

RAVE SPIN

Interim Report

TU858

BSc in Computer Science - International

| **César Hannin**

C21415142

| **Supervised by Bryan Duggan**

School of Computer Science

Technological University, Dublin

November, 2025

Abstract

Aim:

Using an XR headset, give experienced DJ's the opportunity to use their equipment as they normally would, making and mixing music, all in XR, without the need for the physical equipment being present. Using 3D renditions of their existing equipment, such as the famous Pioneer DDJ-FLX4, DJ's will be able to use the 3D renditions the same way they would use their real-world physical counterpart. Anyone looking into becoming a DJ may also use the accompanying tutorials and lessons with this application to learn how to DJ and the accompanying music theory that goes into it also, without needing to spend a single euro, only requiring just passion and curiosity!

Methods:

This will primarily be done using the Godot engine as it has very strong XR tooling, audio processing, and networking capabilities, so it will act as our headset application.

Blender will be used for making the 3D models and textures.

For music integration the project will allow loading audio from disk or from other platforms, from Spotify to Soundcloud, each one offer their own API and solutions for music integration.

Results:

Progress on this project is going solid. Deploying the XR headset (*testing with the Meta Quest 3S*) is working, audio loading and playback is working, and some of the XR controls are working also.

The high priority requirements are all expected to be completed within the project deadline.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

A handwritten signature consisting of two parts. The first part, on the left, is a stylized "César". The second part, on the right, is a stylized "Hannin".

César Hannin

09-11-2025

Acknowledgements

Supervisor and original idea inspiration: Bryan Duggan

Contents

1 Introduction	1
1.1 Project Background.....	1
1.2 Preface and Terminology	2
Controller.....	2
MIDI Controller	3
FX Pad.....	3
CDJ	3
Setup / Session.....	4
1.3 Project Description	4
1.4 Project Aims and Objectives	5
1.5 Concept renders	6
1.5.1 Main Menu	6
1.5.2 First time running	7
1.5.3 DJ'ing away in a session.....	8
1.6 Project Scope	9
1.7 Unique selling point.....	10
2. Literature Review	11
2.1 Introduction	11
2.2 Alternative Existing Solutions.....	11
2.2.1 Tribe XR DJ School.....	11
Pros:.....	11
Cons:.....	11
2.2.2 Alive-In-VR.....	12
Pros:.....	12
Cons:.....	12
2.2.3 Vinyl Reality	13
Pros:.....	13
Cons:.....	13
2.2.4 DJAY	14
Pros:.....	14
Cons:.....	14

2.3	Technologies Researched.....	15
2.3.1	Game Engine	15
2.3.1.1	Unreal Engine.....	15
2.3.1.2	Unity.....	16
2.3.1.3	Flax Engine	16
2.3.1.4	Godot Engine	16
2.3.2	Blender.....	17
2.3.3	FL Studio 20	17
2.4	Other Research.....	18
2.4.1	Audio bus layout	18
2.4.2	Pioneer DDJ-FLX4.....	20
3.	System Analysis.....	21
3.1	System Overview.....	21
3.2	Requirements Gathering.....	21
3.3	Requirements Analysis	25
	Use case 1.....	25
	Use case 2.....	26
	Use case 3.....	28
	Use case 4.....	30
	Use case 5.....	32
4	System Design.....	34
4.1	Introduction	34
4.2	Software Methodology.....	34
4.3	Overview of System	36
4.4	Design System	36
	4.4.1 User interface / Sitemap	37
4.5	Audio Bus Layout	38
4.6	Controller Board (DDJ-FLX4)	39
5.	Testing and Evaluation	41
5.1	Introduction.....	41
5.2	Plan for Testing	41
5.3	Plan for Evaluation	42
6.	System Prototype	43

6.1 Introduction.....	43
6.2 Prototype Development.....	43
Hand interaction and input handling	43
Control behaviour and audio response	44
Bus layout implementation.....	44
6.3 Results	44
6.4 Evaluation	45
7. Issues and Future Work	46
7.1 Introduction.....	46
7.2 Issues and Risks	46
Adjusting to Godot	46
Local vs Global position	46
Is there enough demand on XR	47
Potential legal issues	47
7.3 Plans and Future Work	48
Audio track selection and loading.....	48
Multiplayer functionality.....	48
Menu and UI development.....	49
7.4 Project Plan with Chart.....	49
References	50
Appendix A: Prompts Used with ChatGPT	1
Chat #1	1
Starter prompt.....	1
Original Text:.....	1
Refined Text:.....	1
Original Text:.....	2
Refined Text:.....	2
Original Text:.....	3
Refined Text:.....	3
Original Text:.....	4
Refined Text:.....	4
Appendix B: Additional Code Samples.....	1
Arena.gd	1

Controller.gd.....	2
GD_Main_Menu.gd	3
Highlight_Mesh.gd.....	4
Highlight_Mesh_Slider.gd.....	5
Appendix C: Diagram of FLX4 layout:.....	1

1. Introduction

Rave-Spin is an XR-based DJ application designed to provide a portable, immersive alternative to traditional DJ equipment. Conventional DJ controllers and mixers are typically large, heavy, and expensive, making them difficult to transport and inaccessible for many beginners. High-end setups can cost thousands of euros and often require a solid grounding in music theory and DJ techniques before a user can meaningfully engage with them.

This project addresses these barriers by emulating the DJ environment in extended reality (**XR**). Rave-Spin aims to allow experienced DJs to emulate a full performance setup using only a headset and hand tracking, enabling them to practise or perform without carrying physical hardware. At the same time, the system is intended to support complete beginners by providing an accessible, low-cost entry point into DJ'ing that does not require investing in physical gear.

In addition, the application will include guided tutorials with gamified elements to make the learning process more engaging. These tutorials are intended to teach core DJ skills and concepts in an interactive way, helping users build confidence and competence over time. The overall goal is to combine the flexibility of XR with intuitive interaction design to create a portable DJ experience that is both practical for experienced users and approachable for newcomers.

1.1 Project Background

Effective DJing draws on a combination of musical, technical, and performance skills. Some individuals enter the space with a strong foundation in music theory, while others begin with little to no prior knowledge, which can significantly limit their ability to progress. In addition to understanding elements such as rhythm and harmony, aspiring DJs must also learn to operate a range of hardware and software tools, each with its own workflows and terminology, pros and cons, and potential stock availability issues.

While it's possible to teach yourself everything you may need using online resources, the whole learning environment can be quite fragmented. Platforms such as YouTube provide loads of tutorials, but they're sometimes tied to specific, commonly used controllers or software. Given the wide variety of DJ equipment available, many devices might receive limited coverage, leaving beginners at a loss if they can't find enough information or tutorials on their chosen equipment which can be confusing and discouraging. Studies have also shown that XR has been a great tool for enhanced learning so it could prove useful here too [1]!

For experienced DJs, the challenge is different but related. Professional-grade setups are often physically large, heavy, and expensive, making them impractical to transport regularly, especially for practice or casual use. The Pioneer DDJ-FLX10, the gold standard of DJ controllers, is 40cm long and 6.7kg, it's a challenge to move just that alone [2]. A system that can emulate familiar DJ workflows in an XR environment has the potential to reduce reliance on the physical hardware, enabling DJs to rehearse using only a headset, without the logistical overhead of carrying full equipment.

1.2 Preface and Terminology

Throughout this report and this entire project, several terms will be used to refer to different components and equipment. As these terms can mean different things to different readers depending on their background, here's a table to clarify the terminology that'll be used for this report.

Controller

A Controller refers to a **nearly** all-in-one piece of equipment. Typically combines two or more virtual “decks” with the jog wheels and a central mixer section.

Usually includes built-in FX pads and basic effects controls.

For most use cases, it functions as a near complete general solution for mixing and manipulating tracks.



Figure 1.2A –DDJ-FLX4 Controller



Figure 1.2B – DDJ-FLX10 Controller

MIDI Controller

Refers to a Piano-like device used to send MIDI data to software or hardware instruments.

It doesn't make its own sound, instead it controls parameters such as pitch, velocity and other aspects of whatever virtual instrument or audio source it is mapped to.



Figure 1.2C – Nektar Impact GX Mini

FX Pad

A pad-based performance controller consisting of multiple pads, each mapped to a specific sample, loop or effect.

Pressing a pad triggers the associated sound or effect, allowing rapid, rhythmic triggering during a performance.



Figure 1.2D – Pioneer DDJ-XP2

CDJ

Standalone media player designed to control only one track at a time.

Usually includes a large jog wheel, transport controls and a display, and is commonly used in professional and club-standard DJ setups.

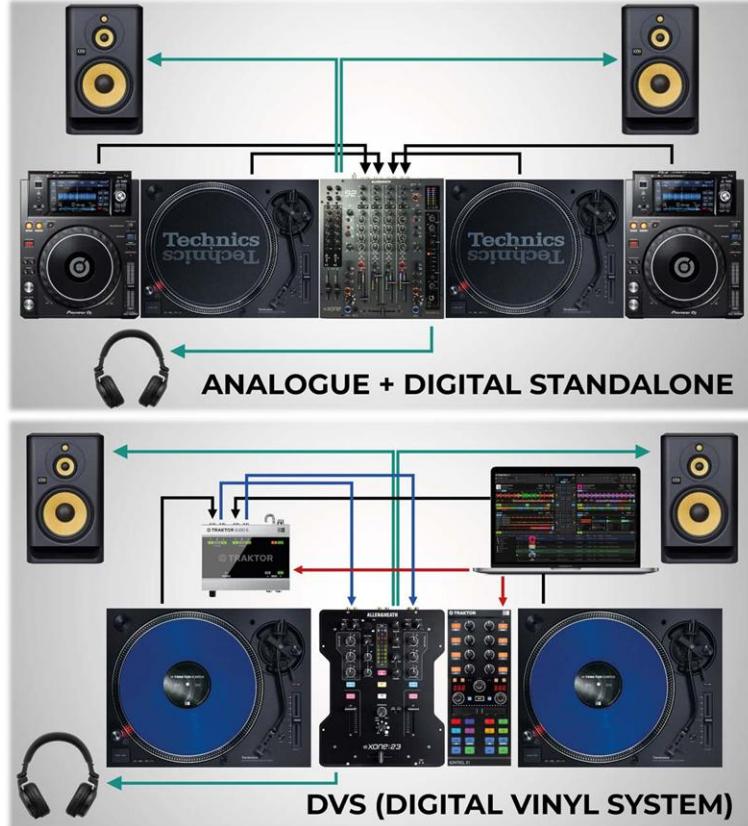


Figure 1.2E – Pioneer CDJ-3000

Setup / Session

The **Setup** refers to the complete collection of equipment used by a DJ. This varies from person-to-person, it may include one or more controllers, CDJs, a mixer, a laptop, an audio interface, or more. Loads of combinations exist depending on the user's preferences, budget, and style.

The **Session** refers to the period of time that this setup is being actively used by a DJ(s), during which time they could be mixing, performing, recording, etc...



1.3 Project Description

This project will focus on emulating the Pioneer DDJ-FLX4 at first because it's a really popular entry-level DJ controller that includes most features an aspiring DJ could want [3]. By targeting this controller first, the system can support many of the interactions and workflows that any DJ will likely encounter with most other controllers, making support for a new controller very quick.

Rave-Spin aims to provide an XR-based environment in which users can learn and practise DJ skills using a virtual model of the DDJ-FLX4 before committing to purchasing physical equipment. The controller itself typically costs in the region of €200–€330, which represents a significant upfront expense for many potential learners. By offering an interactive XR alternative, the project seeks to lower this financial barrier. Users can get familiar with the layout, experiment with mixing techniques and follow structured tutorials without needing to buy hardware in advance.

If, after using the system, a user decides that DJ'ing is not for them, they can simply stop using the application without having made any financial investment. Conversely, if they decide to progress further, their experience in XR should transfer directly to the physical controller, supporting a smoother and more confident transition into using real-world DJ equipment.

1.4 Project Aims and Objectives

The overall aim for this project is to have a fully working XR replication of the DDJ-FLX4, which can then be used by DJ's to mix their tracks as they would with the real physical hardware. As the project progresses, the project will include tutorials and gamification elements to encourage learning how to DJ well. Other features will hopefully become available as the project progresses. The features that will be immediately required to have a minimum viable product would be:

- Accurate 3D rendition of a Pioneer DDJ-FLX4 and an accompanying XR User Interface.
- The ability to load audio tracks from local disk.
- Reliable hit detection for all interactive controls (*buttons, sliders, knobs, jog wheels, etc*).
- Correct mapping of all controller interactions so that changes to controls on the DDJ-FLX4 model accurately affect the final audio output.
- Audio effects (*e.g. reverb, delay*) functioning in real time.
- The ability to record and save audio from DJ sessions for later playback.

The following extended features are not required for the minimum viable product but are planned for later iterations and prototypes:

- Interactive tutorials covering core DJ skills and equipment usage, initially focused on the DDJ-FLX4. These tutorials will incorporate gamification elements to make the learning process more engaging and fun.
- Local-network social connectivity that allow users to join each other's sessions, enabling collaborative DJ'ing, remote teaching, and real-time feedback.
- Support for additional equipment, like more controllers than just the DDJ-FLX4. Including CDJs, FX pads, and Midi controllers
- Support for microphone input within XR sessions.
- Integration with 3rd party music and streaming platforms (*e.g. Spotify, Apple Music, YouTube Music, SoundCloud*) to download tracks directly from a user's existing music library and playlists.
- Inclusion of further XR-based equipment such as sequencers and CDJs to support more complex and modular setups.
- Tutorials that extend beyond performance equipment to cover practical aspects of live sound, such as speaker placement, wiring configurations, gain staging and dB levels, with the goal of helping users prepare reliable audio setups for future gigs.

1.5 Concept renders

1.5.1 Main Menu

The main menu will serve as the starting point for the application. From this screen, users will be able to:

- Start an XR DJ session immediately.
- Host or join a shared DJ session over a local or online network.
- Access tutorials covering different aspects of DJ'ing and equipment usage.
- Adjust application settings (e.g. *audio*, *language*, *accessibility options*).



Figure 1.5A – Draft render of Main Menu

1.5.2 First time running

When Rave-Spin is launched for the first time ever on the users' headset, the user will be presented with the option to jump to the tutorials instead so that they can have a better understanding of the fundamentals. If the user knows their stuff they can just simply select "skip" and not do any tutorials.



Figure 1.5B – Draft render of Tutorial prompt

1.5.3 DJ'ing away in a session

During an active session, the user will be placed in an XR environment featuring a virtual DJ setup. In the current render mock-up, a Pioneer DDJ-FLX6 is shown, but the intended interaction is the same. Users will interact with the virtual hardware as they would with its physical counterpart, using hand tracking to operate the controls, like the jog wheels, sliders, knobs, pads, and buttons.

The visual layout and control behaviour will be designed to closely mirror real-world equipment, allowing experience gained in Rave-Spin to transfer directly to physical DJ gear.

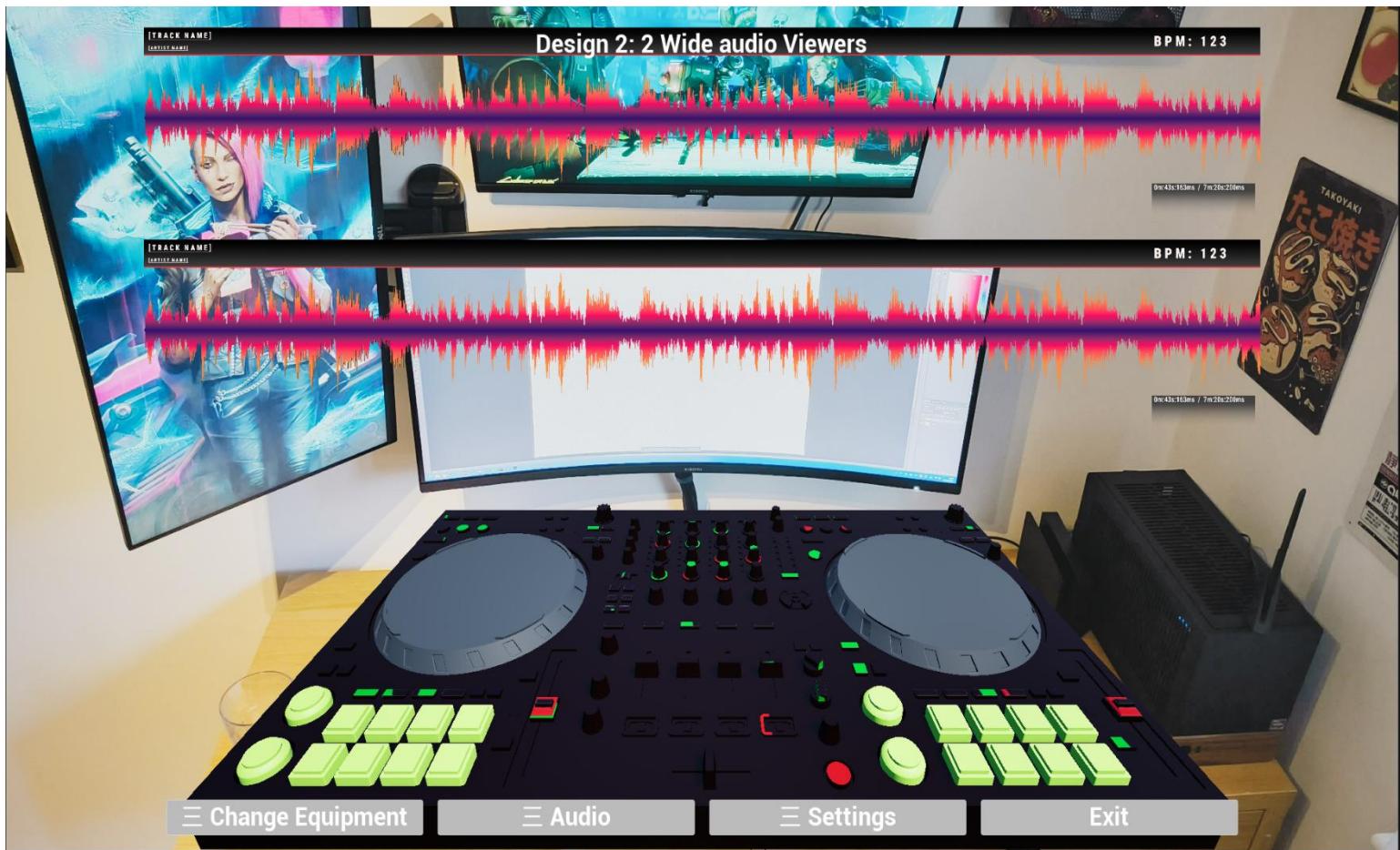


Figure 1.5C – Draft render of a Rave-Spin session

1.6 Project Scope

The scope of this project is centred on 2 main goals:

- To develop an on-the-go XR tool that emulates popular DJ equipment closely enough to be usable by experienced DJs with minimal adjustment. While it is not intended to completely replace dedicated hardware, the system should function as a practical, portable alternative or “companion tool”, particularly useful when travelling or when access to physical gear is limited.
- To provide an accessible learning environment for beginners who own an XR headset but don’t yet have DJ hardware. Rave-Spin will introduce users to what a DJ does, how DJ workflows operate and how to build core skills, allowing them to practise and improve without needing to purchase physical equipment in advance. If, after using the system, a user chooses to invest in real hardware, the knowledge gained should transfer over. If they decide it’s not for them then they’d have avoided any unnecessary expenses.

To keep the project achievable within a one-year development timeframe, the first few prototypes will focus mostly on the Pioneer DDJ-FLX4, a widely used and well-reviewed entry-level controller. Later prototypes will ideally support additional equipment, like a MIDI controller, an FX pad, another famous controller, MAYBE a CDJ if there’s enough time and demand. The intention is to implement good solid accurate examples rather than an exhaustive catalogue, the project is Open source so anyone who want’s to add their own equipment will hopefully be able to do so easily.

Features such as integrated light shows have been considered but are unlikely to be included within the current scope. A major challenge for the project will be balancing performance and accuracy. The application must maintain a consistently high frame rate in XR while still providing a precise and responsive emulation of real DJ tools. Actions done with the equipment will need to feel accurate, as any lag or latency could dramatically worsen the experience.

Some devices, including the DDJ-FLX4, still rely on user interfaces for tasks such as loading and managing tracks. To address this, the project will include a modular UI layer that can be reused across different virtual devices where similar functionality is required.

The project will also only be tested on the Meta Quest 3S. It’s very popular which is ideal and my results will be similar to a wide range of XR users, but because I’ll be using this and other headsets available at TU Dublin, the system is not intended to guarantee 100% compatibility or acceptable performance with all possible XR platforms.

Rave-Spin is not designed to render dedicated DJ hardware obsolete. It’s intended to complement existing tools by offering a flexible, and a practice environment all in XR.

1.7 Unique selling point

What will make Rave-Spin unique is the fact that there's no other app exactly like it on the market for this price point.

A tool of the DJ's

By the DJ's

For the DJ's

For people who don't know how to DJ, or are learning from scratch, It will ideally be a fantastic learning tool, hopefully teaching more people about how fun and easy it can be to DJ and make the scene more varied and expressive.

For experienced DJs, it will allow them to keep jamming away even if they're away from their setups. They can go on holidays with a small backpack and still mix and experiment with new tracks right away.

It will be completely free and open source, using FOSS (*Free open source software*) tools to accommodate this, with a great focus on code readability and documentation so that anyone else who wants to expand on it, maybe add more equipment from brands I've never heard of before, may do so easily.

2. Literature Review

2.1 Introduction

This section will cover how I will structure the project for development, as well as the research that went into it.

2.2 Alternative Existing Solutions

2.2.1 Tribe XR DJ School

Tribe XR DJ School lets users learn and perform DJ sets using realistic virtual hardware based off real equipment [4]. Users can use various controls, effects, mix tracks, and it also comes with interactive tutorials, multiplayer sessions, and live classes with mentors. Users can also import their own music and stream performances via Twitch or other platforms. Tribe XR runs on Meta Quest, SteamVR, and PC VR systems. It is nearly exactly what Rave Spin aims to be and accomplish.

Pros:

- Amazing catalogue of effects and gamification elements for learning.
- Highly rated and polished.[5]
- Dedicated community to provide support and lessons.[6]
- Cross platform, able to use SteamVR, Oculus, whichever XR headset the user wants. [6]
- Import your own tracks.

Cons:

- Only a paid subscription model. No option for a free plan, only a free 7-day trial.
- Many users have mentioned issues regarding latency.
- Equipment simulation is accurate, but users have reported issues with gestures not being recognised, making moving knobs and controls difficult sometimes.[5]
- Steam version is still in early access nearly 6 years later.



Figure 2.2.1 A,B,C – Images from Tribe XR DJ School

2.2.2 Alive-In-VR

Alive-In-VR is an XR interface for Ableton Live [7]. Rather than making audio itself, it sends commands to Ableton and lets you use the features that Ableton offers. In Alive-In-VR you can place blocks which represent clips, instruments, mixer controls, etc, around you in XR and interact with them. It doesn't try exactly replicate physical hardware but it still gives you good control over your music.

Pros:

- Is a VR interface for Ableton Live. This allows it to use a mature, full-featured industry standard DAW [7].
- Uses easy to grab blocks around you in 3D space instead of trying to perfectly replicate
- real life equipment, much easier to control.
- Great performance, very little reported latency[8].

Cons:

- Does NOT generate audio itself. It's just a controller / interface for Ableton Live which requires a valid paid licence.
- No educational aspects. You're expected to fully know the software and terminology.
- Doesn't simulate a virtual audience or venue. It's more of a VR interface tool than a performance simulator.
- No community behind it, which means no community support if any issues arise.
- Users have complained about polish. Appears tacky.

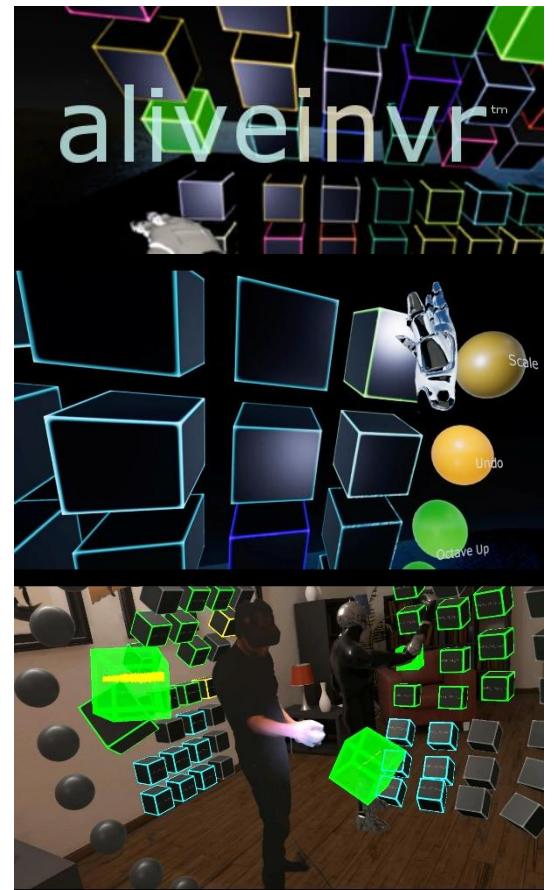


Figure 2.2.2 A, B, C – Images from Alive-In-VR from Steam

2.2.3 Vinyl Reality

Vinyl Reality is a XR app that simulates vinyl-style DJing. It has 2 turntables, and 2-channel mixers

with EQ, filters, gain, etc, and supports importing your own tracks. You can DJ in virtual environments with reactive crowds, lighting effects, and multiple camera angles. You can record your sessions or stream them and use custom or built environments. There's a short interactive tutorial to teach you the basic controls. It also has RekordBox integration (*a very famous app for interacting with DJ controllers*) [9] [10].

Pros:

- Great deal of realism and fidelity, able to work with turntables, mixer with EQ, filters, gain,
- etc
- Import your own tracks
- Virtual cameras can show you on a screen to see how you look while DJing, this also
- includes several environments (custom ones too), reactive crowds and light shows.
- Beginner friendly with tutorials.
- Highly rated and polished.
- Cross Platform support

Cons:

- Many users and the developer, have mentioned issues regarding latency. [10]
- It does not support elements such as sequencers or synthesizers with MIDI tracks.
- Users have reported that some aspects are too oversimplified. I could not find anything that immediately stuck out so will require further investigation.
- Some prior information is required. Tutorials do not teach you from the ground up.

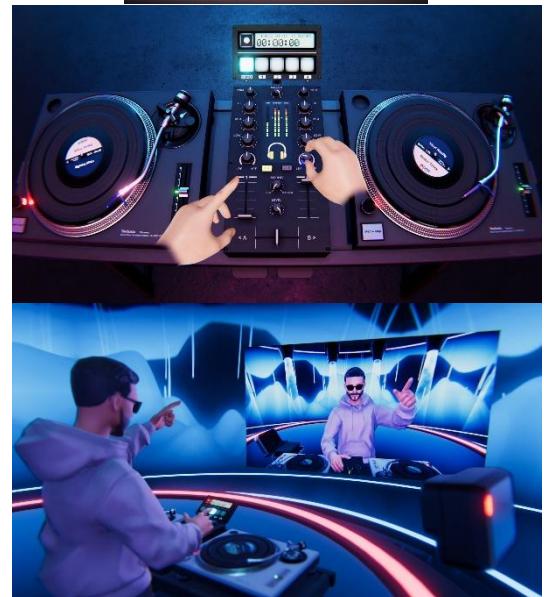


Figure 2.2.3 A, B, C – Images from Vinyl Reality from Steam

2.2.4 DJAY

Djay is a DJ app that lets users mix music and do live performance on VisionOS and Meta Quest. It

comes with virtual 2-turntable and mixer interface, integrates music streaming, supports MIDI/controller hardware. Depending on the platform the user is using and subscription plan, it includes various effects, auto-mix, looping, crossfader, etc. [11] [12] [13]

Pros:

- Cross Platform support
- Import your own tracks (Spotify, Apple Music, and Soundcloud integration included)
- In-depth mixing and channel separation, such as vocal separation.
- Very user-friendly
- Very feature rich
- Highly rated and polished with what appears to be user-friendly UI / UX
- Free tier has many of the essentials and is sufficient
- Of all the software listed here, DJAY ranks the highest in my opinion.

Cons:

- Pro tier is paid which locks off many features such as gesture controls, different views, custom playlists, and more. [12]
- Some features are disabled for certain streaming sources (Apple Music, Spotify) due to licensing.
- Some users have reported that they feel that development of new AI features has taken priority over core usability features, like addressing crashing on slightly older hardware.
- Does not Emulate real-world Controllers, it only interacts with them via USB.

This shows that the idea can be done. These preexisting systems allow users to mess with virtual turntables, mixers, sequencers, and more, all in XR environments, which is very much what Rave Spin aims to accomplish. The project is based off this finding. By combining interactive learning of being a DJ, and having simulated or livestreamed crowds, the project aims to reduce stage anxiety while also teaching how to become a real-world DJ. The inclusion of livestreaming, as seen in DJAY, via Twitch, Discord, and/or YouTube integrations would hopefully help with community engagement and help with the aim of reducing stage-fright.



Figure 2.2.4 A, B, C – Images from DJAY official website

2.3 Technologies Researched

For this project there were several technologies that will be required for the different aspects.

2.3.1 Game Engine

For the project scope, it's unrealistic to attempt to try creating an engine to handle rendering and heavy-duty audio processing. I have made a game engine before in the past and from first-hand experience know about the high difficulty of using DirectX 11 to get simple things rendering. As I'm targeting low-powered android hardware, learning OpenGL or Vulkan, and how to do XR rendering on top of that would be unfeasible for this timeframe. Although a custom solution would theoretically offer the best performance, as I would have complete control of thread groups and resource management, and I have previous experience with FMOD so audio manipulation wouldn't be an issue, to complete this in the timeframe a preexisting engine would be required.

2.3.1.1 Unreal Engine

I have made a couple of games before in Unreal Engine and have a great deal of experience with it. I am extremely comfortable with C++ and blueprints scripting. I am also well aware about the many faults with this engine. Unreal engine, when targeting hardware like the Meta Quest 3S requires several changes, such as changing the RHI to Vulkan so that it may be used by Android. The rendering method also needs to be changed to forward rendering instead of the default deferred rendering so that it may run faster while using fewer resources. From a rendering perspective, unreal engine is possibly the best contender. Unreal Engine also boasts its interoperability with VR. It's an industry standard engine and has great support for android.



I did not choose unreal engine for only 2 reasons:

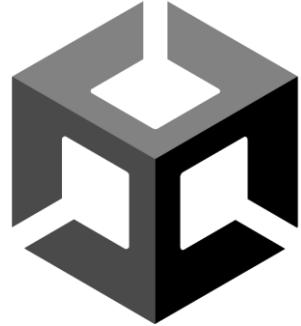
1. It is **not open source**. It is my plan to make this application fully free, and ideally open source so that others may add their own equipment and customise the program however they may wish. Unreal Engine from my own experience does not have an easy mechanism to allow for advanced after-market user modifications, not without the release of a "mod kit" which I would have to develop and this would take an unknown toll on the project timeframe. Licensing issues also exist for developing a "mod kit" and is uncommon for Unreal Engine titles.
2. Previous bad experiences with **audio issues**. Unreal Engine default audio engine has fantastic audio capabilities, and it is highly sophisticated, especially for games. For this project, low latency, and multiple sounds being in memory and being able to be played at once is of upmost importance, and from previous projects I have run into issues with Unreal

such as audio cut-offs, pitch multipliers only ranging from 0.5 to 2.0, drops in performance when multiple FX were active, and more issues. Although these issues may have been fixed with newer engine versions, or could have been fixed by editing the engine source code, or implementing my own audio subsystem (perhaps with FMOD), I believed this to be the major issue with this approach. If I had more time, I would have given unreal engine a shot, but for the timeframe provided, something that would have provided more certainty would be preferred.

2.3.1.2 Unity

Unity is a game engine which also is highly regarded in the industry. It has fantastic XR capabilities and can render easily on very weak hardware.

It however, is **not open source**, and the source code, unlike Unreal Engine and Godot, is completely hidden. I have never used Unity before in the past so a learning curve would undoubtedly be present. I also found users complaining about audio latency and performance issues, particularly when a great number of FXs are active.



Like Unreal Engine, it is not open source, so it suffers from the same issues that stem from that.

I did not choose Unity for these reasons.

2.3.1.3 Flax Engine

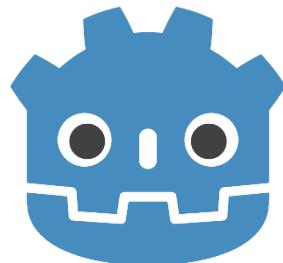
Flax Engine is another game engine that also supports Android. I have recently started using this engine and although it shows great promise, it still doesn't support VR. The online Trello Board showing the engines progress specifically mentions VR/XR support as 20% completed. It has stayed that way since 2016.



2.3.1.4 Godot Engine

Godot engine version 4.5 is the engine I decided to go for as it has very robust XR functionality, such as gesture controls, spatial awareness, and multiple libraries to extend it even more.

The default audio engine is also extremely modular and supports custom bus layout but the main thing which sold me on it was that projects created by Bryan Duggan, such as his *Godot audio sequencer*, was able to easily handle multiple concurrent sounds, all playing on tempo, at once with almost no performance loss. This proved that the default audio capabilities alone were capable enough to handle what this project aims to do.



Godot is also open source and features a rich ecosystem of plugins and tools which will prove to be instrumental in the projects progress, such as the XR toolkit plugin. The community behind Godot is extremely large, especially recently as many Unity developers have moved over to Godot. As progress continues, I'll see if it's possible to use a plugin for Twitch / YouTube / Discord integration. If not, then I will compile my own plugin to use with Godot. I have already done this functionality in C++ in previous projects so it may be possible to copy the code over.

One issue that exists is that I have never used Godot before, and I have already found complications with its scripting language "GD-Script". Although using C# is possible, it is recommended not to use C# as it has slightly lower performance than GD-Script. If GD-Script every becomes too much of a hindrance I will explore focusing on C# scripting instead.

| 2.3.2 Blender

To make the various 3D models and animations, I will be using Blender. Like Godot, it's free and open source and has an incredible ecosystem of plugins and tools and the community is equally large and enthusiastic. Another fantastic feature of Blender which is of particular importance to this project is its' ability to export to GLTF file format. This means that whatever names I give the various meshes and components remain persistent between Godot and Blender, and similar meshes can be "linked" instead of unique.



For example, this means that when we import a model into Godot that has 50 cylinders, rather than Godot storing 50 individual cylinders in memory so that it can render each one individually, it can instead keep just one cylinder in memory and render it 50 times. It's small savings like these that will help with the applications performance on weaker hardware.

| 2.3.3 FL Studio 20

For making sample tracks that the user may use during the tutorials, I will be using FL Studio 20. It is the Digital Audio Workstation (DAW) that I have the most confidence and experience in. It's one of the industry standards due to its reliability and features set. Any other software is fine as it is only .wav, .mp3, or .ogg audio files that we want.



2.4 Other Research

In this section I'll cover some of the extra research I had done regarding other components of the system

2.4.1 Audio bus layout

To make an accurate emulation of the Pioneer DDJ-FLX4, or any other DJ equipment, it's important to know how audio is routed through it. The main flow of audio will usually follow this path [14] [15] [16]:

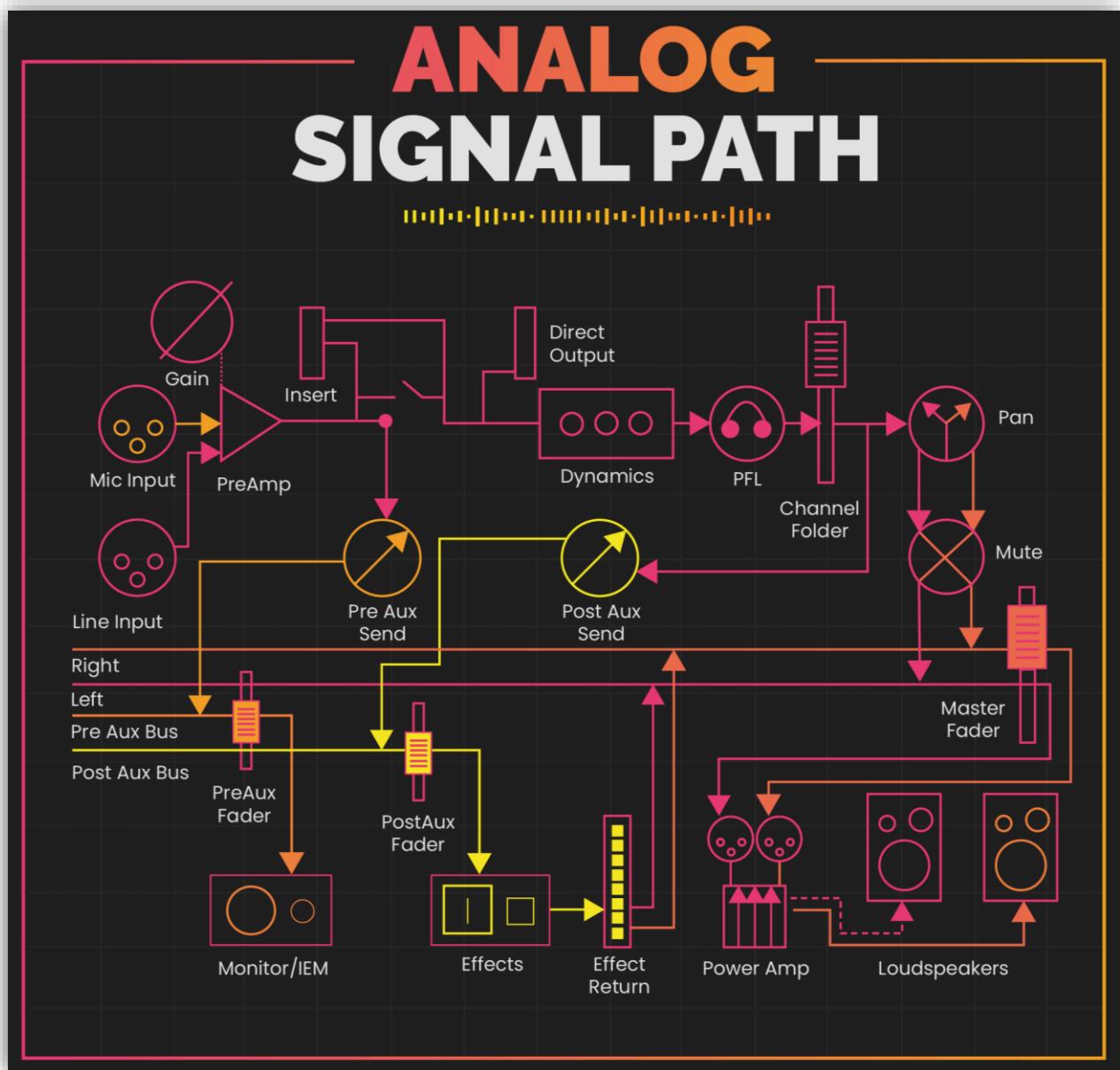


Figure 2.4.1 A – Diagram of normal Audio signal path

From this we can see that there's 2 important input sections:

- Channel X Input (Imagine this is our "Line input")
- Channel X Effects

Whatever audio track we put into Channel X (*The pioneer FLX4 has 2 decks, so 2 channels*) will be mixed in with the master output [17]. This would mean we'd an audio bus for our Channel 1 input (to visualise the audio waveform before any effects are applied), Channel 1 FX, do the same 2 for Channel 2, and then a master output. This should mean we'd probably need about 5-6 audio busses. In total. Here's a sample diagram of what I'm thinking:

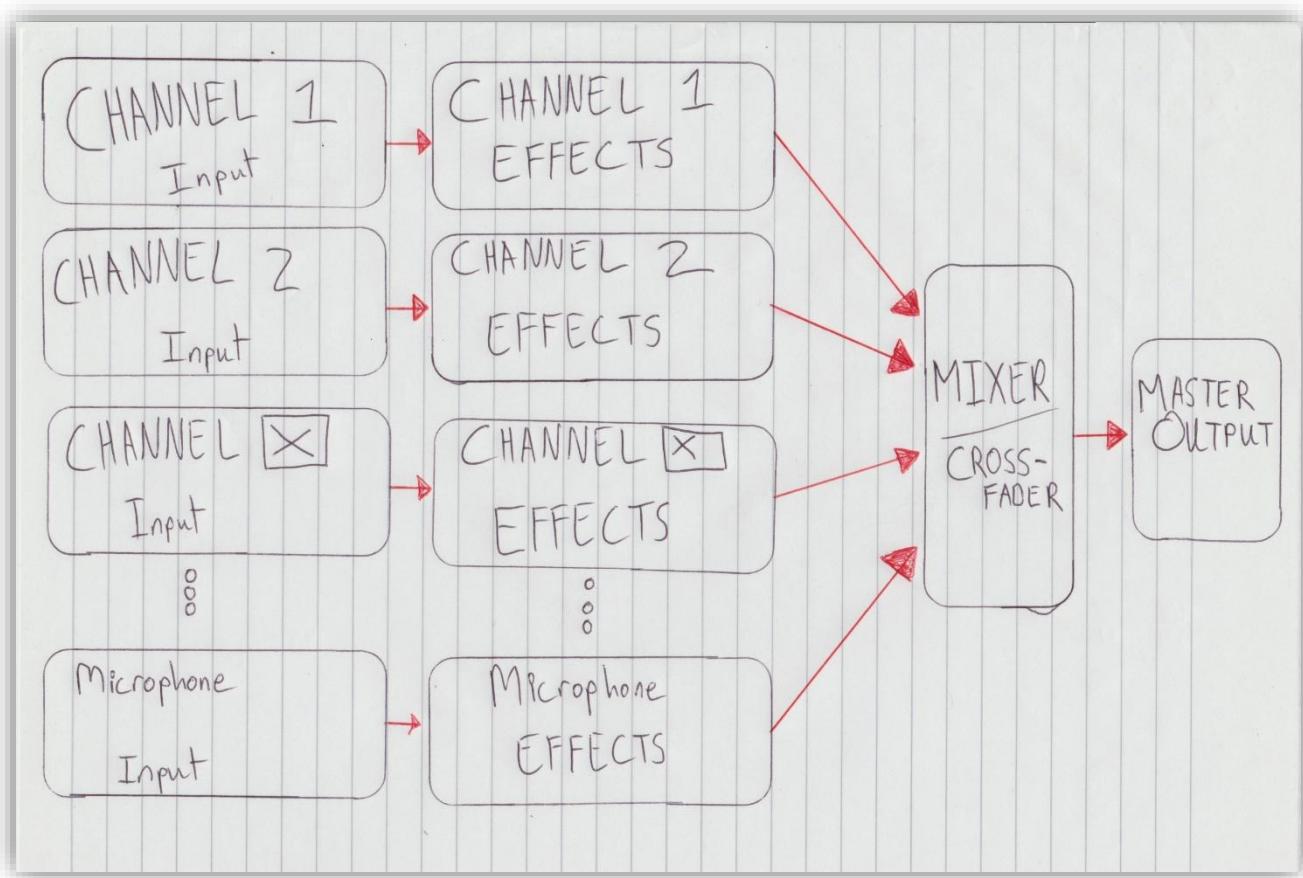


Figure 2.4.1 B –Sketch diagram of Audio bus layout for Rave-Spin

All inputs will be taken in their original form first. This will let us analyse their waveforms before any effects are applied. We can check easily if any given audio wave form is normal:

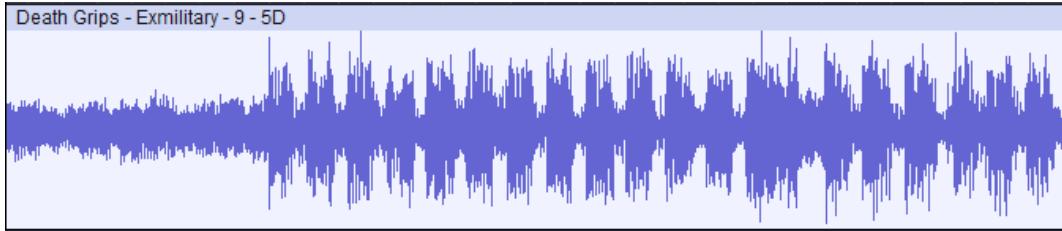


Figure 2.4.1 C – Normal Audio waveform

This is a normal audio waveform, should be ok to apply effects to!



Figure 2.4.1 D – Undersaturated Audio waveform

This audio is likely going to be too quiet. We need to apply more “Gain” (basically its’ volume) before we apply effects otherwise, we might not even hear them.

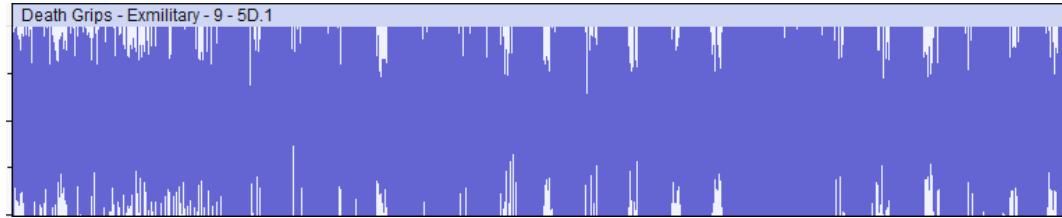


Figure 2.4.1 E – Overblown Audio waveform

Slim chance that this audio will be so loud it’ll destroy your speakers. Need to reduce its Gain otherwise everything, effects included, will be a blown-out / muddled mess.

2.4.2 Pioneer DDJ-FLX4

The FLX4 is one of the more common and popular DJ controller board out there [18] [19]. It does almost everything a DJ could want and it’s fantastic for beginners to. It would be like the Ford Fiesta of the DJ world. It does 90% of everything you need, and if you know how to use this board, you’ll likely have no issue with most other Controllers.

Thankfully there’s loads of resources out there about how this board works exactly. Please check Appendix C for the diagram of how the controller is laid out.

3. System Analysis

3.1 System Overview

So from the users perspective, they'll launch the application on their XR headset (*at the time of writing, will assumed to be the Meta Quest 3S as that is the headset we have for testing*).

The user will then be able to invite others whom also have the app to join their sessions or join the sessions of others. This will be useful for teachers/students who might want to get lessons for how to DJ, especially if teacher and student are long-distance. The user will also be able to teach themselves how to DJ by using the pre-included tutorials.

These tutorials will have gamification elements which can make it fun for the user to learn and hopefully will encourage them to continue and try to keep learning. If users would like to skip the tutorials they can just launch straight into their setup. If they have saved their previous session, they can launch right where they left off, it will have their preferred Controller board (*by default will be the DDJ-FLX4*) being emulated right in front of them.

Other equipment such as sequencers and Midi controllers will also be emulated and shown right in front of the user and tracks preloaded ready to go, exactly how they left it. Or if they want they could start fresh and maybe try use a different controller board, maybe a FLX10[19] if they want to try give that a go.

Any changes to the initial setup will be recorded and saved so that when the user exits the app, and returns, they can pick up right where they left off. They may also go into settings and change certain aspects, such as font sizes of the different texts, colourblind adjustments, and more.

3.2 Requirements Gathering

The main stakeholders for his application will be

- Hobbyists who want to learn how to DJ, possibly from scratch, and those who already have a degree of preexisting knowledge.
- Experienced DJs who want something that can emulate their setups while on-the-go with their XR headset.

To effectively gather the requirements of what would be needed I'll make a form, ideally using Microsoft forms, asking people first for their experience level. This will hopefully split up results into 3 distinct groups:

1. The experienced DJs who just want a portable tool to emulate their current setups;

2. The intermediates who know a bit but want to learn more about how to DJ or use higher end equipment;
3. The beginners, people who have never DJ 'ed before and would like to learn how without spending money on expensive equipment.

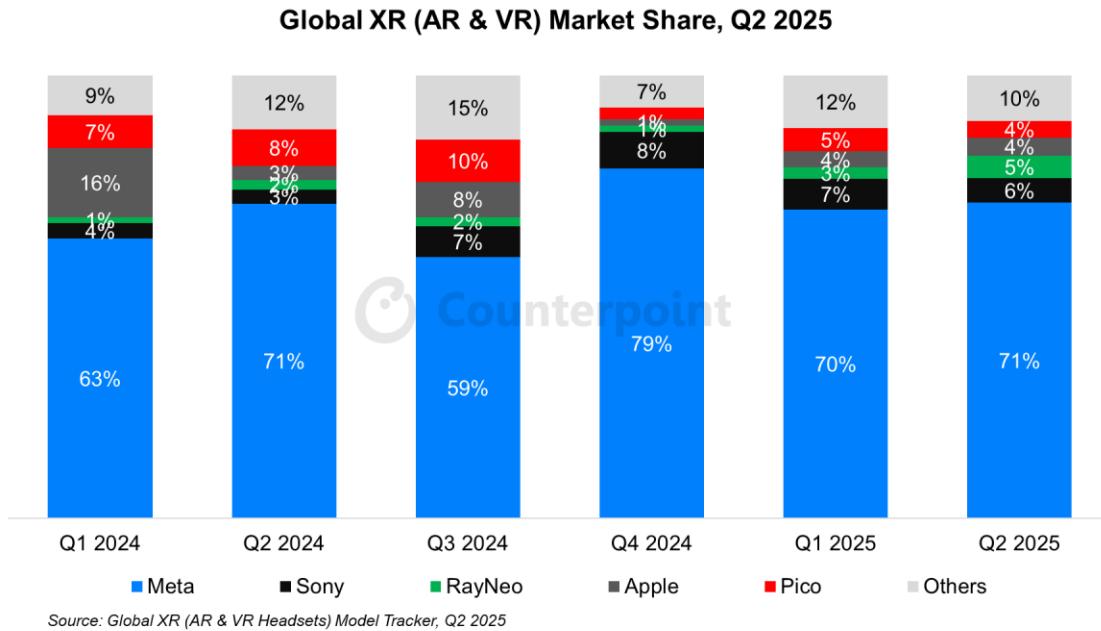


Figure 3.2 A – XR Headset Marketshare for 2024 & 2025

We know that the Meta Quest 3 headsets are the most popular XR headsets, so I'll be targeting them first and foremost. To make sure the tool is accurate, we also need minimal latency.

I am also one of the stakeholders as although I may not have a great deal of experience with DJ'ing on higher end equipment, I have a very basic but solid foundation of knowledge and would like this tool even for just personal use.

There are several functional (what the system should do) and non-functional (HOW the system does them) requirements for this system.

To help with the evaluation and to see the greatest areas of focus, I have this questionnaire which will be sent out via MS Forms:

Question 1. How would you describe your DJ level:

- None
- Beginner (Know the basics such as reverb, EQ, etc...)
- Intermediate (I know the basics and can make decent mixes and tracks)
- Advanced / professional (I know my shit)

Question 2. Do you own an XR headset:

- Yes
- No

Question 3. If you were to use an app built for DJs, everyone from experts to absolute beginners, what would be most important (1: really important, 5: Not important at all)

- More controllers (Like the DDJ-FLX10, FLX6, maybe controllers from different brands)
- More different equipment (like Midi controllers, sequencers, etc...)
- The ability to have others join your session in XR and mix with you (if you allow them, they can also be set to view-only)
- Interactive tutorials which teach you everything you'd need to know about how to DJ. (could have scores too to make them more fun)
- Being able to load in Audio from Spotify / Apple Music / Soundcloud
- Being able to use this on Windows/Linux, not just XR
- Custom environments. Have your sessions with whichever backdrop you want.
- Other

Question 4. If you wanted to learn how to DJ how would you prefer to learn:

- Interact hands-on with a near 1-to-1 replica in XR of the most common DJ equipment
- Online tutorials / reviews (Youtube, ChatGPT, etc...)
- Lessons (in-person)
- Lessons (Online)
- Other

Question 5. If you are a DJ, and were thinking about getting new equipment, what would be most important to you:

- Interact hands-on with a near 1-to-1 replica of it in XR of that specific equipment first.
- Interact hands-on with the real thing (in a shop/demo room).
- See online reviews
- Other

Question 6. [ONLY if experience level > intermediate] (Assuming you have access to an XR headset) How likely would it be that you would use an app on your XR headset that accurately emulates your DJ setup and equipment? (you can specify the equipment you want; it doesn't need to match exactly if you want)

- I would never use it
- I would use it only if I wanted to try out new equipment I'm planning on buying.
- I would use it if I'm on-the-go and can't take my whole setup with me.
- I would use it just for fun / causal DJing.
- I would use it more than my physical setup.
- Other

Question 7. *[ONLY if experience level == beginner]* (Assuming you have access to an XR headset) How likely would it be that you would use an XR DJ app that has interactive tutorials?

- I would never use it
- I would skip the tutorials, just mix around with the equipment and learn that way.
- I would only play the tutorials and then skip everything else.
- I would use it yes.
- Other

Question 8. *[ONLY if headset == no]* Would you consider purchasing an XR headset just to use this app

- No
- Maybe, / borrow a friends' and decide then
- I'd just see if my phone has XR mode or something
- No, I'd see if I can get it working on Windows / Linux / MacOS
- Yes

Question 9. *[ONLY if experience level >= intermediate]* What would you consider to be the most important things that aspiring DJs should learn or be aware of?

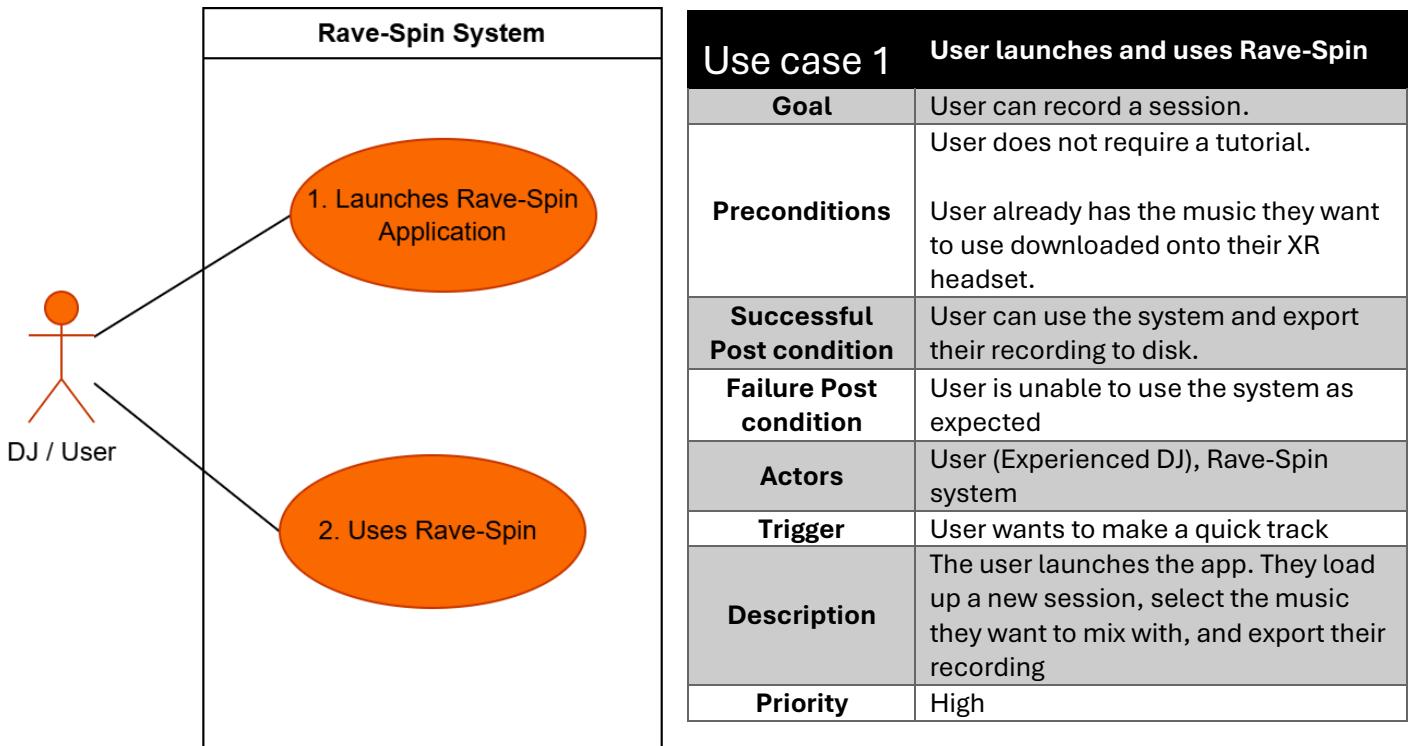
- [text field]

Question 10. If are an experienced DJ, or are learning/practising, what are some issues that you faced or are facing now, like equipment availability, Dodgy software for certain components, legal issues (if legal please let me know which country), etc...

- [text field]

3.3 Requirements Analysis

In this section we'll cover the different use case diagrams that our system will need. Priorities will be of Low, Medium, High.



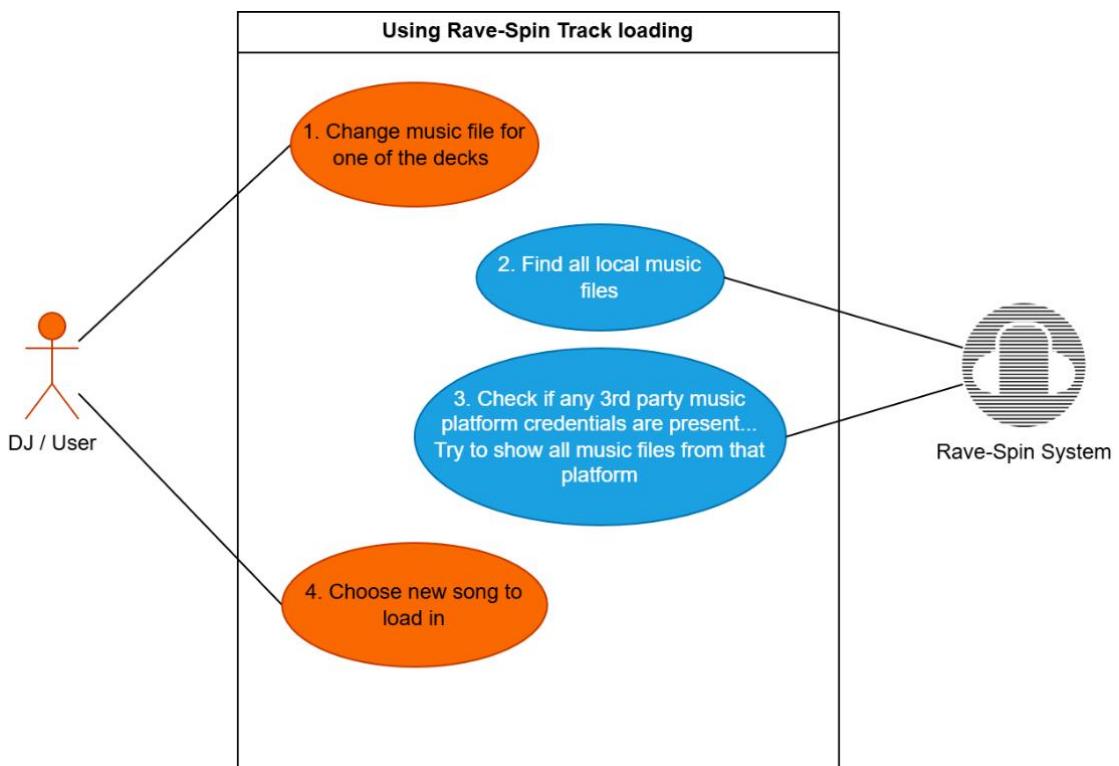
Main Flow		
Step	Action	Alternate
1.1	System remembers previous settings and setup information	AF 1.1
1.2	User picks the Music file they want to use as the track input audio	
1.3	User starts recording their session	AF 1.3
1.4	User wants to change the Music file for a particular track	AF 1.4
1.5	User ends recording and their session	End of Main Flow

Alternate (Exceptions and Error) flows		
Step	Action	Proceed to alternate step
Alternative flow 1.1 - Main Flow 1.1: “The program has never been run before, no previous session information exists”		
1.1	Sets default settings for all variables and equipment.	Main Flow 1.1
Alternative flow 1.3 - Main Flow 1.3: “There is not enough space to save our recordings to”		
1.1	Notify user that there isn't enough space to save any recordings, urge them to free up space if they want to save anything.	Main Flow 1.1
Alternative flow 1.4 - Main Flow 1.4:		

“Credentials for 3 rd party music library are invalid”		
1.1	Warn user their credentials are invalid and must be re-entered	
1.2	User re-enters their credentials	Main Flow 1.1
1.2 Alt	User ignores this and chooses a local music file instead	Main Flow 1.1

Non-functional Requirements	
Secure Credential storage	Credentials for access to Spotify, Apple Music, Soundcloud, YT music, etc, should be securely stored.
Offline use	The main system should be strictly offline. The only elements which require online access is 3 rd party music platforms

Use case 2 User loads in a new song	
Goal	User can select a new song for one of the tracks
Preconditions	User already has the music they want to use downloaded onto their XR headset.
Successful Post condition	New music is loaded in and all pre-existing effects will be applied to this new song
Failure Post condition	User is unable to select new music
Actors	User, Rave-Spin system
Trigger	User wants to make a choose a new song, either from local or from Spotify
Description	The user is already in a session, and has songs for each of their tracks. The user would like to change one of these songs for a new one. It could be from local disk or from a 3 rd party provider like Spotify, Soundcloud, etc...
Priority	Medium



Main Flow		
Step	Action	Alternate
2.1	User opens menu to choose new music to load	AF 2.1
2.2	User selects the song they want	AF 2.2
2.3	User starts DJ'ing with that new music	End of Main Flow

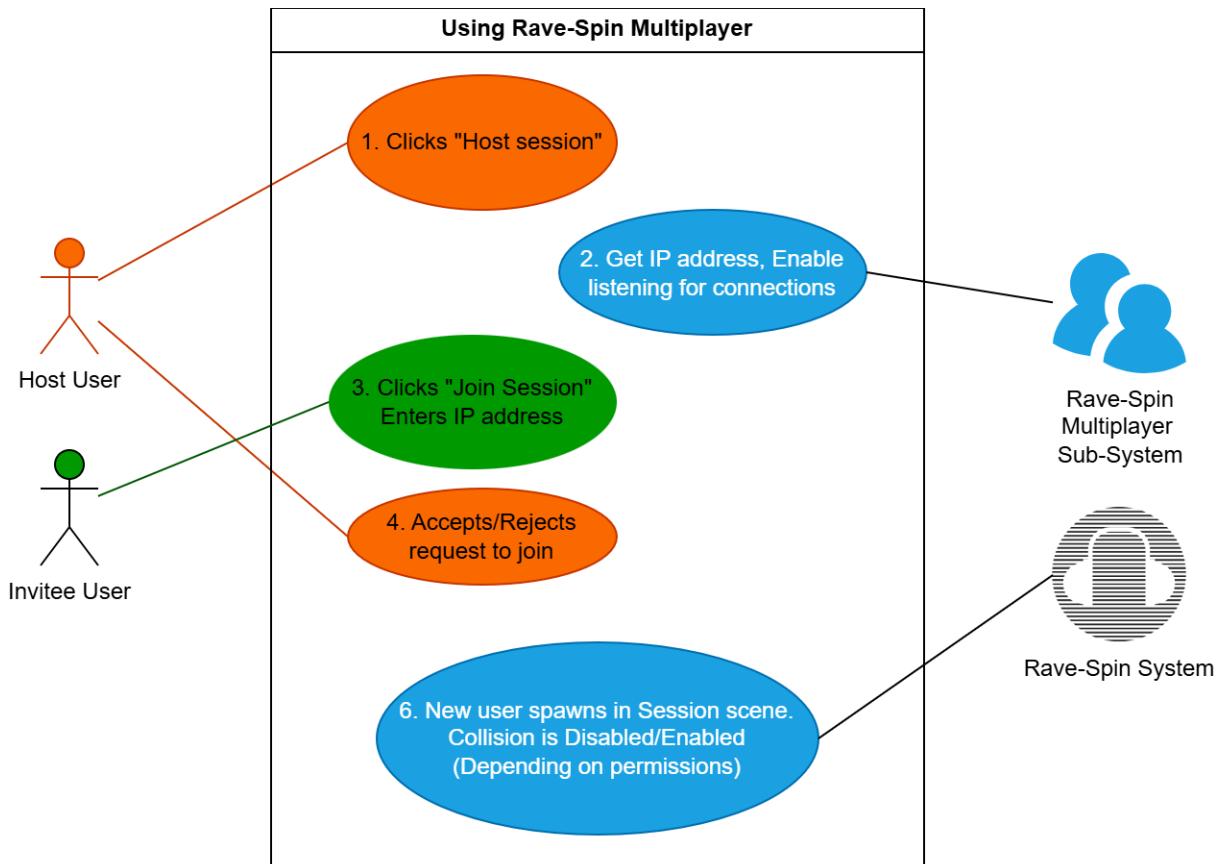
Alternate (Exceptions and Error) flows		
Step	Action	Proceed to alternate step
Alternative flow 2.1 - Main Flow 2.1: “Credentials for 3 rd party platform are invalid”		
2.1	Alert the user that access to the requested platform is not available because their credentials must be re-entered.	Main Flow 2.2
Alternative flow 2.2 - Main Flow 2.2: “We cannot download or access music from 3 rd party platform”		
2.1	For whatever reason, most likely due to network issues, Rave-Spin can not load any audio from the 3 rd party platform. Alert the user with the relevant error message, suggest user try re-enter credentials if it's not a network issue.	Main Flow 2.2

Non-functional Requirements	
Secure Credential storage	Credentials for access to Spotify, Apple Music, Soundcloud, YT music, etc, should be securely stored.

Use case 3

User invites another user to join their session

Goal	Host-User can invite another Rave-Spin user (invitee) to join their session. Having both users share the same equipment like you
Preconditions	Both users are on the same local network.
Successful Post condition	Invitee-User can join the Host-Users' session and interact with the Host-Users' setup. Both users hear the same audio.
Failure Post condition	<ul style="list-style-type: none"> - Users cannot join or host a session. - Audio does not match between both users.
Actors	Host-User, Invitee-User, Rave-Spin Multiplayer Sub-System
Trigger	A user wants to have someone else join their session.
Description	The user will host their session over LAN. They also enable anyone who joins to interact with their setup. The host-user will then tell the user they'd like to invite them to their session and will tell them the code to enter so that the invitee-user can join.
Priority	Medium



Main Flow		
Step	Action	Alternate
3.1	User selects “Host new session”	
3.2	Show User what the IP to connect to is	
3.3	User invites friend to join, showing them the IP to use	
3.4	Friend selects “Join online session”	
3.5	Friend enters IP address	AF 3.5
3.6	Friend Connects to Users’ session	AF 3.6
3.7	Friend can interact with the Hosts’ setup.	AF 3.7

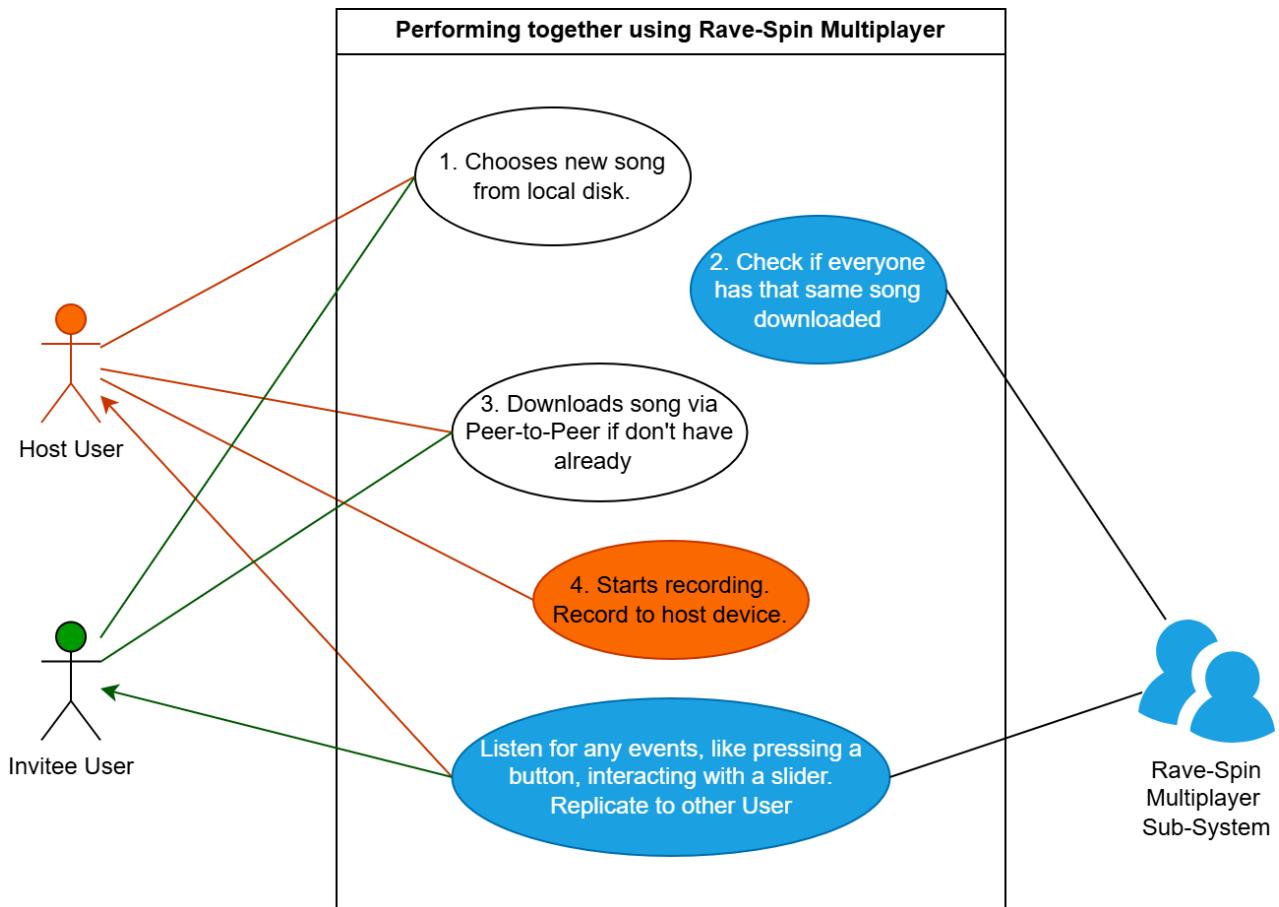
Alternate (Exceptions and Error) flows		
Step	Action	Proceed to alternate step
Alternative flow 3.5 - Main Flow 3.5: “No session can be found on the LAN with that IP address”		
3.1	Alert the user that the IP could not be found on the local network. Maybe it was entered wrong?	Main Flow 3.5
Alternative flow 3.6 - Main Flow 3.6: “The handshake between the host and the client did not succeed”		
3.1	The Friend was able to find the session on the LAN, but the Host rejected the connection. Prompt them to try again	Main Flow 3.5
Alternative flow 3.7 - Main Flow 3.7: “The Invitee is not allowed interact with the Hosts’ Setup (listen-only)”		
3.1	The Friend can join, but their collision has been disabled in the Hosts’ session. They cannot interact with the Host setup.	Main Flow 3.5
3.2	The friend must ask the host to enable collision	If “yes” – Main flow 3.8

Non-functional Requirements		
Secure UDP Client-server connection	Must only be able to transmit strictly necessary data. Audio data must be sent using fixed sizes as to prevent a potential vector of cyber-attack.	

Use case 4

2 Users play together, change song, and record

Goal	2 Users are in a session together on the same network. They want to play and record together.
Preconditions	Both users are on the same local network. Both have songs downloaded onto their headsets. Both Headsets have sufficient storage.
Successful Post condition	Both users can effectively DJ together, choose new songs to swap out mid-performance and record their performance
Failure Post condition	Severe lag / latency makes mixing together impossible / impractical.
Actors	Host-User, Invitee-User, Rave-Spin Multiplayer Sub-System
Trigger	Both users want to play together, interacting with the same setup, and record their performance.
Description	The Host user will be hosting a session which the invitee user has already joined. The invitee has been allowed to interact with the Host users' setup for this session. They both want to perform together and record something.
Priority	Low



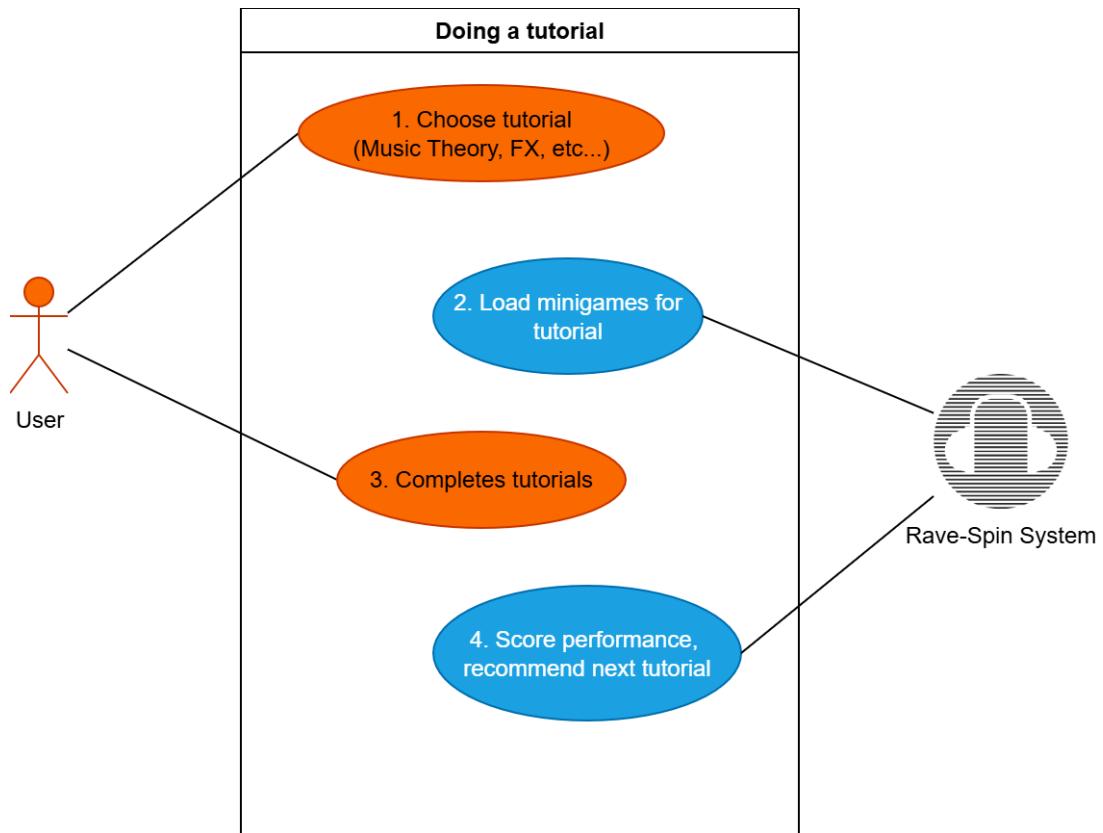
Main Flow		
Step	Action	Alternate
4.1	Host-user chooses song to be input into their controller	AF 4.1
4.2	Host-user begins recording session	
4.3	Host-user performs an action, for this use case let's assume they increase the volume of Track 2.	
4.4	The host-user's multiplayer sub-system has been notified and tries to synchronise the invitee with the new change.	
4.5	Invitee has their instance of Track 2 updated. The volume of Track 2 has increased.	
4.6	Invitee performs an action, for this use case let's assume they increase the reverb on Track 1.	AF 4.6
4.7	The Invitee multiplayer subsystem sends a notify to the host subsystem with a timestamp of the change.	
4.8	Host-user has their instance of Track 1 updated. The reverb on Track 1 has increased.	
4.9	The host finishes the recording	
4.10	Ask host-user if they want to send the recording to Invitee	AF 4.10
4.11	Ask invitee if they want to receive the recording	

Alternate (Exceptions and Error) flows		
Step	Action	Proceed to alternate step
Alternative flow 4.1 - Main Flow 4.1: “Not everyone has the same song”		
4.1	Alert the user who chose the song that the other user doesn't have the song downloaded. Do they wish to proceed and share the song with them?	
4.2	The other user who doesn't have the song is asked if they would like to download the song?	
4.3a	The other user agreed to download the song	Step 4.4a
4.3b	The other user did not agree to download the song	Step 4.4b
4.4a	Begin an FTP file transfer over WLAN between the User who has the song, and the user who doesn't.	
4.5a	File has been downloaded and now both users have the same song	Main Flow 4.2
4.4b	Alert the user who has the song that the transfer was denied. Recommend they choose another song.	Main Flow 4.1
Alternative flow 4.6 - Main Flow 4.6: “In the time that it took for the Invitee to do action ‘X’, the host-user had done a more recent action on the same item”		
4.1	For the host-user, ignore the action from the invitee.	
4.2	For the host-user, try to synchronise the newer change with the invitee.	
4.3	For the Invitee, their instance has been updated with the newer change	Main Flow 4.3 / 4.6

<p style="text-align: center;">Alternative flow 4.10 - Main Flow 4.10 / 4.11: “The request for file transfer has been denied”</p>		
4.1	Alert the relevant user that the transfer was denied and that only the host has the recording	

Non-functional Requirements	
Secure UDP Client-server connection	Must only be able to transmit strictly necessary data. Audio data must be sent using fixed sizes as to prevent a potential vector of cyber-attack.
Timestamps	If 2 changes are received by the host at the same time, the newest is applied.
File transfer	Transferring of music files is required for both users.

Use case 5 A User does a tutorial	
Goal	A user wants to do a tutorial on a topic.
Preconditions	None.
Successful Post condition	User completes tutorial and gets a score, being recommended a new one afterwards.
Failure Post condition	User does not complete tutorial
Actors	User, Rave-Spin
Trigger	User is curious about a topic, it could be about Music theory, what different effects are, beat-matching maybe, for this use case let's assume Music Theory.
Description	The user will choose tutorial on music theory. They may want a refresher course or are completely new to the topic, hopefully this will give them some extra help. They'll be given a minigame at the end and scored based on how well they do.
Priority	Low



Main Flow		
Step	Action	Alternate
5.1	User chooses a tutorial.	AF 5.1
5.2	User is given information about the topic.	
5.3	The user is then given a minigame based on that topic. Could be a Quiz, or could be an interactive exercise.	
5.4	The user is given a score based on how well they did.	
5.5	The system recommends the user another interesting topic to research.	

Alternate (Exceptions and Error) flows		
Step	Action	Proceed to alternate step
Alternative flow 5.1 - Main Flow 5.1 to 5.5: “The user exited the tutorial prematurely”		
5.1	Mark the tutorial as “Partially Read”	
5.2	Ask the user next time if they’d like to continue from where they left off	
5.3	If the user says “Yes”, skip the menu and start the tutorial from the last saved point.	

Non-functional Requirements	
Scoring system	A system of standardised scores that can be engaging and fun for the users.

4 System Design

4.1 Introduction

This section will cover how I will structure the development of Rave-Spin. The ideal for the project would be a fully integrated system all on the Meta Quest 3 headset that can be used with just your hands. No external tools or accessories required.

4.2 Software Methodology

When developing a project, there's multiple different methodologies and approaches you can take. All with their own pros and cons. For the sake of this particular project, it'll be just me working on it so that must be taken into consideration.

Some methodologies exist such as **Agile**, which is an iterative, cycling approach to software design, with daily reviews on the teams progress and "sprints" which includes planning, task allocation, execution, review, and adjustment phases which allows for quick feedback, address issues as they arise, and enhance project quality progressively, is inappropriate for this project as it's just a solo developer working on this.

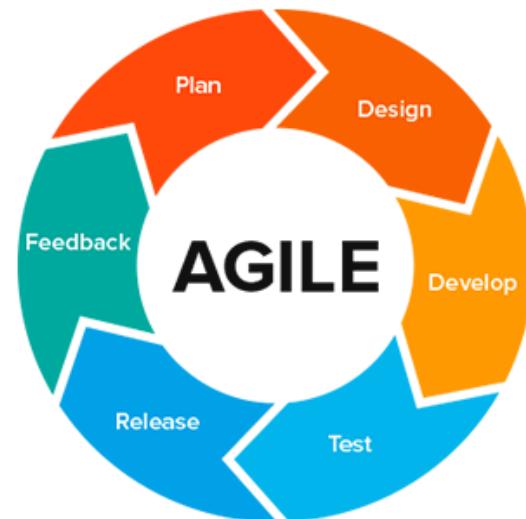


Figure 4.2 A – Agile diagram

Another popular methodology is the **Waterfall** methodology which is a linear approach to software development breaking the whole process into 7 distinct steps. Changes to the initial requirements can be slower but the path of development is straight forward from requirements gathering to deployment and then to maintenance.

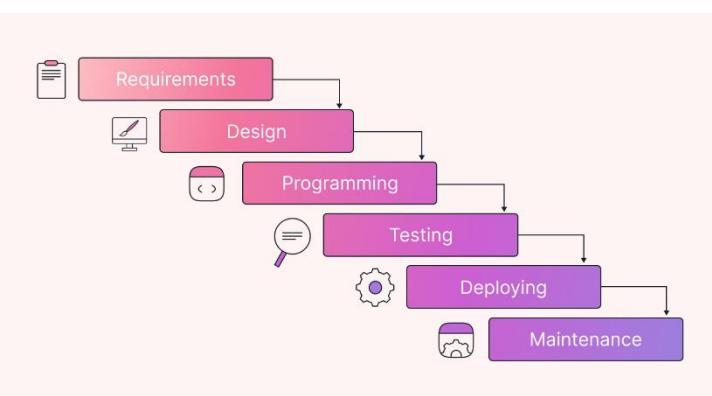


Figure 4.2 B – Waterfall Method diagram

Rapid Application Development is the methodology I'll be going for as it provides the best of both worlds. It's perfect for a solo developer, and can handle changes to requirements quite well. It mostly focuses on making prototypes, testing them, get feedback, and iterate on the prototype some more. You keep going until the prototype seems solid and you then move to the construction phase which is where you refine it and make hopefully release it then afterwards.

It's this focus on prototype development and feedback which will be useful because I can get feedback on what works and what doesn't quickly without delving too deep on something which isn't worth it, and focus more on what does work, and as I would also be using the end product for my own personal use too, I could also give valid feedback. Requirements might change so waterfall is no good for the project timeline and agile is aimed towards a team so as a solo developer that doesn't work either.

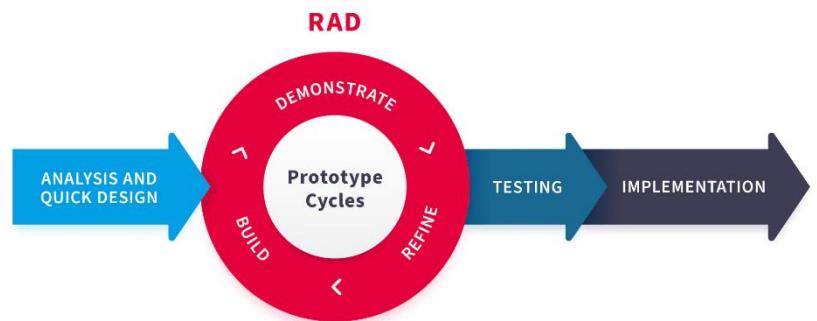


Figure 4.2 C – R.A.D diagram

4.3 Overview of System

For this project, I'm implementing a monolithic architecture. All components from rendering, audio processing, multiplayer via users joining via LAN, credentials, can all be performed by the XR headset using when powered by the Godot engine. In future I could implement a Client-Server architecture where certain equipment data and models could be sent to the headset from a server or allow people to join each others sessions from anywhere in the world, not just Local Area Network, but for now this is what we'll be focusing on for this project timeframe. Please see this simplified diagram for a quick overview of the system.

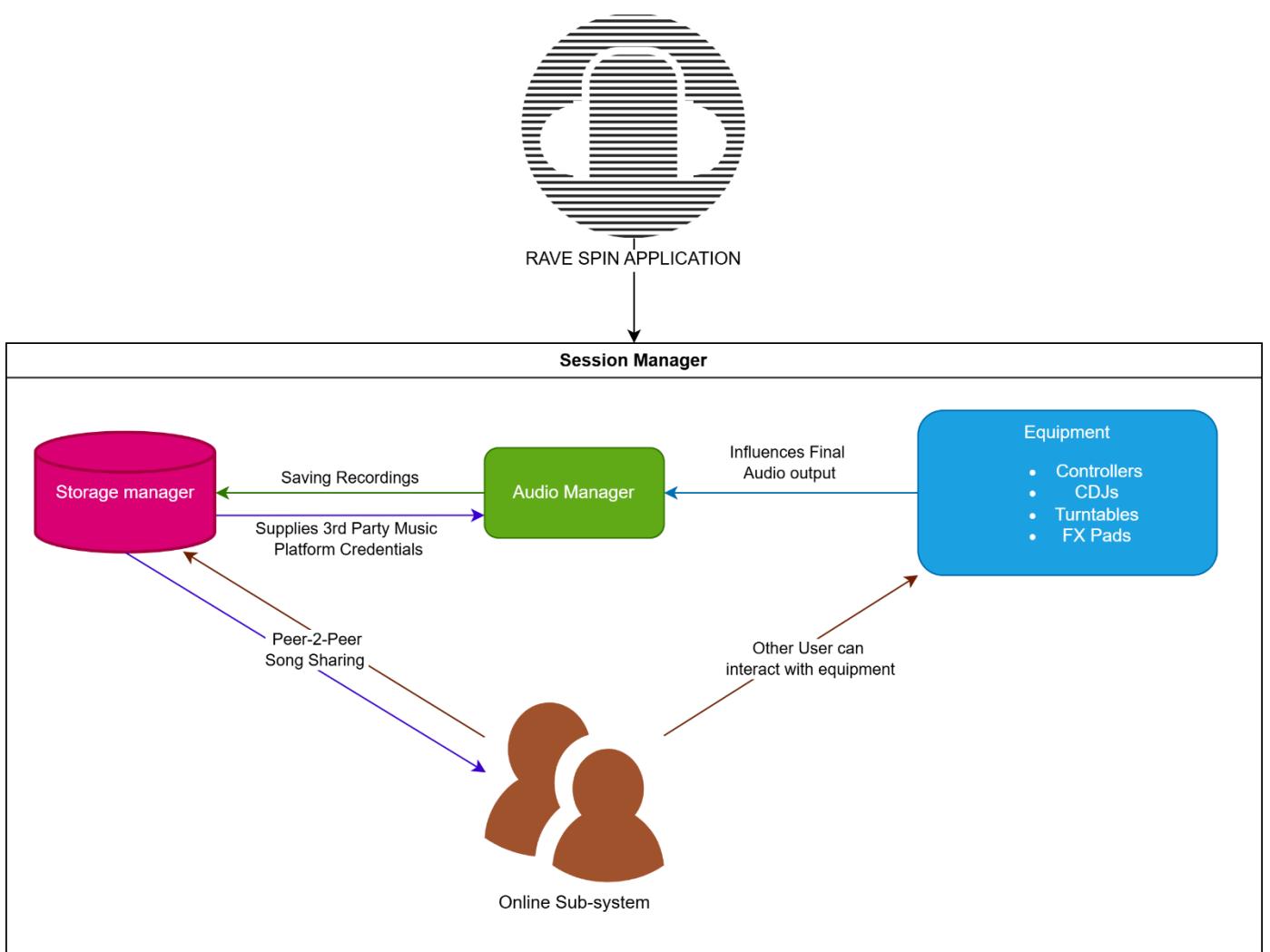


Figure 4.3 A – Overview of Rave-Spin Architecture

4.4 Design System

This system is going to have several different parts. In this section we'll discuss the different aspects of the system.

4.4.1 User interface / Sitemap

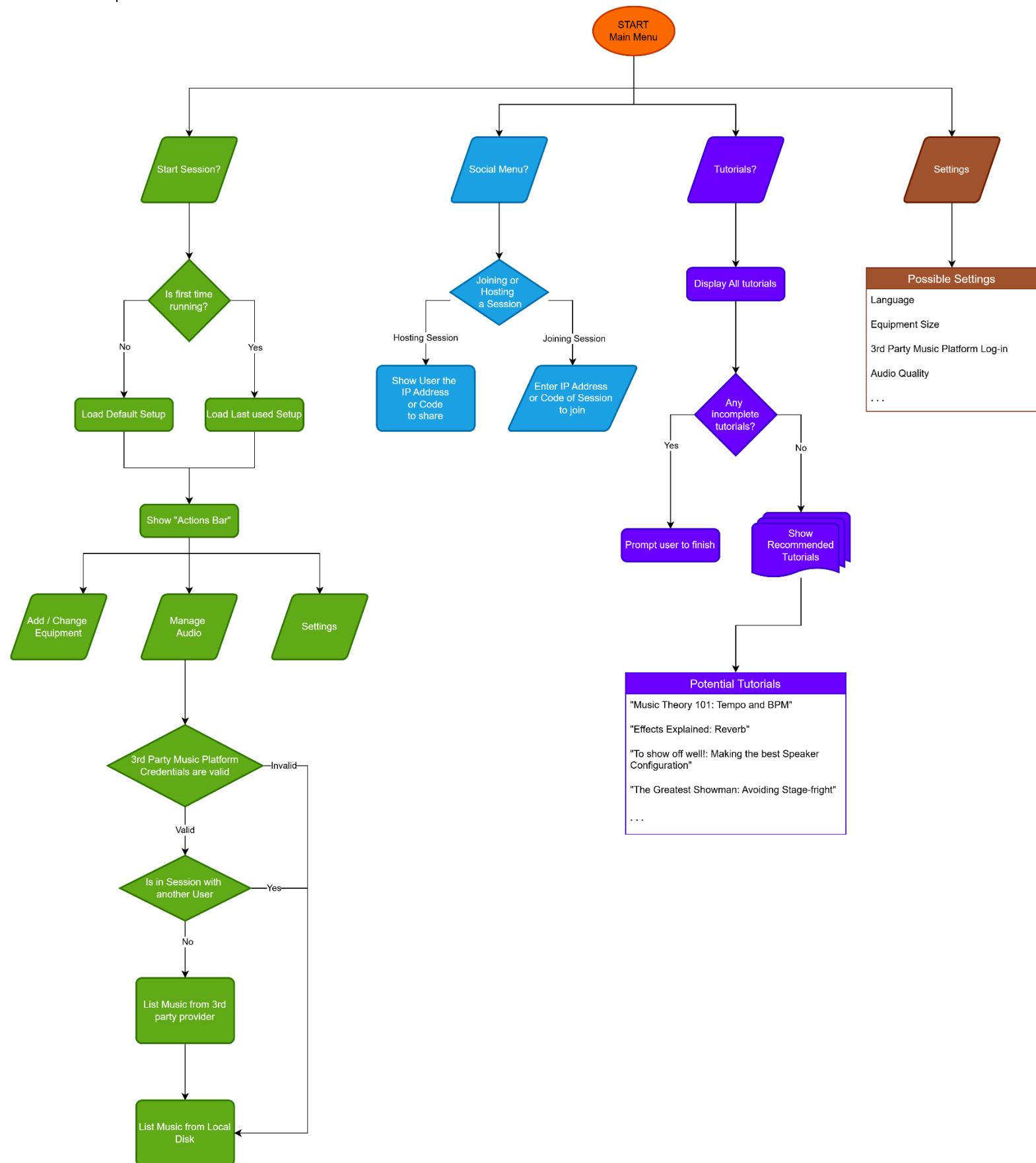


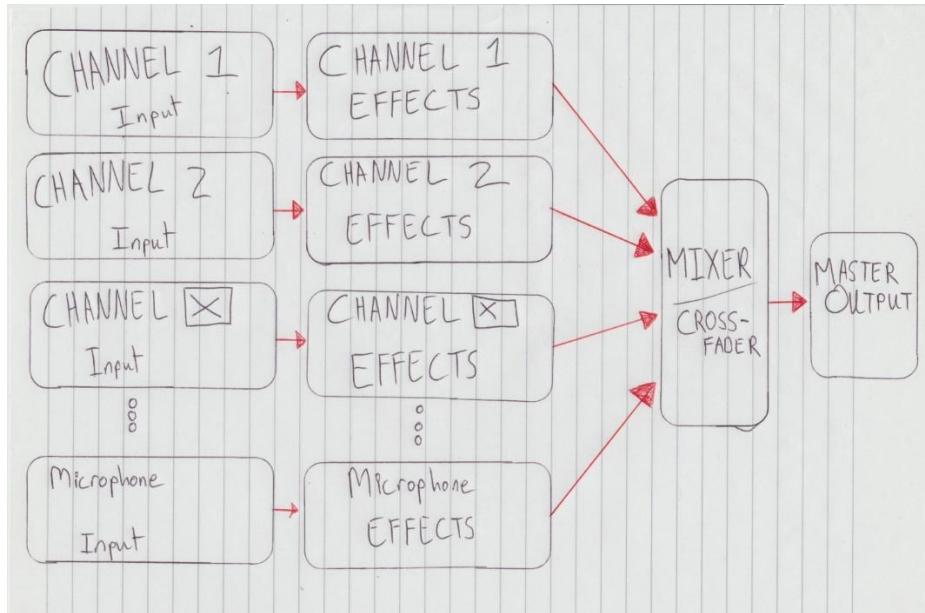
Figure 4.4 A – Main User interface flow map

As we can see in the diagram in **figure 4.4 A** the User interface should follow this general structure. This also demonstrates the general flow of the application and how the user could operate it.

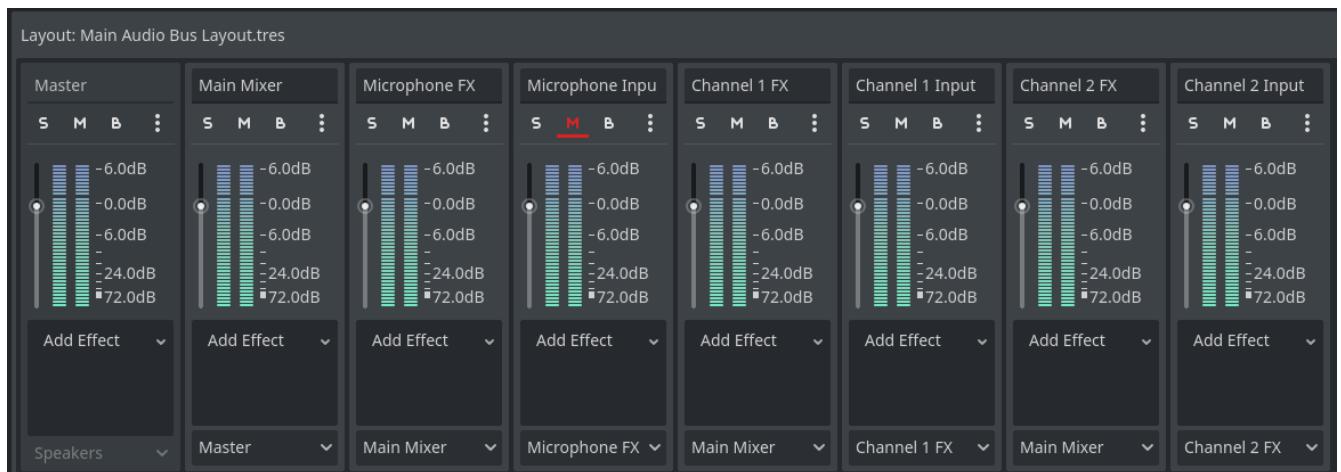
4.5 Audio Bus Layout

As shown in research section, we need to have appropriate Audio bus layout. What we're going to do for Rave-Spin is for each of our tracks (*imagine our “tracks” as what will get our music files, apply the effects to them, and then output the end result. This is a track, each stage it went through was a separate channel*) Assuming we have 2 tracks that will both be fed music as an input, each track will need to have the channel bus with just the original audio data with no effects applied. This should then go to channels’ effect bus which will apply our effects, which then gets sent to a mixer which can influence how much this track will affect the final master output.

We can see the quick sketch diagram here:



The here's the Godot implementation:



4.6 Controller Board (DDJ-FLX4)

The way I aim to replicate this is that the board will be a Godot 3D Node. It will take Inputs such as what players to use, the Audio file to use, and what channels to use.

It will operate as it should on the audio. Sending original audio to the relevant input channel. Applying whatever effects as is determined by the node itself, then send it's output to the relevant channel. It will also influence the mixer and decide how the final output channel will get it's audio.



Figure 4.6 A – The FLX4 with the Controller.gd script attached

The model itself is comprised of 2 main parts. The 3D mesh which is made in Blender (Please keep in mind the model is not complete) and in **figure 4.6 A** we can see how it's laid out. In Blender I'm re-using the same meshes as much as possible to help with memory usage.

Because Blender can export to GLTF, and Godot natively supports GLTF, anything I import into Godot will retain this same hierarchy. Repeat meshes aren't saved separately and rendered as unique, they're recycled, which can help with lowering memory usage, which we can see in **figure 4.6 B**.

Blender Layout

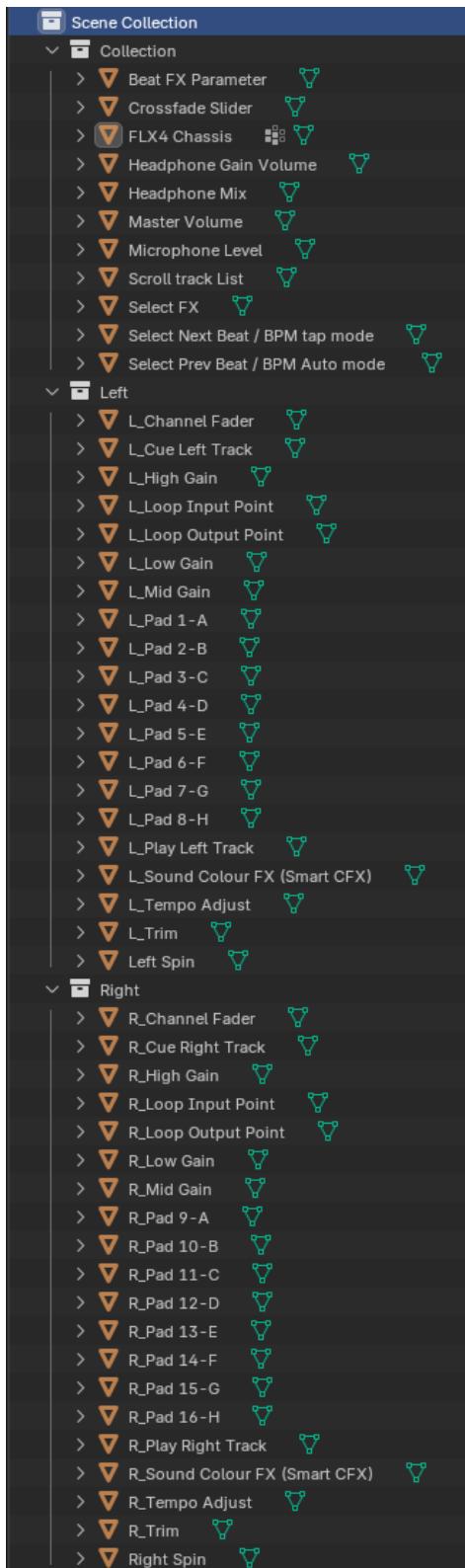


Figure 4.6 A – Layout of prototype FLX4 in Blender

Divided into General controls, Left (Track 1) controls, and Right (Track 2) controls

Godot

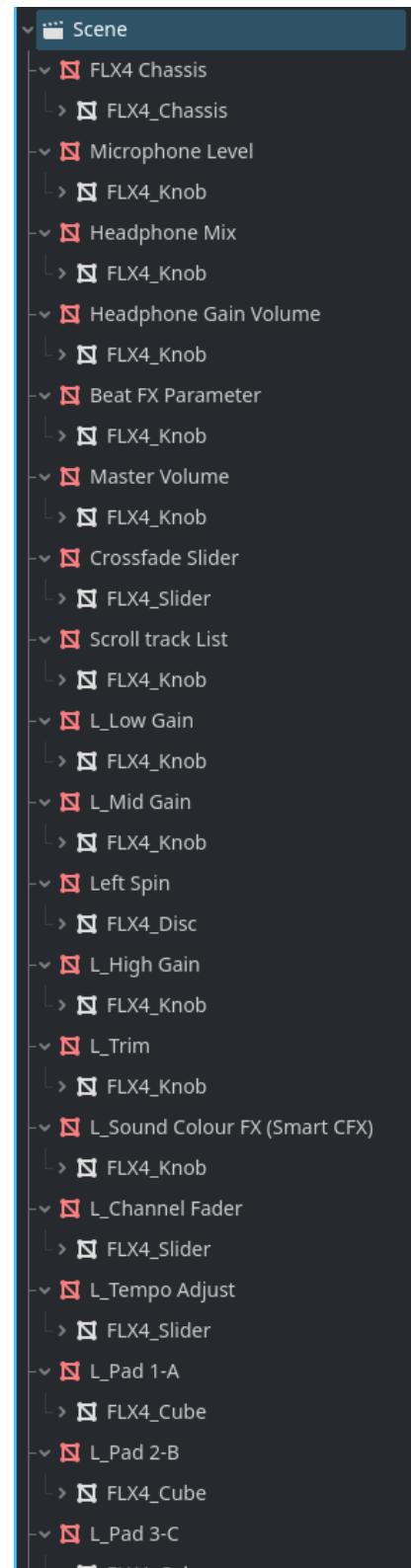


Figure 4.6 B – Layout imported FLX4 in Godot

The names stay the same and meshes are re-used / recycled instead of all being unique.

5. Testing and Evaluation

5.1 Introduction

A key part of the testing and evaluation process, will be to determine the key areas of focus, a questionnaire will be sent out to individuals whose experience DJing could range from experts, intermediate, and absolute beginners. Professional and intermediate DJs could have different requirements and desire different features than those whom are just starting off. By identifying what features are most important to these 3 distinct groups, I can make better use of my time and put more effort and work into the features that matter most.

5.2 Plan for Testing

For testing, I have already been in touch with a good few Irish and UK DJs who have expressed strong interest in evaluating the prototypes produced during this project. Given that I have adopted the Rapid Application Development (RAD) methodology, which places emphasis on iterative prototyping and continuous user feedback, this is well suited to the planned workflow. Each prototype can be deployed to these DJs for hands-on evaluation, with their feedback feeding directly into subsequent development cycles.

One issue is that several of the Irish DJs do not currently own XR headsets. To address this while still obtaining realistic user feedback, we've agreed that I will travel to them with my own XR headset, preloaded with the latest prototype build, and we can test it out then!

This will allow them to test the system in person and provide immediate feedback on usability, realism and overall experience. For system testing, we will work through the defined use cases from Section 3.3. Successful completion of these use cases without major issues will be treated as an indicator that the corresponding system requirements have been met.

For unit testing of individual components, I will create dedicated Godot scenes that isolate the element under test. These scenes will contain only the relevant unit (*for example, a specific control surface or interaction module*), with any required supporting data such as audio tracks pre-loaded for the scene. This approach is already in use for the “Arena” scene in Godot, which currently contains only the DDJ-FLX4 model. Predefined music and interaction behaviours are used there to verify that controls respond as expected and that the underlying logic functions correctly in isolation.

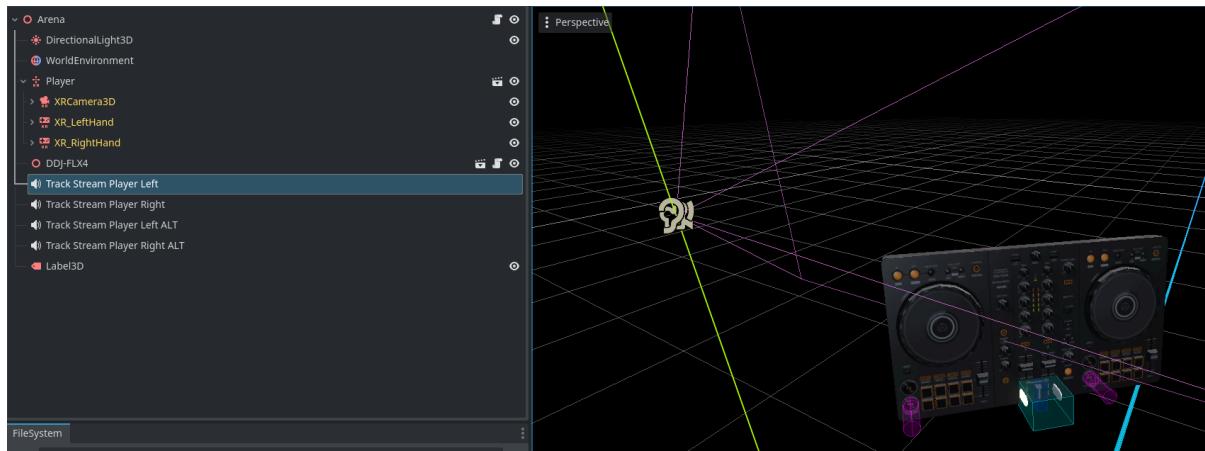


Figure 5.2 A – Screenshot Unit testing “Arena” godot scene

5.3 Plan for Evaluation

My own experience at this current moment in time, would be that closer to a beginner, so the difficulties that I may run into may be very similar to what another beginner DJ may experience. I will still offer the prototype testing to other beginner DJs anyway to verify my results.

For the 3 main groups of DJs, the beginners, the intermediates, and the professionals, each month during the testing phase of this project, I’d like to have 1-4 people from each group try out the latest prototype and get their feedback, get them to try make a super quick mix.

If they’re a beginner try out a tutorial they’ve never done before and get their feedback if it was useful, interesting, boring, etc... and see if I could test multiplayer out and have a DJ’ing session with 2 users at the same time.

I would deem 4-6 prototype testing sessions to be enough but as some of the DJs who have agreed to test it have shown an unbelievable degree of enthusiasm and excitement for this project. I may conduct more testing sessions, both to get even more accurate feedback and partly to get them to shut up asking about it.

Before testing, I will prepare a feedback form and ask for their thoughts mid-use so that I can get a detailed report of their thoughts and experience.

6. System Prototype

This section describes the current prototype of the Rave-Spin system, focusing on how the Pioneer DDJ-FLX4 has been replicated in XR, how interaction and audio behaviour are implemented, and what technical issues have been encountered so far.

6.1 Introduction

The primary objective of the prototype is to create an accurate virtual replication of the Pioneer DDJ-FLX4 that closely emulates the behaviour of the physical device. The prototype focuses on modelling both the visual layout and the functional aspects of the controller so that user interactions in XR produce outcomes that are consistent with real-world usage.

6.2 Prototype Development

Prototype development so far has concentrated on three main areas:

- XR interaction
- control-to-audio mapping
- audio bus layout

Hand interaction and input handling

At present, XR hand interaction is implemented using a small collision shape at the tip of the index finger for each virtual hand. This allows the user to “touch” controls on the DDJ-FLX4 model, which is sufficient for initial testing but not yet natural or intuitive.

What I want to happen is buttons should be activated through a pointing and tapping action, like how a user would press them in real life. Sliders and knobs should respond to a “pinch” gesture, with their values changing smoothly as the pinched hand moves along the appropriate axis. The jog wheels should respond to a flat-hand gesture or simple contact, letting users nudge and scratch them in a realistic way.

Control behaviour and audio response

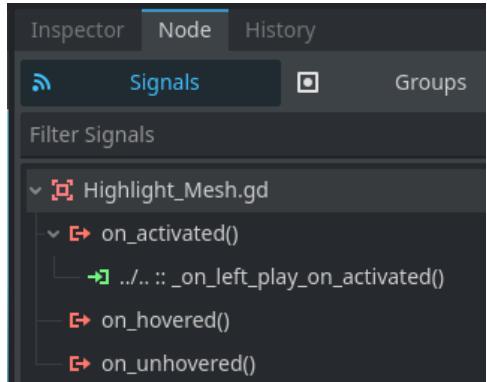
Each hardware control on the DDJ-FLX4 (e.g. *faders, knobs, jog wheels, FX pads*) should affect the corresponding audio signal in the same way as on the real device. So far only Play / Pause is functional.

Bus layout implementation

The prototype also aims to replicate the internal bus layout of a normal Controller as closely as possible within the Godot engine while preserving performance. At the moment the individual tracks are routed through the different channels correctly and concatenate and output as expected.

6.3 Results

The prototype will be evaluated against the test cases defined in Section 3.2 “Requirements analysis”. The main issues encountered so far are as follows:

- **Spawn position of the controller:** When entering a session, the virtual DDJ-FLX4 initially appears far away from the user. A temporary workaround has been to switch from a room-scale boundary to a stationary boundary on the headset, which effectively pulls the controller closer to the user. Without this workaround, the controller must be manually repositioned in Godot by dragging it closer to the origin.
- **Area3D and collision detection issues:** Collision events and signals for Area Collision nodes did not function at all at first. This was fixed after discovering and correctly configuring settings in the “Node” tab in Godot. Once hooked up, Area3D nodes and their collision shapes began responding correctly to hand interactions.
- **Slider tomfoolery:** Slider controls currently do not behave correctly. When the user attempts to move a slider horizontally from left to right, the slider may jump to a random in global space and then move vertically instead of tracking the hand’s intended horizontal movement. This is probably a mess up between local and global positioning.
- **Headset deployment issues:** Multiple issues have been encountered when deploying builds to the XR headset, mostly build configuration errors. These issues have since been fixed.

Despite these issues, the prototype has successfully demonstrated that the DDJ-FLX4 can be done in XR, basic control interaction is possible, and an initial version of the audio routing structure is in place. We saw this with Tribe XR school (see section 2.2.1) as they were able to do it with few issues.

6.4 Evaluation

The prototype will be considered highly successful if it can emulate the DDJ-FLX4 with near-perfect functional accuracy in XR so that experienced DJs can use it in a way that feels natural and familiar. At the current stage of development, this is not anywhere remotely CLOSE to true, but meaningful progress has been made towards it!

7. Issues and Future Work

7.1 Introduction

This section summarises the current progress of the project and outlines the work that remains to be completed. It also identifies key issues and risks encountered so far, and describes the planned future development required to move from the current prototype towards the intended final system.

7.2 Issues and Risks

Adjusting to Godot

Godot has been a nightmare to adjust to. Setting up OpenXR was ok due to the classes and sample projects provided by Bryan Duggan. I am most familiar with the structure and workings of Unreal Engine, as mentioned previously I cannot use Unreal engine primarily due to the audio issues I mentioned. Moving from “Unreal logic” and structure, to Godot, was a nightmare. One example was getting Signals to work in Godot which proved to be cumbersome at first as I could not figure out why an entering an Area3D wasn’t triggering code in another node despite the code and names being the exact same as the previously mentioned sample code. It was only after discovering the “nodes” section that I realised you had to manually hook up signals. I spent days debugging this before I realised this. Another issue was that Godot Nodes can only have one script attached. If this were Unreal, I would most likely have had different actors (*Node3D in Godot terminology*) interacting with each other via casting function calls and interfaces. In Godot doing things this way is not immediately obvious. I may need to refactor much of my code later but learning how to do code “The Godot Way” is proving to be a challenge.

Local vs Global position

Another issue is with Godot Local vs global position confusion. When trying to make a slider for the prototype, rather than the slider going “2 units to the right” it may snap to a whole different position altogether and go 2 units in a whole other direction. The Godot XR toolkit plugin already comes with prefabricated slider functionality so I will research how they work and consider using them instead.

Is there enough demand on XR

One major risk that may occur, is after I finalise the results from my questionnaire, if enough people mention that they would prefer a Desktop application instead of an XR one, I may need to investigate how I could accommodate for both systems. On paper this could be easily implemented, perhaps a fixed camera instead of one whose location is based upon the local XR headset and checking for mouse and keyboard input instead of XR controllers, but more testing will be needed to verify this should the questionnaire results show that this will be something that needs serious consideration.

Potential legal issues

I don't know if Pioneer would be happy that I'm aiming to make a perfect emulation of one of their most popular products, so if further research will be needed to check and see if this project would be able to last outside of an educational project. The fact that the whole project, from start to finish is FOSS should give some leeway, but I am not sure as my legal knowledge is minimal at best. If this project were to go public in a finished state, I may need to remove Pioneer branding

7.3 Plans and Future Work

The immediate priority for future work is to improve the feel and reliability of the core XR controls and interactions. At the moment, many interactions still feel unpolished; sliders are unreliable, and only buttons are implemented. Before expanding the feature set, the goal is to ensure that the DDJ-FLX4 emulation is smooth, responsive and enjoyable to use.

If the core equipment does not behave convincingly, the project fails in its primary aim of providing a useful DJ tool in XR.

Audio track selection and loading

Another key task is to implement robust audio track selection within the application. Currently, tracks are predefined in the Godot editor, which is not acceptable for real usage.

The next steps are:

- Implement track loading from local disk, allowing users to choose their own audio files at runtime.
- Only after that, investigate integration with 3rd party music platforms like Spotify or SoundCloud.

Multiplayer functionality

Multiplayer support, specifically the ability to join other users' sessions, is another major area of planned work. I have already consulted with a previous Final Year Project student who implemented multiplayer in Godot. Their main point was that, at a basic level, joining another user's session largely requires knowledge of the host's IP address.

I will conduct further research into the networking requirements and constraints in Godot, but this suggests that LAN-based multiplayer is realistically achievable in the project timeframe. The initial focus will therefore be on local network sessions, with wide-area or internet-based multiplayer considered too low a priority for development.

Menu and UI development

A main menu and supporting UI will also be needed. Gonna need:

- Starting or joining WLAN DJ sessions
- Selecting and changing equipment
- Choosing and managing audio tracks
- Settings for audio quality options, general preferences, language, etc...

The requirements for what will be needed can be seen in **figure 4.4A.**

7.4 Project Plan with Chart

The project is divided into 5 primary milestones, so far things are proceeding as planned; their progress should be seen in the Gantt chart below:



References

- [1] M. Conrad, D. Kablitz, and S. Schumann, ‘Learning effectiveness of immersive virtual reality in education and training: A systematic review of findings’, *Comput. Educ. X Real.*, vol. 4, p. 100053, Jan. 2024, doi: 10.1016/j.cexr.2024.100053.
- [2] ‘Pioneer DJ DDJ-FLX10: All specifications & features’, Pioneer DJ. Accessed: Nov. 23, 2025. [Online]. Available: <https://www.pioneerdj.com/en/product/controller/ddj-flx10/black/specifications/>
- [3] H. H. published, “A strong contender for the best beginner’s DJ controller on the market right now”: Pioneer DDJ-FLX4 review’, MusicRadar. Accessed: Nov. 23, 2025. [Online]. Available: <https://www.musicradar.com/reviews/pioneer-ddj-flx4-review>
- [4] ‘Tribe XR partners with AlphaTheta Corporation to bring Pioneer DJ – the world’s leading DJ hardware and software brand – to virtual reality - News - Pioneer DJ News’, Pioneer DJ. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.pioneerdj.com/en-us/news/2021/tribe-xr-partners-with-alphatheta-corporation/>
- [5] ‘Tribe XR | DJ Academy on Steam’. Accessed: Oct. 12, 2025. [Online]. Available: https://store.steampowered.com/app/877850/Tribe_XR_DJ_Academy/
- [6] ‘Tribe XR DJ Academy’. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.tribexr.com/#features>
- [7] ‘Ableton Live in VR lets you play as a disembodied Daft Punk head’, CDM Create Digital Music. Accessed: Oct. 12, 2025. [Online]. Available: <https://cdm.link/newswires/ableton-live-vr-lets-play-disembodied-daft-punk-head/>
- [8] ‘AliveInVR on Steam’. Accessed: Oct. 12, 2025. [Online]. Available: <https://store.steampowered.com/app/674180/AliveInVR/>
- [9] ‘Vinyl Reality - DJ in VR’, Vinyl Reality. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.vinyl-reality.com/>
- [10] ‘Vinyl Reality - DJ in VR on Steam’. Accessed: Oct. 12, 2025. [Online]. Available: https://store.steampowered.com/app/642770/Vinyl_Reality_DJ_in_VR/
- [11] ‘Algoriddim djay Pro 5’, Digital DJ Tips. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.digitaldjtips.com/reviews/algoriddim-djay-pro-5/>
- [12] ‘Algoriddim Djay Pro review & performance tests’, DeeJay Plaza. Accessed: Oct. 12, 2025. [Online]. Available: <https://www.deejayplaza.com/en/articles/djay-pro-review>
- [13] J. Hartley, ‘Algoriddim djay Pro Review - We Are Crossfader’, We Are Crossfader - Learn How To DJ Online. Accessed: Oct. 12, 2025. [Online]. Available: <https://wearecrossfader.co.uk/blog/algoriddim-djay-pro-review/>
- [14] T. and How-To’s and N. and Reviews, ‘How does a mixer work? | The Basics’, reverb.com. Accessed: Nov. 23, 2025. [Online]. Available: <https://reverb.com/fr/news/how-does-a-mixer-work-the-basics>
- [15] R. Dejaegher, ‘DJ Audio Routing 303: Analogue Mixers, Guitar Pedals, and External Soundcards’, DJ TechTools. Accessed: Nov. 23, 2025. [Online]. Available: <https://djtechtools.com/2015/06/28/audio-routing-101-analog-mixers-guitar-pedals-and-external-soundcards/>
- [16] DJ.Studio, ‘Mixing dB Levels for DJs - Understanding Gain Staging’, DJ.Studio | Mixing dB Levels for DJs - Understanding Gain Staging. Accessed: Nov. 23, 2025. [Online]. Available: <https://dj.studio/blog/mixing-db-levels>
- [17] J. Hartley, ‘The Inputs and Outputs on a DJ mixer - We Are Crossfader’, We Are Crossfader - Learn How To DJ Online. Accessed: Nov. 23, 2025. [Online]. Available: <https://wearecrossfader.co.uk/blog/the-inputs-and-outputs-on-a-dj-mixer/>
- [18] H. H. published, “A strong contender for the best beginner’s DJ controller on the market right now”: Pioneer DDJ-FLX4 review’, MusicRadar. Accessed: Nov. 23, 2025. [Online]. Available: <https://www.musicradar.com/reviews/pioneer-ddj-flx4-review>

[19] P. Morse, 'The 7 Best DJ Controllers For 2025', Digital DJ Tips. Accessed: Oct. 08, 2025. [Online]. Available: <https://www.digitaldjtips.com/best-dj-controllers-2025/>

Appendix A: Prompts Used with ChatGPT

Before supplying prompts, I uploaded my project proposal document as to provide better context so that the results would be more in-line with what would be desired.

Chat #1

ChatGPT model used was **5.1 “Thinking”**

Starter prompt

“I'm writing my interim report for Rave-Spin, the XR DJ App, and want to make it sound more proper as it's for college. Please re-write re-phrase the prompts I'll supply to you, anything in double brackets [[]] is meant for you for context and is not part of the original text. Use what you know about my project for clues and guides.”

Original Text:

1. Introduction [[Section Header]] This project, Rave-Spin, aims to develop a portable DJ setup in XR so that anyone can DJ on-the-go. The equipment used by DJ's is typically physically large and cumbersome to transport (the Pioneer DDJ-FLX10, the gold standard of DJ controllers, is 40cm long and 6.7kg), and anyone just trying out DJ'ing for the first time might be turned off by the sheer initial cost of the equipment which can easily go into the thousands of euros, as well as the music-theory knowledge that's required. As someone who loves the Irish rave-scene and electronic and techno music, DJ'ing is something I've always been interested in but never got the chance to due to the steep initial cost. This project hopes to help with all these aspects, by allowing experienced DJs to have their whole setups be emulated in XR so that they can DJ on-the-go using their just their hands and headsets, and don't always need to carry their entire setups with them. People who would like to DJ but whom have never done so before could load this application onto their headsets and experience the pre-included tutorials which will feature gamification elements to make the learning experience more fun.

Refined Text:

Rave-Spin is an XR-based DJ application designed to provide a portable, immersive alternative to traditional DJ equipment. Conventional DJ controllers and mixers are typically large, heavy, and expensive, making them difficult to transport and inaccessible for many beginners. High-end setups can cost thousands of euros and often require a solid grounding in music theory and DJ techniques before a user can meaningfully engage with them.

This project addresses these barriers by virtualising the DJ environment in extended reality. Rave-Spin aims to allow experienced DJs to emulate a full performance setup using only a headset and hand tracking, enabling them to practise or perform without carrying physical hardware. At the same time, the system is intended to support complete beginners by providing an accessible, low-cost entry point into DJ'ing that does not require investing in physical gear.

In addition, the application will include guided tutorials with gamified elements to make the learning process more engaging. These tutorials are intended to teach core DJ skills and concepts in an interactive way, helping users build confidence and competence over time. The overall goal is to combine the flexibility of XR with intuitive interaction design to create a portable DJ experience that is both practical for experienced users and approachable for newcomers.

Original Text:

1.1 Project Background [[Sub-Section Heading]] There are several different aspects to being a good DJ. Some enter the profession/hobby with vast preexisting knowledge of music theory, while others may have no preexisting knowledge at all, which can severely impact their ability. For someone starting off getting into DJ'ing, you can absolutely teach yourself using various tutorials from all around the internet. YouTube is especially helpful with this, but there's also a vast array of different equipment that DJs can use as well, all with various different features and quirks, so very typically you'll only find tutorials for the most common equipment. Thankfully many of the tools operate in a similar fashion, but it still can be very confusing. Experienced DJs might also have equipment which can be quite physically large and cumbersome to transport. It'd be handy if there was a way to emulate their setups so they could use it on-the-go if they wanted.

Refined Text:

Effective DJing draws on a combination of musical, technical, and performance skills. Some individuals enter the space with a strong foundation in music theory, while others begin with little to no prior knowledge, which can significantly limit their ability to progress. In addition to understanding rhythm, phrasing, and harmony, aspiring DJs must also learn to operate a range of hardware and software tools, each with its own workflows and terminology.

While it is possible to self-teach using online resources, the learning environment is fragmented. Platforms such as YouTube provide a large number of tutorials, but these are often tied to specific, commonly used controllers or software. Given the wide variety of DJ equipment available, many devices receive limited coverage, leaving beginners to generalise concepts across different layouts and feature sets, which can be confusing and discouraging.

For experienced DJs, the challenge is different but related. Professional-grade setups are often physically large, heavy, and expensive, making them impractical to transport regularly, especially for practice or casual use. A system that can emulate familiar DJ workflows in an XR environment has the potential to reduce reliance on physical hardware, enabling DJs to rehearse or perform using only a headset and tracked inputs, without the logistical overhead of carrying full equipment.

Original Text:

Controller [[What we're going to explain]]

This is what will be referred to by the Controller. Typically, it's a board with 2 or more turntables and built in FX pads. They're nearly a one-stop-shop for everything a DJ will need. [[Shows a picture of the DDJ-FLX4 and FLX6]]

MIDI Controller [[What we're going to explain]]

Refers to a "Piano-like" component that controls the pitch and velocity of whatever audio sample it's linked to. [[Shows a picture of the nektar impact GX Mini]]

FX Pad [[What we're going to explain]]

A board which has different pads which all correspond to different audio samples that be shot-off with a press.

[[shows a picture of the Pioneer DDJ-XP2]]

CDJ [[What we're going to explain]]

Meant for controlling one track. Has a turntable-built in.

[[shows picture of Pioneer CDJ-3000]]

Refined Text:

Controller

In this report, a controller refers to an all-in-one DJ controller: a single unit that typically combines two or more virtual "decks" (with jog wheels) and a central mixer section, often including built-in performance pads and basic effects controls. For most use cases, it functions as a near complete solution for mixing and manipulating tracks.

MIDI Controller

A MIDI controller refers here to a keyboard-style device used to send MIDI data to software or hardware instruments. It does not generate sound on its own; instead, it controls parameters such as pitch, velocity and other aspects of whatever virtual instrument or audio source it is mapped to.

FX Pad

An FX pad is a pad-based performance controller consisting of multiple pads, each mapped to a specific sample, loop or effect. Pressing a pad triggers the associated sound or effect, allowing rapid, rhythmic triggering during a performance.

CDJ

A CDJ refers to a standalone digital media player designed to control one track at a time. It typically includes a large jog wheel/platter, transport controls and a display, and is commonly used in professional and club-standard DJ setups.

Original Text:

For testing, I have already been in contact with a huge number of Irish and UK DJ's who expressed great enthusiasm to test out the various prototypes that will come up during the development of this project. As I have decided to use the Rapid Application Development methodology, which relies heavily on iterative prototypes, this would be ideal as I'd have prototypes ready for them to test and give feedback on. One issue I will have though is several of the Irish DJs have already stated that they do not have XR headsets.

To remedy this and get their accurate feedback, we have already agreed that I would drive up to them with my own XR headset with the latest prototypes loaded on, and get their immediate feedback in real time. For system testing we will be running through the use cases mentioned in section 3.3. If they are able to complete with no issues then that will be considered a success. For unit testing of individual components, I will have dedicated Godot scenes which will have nothing except for the unit I wish to test. All other information such as music data will be hardcoded into these scenes. I have already done this with my "Arena" scene in Godot which has the DDJ-FLX4 and nothing else. Music is already predefined and I can test and make sure the components and controls work as expected.

Refined Text:

For testing, I have already established contact with a substantial group of Irish and UK DJs who have expressed strong interest in evaluating the prototypes produced during this project. Given that I have adopted the Rapid Application Development (RAD) methodology, which places emphasis on iterative prototyping and continuous user feedback, this arrangement is well suited to the planned workflow. Each prototype can be deployed to these DJs for hands-on evaluation, with their feedback feeding directly into subsequent development cycles.

A practical challenge is that several of the Irish DJs do not currently own XR headsets. To address this limitation while still obtaining realistic user feedback, we have agreed that I will travel to them with my own XR headset, preloaded with the latest prototype builds. This will allow them to test the system in person and provide immediate qualitative feedback on usability, realism and overall experience. For system testing, we will work through the defined use cases from Section 3.3; successful completion of these use cases without major issues will be treated as an indicator that the corresponding system requirements have been met.

For unit testing of individual components, I will create dedicated Godot scenes that isolate the element under test. These scenes will contain only the relevant unit (for example, a specific control surface or interaction module), with any required supporting data such as audio tracks hardcoded into the scene. This approach is already in use for the "Arena" scene in Godot, which currently contains only the DDJ-FLX4 model. Predefined music and interaction behaviours are used there to verify that controls respond as expected and that the underlying logic functions correctly in isolation.

Appendix B: Additional Code Samples

Arena.gd

```
1  extends Node3D
2
3  var xr_interface: XRInterface
4  @onready var environment:Environment = $"WorldEnvironment".environment
5
6  func enable_passthrough() -> bool:
7      if xr_interface and xr_interface.is_passthrough_supported():
8          return xr_interface.start_passthrough()
9      else:
10         var modes = xr_interface.get_supported_environment_blend_modes()
11         if xr_interface.XR_ENV_BLEND_MODE_ALPHA_BLEND in modes:
12             xr_interface.set_environment_blend_mode(xr_interface.XR_ENV_BLEND_MODE_ALPHA_BLEND)
13             return true
14         else:
15             return false
16
17 func _ready():
18     xr_interface = XRServer.primary_interface
19     if xr_interface and xr_interface.is_initialized():
20         print("OpenXR initialised successfully")
21
22     # Turn off v-sync!
23     DisplayServer.window_set_vsync_mode(DisplayServer.VSYNC_DISABLED)
24
25     # Change our main viewport to output to the HMD
26     get_viewport().use_xr = true
27     enable_passthrough()
28
29     else:
30         print("OpenXR not initialized, please check if your headset is connected")
31 |
```

Controller.gd

```
1  extends Node3D
2
3  @export var Track_1_AudioSource : AudioStream;
4  @export var Track_2_AudioSource : AudioStream;
5  @export var Track_3_AudioSource : AudioStream; # Only Tracks 1 & 2 are priority
6  @export var Track_4_AudioSource : AudioStream;
7  # Too many issues with Init'ing
8  #var AudioSourceList : Array[AudioStream] = [Track_1_AudioSource, Track_2_AudioSource]
9
10 @export var Use_2_Track_Bus_Layout = true
11 var Bus_Layout : AudioBusLayout
12 @onready var AudioPlayerList = ["../Track Stream Player Left",
13     "../Track Stream Player Right",
14     "../Track Stream Player Left ALT",
15     "../Track Stream Player Right ALT"];
16
17 func CheckRange(minimum, maximum, value, function_name = "") -> bool:
18     if (value < minimum or value > maximum):
19         if (function_name != ""):
20             print(function_name, " was given an invalid int, it's only between ", mi
21         else:
22             print("Incorrect int given, it's only between ", minimum, " and ", maxim
23         return false
24     else:
25         return true
26
27 func LoadTrackIntoMemory(p_which_track : int):
28     if(CheckRange(0, 3, p_which_track, "LoadTrackIntoMemory")):
29         print("Spawned Player for Track ", p_which_track)
30         var current_stream : AudioStream;
31     match p_which_track:
32         0:
33             current_stream = Track_1_AudioSource
34         1:
35             current_stream = Track_2_AudioSource
36         2:
37             current_stream = Track_3_AudioSource
38         3:
39             current_stream = Track_4_AudioSource
40
41     AudioPlayerList[p_which_track].stream = current_stream
42     print("Loading Track ", p_which_track, " into memory now")
43
44 func Play_Pause(p_which_track : int):
45     if(CheckRange(0, 3, p_which_track, "Play_Pause")):
46         var current_player = AudioPlayerList[p_which_track]
47         if(current_player.playing):
48             print("Pausing Track ", p_which_track)
49             current_player.stream_paused = true
50         else:
51             print("Playing Track ", p_which_track, " now @ ", current_player.get_pla
52             current_player.stream_paused = false
53             current_player.play(current_player.get_playback_position())
54
55 func _ready() -> void:
56     LoadTrackIntoMemory(0)
57     LoadTrackIntoMemory(1)
58     if (Use_2_Track_Bus_Layout == false):
59         LoadTrackIntoMemory(2)
60         LoadTrackIntoMemory(3)
61     #CreateInteractableControl(btn_PausePlay_ref, E_CONTROLTYP
```

GD_Main_Menu.gd

Just for the Main Menu scene just to verify that it loaded

```
1  extends Node3D  
2  
3  func _ready() -> void:  
4      print("Main Level Loaded")
```

Highlight_Mesh.gd

```
1  extends Node3D
2
3  @export var Target_Mesh : MeshInstance3D
4
5  signal on_hovered
6  signal on_unhovered
7  signal on_activated
8
9  var fully_exited = true
10 @onready var highlight_mat : StandardMaterial3D
11 @onready var activated_mat : StandardMaterial3D
12
13 enum E_ActivationStates {
14     Exited,
15     Hoovered,
16     Pressed
17 }
18
19 func Load_Mats():
20     highlight_mat = load("res://Art/Materials/M_Item_Hovered.tres")
21     activated_mat = load("res://Art/Materials/M_Item_Activated.tres")
22
23
24 func _on_ready():
25     Load_Mats()
26
27
28 func HighLight(p_state = E_ActivationStates.Hoovered):
29     if (highlight_mat == null or activated_mat == null):
30         Load_Mats()
31     match p_state:
32         E_ActivationStates.Exited:
33             Target_Mesh.material_overlay = null
34             on_unhovered.emit()
35         E_ActivationStates.Hoovered:
36             Target_Mesh.material_overlay = highlight_mat
37             on_hovered.emit()
38         E_ActivationStates.Pressed:
39             Target_Mesh.material_overlay = activated_mat
40             on_activated.emit()
41
42
43 func _on_highlight_area_entered(area: Area3D) -> void:
44     HighLight(E_ActivationStates.Hoovered)
45     fully_exited = false
46
47
48 func _on_highlight_area_exited(area: Area3D) -> void:
49     HighLight(E_ActivationStates.Exited)
50     fully_exited = false
51
52
53 func _on_activation_area_entered(area: Area3D) -> void:
54     HighLight(E_ActivationStates.Pressed)
55
56
57 func _on_activation_area_exited(area: Area3D) -> void:
58     if (fully_exited):
59         HighLight(E_ActivationStates.Exited)
60     else:
61         HighLight(E_ActivationStates.Hoovered)
62 |
```

Highlight_Mesh_Slider.gd

```
1  extends "res://Scripts/Highlight_Mesh.gd"
2
3  var active = false
4  var hand_ref : Area3D
5  var starting_pos : Vector3
6  @export var Value = 0.5
7
8
9  func _on_activation_area_entered(area: Area3D) -> void:
10     if (area.name == "Hand"):
11         print("Slider Activated")
12         active = true
13         hand_ref = area
14         starting_pos = Target_Mesh.position
15
16         HighLight(E_ActivationStates.Pressed)
17
18
19  func _on_activation_area_exited(area: Area3D) -> void:
20     active = false
21     #hand_ref = null
22     if (fully_exited):
23         HighLight(E_ActivationStates.Exited)
24     else:
25         HighLight(E_ActivationStates.Hoovered)
26
27  func _process(delta: float) -> void:
28      var max_pos = "$Highlight/Max point".position.z
29      var min_pos = "$Highlight/Min point".position.z
30
31      var hand_pos = $"../../Player/XR_RightHand".position
32      if (active == false):
33          var format_string = "Rhand pos: X: %f, Y: %f, Z: %f"
34          var actual_string = format_string % [hand_pos.x, hand_pos.y, hand_pos.z]
35
36
37      if (active):
38
39          if ($Highlight/Activation.position.z <= min_pos):
40              $Highlight/Activation.position.z = min_pos * 0.9
41
42          elif ($Highlight/Activation.position.z >= max_pos):
43              $Highlight/Activation.position.z = max_pos * 0.9
44      else:
45          $Highlight/Activation.position = to_local($"../../Player/XR_RightHand".position)
46          $Highlight/Activation.position.y = 0.1
47          $Highlight/Activation.position.x = 0
48
49          Target_Mesh.global_position = $Highlight/Activation.global_position
50          var alpha = ($Highlight/Activation.position.z) + 0.5
51          print("Moving Slider, min_pos: ", min_pos * 0.9, " max_pos: ", max_pos * 0.9, " Alpha = ", alpha)
```

Appendix C: Diagram of FLX4 layout:

Pioneer Dj

DDJ-FLX4

Hardware Diagram for rekordbox for Mac/Windows



Pioneer Dj

Group	No.	Name	Function	Function (+SHIFT)
1.DECK Functions are the same for left and right decks.				
	1	PLAY/PAUSE	Play/Pause	
	2	CUE	Set/Call cue, Back cue	Jump to track start
	3	SHIFT	SHIFT	—
	4	JOG	Scratch (Vinyl mode on *)	Search
		wheel side	Pitch bend (Vinyl mode off *)	
	5	IN	Loop in	—
		loop on	Adjust Loop in point	—
	6	OUT	Loop out	—
		loop on	Adjust Loop out point	—
	7	4 BEAT/EXIT	4 Beat Loop on	Active Loop on/off
		loop on	Loop exit	
	8	CUE/LOOP <	Call cue/loop	Delete cue/loop
		loop on	Loop half	
	9	CUE/LOOP >	Call cue/loop	Save cue/loop
		loop on	Loop double	
	10	BEAT SYNC	Beat Sync on/off	Tempo Range
		press and hold	Set master deck	—
	11	TEMPO	Tempo control	
2.EFFECT				
	1	FX CH SELECT	Select Beat FX channel	
	2	FX SELECT	Select Beat FX	Select Beat FX (reverse order)
	3	BEAT <	Select beats for Beat FX	BPM auto mode on
	4	BEAT >	Select beats for Beat FX	BPM tap mode on
	5	LEVEL/DEPTH	Beat FX parameter	
	6	BEAT FX ON/OFF	Beat FX on/off	Release FX on
3.MIXER Functions are the same for CH1 and CH2.				
	1	MASTER LEVEL	Master level volume	
	2	HEADPHONE CUE (MASTER)	Master out monitor on/off	—
	3	TRIM	Trim	
	4	EQ (HI, MID, LOW)	EQ (HI, MID, LOW)	
	5	CFX	SOUND COLOR FX / SMART CFX	
	6	HEADPHONE CUE (CHANNEL)	Channel monitor on/off	Tap BPM
	7	CHANNEL FADER	Channel fader / SMART FADER	Channel fader start
	8	CROSSFADER	Crossfader / SMART FADER	Crossfader start
	9	MIC LEVEL	Mic level volume	
	10	SMART CFX	SMART CFX on/off	SMART CFX preset selection
	11	HEADPHONES M/D	Headphones monitor volume balance	
	12	HEADPHONES LEVEL	Headphones monitor level volume	
	13	SMART FADER	SMART FADER on/off	—
	14	MIC ATT.	Mic Attenuator	
	15	Android MONO/STEREO	Select MONO/STEREO for an Android device audio output	

*1 Vinyl mode can't be turned on/off via the DDJ-FLX4. But it can be turned on/off using keyboard shortcut of rekordbox.

Group	No.	Name	Function	Function (+SHIFT)
4.BROWSE				
	1	BROWSE	turn	Zoom in/out of enlarged waveforms
		press	Move cursor between track list and tree view	
	2	LOAD (DECK1)	Open a folder	
		press twice	Load a track to deck1	Move the cursor into related tracks
	3	LOAD (DECK2)	Instant doubles	—
		press twice	Load a track to deck2	Browse view
5.PERFORMANCE PADS Functions are the same for left and right decks.				
	1	HOT CUE	HOT CUE mode	KEYBOARD mode
	2	PAD FX1	PAD FX1 mode	PAD FX2 mode
	3	BEAT JUMP	BEAT JUMP mode	BEAT LOOP mode
	4	SAMPLER	SAMPLER mode	KEY SHIFT mode
	5	PERFORMANCE PADS	HOT CUE KEYBOARD	Set/Call Hot Cue Delete Hot Cue Playback starts from the Hot Cue point, with the sound pitch assigned to that pad Change the pitch set to the Performance Pad
		PAD FX1	PAD FX on	PAD FX on
		PAD FX2	PAD FX on	PAD FX on
		BEAT JUMP	Move playback position	Change the number of beats set to the Performance Pad
		BEAT LOOP	Auto Beat Loop on/off	—
		SAMPLER	Play a sampler sound	Stop a sampler sound
			Load a track to a sampler slot	
		KEY SHIFT	Change the key, with the sound pitch assigned to that pad	Change the pitch set to the Performance Pad