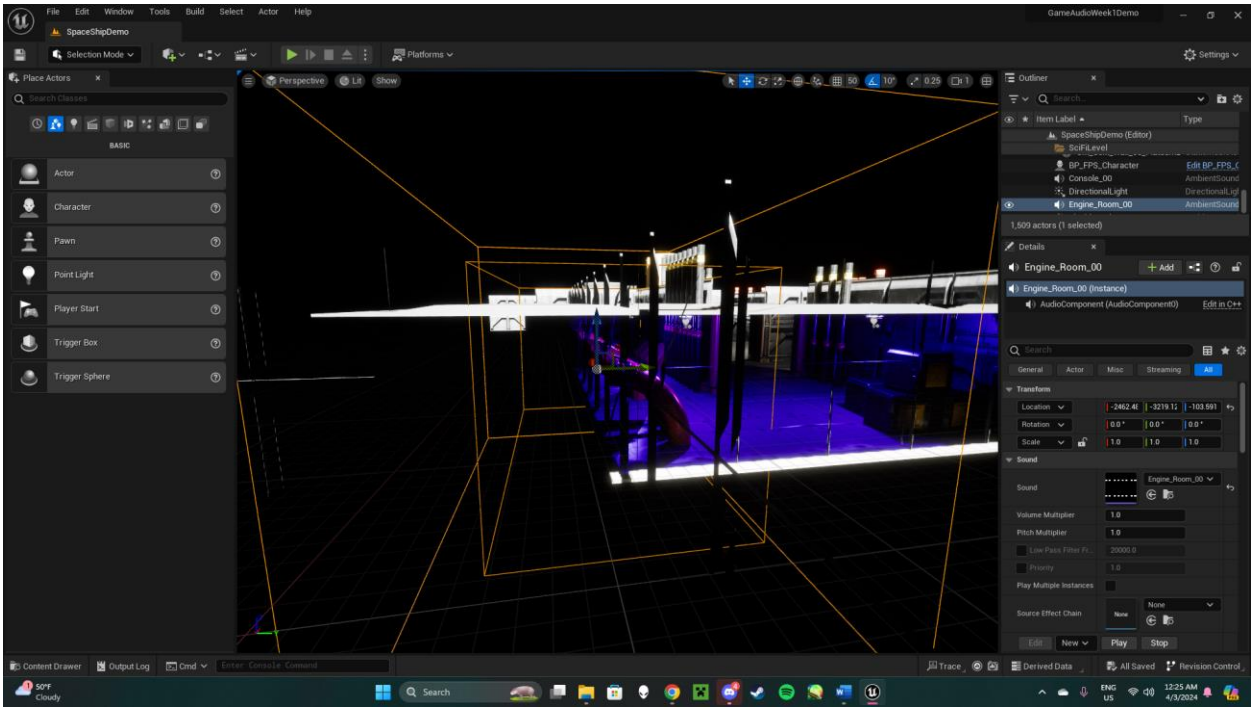
COMP-1871 REFLECTIVE LOG

Kirk Hogden, 001115381

Contents

Lab Task 01

Ambience



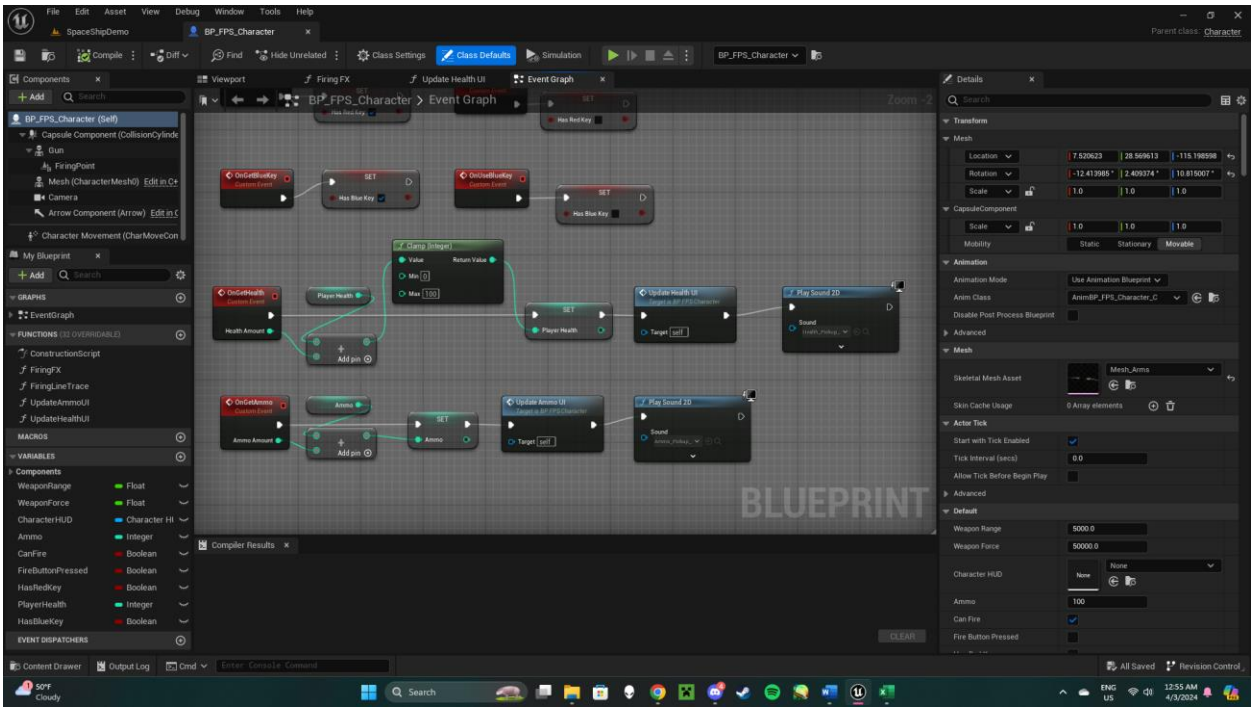
After getting comfortable with the attenuation system, I discovered that audio game objects don’t have to strictly have to use spheres unlike in Unity. Using the box shape for the inner/outer radius attenuation settings, it helped me to prepare sounds related to ambience. When giving ambience to the engine room, I was able to fit the whole box radius into the room to make sure the player only hears those sounds when in that area.

Game Analysis

Bomberman 64 was my chosen game because the limitations of audio in older game titles. When filling in the descriptions, I explained short summaries of the purposes for why those sounds were playing, as to show what they indicate towards the player.

Challenges

Transferring from Unity to Unreal took a while to get familiar with. How both game engines approach sound are different. Instead of creating a audio source and applying a sound clip to it, Unreal Engine has each sound file contain their own properties. Once I started to understand how to use the tools Unreal Engine provides, I was then able to apply sound settings with ease.



This would also by my first time coding through event graphs. While it felt intimidating at first, the lab tasks helped me understand how to get specified audio clips to play when an event is called. Reading the name of the nodes and understanding the concept that instructions run through node whiles helped me figure out what to do with the PlaySound2D node.

The chosen gameplay of Bomberman 64 contained many sound queues happening all at the same time. This proved difficult to keep up with, however it allowed for more sounds to record in a short span of time.

Lab Task 02

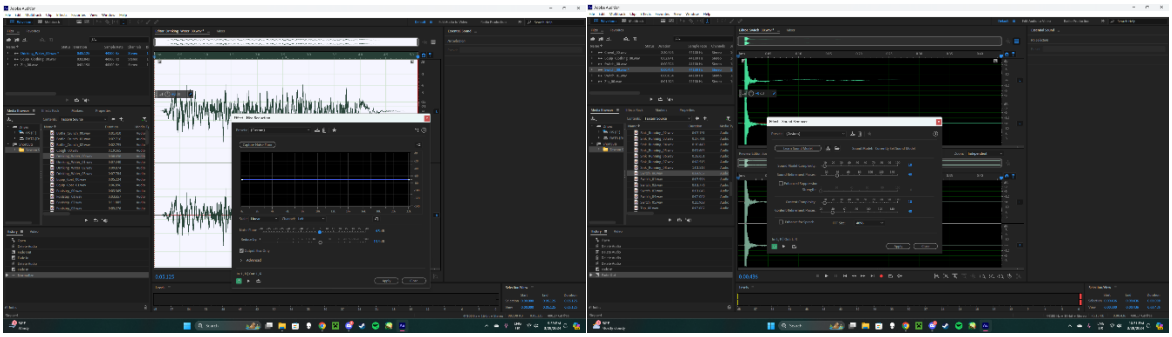
Sound Recording

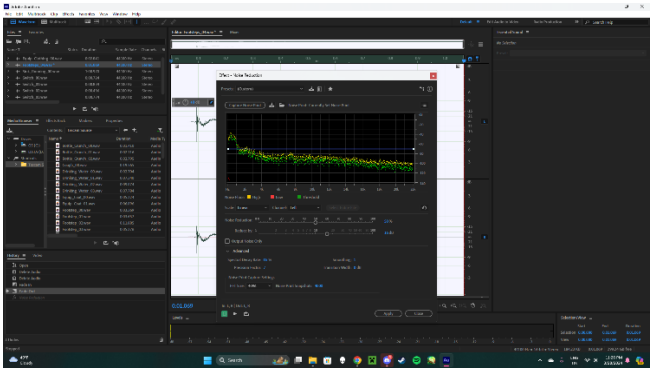
To capture sounds outside, I chose a Tascam paired with earphones to hear back recordings with. When it came to ambient sounds, I would capture a minimum of two minutes considering given feedback from another student that shorter ambient audio files come off as repetitive. I took advantage of a water bottle I purchased and recorded sounds of myself drinking from it, as I figured it would suit for survival games that require hydration. The bottle cap would provide useful for collision sounds, as I'd tap it on different surfaces such as metal or concrete.

With the sounds recorded, a USB cable was used to connect the Tascam into a computer. This allowed me to drag the files from the micro SD onto the computer. I kept all the files as more sounds meant different opportunities.

Sound Clean-Up

For all sounds, fade in and out effects were applied for all Tascam recordings used, This assured no pops can be heard in the clips. Before using the normalise effect, various sound removing features were used. Using the noise reduction tool helped remove unwanted background noise that could be heard in the Tascam recordings.





While noise reduction was used for removing most background noise in sounds, hiss reduction and sound remover worked better with other sound files.

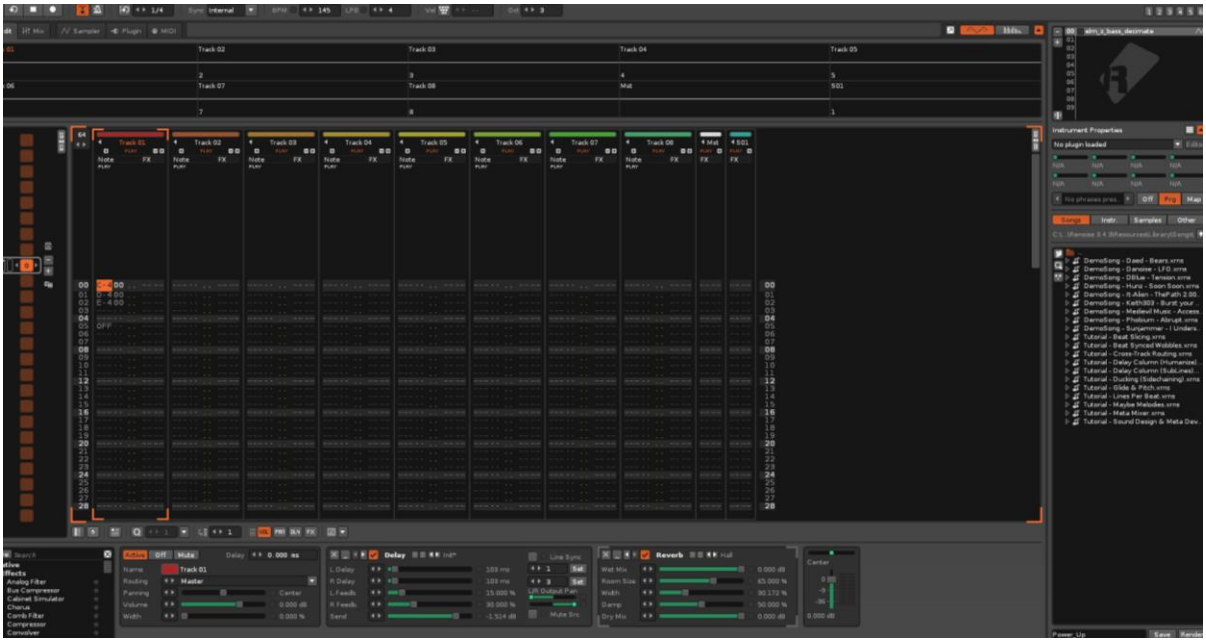
Challenges

Before taking recordings with the Tascam, it was worth learning how to use one first and getting some practice before I felt confident going outside with it. When getting recordings of places such as the Greenwich Market, I avoided spots where I was likely to capture other people's conversations. This proved difficult to do as I wanted to make sure the ambience audios were past two minutes, but with so many people in one area, some recordings included people walking past the Tascam while talking.

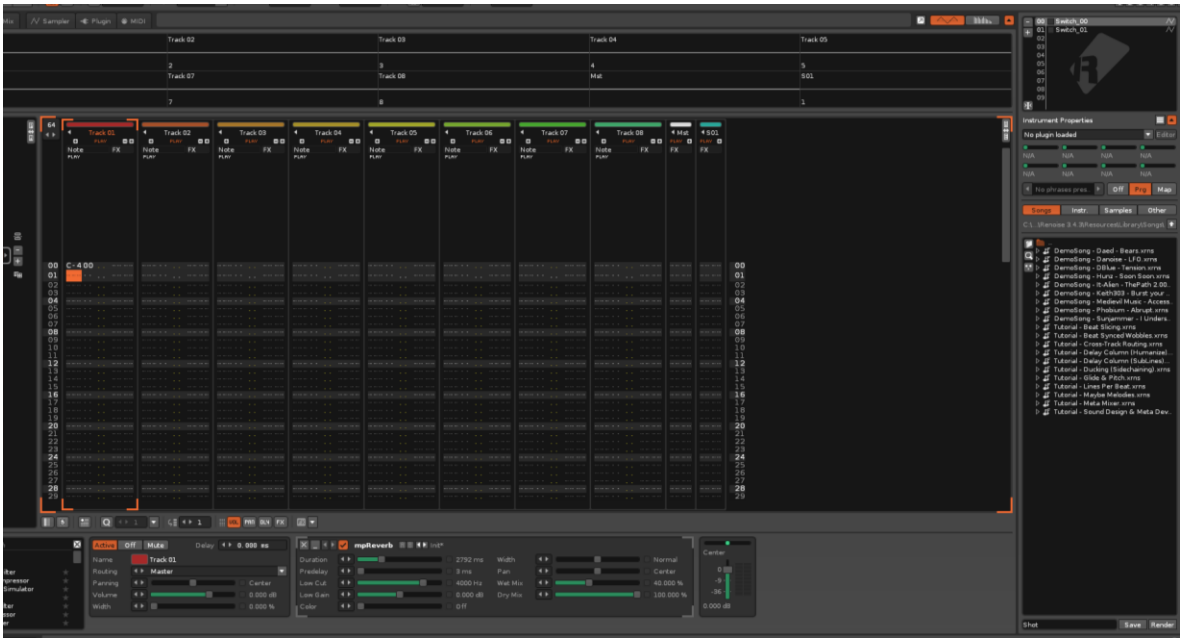
Lab Task 03

DSP Effects

I took different instrument sounds and explored with the DSP effects provided. I would play around with them until I understood what each effect and feature did.



The power up sound effect uses the combination of the reverb and delay DSP effects on a bass instrument. This gave a old school feel for gaining some form of power up



Using the switch clicking sounds I recorded with the Tascam recorder, I had the idea of recreating a gunshot sound by getting it to echo. The Reverb was used, however in the end it didn’t sound how I hoped it would. It was kept regardless as it could hold other potential, such as something dropping in a distant cave.

Modulation

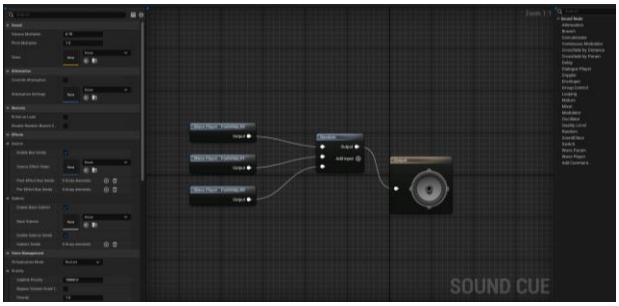
To create the blizzard sound, I took a noise sound effect and applied a pitch envelope on the modulation. Using the graph, I created a wavy pattern so that the pitch adjusts. This gives the realistic sound of wind changing strength as it blows. The envelope length had to be increased by a huge number, so that the change of pitch could last for a realistic set of time for a blizzard.

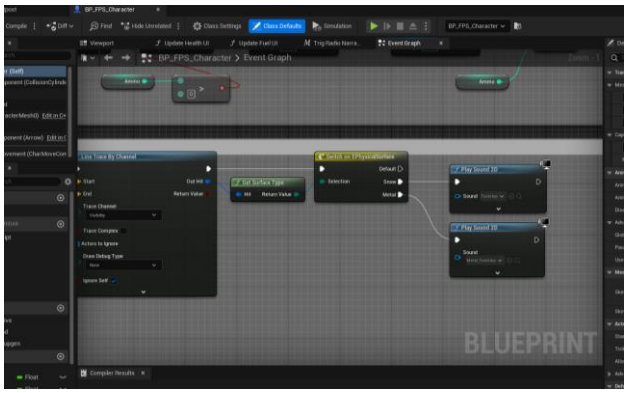
Challenges

No Tascam recordings were done during the production of most sound effects in this week, so I had to try DSP effects using instrumental sounds. Without these recordings, the most I could do is pick a random instrument and try to figure out how I could manipulate it into sounding like something else. This challenge proved beneficial as it boosted me to be creative and use my imagination.

Lab Task 04

Sound Cues



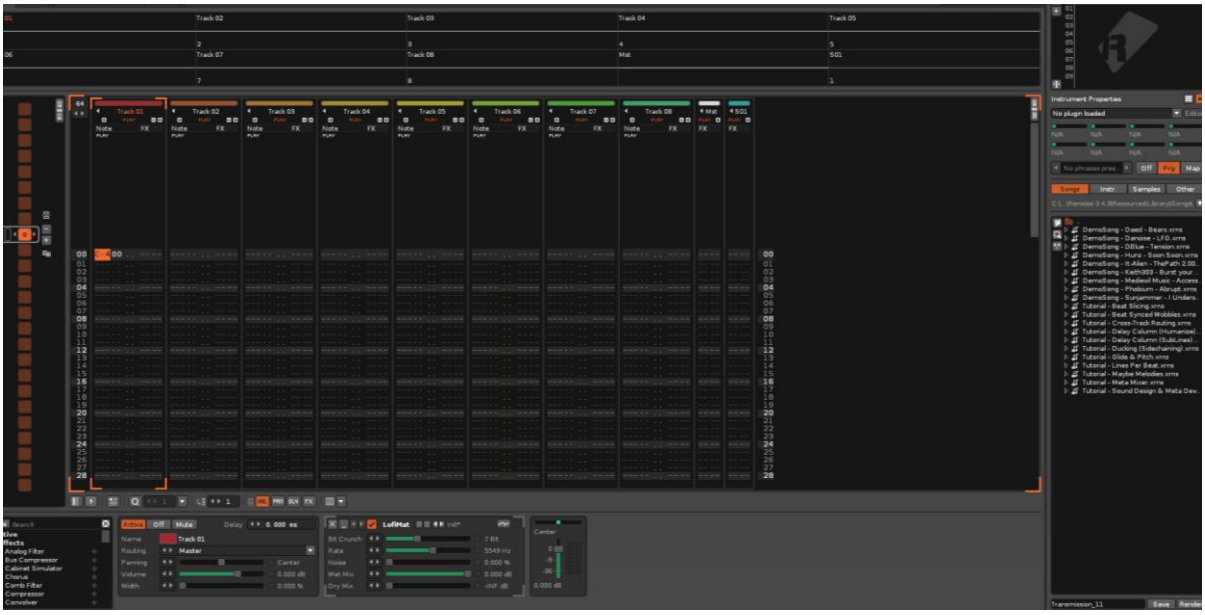


By creating a sound cue for footsteps, a random node was inserted which allowed me to connect multiple wave players into it. This made coding easier as I could simply refer to the sound cue to play, rather than coding the footsteps to play a different audio file each time. The same scenario was applied for metal footsteps.

Environmental Sounds

I made use of the blizzard sound composed in lab task three, as I felt it fitted the scene. The volume multiplier had to be lowered down so that other sounds can be heard over it. Throughout gameplay, I couldn't notice when the sound effect was looping.

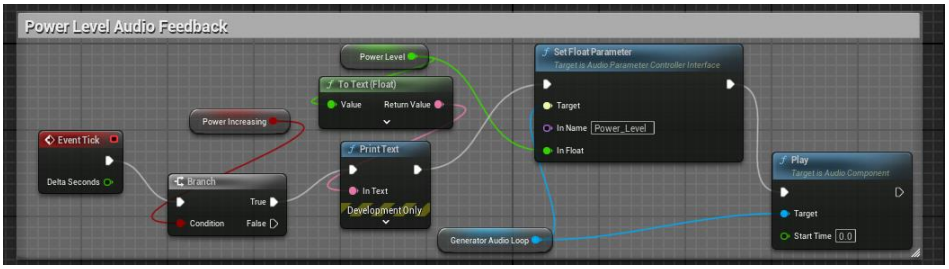
Radio Transmissions

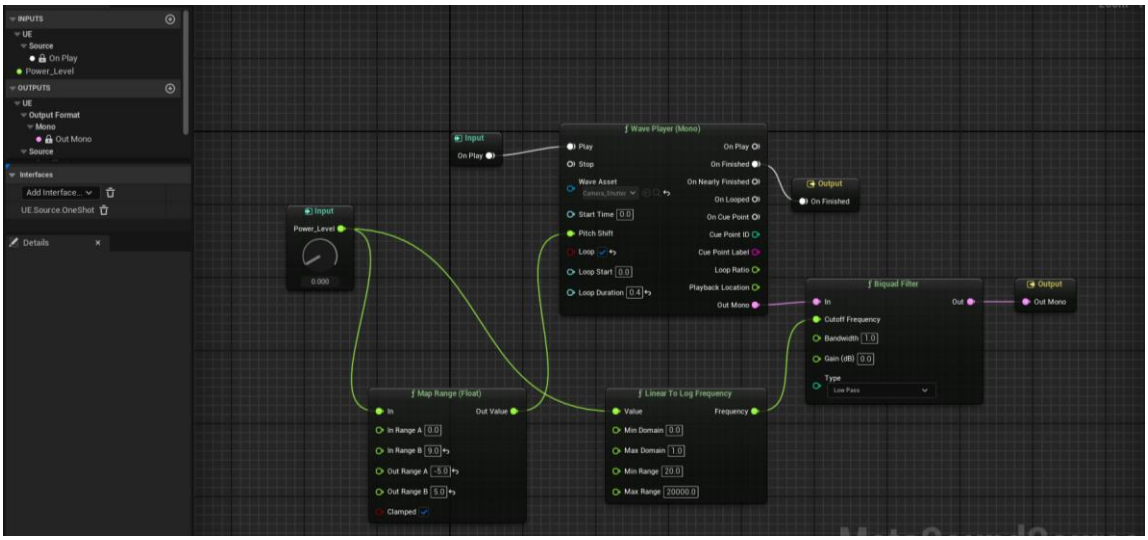


I took each recording into Renoise and would apply a LofiMat DPS effect. I chose this effect because by using the bit crush slider, I could distort the transmission audio to make it sound as if it were coming from a radio with low signal. The bit crush was kept at a reasonably low value, as making it too low could make it difficult to make out what is being said.

Lab Task 05

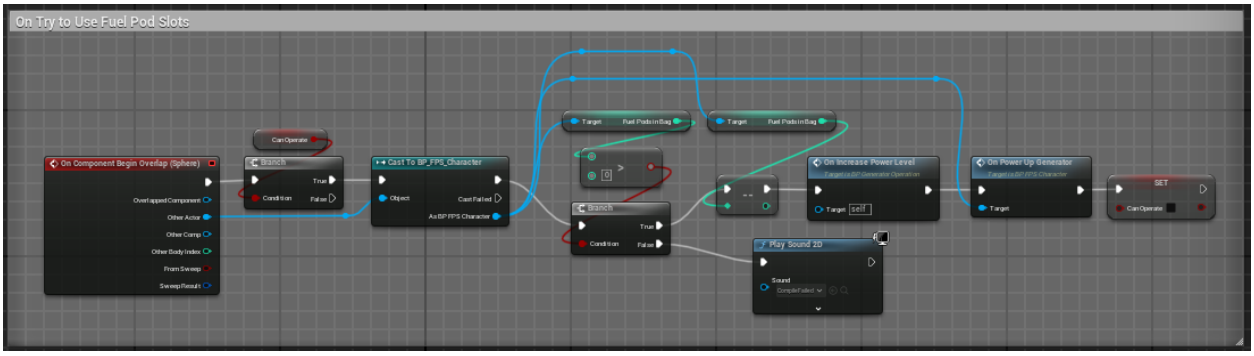
Generator Sounds

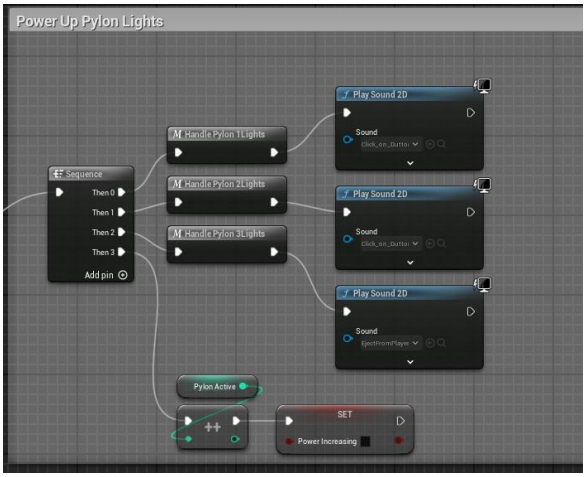




A meta sound source was created to give interactivity for the generator audio. This meta sound source contains a input that stores the value of how much the generator is charged by. The input value controls the pitch of wave player which plays the charging sound clip. For better control over the range of the pitch value, the input node is fed through a map range node which clamps the range of the final calculated pitch.

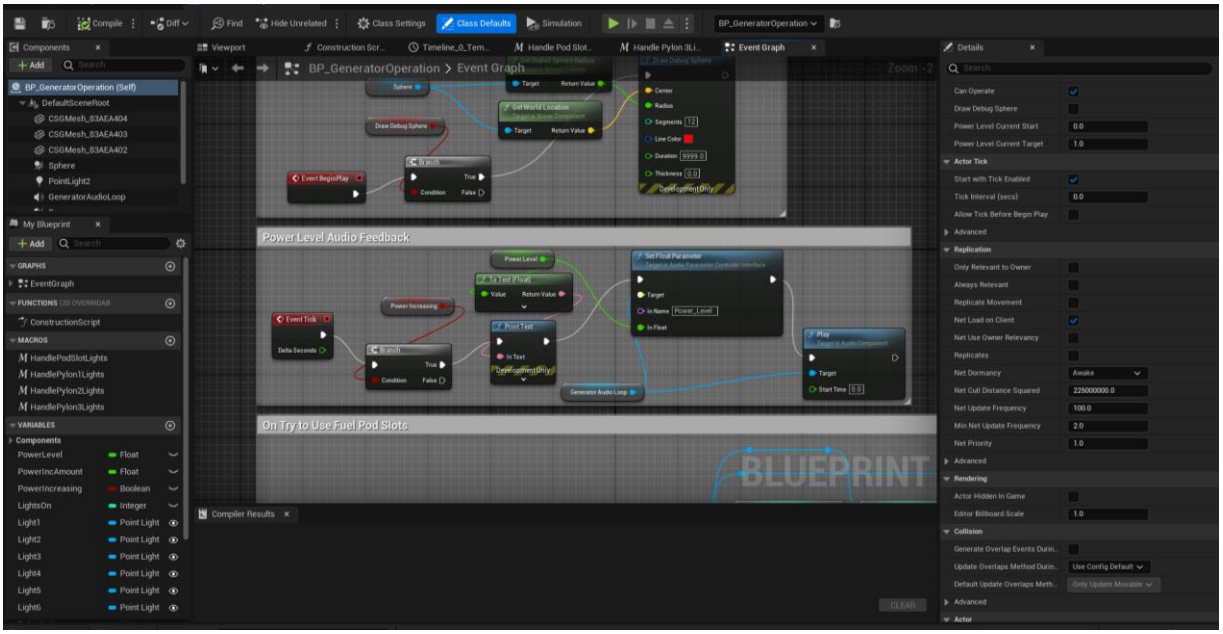
The input node receives its data from the generator's event graph, in which it updates whenever the player feeds a fuel pod into it. This is done through the set float parameter which outputs the data to the meta sound source.





The error sound for when the player walks up to the generator with no fuel pod required the use of the branch node. The sequencer allowed me to user multiple audio clips depending on the amount of fuel pods being placed into the generator. Because of this, I used a different audio clip when the final fuel pod is placed in, which acts as a signal that the player completed their objective.

Challenges



I was unable to get the generator sound to play when it was in the process of charging up. Different methods I'd try would make the situation worse, where sometimes the charging sound wouldn't play at all. Eventually I left the event graph in a state where the generator plays sound when it's not charging up, at least having a difference in pitch depending on how many fuel pods it had. The issue could possibly be that the play node is being repeatedly called when the generator is charging up.

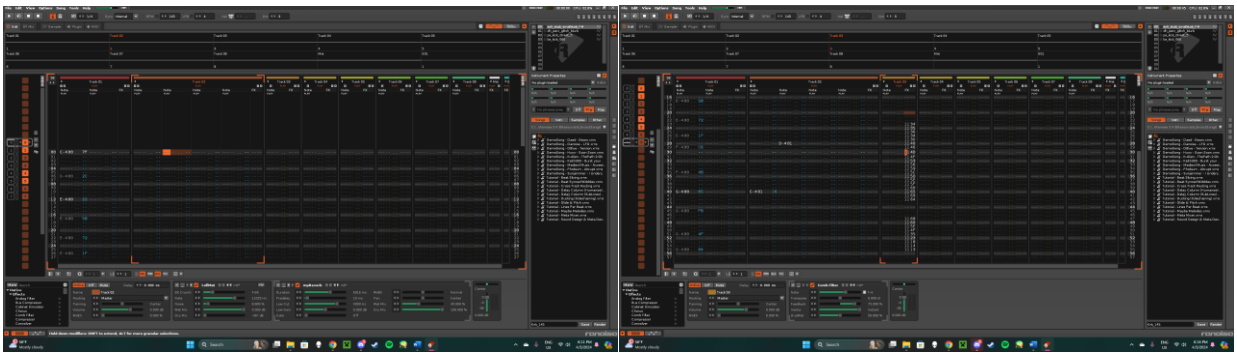
Lab Task 06

Idea

With the image on Moodle, I could vision an action-adventure game that would involve racing against time. I aimed to make a music theme that could support the mood of running through a colourful course while immersed with beating a goal in time.

DSP Effects/Tempo

The tempo was adjusted to be 145. This helped increase the pace of the music and fit the mood of the player rushing to race through the course. When trying to increase it higher, this resulted in the loop being too short, which could become too repetitive for the player's ears.



For each track, tried different DSP effects to see what suited most for each instrument. Other instruments I would combine with multiple effects to give a unique sound. Overall, I would experiment by removing or inserting different effects, then tweaking the settings to achieve further results.

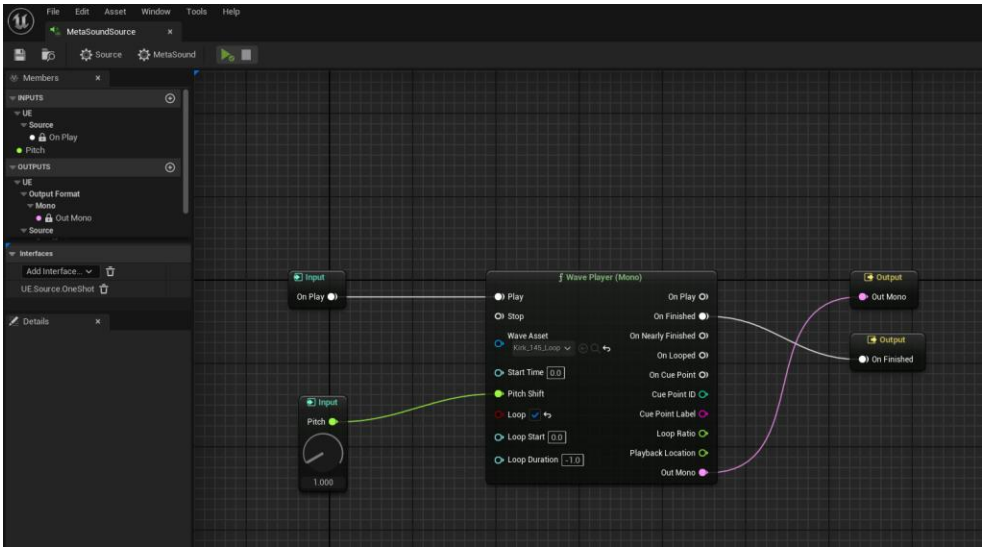
During track three, I applied a comb filter and allowed the instrument to change the note value over time by recording and holding down right click on that DSP. This created a robotic-like sound effect to play every few seconds in the soundtrack.

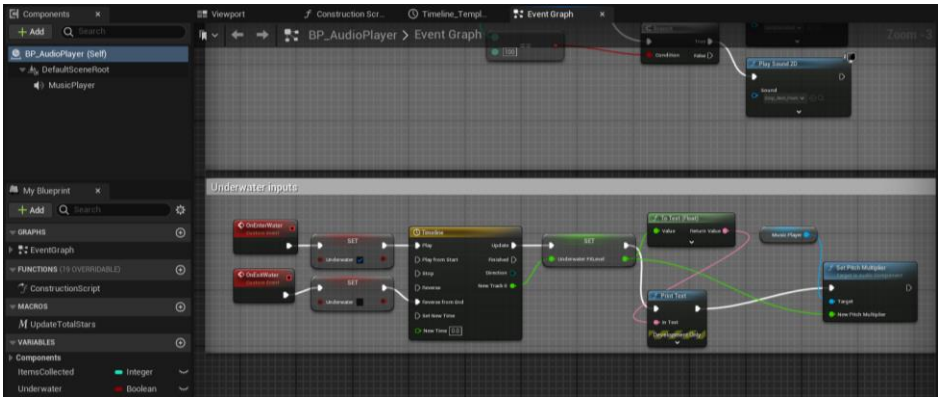
Challenges

This was the first soundtrack I was creating with Renoise, as my prior times using it would be for creating unique sound effects. It took a while trying to understand how to make audio that could last longer. I eventually learned how to create sections in the song, but found that I had to copy and paste the same melodies from each track by each note. This came off as time consuming.

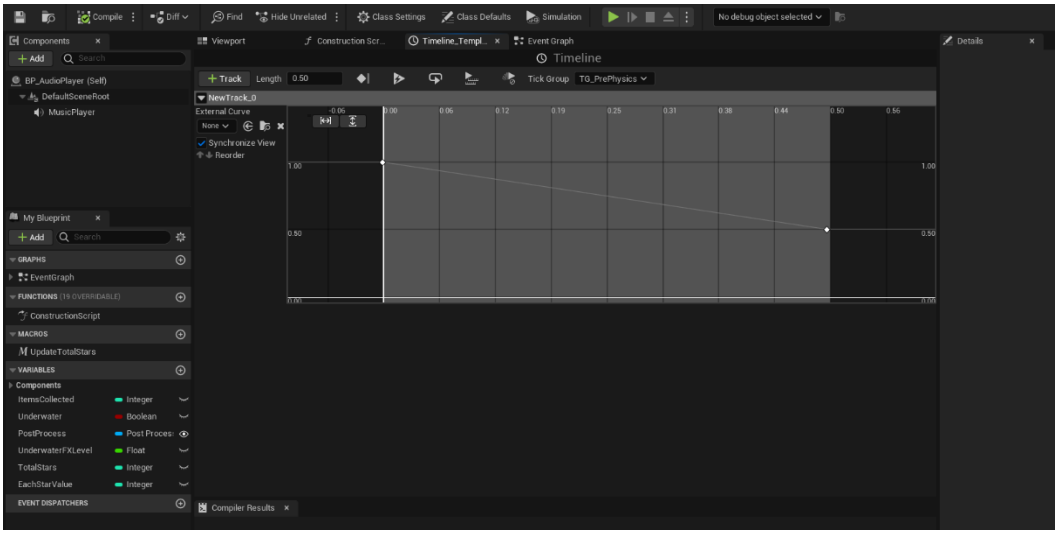
Lab Task 07

Adaptive Music





A meta sound source was created for the music, which would allow control over it's parameters in coding. A pitch input node was added so that the music could adapt depending on if the player was submerged in water or not. Although the meta sound was used to play music from, the audio's pitch multiplier was used instead in the end to control the pitch.



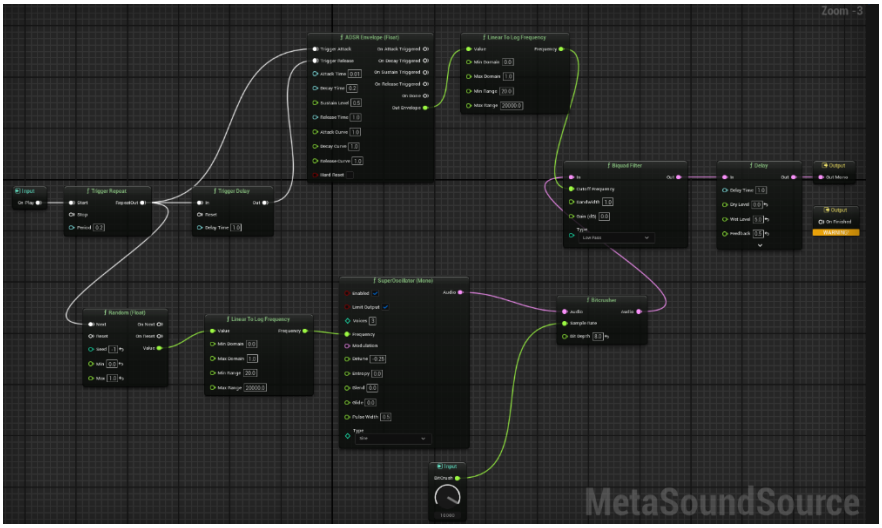
The timeline editor had to be adjusted, so that the music could play at a normal pitch when on land, and at a lower pitch when submerging into water. The starting point was dragged up to one, and the end point was lowered down to zero.

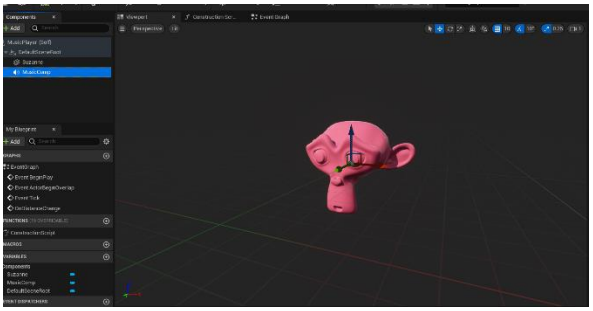
Challenges

The music file plays through a meta sound source because originally I planned to adjust the pitch through float parameters. Whenever the pitch value were to change however, it’d cause the music to start from beginning. In the end, the pitch multiplier attribute of the audio component was adjusted instead. Alongside the pitch multiplier, the volume multiplier was lowered so that sound effects could be heard easier.

Lab Task 08

Generative Music

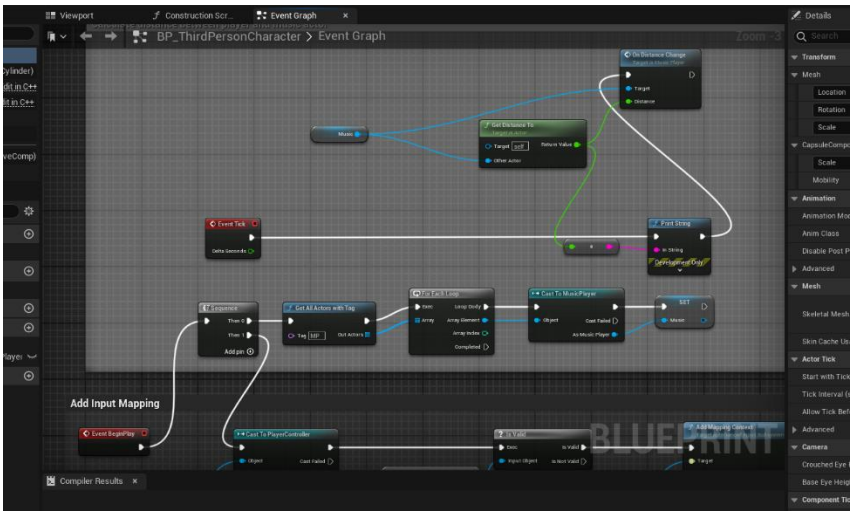




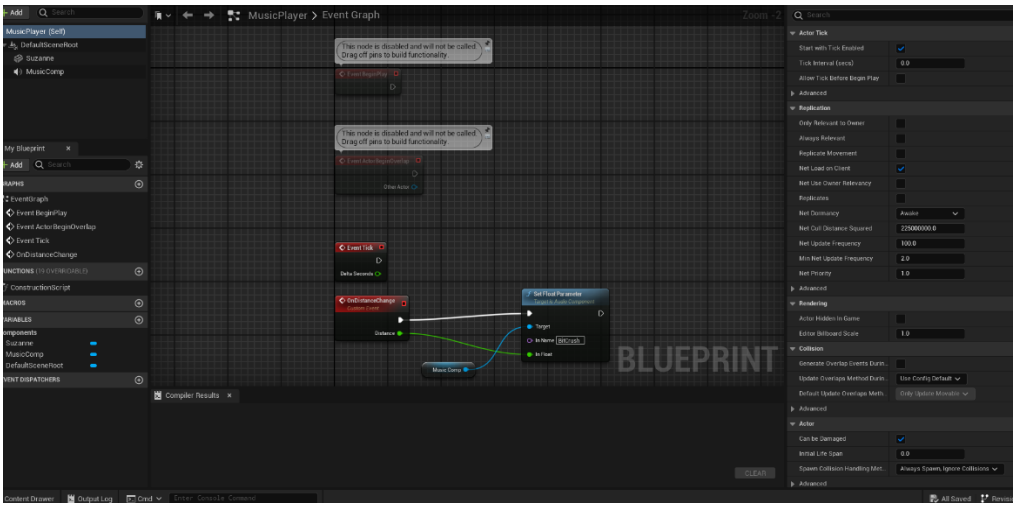
A synthetic generative music system was created, which uses a random seed whenever it’s executed. This was created for the purpose of making soundtracks seem less repetitive for the user when they’re gaming, allowing them to feel immersed every time they game. To further improve this, interactivity was applied to the music in which it starts to sound more distorted when the player gets closer towards a Suzanne mesh.

Music distortion was done through a bit crush node in the meta sound source with a input parameter to control the sample rate. The player class calculates the distance between it and the Suzanne mesh. The Suzanne class uses a custom event to take the return value and input into the meta sound source.

Challenges



Being new to event graph coding, it took a while trying to implement the mechanic of making the music become distorted based on the player's distance to the Suzanne mesh. Different nodes were explored to try find a reference for the Suzanne instance, but most didn't work.



Eventually, a working distance calculation system was implemented, but it wasn't feeding into the generative music's meta sound source. This was when the idea came of having the Suzanne class send the float value instead through a custom event.

Before choosing the sample rate to be interactive, I tried connecting the node wires to the bit depth of the bit crush node instead. It was hard to try and make the music sound noticeably different on player distance with this, because it would only start to sound distorted when the value got near to one.

Appendix A

Health pickup, Freesound.org, October 28th 2013,

https://freesound.org/people/juancamiloorjuela/sounds/204318/

Ammo pickup, Freesound.org, October 10th 2022,

https://freesound.org/people/BBBBilly/sounds/653032/

Computer Console, Freesound.org, November 11th 2020,

https://freesound.org/people/unfa/sounds/543968/

Engine Room, Freesound.org, January 15th 2021,

https://freesound.org/people/FiveBrosStopMosYT/sounds/554414/

Water Tank, Freesound.org, March 23rd 2021, https://freesound.org/people/titi2/sounds/564426/

Steam Pipe, Freesound.org, July 5th 2015,

https://freesound.org/people/visions68/sounds/278999/

Coding assistance for finding distance between actors, December 6th 2015,

https://forums.unrealengine.com/t/get-distance-from-one-player-to-an-npc/49283