Development of a Musical VR Videogame

Aitor Álvarez Martín

Director: Lasse Löpfe

Grau:

Curs: 2022-23

Universitat: UPC-CITM

Index

Summary

In this University Final Thesis I will be developing a virtual reality rhythm based video game. In order to do so, I will begin by researching the history of virtual reality and rhythm games, analyze the state of the art in the genre and develop a prototype in the Unity engine.

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Keywords

rhythm, games, music, immersion, BPM, timing, virtual reality, Unity.

Glossary

Music / rhythmic games: Rhythmic / music games are video games that require players to synchronize their actions with a beat or rhythm of music, often by pressing buttons or moving in time with the music.

Virtual reality: Virtual reality is a gaming platform in which players use certain peripherals such as headsets or haptic controllers in order to increase the immersive experience. Virtual reality is characterized by fully immersing the player into the game world instead of providing interactions with the real world as platforms such as augmented or mixed reality do.

Game Engine: Software specifically designed for the creation of video games. Game engines abstract much of the complexity behind making video games into a prepackaged, easy to use system.

BPM: Beats per minute. It is widely used as a metric for measuring the speed at which a song is played and perceived.

1. Introduction

1.1 Motivation

The motivations that led me to choose this TFG theme are my love for music and video games and the possibility of combining both into a single project. Following this concept, this project represented a middle ground between a trial and a fun personal challenge that can be added to my resume.

This combination of music and video games has always fascinated me since I discovered Guitar Hero at an early age. As years passed I tried more titles from the genre and became hooked with the genre and decided to try as many musical games as I could. This and my general passion for video games led me to Virtual reality and the musical titles developed for the technology.

For this project I want to focus on the development and the design of the gameplay of the videogame, exploring topics like timing, combos, difficulty levels, and flow.

1.2 Problem formulation

Instead of a problem, my thesis poses as a guide in designing and programming a musical VR video game for the Oculus Quest 2 that later can be released for free on a digital platform. This project will document the whole process.

To achieve the completion of the thesis I must achieve the following points in the following order:

1. Program the videogame based on the design given by the company and my improvements over it.
2. Implement the assets given by the company and create an appealing UI.
3. Go through the whole process of publishing the video game.
4. Document the whole process to help future developers.

Having these three concepts as the main statements of the undergraduate thesis project, the development process will face all the main phases that a videogame encounters in its production.

1.3 General objectives

The general objectives of this project are to develop a music video game and go through the whole process of production, post-production, and release of the title following the standard method followed in the video game industry. This process includes the programming, the design, and the release of the musical VR title.

This thesis will document all the development and will dig deep into each decision taken during the whole creation of the product.

1.4 Specific objectives

1. **Programming**
    - 1.1. **Project setup (XR).**
    - 1.2. **Creation of the environment.**
    - 1.3. **Program basic objects.**
        - 1.3.1. Drums.
        - 1.3.2. Drumsticks.
        - 1.3.3. Extras.
    - 1.4. **Implementation of the music system.**
        - 1.4.1. Song information extraction.
        - 1.4.2. Song BPM extraction.
        - 1.4.3. Visual representation.
        - 1.4.4. Sound representation
    - 1.5. **Menus and UI.**
        - 1.5.1. Start menu.
            - 1.5.1.1. Song selection button.
            - 1.5.1.2. Options.
            - 1.5.1.3. Exit.
        - 1.5.2. Song selection.
        - 1.5.3. Pause Menu.
            - 1.5.3.1. Resume.
            - 1.5.3.2. Options.
            - 1.5.3.3. Exit.
    - 1.6. **Boss.**
        - 1.6.1. Boss mechanics.
2. **Design.**
    - 2.1. **Objects.**
        - 2.1.1. Drums and Drumstick hits.
    - 2.2. **Songs.**

1.5 Reach

This thesis about the development of a music rhythmic game has the potential to reach a broad audience, both within and outside the academic community.

Within the academic community, the thesis can contribute to game development, music theory, and human-computer interaction. It can provide insights into the design and implementation of rhythm-based gameplay mechanics and explore the design of new mechanics and features for the genre. This thesis will also dig into how music as a mechanic can enhance the gaming experience.

Outside the academic community, the thesis can reach game developers, musicians, and gamers themselves. Game developers can use the insights from the thesis to improve the design and development of rhythm-based games, creating more engaging and enjoyable experiences for players. Musicians can use the thesis to explore the intersection between music and technology, which can inspire someone to create new forms of artistic expression. Gamers can benefit from the thesis by understanding the games they play and how music contributes to their enjoyment. They also can ignite the spark in someone to start developing games.

2. State of the art / Theoretical framework / Contextualization / Market study

Before kicking off the development, it is necessary to explain what this thesis is about. The development of a musical video game refers to the process of creating a virtual reality video game where mechanics are focused on music and rhythm. Hence, the process of developing a virtual reality music video game involves creating a virtual gamified experience where users, through the use of a special peripheral, can play with their own physical movements in a game where the objective is to collect points or finish a level while following a beat, in this case, a song.

This type of video game, although it may seem like a recent development in the industry, actually has roots dating back to the 1970s. I am sure that all of us have at one point in our lives experienced or at least seen someone play a rhythmic game classic like Dance Dance Revolution in arcades, rocking to Guitar Hero, singing in a karaoke bar, or playing with classic fun electronic games like Simon.

Eventually, games transitioned from arcade consoles, where people danced atop machines pressing buttons with their feet, to being played at home with devices like guitars and drum kits in games like Rock Band. Recently, with the advent of virtual reality, not only do you feel like you are playing a real instrument, but you also experience the sensation of performing on stage for an entire stadium.

2.1 Video game development

The process of developing a video game involves combining a range of disciplines such as programming, game design, art and animation, audio engineering, quality assurance, and project management to create an electronic game that interacts with the user through external inputs and generates visual responses on a display. In the case of classic video games, the inputs are usually controllers or joysticks, and the display is a separate screen. Nowadays, this concept is changing rapidly with the appearance of mobile devices and virtual reality.

The development process typically starts with conceptualizing the game's basic elements, including the story, characters, gameplay mechanics, and visual style. Once the core design is in place, usually compiled in a Game Design Document (GDD), developers use specialized software tools and programming languages to build the game's framework and mechanics.

These specialized tools range from video game engines to very specific software for tasks like 3D modeling, animation, or audio mixing. During the development process, the development team works together to design and create game assets such as 3D models, animations, sound effects, and music. They also test the game repeatedly to identify and fix bugs and ensure the gameplay experience is smooth and enjoyable.

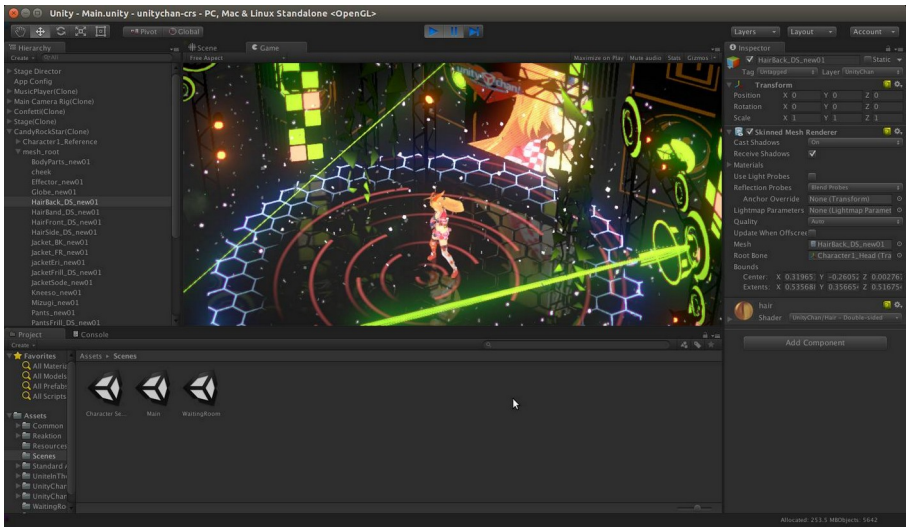
Because of this, video game development requires a diverse set of skills and expertise and involves collaboration between multiple teams and departments. It can take several months or years to complete a game, depending on its complexity and extent. To succeed in developing a video game, the individual or the team requires creativity, technical proficiency, attention to detail, and, if the objective is to release and sell the game, an understanding of the target audience and the competitive landscape of the gaming industry.

2.1.1 Video game Engines

Videogames can be developed through a lot of mediums and programming languages such as C and its variants, Python, or even Scratch. Luckily, the tedious process of creating an entire video game from scratch has changed and improved tenfold in modern times. The appearance of video game engines changed the development process altogether, consolidating most of the resources needed to create a video game into a single program.

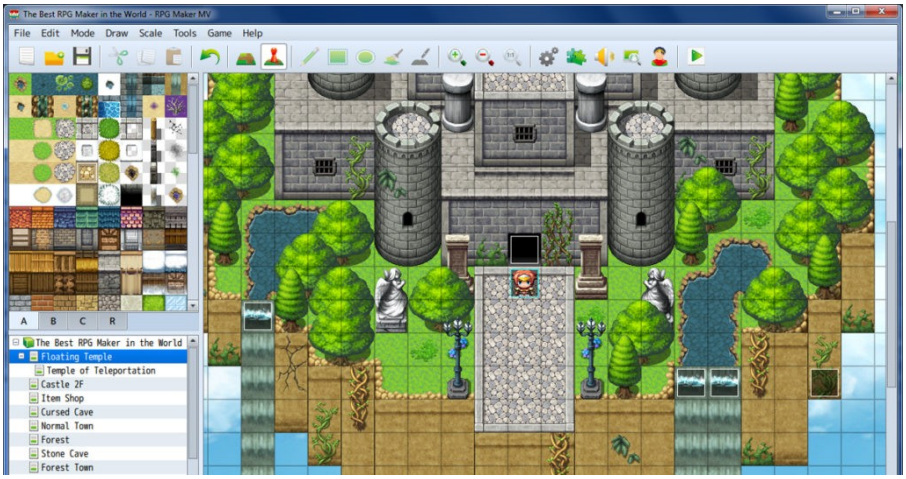
A video game engine is, by definition, a software framework designed to simplify the development of video games. It provides tools and libraries that enable developers to create games more efficiently. Game engines typically include features such as graphics rendering, physics simulation, audio processing, input handling, and artificial intelligence. They work by separating the core software components, like the three-dimensional graphics rendering systems, from the assets like the art, 3D models, game worlds, and the specific rules that create the user's experience.

The use of game engines has significant benefits for video game developers, such as a huge reduction in the time and resources needed for development or the simplification of porting games to different platforms. However, it is important to remember that video game engines require specialized knowledge and have their own learning curves, which require additional time and resources to master. Despite these challenges, video game engines continue to be a vital tool in the creation of successful video games, playing an increasingly important role in the growth and expansion of the gaming industry.



https://diocesanos.es/blogs/equipotic/2015/09/01/unity-editor-de-videojuegos-nativo-para-Linux/

https://docs.unrealengine.com/5.0/en-US/level-editor-in-unreal-engine/



https://hard-drive.net/hd/video-games/new-rpg-maker-update-quietly-deletes-program-one-week-after-install/

2.1.2 External Programs

VR game development also relies on external programs such as 3D modeling software, Oculus Client (PC to VR connection), digital audio workspaces such as Ableton Live and many other tools each specialized for a particular part of the process.

2.2 Virtual Reality Development

The etymology of the term “virtual reality” has its roots in the late 14th century where the Latin word *virtualis* is defined as "influencing by physical virtues or capabilities, effective with respect to inherent natural qualities". Other definitions appear throughout history, but a definition of virtual in the context of computers is first found in 1959 as: "not physically existing but made to appear by software".

However, the concept of a different reality that exists over the real one creating an apparent or virtual world where one can reside while not noticing their real surroundings has been around since Plato created “The Cave” allegory in his work, “The Republic”.

Therefore, by making use of each of the two words from Virtual Reality’s individual definition, we can come to a definition of it as an existence different from the real world which can be perceived as real or that has been manufactured to feel as real through bypassing sensory stimuli. Thus, any experience that changes the perception of reality for the user being this experience momentary or permanent can be considered as virtual reality.

VPL Research’s full body suit

In later years and with the apparition of various pieces of fiction that use the concept of virtual worlds but in a hyper technological setting, the general public's perception of virtual reality has become that of the creation of secondary fake realities through the use of technological devices like headsets (Neuromancer, William Gibson, 1984), holodecks (Star Trek: The Next Generation, 1987) or full body immersion (The Matrix, 1999). This new look into virtual reality has shaped the concept into what we consider it to be nowadays: a separate reality hosted by a machine that has a different set of rules and concepts that can be accessed by individuals through external devices.

The modern widespread concept of virtual reality is attributed to Jaron Lanier, a computer scientist who developed some of the first commercial grade virtual reality hardware in the late 80s. Under the name VPL Research, he designed and built a full body sensor suit and a virtual reality headset very similar to current commercial grade gaming headsets.

During the 90s and early 2000s, virtual reality had a period of slow progress, mainly attributable to the hardware needs of the platform, which outpaced the speed of microprocessor development. It was not until 2012 that virtual reality became mainstream with Oculus releasing the Rift, a commercial grade virtual reality headset specialized for gaming. Oculus was then acquired by Facebook in 2014 and in 2016 HTC entered the market with the Vive. In the following years, other companies revealed their VR offerings, Apple being the latest in 2023.

In the software realm, support for VR development has mostly relied on retrofitting solutions into already existing engines. This has allowed developers to smoothly transition most of their existing assets into VR, but compatibility issues are still commonplace and online documentation is rarely up to date. Furthermore, VR platforms have lacked standardization due to providers competing for market dominance, which makes the process even more difficult. This, along with many other external factors, has kept VR in the niche, unable to compete with other more accessible platforms.

2.3 Rhythm video games

While musical games are those that use music as a gameplay element, rhythm games are a subgenre of action games in which the main challenge to the player is their sense of rhythm. Some of the most famous examples of rhythm games are Dance Dance Revolution or Guitar Hero. The player engages with the game by taking actions in an exact timing usually given by a beat. Their performance is then measured and used as the score.

One of the first commonly known examples of electronic musical games is Simon, a handheld device in which players took turns to press complicated sequences of buttons. The title of the first rhythm based game is debated between Dance Aerobics (1987) and PaRappa the Rapper (1996). Although Dance Aerobics came nearly 10 years earlier, its classification as a rhythm game is not clear.

Up until 1998, the primary market for these games was Japan, with some niche users in Europe. The first game to achieve success outside of Japan was Dance Dance Revolution, which featured pressure sensitive pads the players could dance on.

Success of the genre within the Western market came in 2005, with the release of US made Guitar Hero. The genre grew but reached a point of stagnation in the early 2010s, where widespread personal computer devices made peripheral based games hard to sell.

Even so, the genre still has a significant audience and has had moderately successful releases since then.

2.4 Virtual Reality Musical/Rhythm video games

With the arrival of VR, rhythm based games have found new ways of engaging with players by taking full advantage of the possibilities the platform has to offer. The freedom of expression VR gives its players ties very well with many of the principles rhythm games are built upon. The genre has achieved some mainstream popularity again with top charting games such as Beat Saber.

Features such as the VR display, haptic controllers, body sensors and motion tracking offer a whole new way for developers to connect with their players, opening the door to full body experiences. Music and especially rhythm tie very well into these new technologies, taking them to the next level.

2.5 Videogame analysis

To gather information about the development of this thesis, I have analyzed 3 classic titles that belong to 3 distinct types of musical games which mechanics are completely different from one another. The main goal here is to extract all the bits of essential information that made these three titles great. To do so, I will separate each analysis into 3 sections: mechanics, immersion, and conclusions.

The three titles that I am going to analyze are Rhythm Doctor, a game based on a single click of a button, Guitar Hero 3, considered by most the epitome of the saga, and Beat Saber, which lately has become the first in rankings of VR musical games.

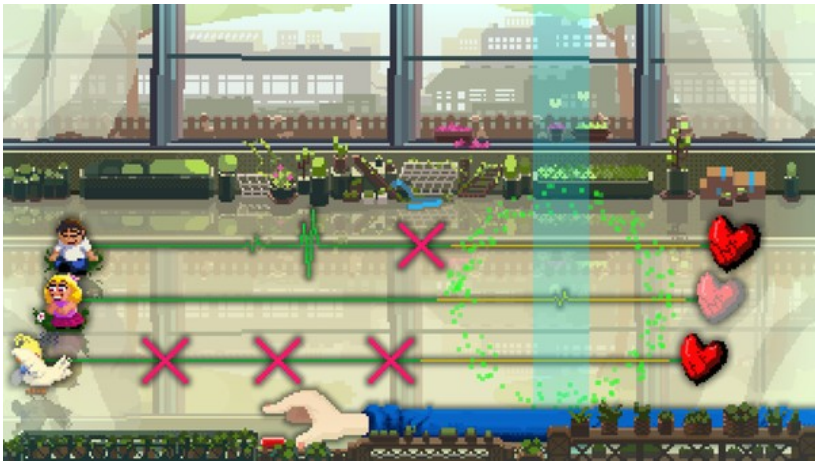
2.5.1 Rhythm Doctor

As described by its developers, the 7th Beat Games, Rhythm Doctor is a one-button rhythm game that aims to invisibly teach complex music and rhythm theory. The core mechanic is simple: slam your spacebar on the 7th beat to save patients in a hospital. Every level has a twist on that concept: polyrhythms, offbeats, hemiolas, and irregular

time signatures. It is a game designed to make you learn rhythm theory without realizing it. It was released on Steam on the 26th of February 2021.

2.5.1.1 Mechanics

In Rhythm Doctor, the main mechanic involves timing your actions to the beat of the music to press the button always on the seventh beat of a 4 by 4 compass, which is the most common one used in music. The game consists of a series of levels, each featuring a unique piece of music and a different set of challenges.



During gameplay, a series of icons will appear on the screen in time with the music. These icons represent the beats of the music and are synchronized with a visual cue in the background. As the icons reach a certain point on the screen that symbolizes the 7th beat, the player must press any key on the keyboard in time with the beat.

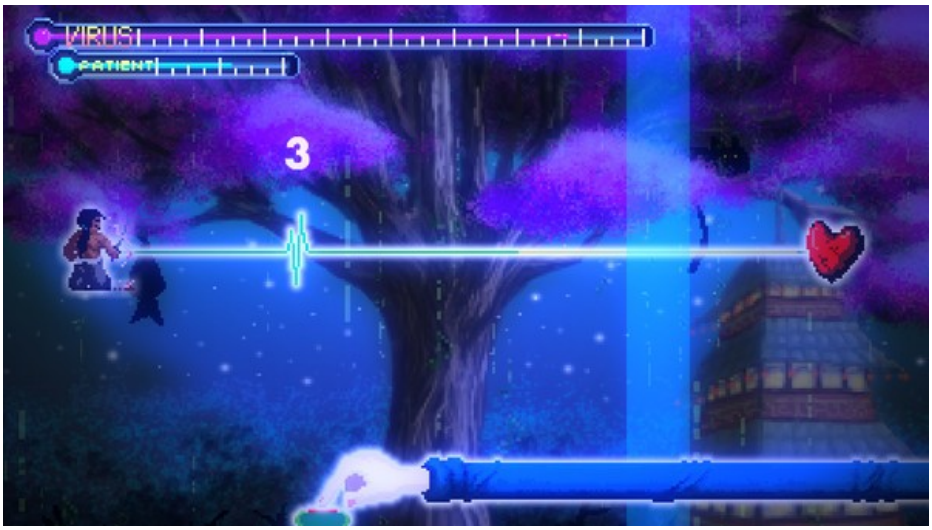
Rhythm Doctor also features a "life" system. If the player misses too many beats or fails to respond to them, they will lose a percentage of their life. Once all their life is lost, the player must restart the level. As the game progresses, levels become increasingly more challenging and require greater precision and accuracy in timing.

life representation in rhythm doctor

The game advances forward by levels and sections of levels. To move to the next level you must complete the previous one and so on. This was designed not only to create a progression through the game but also to teach the player about the gameplay and music altogether. The level structures are compiled in sets of 4 with a final level that acts as a boss. These sets of levels have a different mechanic that changes the gameplay without changing the input. You will still play with just one finger, but the

music and its representation will change between syncopation, polyrhythms, and irregular time signatures between others.

Each boss represents the most difficult level of each set and also acts as a test. It represents both the ending of the set and a challenge to beat to be able to face later more difficult challenges. Because they act as tests also they are the levels that squeeze to the maximum the quirks of each musical concept. The first boss represents keeping your rhythm by counting and not hearing the beat, the second represents polymorphism, and so on.



Boss level

Finally, rhythm doctor does not have an intricate ranking or a profound punctuation system to rank how you completed the game, but it does have a marking system that symbolizes how many fails you made during one level. it goes from F if you lose all the way up to S meaning that you completed the level without fail. If you complete the level with an S, you unlock new versions of the level as you progress.

2.5.1.2 Immersion

The game follows a group of doctors that have developed a new technology to allow musical treatment from remote locations and you are one of these remote doctors. You can not communicate with them directly, but you express your help by completing the levels, which symbolize each patient from the hospital and their treatment.

Because this game is really easy to get into due to the single-click mechanic and the refined design that has a really good difficulty curve it is really easy to just start the game, plug in your headphones and enter every one of the levels completely focused.

2.5.1.3 Conclusions

By simplifying all the mechanics to just one click, and using it as a modal switch between all the different musical concepts in the game, you create a fun and simple experience for the users that hides a complex design that is even capable of teaching players about music theory.

This could help my design by allowing me to rethink classical mechanics into more simple modes while I increase the complexity of the concepts behind them without changing the gameplay. This would help both my project's gameplay and immersion.

2.5.2 Guitar Hero 3

Guitar Hero III: Legends of Rock is a music rhythm game developed by Neversoft and published by Activision. It was released in 2007 and is the third main installment in the Guitar Hero series and the one with more renown. In the game, players use a guitar-shaped game controller to simulate playing lead or bass guitar parts in various rock songs by hitting colored notes on the controller in time with scrolling notes on the screen. This was later expanded to drums and vocals.

Guitar Hero III features a single-player campaign mode where players progress through different levels and unlock new songs and characters. It also includes a cooperative mode where two players can play together on the same console and a competitive mode where players can compete against each other. The game features a tracklist of over 70 songs, including classic rock tracks and new hits from various genres.

2.5.2.1 Mechanics

To be able to play Guitar Hero 3 the players will need to acquire a dedicated guitar-shaped controller for the corresponding console (PS2, PS3, Xbox360, Wii, PC, or Mac). This controller allows the player to integrate with the specifics mechanics of the game while giving an extra level of immersion. This controller has 5 colored buttons that represent frets in a guitar and a slider bar that turns 45 degrees over an axis sideways that imitates the plucking of a guitar string

Guitar Hero 3 features multiple gameplay modes, including Career Mode, Quickplay, and Multiplayer, that offer different challenges and gameplay experiences. We will be focusing on the single-player because the other modes are just extensions of it.

The single-player mode starts with a song selector separated by gigs. Each gig represents a concert in a specific venue and a difficulty level. The last concert always represents a “boss” but in this case, it feels just like an extra cool song to play and it's presented more like the finisher of the concert. The main gameplay loop is to play each song in order until you complete the game and after or while doing so replay the player’s favorite songs as many times as they like while trying to achieve higher scores. Depending on the number of notes that you missed in the song you will get a valoration out of five stars, being 5 stars a perfect level.

As I mentioned before, the video game's song selection menu is separated between "gigs" or concerts. Each gig has a small number of songs. The game's system works with 2 parallel difficulty progressions. The difficulty rises with each song until you arrive at the last concert and complete the game, but on top of this, each song has four levels of difficulty: Easy, medium, hard, and expert. On the easy level, only the first three colors appear, red and yellow and the amount of notes that appear is small. In medium an extra color is added, the blue notes, and the amount of notes also increases. On hard, the complete 5 colors are used and the amount of notes is moderate. Last but not least in Expert the amount of colors is the same as in Hard but the number of notes is pushed to the limit.

When entering a song, you will be faced with what is called a "fret highway", which consists of the back of the guitar fretboard extending to infinity with 5 strings from where colored dots appear at the rhythm of the song representing the notes on the guitar. The dots move toward the player, and the player must hit the corresponding buttons on the guitar controller when the dots reach a certain point. On the left, you can see the song score and multiplier, on the right you can see the star meter and the rock meter, and on the middle you can see the fret highway.

The second type of basic note that appears is the sustain note. It is represented as a long strip of the color of the note that starts with the regular dot shape. You will need to press the button for the whole duration of the strip. If you lift your finger before you will miss the note, break your combo and start getting points for the remaining strip.

There are no life meters in the game because they use a concept called “Rock meter” that represents the global feeling of the public of each gig. At the end of the day, it measures how well you are playing by counting the number of notes you have hit and missed. if you hit more notes than you miss you can keep playing, if you miss more than you hit you lose. The meter is represented by a meter that changes between 4 states: Green when you are really safe, yellow represents that you are missing some notes, Red means you are close to failing, and flashing red means you are running through the edge and if you miss a couple more you will lose.

On top of the normal notes, there are also the star notes which the players can hit in order from start to finish and gain one charge of star power to. When you obtain three charges you will be able to power up your guitar by shaking it or pressing a button on it and duplicate your multiplier in the game. This power-up will fade with time. You can have a maximum of six charges, being the longest time you can activate the power up for and being three charges the shortest.

The guitar also has an extra mechanic in the form of a makeshift tremolo lever or whammy bar that changes the sustain note's pitch. Although at first, it may seem that you can get more points through the use of this functionality, it has been proven wrong by the community. But there is still another function for this mechanic apart from helping the immersion and that is if you use it in the middle of a star phase you are able to charge your star meter faster. You can also fill up the rock meter faster through the use of the whammy bar which allows you to survive dire situations in songs if you know how to use it correctly.

2.5.2.2 Immersion

Compared to Rhythm Doctor, the guitar hero I would say has a couple of extra levels regarding the immersion that is able to achieve. This is attained through three main points: The controller, the environment, and the song repertoire.

Let's start with the controller, which is shaped and acts as a guitar both in functionality but also in feel in the case of the Wii version uses the vibration and the internal speaker of the controller to recreate the sensation of a real guitar. As of today the concept of an external peripheric used specifically to play a particular game is kind of outdated, but the Guitar hero's guitar in particular is one of the best if not the best peripherals created for a video game in history after following its trajectory.

It is easy to learn because you can relate to the real object, it's light which makes it easy to use for extended periods of time, it's flashy, cool, and fun and it lasted through 7 big titles and all the band-centered and expansion games of the saga. And last but not least it makes the player feel like a real rockstar.

The in-game environments represent musical venues like festivals, bars, and crazy places like hell that fit perfectly the rock and metal theme of the videogame, and the characters that you can choose from to play with personify the player inside the experience.

Finally, the video game's setlist is wide and entertaining. It is composed of songs from real and famous rock groups from the 70s, 80s, and 90s like Black Sabbath, AC DC, KISS, The Killers, and many more. This completes the overall flavor of the game and gives the last touches to the game completely hooking the players.

2.5.2.3 Conclusions

Guitar Hero 3 is the maximum exponent of this genre in consoles and has represented it through the 2000s attracting hundreds of thousands of players to rhythm games. Its fun and easygoing attitude mixed with the rock aesthetics and the innovative concept they came up with for the controllers makes this game, in my opinion, my go-to in terms of personal bests.

Mechanics-wise the core concepts that we can extract are the difficulty levels for each level, allowing players to select the experience that fits them, the concept of different types of notes that represent different musical concepts, and the star power mechanic, that rewards the player after completing a perfect section and allows them to earn more points while playing through a flashy and cool section after activating the power up which adds to the immersion.

Although I like the idea of an extra peripheral like the guitar, in the case of a virtual reality game that focuses on being played only with the basic headsets it represents a problem. But there are concepts like haptic feedback that can be generated through the vibration of the controller that can be used to represent certain things in the game.

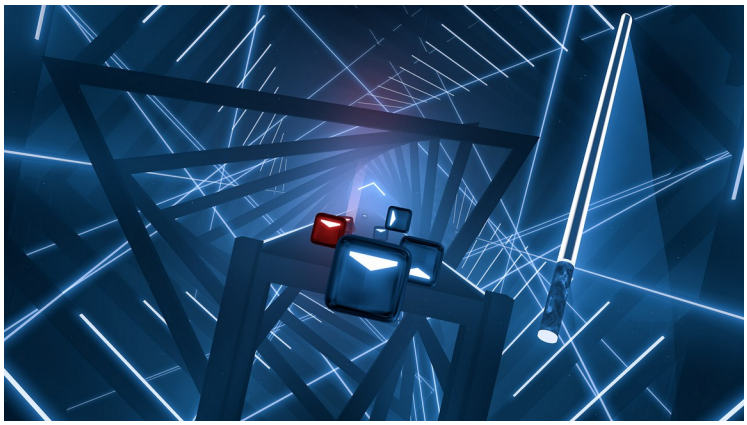
When creating a rhythm game, Guitar Hero 3 is an example of a well-tough and perfectly executed product for the general public.

2.5.3 Beat Saber

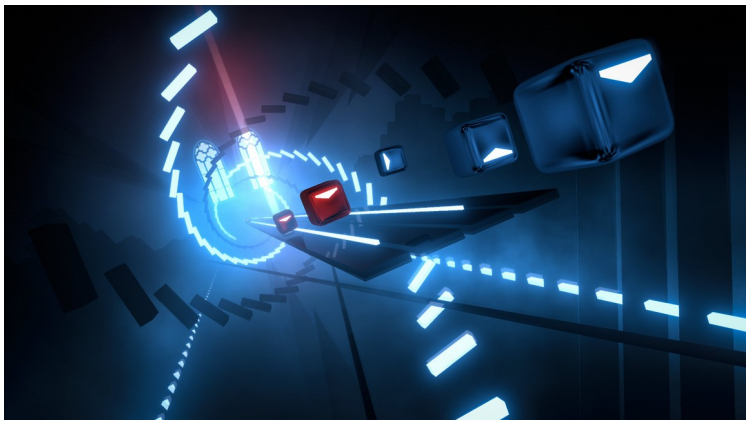
Beat Saber is a virtual reality game developed by Beat Games that was released as an early access in 2018 where you use lightsabers to slice blocks that fly toward you, all while moving to the beat of the music. The game has different types of music and difficulty levels, and you can even create your own levels using the editor. The game has received a lot of praise for its unique gameplay, immersive experience, and catchy music. It's the favored choice for rhythm game players in virtual reality.

2.5.3.1 Mechanics

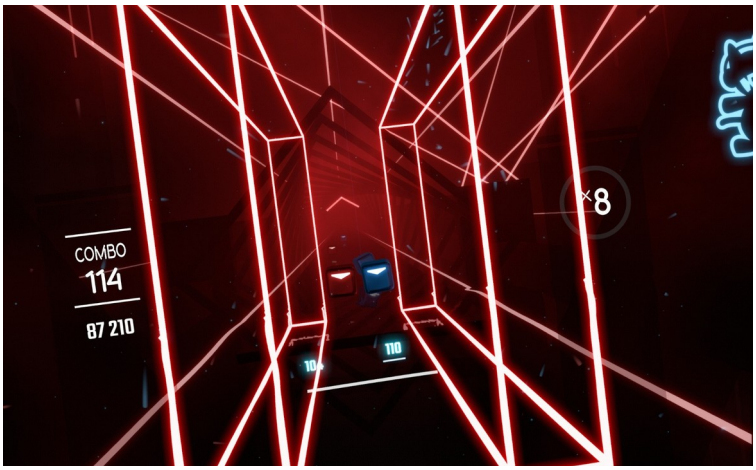
The main gameplay loop in Beat Saber is based on cutting flying blocks at the rhythm of music with lightsabers. You have a different colored saber in each hand, blue on the right one and red on the left one. The flying blocks come straight at you with an arrow drawn in the front face which symbolizes the direction that you have to cut it from or a circle which means you can cut it in any direction.



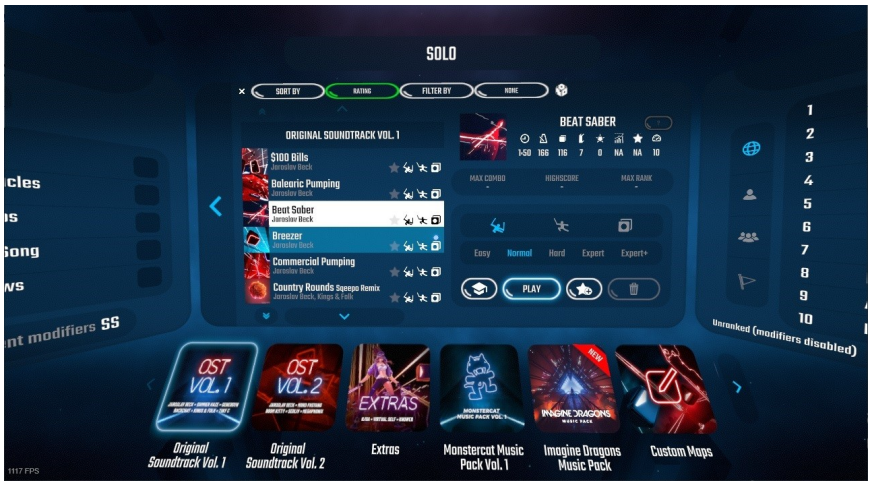
As in Guitar Hero, when you cut a block you earn points and you accumulate a multiplier that resets every time you miss one block. the more blocks you cut in a row without missing the more multiplier you get and the more points you earn. There is no progression so you basically have different levels that you can play infinite amounts of times to try and get a higher score each time.



On top of cutting blocks, you will have to evade spiked balls and walls that will come from the same direction as the blocks which will lower your high score and make you lose. You will have to do this while trying not to move around too much because of the environment but it is easy once you get used to it. If you miss a lot of hits in a row you will lose and the menu screen will appear in order for you to retry, select another song, or exit the game.



The songs available for you are mostly created for the game in particular by their development team's composer or by freelancers, but as time went past extra DLCs appeared with songs from Lady Gaga, Linkin Park, and BTS between other big musical groups.



2.5.3.2 Immersion

Apart from the other two titles that have been discussed previously in this Thesis, Beat Saber does not have any type of narrative or progress in a story, so there is no effort applied to making the player follow an argument. This eliminates the possibility of immersion through personal identification with the main character or an actual representation of a virtual user in-game as in rhythm doctor.

But in the case of this game, these types of immersion are not needed in any way. It presents an environment where you have to follow a small set of rules and lets the virtual reality headset complete the circle. The strength of virtual reality is that the feeling of embodiment that normally has to be created through a narrative or through the use of external peripherals that act as a medium is bypassed completely.

A virtual reality headset can indeed be categorized as a peripheral, but the difference here is the relationship of the user with this headset. Through the lenses and their disposition inside the VR headset, the user's brain interprets the visual representation as the real environment that is in front of them. Indeed, the environment is not a one-to-one representation of reality because technology has not yet arrived at the point of being able to replicate completely real environments, but it's good enough to affect the user's perception.

After the user's first contact with the game, the second step is to represent its movements, and that is when the controllers come in. They are designed to specifically function by imitating human interactions with their surroundings through the use of the hands in the form of grabs, pokes, hits, triggers, vibrations... When the user sees the environment as reality, and it's embodied through the controllers, the game just feels like a cool activity to be doing in this new reality that you are in and it could be seen as playing in an arcade or a simulator.

On top of that, the visuals of the game with all the lighting that goes over the user's head, the sound, and the use of two lightsabers which is one of the selling points of the game make the user forget that it is playing through a display.

2.5.3.3 Conclusions

This game goes straight to the point, you boot it up and start to play right away without any type of follow-up with characters, or any progression both in narrative or in the form of a character's development. This is one of its main strengths and also one of the general strengths of a lot of rhythm games, which is that the gross amount of players want to select their favorite songs and play them right away without any context apart from the environmental one.

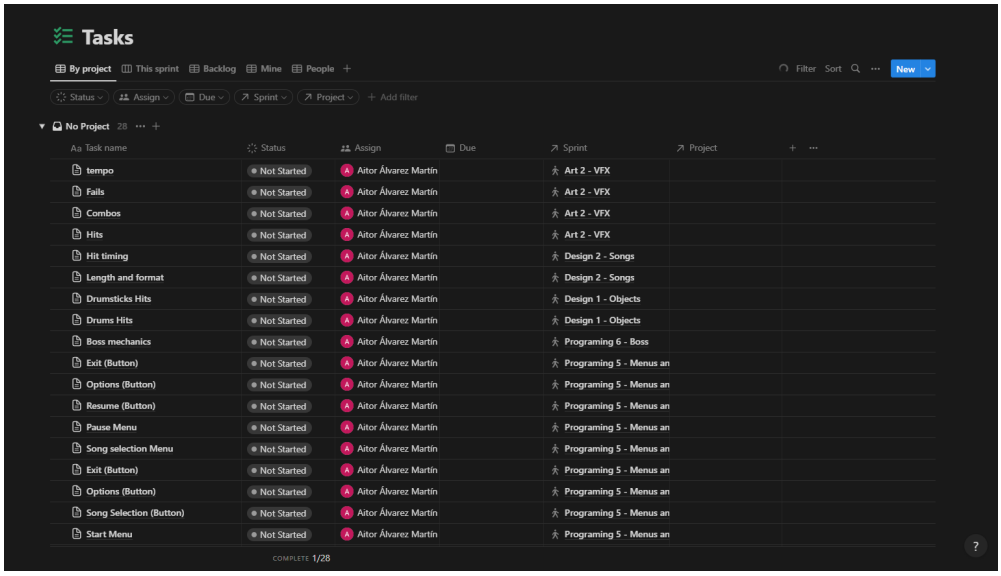
Both rhythm doctor and Guitar Hero do follow this concept also because each level is independent and can be replayed indefinitely, but they have a line of narrative even if it is almost non-existent which is the case of guitar hero that acts as a bit of extra flavor.

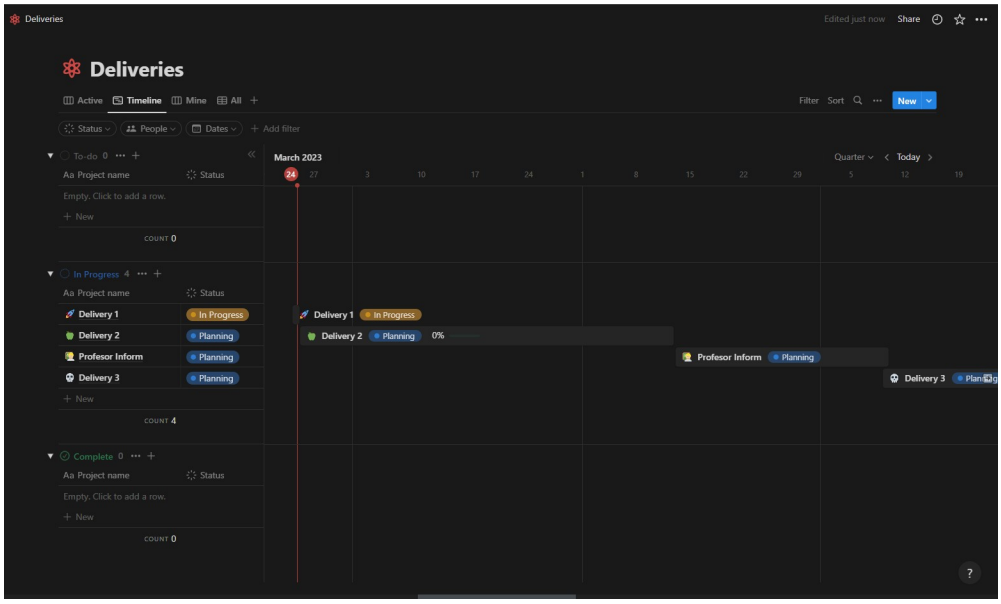
What can be taken from Beat Saber is clean, great, and well-designed gameplay that has an extra layer of UI on top to be able to choose what you want to do in-game. That sturdy core gameplay is what needs to be designed as a basis for any good rhythm-based video game and that clean immersion is a must also. On top of this, any type of concept can be built to give the game flavor and uniqueness.

3. Gestió del projecte

To keep track of my thesis I have created a Notion workplace. There I will record all the progress that I have made and the remaining work objectives that I have to complete in order to finish the project.

In order to do so I have separated the thesis by deliveries, each one with its respective sprints and each sprint with the respective tasks attached to them. This allows me to keep establishing a calendar for the development of the game and not to establish a start and end point for it.





Each sprint represents a week, having a total of sixteen sprints from the first delivery until the final presentation. These sprints are separated between programming, design, art and finally polishing and publishing.

By following this methodology, I can apply the principles of agile to manage my project, breaking down complex goals into manageable tasks, prioritizing my work, and adapting to changes as I go.

Additionally, I will be using Github to handle the unity project on the cloud and save my progress on both computers, Google Drive to save all the assets that I gather outside the project, and Notion to keep track of my database of documents, links, and information.

3.1. SWOT

These are the strengths and weaknesses of the thesis and its development

	Positives	Negatives
Internal Origin	**Strengths** All musical games have a similar core, so there are a lot of finished examples that will aid the development. Personal experience with this type of game and the will to create a great experience.	**Weaknesses** I have Little experience in handling projects of these dimensions by myself. There is little to no documentation about the development of musical titles, so I will have to face the development based on VR games and musical games separately.

	Opportunities	Threats
External Origin	The creation of a good video game that both entertains and can appear in my resume. Improve my programming and designing skills through development.	The time limit can represent a problem for the completion of all the different parts of the development. Problems and bugs can appear during the development and represent a hindrance to the process.

3.2. Risks and contingency plan

The possible risks identified in this project, and their corresponding solutions are the following, ordered from minor to major.

Risc	Solució
Bugs and small delays	Use the time dedicated to that section of the project to search for information about the problem and fix it
Hardware Failure	Keep a backup on the cloud (github, google drive) to have always access to the project from both my computer in case one or both fail.
Missing Assets, Songs, or FX	Dedicate some time to create these assets myself or find them on the internet and dedicate some of the budget to acquire them.
Failure to create a good immersive video game	Narrow the scope to create something that prioritizes quality over quantity
Failure to arrive at the final delivery with a complete thesis	Deliver what I have achieved explaining the reasons why I did not arrive at the final objective

3.3. Initial costs analysis

For the video game that I will be developing there are not any real initial costs. The programs that I will be using are all free, they have a student license (Unity) or they

have a completely free release (Blender). On top of this, I will be using my equipment which consists of my desktop computer, laptop, and Oculus Quest 2 headset.

Furthermore, every song and sound that I may need will be first selected from a royalty-free library or can be created by myself. Also, all the 3d models have been provided by the company, and in the case that I would need any new specific model I can also create it myself.

Nevertheless, it would be beneficial to have a small sum of money dedicated to any inconvenience that could appear during the development like any problem I could have with my equipment or if I need a specific song, FX, or 3D model for the final product.

I will invest an initial sum of 300€ for any inconvenience and needs I could have in the development. This will be separated into 150€ for equipment failures, 75 € for any license that I may need and the remaining 75€ for any model or audio that I could need.

Acquisitions	Investment
Equipment reparations	150€
Needed licenses	75€
Extra Assets	75€
Total	**300.00€**

4. Methodology

For the development of this project I will separate the process into several stages. pre production, production, and postproduction. Each stage is critical to the success of the game, and they require careful planning, coordination, and execution to ensure that the final product is engaging, fun, and bug-free.

4.1 Pre Production

is the initial stage of game development, and it will be primarily focused on programming and planning. This is where the game's core concepts will be developed and established. I will work on defining the game mechanics, and the programming of each section, as well as the prototyping and the final version of the skeleton of the game. Preproduction is essential to get a clear understanding of what the game will look like and what it will be able to do.

The first weeks will be centered on programming and building all the different basic sections of the video game such as the drumsticks, the drums, the menus, and the mechanics that will be applied to the mechanics.

This section will be represented in notion's sprints as programming. The first six sprints will be dedicated to preproduction and will set the bases for the project.

4.2 Production

Will be the next stage of game development, and it is where the design and art phases occur. During production, I will implement the 3D models, music, sound effects, and final levels. This is where the characters, backgrounds, and environments come to life.

The design period that is formed by 3 sprints will be where I will implement and test the mechanics of the game over the skeleton created in pre production. Because of the nature of this project, the game feels and gameplay needs to feel not only fun and smooth but need to make the player enter the "zone". The flow of the game needs to mix the gameplay and music for the player to have a feeling of immersion and this will be the period where it will be implemented.

The Art period follows right after the design period and is where all the final assets that hadn't been added through the preproduction and production will be implemented. All the menus, the VFX, the final audio clips, the remaining 3d models that need to appear on the scene, and the final visual polishing of the game will be added in this period.

To finish this section, the last sprint that finalizes it is the one called general polish and will be the last week of production. This period will be dedicated to the final touches for the game to be as good as it can be. The last bugs that appear and the last design and art changes will be applied here.

During this whole process, I will be testing the game with two different groups. One is formed by a small group of external testers and another one is formed by game developers. Because of the short amount of time that there is, the information that I need from the testers needs to be complete and fast and that is the reason why I choose this group.

4.3 Post Production

Is the final stage of game development and its release. Once the game has been thoroughly tested and is ready for release, it will be made available to the public. In order to have a decent release, I will have to upload the game to a digital platform like Steam or Itch.io and create somewhat of a campaign that follows the development of the product.

5. Project development

5.1 Pre Production

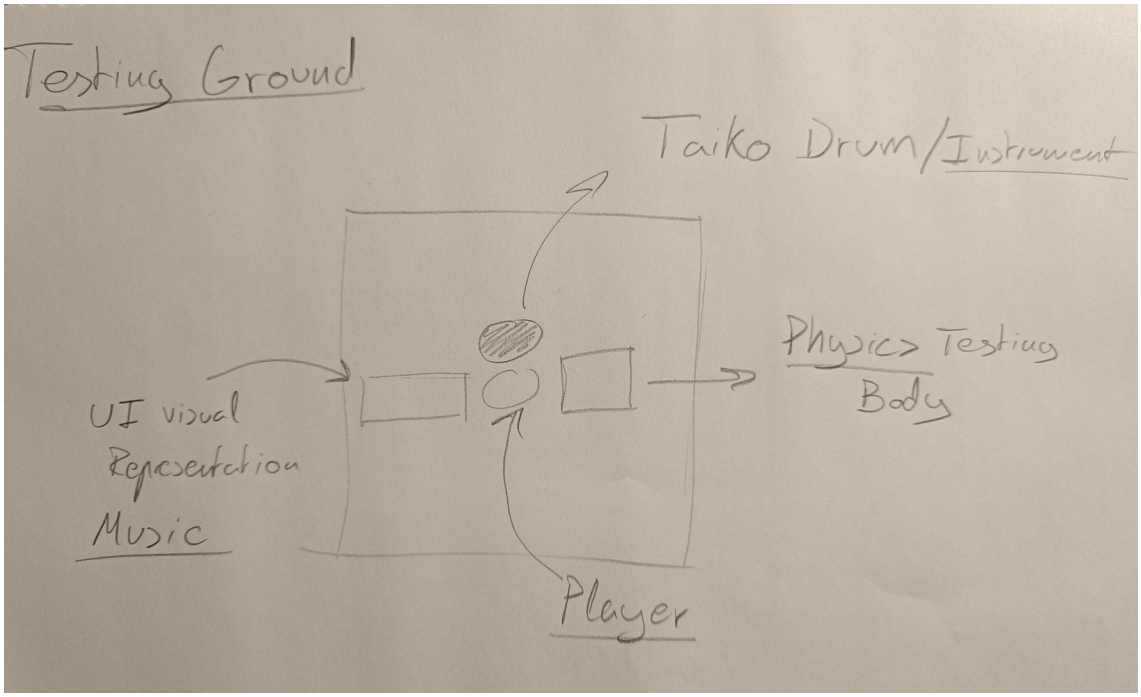
5.1.1 Research

As I mentioned before, musical videogames are one of my personal preferences. So the previous research was made to state a basis on where to start developing more than to influence the actual design. This research acted more as a list of what to do and what to avoid during the process.

What I wanted to develop was a classic musical title based on games like the previously mentioned guitar hero where you play through following a song, and combining it with a virtual reality experience that gives the player an extra level of freedom. This environment will allow the player to get as close as possible to the real deal.

Once I established the overall design for the videogame I separated the development into two main areas, VR development and Musical/Audio development. These two sections represent the two core pillars, and they both needed to be equally important for the project to succeed.

And to kick-start the project I established that I needed a small testing ground that would act as the basic scenario for this type of game, a small ground with the player, the instrument and the UI that represents the rhythm. This would be later replicated in the various scenes as the cardinal stone of the gameplay.



5.1.2 Design

I started by defining what type of game I wanted to develop to fit the subject of this thesis. The main two routes that I could follow were to develop an innovative game that aimed to revolutionize the industry, or I could design a game that represented the core aspects of a classical musical video game and dig into all the aspects of the subject.

In the end I chose the second option, while deciding to try and make it somewhat appealing and innovative visually, the mechanics and dynamics of the game are really similar to the previously mentioned titles. This does not mean that this project aims to copy a musical video game title, it means that I think these similar points in my opinion are the basis of what these kinds of titles are built upon.

The first thing that I designed came to my mind as the first thing that this type of games use as their main stage in, and that is to follow the rhythm of a song or musical composition to earn points in order to beat a level or a leaderboard. This level is represented with the song or composition as the level name and theme.

Rock you like a hurricane, tier 5, song 2, guitar hero 3, 2007.

In terms of the art of the game and the visual design, the tutor in charge of this thesis Lasse Loepfe and the teacher Marc Ripoll were working on a previous VR musical project that was on hold, and I was lucky enough to be allowed to use these 3D models for the project. The aesthetics of these assets were based in a classical japanese setting, with classical topics like temples, buddhist monks, the moon, and most importantly, japanese traditional instruments.

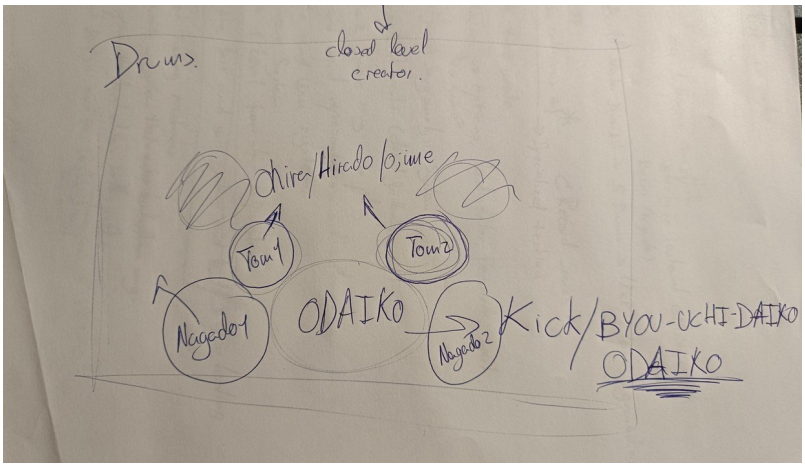
The previous aesthetics were centered around the use of a type of japanese drum called the Taiko, which comes in different forms and sounds. The selected one was the

most common one, the Nagado-daiko. The Nagado-daiko is a simple round drum with two sides normally used with a stand.

I'll also be using the stand in some cases, the Bachi which are the drumsticks for this type of drums, and the models for some other types of drums which I will explain later.

With these two concepts, the japanese setting and the drums, I thought about a game loop where you would select a song, you would load this song and then enter directly to the temple with a couple of drums in front of you that would represent your stage. Over these drums they would appear three note highways, like the ones in guitar hero, representing the three drums and their respectives parts of the song to come. You would hit the drums at the same time as the notes and get points for your skill at hitting as close to the tempo as possible.

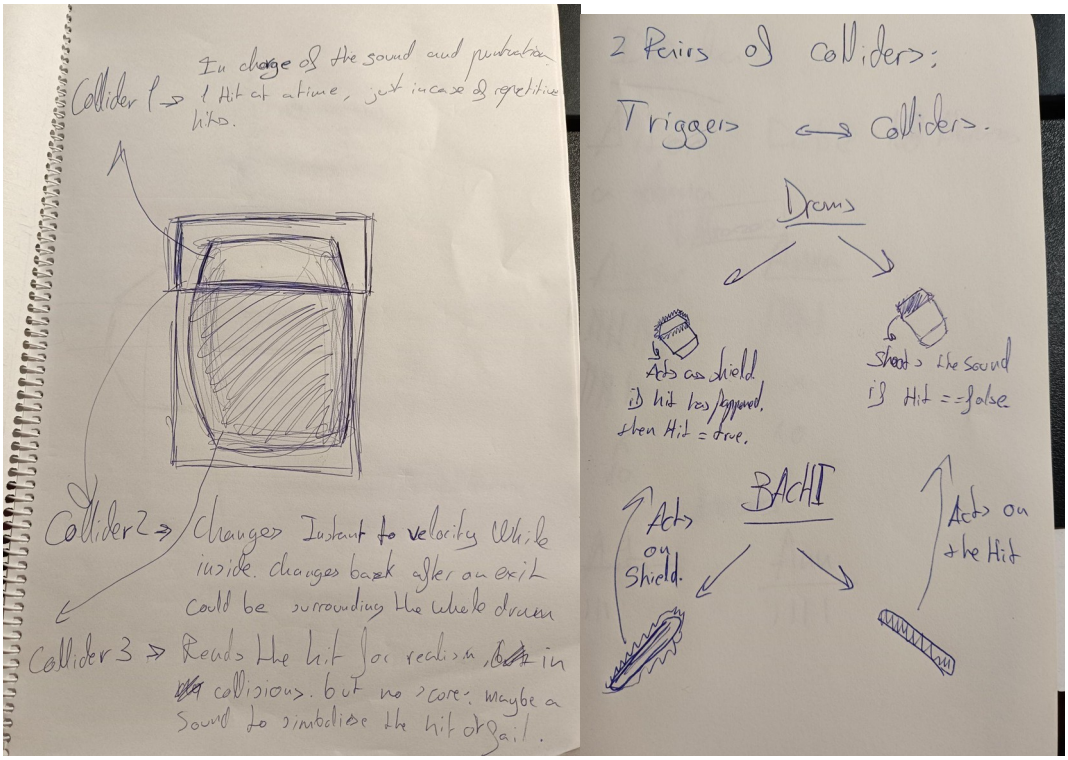
With this in mind, at first I thought about creating a drum kit, but this idea got discarded quickly because of the added complexity of the songs and the gameplay even though it would be a really cool idea for hardcore players.



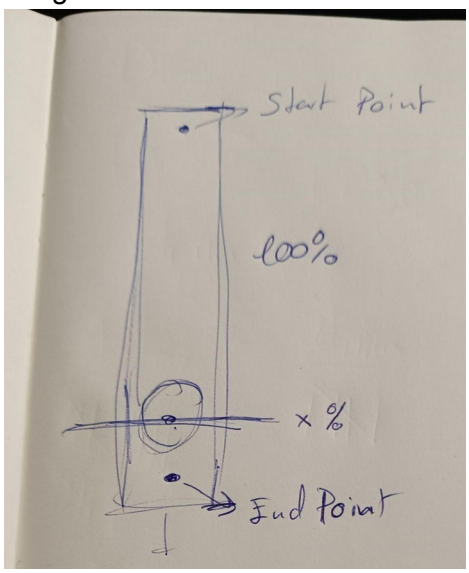
Drumkit style early design

After each song, a screen with the information about how good you did would appear and the option to try again or to go back to the menu would be given to the player. This will be the main game loop of the project.

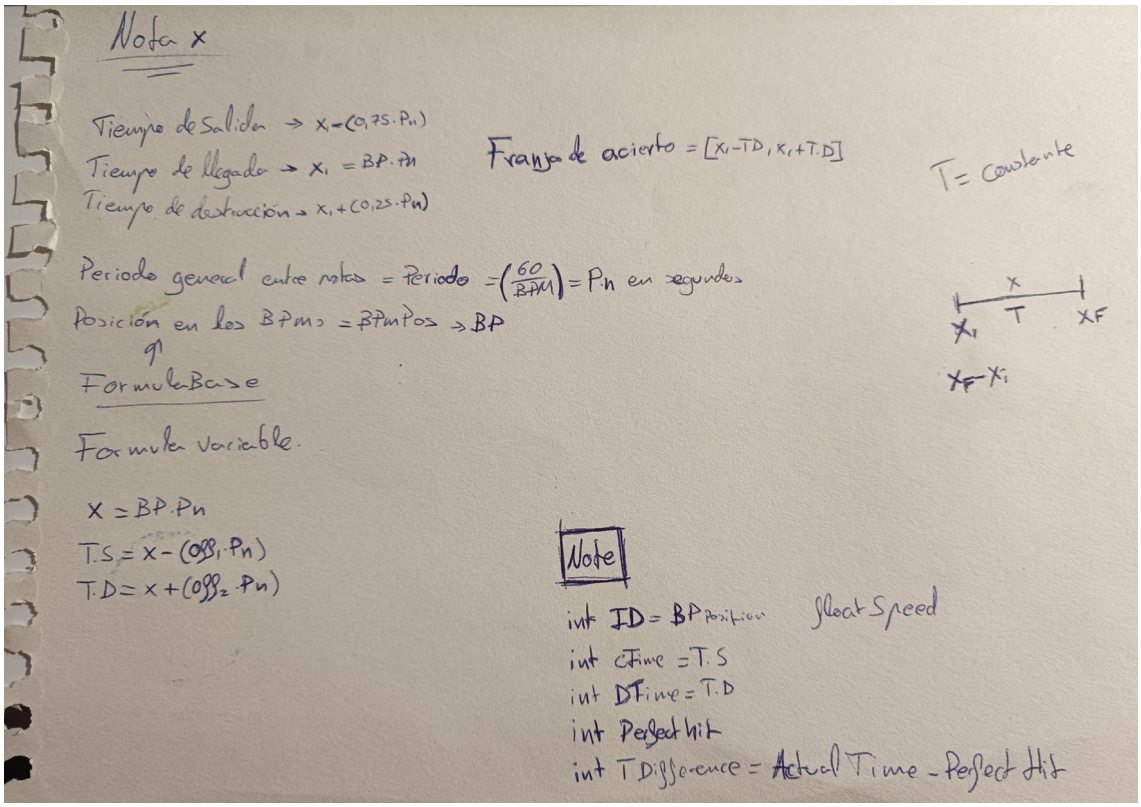
For the design of each part I drew a small scheme of how I wanted to implement it, starting with the drum. It would have two separated sections, one for the sound collision physics and another one just for the regular collisions. Inside the sound collision section, there would be 2 colliders, one that reads when the bachi has closed in to hit the target and act as a trigger, and another one to act as the physical part, because the XR toolkit system has some quirks and in some previous tests the bachies sounded repeatedly like a machine gun because of the repetition of the onCollision() Unity function call.



After this, I thought about the UI and designed the note highway and the note's representation. My idea was to make a note highway that would carry just one note or specific type or note for each drum so they would act independently. The looks of this note highway was the following:



After thinking about this highway I realized that I needed some type of formula to measure how to count when a note is close to the hitting point so we could calculate the score for each hit.

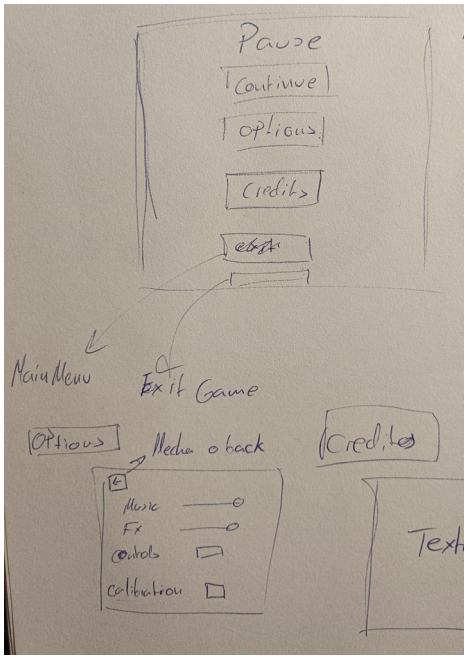


This formula will take into account the BPM, a concept in music that calculates the amount of beats in a minute. Being that a beat is the basic unit of time in a musical piece when looking at the tempo and the length of it. A song that has a BPM of 120 would have 2 beats per second. and the formula done the other way around shows us that each beat appears every 0.5 seconds.

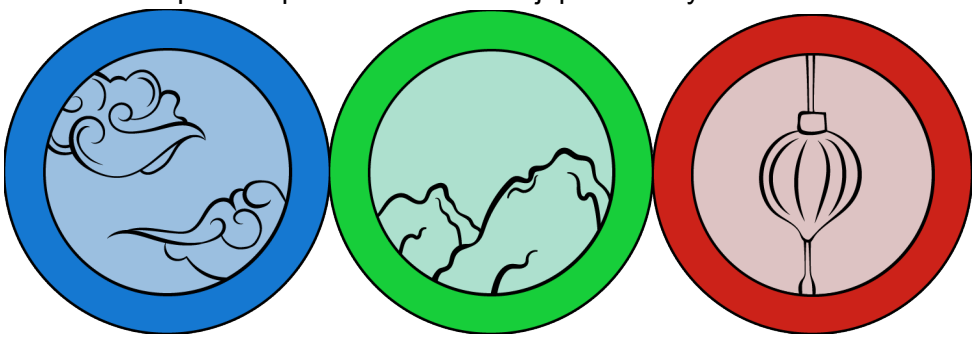
With this, if you know the distance between two points, being this two points vector3 in Unity and the start position and end position of the highway you can make it as big as you want because the period between beats will be always the same for the song and you can calculate the speed of each note by dividing this distance by this time period.

I also realized that I needed variables to calculate if the hit was before or after the actual beat, and for this I established offset 1 and offset 2, that measure half a beat in front and after each note and if the beat is inside this time space then it would count, and the bigger the time difference, the less points teh player would get. I made it this way to be able to calculate all the results at the end and to eliminate an in-game score marker to make it as immersive as possible.

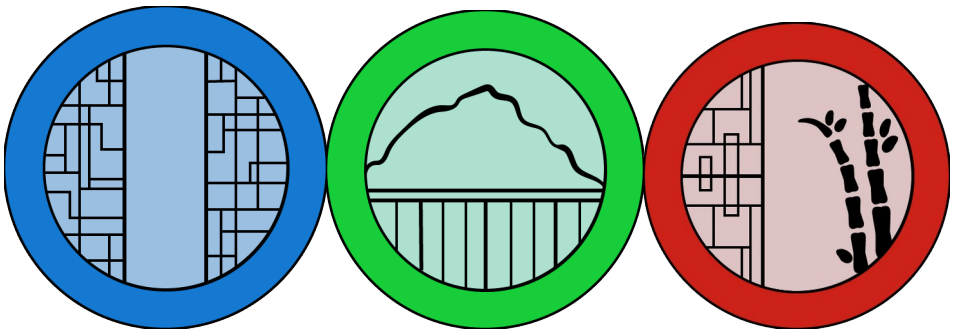
I also created a small menu scheme for the UI, nothing special but still necessary.



A really good friend of mine helped me with the design of the notes, so I had two different types to choose from, one more based on nature, and another ones more based on classic Japanese places. Both with a japanese style.



Natural type of notes



Japanese style notes.

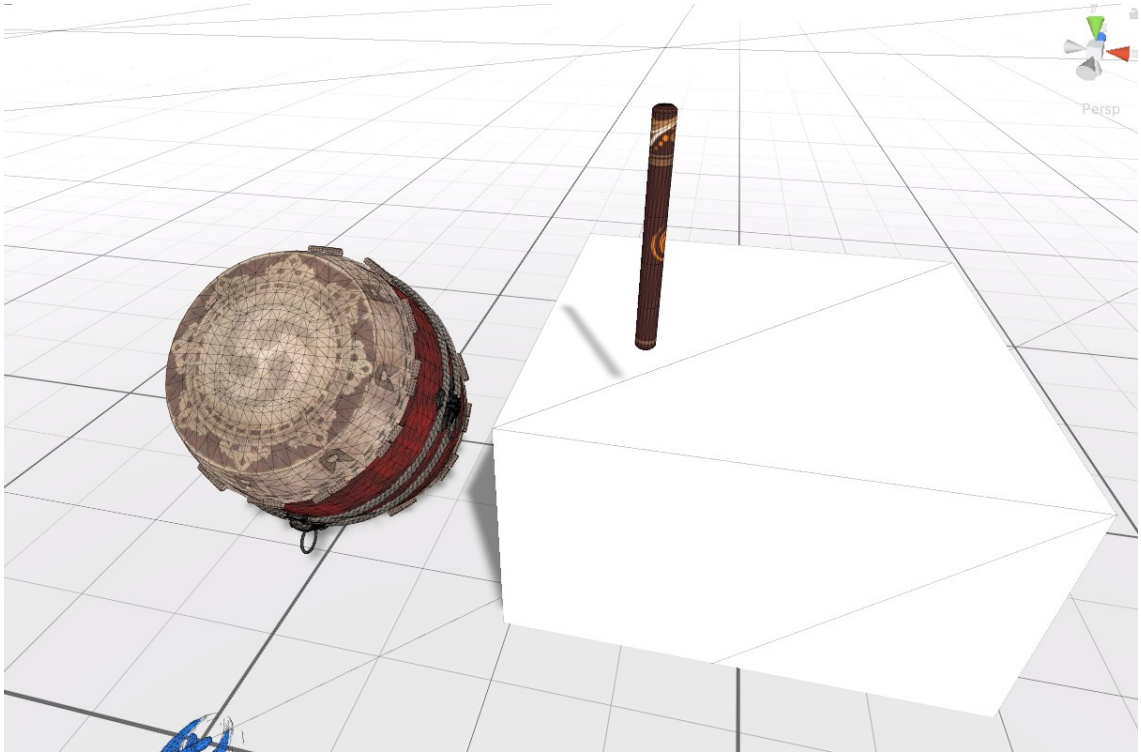
5.2 Production

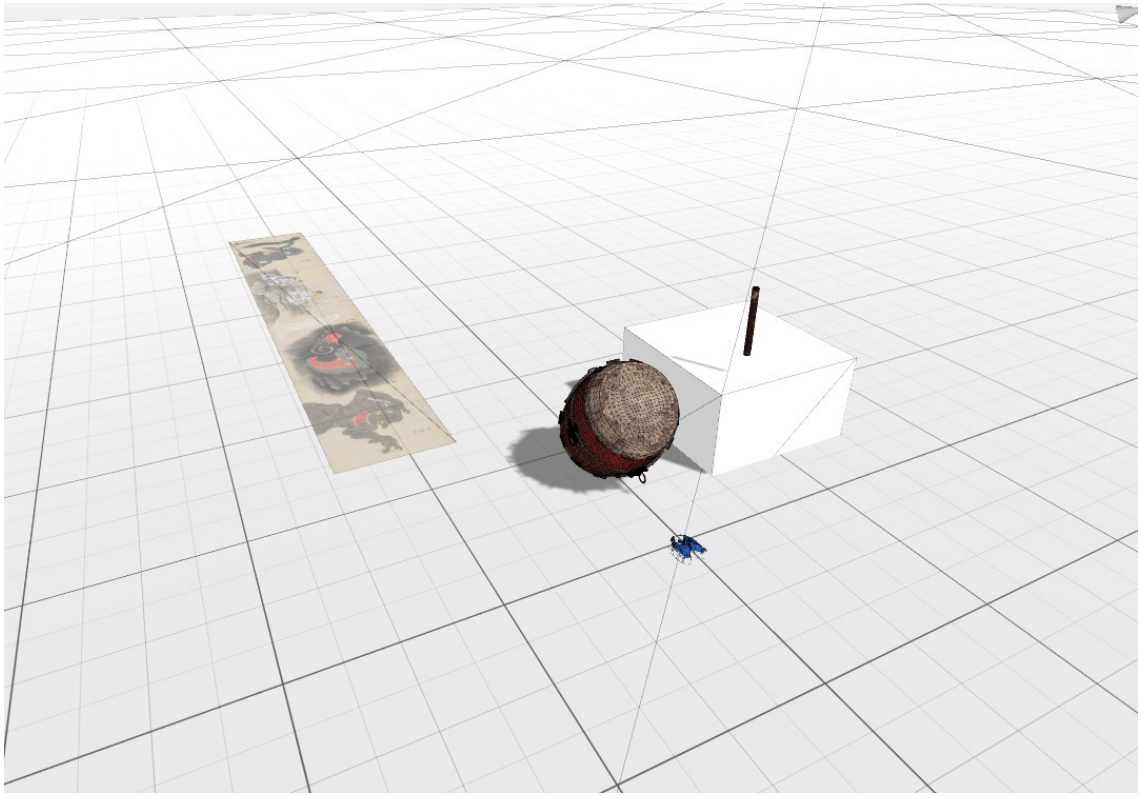
I started by creating a sample scene with the scheme that I have shown in the research. A big white space with a drum, a box and the XR origin which is the player's basic VR functionality unit.

First of all I focused on the physics of the bachi and the sound shot system that fires the sound with each hit. To do this, I separated the bachi physics into the bachi and the drums physics into the drum to keep them as two sides of the system. The bachi has two colliders, one is a trigger and the other one is a normal collider, one acting as the real bachi's body and the other one triggers the drum's audio event. The basic physics system calculates its position and does not allow the stick to go through any object, having a recoil so it looks like a real life object.

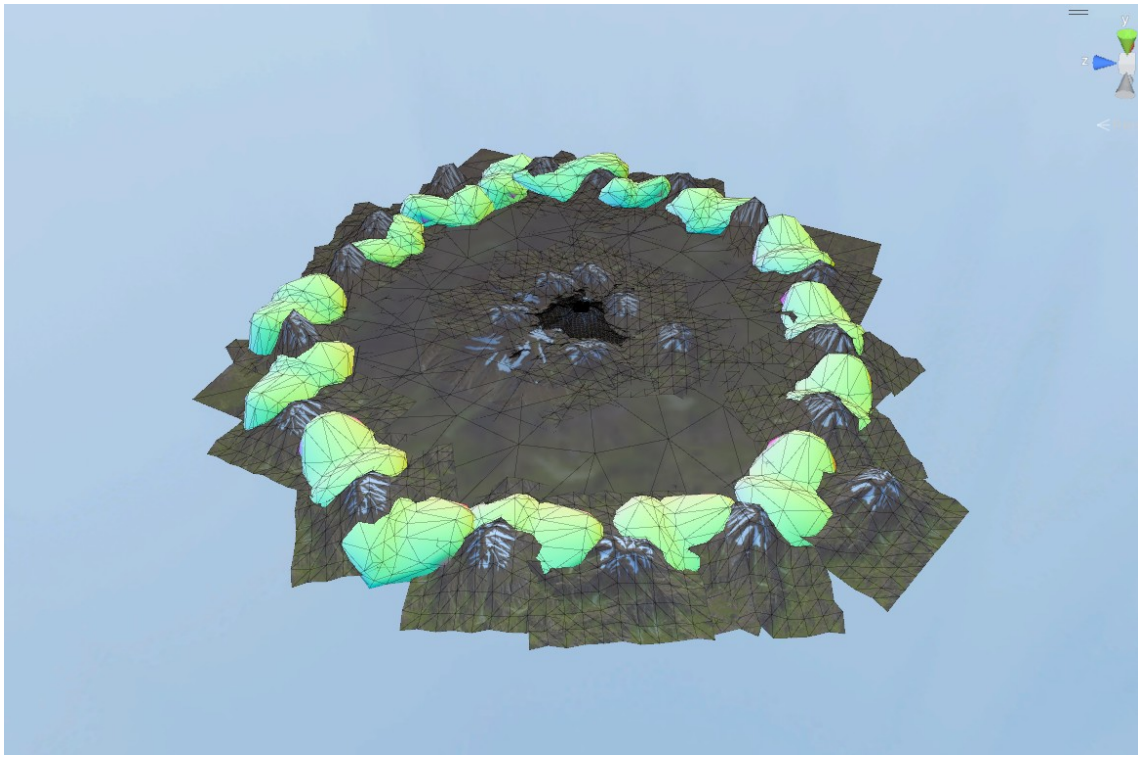
The drum detects when a bachi has hitted its trigger collider and fires a sound while the other collider detects if the bachi is too close and controls when the bachi will resent its position to fire another shot. This system gave me a lot of problems at the start because it was based completely on physics and was an operation between two colliders, one in the drum and another one in the bachi. This created bugs in the audio system and detected more collisions than necessary. That's why now these collisions are detected by the normal colliders and the sounds are reproduced by the trigger colliders.

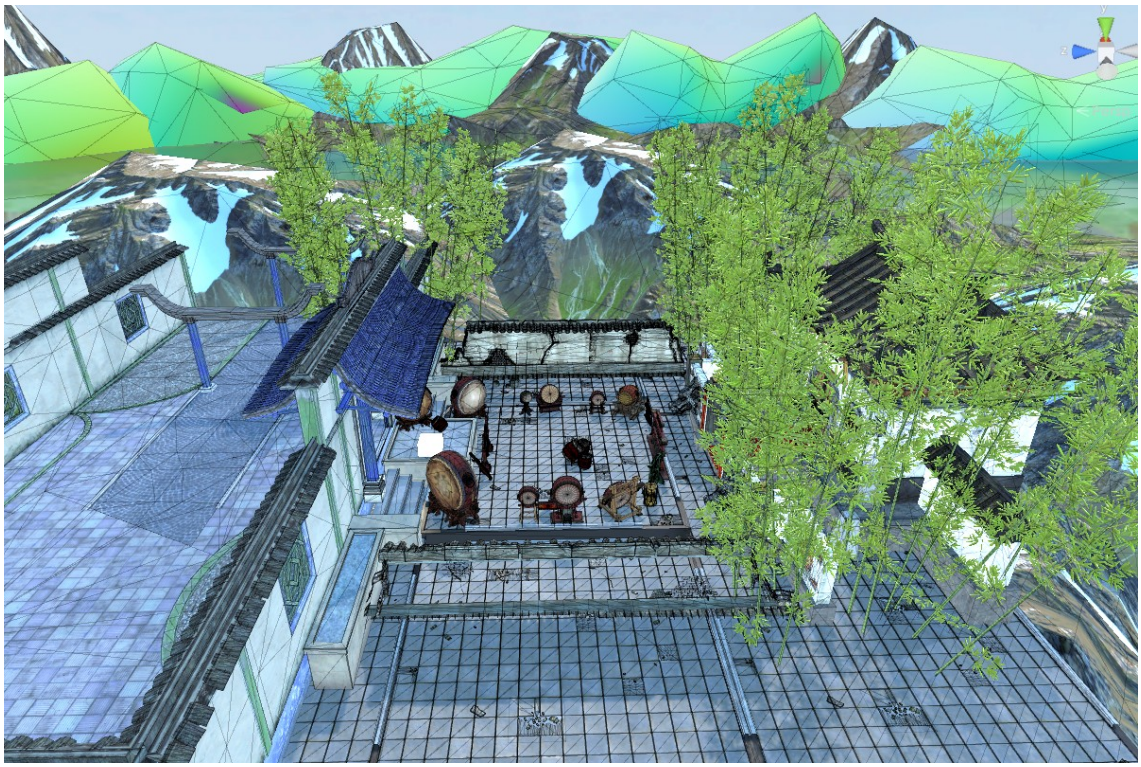
I implemented the formulas for the highway into a couple of scripts inside the gameobject called path and applied them to the scene. These shots are made by a system I created that analyzes a song given the audio file, but I was never able to really extract all of the flavor of this system, and I could not put them together with the physics system at the end of the day. They work really well independently but do not have a link in common between them.





While working on this, I created 3 more scenes. One that acts as the song main scene, another one meant for a sandbox and another one for the main menu.





5.3 Post Production.

The development process of the thesis hit a wall when the combination of the mechanics arrived. I needed a system to be able to create each note independently when a specific type of sound played in the song. To do this I did not have enough

knowledge of audio physics to be able to isolate specific instruments from a track or to know when they have played after the previous work was done. So I started searching and found an SDK called Maestro - Midi player tool kit.

This library allowed me to import a midi file with specific events when played that could be called anywhere from the project. This was a huge leap forward in development, but sadly I still was not able to completely make it work despite my efforts.

At the end of the day this project was way bigger than what I had imagined and required a lot more knowledge than the one that I thought would be enough, but I have learned a lot through the process and know that the problems that appeared where just solved with more time and effort, which in my case I run out because of my personal error at measuring the dimensions of this thesis.

With these two sections combined the rest of the small details like the score or the game loop would have been easy to complete, and would give this game a completely different look. But I am still proud of what I was able to accomplish at least for myself. Is not the best but I have invested a lot of time into it and I can still look at it with a straight face.

8. Bibliography

- The whole bibliography is in this notion's anex:
- https://fantastic-maple-40c.notion.site/f09c05f0c37448fab6037ccfca305afd?v=fc6dd3c97db94b9b9a25a6023b9d8473&pvs=25