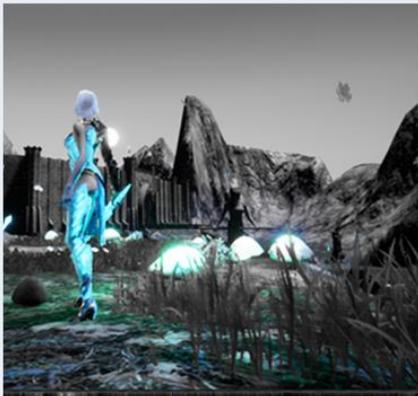


WINTERSPRING

Modular 3: Game and Character Design Fundamentals

20th December 2019

Eve Ryan



Project Investigation

I was given a brief at the beginning of this project to create a playable level within Unreal Engine Four, to develop my skills within level design and code within the engine. Unreal Engine is a game engine developed by Epic Games, which was first used to show the 1998 first-person shooter 'Unreal'. It was developed for first-person shooter games in the beginning however recently it has been used for creating games of other genres, such as platformers, fighting games, MMORPGs, and RPGs. It uses the code C++ and is used by many due to its free availability. My final outcome is a game centered around my character Aurora; whose soul was stolen by an evil ruler, therefore she has to reclaim the parts of her soul which are frozen around the map and return them to her body at the end so she can be alive again. I thoroughly enjoyed this brief and I found the research fun as I had free reign to create whatever I wanted within the software. I chose my own story, theme, objects and map. I also asked around my peers who are notorious for game playing and they all really enjoyed the story and visuals, which is perfect as people aged between 15-25 are my target audience for this game.

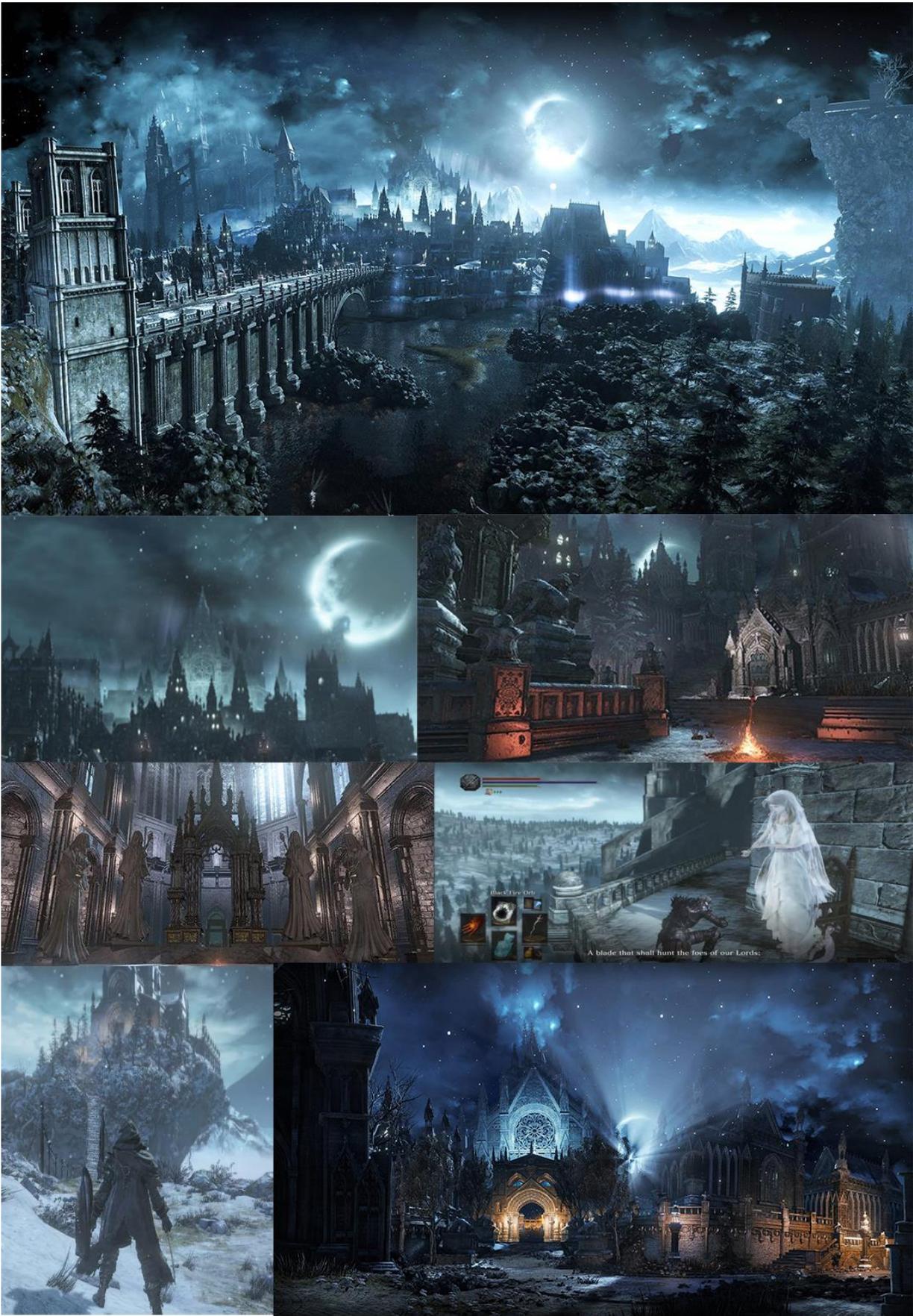
Ideas Generation & Research

I generated many ideas throughout this project, with the help of researching other game environments and themes. The very first thing I did at the beginning of this brief was create mood boards, one revolving around a game which I immediately wanted to take inspiration from, and the others themed around other games which I really enjoy the mood and themes of. These mood boards really helped to spark creativity, and piece together all my ideas based around forests, snow and ice.



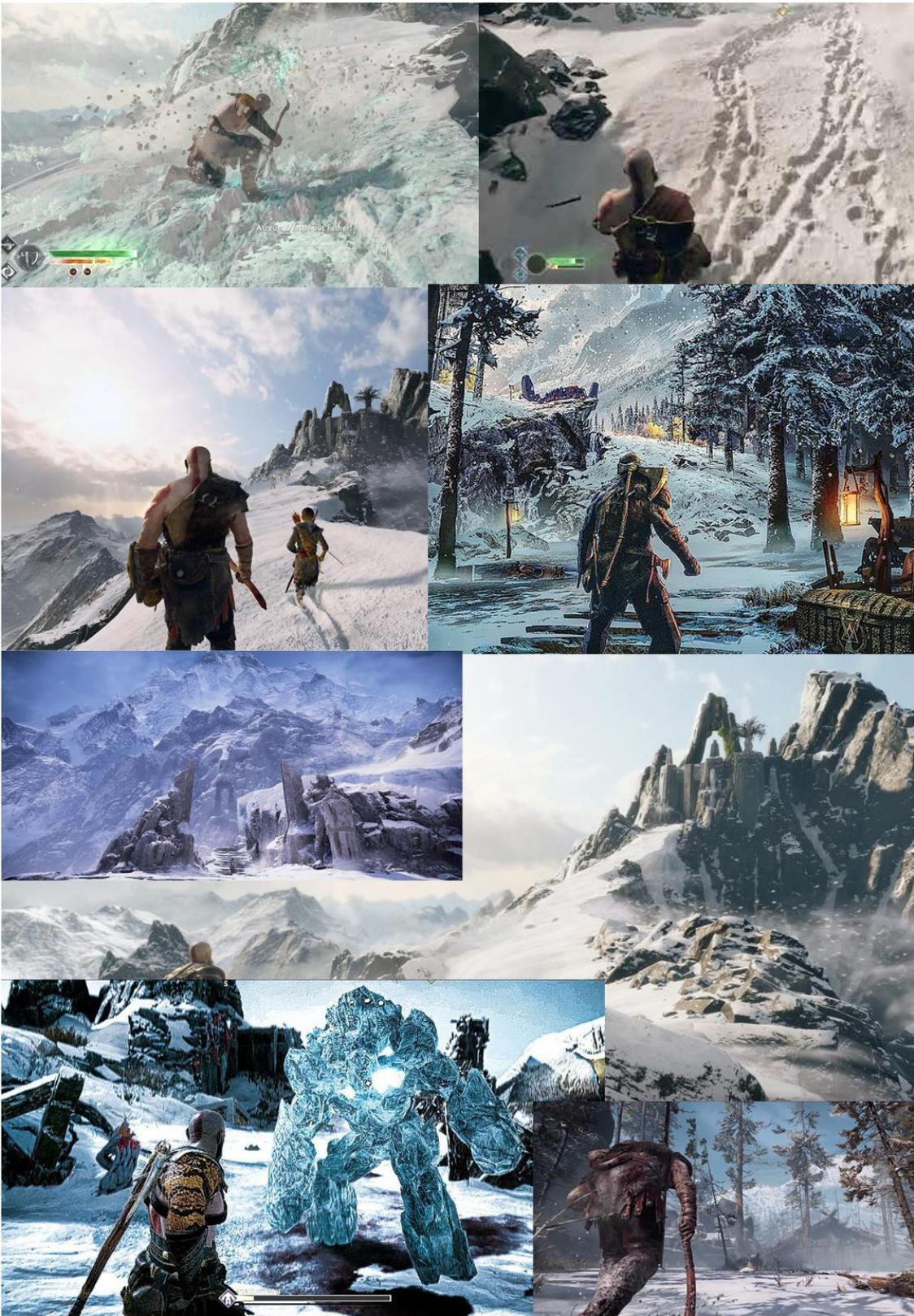
Returns)

(Alice Madness



(Dark Souls)

III)



(God of War)



(World of Warcraft: The Wrath of the Lich King)

After looking through all the free texture packs I had within Unreal Engine, the snowy ruins and trees really sold my idea for the snowy climate, as well as the magical forest climate. The Warcraft, God of War and Dark Souls III game design is some of the best game design I have ever seen for cold climate and the attention to detail is beautiful. As you will see in later screenshots of my game, I have clearly taken inspiration from my research from these games.

I developed ideas for the technical side of my game too. I play games regularly but decided to play some of my favorites again to see why, on the technical side, I enjoyed them so much. The first thing I noticed was I am a very lazy runner in games, and where I could not incorporate a fast mode of transportation apart from running, I decided to use teleport points as they are easy to code. There are teleporters in World of Warcraft: Legion in which you could travel to any teleporter from any other one. There is also a function in where you can travel back to a main location from a whistle which has a cool down, however I found this unnecessary for my level as it only takes 5-10 minutes to complete.





Flight Master's Whistle

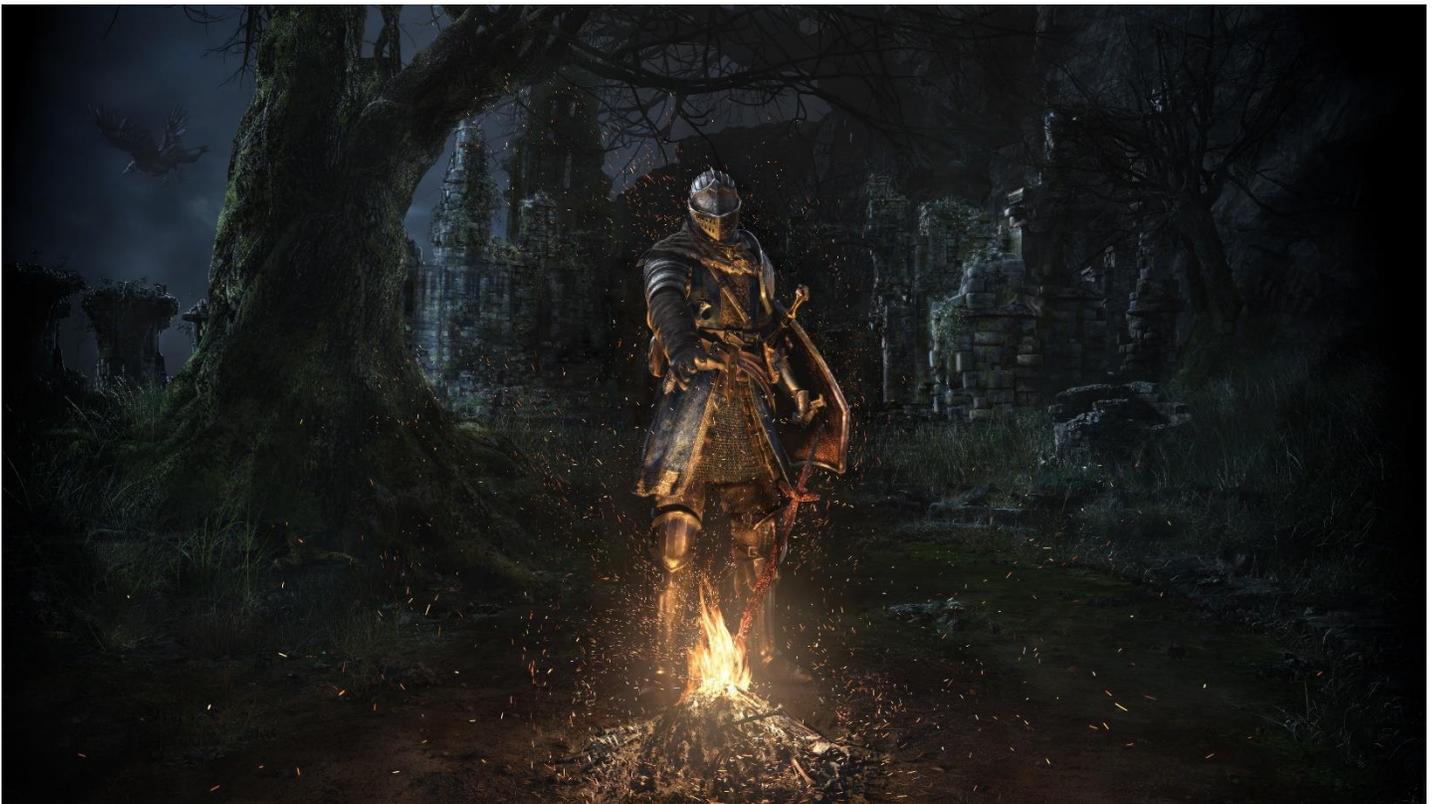
Item Level 1

Binds when picked up

Use: Request a pickup to the nearest flight master.

Only usable in . (5 Min Cooldown)

Another example of teleportation in games which I think is extremely useful are the bonfires in Dark Souls. When you rest at a bonfire you take no damage, and heal to full health, however this would be unnecessary in my game as there are no enemies. The bonfire allows you much like in Warcraft to travel to any other bonfire within the game (which you have already discovered). There are a lot of bonfires in Dark souls, however my map is so small two is enough to get around if you miss an item and need to travel back over a lake etc. I chose to use rune-stones instead of Alien Teleporters or Bonfires, and they glow blue which matches the aesthetic of magic and ice in my level.





(Dark Souls III and Dark Souls II)

I also really enjoy exploration in games, I find what links my favorite games together are the ease of transportation, gorgeous visuals and exploration. It feels like you can immerse yourself way more in the game where you can travel where you want to and go at your own pace without being forced into certain doors. A good example of this is Resident Evil 7: Biohazard. You're in a small enough map to not be wandering for hours, however you can go where you want in a giant farm mansion (and eventually outside) as there are many floors, and many objects to find, so you take your time exploring but aren't getting frustrated with how far you're travelling.



(Resident Evil 7: Biohazard)

However, as well as looking at my favorite games and what made the levels fun, I will also talk about some games I found extremely dull and hated. Unlike Outlast (another great exploration game), Outlast 2 let every player down. The map was all dark, and way too big. At no point did I know where I was going, where I was meant to be going, who was chasing me and why, if I was going in circles or if I had already been somewhere as every building and area looks the same, and are miles apart. This is the best example of what to avoid, as, as a player I thought it was boring, and I didn't care about the story or completing the game in the end.



(Outlast II)

I decided to conduct some of my own research into using Unreal Engine itself. I watched a lot of YouTube videos during the process of creating my level and read through many links which my tutor gave me. Here are the videos I watched and followed when creating my level if I was stuck or needed to research features.

UE4 Tutorial - Audio (Trigger Activated Sound)

<https://www.youtube.com/watch?v=YcEzT3-hAmk>

How to play videos in widgets | Unreal Engine 4 Tutorial

<https://www.youtube.com/watch?v=oguiuWgCCoE>

Unreal Engine 4 - Official Death Screen Tutorial (Beginner)

<https://www.youtube.com/watch?v=68KSATgkt6k&t=501s>

ABK Tutorial Series - 23 - Custom Animals

<https://www.youtube.com/watch?v=2NK4tCphraw&t=1271s>

Unreal Engine 4 " EASY AI ROAM WITH PEOPLE AND ANIMALS WITH DEATH ANIMATIONS!

<https://www.youtube.com/watch?v=5eRBx17733s&t=993s>

Creating A Main Menu - #51 Unreal Engine 4 Beginner Tutorial Series

https://www.youtube.com/watch?v=U_DY1lmHbLc&t=328s

Styling our Main Menu - #52 Unreal Engine 4 Beginner Tutorial Series

<https://www.youtube.com/watch?v=xAL3U05M1kM&t=1s>

Unreal Engine 4 - Loading/Opening Levels

https://www.youtube.com/watch?v=N_q8jTs8gUo&t=39s

How To Create A Main Menu - Unreal Engine 4 Tutorial

<https://www.youtube.com/watch?v=ulUO4EN8BG8&t=2s>

Adding Main Menu Functionality - #53 Unreal Engine 4 Beginner Tutorial Series

<https://www.youtube.com/watch?v=1PeA1hdSU14&t=489s>

How to Set Up a Teleporter in UE4

<https://www.youtube.com/watch?v=78GiRnzZojA>

UE4 - DYING AND RESPAWNING

<https://www.youtube.com/watch?v=gVdPRJVfUuM&t=97s>

Unreal Engine 4 - Object rotation Tutorial

<https://www.youtube.com/watch?v=D1pQrctHqEM>

Unreal Engine 4 - Quick! How To: Scale Materials

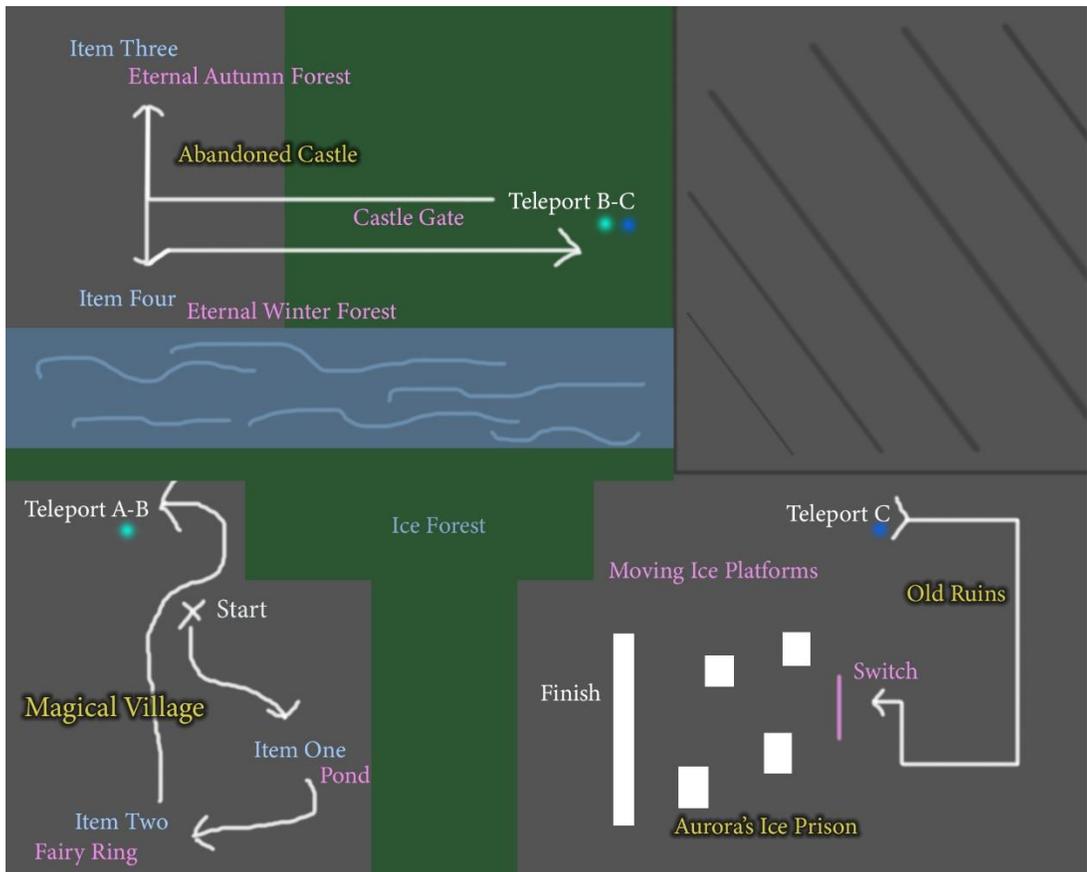
<https://www.youtube.com/watch?v=DXK1pdvBUAM&t=2s>

Unreal Engine 4 Tutorial - Collision (Quick & Easy)

<https://www.youtube.com/watch?v=XLvJnFuk3Cc>

Pre-Production

After I had an idea in my head of what kind of game I wanted, it was time to create a floor plan. I did end up changing floor plans as shown in my development logs below, but I ended up scrapping the idea completely after I had played around with Unreal Engine and knew what was within my reach and what was not. Here is the rough floor plan for my final game I made involving a castle, ice prison and magical forest/village, which I ended up sticking to throughout my project.



My final character I decided to use within my game was Paragon Aurora as she fit the theme of my game perfectly. I built the majority of my game based around sights where she looked like she fit in, as it is hard to make such a stand out character like Aurora blend in.



The texture pack which I used mostly was the Infinity Blade: Ice Lands and Grass Lands. These texture packs contained a lot of structures and assets which fit very well into my game, such as all the parts for my castle, ice kingdom and ruins.

<Back **CONTENT DETAIL** Home Browse Industries Free On Sale Submit Content Help Search Products. Q

Infinity Blade: Ice Lands

Epic Games

★★★★★ 180 10 reviews written | 0 of 1 question answered

Infinity Blade: Ice Lands is the wintery fort set deep within a glacial enclave. It's one of three environments that ship with the Infinity Blade Collection, a massive suite of content designed for high-quality mobile experiences.

Add To Project

OR

Write a Review

Supported Platforms

Supported Engine Versions

4.9 - 4.23

The screenshot shows the 'CONTENT DETAIL' page for 'Infinity Blade: Ice Lands' on the Epic Games Store. The page features a large image of a wintery fort, a star rating of 5 stars with 180 reviews, and a description of the environment. There are buttons for 'Add To Project', 'Write a Review', and information about supported platforms and engine versions (4.9 - 4.23).

<Back **CONTENT DETAIL** Home Browse Industries Free On Sale Submit Content Help Search Products. Q

Infinity Blade: Grass Lands

Epic Games

★★★★★ 460 18 reviews written | 0 of 2 questions answered

Infinity Blade: Grass Lands is the earthy citadel adorned with stone set pieces and beautiful props. It's one of three environments that ship with the Infinity Blade Collection, a massive suite of content designed for high-quality mobile experiences.

Add To Project

OR

Write a Review

Supported Platforms

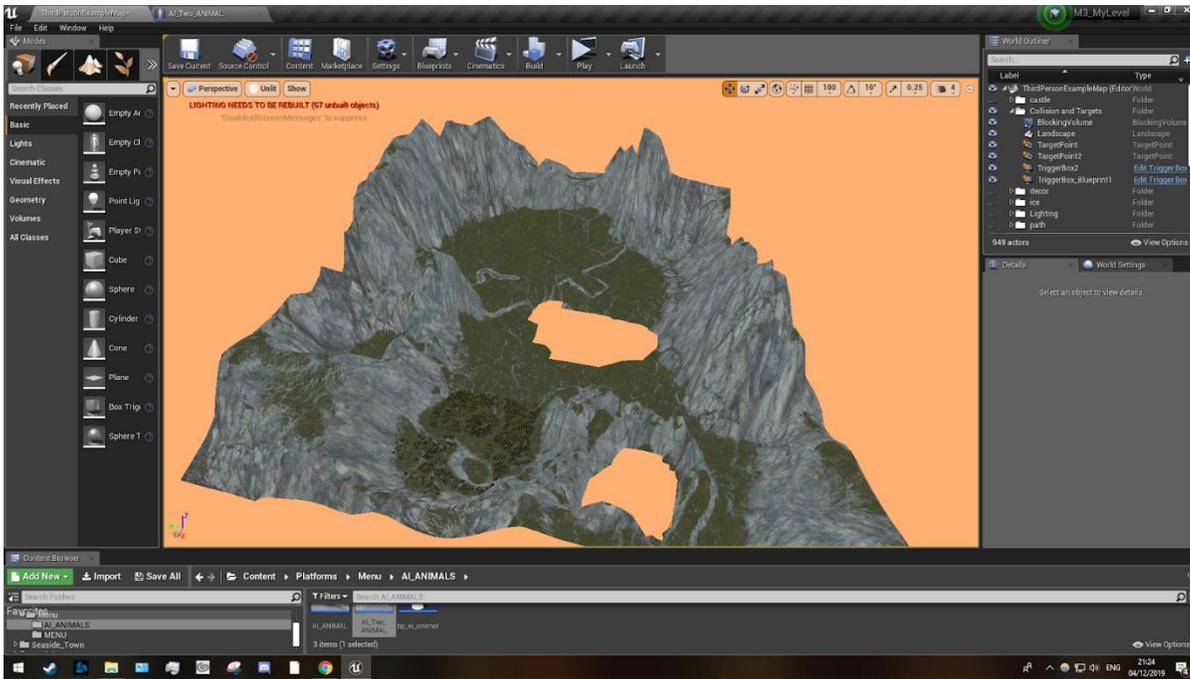
Supported Engine Versions

4.9 - 4.23

My Development Logs

In this blog I will be taking you through everything I have completed so far for Modular 3: Game and Character Design Fundamentals.

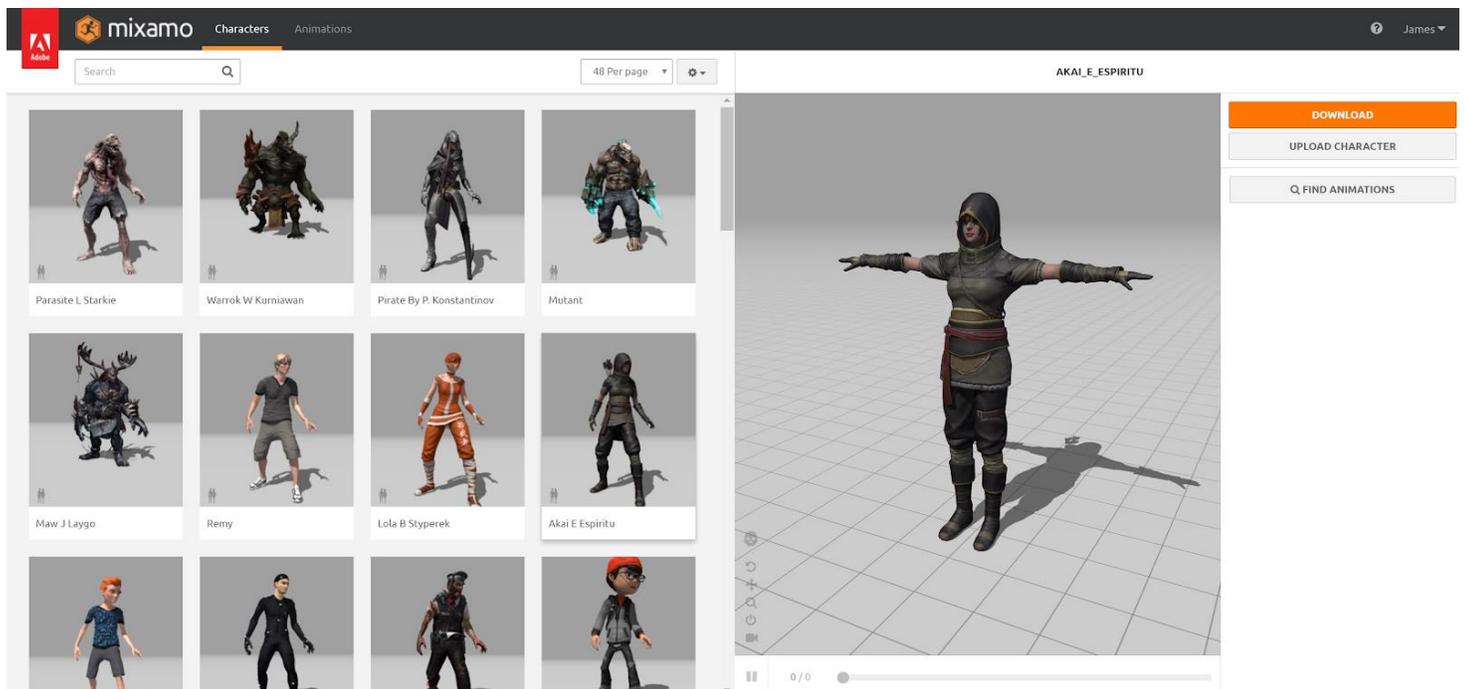
We began using unreal Engine in our first lesson. Firstly, we had to create a floor for where our character would exist. We searched for a texture which we downloaded for free on the unreal shop, and placed it into the world through the materials tab. I chose a grasslands texture. After the ground turned from grey into grass, we used the 'Sculpt' tool to bring in some mountains, so our floor no longer had a horizon, as this is bad in games, due to the fact the player would fall off of the map. After messing around with the different sculpt and flattening tools I had the mountain range I desired.



After this, we were tasked to make a game menu, with two level buttons, which would successfully take our character to level 1 and level 2. To do this, we created a widget blueprint, and named it 'My Menu'. We created two new buttons which I named Level One and Level Two. I selected 'On Clicked Event' in the panel which says 'Details'. Over in the blue print section, I selected the graph tab through the Designer drop down menu and created two event dispatchers and named them 'Level_1_ON' and 'Level_2_ON'. I then called the event for each dispatcher in the event graph. After this I opened the level blueprint and added the event 'BeginPlay' which would start the event. I then created a new widget and an 'Add to Viewport (With the target being User Widget)'. I then made sure we could see the mouse cursor and that the controller would connect to the game. I then continued to code until the events were active, and connected the execution lines, ensuring they were correctly linked. After this was complete, I had to set up the Persistent level system. In the editor viewport, I created two new levels within the levels tag. I placed a cube in level one and a cylinder in level two, as indicators that my code had worked. I also added a target point to each scene. I went back into the blueprint and copied and pasted the 'Load Stream Level' and 'Unload Stream Level' so that the levels would run to the corresponding buttons. Finally, I hid the mouse cursor during the level so that the player would not see the mouse after the menu screen. I wanted to change the spawn location for the different characters, so I changed the target point within the blueprint. I did this by adding 'Set Actor Location', 'Get Player Character' and 'Set Actor Location' along with the desired coordinates.

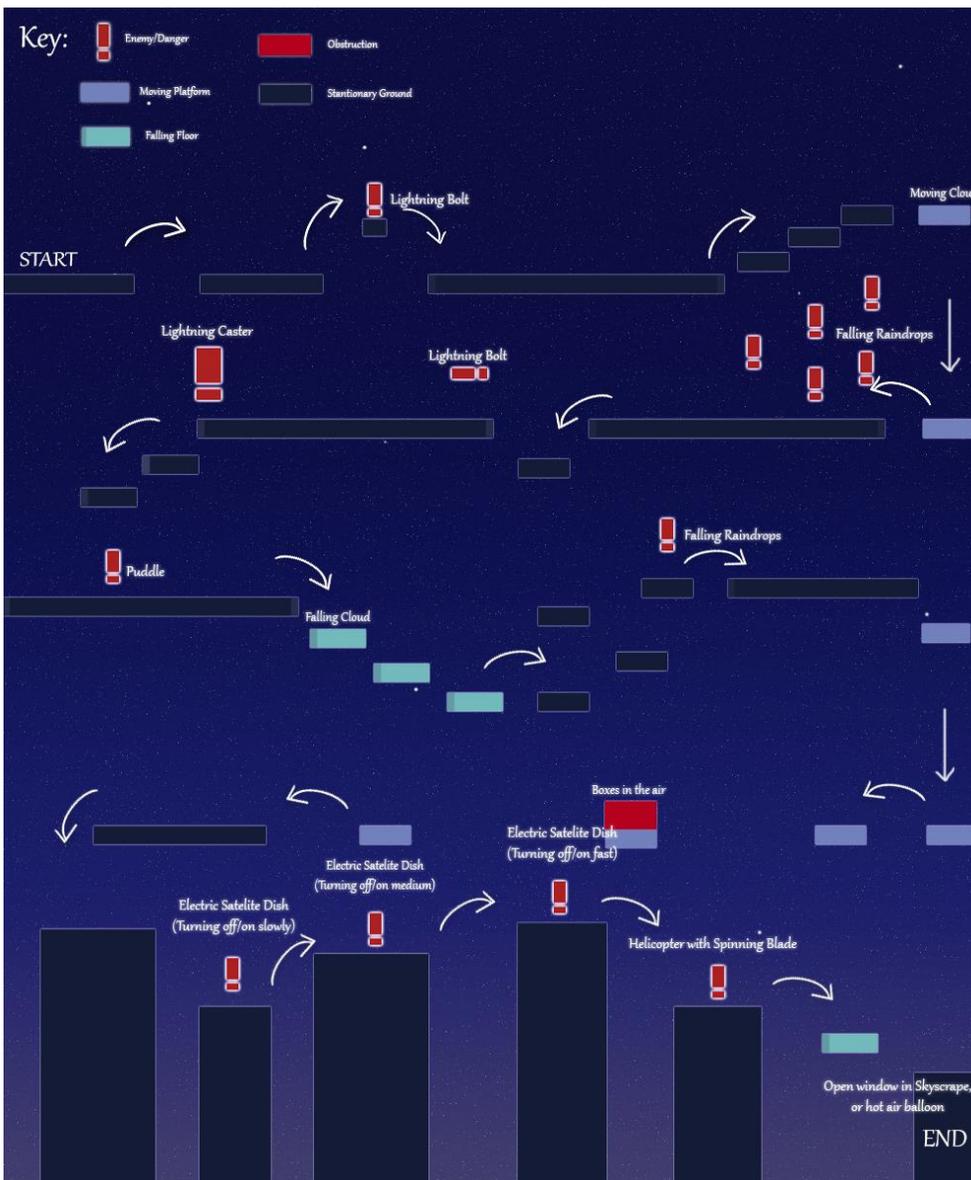
Implementing Characters

After this task was complete, it was time to implement a character into the game. We used the website 'Mixamo' to download a preexisting character design and imported it into Unreal Engine. I chose a basic female character and mapped the character texture onto the mannequin within Unreal engine. The process of this is as follows. In the content browser I created a new folder named 'My Character' so that I didn't lose my blueprints. I imported my downloaded character (as a T-Pose format), which created the textures for my character. I also downloaded a running and walking animation to then apply to my character. I imported these downloaded animations from Mixamo into my new folder, and made sure to untick input mesh, and select the skeleton mesh that belongs to my imported character. I then duplicated the Third Person Character blueprint which already existed and pasted it into my new folder. I renamed it to my character's name. I then created a Blend Space 1D for the first time. I opened this and navigated into the graph editor. After this I made an Animation Blueprint and made a new state machine called motion. I made one called run and one walled walk and opened these in the blend space I created from the asset browser. I played around with my characters speed and the blending of the running to walking animations. I then finalized the blueprint, so everything compiled correctly. I also went into my character's blueprint and added a node so that when shift is held her run speed increases.



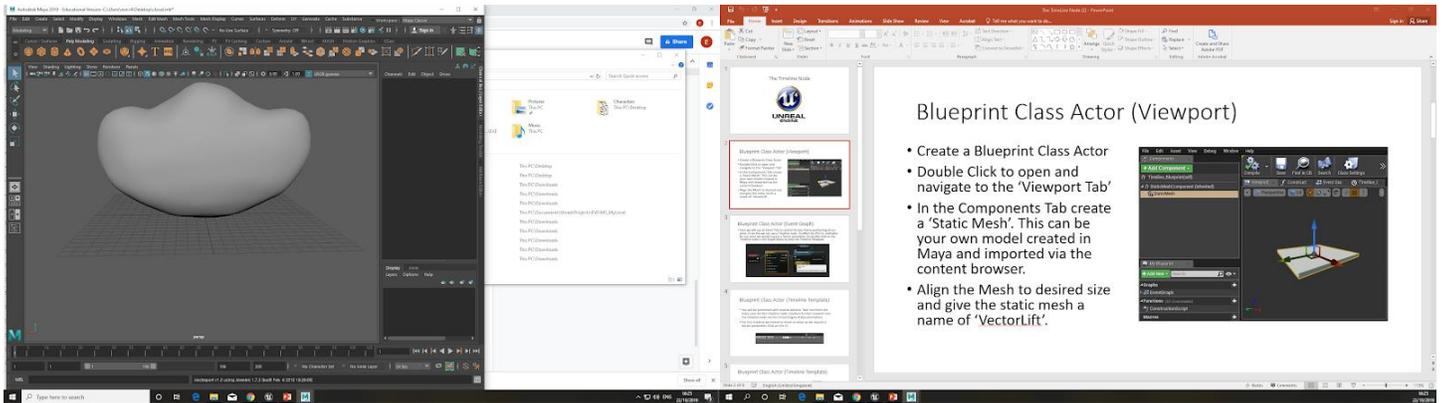
Creating a Floor Plan

Next lesson I decided I wanted to make a floor plan for what would be my 3D platform game. My idea was the character begins up in the stars and clouds, and gradually travels closer to the ground (ending on top of a skyscraper) The color scheme features blues purples and yellows. Here is my rough floor plan

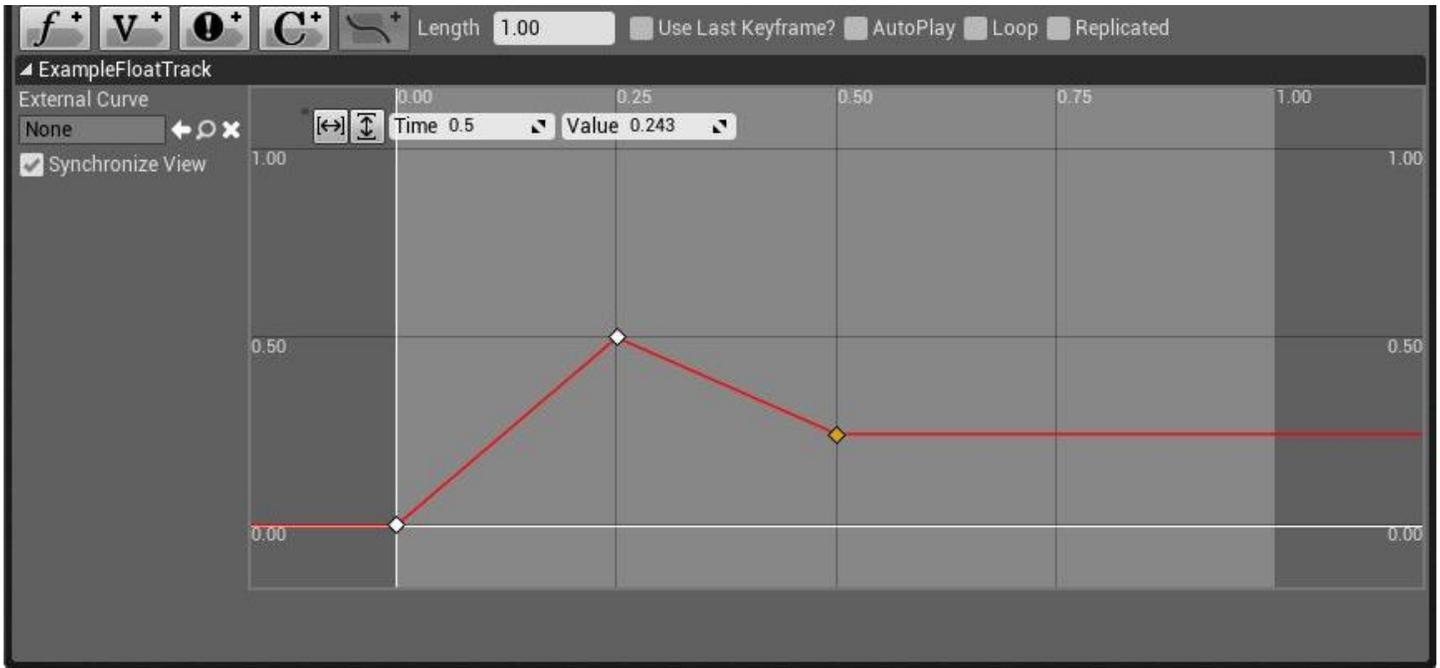


Moving Platforms

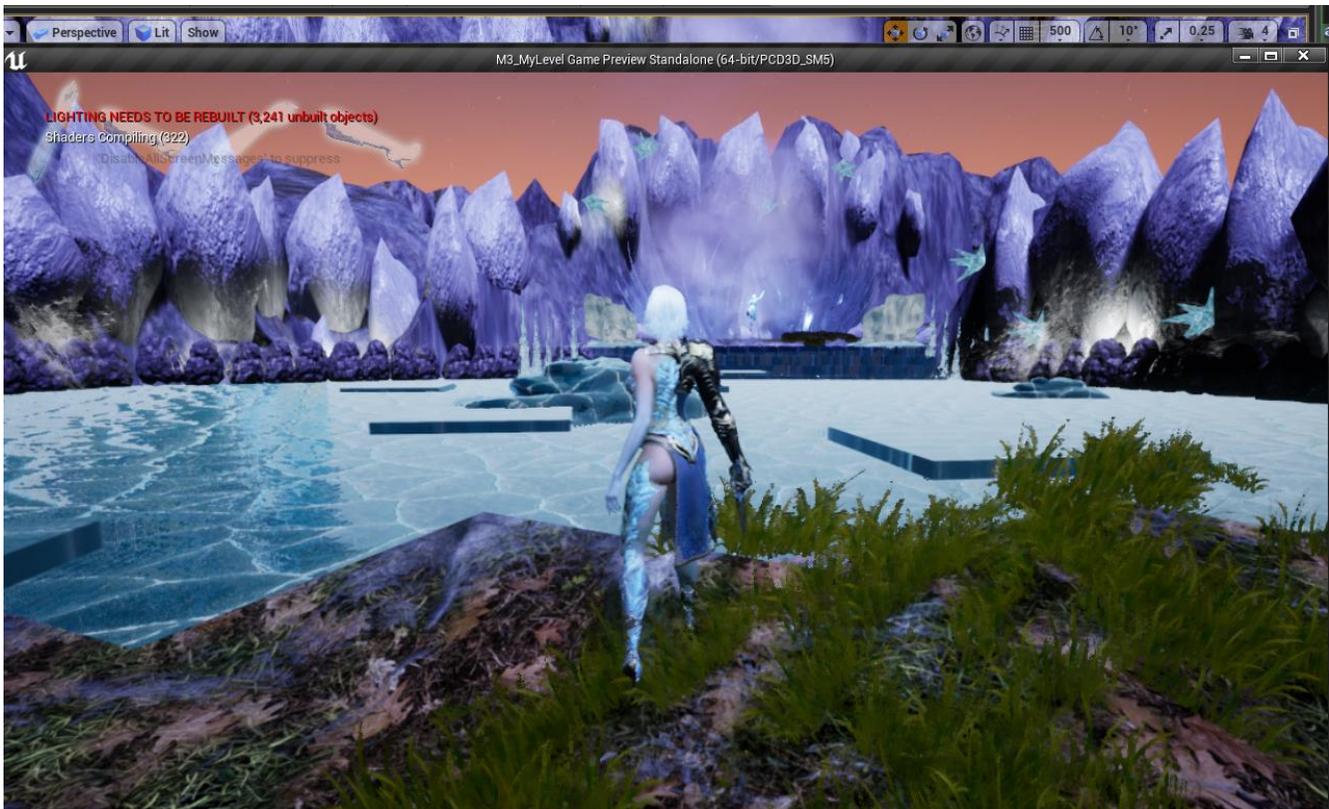
I really want my moving platforms to be clouds, so I decided to sculpt one in Maya. I then decided to try and make a moving platform in Unreal Engine and use the cloud as my moving platform in my Unreal Game.



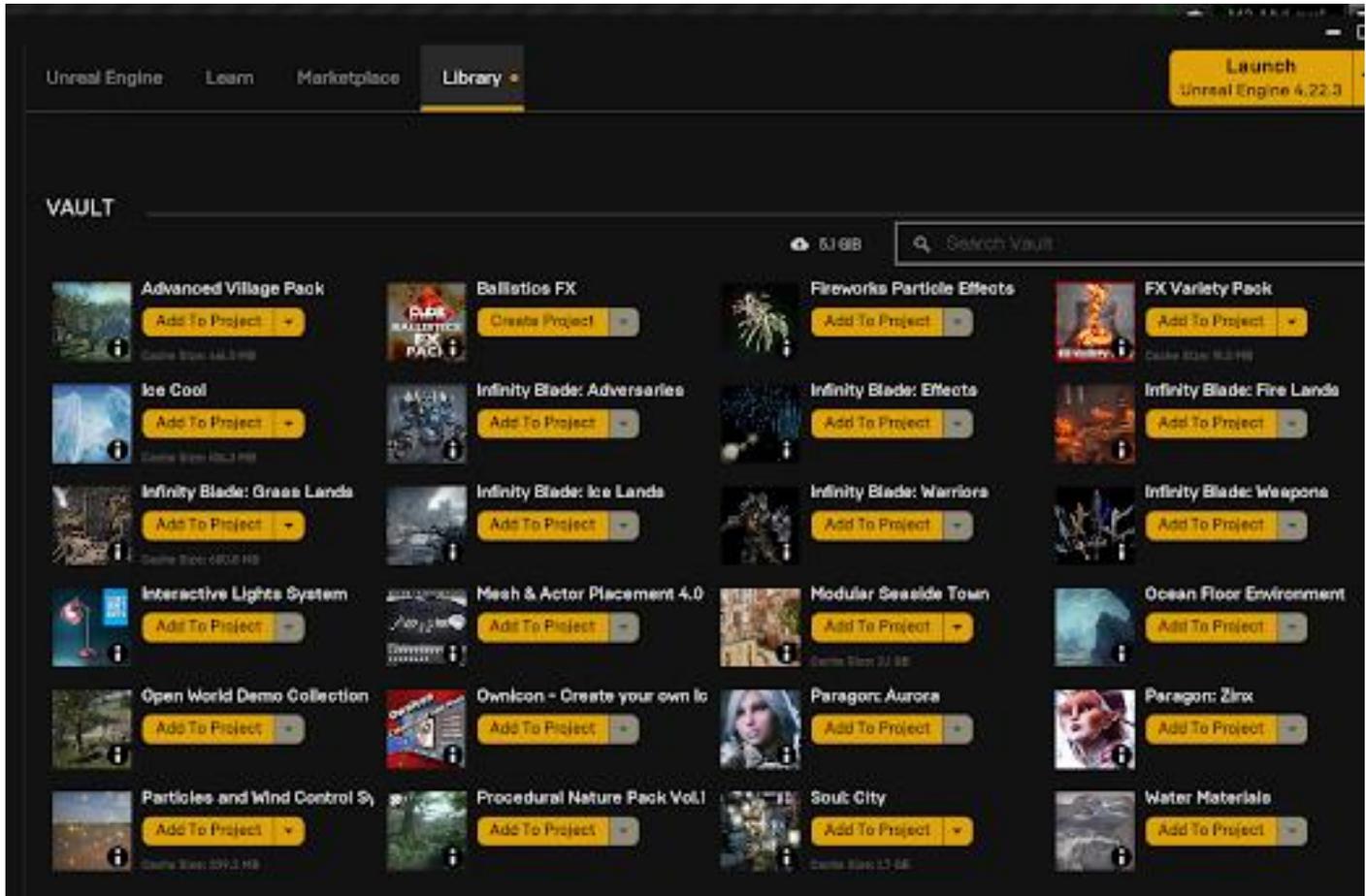
To do this, I first created a Blueprint Class Actor. I navigated to the Viewport Tab and created a static mesh, using my cloud which I imported through the content browser. I made my cloud the correct size and named it 'Cloud'. I used an Event Tick to control the 'Key Frame Positioning' of the cloud. I used a timeline node to do this, which controlled how fast my platform would move and how far/high it would travel. I did this by creating a vector parameter in the Timeline Node and locked the axis I did not want to be affected. I made sure to tick the loop box so that my cloud would constantly be travelling. I set the altitude and timer within the node too. After this I completed the blueprint script and make sure my nodes all were executed correctly. I tested this worked in game and checked the height and speed. I made sure to also add a pause in the timeline node so that my character had time to jump onto the cloud before it ascended.



After this however I decided to scrap the ideas of moving clouds in the sky, as from playing around more in Unreal and thinking more about the theme I wanted to go with, I scrapped the clouds and the city and decided to begin further research. I ended up still using moving platforms, but I instead make many in different directions, ice blocks floating over a frozen lake which is the final stage of my level.

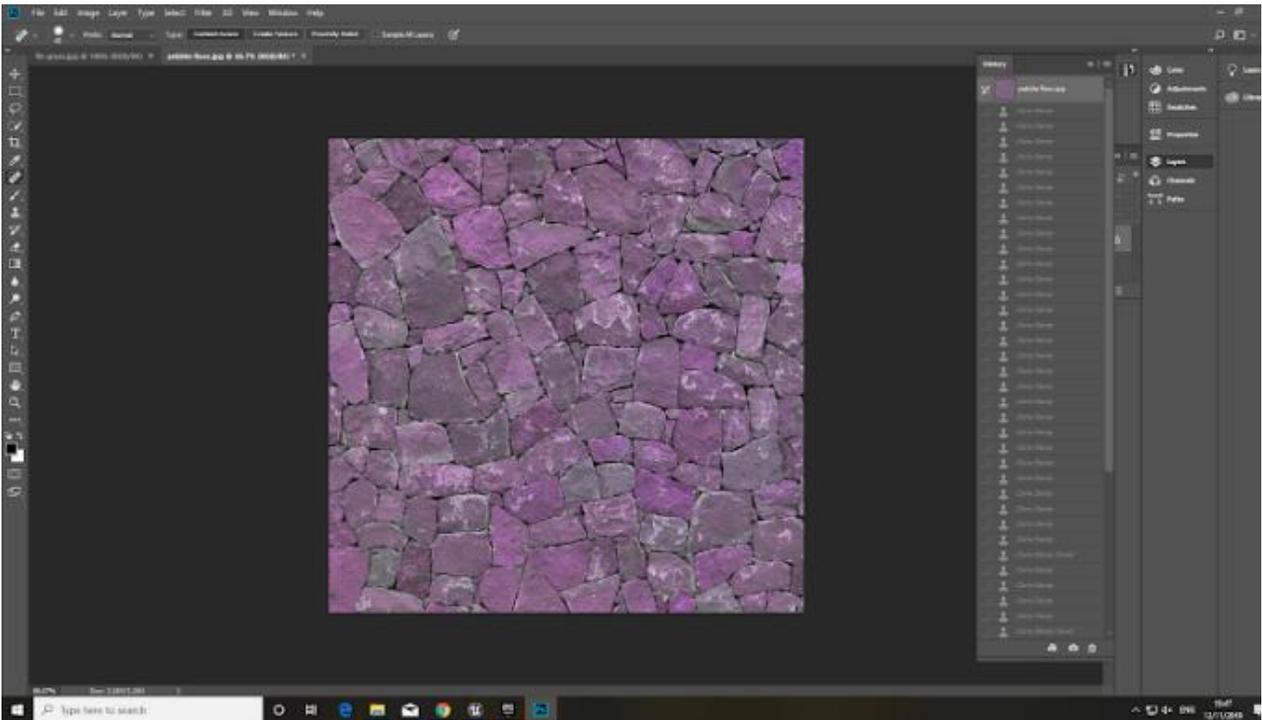
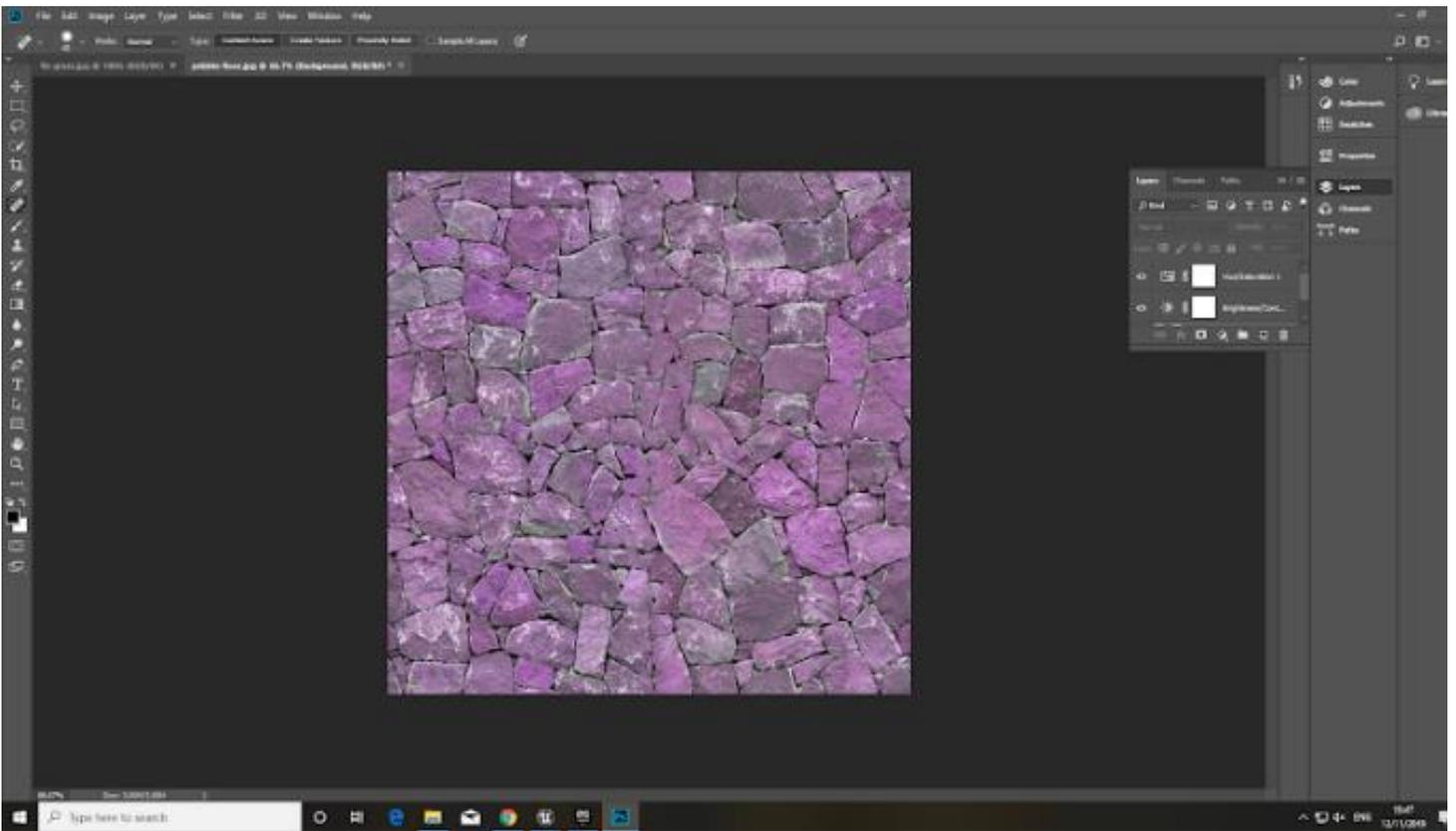


Making a level from sky to ground was way too time consuming and hard to make for a first level attempt, so I moved on to thinking of new ideas. I downloaded a few different texture packs for free off the Unreal Engine Library to see what I was working with. I don't mind having to make a few meshes in Maya, but I want to use preexisting meshes so it is easier for me to create a level. These are the packs which I have downloaded.

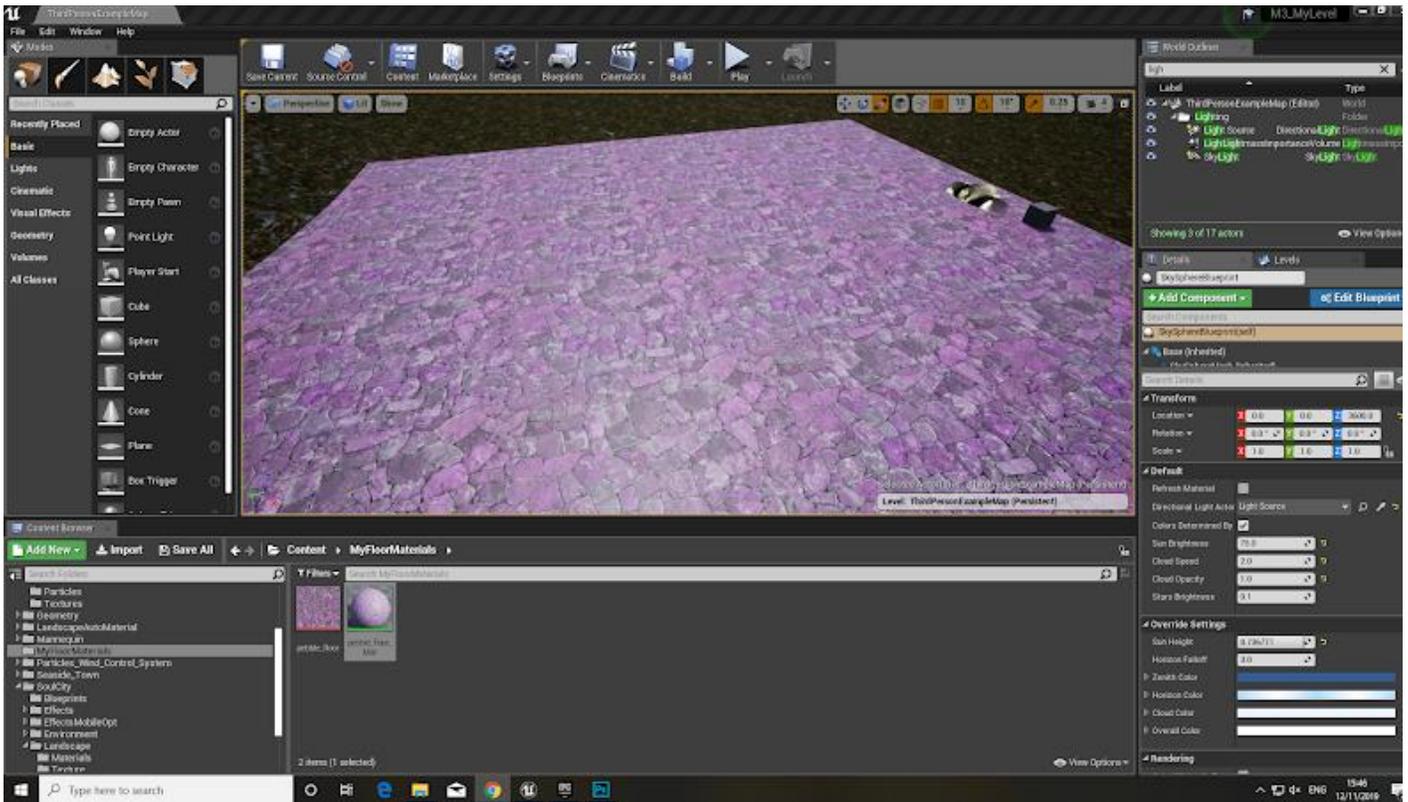


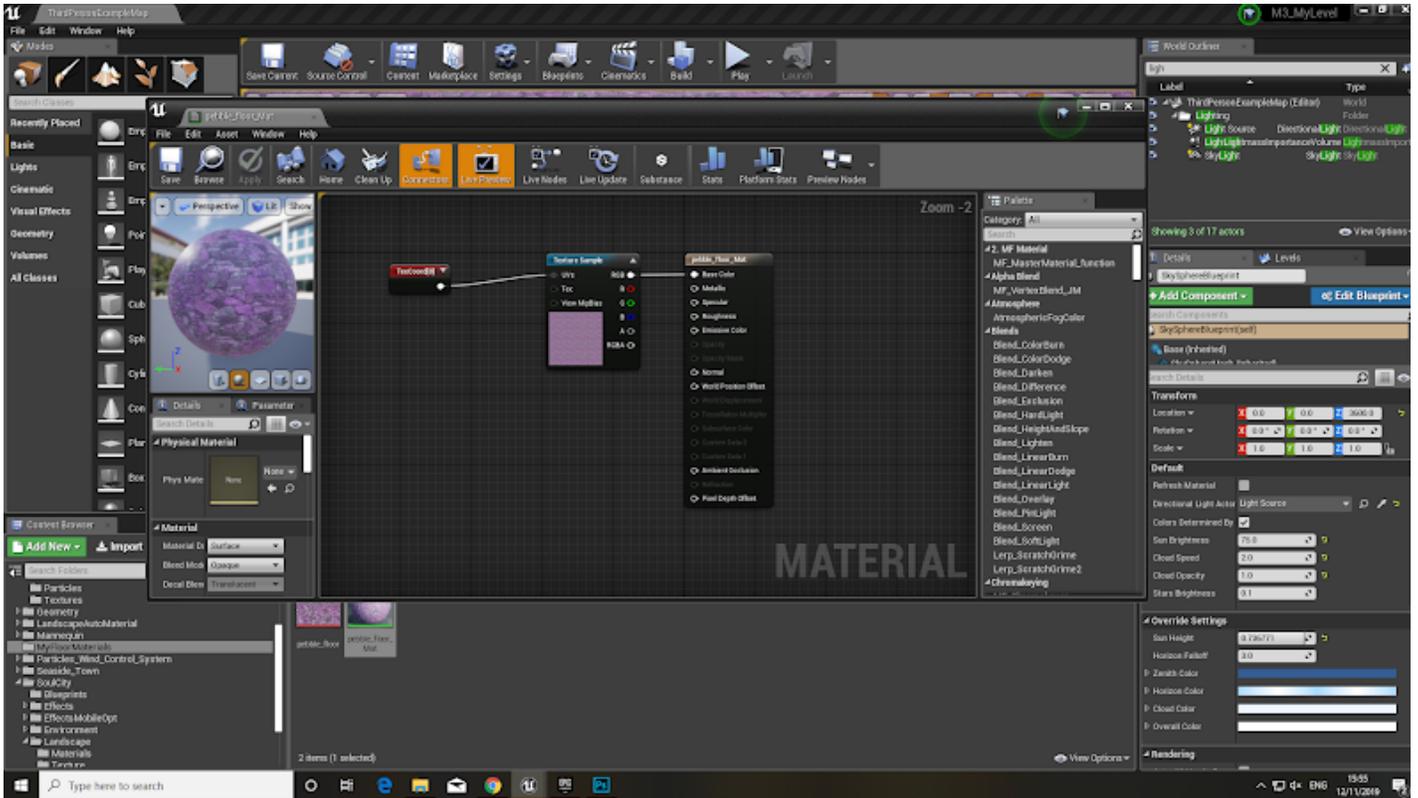
The story behind my new level isn't completely figured out yet, however I think I want to go with something along the lines of a land which has been cursed by some sort of evil being, and the protagonist needs to collect some different objects for a spell to break the curse. It is not polished yet, but I have an idea of the visual aesthetics I want. I decided to try and create my own floor texture, as the magical part of the land I want to be purple and glowing. Even if I couldn't get it down into my level, it was really good practice for learning about textures and how they work.

I found a picture of some cobblestone and coloured it purple using photoshop. I then flipped the image horizontally and vertically from the midpoint so that the corners would match up when placing them as a texture on the floor. I then used the clone stamp tool and the spot fix healing tool to make sure the clashes of stone in the middle blended where the axis had been flipped, as shown below.



After this I decided to test it out in Unreal Engine. I created a plane class actor and set that aside. I then made my floor image texture, before applying it to the plane's empty material slot.





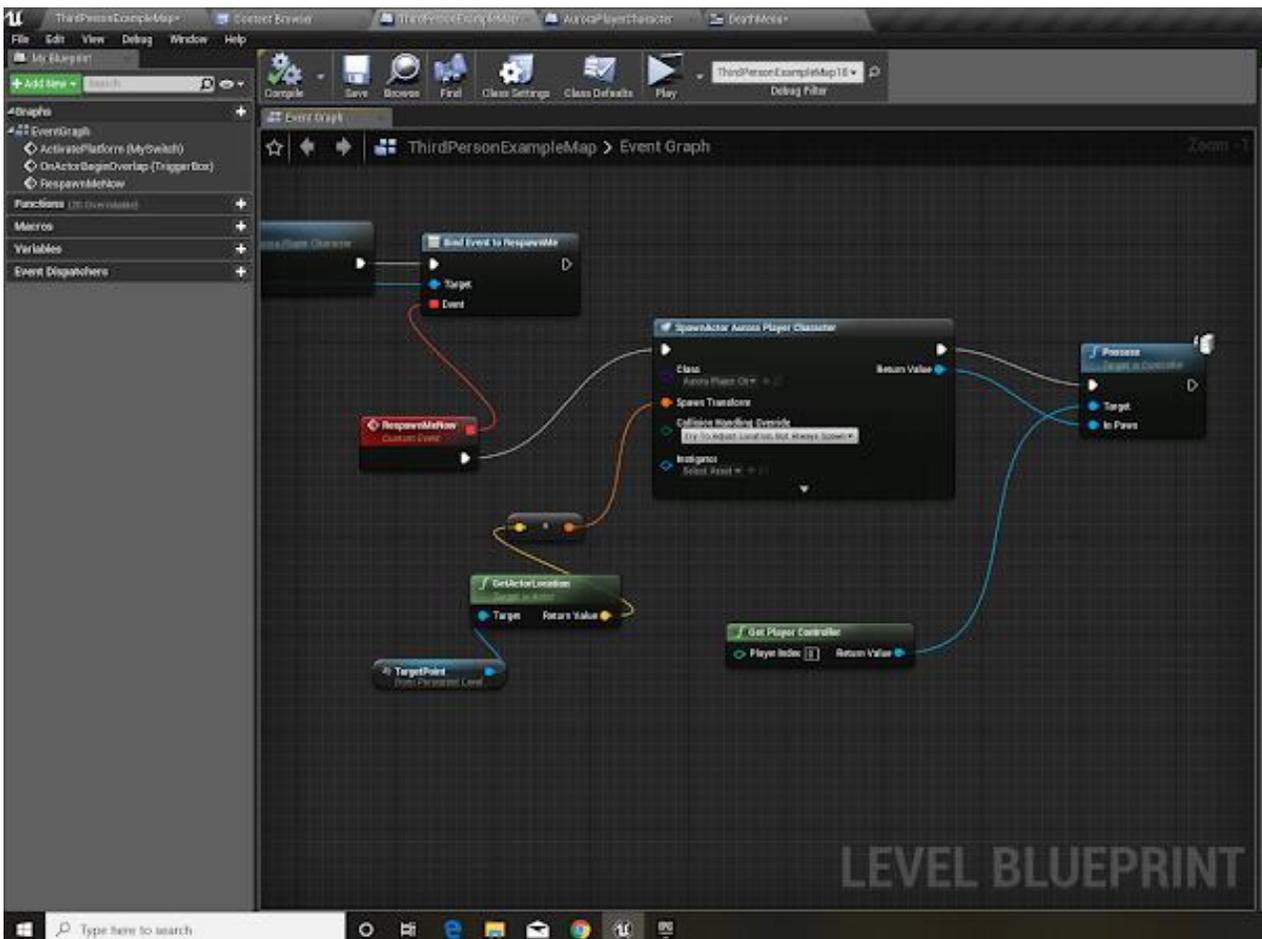
I really like this floor texture; however I think I need to make it a little darker and add some sort of leaf/flower overlay on top of it so that it doesn't look so clean and polished. I ended up not using this floor for anything, however when building my castle, when I tried to lace the brick textures onto my cube, the bricks were way too big and not realistic, but luckily I had learnt how to scale textures so now the bricks on the castle are a realistic size.

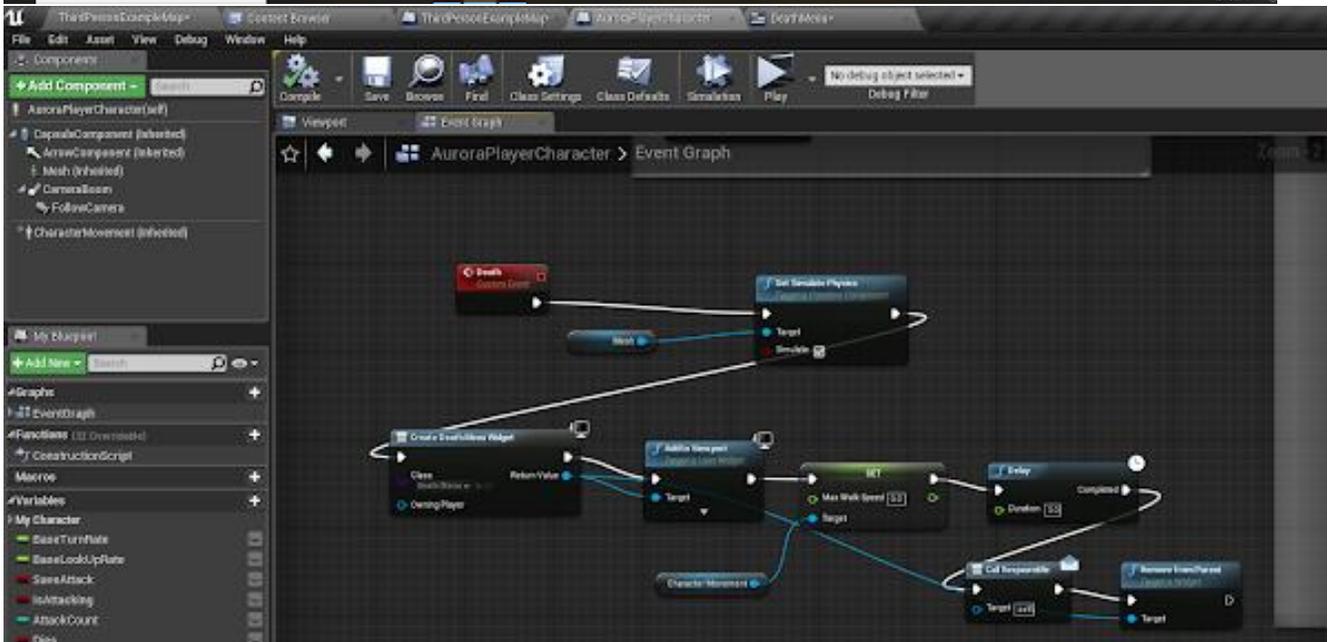
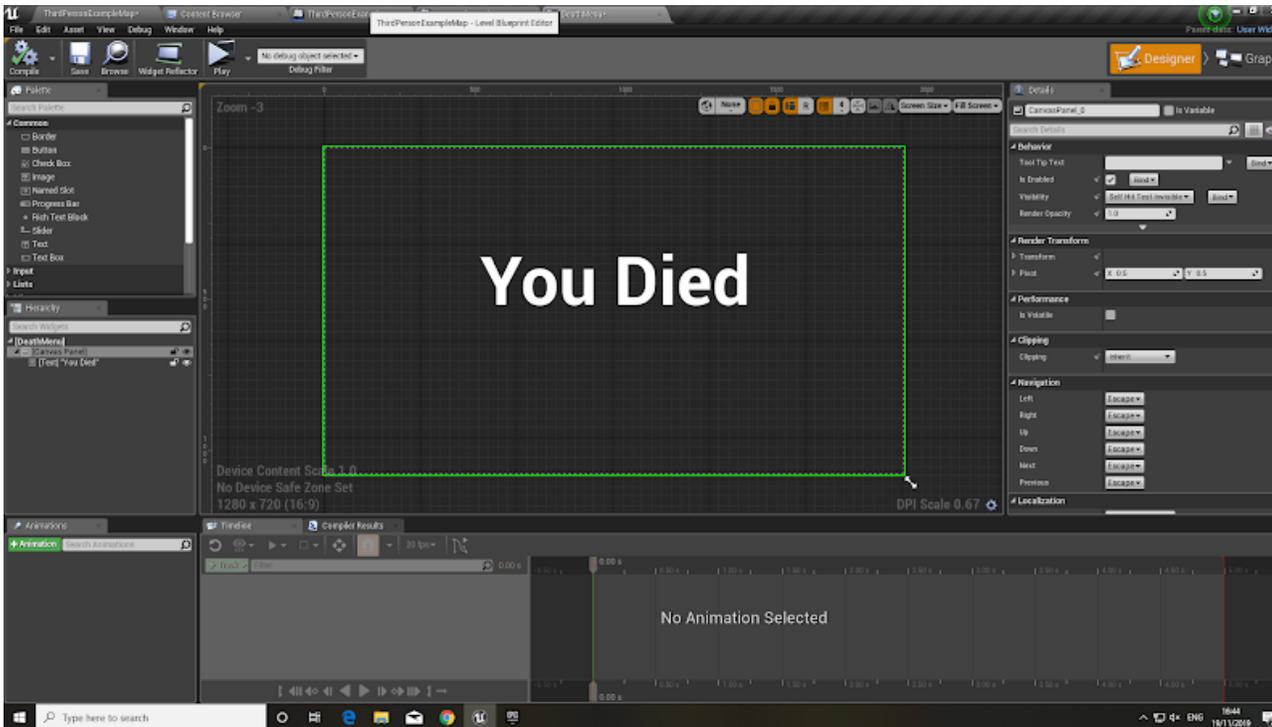


Death Implementation

Next lesson I decided to make a lake that my character will have to jump across moving platforms to get across. But first, I wanted to tackle making my character's death scene that would show if they came into contact with the water in the lake, or anything else that could kill them in game. I made a widget like I did when making my game menu, but this time coded it so that on impact with a collision box it'd make the character a 'rag doll' and make the player unable to move the character after death.

I also made it so that the character respawns at a set target point, which I will have to copy and paste for different death locations so that my character doesn't respawn at the same target point every time. Here is the code and screen below. I am going to edit the death screen however, and maybe take a picture in photoshop that fits the aesthetic of the game. I would love to have the screen fade to purple and then black, with the text saying 'you died' but the animation seems quite hard. I will however give it a try.





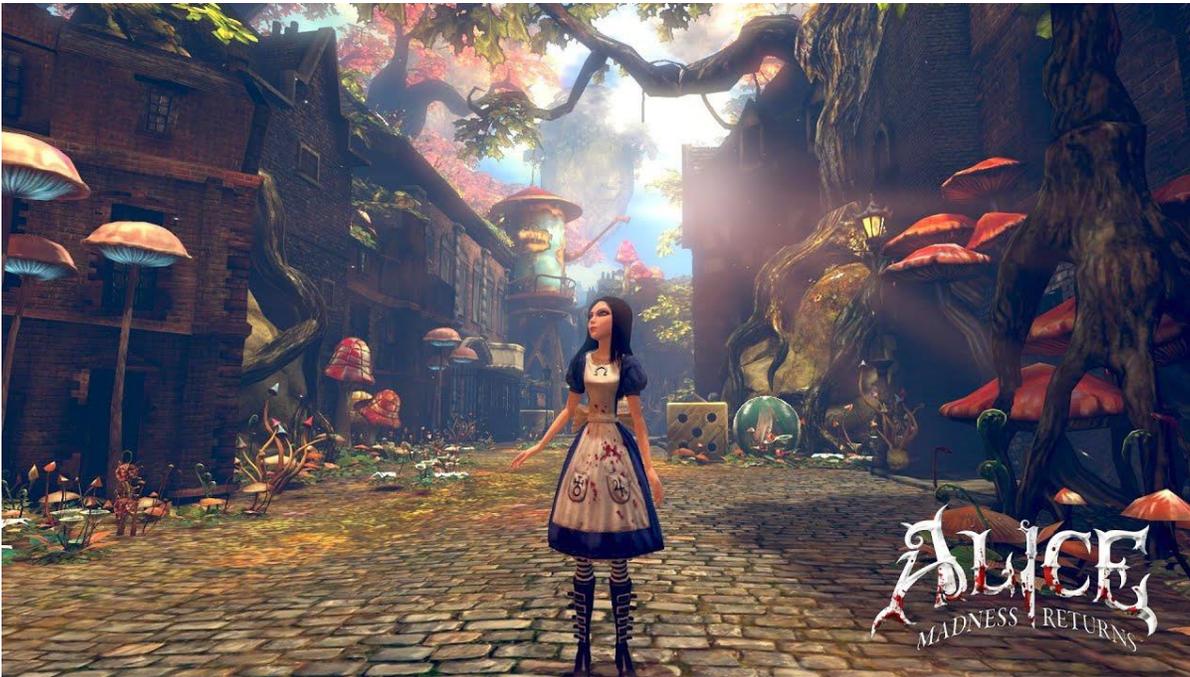
Theme Generation

My main inspiration for my level I takes from a few games. I really enjoy the way snow looks in games, particularly in Dark Souls III in Boreal Valley.



The skybox in this part of the game is also extremely magical. I remember when I first arrived at this part of the game, I was stunned by the visuals; I really want to create something inspired by this aesthetic. The northern lights are also a beautiful touch; however I have to work with what I've got in my texture packs which is restricting. The colour scheme however I can manipulate. Also, something I would love to figure out how to do, from this image, if you walk across the bridge in game, the blurry buildings in the background that look like pictures come into focus and are actually in-game 3D objects. I don't know if I could place atmospheric fog heavily around parts of my level to create a similar effect, but I will play around with different settings to see if I can do this, so the view distance doesn't go across the whole level.

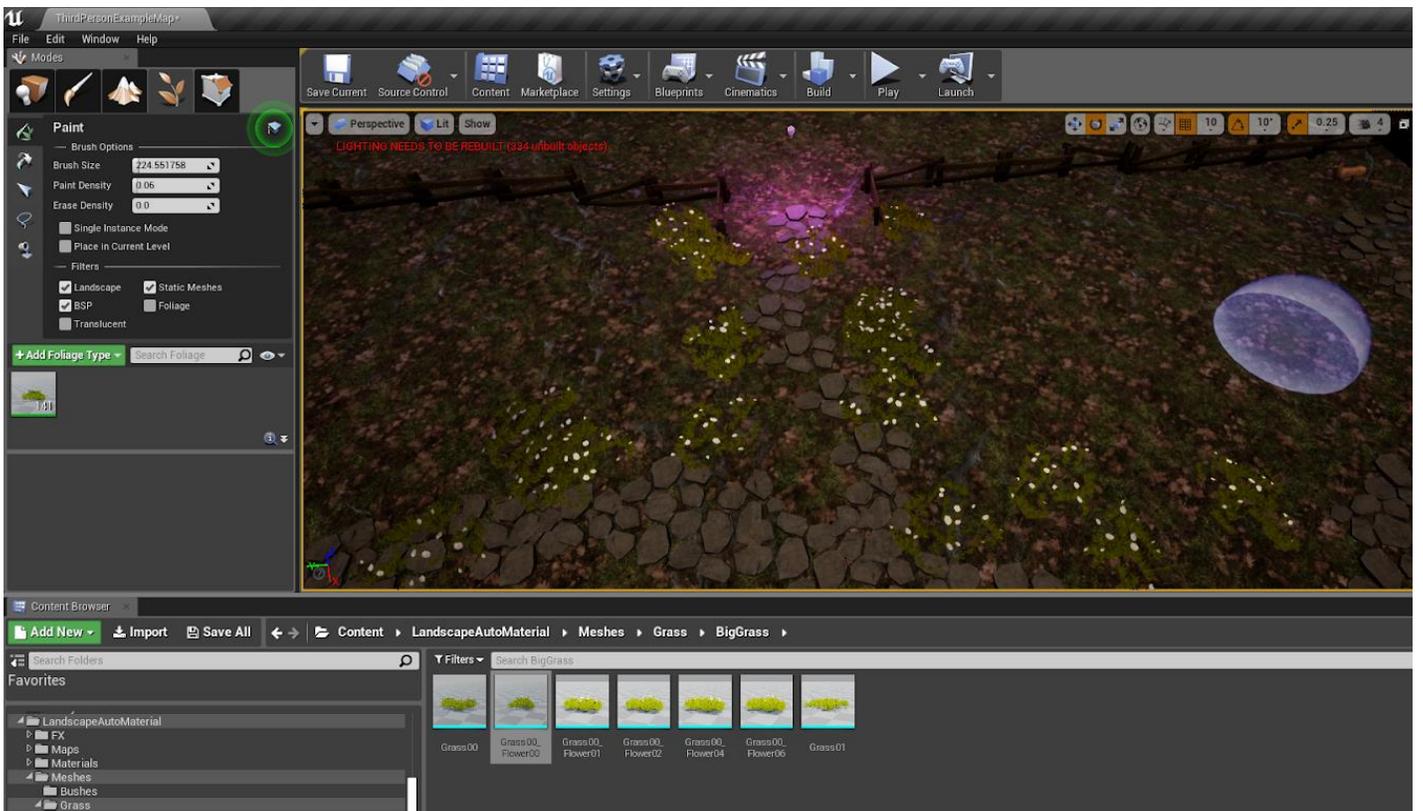
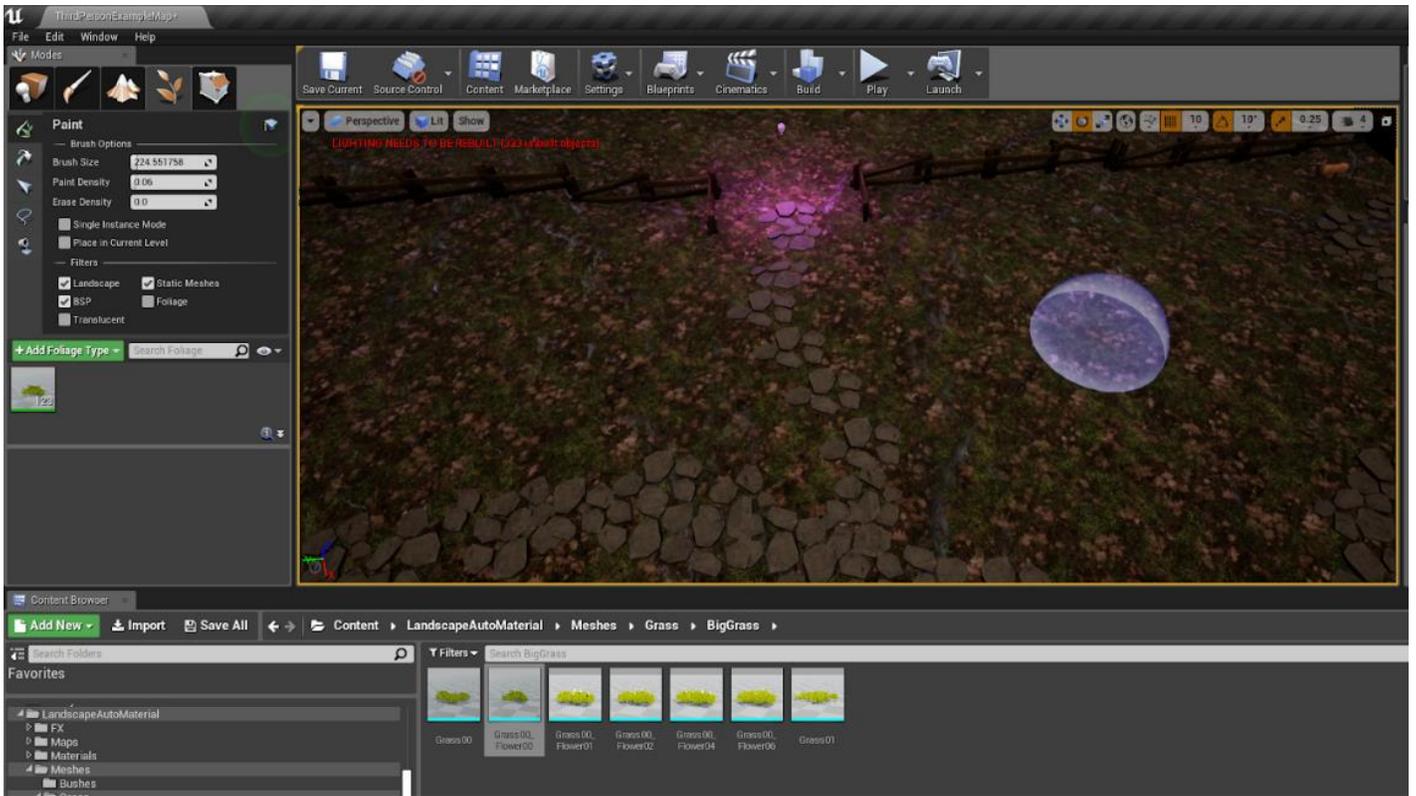
Another game which I really enjoyed the style and look of was Alice Madness Returns.



The oversized mushrooms and flowers on the cobble floor look fantastic, and really bring the fantasy idea I have to life. I also really enjoy the dark twists and visuals in this game, from the bloody horror to the magical world we get to explore. This is something I really want to work on, as the visuals create different responses from certain audiences, and I really want to target a younger, more 'edgy' demographic.

Foliage

Today I created a small village in UE4, and I felt as though the floor looked quite bland. I love the cobblestones and the decor, however it needed some kind of depth and realism to it, instead of a flat texture. I decided I would use the foliage tool to help with this. To access the foliage tool, I had to go into foliage edit mode in the Modes panel. From here, I could add a foliage type, and select which static mesh I wanted to paint onto my floor. I chose a grass mesh, and a grass with flowers mesh, so the floor didn't look too flowery. I decided to test my brush size and paint density, to ensure there wasn't too much or too little grass being painted.



After deciding on a suitable paint density, I decided to paint the rest of my village. The foliage brought it to life, and made it look a lot more realistic and three dimensional than it did before. I will use this feature in the future confidently when painting other foliage, such as trees and other objects in large quantities.



After finishing the foliage of my village, I added more trees and lighting to make it seem even more magical. This is where my player starts the game, so I wanted it to be stunning. I also drew inspiration from the Alice Madness game when creating my oversized houses, trees and cobblestone pathways. It really adds to the illusion of an oversized, magical land.



Foliage was making my game lag crazily when playing and when editing, so I had to come up with a solution to this. I found a function that makes foliage invisible if you are a certain distance away from it, as shown below.



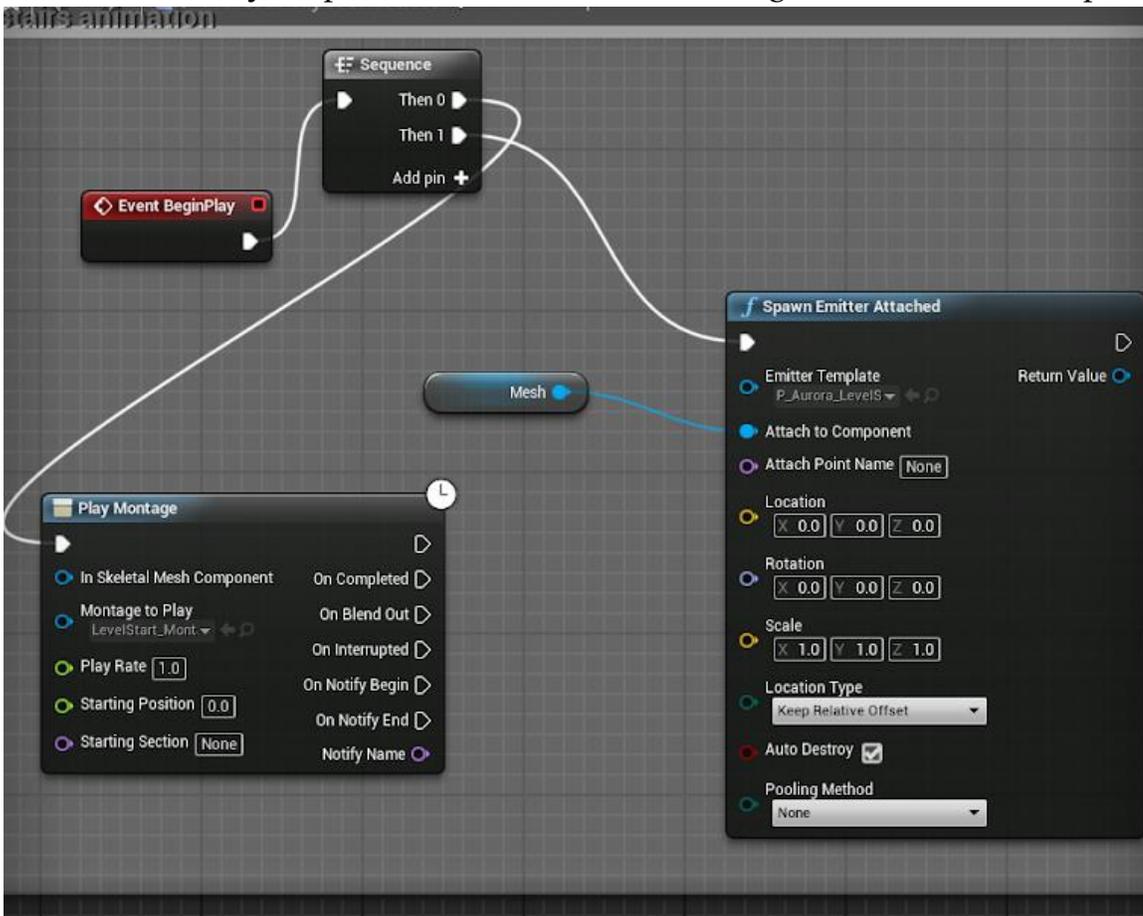


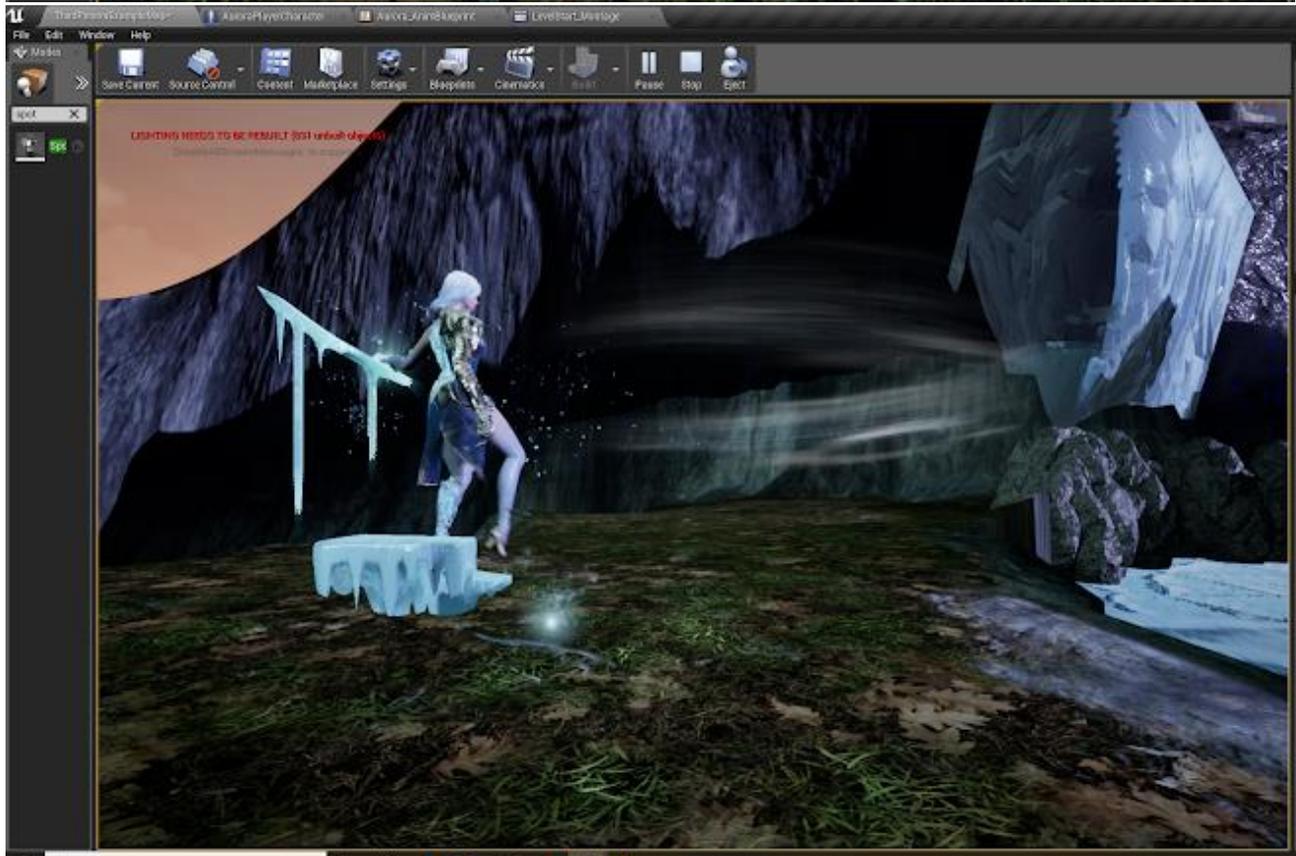
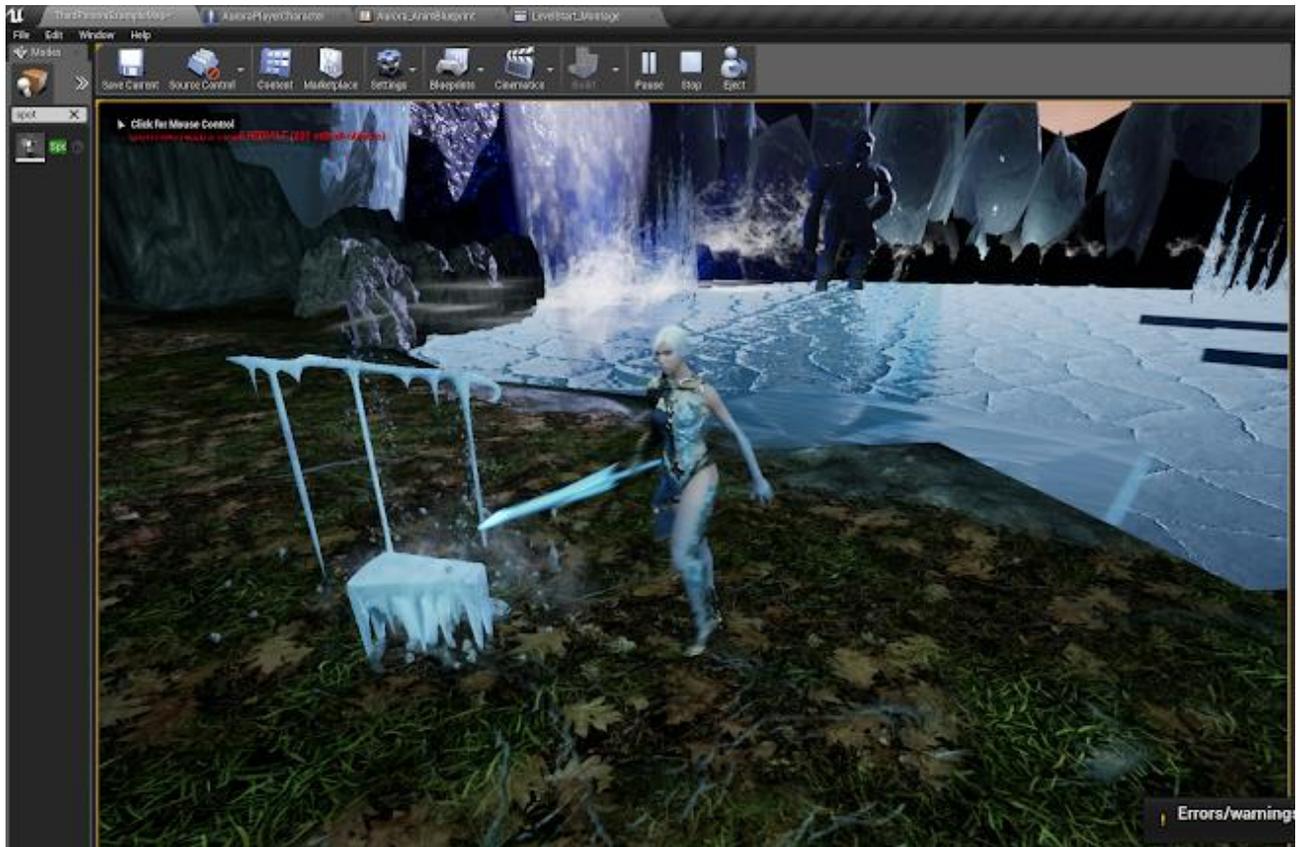
Character Animation

I have chosen a paragon character to be my main character in the game. Her name is 'Aurora' and she is an icy humanoid. I chose her as she suits the icy aesthetic of my level and will fit well into the story looks wise.



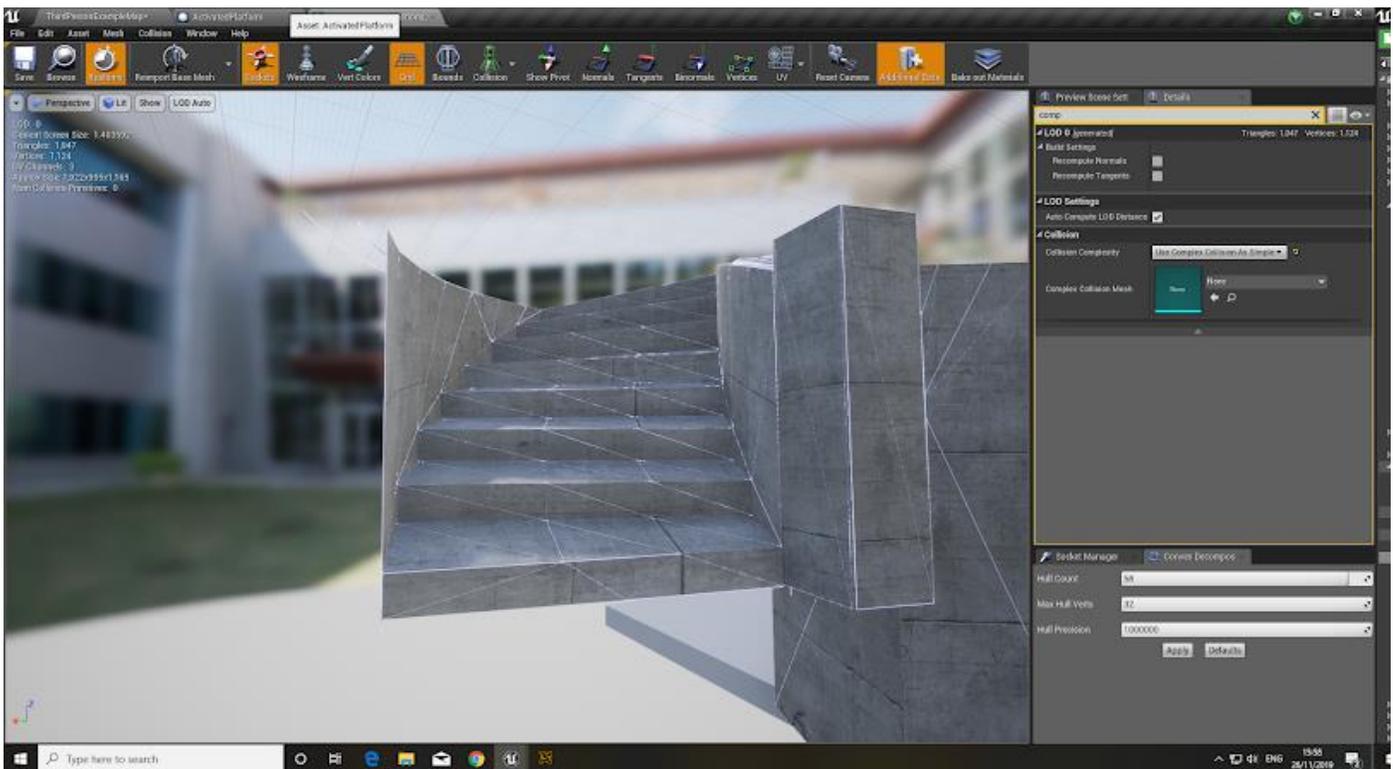
I noticed every time I begin a level, she walks down an invisible set of stairs. I decided to look in her folder which I downloaded to see if I could locate the mesh of the stairs, so I could try implementing the stairs into Aurora's character blueprint when she starts the level and respawns. I do not fully understand how the sequences work yet, however I understand that an EventBeginPlay (when the level begins) Aurora's montage plays (walking down the stairs animation), so I needed to insert the stairs mesh using the Spawn Emitter Attached, that would destroy itself after the montage finished. The code was very simple to create and didn't take long to do with a little help.





Collision

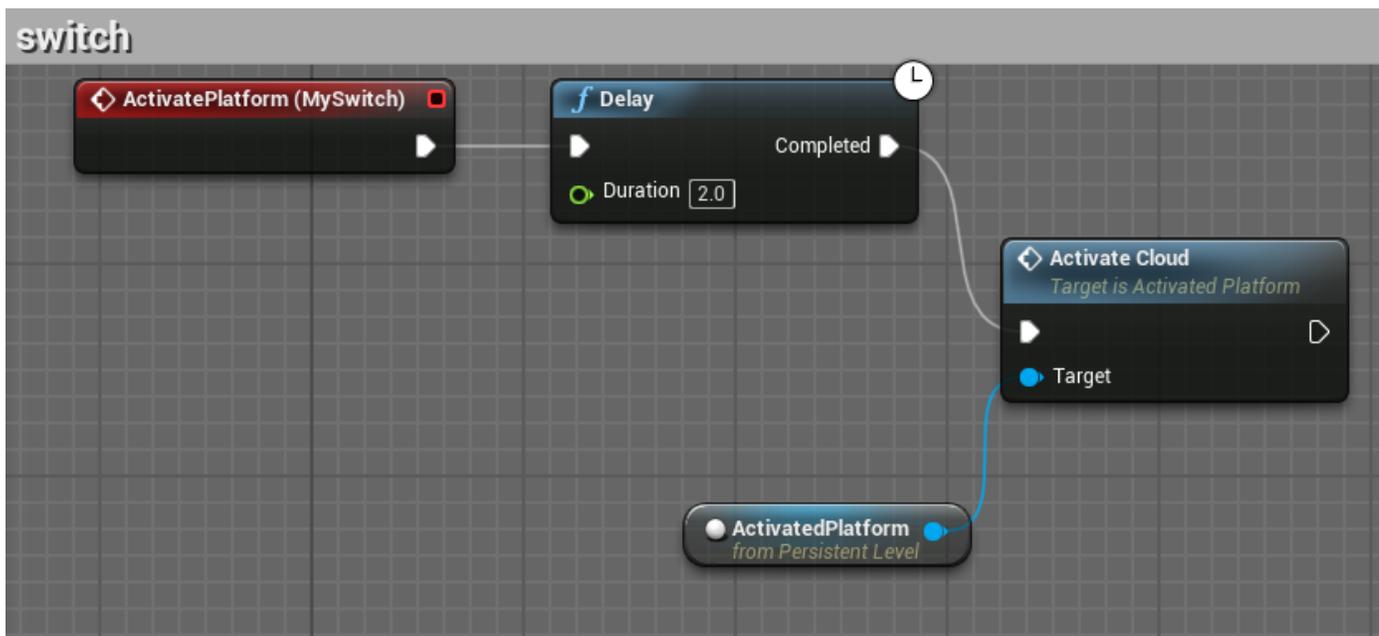
After creating a section in my game up on a hilltop, I wanted to make a moving platform that took my character to some stairs they could walk up, however, when I tested this out in game, my character couldn't walk smoothly up the stairs as the default collision surrounding the mesh was lined up horribly, more like a slope than stairs. To fix this, I decided to go into the collision settings of the stairs and play around with the different collision settings. I clicked on the 'complex collision' button, and thankfully this created the correct collisions for my staircase. My character can now jump up each step without sliding down.

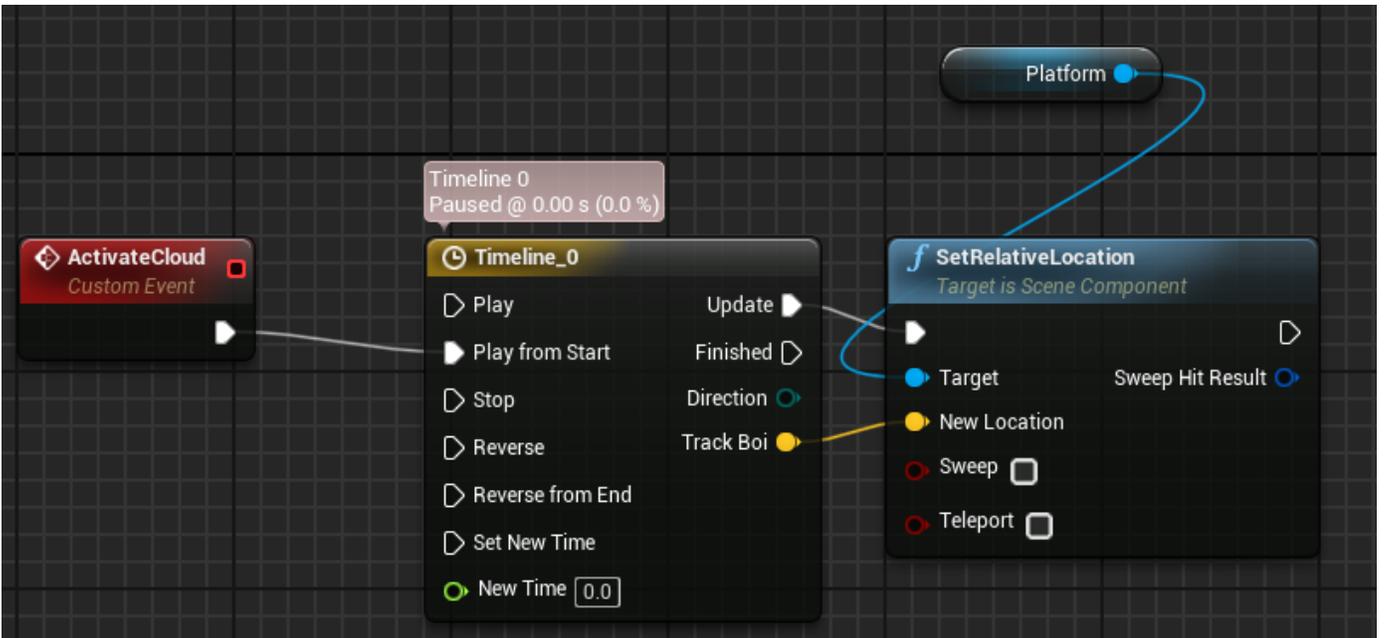
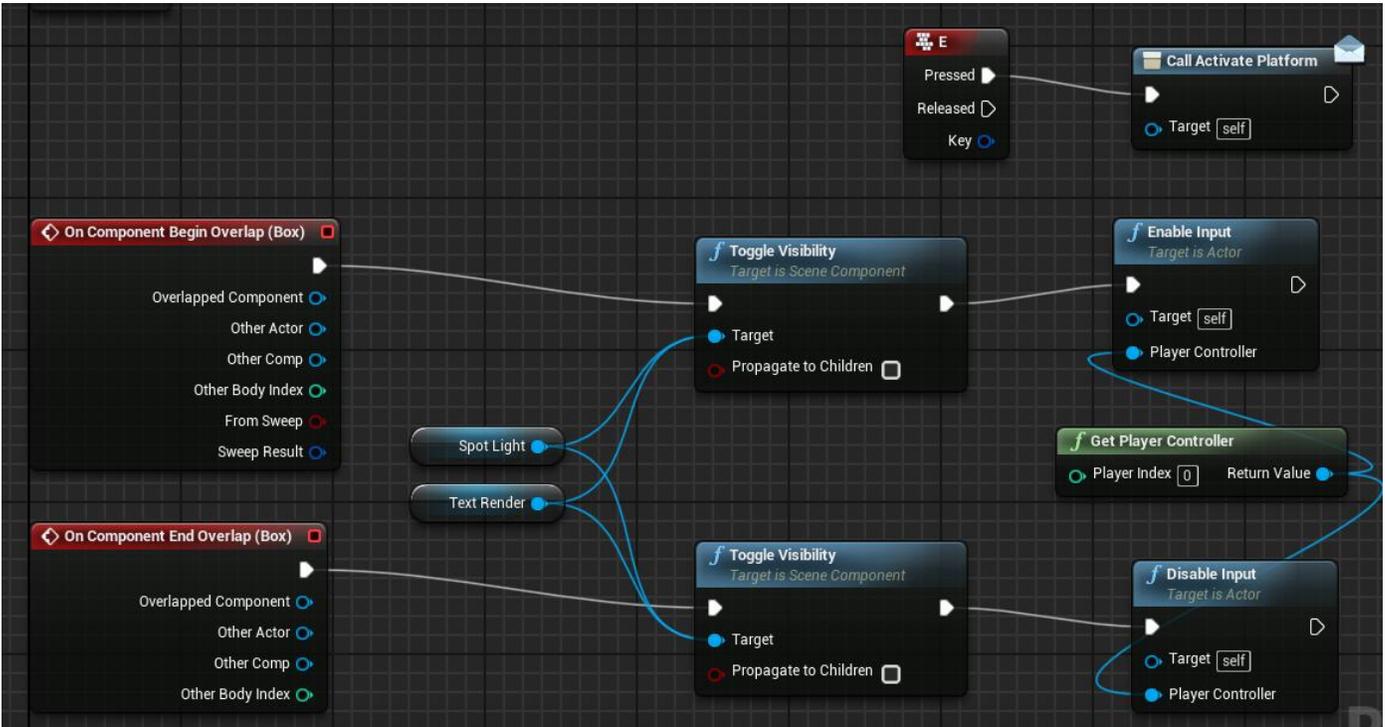


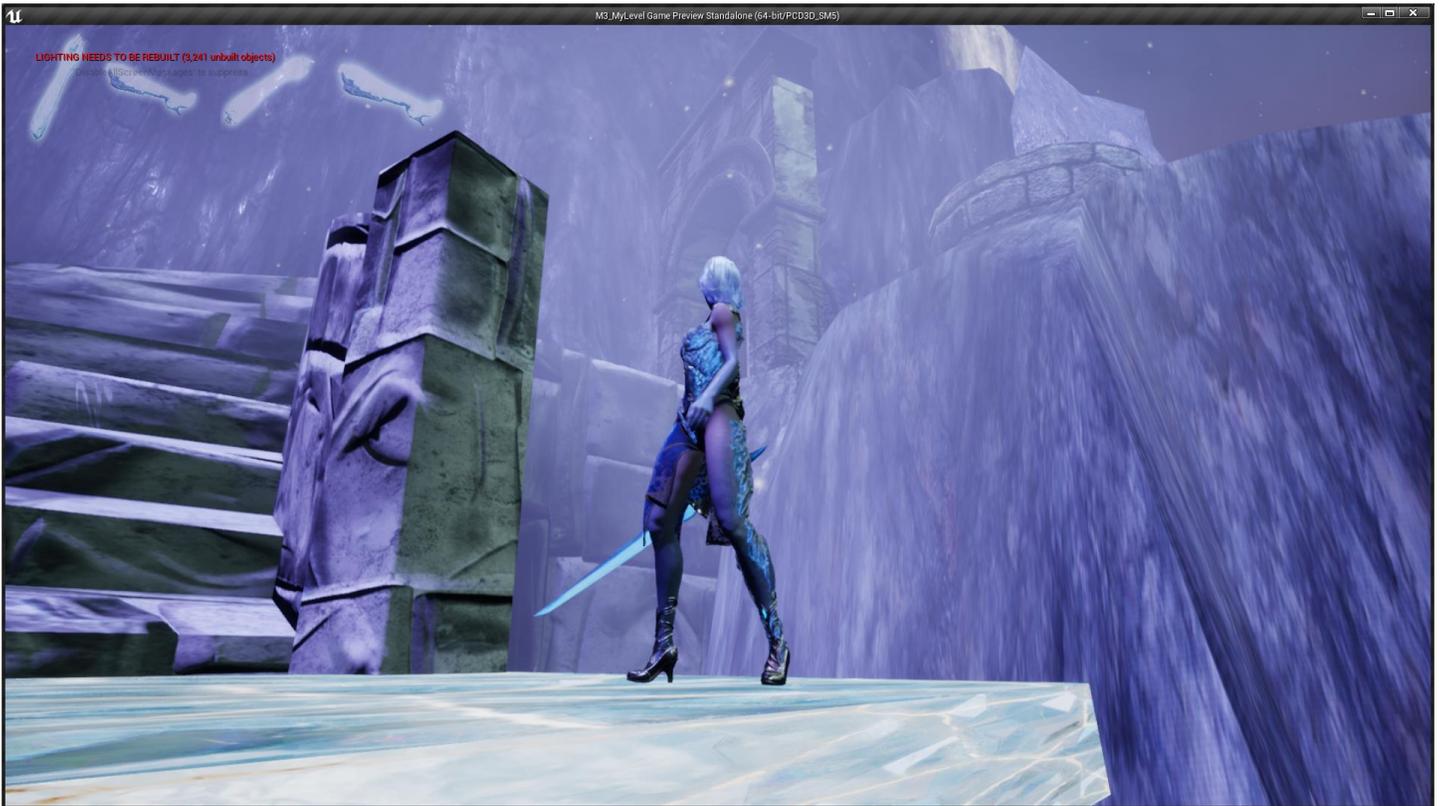
Connecting a Switch to a Moving Platform

After sorting out my stair's collision, I replaced the texture, so they looked beautiful and icy. Now it was time to work on my switch activated platform. First, I created a Blueprint Class Actor and added

come components: a box collision, text render and a spot light. I made sure I set the visibility of the spot light and text to invisible, which would only appear when my character overlapped with the box, which I implemented using the 'on component begin overlap'. I then enabled the input for the player to activate the event. I did this by adding an end overlap and disabled the input when my character was no longer in the box. I then created a dispatcher called ActivateMatineeSecret, which I would call when my player pressed the assigned key, in my case the key was E. I played the BCA into my level and made sure to go into my level blueprint to activate it, creating a flipflop node between playing and reversing my switch. I then connected this switch to activating my platform in a timeline node so it would move to the desired location, the top of the stairs.

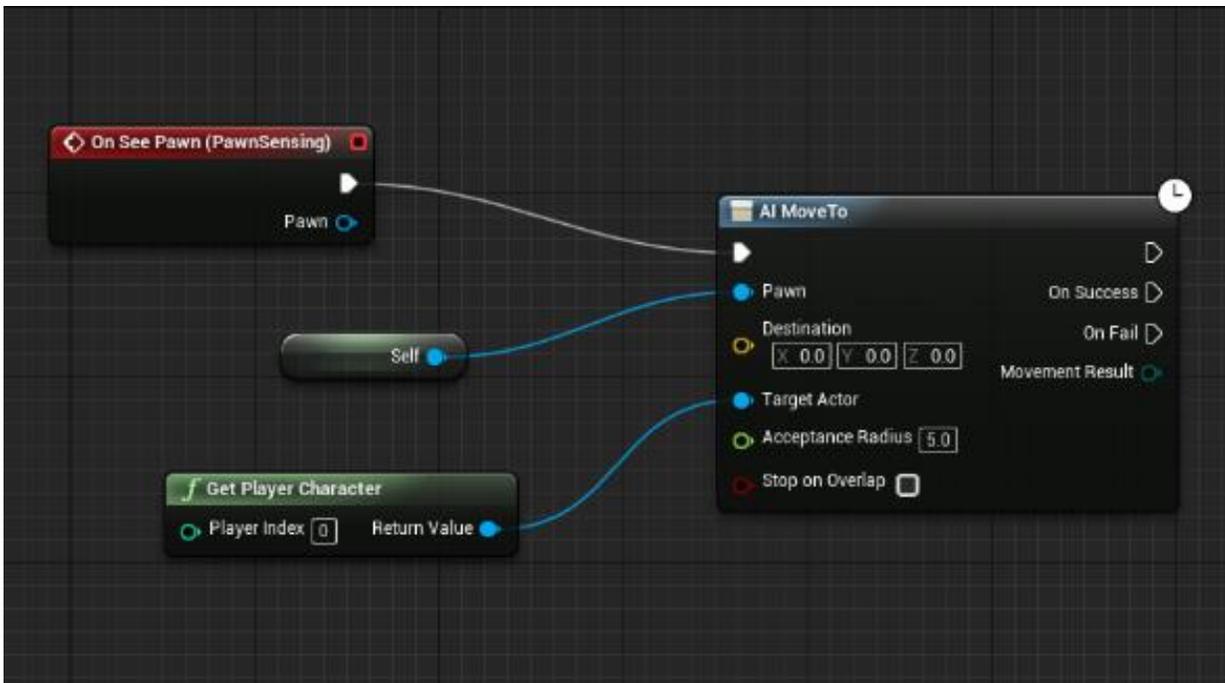


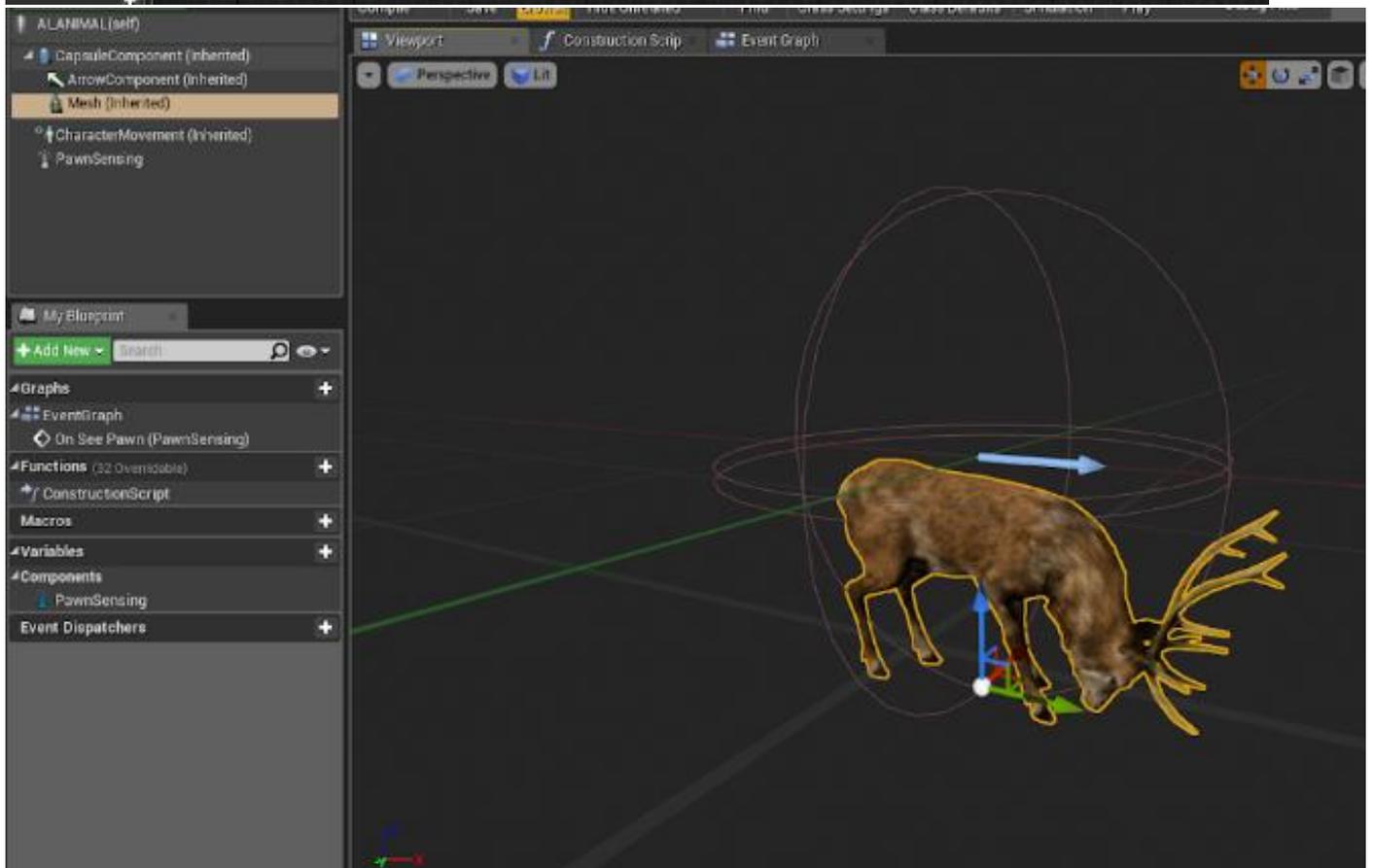
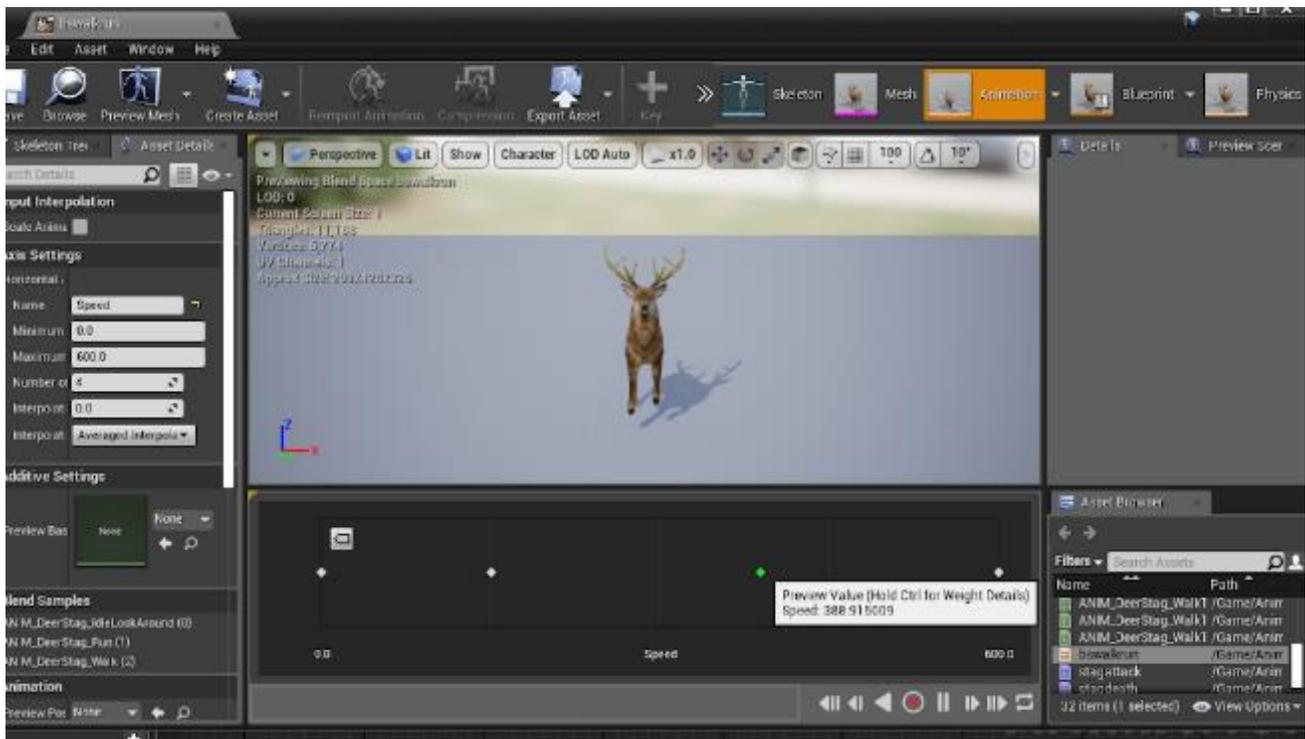


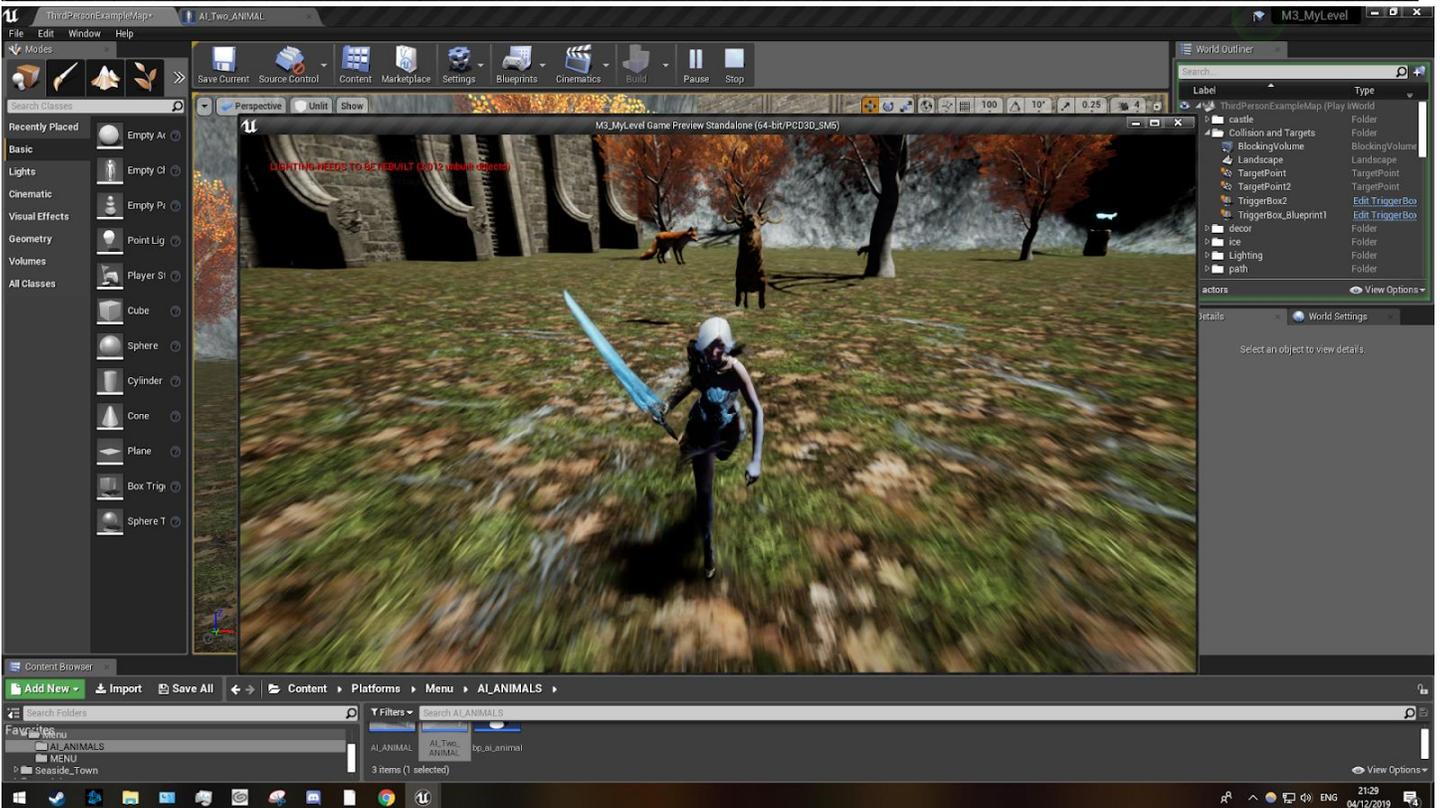
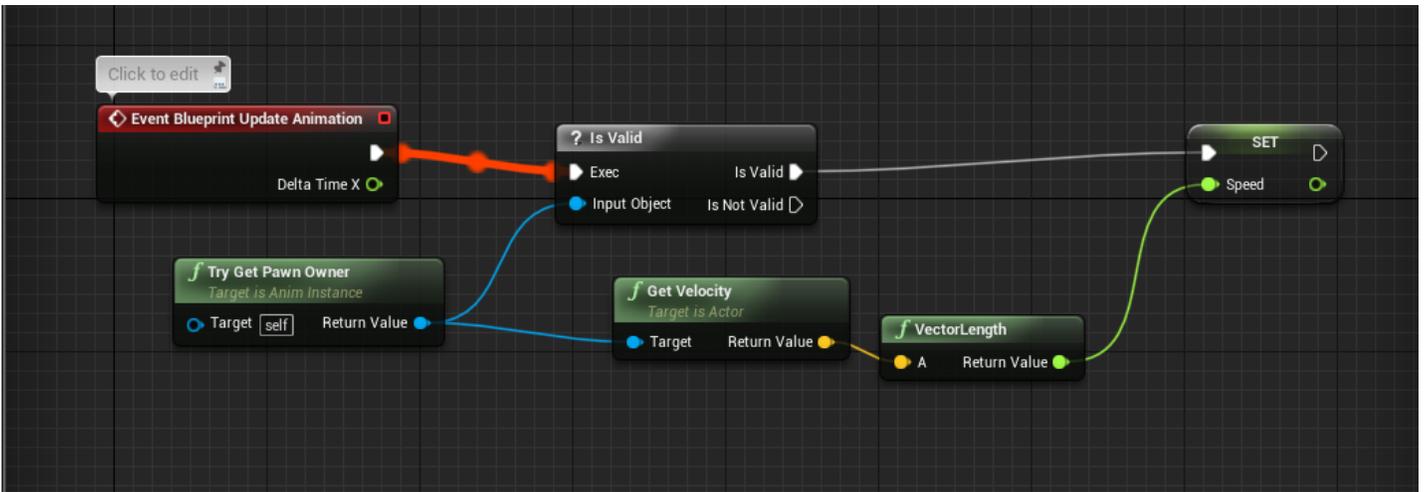


AI Animals

I decided I wanted to have an animal in my game that stands idle and follows my character around a certain area of the game. I chose a Stag from my free animal texture pack which I added to my level and decided to start animating it. Firstly, I created a Blend Space 1D and added in an idle, walk, and run animation. The blend space is used so the animal can have smooth transitions from walking to running. After this was compiled and saved, I opened my Stag's blueprint, and began to add in some code. I made sure that the stag's vision radius was big enough to notice a human realistically, and when they did see my player character, they would follow me. I also created a speed variable, and a state machine, to compile the animations together with the stag. This is not completely what I wanted, I would rather the Stag walk around on its own, however the coding and steps to achieve this is not as simple as I thought.



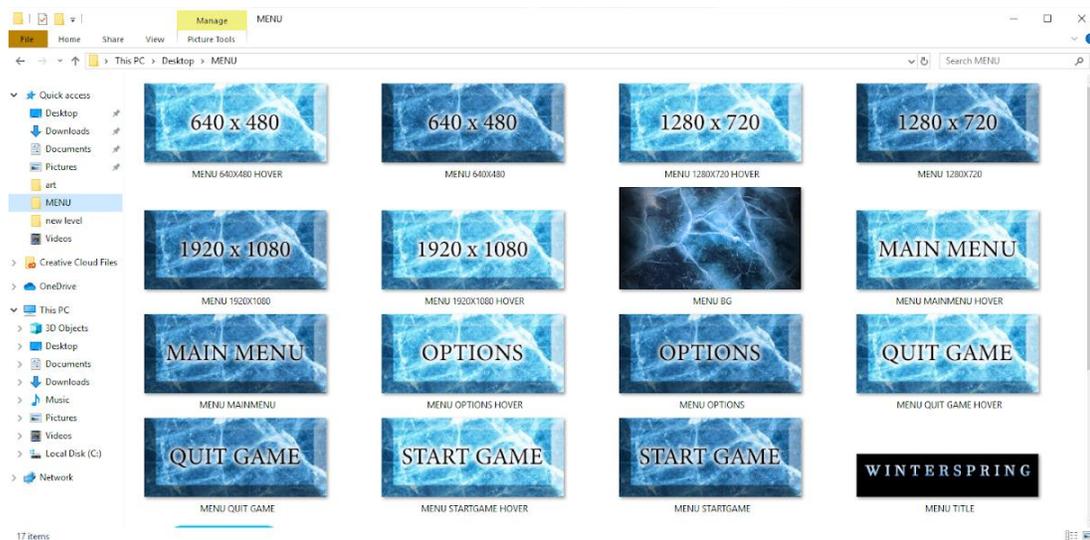




Making a Customized Menu

I had made a menu in the past which looked quite bland, and instead of a background you could just see my character in game. I wanted to instead make a start menu that looked full of personality and matched the theme of the game. To do this, I first had to find a background, choose a name for my level, and create some buttons. I created everything in Adobe Photoshop and found the picture of ice from google images. I also realized that later on, I wanted the buttons to glow when the mouse

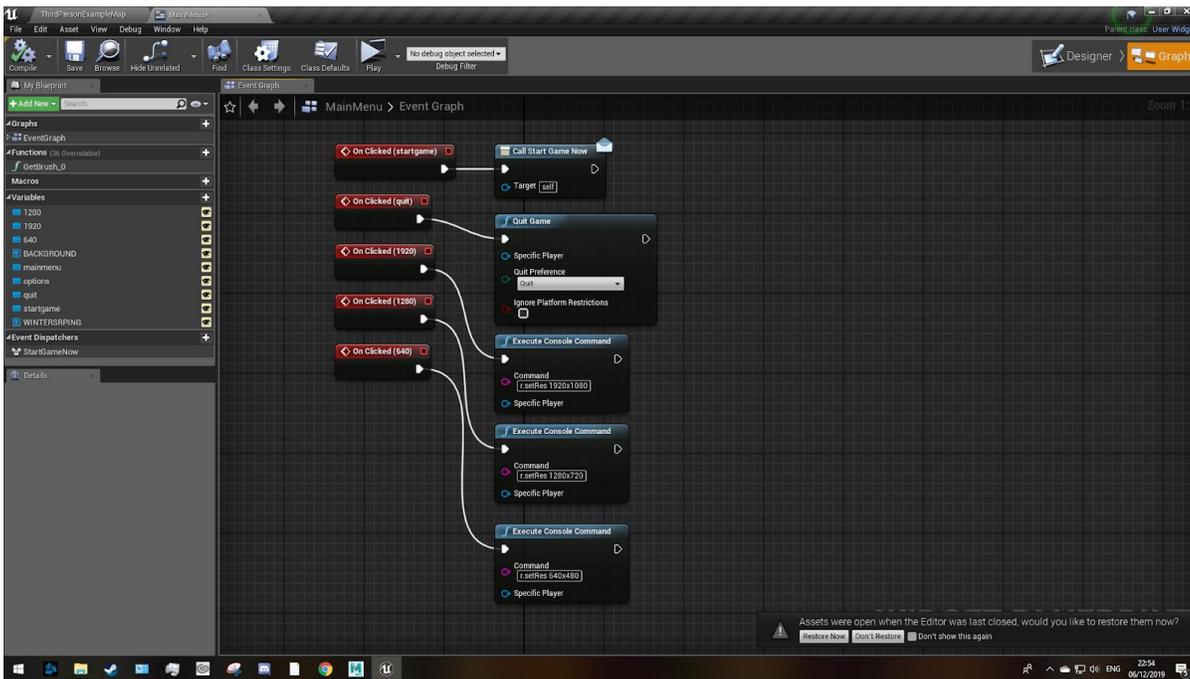
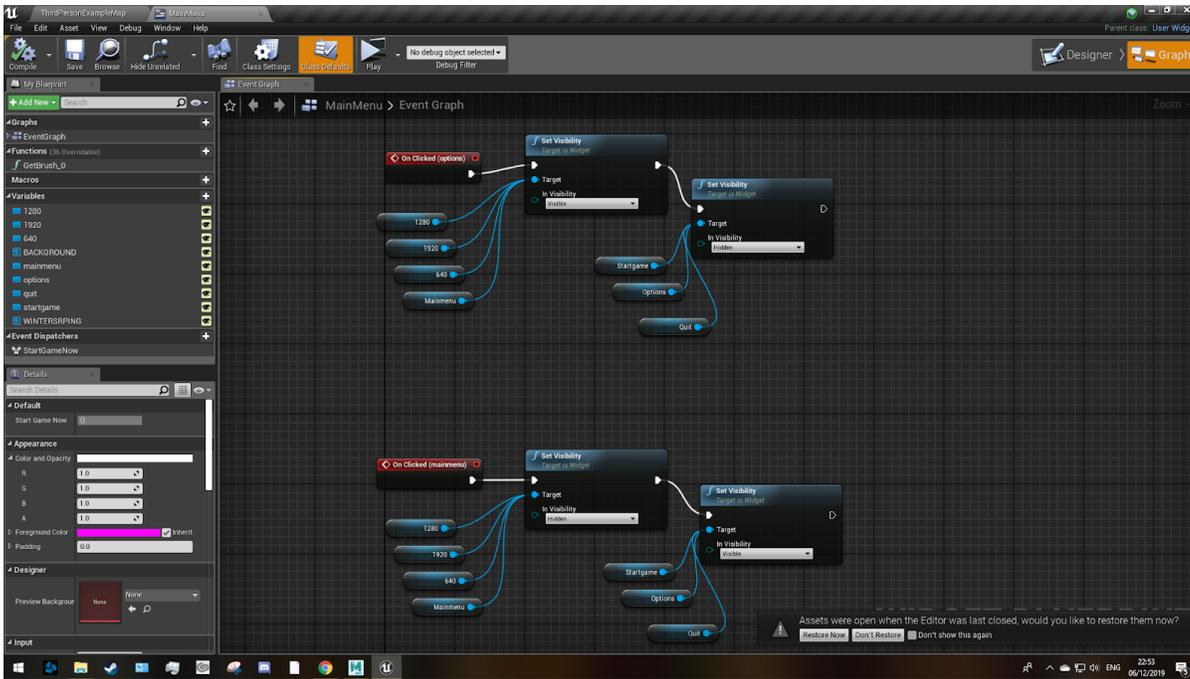
hovered over them, so I upped the brightness in Photoshop of my original buttons and imported the image into the picture option.



After everything was created, it was time to piece my menu together in the widget. As you can see, my buttons are all overlapping in this image. This is because I am going to be adding code in the graph of my widget so that only the top-layer menu is displayed on event begin play. Only will the resolution options be visible when the options button is clicked, as well as the original menu being hidden. I will also be implementing code so that the resolutions will be changing on the screen when clicked. I have taken the time to individually label everything in the details panel, from my buttons to the background so that I don't get confused when implementing data and linking buttons in the graph section of my blueprint.



Here are some pictures of my finished graph in the widget blueprint. I have ensured that all the correct buttons will be visible or invisible at the correct times, as you can see below. I have also added every 'on clicked' event needed in my widget blueprint, so that I know which buttons cast where. My resolution buttons are correctly formatted, and the back to menu button as well as the options button all function perfectly. Lastly, I decided to check my resolution changes worked, I did this by clicking 'play' and trying out the different resolutions. As you can see here my menu is full screen when I chose my monitor's default resolution.



After all this worked correctly, it was time to ensure this worked at the beginning of my level and loaded up before anything else. It was a huge problem at first, as within my research I was instructed

to build my menu within a different 'level' as I couldn't get my menu to disappear off of the screen after hitting play. I had to code the menu level to cast to my other playable level. It was really difficult to figure out, however my tutor helped me fix this, and took away my unnecessary level.

To do this, I coded as follows. Event Begin Play (In my game map) > Create My Menu Widget > Add to Viewport > Bind Event to StartGameNow > Start Game (Custom Event) > Remove from parent. This works perfectly as my menu is not complex enough to warrant using another level, unless it was a game like Oblivion, World of Warcraft or the Sims where you have a lot of data being stored in the menu, such as customizable characters, races, worlds and more. It successfully disappears when 'start game' is pressed and my character lands where she is supposed to be at the start of the game.

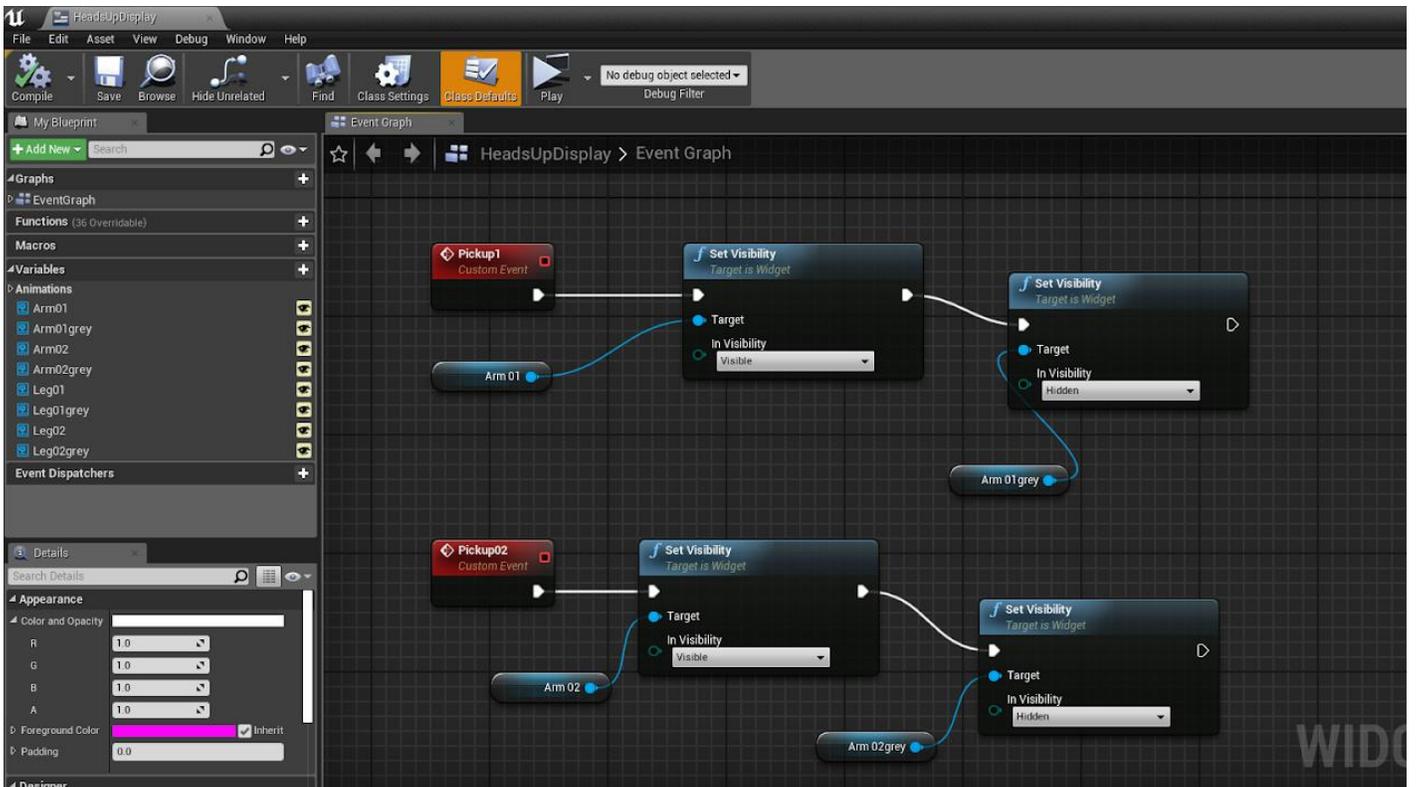
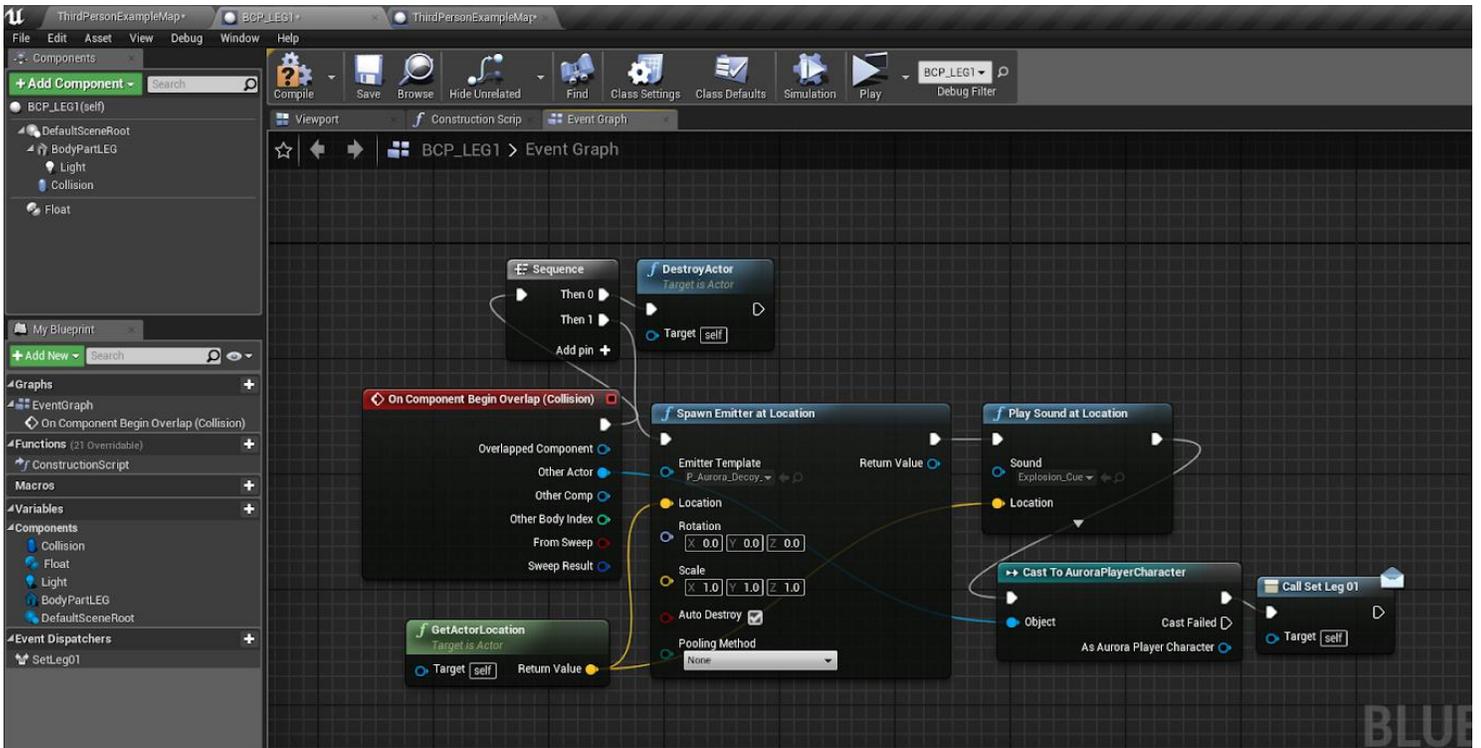
I don't have much more left to do from here in my level. I'd love to do so much more with it but it is impossible without a team and greater knowledge. I just need to wall off one section of my game and implement a function where certain actors can be collected. Simplified, in my game you play as a girl's lost soul, and you must collect the shattered frozen pieces of her body to fuse together again at the end of the level. These actors need to be collected and shown on screen.



Collecting Items & Animation



After this, I needed a display that showed my items as uncollected, however when collided with they showed as collected. Firstly, I created a blueprint with my own created images of my four items, two



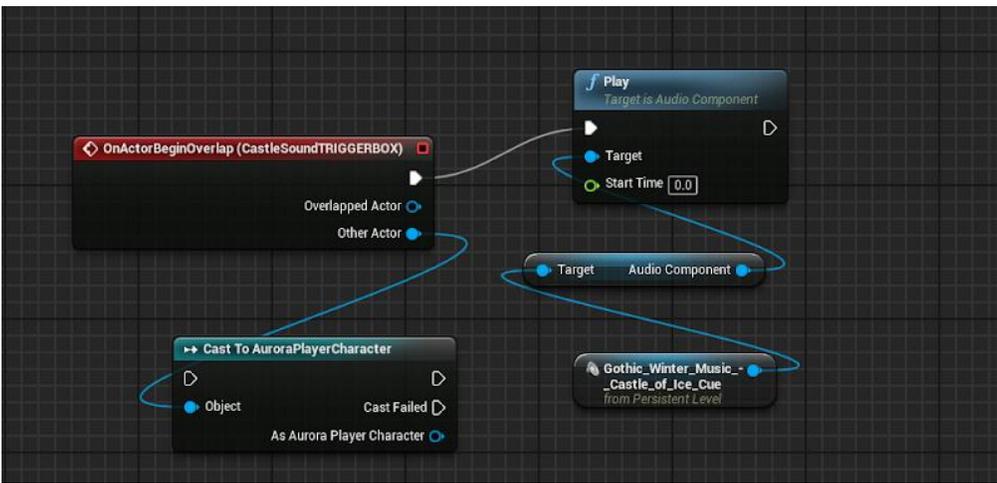
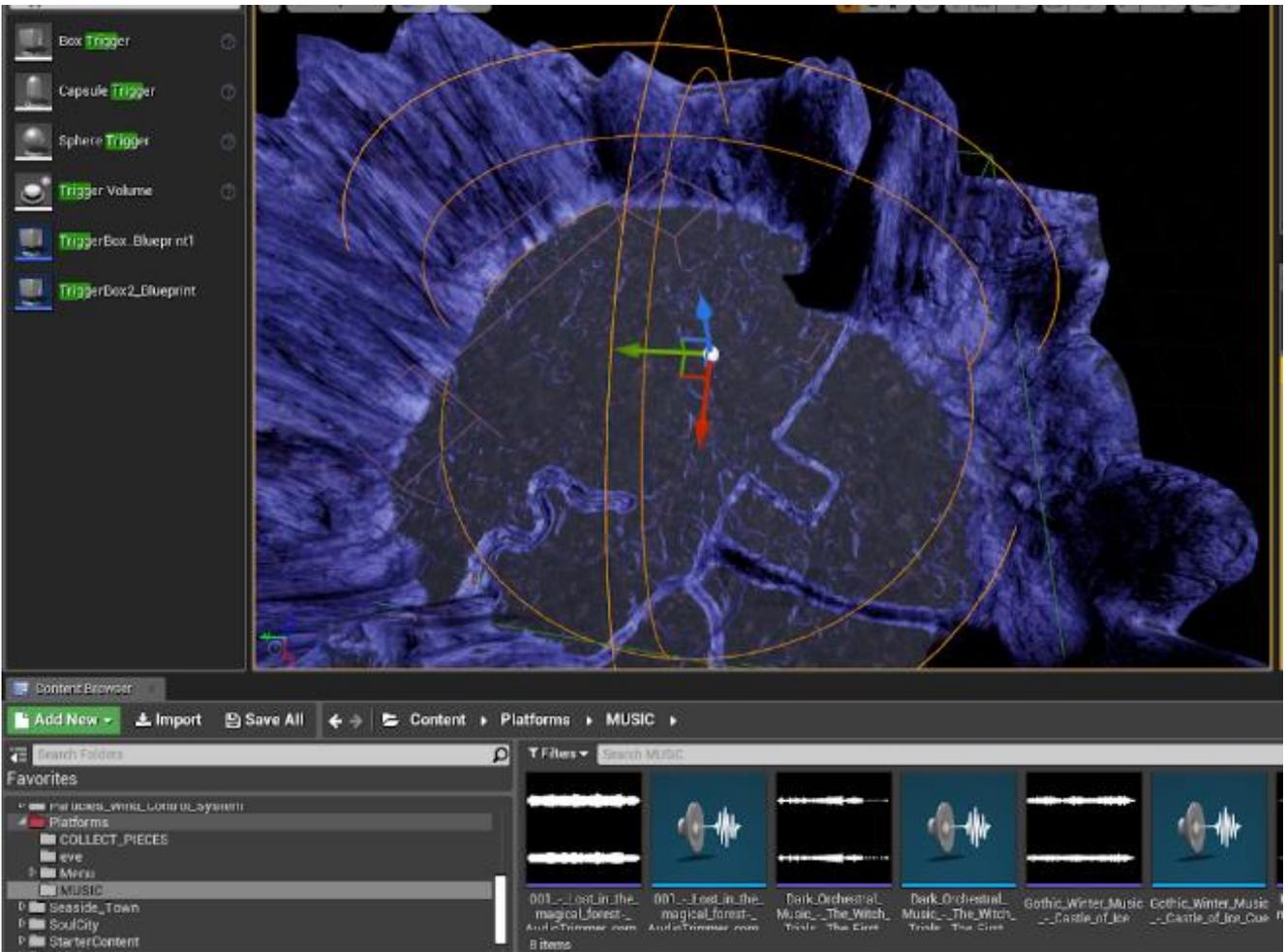


Sound

I'm currently having a lot of problems with sound and I don't think I'll be able to fix it before the deadline, but it is not a necessity. I have implemented all my royalty free ambient music into the game and created trigger boxes that play sound when my character walks into them. My sound files are long enough, they aren't ending, and the loop button is ticked, however when I collect an item the sound fades out and doesn't come back until I enter another trigger box. This is extremely annoying and hopefully I can find a way to fix it next lesson. Here is my code so far, and how I have laid out my trigger boxes and sound.

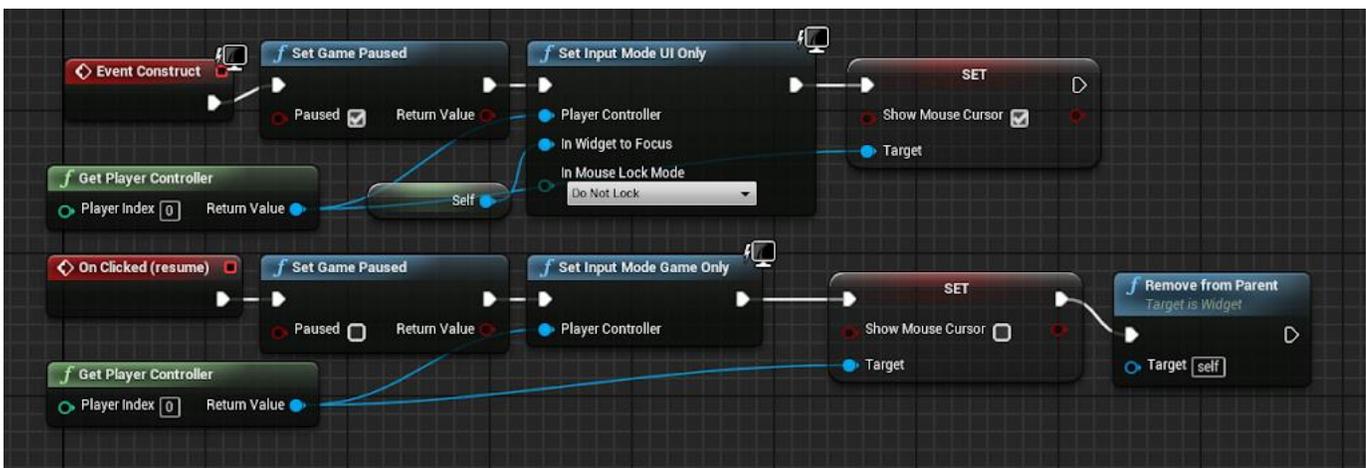
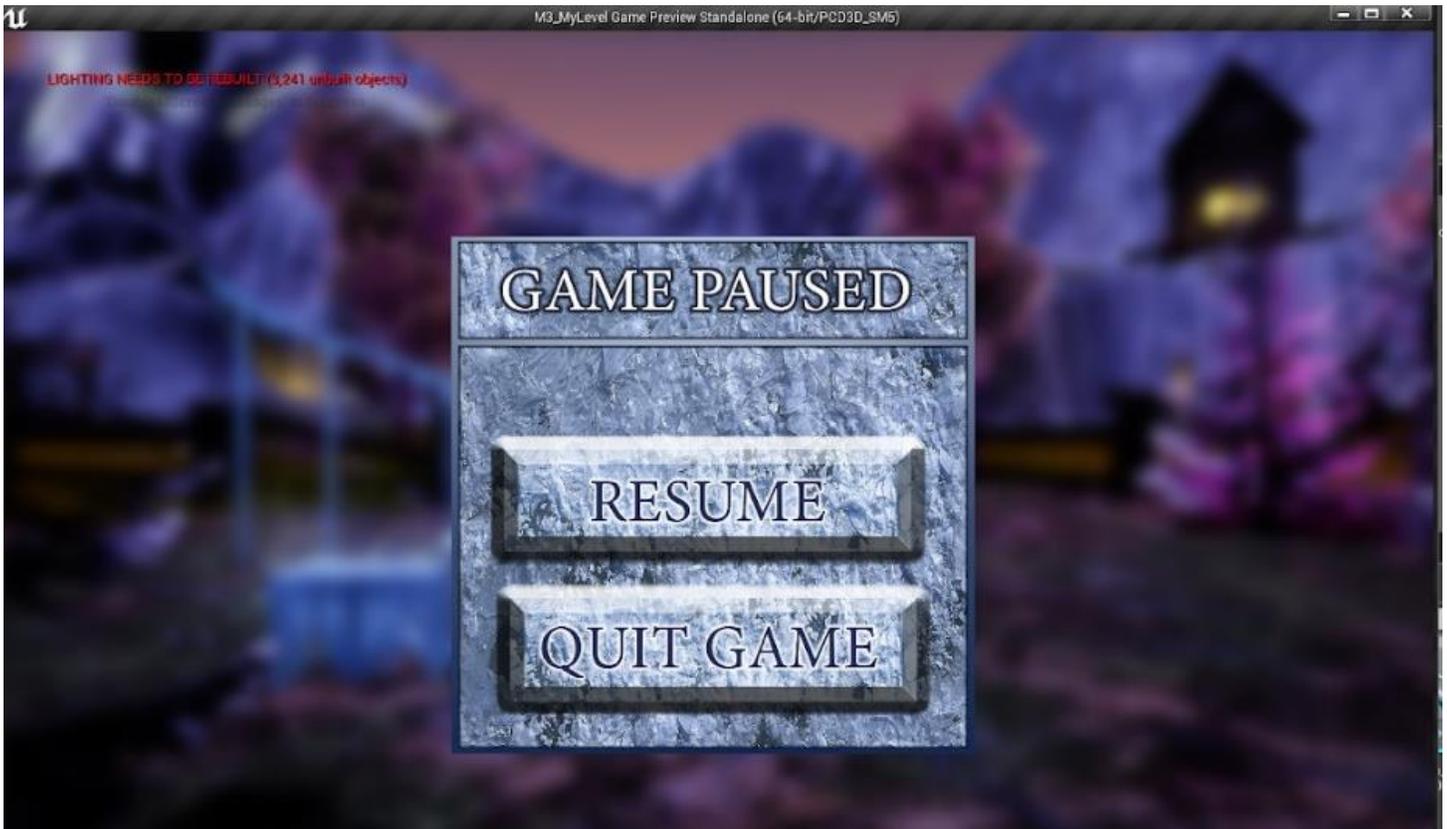


I have since changed these clips to 'cues', however it still does not work. I have also given them priority over other sounds, and that does not work either, but I don't want to remove my sound animation from my items.



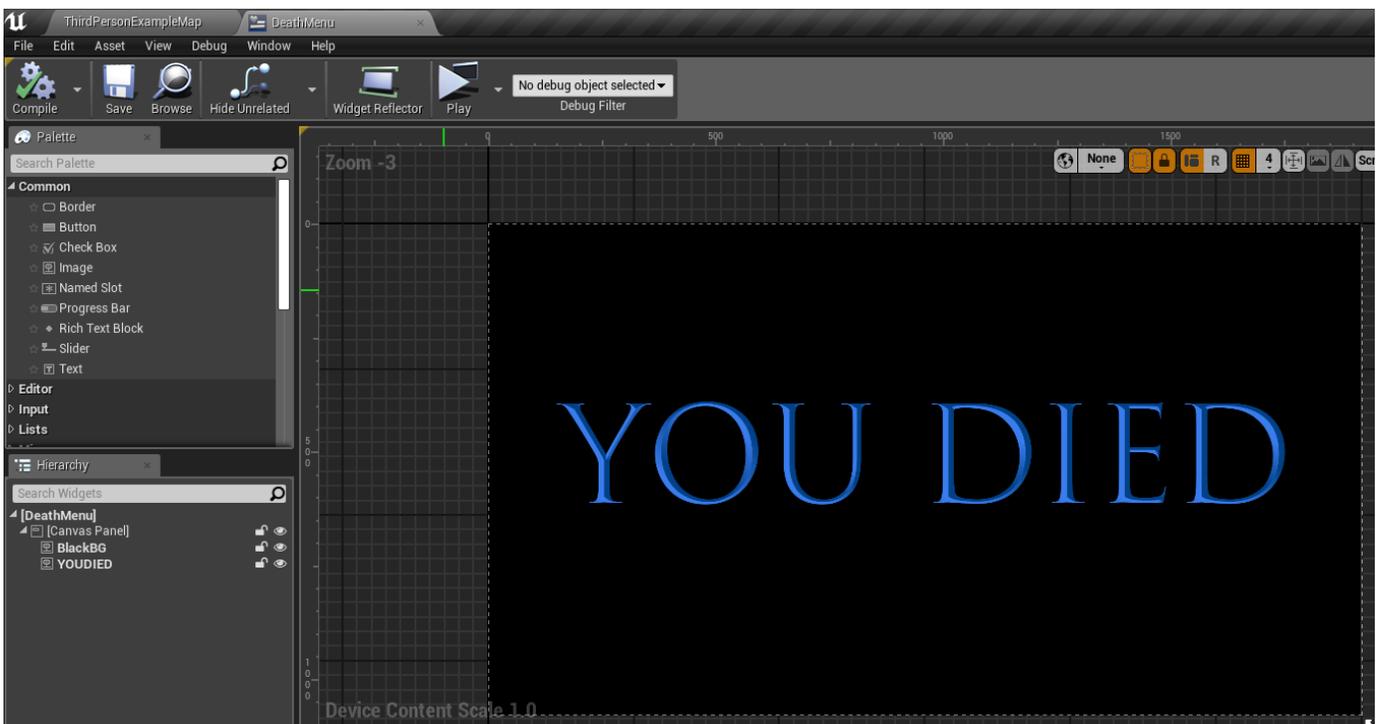
Pause Menu

Today I created a pause menu in a widget. I only needed two buttons this time, just a resume and exit. I designed everything myself in photoshop, however the blur was done in UE4 as I used the wrap tool and added blur. The camera does not move, and I successfully added the mouse to the UI, and took away the mouse on resume. I also made sure that the player could not access the game mode during the pause menu and vice versa. I used the buttons Ctrl+P to pause as ESC quits my game simulation. The player has the option to quit the game or resume with no time passed.

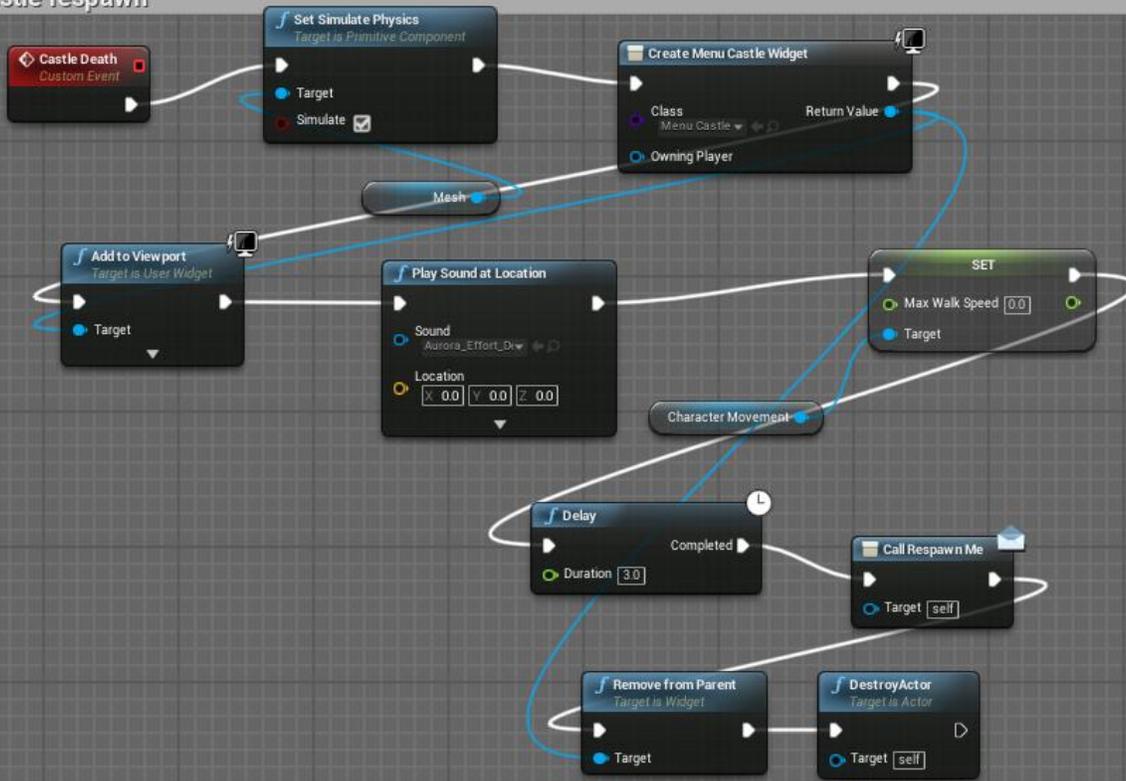


Revisiting Death and Respawn

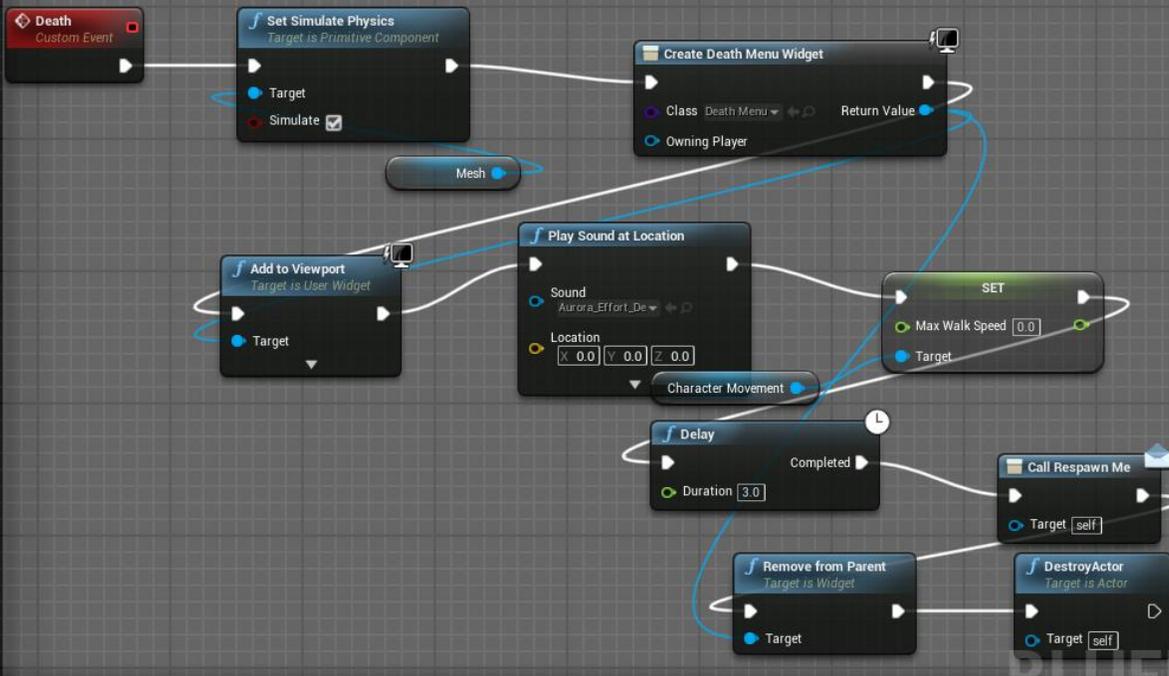
I needed to update my death system, as I have another few places where my character can possibly die unlike at the beginning of my level. To do this, I created a trigger box in the vicinity where if my character came into contact with it, they would die. I opened the level blueprint and coded this the same way as I did with the sound, however instead of it triggering sound, I cast to a custom event in which my character would turn into a rag doll, show the death widget which I created, and during the time it is on screen my character in the background is being teleported to the target point which the specific trigger box is connected to.

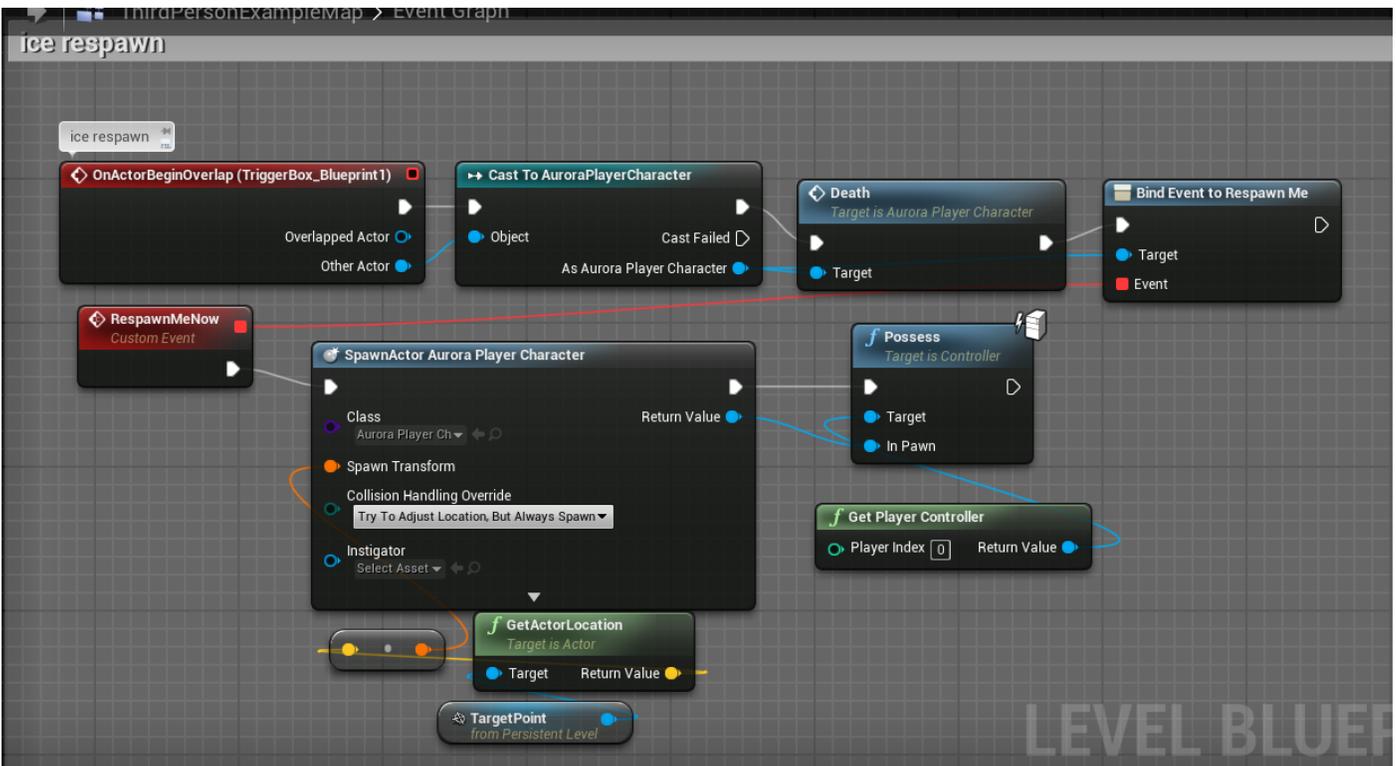
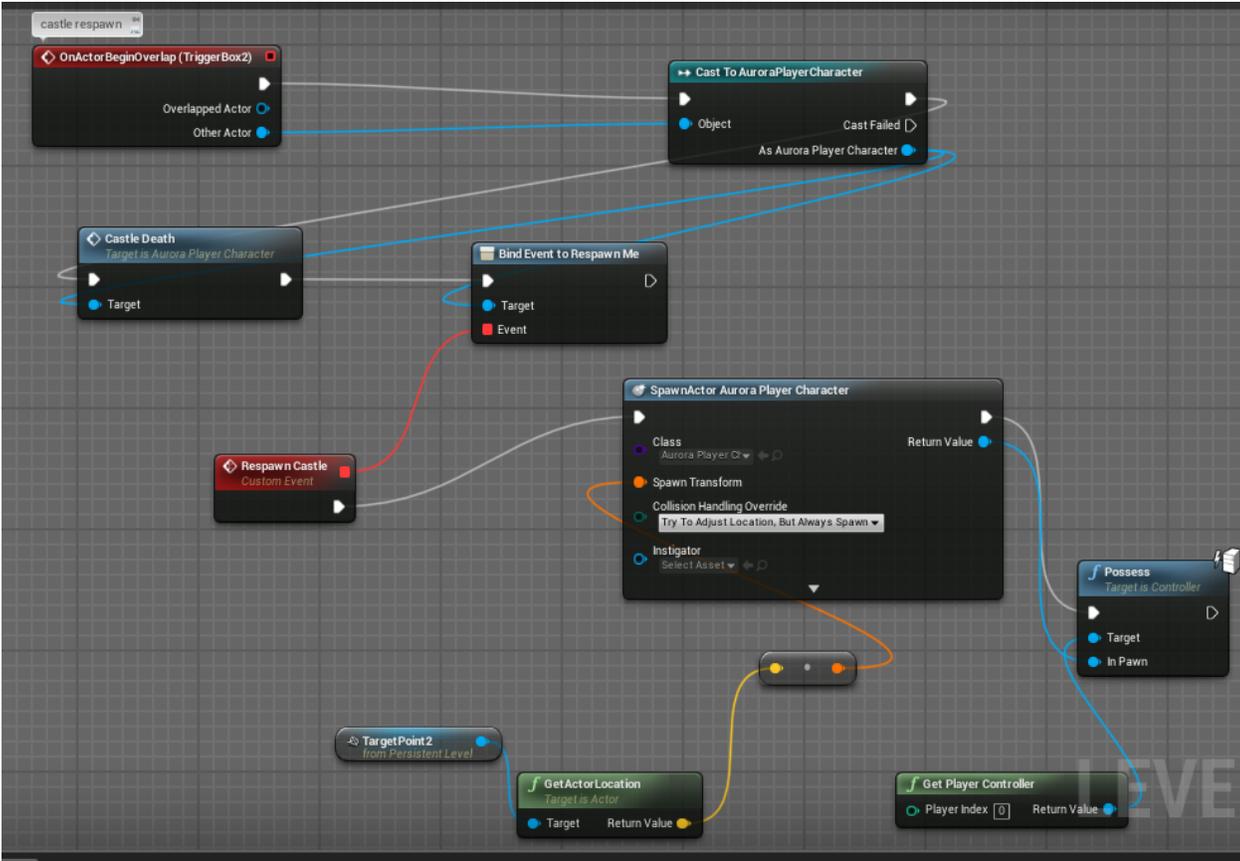


castle respawn



ice death





If I have time, I will set up a health bar and give my character fall damage, and damage from objects like fire or being too cold, maybe a stamina system.

Evaluation and Peer Review

The project I was given was to create a functioning 3D game within unreal engine, the goal to get from point A to point B with obstacles along the way. Within this project I explored different types of game design, game mechanics and different ways to implement effects in the game. I looked at pre-existing games and created mood boards of games such as Dark Souls, God of War, World of Warcraft and Alice Madness returns as inspiration for my game, looking at what made them good games and how I could incorporate those themes into my game. I also received feedback from my peers to see if they enjoyed my game and why, which I will be discussing in this evaluation.

Winterspring is an item collection game, suitable for those aged from 12+. It takes around four minutes for me to complete the game, however I know where I am going and what to do, for someone new playing my game it takes around 7-8 minutes on average to complete, which is the intended play time. I really enjoy the theme of my game, as it centers around ice and restoring the player's body parts which were stolen. The main character is a humanoid being, half made up of ice, and the ligaments you as a player are collecting are four of her frozen body parts around the map. The complexity of my game is very simple; there are four items to collect, and you must piece them together at the end of the game by successfully jumping across moving icy platforms. My game is unique however many designs and visuals in the game were inspired by other games such as Warcraft and Dark Souls.

Ideas and Inspiration

Initially I explored ideas involving medieval themes. I feel as though most people who play the type of game I made enjoy the time period I based my game in. The visuals in medieval games can be greatly exaggerated to what they were, with giant fortresses and fires. It leaves a lot of room for creativity and story development.

My main source of inspiration for game mechanics was World of Warcraft (WoW). I really enjoy the teleportation aspects and flight paths in WoW, as well as tasks involving moving platforms and jump

pads (which I didn't get a chance to incorporate). I also enjoy how climate changes and environment changes in WoW can be so drastic within walking distance, which is what I included in my game from the grassy village to my frozen prison. Another great source of inspiration was Dark Souls II & III. The medieval themes and visuals are breath taking and I really wanted to have the same reaction from my audience. I ensured to have a huge castle wall and a big area to explore with beautiful lighting and environment. Dark Souls III in particular has a great skybox and lighting effects which really inspired all the lights in my game, like the glowing trees in the winter forest and my village, as well as the mushroom circle. DSIII has a lot of depth because of the lighting choices and I wanted to recreate that as much as I could.

Learning Methods

If I were to create another game, I feel as though I have learnt so much through trial and error, YouTube, forums and asking questions in class, I would be able to make the level I made over the last three months in a couple of weeks. I found a channel on YouTube which covered a lot of beginner's techniques when making menus, timeline nodes, basic nodes and variables. I didn't understand what any of these words meant as we were thrown into the deep end in class making complicated code, which made me want to learn and understand as quick as possible. I spent a lot of time doing trial and error techniques to try and make things work/look great within my game. I have developed my skills greatly in this software, as I had never used the software before September, and I have now found myself being asked for help by all of my peers which is nice.

Strengths

One of my strengths within this project, as well as all the other projects is my skills using photoshop. It really helps when designing as I can quickly draw what is in my head. It doesn't take a while to manipulate, and I'm well-adjusted with the tools to generate the ideas in my head into real life. When creating widgets and menu screens, I used my graphic design skills to create a good-looking menu and incorporated that into my game. I have done a lot of concept art and logo design in the past, so this was a skill that came naturally to me. I have a very artistic mind as well as an academic one, so this really benefitted me when coding and making my game look visually appealing.

Another strength is my perfectionism, I knew when starting this project, I wouldn't be satisfied having a half-finished game, that's why I worked so hard to get it as done as possible and try and incorporate as much as I could visually and mechanically. My time management was also an area

which I felt I did well in, usually my time management is very poor, however I enjoyed this project a lot, so I worked hard to spend at least three to four nights a week (aside from the days in university) doing things here and there on my game at home.

Weaknesses

One thing I really struggled with in this project was figuring out variables and calling event dispatchers etc. It really took me a while to understand how they work and what they do, I'm not still not extremely confident using them, however I feel a bit freer to play around with them and not feel scared if my code doesn't work the first time. It took time to recognize errors and instantly know what the solution was or where to fix it, however, now if something goes wrong in my code, for example when my character wouldn't flop during her death event, I knew I hadn't set 'simulate physics'. Working with the program so much has helped to develop my knowledge. Another weakness which I still struggle with is the animation of AI. I do have functioning AI within my game, however it took a long time and I wouldn't be confident doing it again without someone with me or a YouTube video taking me through every step again. I understand parts of it, but not collectively. For example, I understand how the blend spaces work and how to do it, but I don't understand how to attach that to an AI and make it move around.

Peer Review

I asked three of my peers some questions about my game, to see what I could improve on, and what was good about my game and menu system. I asked three questions for three topics, my menu system, gameplay, and visuals.

- How easy/hard did you find my menu to navigate?
- Maybe adding sounds to the buttons or some title music?
- Does my menu fit in with my game?
- Did you enjoy or not enjoy the gameplay?
- Was the game challenging or easy?
- Would you add anything else?
- Was the game's visuals appealing or unappealing?
- Favorite visual aspect?
- What else would you add to improve visuals?
- Anything else you would like to comment on?

All three peers found my menu easy to navigate, and clear to understand. Two peers said the style of the menu and the glowing buttons made it easier to navigate as they knew what they were clicking on. I made my buttons big and clear for this purpose and kept the original menu page simple. All three also agreed that adding title music and sound to the buttons would top off the menu, which is something I didn't have time for, but I will do in the future. All three also said my menu fits the theme of my game perfectly and enjoyed the icy theme all together.

All three of my peers said that they did enjoy the gameplay, due to the exploring factor and the visuals. One said it was 'rewarding' and 'beautiful', another said it had a 'lovely theme and environment' as well as a 'sense of achievement'. All agreed my game had a simple game play, due to the item collection being minimal, however the platforms at the end of the game were challenging, which was my intention. The three peers suggested three different things. Doors unlocking other areas, which is something I could incorporate easily using a trigger box, NPC's that fight, which is another aspect I intended on doing however my game went in a different direction. Lastly, one peer recommended a jump pad, which would fit with my game, but I ran out of time. I will add these into my level in the future as they are fun features I would love to have within my level.

I asked my peers if my game was visually appealing. This is an important factor to me as I feel as though I worked hard on making my level visually stimulating. All three of my peers said they enjoyed the visuals, and loved the snow and ice theme, as well as the lighting. Most peers adored the way my areas all link together following my icy cold December aesthetic, and one even said the particle effect of the snow tied it all together. I asked if they could think of anything that would improve my visuals; one peer said to have a snow storm of some kind at the end of my game to wrap it up, however I ran out of time to create anything big right at the end, but I would love to do something like this. Another peer said more ambient lighting in some areas would look more appealing, and maybe more majesty to my castle to have a bigger impact.

To conclude, judging from my own work, and my peer's comments, I have met the brief in completing a level that takes the player from point A to point B, with a successful functionality of item collection, moving platforms, teleportation, AI roaming, stunning game design and character animation. I have developed my skills greatly throughout this project and I am extremely proud of how much I have learnt and cannot wait to delve further into Unreal Engine next term to create even more spectacular looking environments and levels.

Gameplay Visuals

Starting Zone

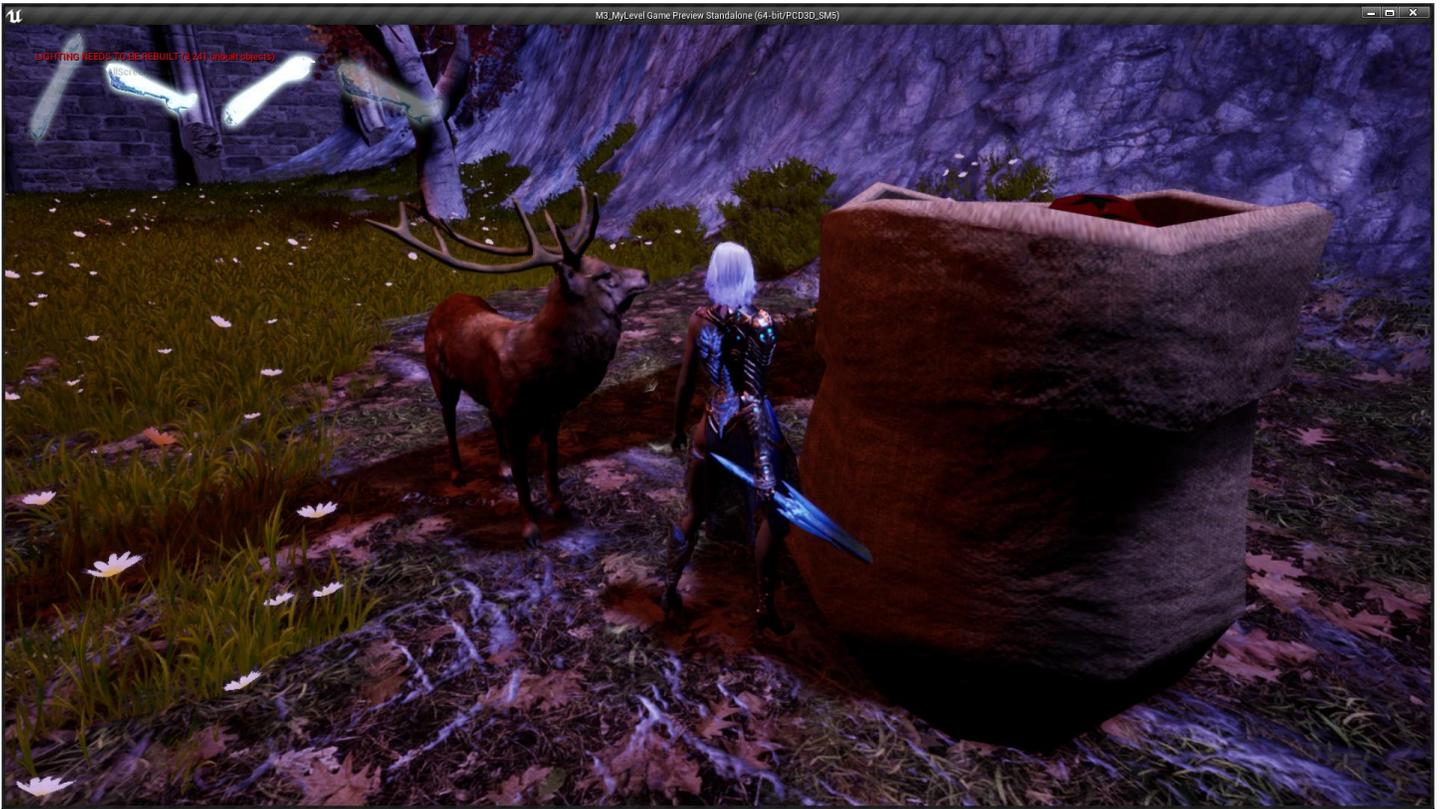




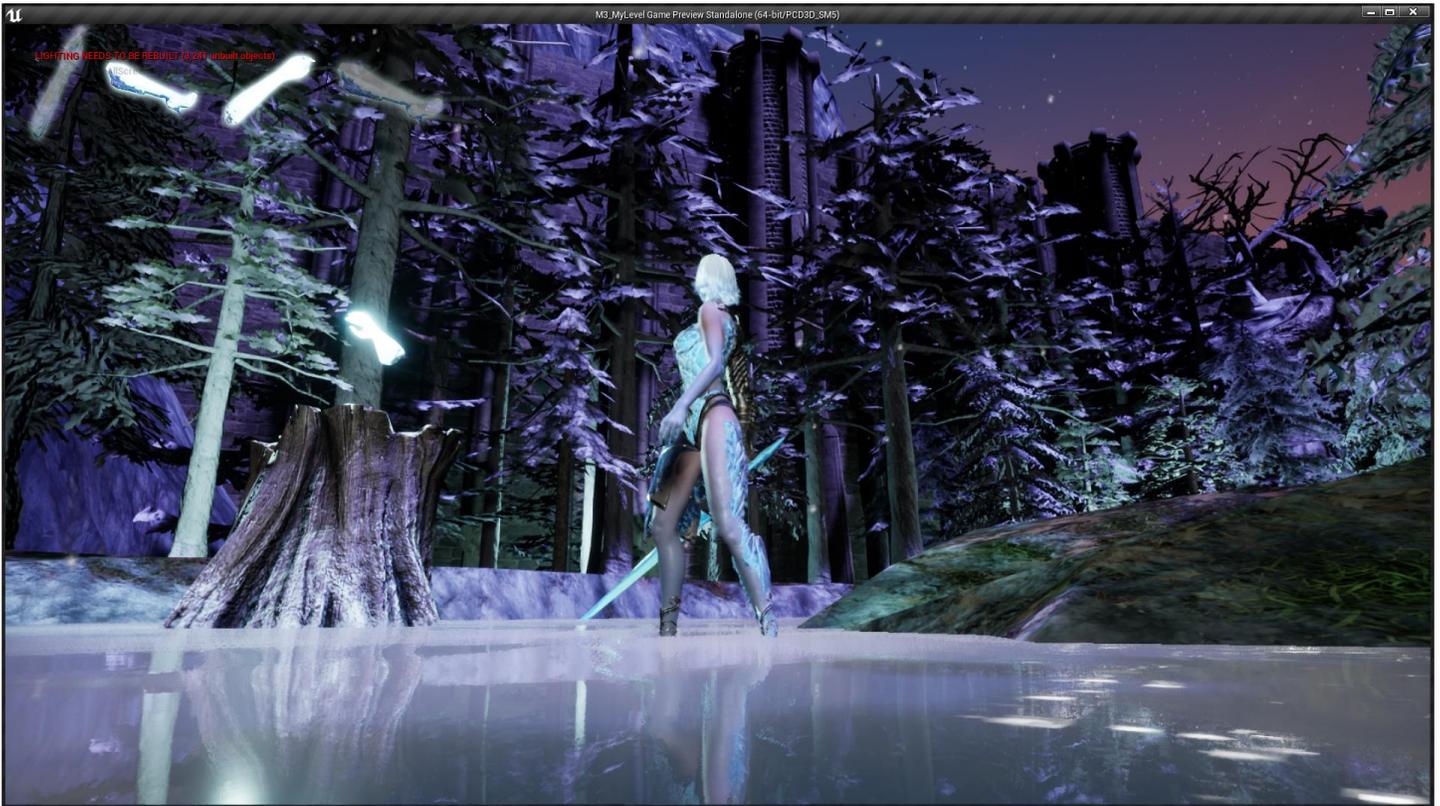
Castle Gates



Eternal Autumn Forest



Eternal Winter Forest



The Ruins



Ice Prison



