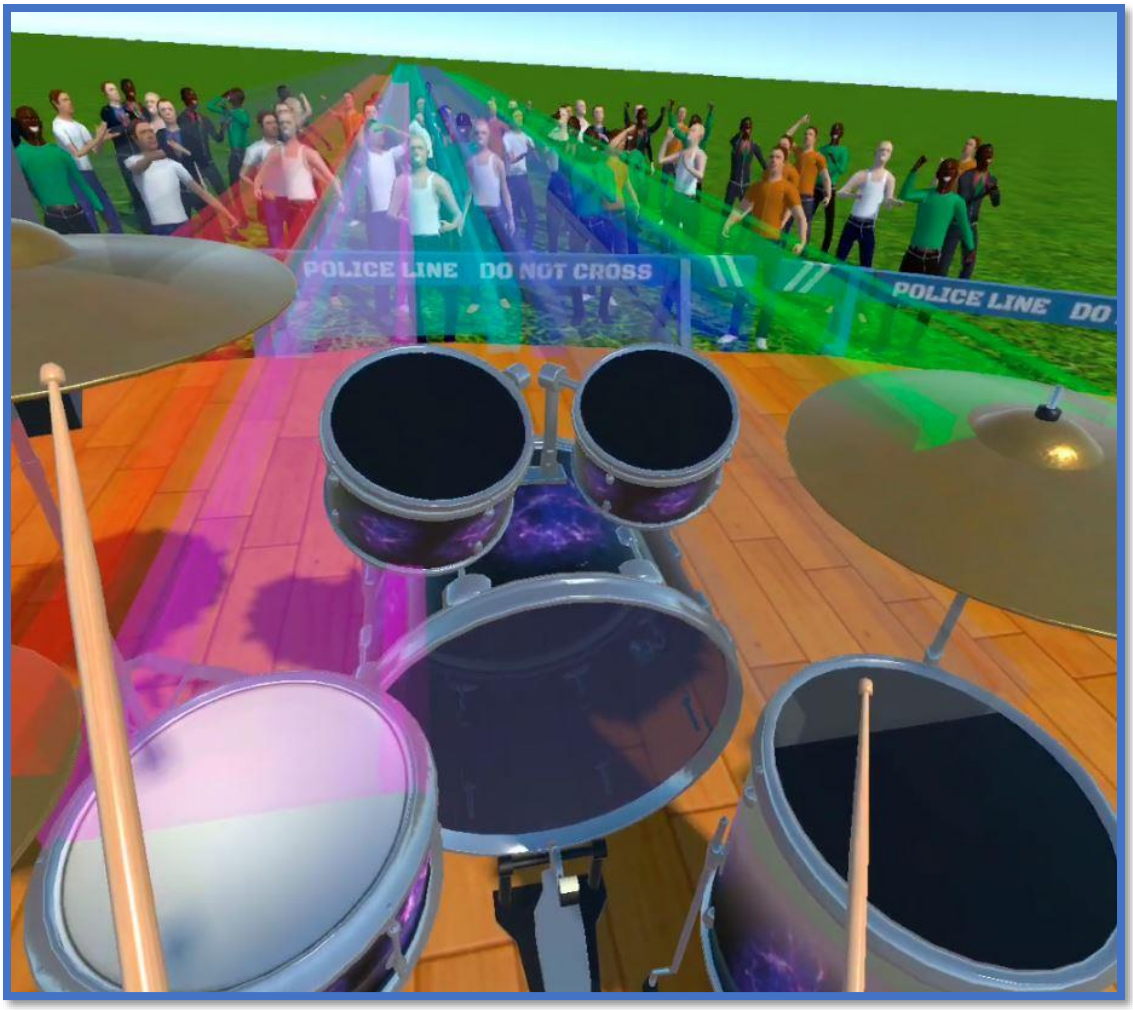
Drum Legends – A VR Game

Names: Barak Biber, Noa Pariente.

Supervisors: Boaz Sterenfeld, Yaron Honen.



Contents

Introduction

Drum Legends is a virtual reality drummer game.

In this game you drum on a virtual drum set which includes even a bass drum!

To play the bass we constructed a DIY bass pedal which anyone can construct at home using basic materials and a Bluetooth mouse as the actual controller. Everything is affordable and easy to acquire.

In the game you can play your favorite songs in front of your fans and have a lots of fun.

Website

Video

Technologies and Platforms

Our VR game was developed using Unity for the headset Meta Quest 2 using the XR plug-in framework and XR Interaction Toolkit. The code was written in C# and edited using Visual Studio Code.

Unity

Unity is a cross-platform game engine which is simple to use and gives anyone the ability to create both 2D and 3D games for many platforms.

Also, to create interactive songs in the we developed a Python script using PyCharm which takes textual tab representations of a song and parse it to ".level" file. We will provide more information in "How to create a song" chapter.

Unity XR plug-in framework

Unity has developed a new plug-in framework called XR SDK that enables XR providers to integrate with the Unity engine and make full use of its features. This plug-in framework provides a way to develop an XR application (AR or VR) and easily support multiple platforms for XR. The XR Interaction toolkit package is a high-level, component-based, interaction system. Which provides easy tools to create cross-platform interactions in XR.

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. It is a free open-source IDE which can be used with a variety of programming languages. There are many extensions for it which enable better features.

Equipment

The Meta Quest 2 as the Virtual Reality headset and a DIY bass pedal which uses a Bluetooth mouse as the electronics making the pedal completely wireless.

Meta Quest 2

Oculus Quest 2, marketed since November 2021 as Meta Quest 2) is a standalone virtual reality headset developed by Meta Technologies, LLC. The Quest 2 was officially unveiled on September 16, 2020.

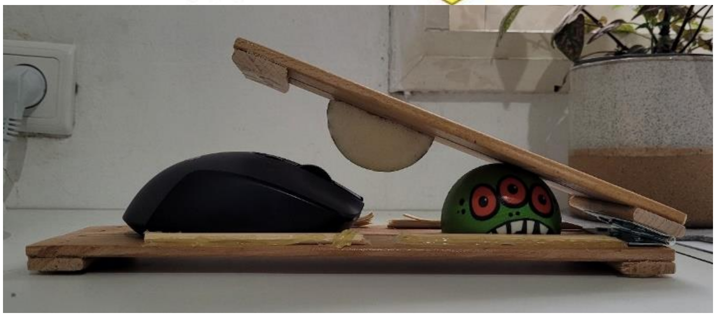
The headset can run as a standalone device without the need for a computer running games installed on the device. Or connected to a computer using either a USB-C connection or over WiFi running desktop games which use the Quest 2 as the VR headset.

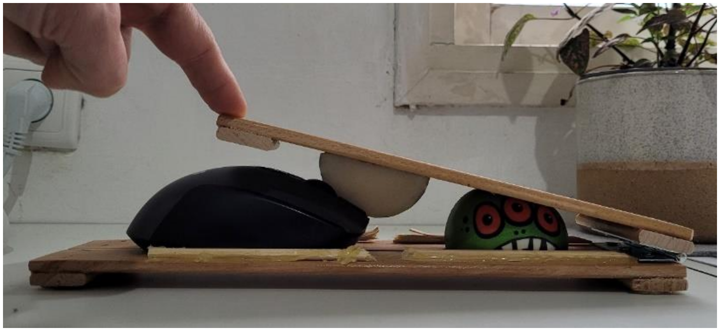
Bass Pedal pairing in Bluetooth.

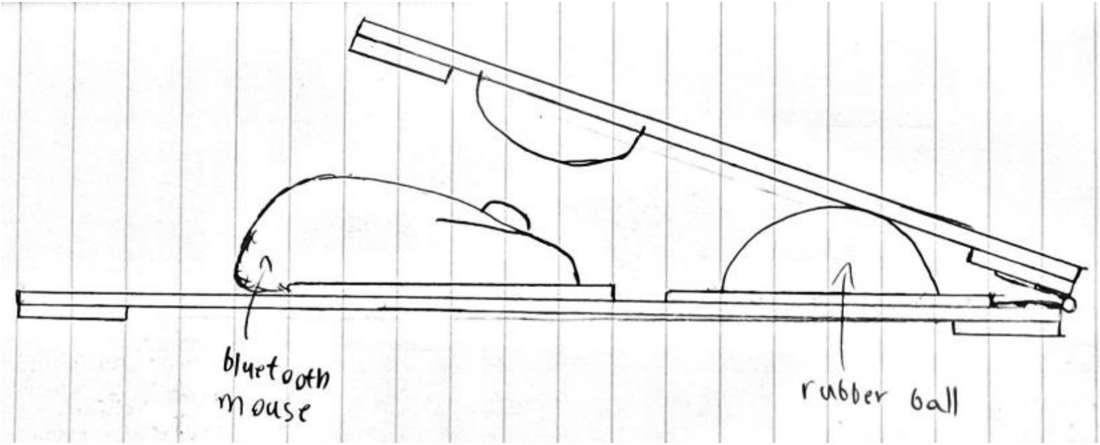
DIY Bass Pedal

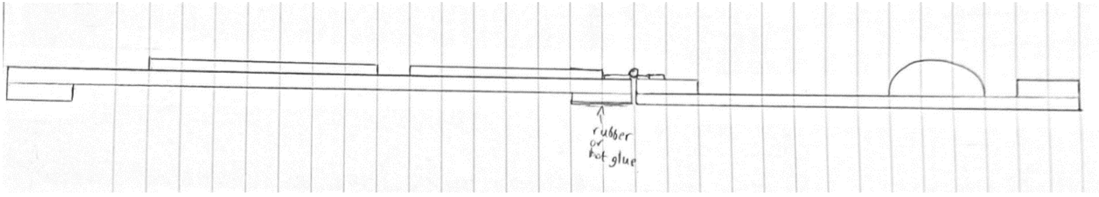
List of Supplies:

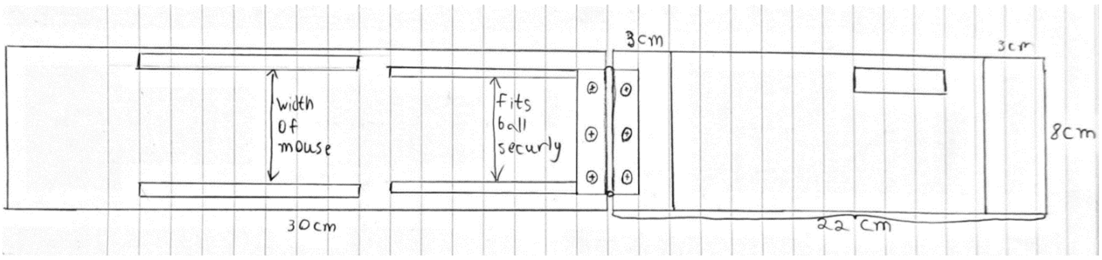
- Hinge
- Hot glue
- Wood
- Chopsticks
- Bluetooth mouse
- Sponge/rubber ball
- Nails
- Screws
- Bluetooth Mouse (We used Logitech G603 Lightspeed Wireless Gaming Mouse)





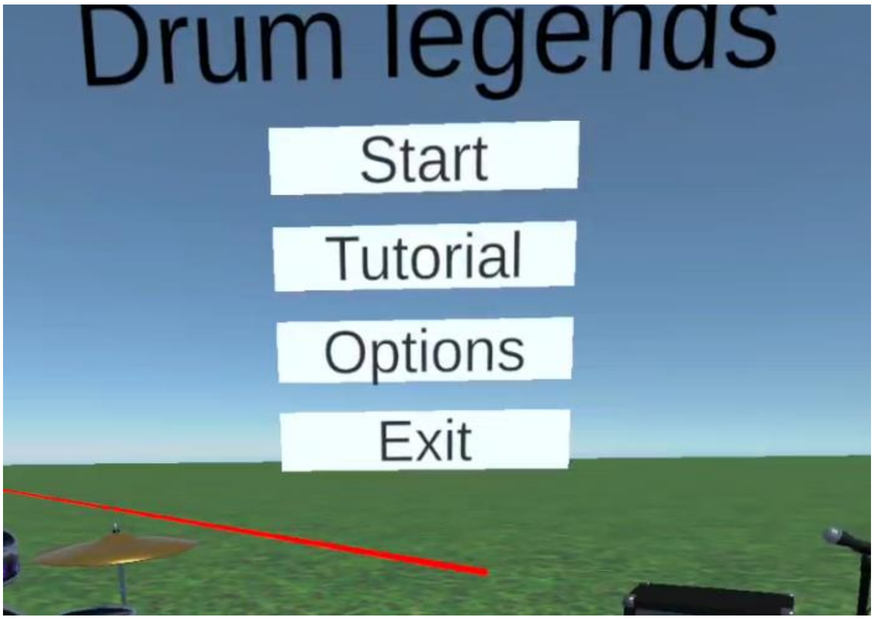

bluetooth mouse
rubber ball


rubber or hot glue


3cm
3cm
width of mouse
fits ball securly
8cm
30cm
22 cm

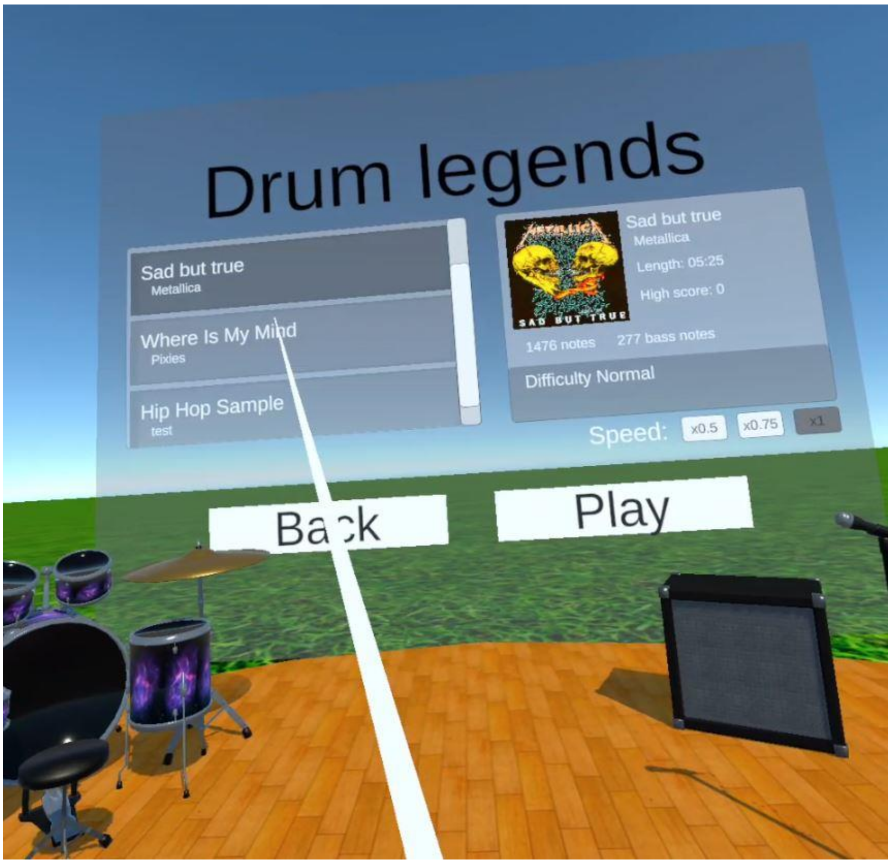
Scenes Overview

Main menu



Start

List of the songs you can play. You can see metadata about the song, name, author, and length. As well as information about the number of notes and specifically bass notes. The player can also see the highest score achieved for the song. There is an option to play the song at slower pace 50% or 75% slower to practice or just take it easy.



Sad but true
Metallica
Length: 05:25
High score: 0
1476 notes 277 bass notes
Difficulty Normal

Tutorial

A short explanation about the Bass drum and how to build the bass pedal. As well as a short tutorial how to play the game with two short samples of music you can try.

Options

In the Options menu there are two options.

- Turning the Bass Notes off, meaning all songs are played without the bass notes
- If playing with bass setting the key which triggers playing the Bass drum, this was done using interactive rebinding, meaning once the player clicks the option the next key he presses is re-binded as the bass key.

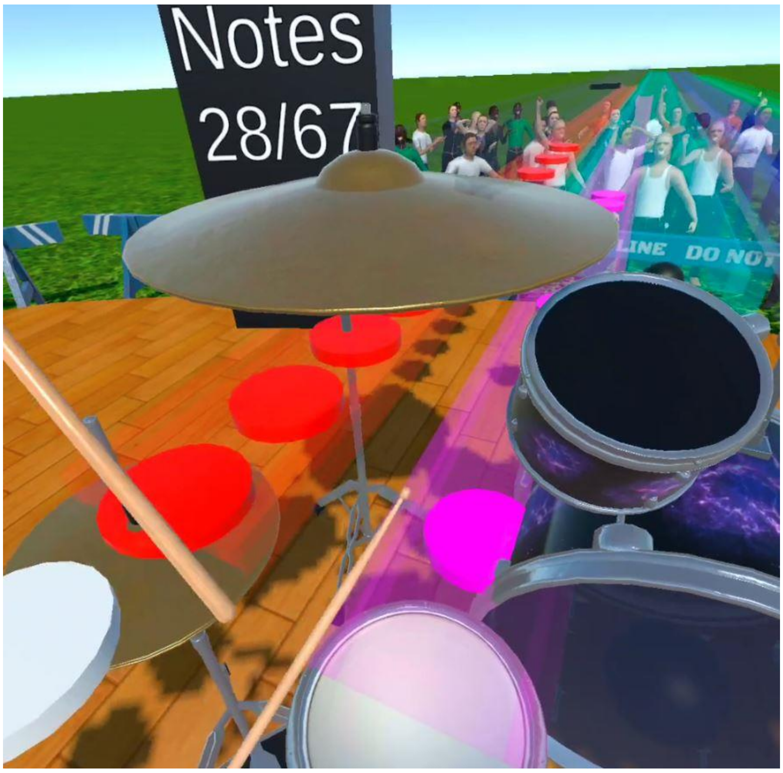
The settings are saved to an XML file and are loaded when the player restarts the game meaning the settings are saved.

Level

The level scene is the heart of the game, in it the song is played in the background and colored notes are moving towards the drum set, when each note reaches the drum the player can hit it using the drumsticks which are the controllers and receive a hit, or if it is a bass note hit it using the key as set in the options menu.

The game keeps the hit count and provides a score which is calculated based on the distance of the hit from the center of the drum, the score is also multiplied if the player is during a combo, meaning he hasn't missed the last X notes.

- 2 note combo provides a multiplier of x2.
- 3 note combo provides a multiplier of x4.
- Any higher combo provides a multiplier of x8.

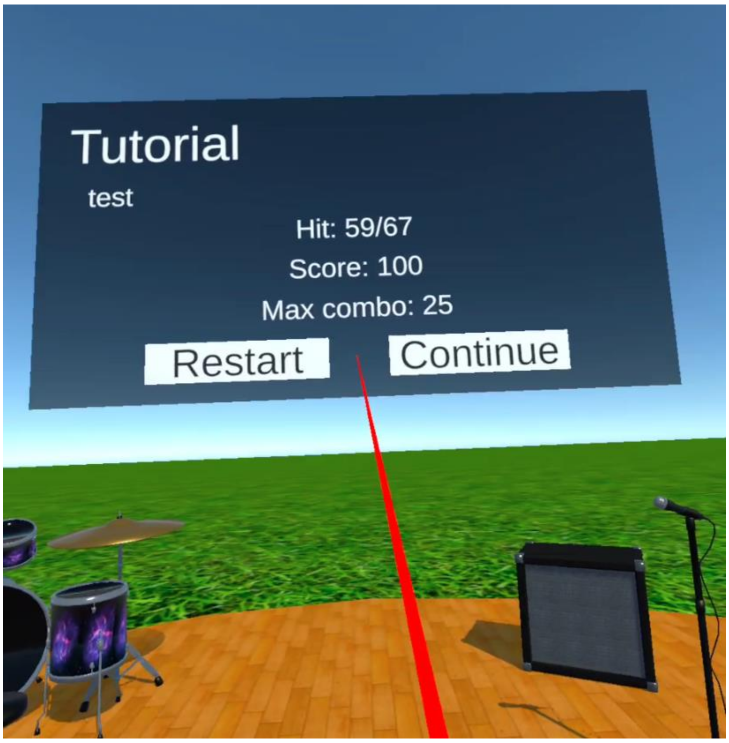


Finish Scene

When the player finishes a song in the level scene a Finish Screen is presented to him.

In it the Song name and Author is displayed, as well as the hit count, score, and max combo.

If the score is higher than any previous score on the same song the game announces to the player he has a new high score and it is saved.



Tutorial

test

Hit: 59/67

Score: 100

Max combo: 25

Song player

This scene is not accessible to the player and is not compiled into the final game but was created in the same Unity project to help construct new songs. We will provide more information about this scene in the next Chapter How to create a song.

Songs

Level file

The levels in the game are created using a human-readable XML file with the extension ".level".

The XML file has the following fields:

- SongAudioFile – A relative path to the audio file.
- Metadata – a dictionary with the following keys
    - AudioType – the audio file encoding, for example MPEG for mp3.
    - Name – The song's name
    - Author – The song's author
    - Difficulty – One of the following difficulties: Easy, Normal, Hard.
    - Length – The song's length in the following format HH:MM:SS
    - Image – Optional, an image file to present in the song selection menu
- Notes – A sorted array by start time of NoteDescriptor elements, each NoteDescriptor describes a single note and has the following fields:
    - start – The time in seconds the note appears in the song
    - drum – The note's drum, it can be one of the following: Snare, FloorTom, TomTomLeft, TomTomRight, Crash, Ride, HighHats, Bass.

The levels in the game

In the main menu scene, all ".level" files are found in the song's directory and are parsed to create the song selection menu using the metadata.

Once a player chooses a song to play, the file is parsed.
The song's time is updated at the FixedUpdate routine and if the next note in the list should be generated, meaning the note's time minus the time it takes it to reach the drum has been reached it is created, and sent on its way.

Creating the level

For creating a level we developed a Python script which takes textual tab representations of a song, for example:

```
C:|x---------------|----------------|----------------|----------------|
H:|--x-x-x-x-x-x-x-|x-x-x-x-x-x-x-x-|x-x-x-x-x-x-x-x-|x-x--x--x--x----|
S:|----o-------o---|----o-------o---|----o-------o---|----o--o--o--oo-|
B:|o-------o-------|o-------o-------|o-------o-------|o----o--o--o----|
```

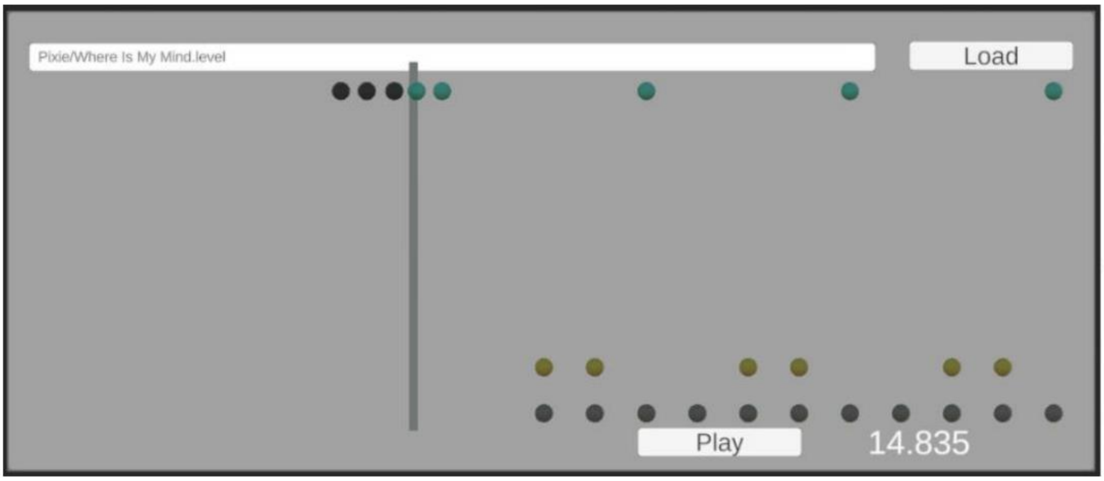
and converts it to a level file.

The script converts it to the games .level file using the provided settings:

- BPM – Beats Per Minute
- FIRST_NOTE_TIME – The time which the first note appears
- DRUM_DICT – A dictionary the maps between the drum name in the text tabs to the drum name in NoteDescriptor
- NOTES – Array of the signs of notes in the text tab
- PATH – path to the text tab
- LEVEL_NAME – Level's name.

- SONG_AUDIO_FILE – The song's audio file name
- NAME – The Song's name
- AUTHOR - The song's author
- DIFFICULTY - One of the following difficulties: Easy, Normal, Hard.
- LENGTH - The song's length in the following format HH:MM:SS

To test the resulting level the 2D scene was developed which plays the song.



The scene plays the song in the background and generates the notes as described in the .level file.

The scene is in 3D but the camera is set as orthographic and with the UI as its world. the notes themselves are 3D and appear between the camera and the UI and are generated in the same manner as the notes in the game so to provide a reliable testing environment.