1-5-2024

AUD216 Project 2 Example

SAE University College

Recommended Citation

SAE University College. (2024).*AUD216 Project 2 Example*[Student Exemplar, SAE University]. Creo.

AUD216 Project 2:

Sound Design and Middleware Implementation

Your Name: [redacted]

Your Email: [redacted]

Facilitator Name: [redacted]

Campus: Perth

[redacted]

[redacted]

Table Of Contents

Game audio implementation and audio middleware

Within Our last project we had a focus on implementing game audio directly within the game creation engine, when thinking about being an audio engineer practically, and working amongst multiple clients using multiple game engine creation, it doesn't make sense to only know how to use unity, so we use middleware.

What is middleware ?
Middleware is "software that different applications use to communicate with each other" (Amazon,n.d), they are made to provide a sort of functionality to be able to join applications so that tasks can be done more efficiently, acting as a bridge between very diverse technologies, tools and databases so it can be integrated into one system (Amazon, n.d). The importance of middleware comes from the fact that instead of developers having to focus on developing a system to enable connectivity between different components of software, having to build a system within the application which exchanges data. Rather they can focus on business logic like marketing and refining important features within their application

How Does It Work ?
Middleware works by enabling developers build apps without having to create customs systems within their application every time they need to connect certain components (In our case it would be connecting the actual game play to audio), middleware works by providing the service to communicate through common computer messaging frameworks which is types of computer coding, like different languages computers speak in like Javascript or C#. (IBM, n.d.) Due to being able to speak the same coding language as the application you wish to integrate certain features in the application like audio triggers when a certain action happens within a game, it is able to handle the communication of data to what the user will be seeing within the application, like connecting a sound to a player's movement. It is able to do all this without having to install the middle where or the engine on either side of the creation process (Williams, 2022)

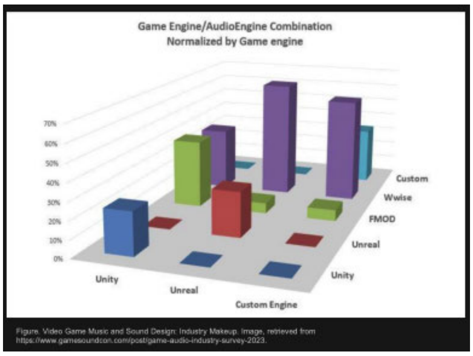


Figure. Video Game Music and Sound Design: Industry Makeup. Image, retrieved from https://www.gamesoundcon.com/post/game-audio-industry-survey-2023.

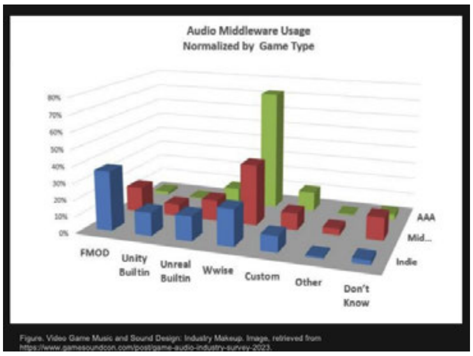


Figure. Video Game Music and Sound Design: Industry Makeup. Image, retrieved from https://www.gamesoundcon.com/post/game-audio-industry-survey-2023.

Why Use Audio Middleware for Game Audio Implementation ?
When there are audio tools within game engine applications, why is it that almost all Triple A game studios and a large number of independent game developers use established audio middleware, the graphs being a testament to this fact seeing that audio middlewares like Wwise and FMOD have the largest use within all types of game development. This is because although game engines have a good simple audio tools which work for audio implementation within some games but the level of audio implementation that games require within today's game market is too much for most in engine sound implementation tools, there is also the option for a programmer or developer to create a whole new system of middleware custom made for a specific game, but this means the sound designer within the game will

have to learn how to use it from scratch and then deal with most the bugs that come with creating any new sort of software leading to wasting time due to the sound designer having to communicate with the programmers rather than having a programmer who is already confident within a certain audio implementation middleware like Wwise saving precious time (Nogueira,2019)

Using Middleware for audio implementation within games is good because it allows you to ;

Multiply and Diversify sound

Audio Middleware is able to improve sounds that need to be repeated so they never end up being the same sound over and over and the player doesn't get tired of them, this can apply to footsteps, ambiences, randomising vocal sounds of players or non playable characters and their respective dialogue. (Nogueira,2019) Most game audio middleware have support for Real Time Parameter controls allowing sound to change based on what is occurring in a game like a care sounding louder during acceleration and quieter during deceleration.

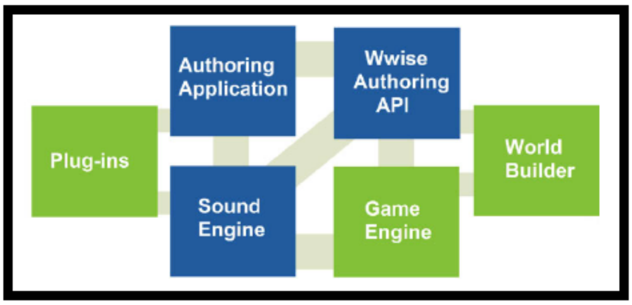
Making a Realistic Ambience

In audio Middlewares like FMOD and Wwise you are able to use "multiple base ambiences of a couple minutes and add multiple insects, birds and other nature sounds in a random order. Doing that, you are able to create a much more realistic-sounding location. And moreover, you can use middleware to smoothly change the time of day and transition between locations with a correct fade curve."(Nogueira,2019), this is useful in creating something more realistic rather than having a minute long ambience on loop that a player gets tired of because they stay within one area of a games map due to the game requirements

Adaptive Music

This is music within a game which is made and cut to fit certain points within the game organically to induce a certain emotion or aura through changing the tone, volume or rhythm. This is hard to do within most in engine game audio implementation tools but within FMOD and Wwise this process is streamlined for like when you are playing a stealth game and you can make a script within the middleware which cues dramatic music for when you are found. This is called called (Nogueira,2019)

Wwise



Wwise is the leading middleware for audio implementation for games made in today's market being used for across the game industry for nearly all operating systems like windows, Linux, MacOs, Playstation, Xbox, Nintendo, IOS and android being used for audio implementation within game like the Assassins creed franchise, Cyberpunk 2077 , The Halo games, Mortal combat and more.(Wwise,2023)

Wwise was made to ;

- provide a complete authoring solution meaning a complete system made to govern the sound across multiple game engines.(Wwise,2023)

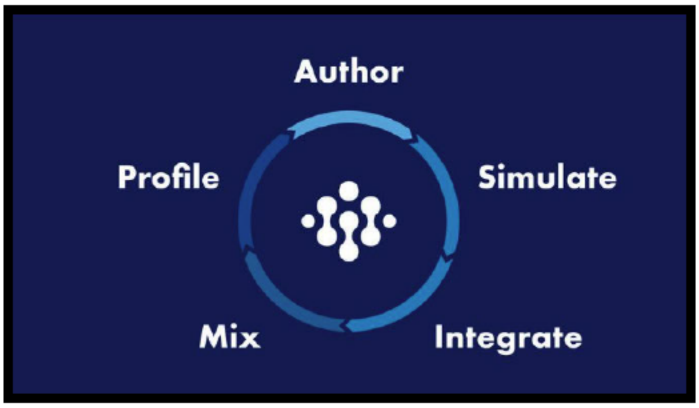
- Redefine production workflow for audio and motion ((Wwise,2023)

- Improve pipeline efficiency.(Wwise,2023)
- Push the boundaries of gameplay immersion (Wwise,2023)

Wwise is a powerful authoring/middleware application which works in a non linear fashion to create audio and motion assets to manage sound, music, motion and creating sound banks meaning it works in a non restrictive manner to work across the majority of popular game engines. It is able to use plugins for generation audio and motion like tone generators and create audio effects like reverb (Wwise,2023) or even things like spatialization which is made to apply the needed effect to a specific sound based on the player's environmental position.

Wwise has native support in 3D spatial audio which was introduced in 2021 which is exceptional for head tracking especially in VR games allowing you to for various XR's and also allows you to have HRTF which is head related transfer function allowing the sound to be realistic to our own hearing meaning that when we turn our head in game it doesn't turn off all the sound within one ear, always being able to hear a little out of one ear.

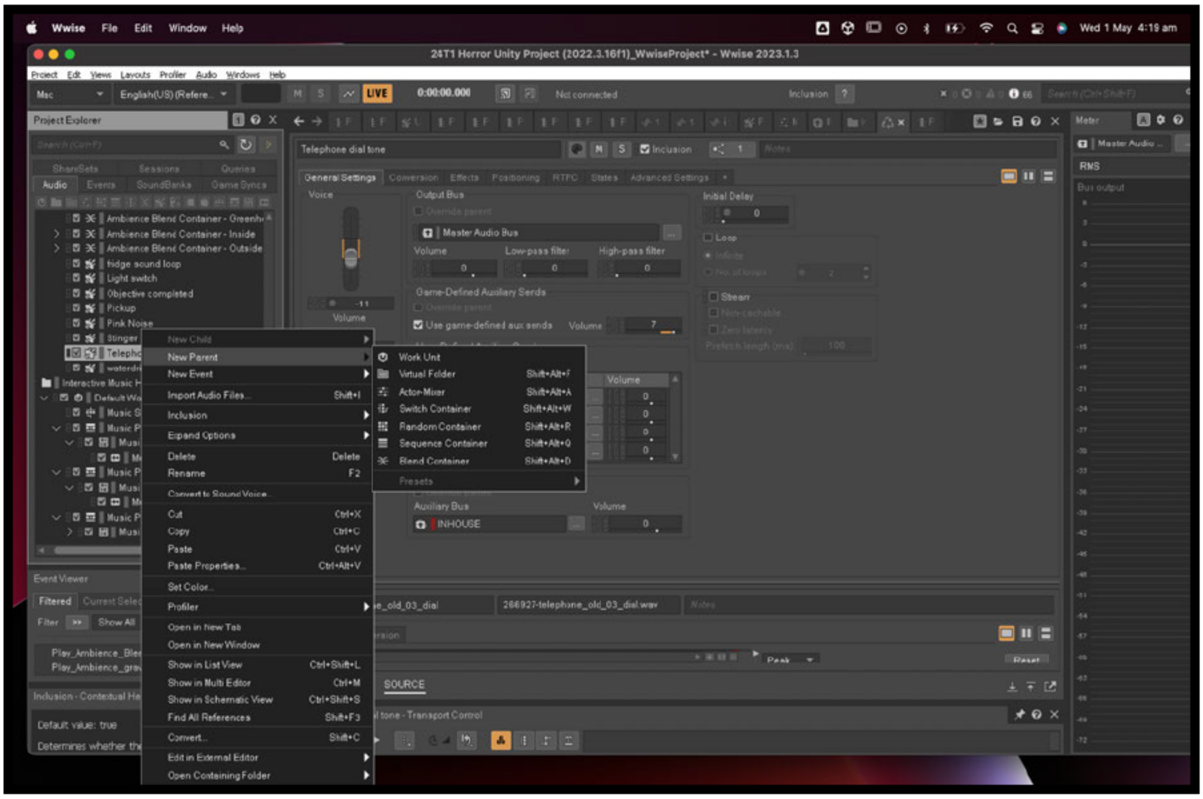
Wise creates a streamlined production pipeline for game audio implementation being able to tightly integrate tools to allow sound designers to perform a variety of tasks in real time within a game



- Author—build sound, motion, and music structures and define properties and behaviours.
- Simulate—validate artistic direction and simulate game play.
- Integrate—integrate early without additional programming.
- Mix—mix properties in game in real time.
- Profile—profile in real time to ensure game constraint compliance. (Wwise,2023)

Audio Asset Documentation

Before working within Wwise I did research into the parameters in which all my sounds would be going into.


24T1 Horror Unity Project (2022.3.16f1)_WwiseProject* - Wwise 2023.1.3
Project Explorer
Telephone dial tone
General Settings
Master Audio Bus
New Child
New Parent
New Event
Import Audio Files... Shift+I
Inclusion
Expand Options
Delete Delete
Rename F2
Convert to Sound Voice...
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste Properties... Ctrl+Alt+V
Set Color...
Profiler
Open in New Tab
Open in New Window
Show in List View Ctrl+Shift+L
Show in Multi Editor Ctrl+M
Show in Schematic View Ctrl+Shift+S
Find All References Shift+F3
Convert... Shift+C
Edit in External Editor
Open Containing Folder
Work Unit
Virtual Folder Shift+Alt+F
Actor-Mixer Shift+Alt+A
Switch Container Shift+Alt+W
Random Container Shift+Alt+R
Sequence Container Shift+Alt+Q
Blend Container Shift+Alt+D
Presets

When dragging your audio you need to create a parent (main folder), the childs are the sub components or contents of the parent folder. Depending on what you want your specific piece of audio to do within your game you need to select between various parents usually stemming from the work unit which is used to help organise all the elements within your project.

Virtual Folder - When I dragged an audio file into the virtual folder the objects placed within the folder are shown within the contents editor in Wwise meaning you can edit the volume, pitch and other parameters of the child objects whilst not leaving the view of the parent

Actor Mixer - When audio is dragged into the actor mixer parent it is grouped and organised to be able to control specific parameters of that sound like the actor mixer for footprints within my project would be able to turn down all the footsteps across the whole game

Switch Containers - This container is very good for something like footsteps within the game as it allows the sound designer to be able to make the decision of which sounds will be plated based of events within the game, like a wood foot step in a wood surface and a tile footstep sound on a tile surface

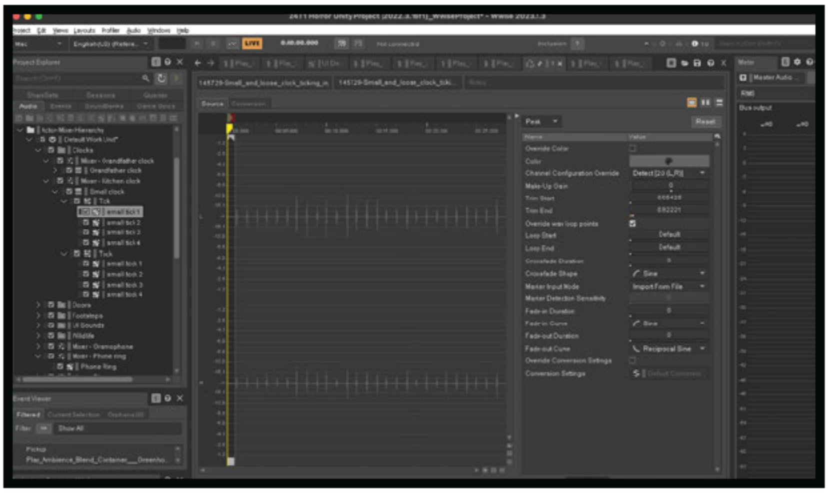
Random Containers - Usually placed within a switch container as a child, for something like the footsteps as it you input a various amount of times and it switches between the sounds allowing you to avoiding repeating the last however many sounds to avoid a repetitive sound

Sequence container - same as switch container but instead it is in a specific order that is desired by the designer rather then random,

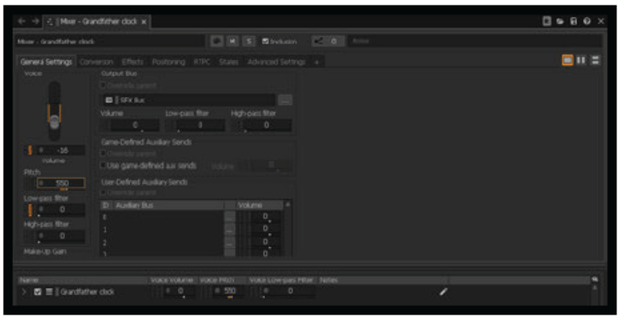
Blend Container - used to group multiple sounds so they are able to be heard at the same time

CLOCKS

The clocks are the first thing you hear upon entry within the haunted house, there were two clocks within the level, The Grandfather clock and the smaller kitchen clock.

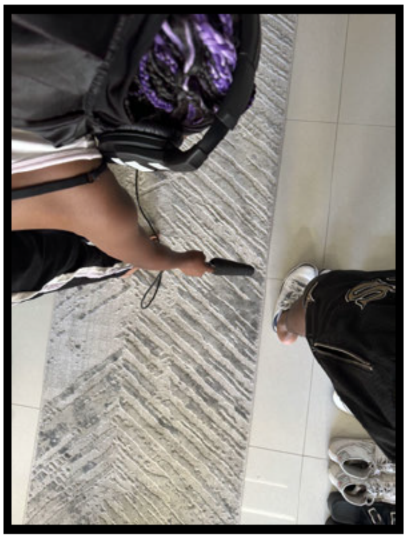


The grandfather clock audio was one long audio file which you don't want, so I went into source and basically cut the sound to 4 ticks and 4 tocks, and placed these within a sequence container and the ticks and tocks being in switch containers as children to the sequence container, this allows for the ticks and tocks to play one after the other and not be highly repetitive. Changed the Pitch of the Clock to 550 making it tick in game at about every one second. The sound for the grandfather clock was within the game files and for the smaller clock I used Soundsnap.



FOOTSTEPS

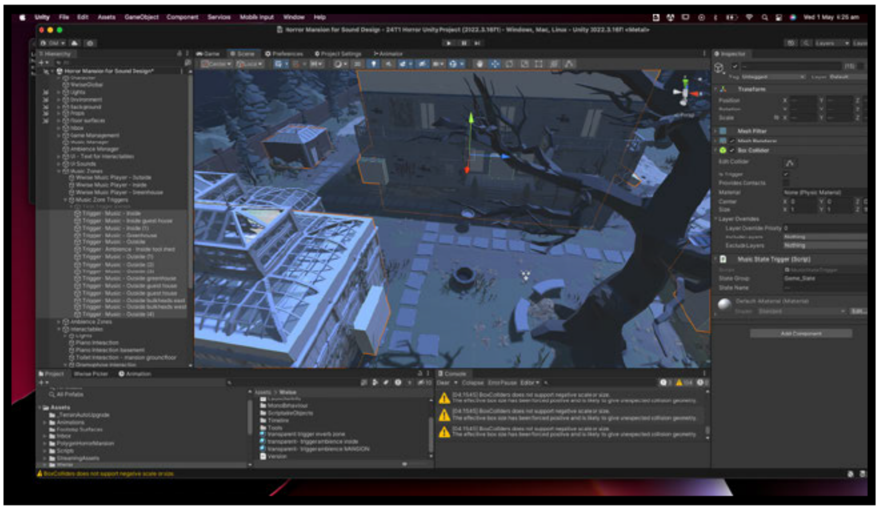
Now knowing what I would be using to implement my sounds within the game I started with the footsteps, the sounds required for the footsteps within the game were given to us besides the tile sounds, either you could drag and drop them if they were



already perfect sound loops of one singular foot step or the specific footstep manually. For the Tile sound, I used the Sennheiser MKH416 shotgun microphone paired with the Zoom F8 to record the sound of my leather shoes on the tiles within the house, I didn't put them on properly to create a the slight squeak of leather as I walked I as I imagined the character wearing leather boots and I came out excellent within the game, the gravel sounds within the game were sourced from sound snap, and applied to the Wwise Project in the same fashion that the clocks were applied to the game.

AMBIENCE ZONES AND TRIGGERS

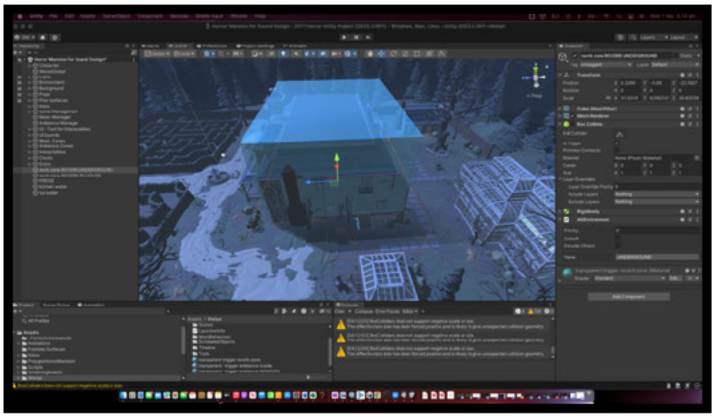
Ambience Zones are highly needed within a game to be able to create the environment around you and Highlight the switch within the environment as the player enters a new area, paired alongside reverb zones you can completely change the environment around the player



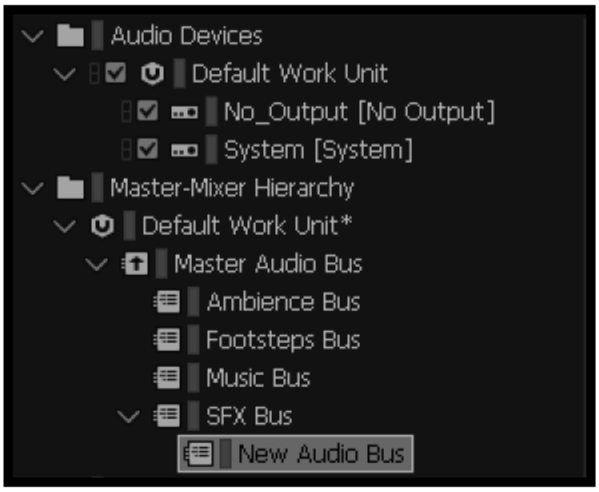
```
[04:15:45] BoxColliders does not support negative scale or size.
The effective box size has been forced positive and is likely to give unexpected collision geometry.
[04:15:45] BoxColliders does not support negative scale or size.
The effective box size has been forced positive and is likely to give unexpected collision geometry.
[04:15:45] BoxColliders does not support negative scale or size.
The effective box size has been forced positive and is likely to give unexpected collision geometry.
```

Ambience zones are created by making a cube object within Unity and making it a trigger meaning that walking through it will start a scripted action. Make sure it isn't a box collider so that the player can walk through, add component which will connect unity to Wwise, Go into Wwise and input your sound within the ambience folder, generate to sound bank then go back to unity and add the source onto the component and it will switch based on the what cube is set to what. I went in and changed all the colours of the cubes, white being outside and orange being inside, also adding them to the shed and guest house as they were missing.

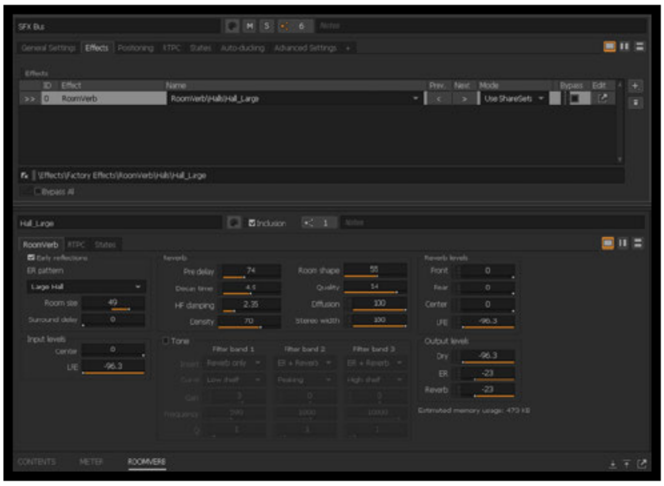
REVERB ZONES



Reverb Zones work the same as Ambience Zones up the part where you add the Wwise Component onto it, adding the Audio Kinetic Environment script allowed for select sounds within the Wwise Project to be routed to an Auxiliary Track with a reverb Plug-in causing the sound within that area to take effect to plug ins placed on that Aux folder. The cubes within unity acting as a trigger to turn on a bus with reverb in a music making DAW. Within Wwise I went into the default work unit and added a new bus folder and named it reverb and added a reverb bus for inside the house and underground in the basement



```
Audio Devices
  Default Work Unit
    No_Output [No Output]
    System [System]
Master-Mixer Hierarchy
  Default Work Unit*
    Master Audio Bus
      Ambience Bus
      Footsteps Bus
      Music Bus
      SFX Bus
        New Audio Bus
```



SFX Bus

ID	Effect	Name	Mode
0	RoomVerb	RoomVerb\Hall\Hall_Large	Use ShareSets

\Effects\Factory Effects\RoomVerb\Hall\Hall_Large

□ Bypass All

Hall_Large ☑ Inclusion

RoomVerb

☑ Early reflections
ER pattern: Large Hall
Room size: 49
Surround delay: 0

Input levels
Center: 0
LFE: -96.3

Reverb
Pre delay: 74
Decay time: 4.6
HF damping: 2.35
Density: 70
Room shape: 55
Quality: 14
Diffusion: 100
Stereo width: 100

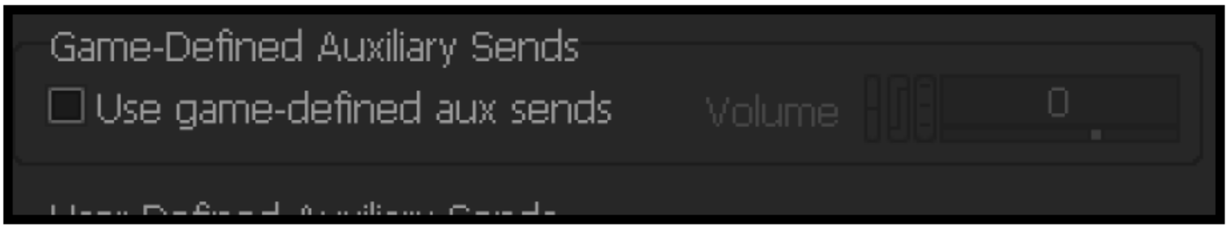
□ Tone

	Filter band 1	Filter band 2	Filter band 3
Insert	Reverb only	ER + Reverb	ER + Reverb
Curve	Low shelf	Peaking	High shelf
Gain	3	0	0
Frequency	200	1000	10000
Q	1	1	1

Reverb levels
Front: 0
Rear: 0
Center: 0
LFE: -96.3

Output levels
Dry: -96.3
ER: -23
Reverb: -23

Estimated memory usage: 473 KB



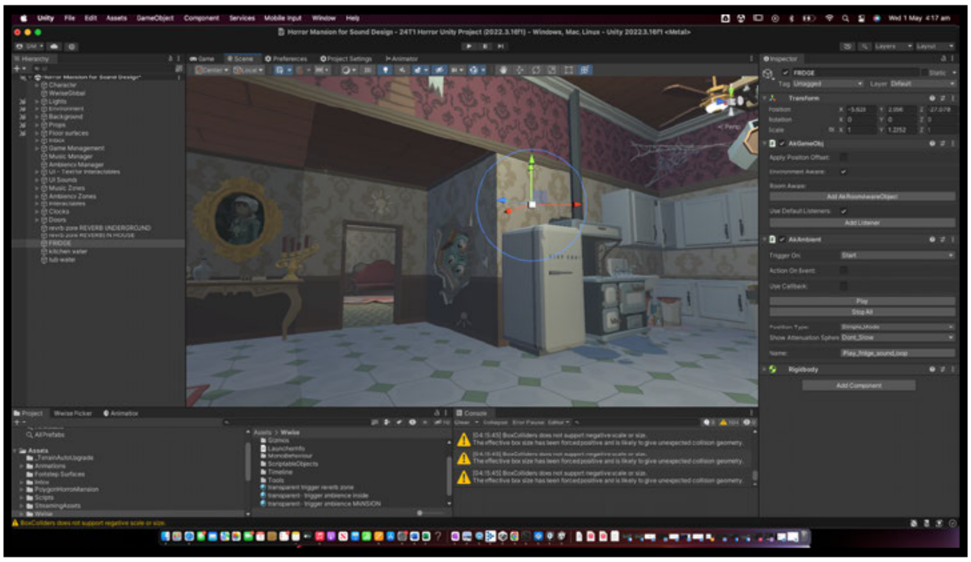
Game-Defined Auxiliary Sends

□ Use game-defined aux sends   Volume 0

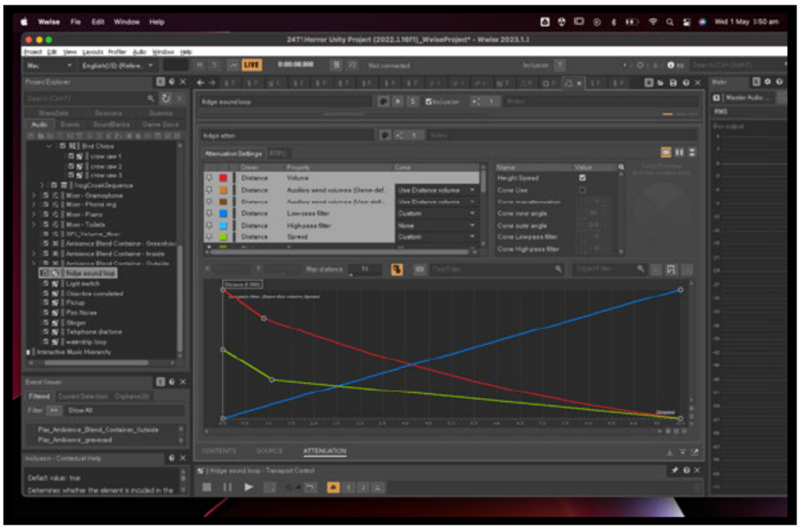
I added the reverb hall plug in to the bus and changed the effects to my desired settings, ticked the 'use game defined aux sends' to choose the sounds which are affected by the the reverb, in my base it was the piano, clocks, footsteps, jump sounds, gramophone and fridge and water dripping sound.

OTHER SOUNDS

I added two other sounds to the project, the 3rd time doing the project compared to the four I added within the first go round of my project, the sounds which made it were the fridge sounds and the sound of water dripping in the sink, missing out the leaves and the whispers I added coming outside of the walls. I liked the silence, it added an eerie feeling.



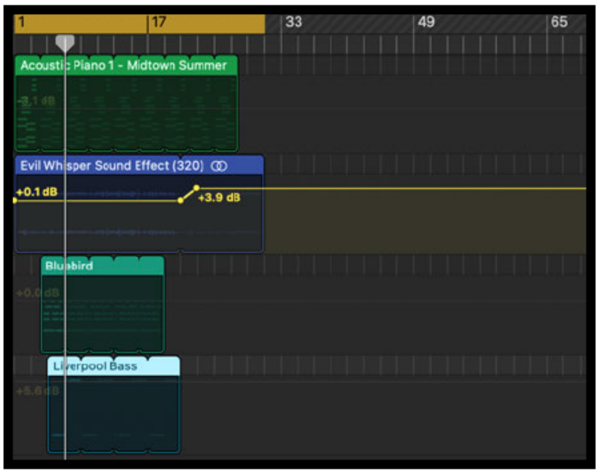
Horror Mansion for Sound Design - 2471 Horror Unity Project (2022.3.16f1) - Windows, Mac, Linux - Unity 2022.3.16f1 <Metal>
Hierarchy
Scene
Game
Inspector
FRIDGE
Transform
AkGameObj
AkAmbient
Rigidbody
Add Component
Project
Console



I added an empty object within Unity, and added the AkAmbient component which is similar to adding a sound source in unity, You then route the sound to unity.

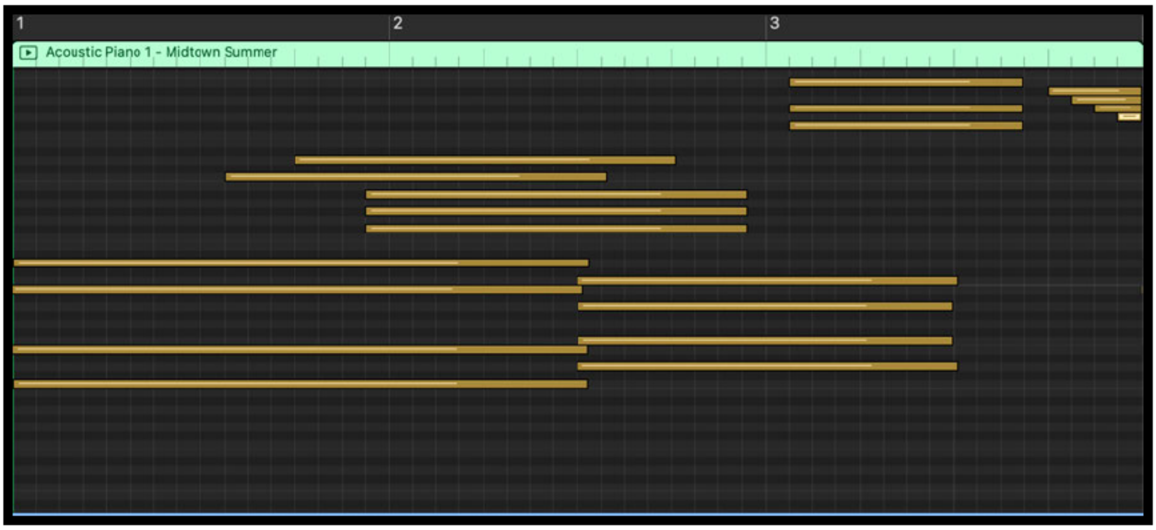
In Wwise you needed to make the sound an event for it to be able to heard within Unity because on it's own its just data within the Wwise file with nothing to do, Once I made it an event, I attenuated it so that the sound is accurate based on the players positioning (RTPC), changing the volume parameter, low pass filter and spread to create occlusion when the fridge or water is out of sight, and making the low end be the most notable thing at distance then the full depth of sound comes out when you are right next to the fridge, and spread is for the occlusion. For the water drip within the sink and the bathtub the max distance is 3 as opposed to the ten on the fridge.

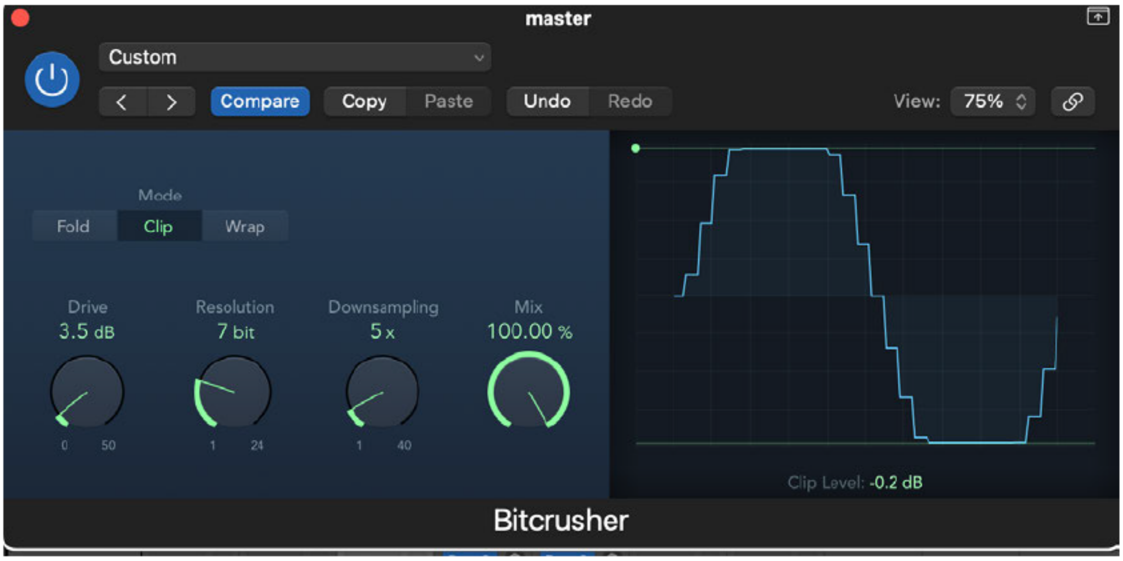
Music and Created Pieces


1
17
33
49
65
Acoustic Piano 1 - Midtown Summer
Evil Whisper Sound Effect (320)
+0.1 dB
+3.9 dB
Bluebird
Liverpool Bass

I made a soundtrack inside the house and inside the green room for the Using syntheses and Logic pro, I didn't end up using them within the final product but they were created, making use of sharp and flat notes to make it seem uncomfortable and scary synth sound. The images surrounding is the midi notes and automation on the music track 1, I added a bitcrusher to all the sounds to distort the sound

Using the same synths to create the open menu and close menu and also the projection complete sound





Mode: Clip

Drive 3.5 dB

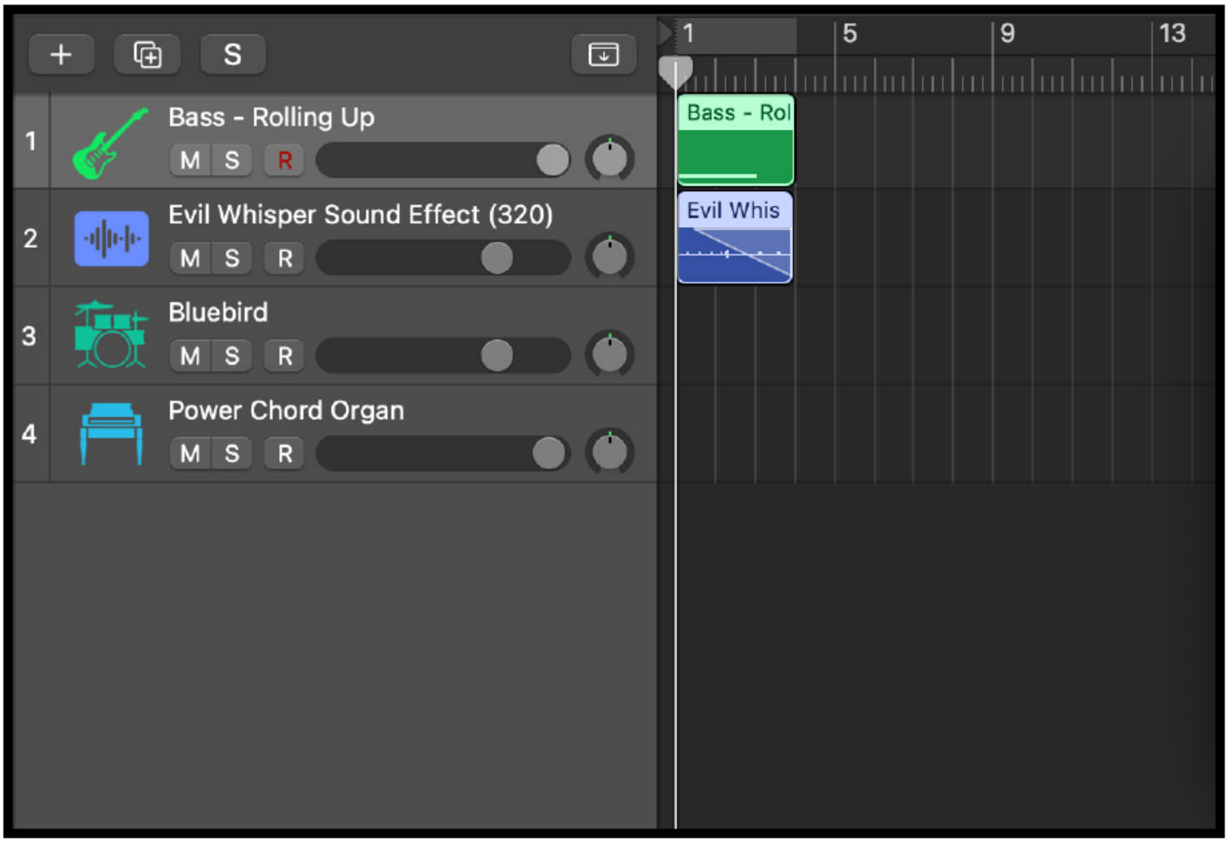
Resolution 7 bit

Downsampling 5 x

Mix 100.00 %

Clip Level: -0.2 dB

Bitcrusher Image.



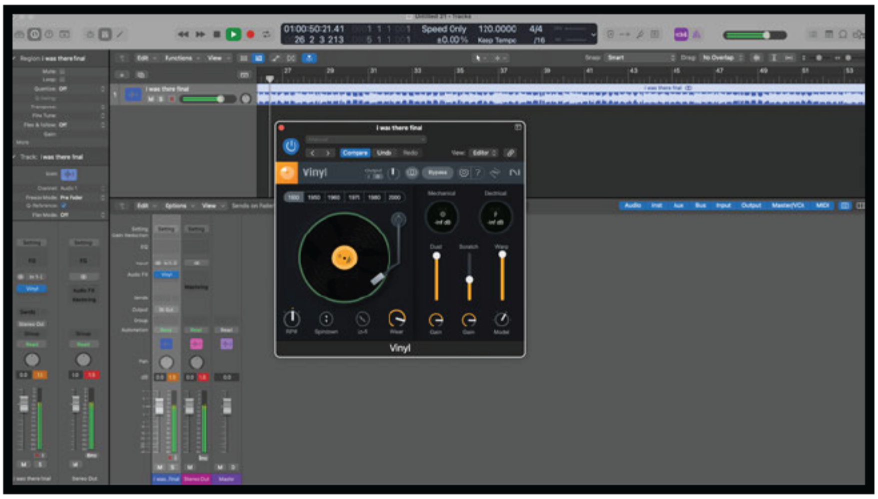
1 Bass - Rolling Up

2 Evil Whisper Sound Effect (320)

3 Bluebird

4 Power Chord Organ

Objection complete sound

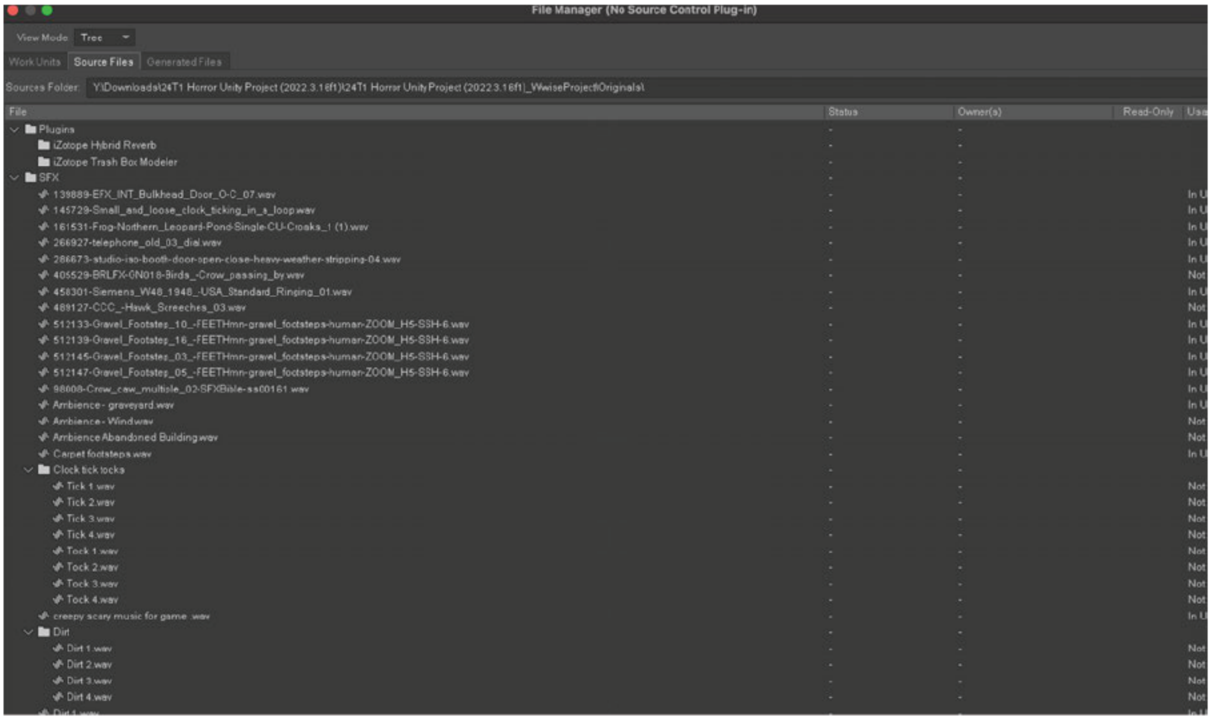


For the gramophone, As an Easter egg I added a song of my own created and used the Izotope vinyl plugin to make it sound like an old bad quality gramophone

The other recorded sounds within this project are the lightswitch and the toilet sounds which were simply recorded using my Phone.

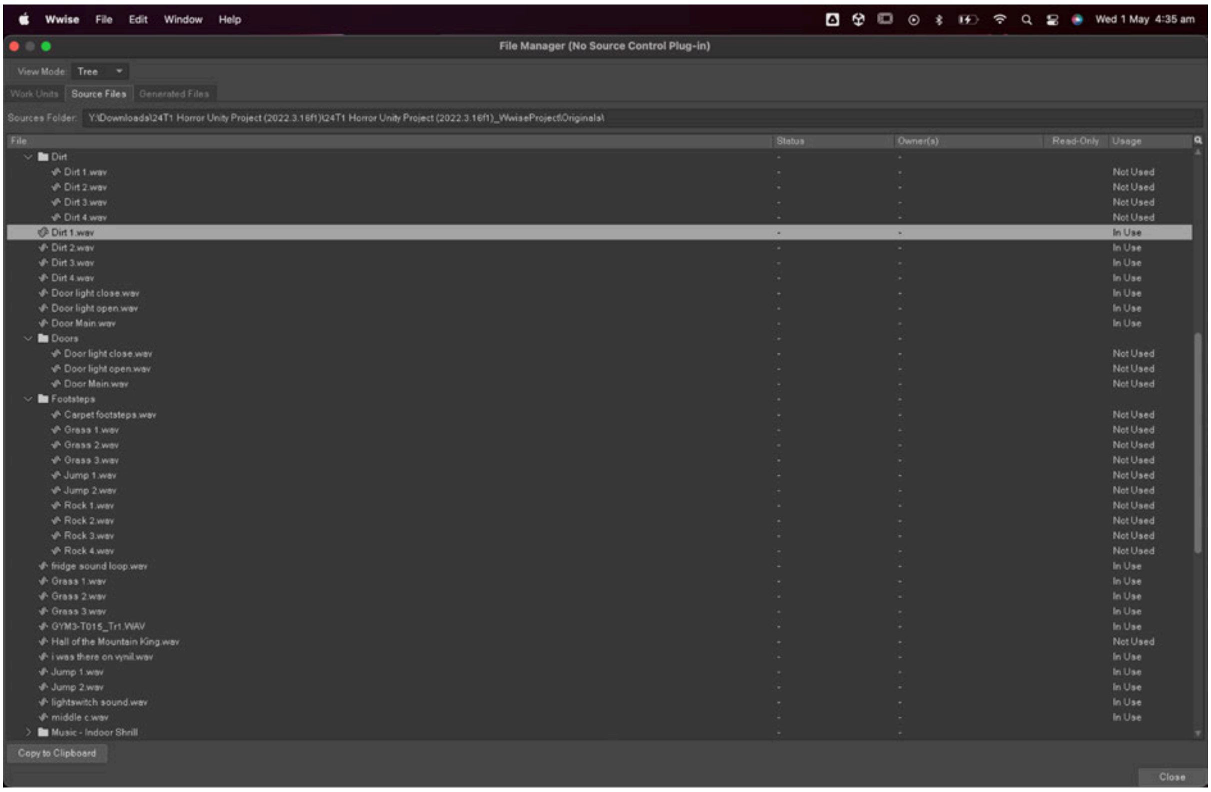
Every other sound within the game was sourced from soundsnap

ASSET LIST



Sources Folder: Y:\Downloads\24T1 Horror Unity Project (2022.3.16f1)\24T1 Horror Unity Project (2022.3.16f1)_WwiseProject\Originals\

File	Status	Owner(s)	Read-Only	Usa
Plugins	-	-		
iZotope Hybrid Reverb	-	-		
iZotope Trash Box Modeler	-	-		
SFX	-	-		
139889-EFX_INT_Bulkhead_Door_O-C_07.wav	-	-		In U
145729-Small_and_loose_clock_ticking_in_a_loop.wav	-	-		In U
161531-Frog-Northern_Leopard-Pond-Single-CU-Croaks_1 (1).wav	-	-		In U
266927-telephone_old_03_dial.wav	-	-		In U
286673-studio-iso-booth-door-open-close-heavy-weather-stripping-04.wav	-	-		In U
405529-BRLFX-0N018-Birds_-Crow_passing_by.wav	-	-		Not
458301-Siemens_W48_1948_-USA_Standard_Ringing_01.wav	-	-		In U
489127-CCC_-Hawk_Screeches_03.wav	-	-		Not
512133-Gravel_Footstep_10_-FEETHmn-gravel_footsteps-human-ZOOM_H5-SSH-6.wav	-	-		In U
512139-Gravel_Footstep_16_-FEETHmn-gravel_footsteps-human-ZOOM_H5-SSH-6.wav	-	-		In U
512145-Gravel_Footstep_03_-FEETHmn-gravel_footsteps-human-ZOOM_H5-SSH-6.wav	-	-		In U
512147-Gravel_Footstep_05_-FEETHmn-gravel_footsteps-human-ZOOM_H5-SSH-6.wav	-	-		In U
98008-Crow_caw_multiple_02-SFXBible-ss00161.wav	-	-		In U
Ambience - graveyard.wav	-	-		In U
Ambience - Wind.wav	-	-		Not
Ambience Abandoned Building.wav	-	-		Not
Carpet footsteps.wav	-	-		In U
Clock tick tocks	-	-		
Tick 1.wav	-	-		Not
Tick 2.wav	-	-		Not
Tick 3.wav	-	-		Not
Tick 4.wav	-	-		Not
Tock 1.wav	-	-		Not
Tock 2.wav	-	-		Not
Tock 3.wav	-	-		Not
Tock 4.wav	-	-		Not
creepy scary music for game .wav	-	-		In U
Dirt	-	-		
Dirt 1.wav	-	-		Not
Dirt 2.wav	-	-		Not
Dirt 3.wav	-	-		Not
Dirt 4.wav	-	-		Not
Dirt 1.wav	-	-		In U

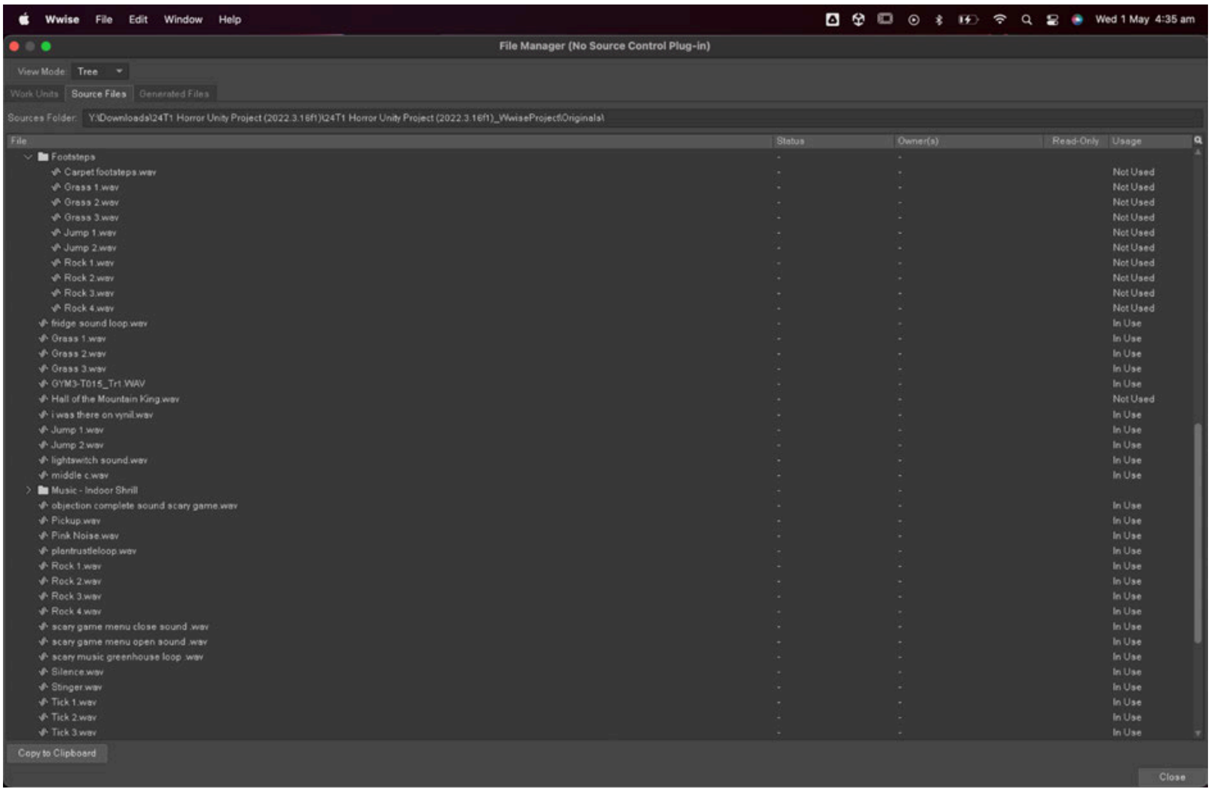


File Manager (No Source Control Plug-in)

View Mode: Tree

Sources Folder: Y:\Downloads\24T1 Horror Unity Project (2022.3.16f1)\24T1 Horror Unity Project (2022.3.16f1)_WwiseProject\Originals\

File	Status	Owner(s)	Read-Only	Usage
Dirt	-	-		
Dirt 1.wav	-	-		Not Used
Dirt 2.wav	-	-		Not Used
Dirt 3.wav	-	-		Not Used
Dirt 4.wav	-	-		Not Used
Dirt 1.wav	-	-		In Use
Dirt 2.wav	-	-		In Use
Dirt 3.wav	-	-		In Use
Dirt 4.wav	-	-		In Use
Door light close.wav	-	-		In Use
Door light open.wav	-	-		In Use
Door Main.wav	-	-		In Use
Doors	-	-		
Door light close.wav	-	-		Not Used
Door light open.wav	-	-		Not Used
Door Main.wav	-	-		Not Used
Footsteps	-	-		
Carpet footsteps.wav	-	-		Not Used
Grass 1.wav	-	-		Not Used
Grass 2.wav	-	-		Not Used
Grass 3.wav	-	-		Not Used
Jump 1.wav	-	-		Not Used
Jump 2.wav	-	-		Not Used
Rock 1.wav	-	-		Not Used
Rock 2.wav	-	-		Not Used
Rock 3.wav	-	-		Not Used
Rock 4.wav	-	-		Not Used
fridge sound loop.wav	-	-		In Use
Grass 1.wav	-	-		In Use
Grass 2.wav	-	-		In Use
Grass 3.wav	-	-		In Use
GYM3-T015_Tr1.WAV	-	-		In Use
Hall of the Mountain King.wav	-	-		Not Used
i was there on vynil.wav	-	-		In Use
Jump 1.wav	-	-		In Use
Jump 2.wav	-	-		In Use
lightswitch sound.wav	-	-		In Use
middle c.wav	-	-		In Use
Music - Indoor Shrill	-	-		



File Manager (No Source Control Plug-in)

View Mode: Tree

Sources Folder: Y:\Downloads\24T1 Horror Unity Project (2022.3.16f1)\24T1 Horror Unity Project (2022.3.16f1)_WwiseProject\Originals\

File	Status	Owner(s)	Read-Only	Usage
Footsteps	-	-		
Carpet footsteps.wav	-	-		Not Used
Grass 1.wav	-	-		Not Used
Grass 2.wav	-	-		Not Used
Grass 3.wav	-	-		Not Used
Jump 1.wav	-	-		Not Used
Jump 2.wav	-	-		Not Used
Rock 1.wav	-	-		Not Used
Rock 2.wav	-	-		Not Used
Rock 3.wav	-	-		Not Used
Rock 4.wav	-	-		Not Used
fridge sound loop.wav	-	-		In Use
Grass 1.wav	-	-		In Use
Grass 2.wav	-	-		In Use
Grass 3.wav	-	-		In Use
GYM3-T015_Tr1.WAV	-	-		In Use
Hall of the Mountain King.wav	-	-		Not Used
i was there on vynil.wav	-	-		In Use
Jump 1.wav	-	-		In Use
Jump 2.wav	-	-		In Use
lightswitch sound.wav	-	-		In Use
middle c.wav	-	-		In Use
Music - Indoor Shrill	-	-		
objection complete sound scary game.wav	-	-		In Use
Pickup.wav	-	-		In Use
Pink Noise.wav	-	-		In Use
plantrustleloop.wav	-	-		In Use
Rock 1.wav	-	-		In Use
Rock 2.wav	-	-		In Use
Rock 3.wav	-	-		In Use
Rock 4.wav	-	-		In Use
scary game menu close sound .wav	-	-		In Use
scary game menu open sound .wav	-	-		In Use
scary music greenhouse loop .wav	-	-		In Use
Silence.wav	-	-		In Use
Stinger.wav	-	-		In Use
Tick 1.wav	-	-		In Use
Tick 2.wav	-	-		In Use
Tick 3.wav	-	-		In Use

Mid-Project Reflection

- **Process:**

“Game development is a multifaceted and collaborative process in which game audio meets at the crossroad of creativity and technology. Technical and creative skills go hand in hand in shaping a game's sonic experience. Implementation, more than any
other aspect of game development, proves this.”(Zdanowicz, G., & Bambrick, S. 2020)” This quote from the before class reading stuck out to me because it shows that your proficiency in a certain technology alone or your creativity alone doesn’t work alone in shaping a games sonic experience but both together are co-dependent in being able to create an immersive sonic experience for games, implementation refers to the tools you use to put audio into a game, in our case its the game engine (Unity) and the Middleware (Wwise), so the challenges I have found within this project through Using Wwise and Unity was obviously using Wwise as it is a completely new software but once I watched some tutorials and watched some of the classes, navigating wise was actually easier than using the Unity in game engine sound tools alone. But also another challenge is that using middleware actually causes more strain on the computer when running unity, unity lags so much more with this project then the last even when it;s the only application running on my device. I am trying to go into campus to do the assignment as it seems to work better on the campus computers, the next steps within my project is the research portion and adding those extra sounds around the game level.

- **Person:**

Due to the project being individual, communication is based purely around feedback on your project. I've let classmates play around with my level and view my sounds and provide feedback on those sounds, telling me it’s too loud or doesn’t fit the vibe, even offering sound packs to use for the game. This is excellent for creating a good sonic experience for a game because you have to assume that you are not the only one listening to the games audio as you are creating it for others to hear not your self so feedback from peers or even people with untrained ears helps to create a piece that is objectively a good listening experience

- **Proficiency:**

Given the last project was based around unity, the navigation and use of unity feels quite easy for me, so given that the game engine that Wwise is playing middle man for is unity that makes one aspect of the project feel less stressful. Creativity plays a huge role in game audio design for implementation of music, character sounds and more but it is also important to stick within the realm of what you would like to hear as a listener so sometimes sticking to the basics is the best option over doing the most to impress other sound designers when in reality that’s not what the targeted audience would want to it’s important. understanding what your audience wants is probably the most important thing over all.

Project-Completion Reflection

- **Appraisal**:

Overall the project was somewhat successful in the fact that I was able to create audio for the game given to us but given the 2 restarts which I did within the project, I did find myself getting less motivated to go the extra mile for some of the aspects of the game audio implementation. Given that I am very happy with the research component of my project, I feel as if I have explained the middle and its use within game audio in a clear and concise manner.

- **Challenges:**

The Largest Challenge I faced during this project was actually my own computer, Using the M1 Mac Mini from 2020 , the 8 Gigabytes of unified memory wasn’t able to handle the the the communication from wise to unity, It was able to open at the start but as soon as I started generating sound banks from Wwise I, wasn't able to open Unity causing download a different version of unity to see if it work work, It did open but would take about an hour to open, so I restarted the game about twice one being a week before the due date, but I kept reaching the same issue,after waiting for an hour and a half for unity to open then immediately crash, I called quits and used a friends computer which was more powerful than mine, for some reason there was about 26,0000 files,so file transfer wasn’t gonna work but we got through it in the end., I learnt that I need a better computer and also to ask for help more option, I always feel as if I am able to work out these problems on my own, usually asking for help is my very last resort after I’ve exhausted all my other options

- **Future Goals:**

For my next project based around game sound design, I will be making use of a computer powerful enough to handle something as taxing as running a game engine but in terms of Wwise, I am currently working on using it for creating sound for a friend of mines game, So hopefully the continued use of Wwise and using it for all type of games will boost my ability to use it and increase my own value for future projects and job prospects.

Declaration

I have not used any AI tools or technologies to prepare this assessment.

Reference List

Amazon . (n.d.). *What is middleware? - middleware explained - AWS*. Amazon. https://aws.amazon.com/what-is/middleware/

Audiokinetic Inc. (2023a). *Documentation: Audiokinetic*. Introducing Wwise. https://www.audiokinetic.com/en/library/edge/?source=WwiseFundamentalApproach&id=introducing_wwise

Audiokinetic Inc. (2023b). *Documentation: Audiokinetic*. Understanding Real-Time Parameter Controls (RTPCs). https://www.audiokinetic.com/en/library/edge/?source=SDK&id=concept_rtpc.html

IBM. (2024, March 12). *What is middleware?* https://www.ibm.com/topics/middleware

Nogueira, T. (2019, July 19). *Audio middleware: Why would I want it in my game?*. Game Developer. https://www.gamedeveloper.com/audio/audio-middleware-why-would-i-want-it-in-my-game-

Williams , A. (2022, November 22). *What is middleware, and how does it work?*. Built In. https://builtin.com/software-engineering-perspectives/middleware

Zdanowicz, G., & Bambrick, S. (2020). *The game audio strategy guide: A practical course* (Chapter 8, pp.245). Taylor & Francis Ltd.