would not generate money. It would be a massive achievement for Abyss if it was Green Lighted on Steam (context below). The game would be available to download for free with an option to donate to help fund development. This has launched hundreds of games on steam and the service is an excellent way to get a foot in the door of the game development business. This is especially beneficial to first time developers as they will always have a reference and feedback from users which in a sense is free testing and market analysis.

As the project is a one-man development team it is almost impossible to create a game that can justify a heavy price tag so this will reflect in the game in comparison to a game that may cost 30-70 euro. Within the timeframe to get a game that looks intriguing and has a compelling atmosphere combined with a non-rage inducing control and gameplay system would be considered a massive success for the development and Abyss itself.

1 Introduction

1.1 Background

Gaming as an industry is growing rapidly every year, gaming companies are making immense amounts of money from their IP's and it is truly an inspiring industry where anyone with an idea and some determination can achieve success

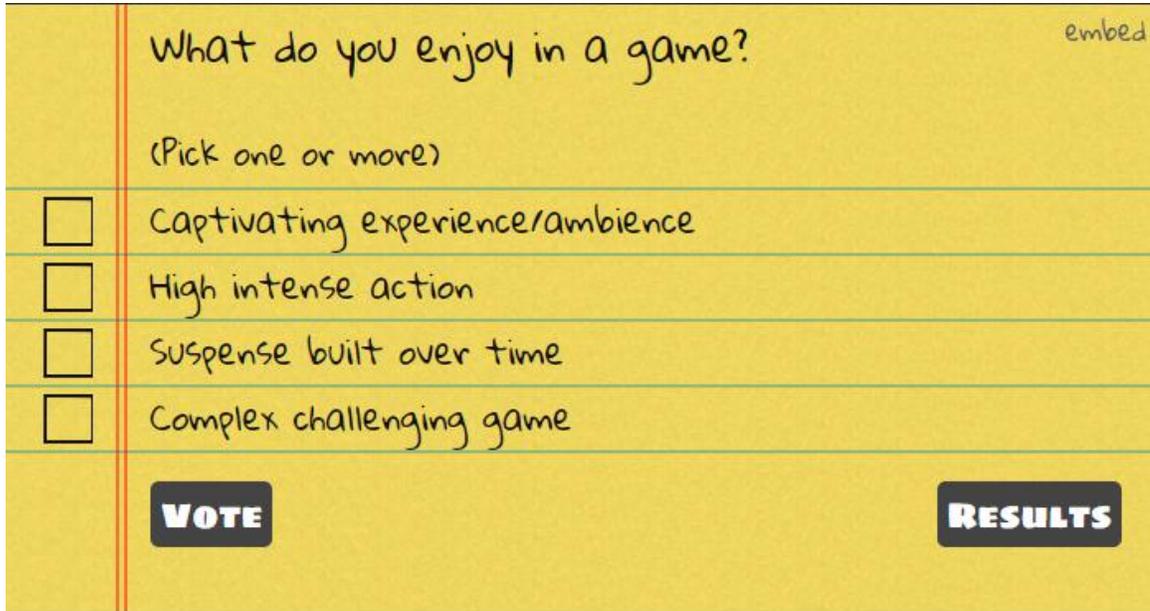
I completed some research on which game genres are the most attractive to users and what I can achieve in the time period. According to 3 leading industry websites – GameFaq, TheTopTens and GamersDecide the Action Adventure/RPG genre is consistently location in the top 3 in genre lists.

The genre of action/adventure is very common in all forms of media, from movies like Indiana jones, Lara croft to games such as Skyrim, Tomb raider and uncharted. They all have one thing in common, excitement.

Skyrim is an action role playing game that is open world. There are countless areas for the player to explore which include caves, ruins cities and forests. It is an incredibly successful game. Not only due to the vast story line but because of the sheer beauty of the scenery, it's a whole different world. What I love about this game is there is so many areas to explore and find different enemies and loot, countless hours can be spend roaming the map without touching a single main quest.

Tomb Raider is an action-adventure game, it's an extremely successful franchise about Lara Croft an explorer, and she discovers that her father left behind a trail of legends and goes on countless adventures throughout different parts of the world exploring caves, tombs and the sea. The game includes conquering enemies and solving puzzles and also dodging traps.

As well as picking a genre I had to decide on a path that the project must take in terms of style of game play. I created a small survey online that I distributed between friends and colleagues asking them a simple question.



The image shows a screenshot of a yellow-themed online survey. At the top right, the word "embed" is written in a small font. The main question is "What do you enjoy in a game?". Below the question, it says "(Pick one or more)". There are four options, each with a checkbox: "Captivating experience/ambience", "High intense action", "Suspense built over time", and "Complex challenging game". At the bottom left, there is a dark blue button with the word "VOTE" in white. At the bottom right, there is a dark blue button with the word "RESULTS" in white.

The results showed that there is no one best game style. There was votes on all across the board, personally I enjoy a captivating ambience and some suspense. They did hold a nice portion of the votes so I felt my gut choice had been partially justified.

1.2 Aims

The aim of Abyss is to create an immersive action cave exploring game. The player will have to overcome obstacles that face them and gather the loot around the cave.

1.3 Technologies

Unreal Engine 4

A relatively new engine that has only been made public to everyone for free since March 2015, the game engine I choose to use is Unreal engine 4. This engine has great potential in both diversity and graphics and easy gameplay design. Because it uses visual coding it gives a lot clearer insight on how a game would work and process and I find this invaluable. One of the main things I want from my game is to have really good graphics and I found unreal would be best to achieve this.

With blueprints visual coding aspect it's easy to get to grips on what needs to be in a game such as UI widgets, basic game mechanics. Visual scripting code is becoming an industry standard for game design and since I want to pursue a career here I thought it would be beneficial to incorporate it into my project.

Persona Animation is used to make changes to Unreal character skeletons, character meshes, skeleton sockets and animation classes just to name a few. This tool allows you to view animation sequences and merge targets as well as set up blend spaces for animations. You can also use this to alter in game physics and other properties such as collision detection for actors by using Unreal's physics asset editing tool. This makes importing external characters and animations much easier to do and allows for transfer of characters between projects.

Also due to the huge popularity boost of Unreal Engine 4 there have been more and more learning tools and tutorials being developed and this also highlighted my decision to use it.

The unreal engine marketplace also has countless paid and free assets to be made use of from blueprints to materials it is a great source.

Blender

Blender is a free open source 3D creation suite, blender has the potential to model, rig, animate, simulation and rendering. And because its open source there are countless online tutorials and aids that can be made use of it difficulty is struck. I used blender to make assets that I used in my project.

Mixamo Fuse

Fuse is an Adobe software that is relatively new in terms of its completeness. When starting this project Mixamo Fuse had been up and running for roughly a year and a half. It is a software that aids in the creation of custom 3D character models, using Fuse I was able to create highly detailed character models that suit my game to a tee. The software allows for basic modeling, giving you a shell pretty much. I the applied a skeleton to the model character shell. This is done by a process called rigging. Rigging is basically attaching a skeletal mesh to a character model, so for instance to add hand gestures to a character, the skeleton of the character must have a well-defined skeletal structure in its hand. After the model contains a skeleton you can then upload the character to another Adobe web app called Mixamo. This converts the character I created on Fuse to a data type compatible with Unreal engine, .FBX to be exact. You can then download the character as a .FBX file and import it to a project.

Audacity

Audacity is a free sound editing software which I used to create sound effects and merge multiple sounds together.

The software is installed on NCI pcs so it was very useful

1.4 Game Analysis

GAME DESCRIPTION

Genre:

- Action adventure – designed to be fun and allow users to explore.
- Time-Trial – completing objective against the clock.

Game Elements:

- Collecting – The player has to collect objects to complete the goal of the game
- Exploring – The player has to explore the map/area to progress further in the game
- Surviving - The player must not be taken out by the AI within the game.

Game Content:

- Realism (A game type that is similar to reality)
- Action (Actions are used such as jumping)
- AI (Adversaries to the player)

Theme:

- Exploration
- Slight thriller ambience.

Style:

- Realism

Game Sequence:

- Linear – Time trial game allows for competition against friends.

Player:

- Single Player

GAME REFERENCE

Player Immersion:

The player will receive enjoyment that focuses on these headings

- Potential scary elements
- Visual (The graphics of the game are too be good)
- Audio (Sound effects that add to the game)

GAME TECHNICAL:

Technical From:

My games graphics are

- 3D (

View:

The game will be viewed by the player in

- Third Person (Viewed as if there was a camera a few feet behind the player almost looking down a touch.)

Platform:

- Unreal Engine 4

Device:

- PC

GAME SALES

Consumer Group:

- Children to Adults that play games.

Distribution:

- Steam – A digital distribution platform developed by Valve Corporation offering digital rights management multiplayer gaming and social networking services. Steam provides the player with installation and automatic updating of games on multiple computers, and community features such as friend lists and groups, cloud saving and in game voice and chat functionality.

2 System

2.1 Requirements

2.1.1 Functional requirements

The ability to use a key board and mouse or a controller.

- The system must be able to accept input.

Start Game.

- The user should be able to start a game when launching for the first time.

Visible/Playable Character.

- When playing the game a Character should always be present and unless in a cut scene the player should remain control of the character throughout.

Character Animations.

- The Character should have appropriate animations when moving such as running and jumping.

Quit Game

- The user should be able to exit the game and return them to the desktop.

Pause Game

- The user needs to be able to pause the game so they can do something else or quit.

2.1.2 User Requirements

The user must have a computer or laptop with a graphic card and a modern OS such as Windows 7+. The game was developed to work on Windows 7+ whether it works on older versions of Windows is presumed but not tested.

2.1.3 Usability Requirements

The game UI menus must be easy to understand and easy to see, have clear well defined tasks, as well as do what it says on the tin. Some other usability requirements are:

Understandability:

- All elements must be easy to comprehend
- The game should be easy to understand.

Learnability:

- The user documentation should be up to date and comprehensive
- The game should be simple to learn

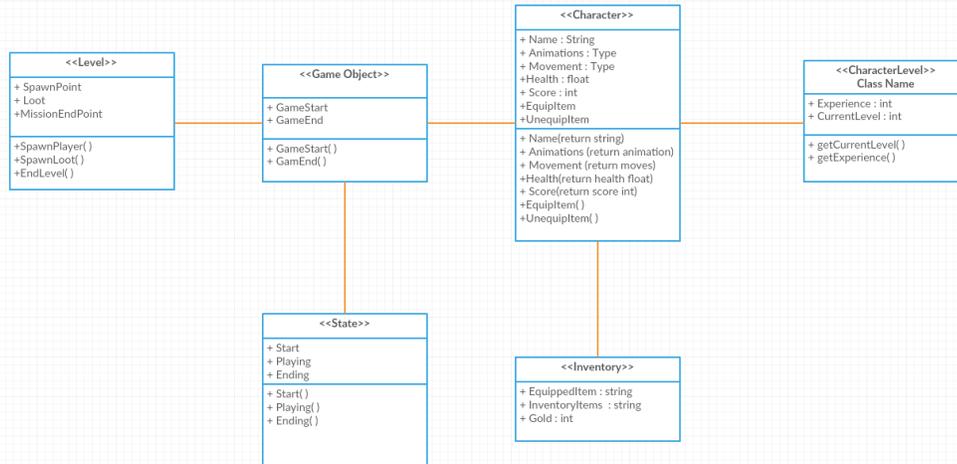
Operability:

- All interface actions should be consistent and lead to expected results.

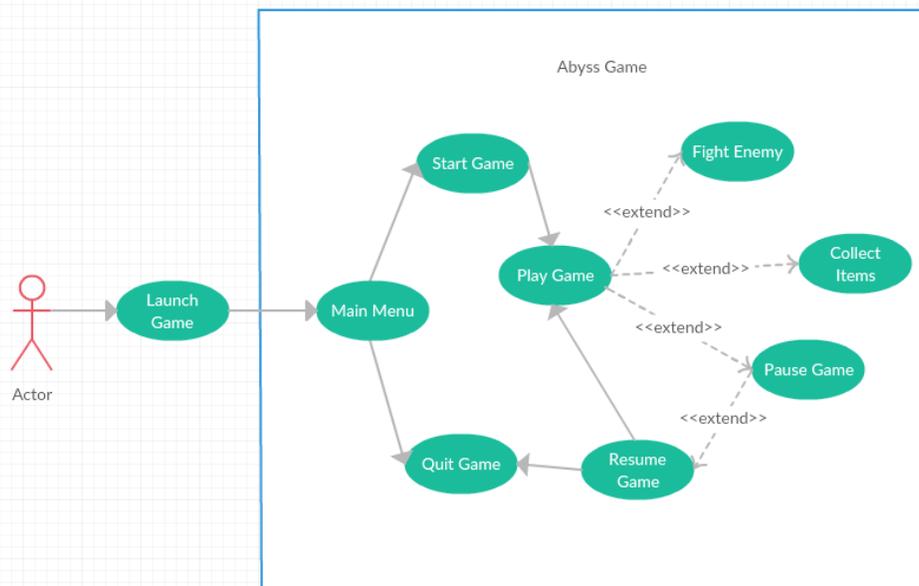
Attractiveness:

- The UI elements layout and colour scheme should be appealing

2.2 Design and Architecture



2.2.1 Use Case Diagram



2.2.2 Requirement 1 <Main Menu>

2.2.2.1 Description & Priority

This is what the player will see when the game is loaded up, they will have access to a Start Game button and Quit game button.

2.2.2.2 Use Case

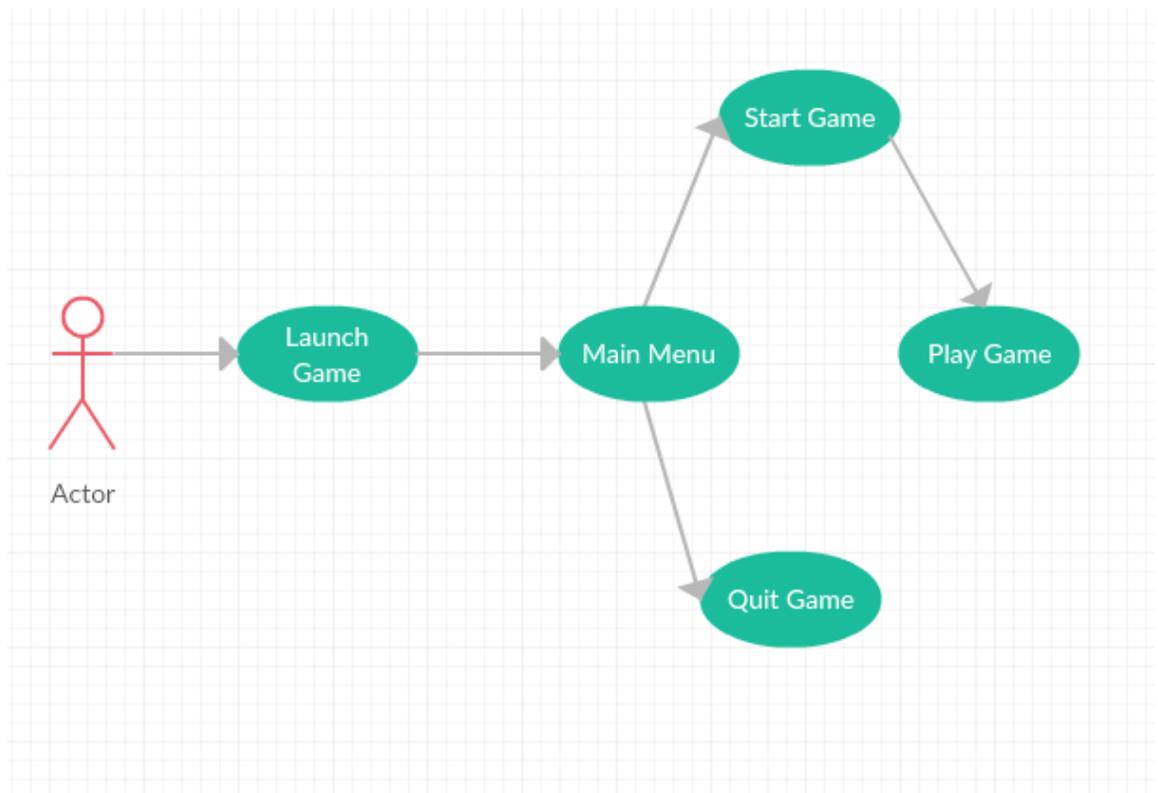
Scope

The scope of this use case is to display the options he user has when the game is loaded.

Description

This use case describes the first interaction with the game after running it.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode

Activation

This use case starts when a player starts the game

Main flow

1. The system identifies the game is loaded
2. The Player sees a menu.
3. The Player selects an option.
4. The Start option is pressed and the user is brought into the game.

Alternate flow

1. The system identifies the game is loaded
2. The Player sees a menu.
3. The Player selects an option
4. The Exit button is clicked and the game closes.

Termination

When an option is selected.

Post condition

The systems continues with the option selected

2.2.3 Requirement 2 <Play Game>

2.2.3.1 Description & Priority

The ability to play the game and complete all of the in game functions

2.2.3.2 Use Case

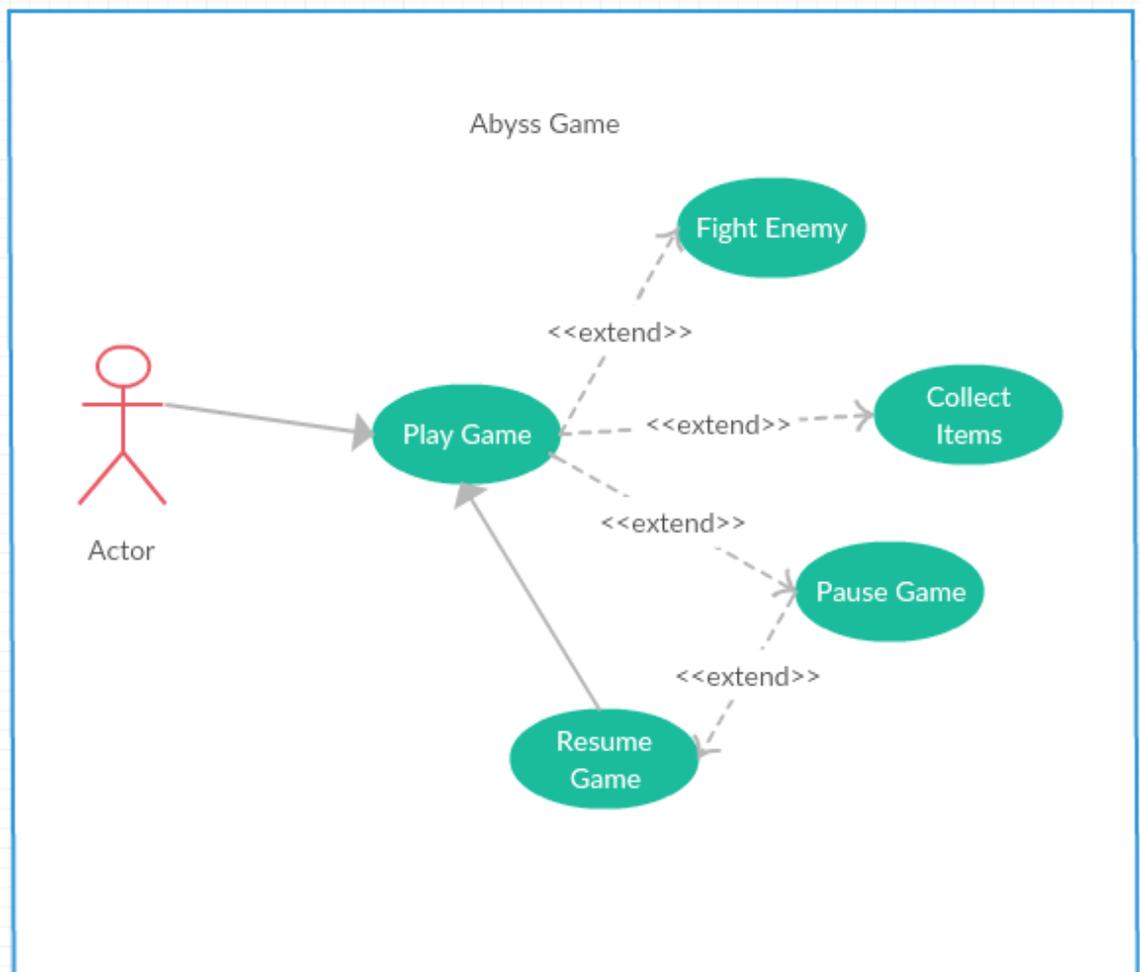
Scope

Play the game and use its functions.

Description

This use case describes the general usage of the game.

Use Case Diagram



Flow Description

Precondition

The system is in the start menu.

Activation

This use case starts when a Player selects the Start game option.

Main flow

1. The system identifies that a player is using the game and a level is loaded
2. The user can proceed to play the game.
3. The system records the number of items the player collects.
4. The player must either run or hide from the enemy.
5. When the player has collected enough items they must escape.

Termination

The user pauses then the user quits game.

Post condition

The system is closed

2.2.4 Requirement 3 <Collect Items>

2.2.4.1 Description & Priority

Collecting and storing in game items

2.2.4.2 Use Case

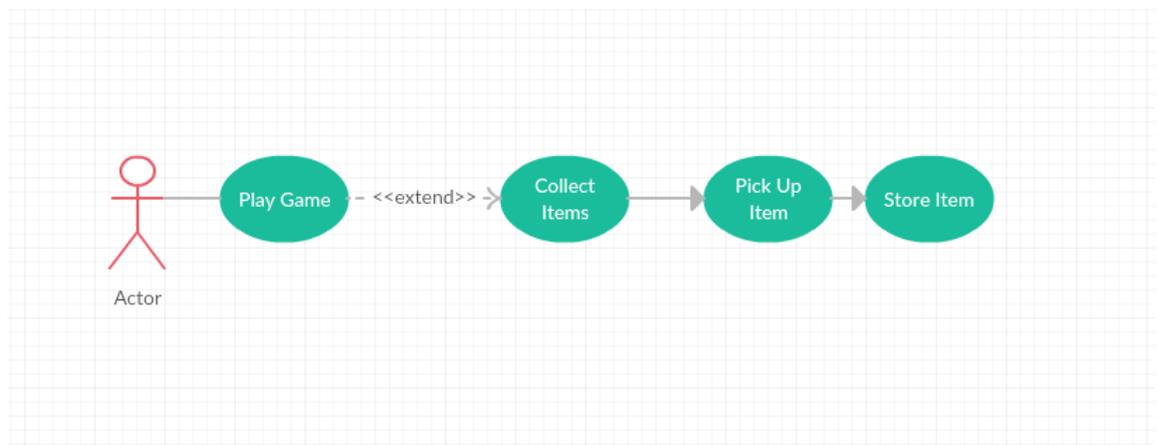
Scope

Giving the player an objective to collect items.

Description

This use case describes process of collecting and storing items.

Use Case Diagram



Flow Description

While the player is playing the game there are items in the level, players can collect them and store them.

Precondition

PC is on.

Activation

This use case starts when a user is playing the game.

Main flow

1. Playing game.
2. Finds item.
3. Picks item up.
4. Stores item.

Termination

When all items are collected or if game is Quit..

2.2.5 Requirement 4 <Launch Game>

2.2.5.1 Description & Priority

Launching game from Desktop.

2.2.5.2 Use Case

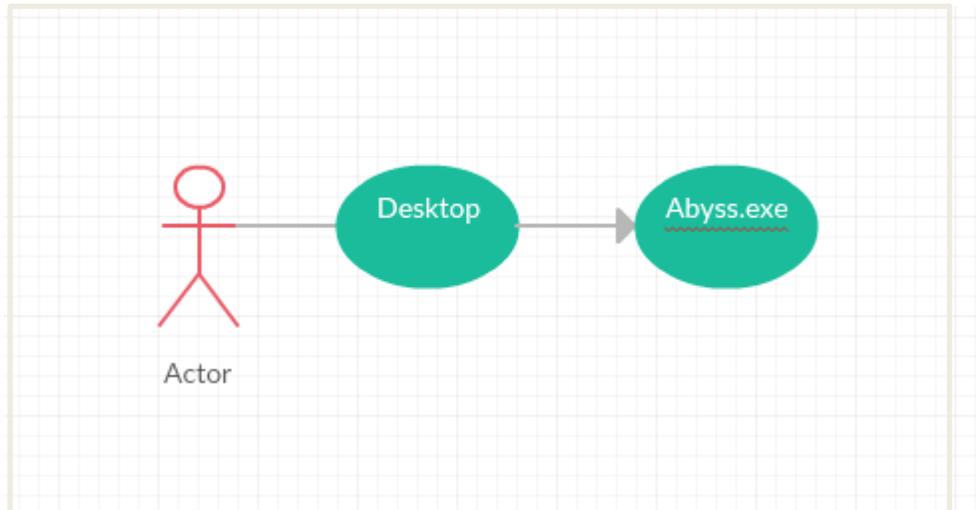
Scope

Running the game.

Description

This use case describes the launching of the game.

Use Case Diagram



Flow Description

Precondition

PC is on.

Activation

This use case starts when a user wants to open the game.

Main flow

1. Launch Windows.
2. Find the program.
3. Launch exe.

Termination

When the game is launched.

Post condition

The game is launched.

2.2.6 Requirement 5 <Pause Game>

2.2.6.1 Description & Priority

Pausing the game from the playing game use case.

2.2.6.2 Use Case

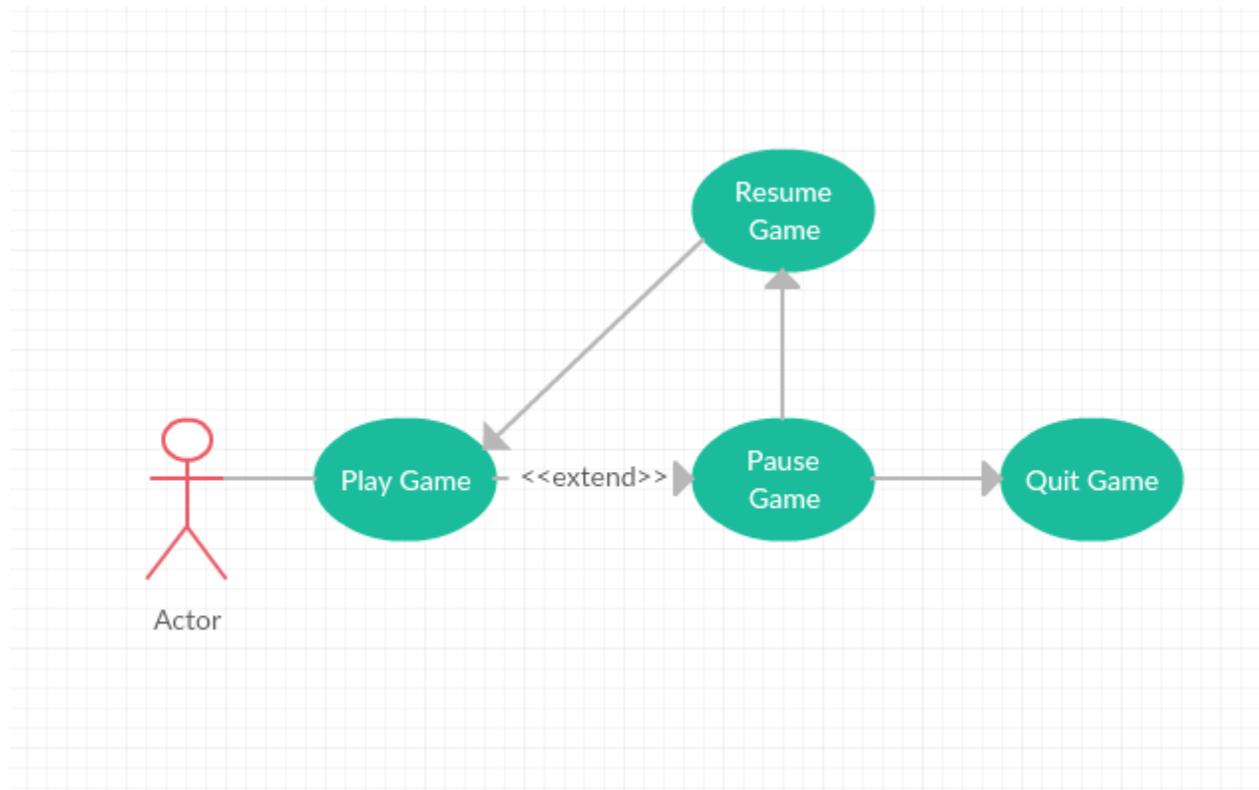
Scope

Being able to pause the game and resume from the same location.

Description

This use case describes the processes of pausing and unpausing the game

Use Case Diagram



Flow Description

While playing the game the user can pause to bring up the pause menu, and when this menu is on screen the game is paused. From this screen the user can return to playing the game or quit the game.

Precondition

Game is playing.

Activation

This use case starts when a user presses the P key while in game.

Main flow

1. Playing game.
2. Press the pause button (P key).
3. Game pauses.
4. Return to the game using the Resume button.

Alternate flow

1. Playing game.
2. Press the pause button (P key).
3. Game pauses.
4. Quit the game by using the Quit button.

Termination

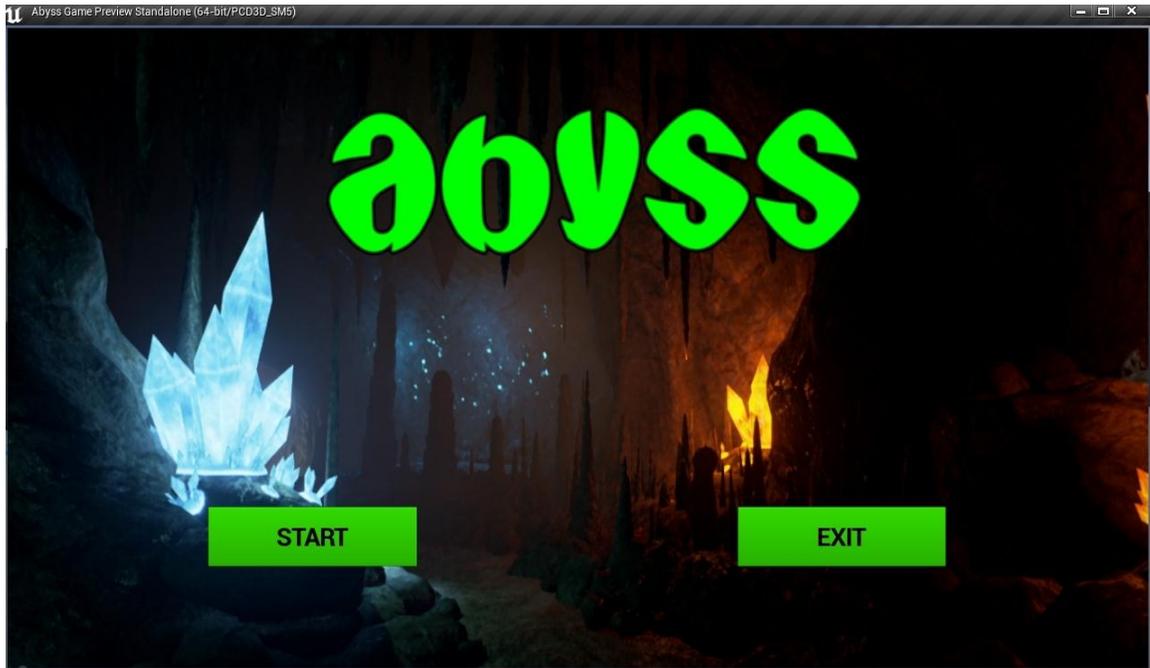
When Game is Resumed or Quit.

Post condition

The Game Resumes or Game is terminated.

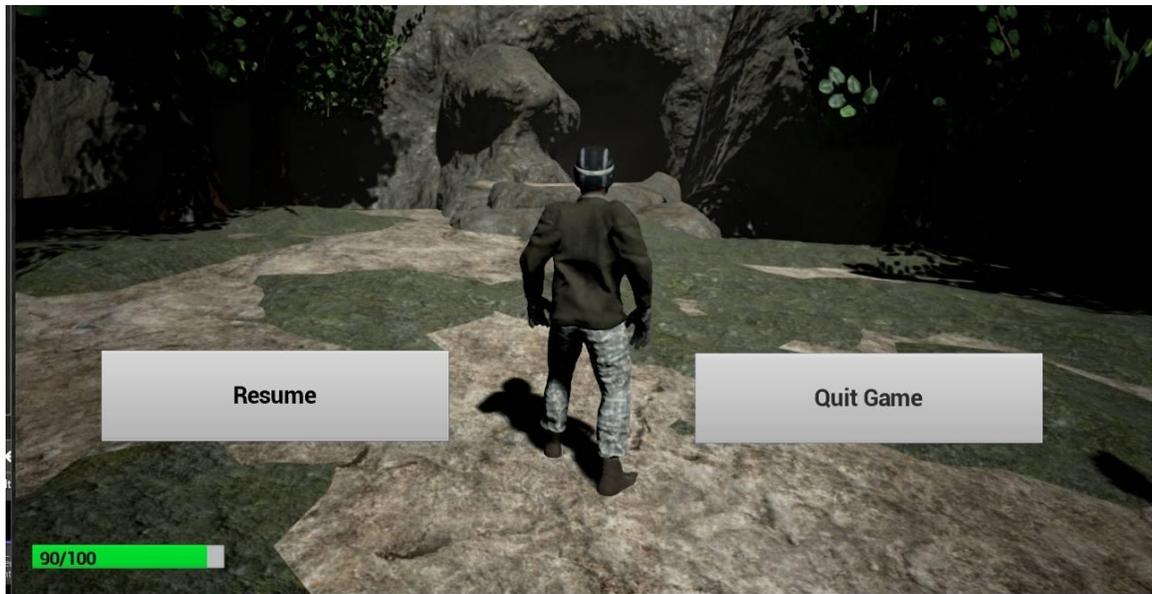
2.3 Graphical User Interface (GUI) Layout

Main Menu Screenshot



This is just the basic main menu that allows users to start the game or close it. It uses in game still frames and a contrasting colour pallet for a more vibrant and attractive menu

Pause Menu Screenshot



Simple but efficient pause screen, allows users to exit the game and pause progress. Will also allow users to leave the game idle then come back once they are ready.

2.4 Testing

Testing is a vital part of any software project so I felt it would be best to perform multiply types of testing while developing the project. This is important as it should ensure that the software acts as intended throughout usage in every way while assuring that requirements can also be met.

2.4.1 Usability Testing

Usability testing was completed with the goal of distinguishing how unique users interacted with the UI aspect of the game. I used a pool of 5 participants to test this. Usability Testing gave insight on how people saw the game and if they liked the menus and overall look of the game some of the results are as follows:

Some of the testers did not appear to like bright yellow for text as it was kind of hard to see, green was suggested so I ended up switching to that.

Another suggestion was making the game a little brighter. I did this by adding a flashlight to the character.

2.4.2 Unit testing

For Unit testing a test plan was put in place that was completed after making changes to the project code or interface. The test plan was something similar to as follows:

Can the user start the game?

Is the camera still showing the character?

Is the character applying to correct physics? (i.e. Is the character interacting with the map as intended, Not being able to walk through walls and being grounded by gravity etc)

Do the WASD keys allow the character to move and jump?

Does F still turn on the flashlight?

Does P still pause the game?

Does the player still take damage/regain health?

Can the player collect objects?

Are these objects counted?

Are the hit boxes between player and objects working?

Is the AI working?

Are animations working?

Can the user quit the game?

The test plan is quite easy to plan out as generally you can inadvertently check multiple parts by doing one.

2.4.3 Customer testing

For this section it was important to test a variety of users so the game can be evaluated from an average user's standpoint. Mainly by taking casual gamers and experienced gamers and seeing how they find the controls and gameplay.

It was evident that the expert gamer could almost instantly pick up on the games controls and mechanics due the fact that I stuck with the industry standard for pc game controls, WASD keys for movement, P for pause etc. One thing I did notice was that Left handed players struggled with the controls more due to mouse and keyboard placement. To solve this problem I made it so the arrow keys on the right of the keyboard also allow the player to move.

Casual gamers also took to the controls fairly easily after some practice. I think the major benefit of this testing was that out of the 7 people that tested it for me, 4

different minor bugs were found, which consisted of an area where the player could get stuck, a wall which could be made transparent by walking right up to it, a glitch where the flashlight would break if F was clicked multiple times in quick succession and animation errors when a player jumped over an object.

The testing participants consisted of 3 Females and 4 Males, all of them had gaming experience some minor and some expert. Ages varied from 6 – 42. I also tested the game with 3 Non-gamer participants. They struggled with the controls and found it difficult to control the camera and move at the same time. I solved this problem by lowering the speed at which the user can rotate the camera making it easier to control.

From this I gathered that this game is targeted mainly at gamers who have some sort of experience. It goes hand in hand with the fact the game is graphically intensive, generally there is a correlation between people that have a PC powerful enough to play heavy graphical games and that play games.

3 Non-Functional Requirements

3.1 Performance/Response time requirement

The performance requirements for the game are:

Operating system: Windows 7/8 64 bit

Processor: Quad-Core Intel or AMD 2.5 GHz or faster

Memory: 8GB Ram

Video Card/ DirectX Version: DirectX 11 compatible graphics card

3.2 Availability requirement

The game will be available for free download from a site such as Github.

Could also be available on a custom website made for the game

The reason the game is free would be to reach a greater audience and receive more feedback from a first project.

It could be also distributed from Steam on PC. The Green Light program promotes and releases games made by developers on the steam platform from as early as alpha stage to spread awareness and even help with funding.

Unreal Engine 4 Market is another platform for selling/distributing products.

3.3 Security requirement

There is no major security requirement for the game, it would not require any internet connection to play the game, therefor they only need to download the game and it will run. AKA have rights to download on the current machine of use.

3.4 Portability requirement

The game will be able to download onto most PC and laptop machines.

3.5 Data requirements

The player must be able to save their current progress. And be able to load it at a later time

3.6 User requirements

Understanding of how to use a controller/keyboard/mouse

Have access to a computer/laptop

3.7 Environmental requirements

A computer/Laptop will be needed to run the game.

3.8 Usability requirements

The game will begin with a tutorial which will explain the basic game controls and show the player how to play and use the game.

3.9 Extendibility requirement

In the future I could add in Downloadable content which could add in even more questions and quizzes or even different moons of the planets to expand the game. It can also be updated with more new information we learn about the solar system.

Reusability requirement

The assets that I will make in this game can be used in future DLC or even future projects that I might start.

4 Implementation

Abyss is a basic indie game that is based around a simple objective and backing it up with cool characters and awesome graphics.

To quickly give a sort of legend on blueprint code:

Red Nodes are events.

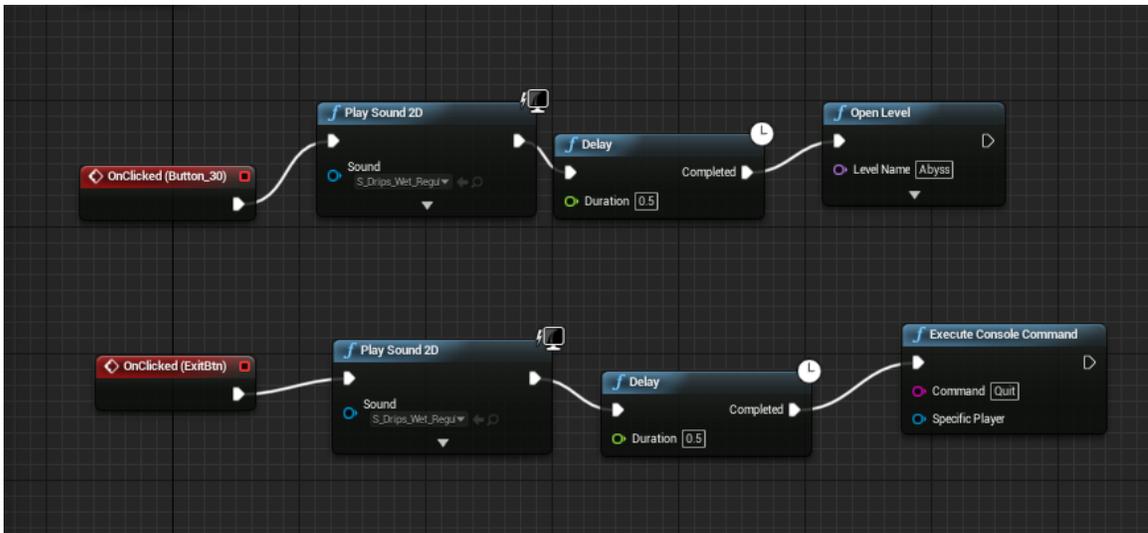
Blue Nodes are functions.

Green Nodes can be references, vectors and mathematical instances.

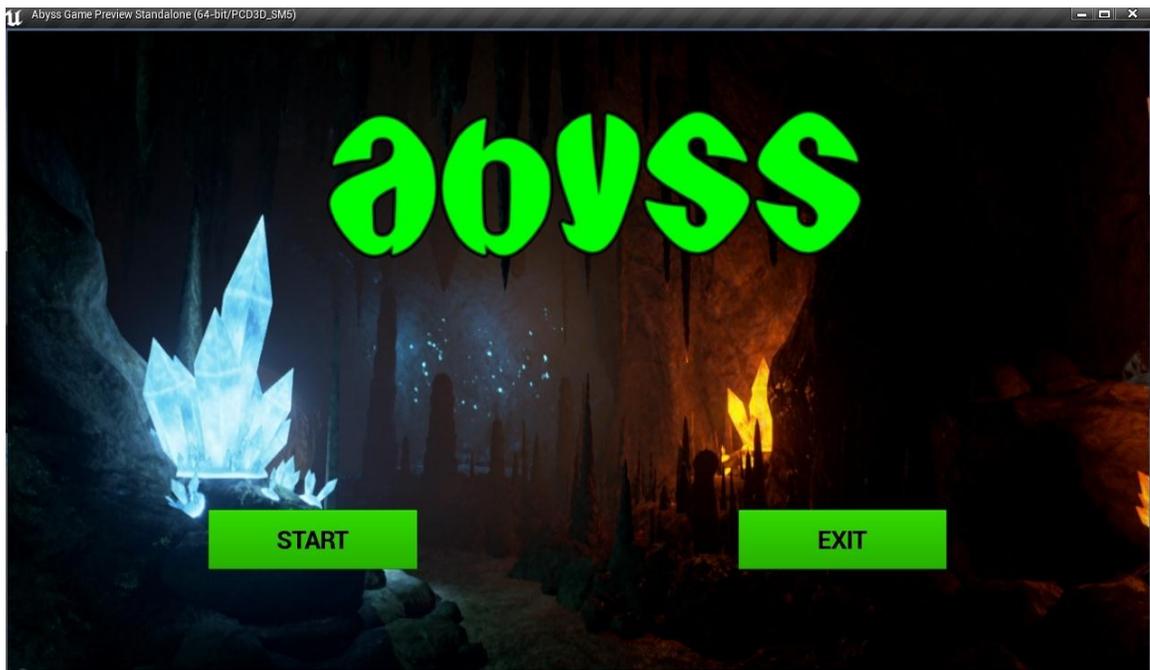
Green Function nodes are generally Get/Set applications.

Grey Branch nodes are very similar to IF statements.

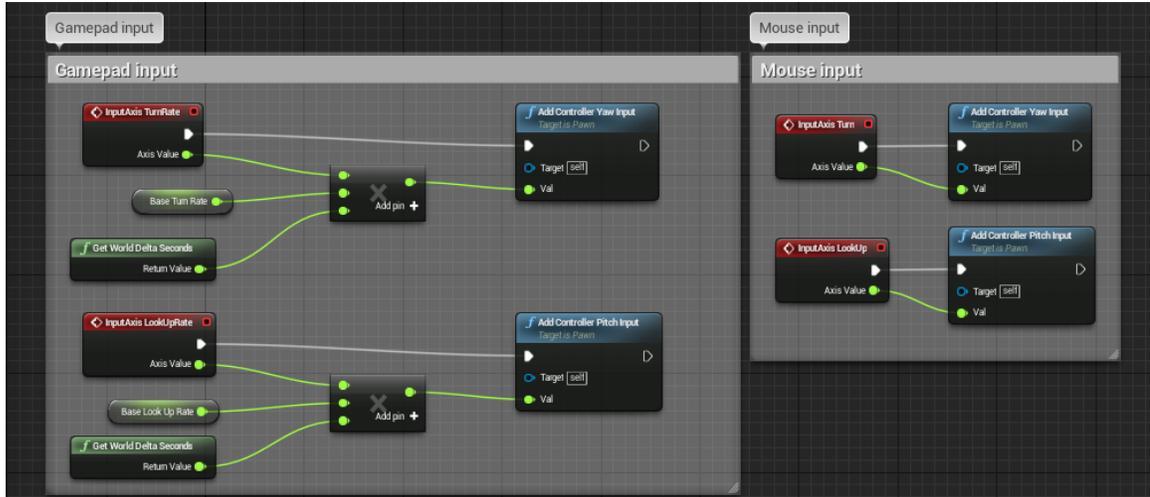
4.1 Main menu



The when the game starts it loads a widget, a widget is basically a screen in which you can have buttons, images and display that can complete functionalities. This is the main menu widget blueprint. The two red event nodes are OnClickListeners. They are attached to the buttons on the widget displayed below. The second set of nodes are PlaySound functions that play some audio when the button is clicked, I added a delay so that the sound could play before the the next function is called. The last set of functions redirect the user from the main menu to their respective functions (The Start Game and Quit Game)

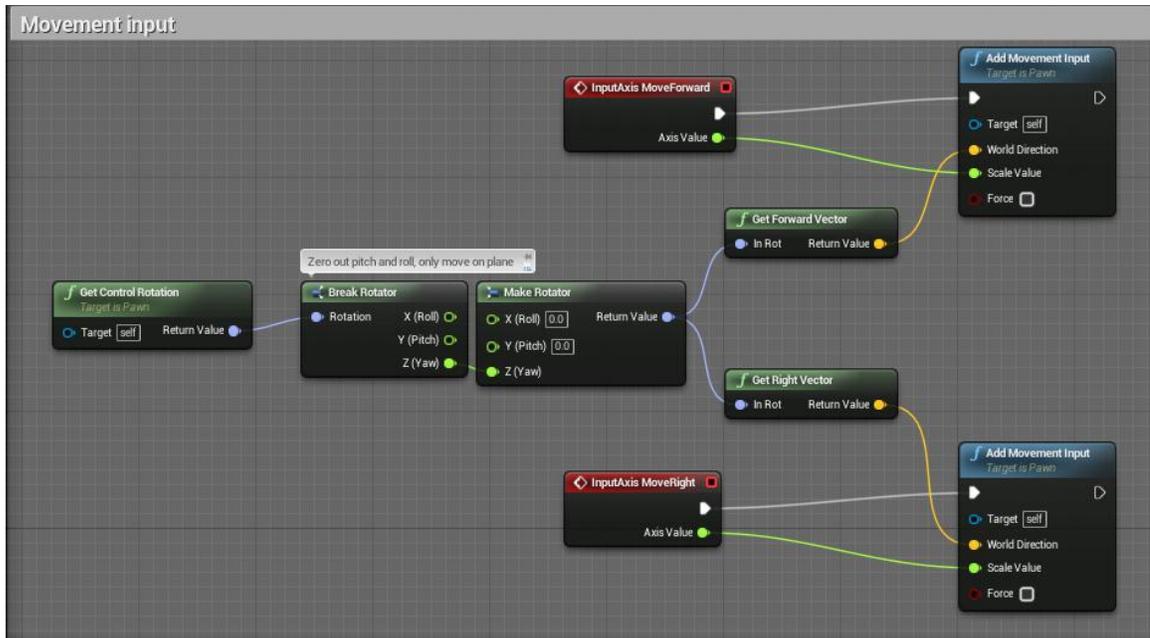


4.2 Input Recognition



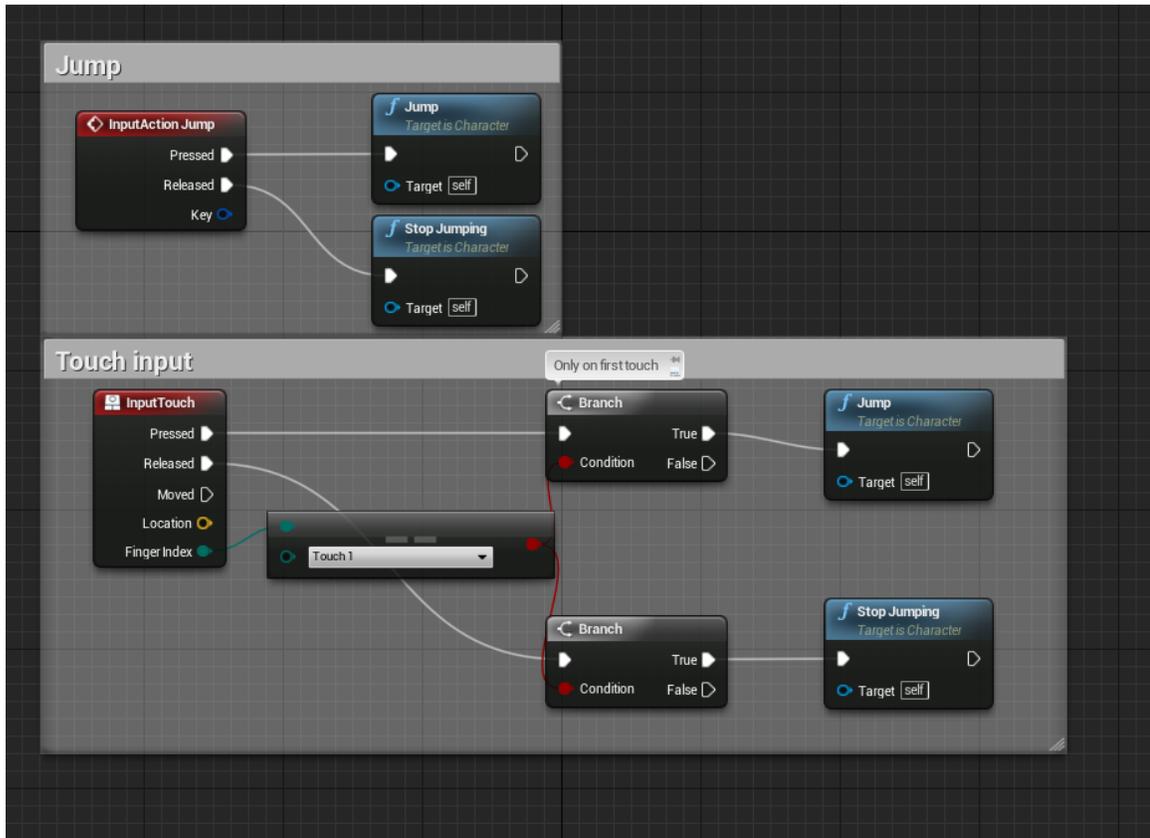
These two sets of blueprints allow both controller and keyboard inputs to work and gives mapping ability (choosing respective button functions on a controller and having them work as it would on a keyboard Eg - On keyboard flashlight is mapped to F, on a controller you can map that to the X button etc).

4.3 Player Movement



These nodes all for the rotation of the camera, they get the players current location, set directions from left, right, forward and backward from the given location. Then take inputs from the class above and apply them to the movement input functions which allow the user to move the character in the 4 cardinal directions. Combinations of 2 directions also work to move diagonally.

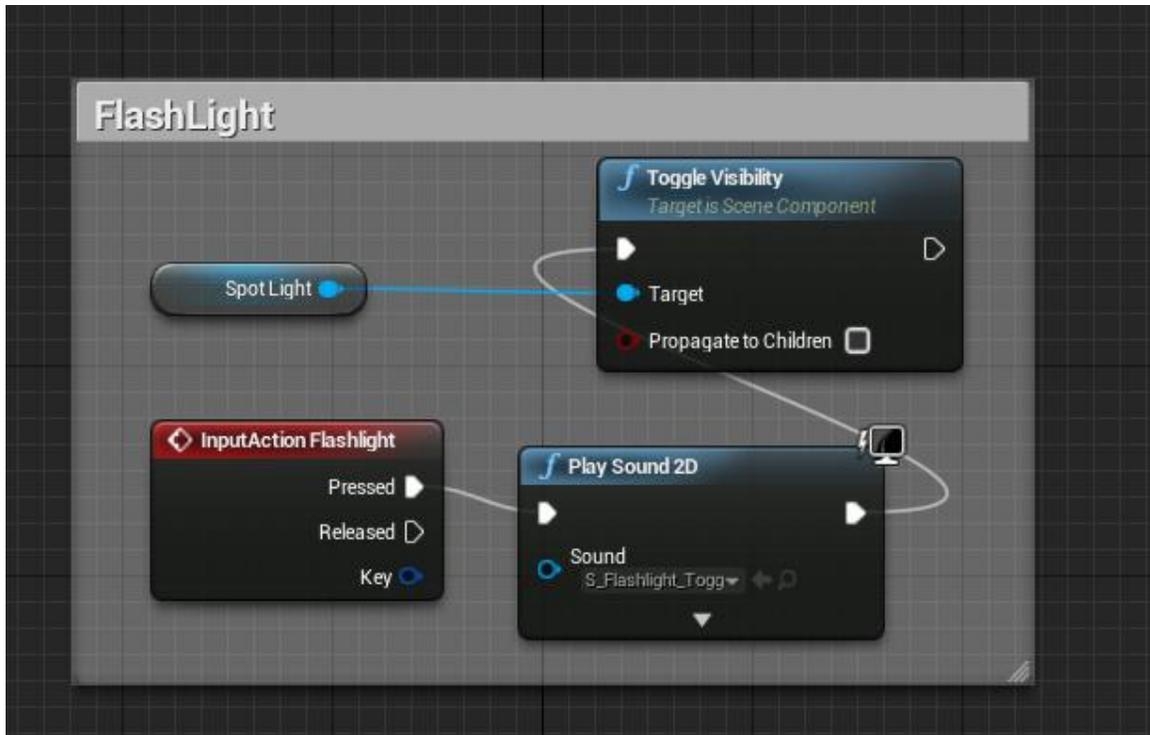
4.4 Jumping



Jumping is a separate type of movement as the play cannot make any other motions while jumping. Firstly the action jump has to be assigned to the character, from there you can set the speed and height at which the character can jump.

Secondly because jumping is a timed action it must have a set beginning and end. The function Stop Jumping is used to end the jumping animation and bring the character back down to the ground. A Branch is used to determine when the character is jumping and not jumping.

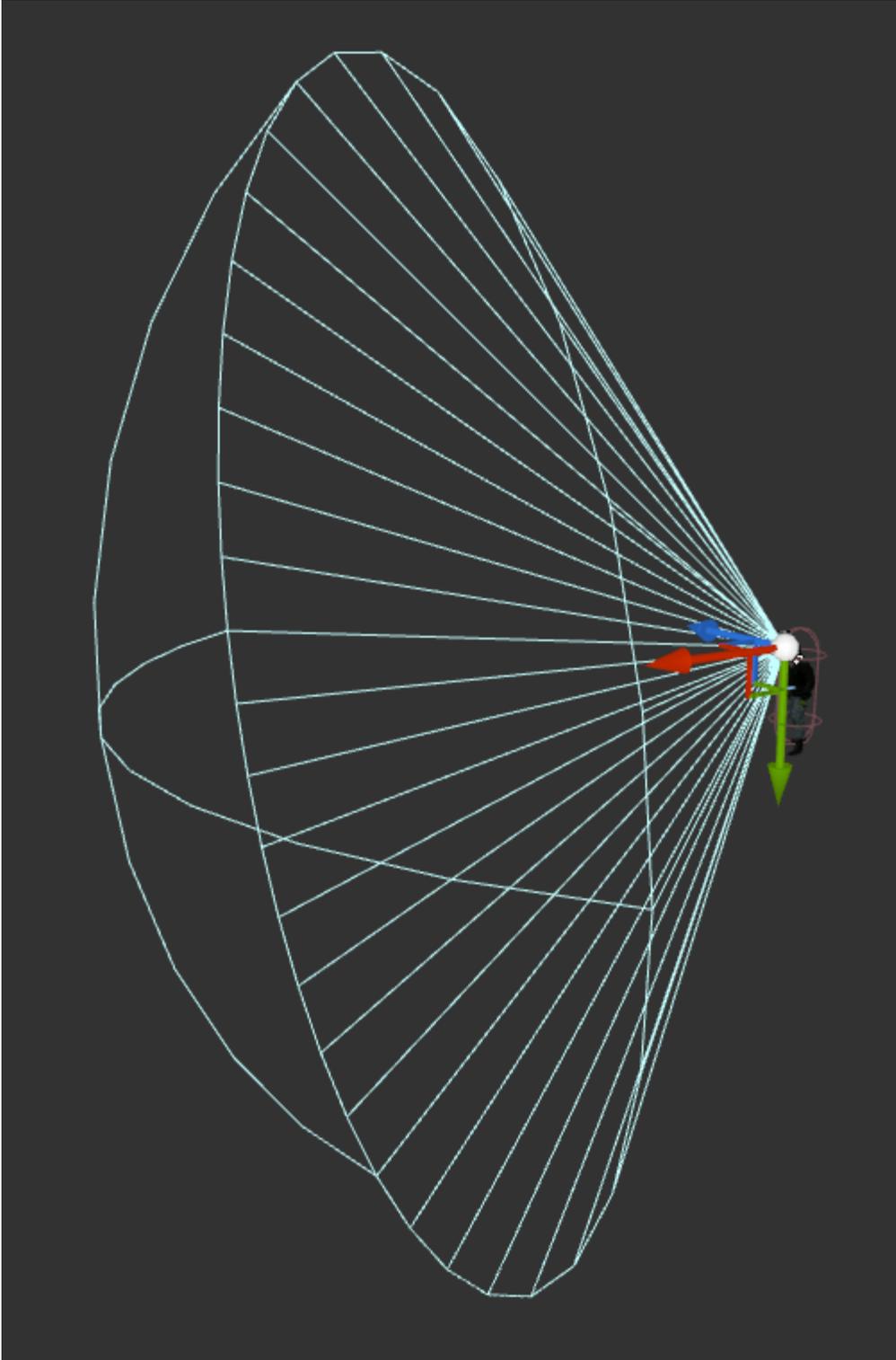
4.5 Characters Flashlight



The Event in this class is an InputAction which is basically the same as an OnClickRelease in java. The class states that when a key is pressed (set to F in system functions) it will play a sound then toggle the visibility of an actor reference called spotlight.

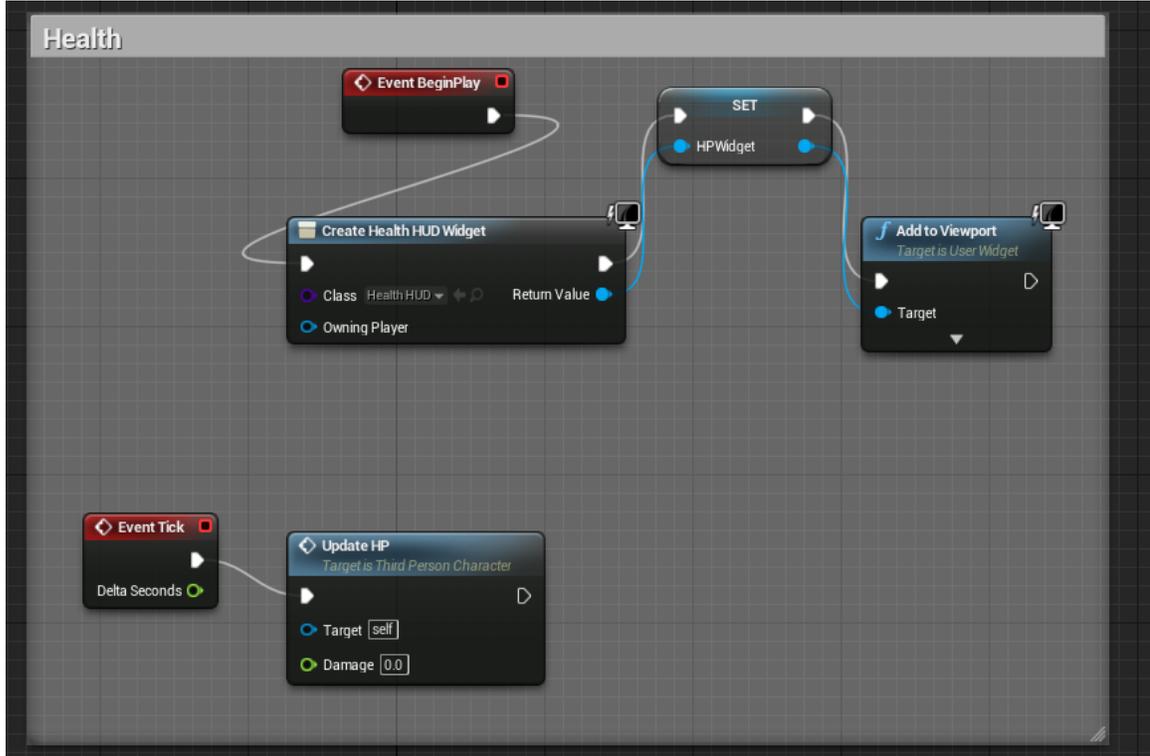
An actor is an object in the game world.

Spotlight is a reference to a lightbulb element that I have attached to the character as shown below.



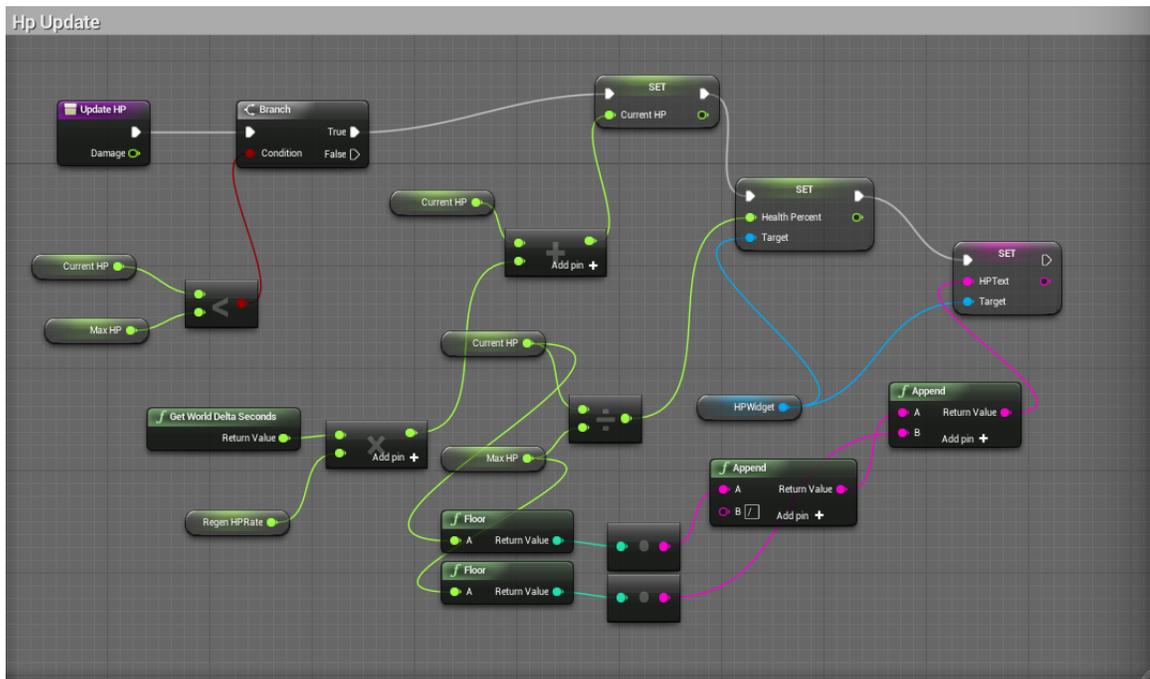
What you see here is the character and a cone coming from the spotlight, the cone indicates the light coming from the spotlight. This gives the effect of a flashlight.

4.6 Health HUD and player Health



This class is used to display the player's health on screen. The Event BeginPlay node basically states that when the game starts run this code. The Event node immediately links to the Create Widget function node which in turn displays a widget on screen with information about the player's current health. It also sets the players Health (HP = health points).

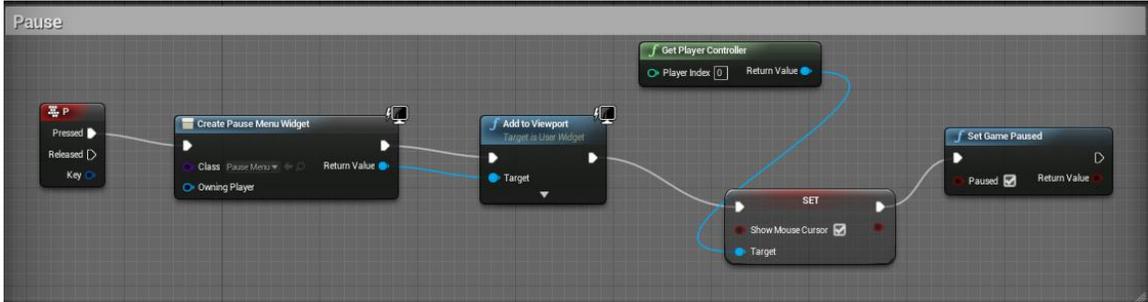
Below that there is another Event, Event Tick this states that on each tick call the Function Update HP.



Update HP is a rather big and messy function that takes lots of variables and compares them to one another. A brief tour of how it works goes as follows:

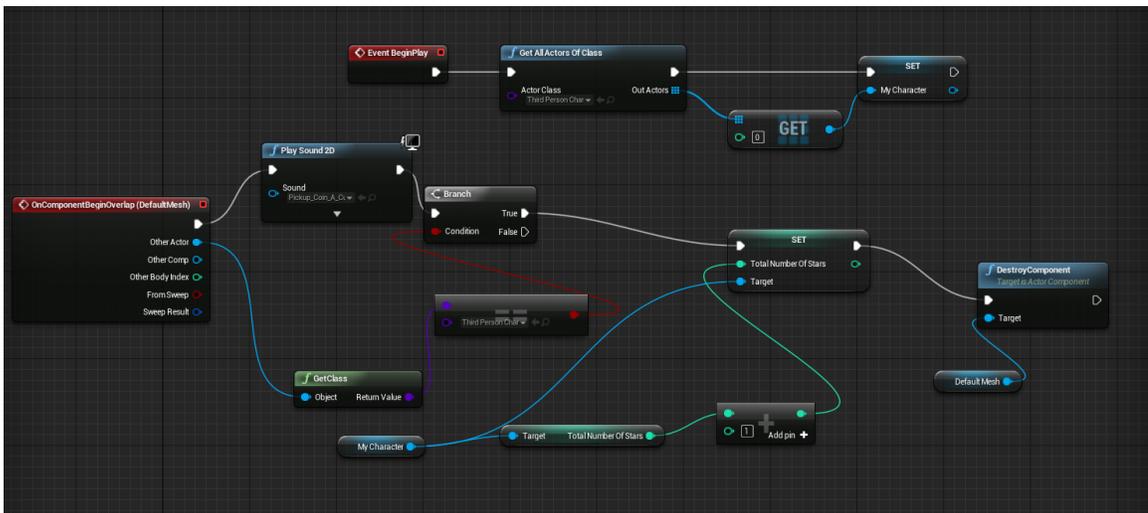
The function is called in the previous class, it starts off by using a Branch node which will only output a true value if its condition is valid, the condition being that the character's Current health is less than its max health. If it is not less it would not be necessary to update as it would be full. A set node is then used to determine current health by taking the value from the game. The purpose of this class is for the character to regain health over time. So checking current health every tick is important. To regain health this function takes current health and adds 5 if the character is not full. Once this is done it returns the value to the widget and it is displayed onscreen.

4.7 Pause Screen



Basically this function happens on P key press. It calls the Pause Widget that I created and adds it to the viewport (Screen). It then sets the cursor to visible so the user can click the options available. It also removes control of the character so they will not try to move while paused. The last node pauses the world so that no more time can pass, AI will freeze and no other functions other than the pause menu screen will work.

4.8 Gem Pickups



This blueprint allows the player to interact with objects in the game such as collectable Gems. The goal is to collect these gems and escape in the best possible time.

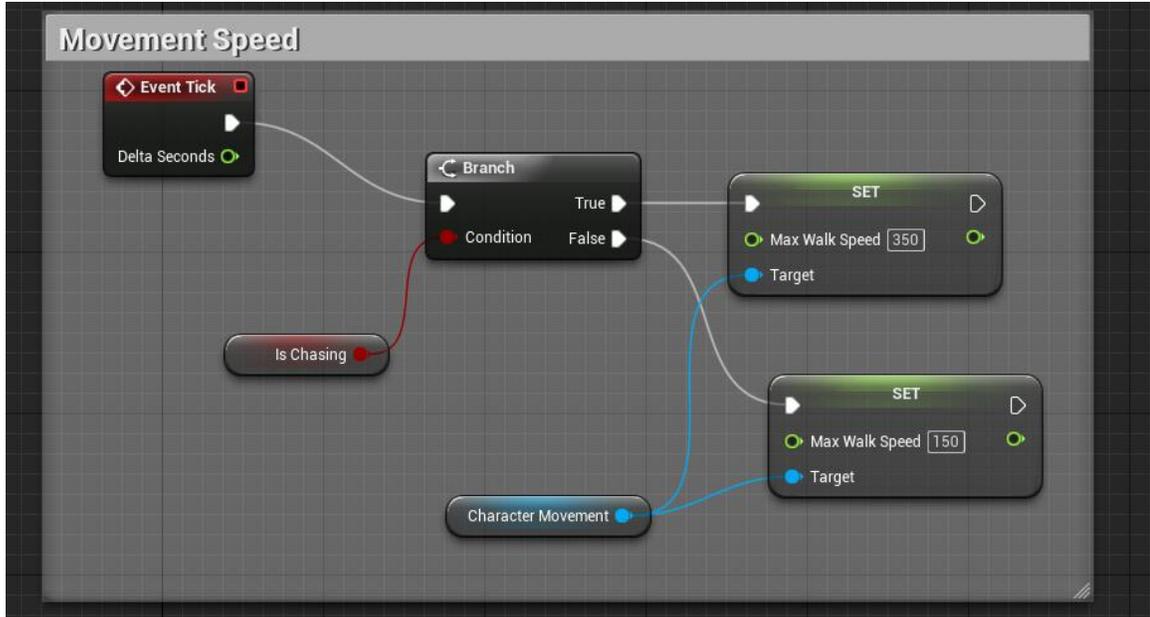
The blueprint starts with an Event BeginPlay that allows all the actors within the games class to be interacted with. The second event node is when the character overlaps with an intractable object. In this case of gems, the character will be able

to collect and store them. When the player collects a gem a sound will play and the gem will be stored. The amount of gems a player has is available to see on screen.



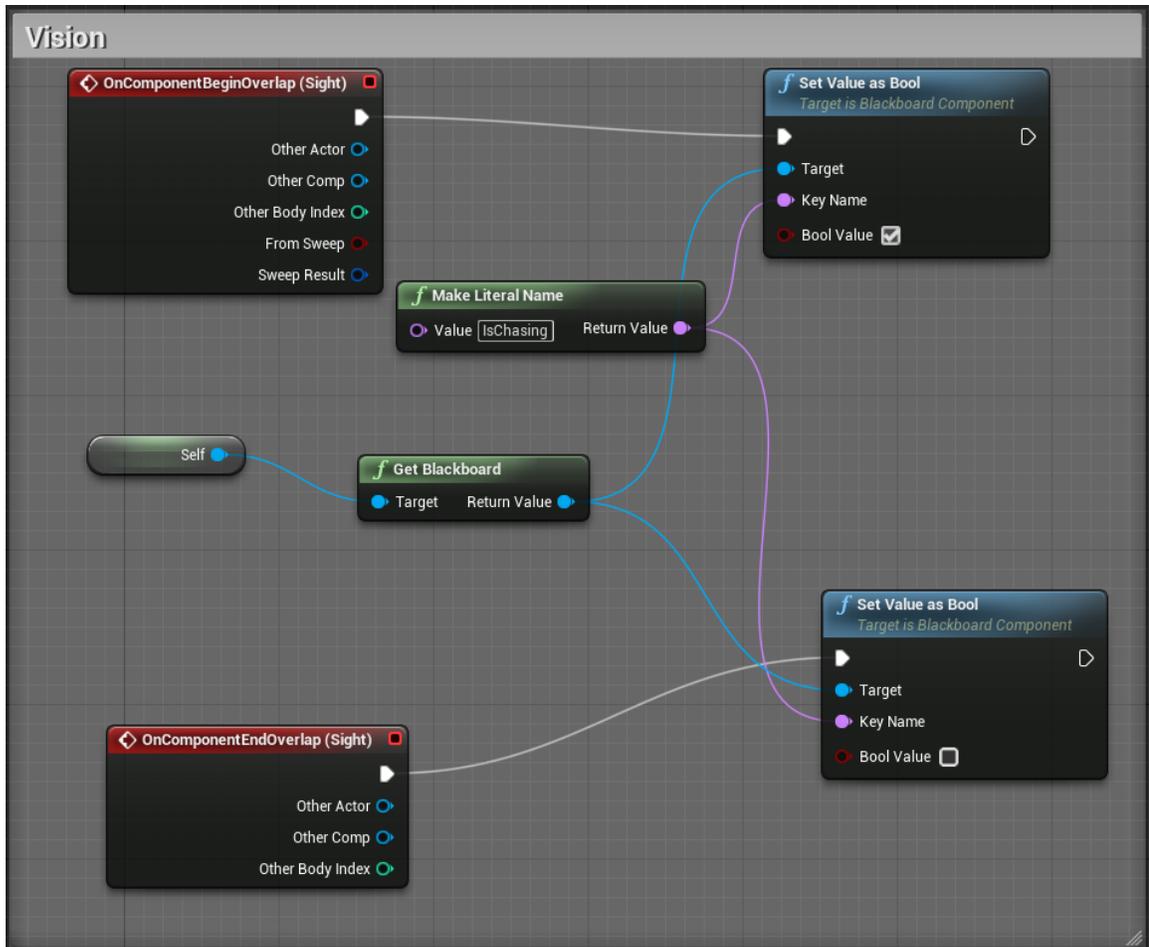
4.9 NPC (Non Playable Character) Enemy

The NPC has multiple modes which will be explained further in the AI section below. The NPC is a mutant cave dweller with high detailed design and functions.



When the NPC is in Chase mode his max movement speed is increased so he can catch the player. This is done by using a branch and a condition stating that if the NPC is chasing move speed increase is the true output.

I have also given the NPC the ability to see the player, and when it sees the player he will change to chase mode.



As you can see above there is an Event called Sight, when this occurs the Boolean value behind the mode IsChasing is set to true, thus activating the AI's chase mode. How this works will be explained in more detail below.

4.10 AI Aspect of the game

In my project I decided to add an NPC enemy AI that would roam the level and if come across the player chase them and eventually attack. The attacking I have yet to fully implement I have the animation created currently but I have not managed to incorporate it in as of yet, but will definitely have it for the final product.

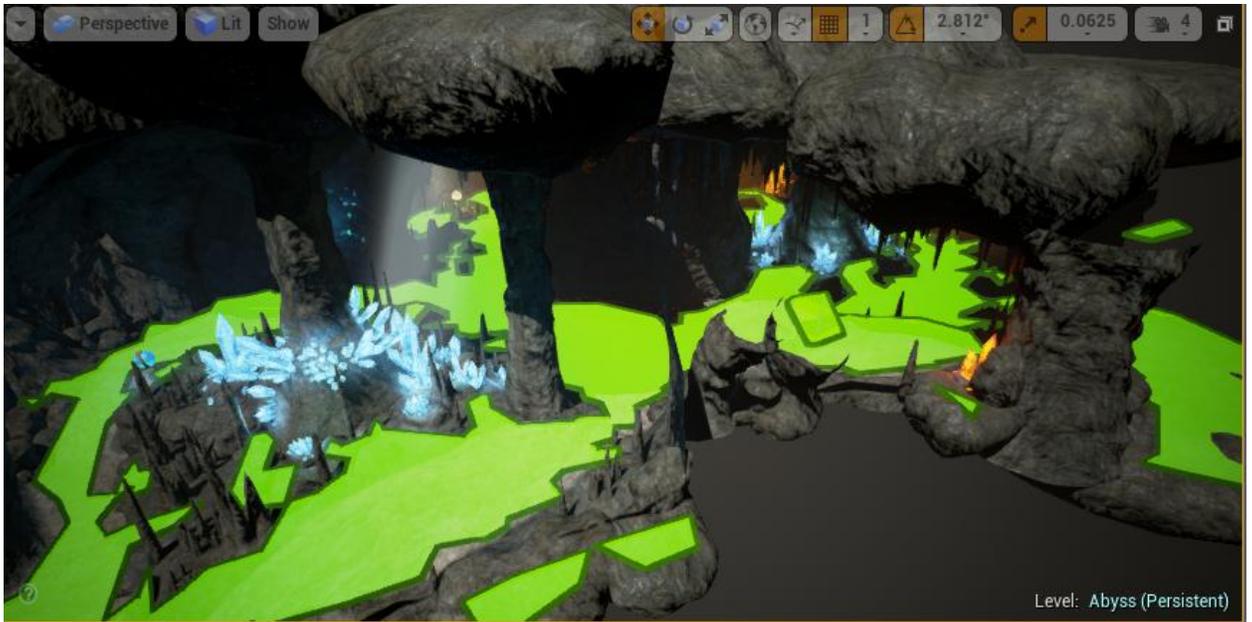
4.10.1 *How it works?*

The process of using behaviour trees in Unreal is as follows:

1. Have a character class or object that you wish to apply AI to like a human npc mesh.
2. Create an animation blueprint so that character will be able to react and move naturally etc.
3. Have an AI controller class that will work with the character class to compile actions.
4. Create a behaviour tree to act as a brain for the AI
5. Create a Blackboard blueprint class which can be thought of as the AI's memory.

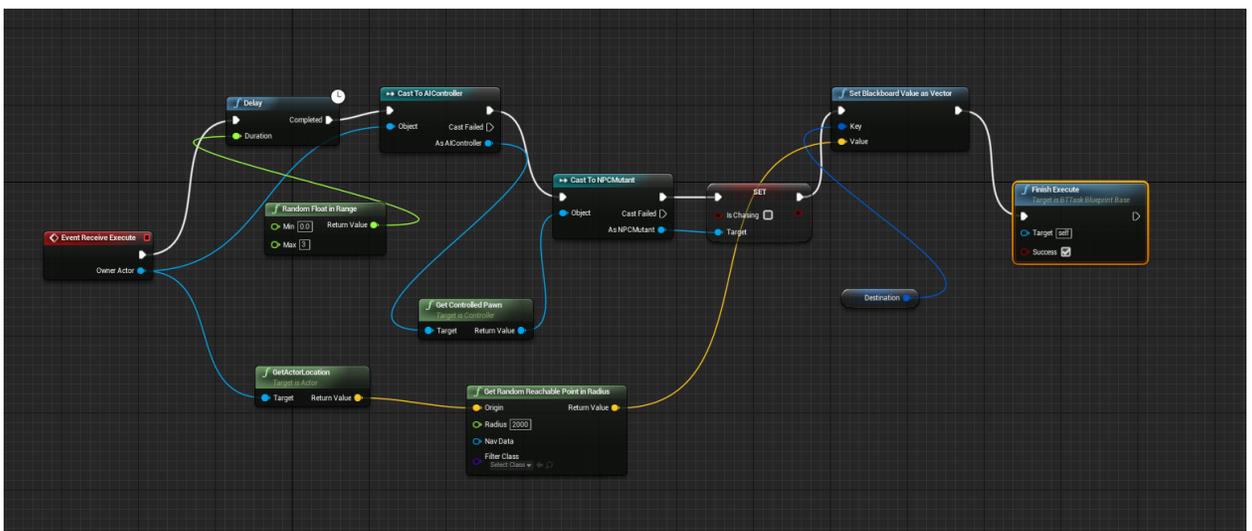
As well as having a character there must also be a terrain for the character to traverse. To combat this I used a NavMesh.

Currently the AI will roam around the map through a NavMesh that I created on Unreal Engine. Which can be seen below indicated in green.



The green area is the AI's field of potential pathing. Although the AI can traverse along anywhere within the NavMesh I have given the AI a patrol radius of around 2000 units.

The patrol or roaming behaviour is created in a blueprint class. (May need to zoom)

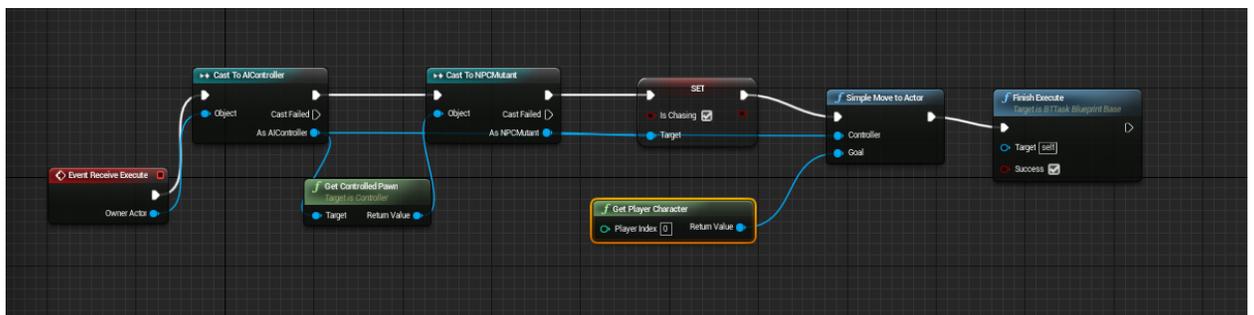


The red node on the left indicates that an event has begun, in this case it is the event of the game starting. The first thing that occurs is actually a function that causes a delay in the NPC characters decision. I added this delay to create for a more natural transition process between an idle character and a moving character.

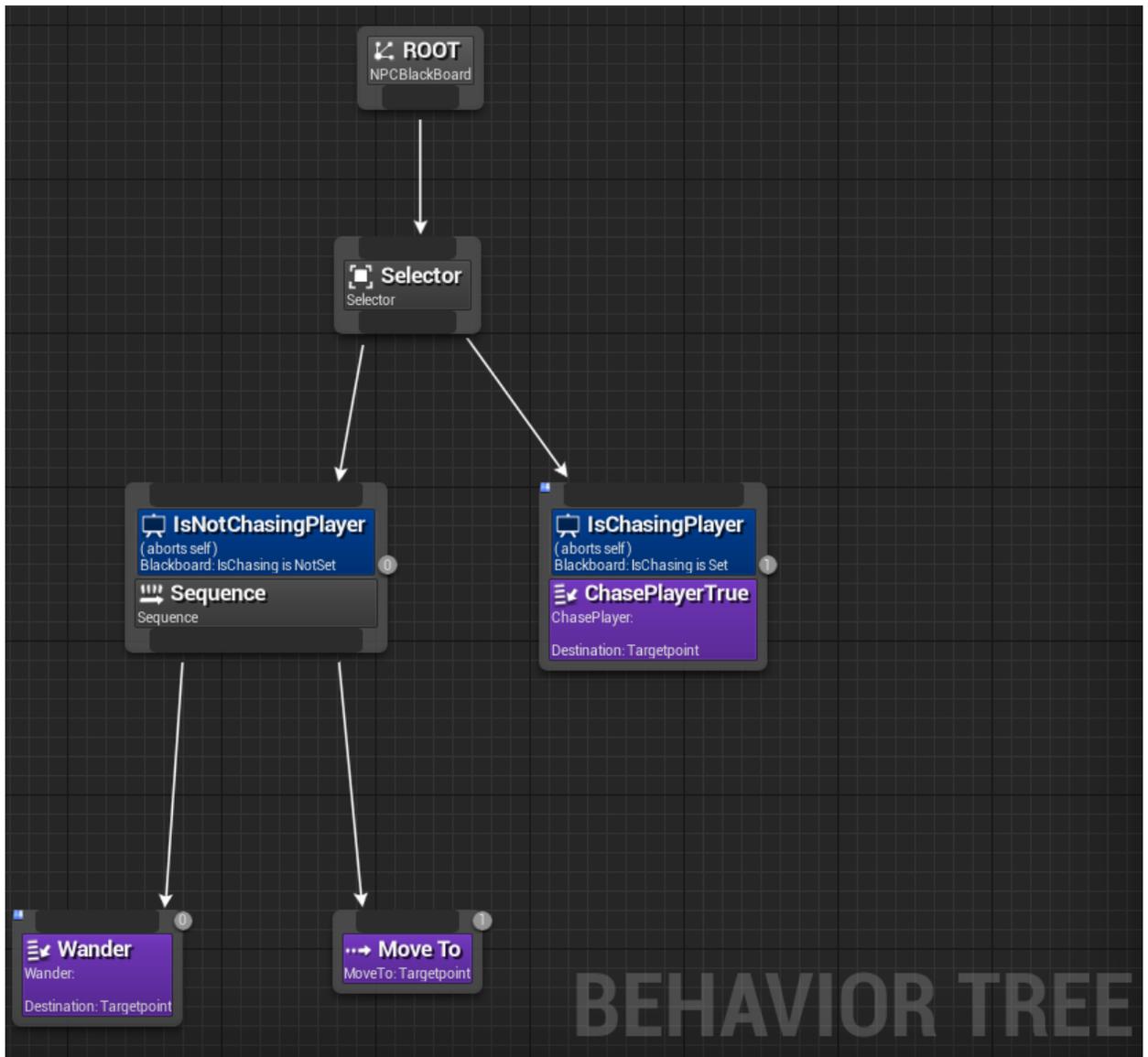
The second function that happens is the assignment of a random destination point within 2000 units. This target will be the area the NPC goes to investigate, once it reaches the target another point is assigned.

This function will keep the NPC wandering the NavMesh until something or someone interrupts it.

Once the NPC is interrupted by the player a new class will come into play which allows the NCP to chase the character once they are in its line of sight. The line of sight is created by giving the NPC a field of view a set distance in front and when a player collides with that field of view this event is triggered.



The behavior tree combines both of the above classes to form the ability of changing states for the NPC.



The selector node will allow the AI to choose either 2 of the children nodes isChasing and isNotChasing. These 2 states are defined here and in the blueprint classes above. The isNotChasing node also has to states wander which is basically when the NPC will roam the NavMesh and MoveTo which lets the NPC move to a specific location.

4.10.2 *Unreal Engine Behaviour Trees.*

After some research about how Unreal Engine handles AI I decided that the best and most efficient method for me was Behaviour trees that can be accessed through Unreal Engines client.

Behaviour trees in UE are event driven, which is massively important for game performance as it will avoid the constant checking and potential pruning that would happen otherwise. For example the AI will only chase the target once a player comes into view, the event of the player entering the AI's line of sight.

Main components of UE Behaviour Trees.

- **Decorators:**
Unlike traditional behaviour trees, in UE a decorator system is used to determine conditions.
These decorators make for easy reading and will allow the user to navigate the tree and see whether a node is active or inactive.
This model is very beneficial as it is easy to keep track of leaf nodes and their functions, how they are being implemented and the conditions that they are applied under.
- **Simple Parallel Nodes:**
Only allow two children, one of which must be a task node and another a subtree or another function. A benefit of this would be how easy it is to optimise.
- **Composite nodes:**
Decision making nodes that will continue the tree. (Selectors, simple parallel nodes etc)
- **Service Nodes:**
Works with Composite nodes to register call backs to update the AI's current state, for example if the AI is no longer chasing the target, it's time to start searching again.

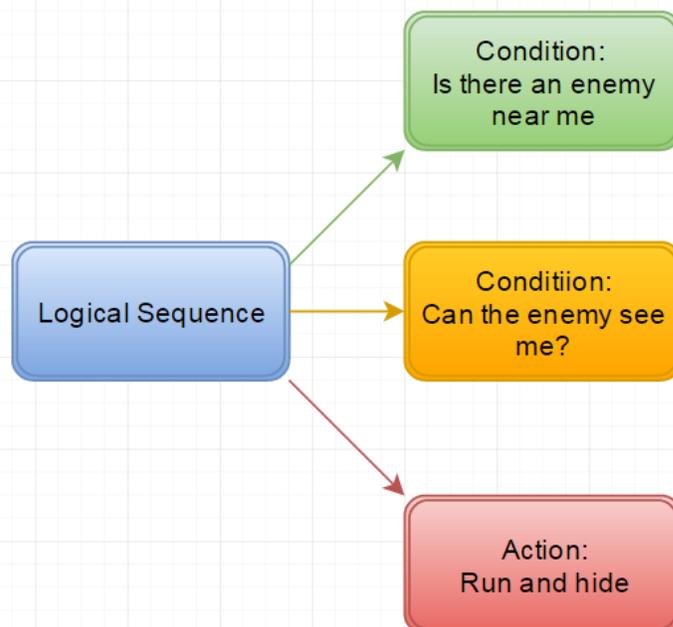
Advantages of using Unreal Engine Behaviour trees.

- Easy to use in terms of functionality.
- Visual coding can make for quicker fixes in syntax errors and bugs.
- Less executing trees due to event driven system.
- Synergy with UE character interface allows for a clear connection between AI construction and NPC character mesh.
- Being a part of the game engine I am building my project on was the deciding factor as it makes perfect sense to use the tools that are in front of me rather than search for an outside source.

As a game, I felt my project was fit into the kind of genre that needed some form of AI attached to it. The main AI NPC I went with in my game was an enemy. It added higher level of atmosphere and pressure to the game.

4.10.3 *The Technical Architecture*

The logical sequence behind the AI interaction is highlighted in the diagram below.



5 User Manual

5.1 Control scheme

W – Move forward

A – Turn left

S – Move backward

D – Turn right

F – Flashlight toggle

P – Pause Menu

Spacebar – jump

Mouse – Control Camera

6 Appendix

6.1 Project Proposal

Gaming as an industry is growing rapidly every year, gaming companies are making immense amounts of money from their IP's and it is truly an inspiring industry where anyone with an idea and some determination can achieve success.

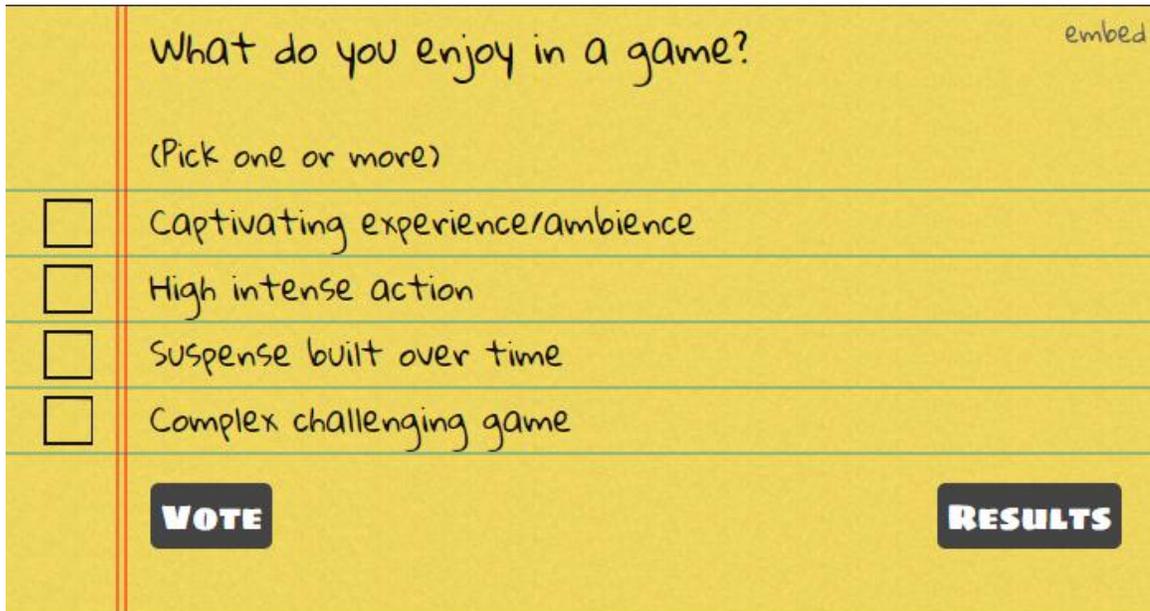
This is one of the reasons I chose the gaming stream of the BSHC course. I have an innate interest in the subject which makes the approach to working more enjoyable.

When starting this project I struggled to come up with an idea that I could find clear direction on so I completed some research on which game genres are the most attractive to users and what I can achieve in the time period.

According to 3 leading industry websites – GameFAQ, TheTopTens and GamersDecide the Action Adventure/RPG genre is consistently located in the top 3 in genre lists. After some more research on genres I decided to settle on an Action Adventure style game.

I feel this genre suits me as it is something that I enjoy playing myself and I know what the industry standard for these games is like. I can compare my idea and soak up ideas from other games that I have played such as the Skyrim and Fallout series made by Bethesda Soft.

As well as picking a genre I had to decide on a path that the project must take in terms of style of game play. I created a small survey online that I distributed between friends and colleagues asking them a simple question.



The image shows a screenshot of an online survey. The survey title is "What do you enjoy in a game?" and it includes the instruction "(Pick one or more)". There are four options, each with a checkbox: "Captivating experience/ambience", "High intense action", "Suspense built over time", and "Complex challenging game". At the bottom of the survey, there are two buttons: "VOTE" and "RESULTS". The word "embed" is visible in the top right corner of the survey interface.

The results showed that there is no one best game style. There was votes on all across the board, personally I enjoy a captivating ambience and some suspense. They did hold a nice portion of the votes so I felt my gut choice had been partially justified.

I feel one of the strengths of this project for me is that as a person who has played games from a young age, I have an excellent resource for in-depth knowledge of games within myself. Having this bank of knowledge allows me to make executive decisions on what would be appropriate for a game and what would be an inefficient addition.

6.1.1 Objectives

I aim to make a game that will create an atmosphere in which players will feel submersed in an adventure akin to an Indiana Jones movie or the excitement of a Lara Croft.

I want the game to look as realistic as I can make it in the timeframe and I want it to feel like anything can happen if one misstep is made.

The game will need to be smooth and using Unreal Engine I think I can accomplish that. Fluid character movements and realistic physics are a necessity.

A third person adventure game would be the official genre of my game, but I may incorporate a first person option depending on how I progress. The object of the game is to explore the level, find your way, try find loot, and if possible escape with your life.

With an adventure game, suspense is usually an important factor, I hope to give the players the need to anticipate what's coming next. I want to add some form of non-playable characters, whether they are friend or foe is yet to be decided. I would like to add some form of adversary for the main character within the levels to add more danger and excitement, maybe some sort of animal or monster that dwells in the levels.

My main objective aside from making the game, is to make something that I can honestly say is a good game. As a gamer I would like to be proud of my creation, this will be a good driving force for me as it's what I would like to do in the future.

As a secondary objective, depending on how I progress with the project I would like to incorporate some branding into my project, such as social media pages, a game website maybe even some 3D models if I have a character to show.

6.1.2 Background

The Game takes inspiration from multiple sources such as films like Indiana Jones and the horror film The Descent and games like Tomb Raider, Skyrim and Fallout.

I'll add some screenshots as to pain a better picture of what I am trying to accomplish.

The developers of Skyrim and Fallout, Bethesda Softworks have made amazing games that I loved and had such great depth with captivating storylines, quests and areas that were designed with great detail.

From playing these games myself I have taken a liking to adventure games and I enjoy solving puzzles within the levels and fighting off enemies to get to the goal at the end.

I feel a lot of these games took some inspiration from how Indiana Jones movies played out and because of their immense popularity I feel that it is a solid idea to go with.

Another inspiration I have is a horror game called Amnesia, it's creepy atmosphere and high intensity horror moments make for an awesome game and I hope to add some of these elements into my own game.

Game Type: Single Player Adventure

The User will play as one main character, there will be a set of levels with some variation of level type such as explore and find loot, escape danger and maybe some sort of driving mission.

I feel that if I have multiple levels with different aspects it will make for a solid project and hopefully an impressive game that players will enjoy.

6.1.3 Technical Approach

I plan on approaching this project in phases.

Phase 1:

Basic level design,. The level design will constantly evolve as I the project progresses.

Phase 2:

Character design and Functionality within levels, ability to use items, pick up loot, use torches etc.

Creating the interface, and game menus.

Phase 3:

Top up detail, potentially add AI characters and cut scenes and have story progression and an ending.

I will be using resources such as Plurasight, YouTube and the Unreal Engine Forums to assist me, as well as my peers. With all these resources I'm sure I will be able to achieve my goals.

6.1.4 Project Plan

6.1.5 Technical Details

Systems used

- Unreal Engine 4.9
- Unreal Engine Blueprints
- Adobe Fuse

6.1.6 Evaluation

I hope to keep to a rigid change management process. I will have version control throughout my project and will test any changes I make as I go along. Primary testing will make sure that the game is working and the core mechanics are running smoothly were as secondary testing will be for visual.

I will conduct some peer testing within the class and from with I will get a good idea of what needs to be improved and also a good opinion on how people feel about the game.

6.2 Monthly Journals

Month: September

Introduction

Being back in college after such a long break is taking some getting used to, I am trying to re-motivate myself and get back into the swing of things. I had struggled to come up with an idea for my project for the first 3 weeks. In the end I decided that if I am going to be putting a lot of effort into something, then it should be something that I like and have a passion for. And with that I decided to make a game that I would enjoy.

My Achievements

I finally came up with my project idea and uploaded my project proposal. I started looking at unreal engine and how my character would be seen in the game. I've created a function that allows players to switch between first and third person.

My Reflection

I find that I'm having a bit of a slow start and I need to work on that. I will try harder in October to get more done.

I'll have to constantly think of additions to my project. I probably will never be satisfied with the quantity of content within the game but I hope to achieve content of decent quality.

Intended Changes

I will try to achieve more of my goals this month and notch off a couple of tasks from my list. I am struggling to find time to work on unreal engine as its not installed on the computers in 3.06 or 3.01 which is where I can study and do work mostly. My laptop cannot run unreal engine so that is also a bit of hindrance and hopefully this can be resolved soon

Month: October

Introduction

This month has been tough in terms of project deadlines, I have mainly been working on things from other modules and have not progressed in any major way in my project. I feel that everything else is getting in the way currently so I have to try dedicate more time to working on it. I have managed to complete my requirements specification. I also had a meeting with my supervisor who was very helpful with setting me on the right track for the requirements specification.

My Achievements

- Successful meeting with supervisor
- Completed requirements specification

My Reflection

I have not been making enough time to work on project and I am definitely falling behind.

Unreal Engine is still not installed on the computers so it is basically impossible for me to work on my project in college.

Intended Changes

I will try to achieve more of my goals this month and notch off a couple of tasks from my list. I am struggling to find time to work on unreal engine as its not installed on the computers in 3.06 or 3.01 which is where I can study and do work mostly. My laptop cannot run unreal engine so that is also a bit of hindrance and hopefully this can be resolved soon.

Month: November

Introduction

This semester has gone by very fast, I've definitely falling massively behind. I've managed to at least create a character with animations. I'm hoping to use December to make some heavy progress and try attempt to catch up a bit with the schedule. I cannot make valid excuses as to why I have been struggling, I have just failed to balance project work with work in other modules and my own job outside of college. The fact that I could not do any work on my project in college until room 3.04 was made available 2 weeks ago also didn't help.

My Achievements

- Created character
- Applied animations to the character

My Reflection

This semester for project has been terrible. At least I'm doing decently in other modules.

Intended Changes

I'll be trying to get a good chunk of project done over the break, Christmas exams may suffer but what can you do.

Month: December

Intro

December was a very busy months in terms of personal life, I was working a lot and spent a lot of time with family over the Christmas period.

This month was a turning point in my project none the less. My character was updated to its final model. It now looks how I want it too and I am happy with my final product. As shown below. I have also begun rigging the skeleton and importing it into Unreal Engine.

I have begun working on meshes that will make up the contents of the map and hope to have the majority of the map completed by the end of January.



My Achievements

Finalized character and skeleton.
Creating meshes for the project.
Acquiring assets for use in project.

My Reflection

After a slow start my project is finally coming along and I'm aiming to have a decent amount done for the midpoint presentation.

Intended Changes

Rather than changing anything I want to continue on a good path of wor

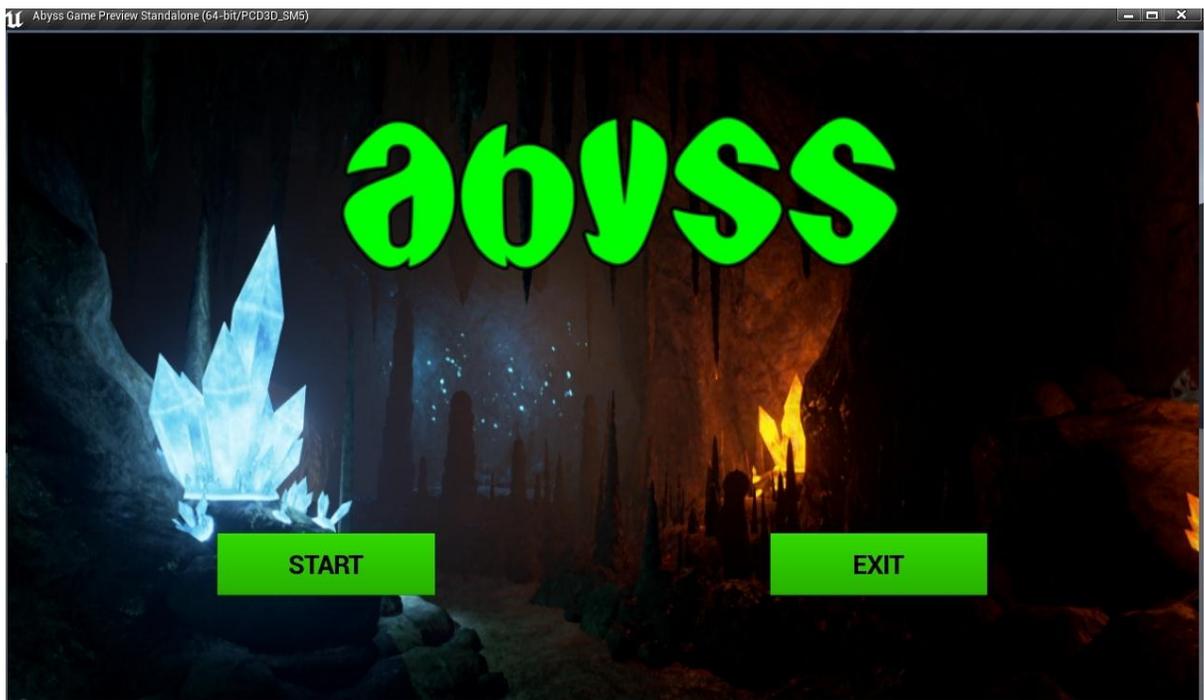
Month: January

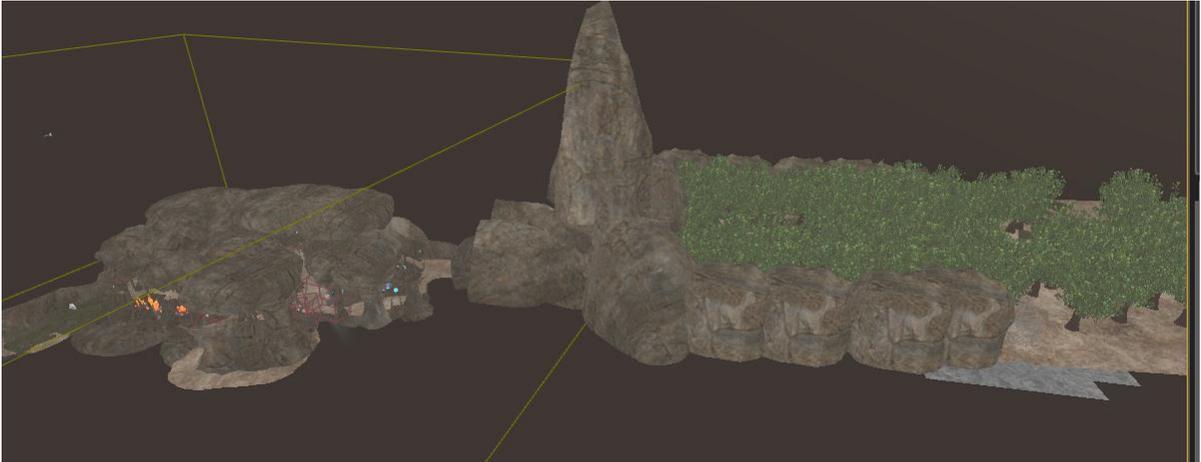
Intro

I committed most of early January to exam prep and completion.

I have continued to make good progress on my project I have worked a lot on my map and I have basically completed a level with my character on the map.

I have worked on some of my requirements like my Main Menu and health system and they are basically completed. I would like to have more at this point but I feel like I can achieve a decent project/presentation and report based on what I have done so far. I still have a lot of time to continue working on my project and I will continue to try and improve it in as many ways as I can.





My Achievements

- Created a playable level.
- Created a functioning main menu.
- Added a visible health system.

My Reflection

Pretty worried about the midpoint presentation in terms of the amount I have completed but this is not the final product so as long as I can voice my plans and show what I can achieve in the future I feel it can go well.

Pretty solid month in all in terms of productivity.

Intended Changes

Continue on after mid-point presentation and add more features and complete all of my requirements.

February Intro

This month I managed to create a form of adversary for the player, I intend to add AI to this enemy to make the game more challenging. This character is very fitting for the map and I really like this addition.



I have created a working pause feature so the player can stop playing and resume from the place they paused. The user can also quit the game from this menu.

I also had my mid-point presentation. The presentation itself went fine, my report seemed to be lacking though. I am not unhappy with my grade as I do think it is fair, I just would have liked to get higher so I will use that and try get some advice from my supervisor on ways that I can up my score for the final report.

My Achievements

Created an NPC enemy character.

Created a functioning pause menu.

Successful Midpoint.

My Reflection

Looking forward getting working on more functionality and improving my project all round.

Intended Changes

Continue on a good path.