GLOBAL SCALE GDD

Preliminary Design

BACKGROUND INFO

Target Audience

This game is intended for a typically younger audience or people who have limited experience controlling a 3D character.

Similar Games

There are 3 main games which our game will take inspiration from.The game's environments and overall mood will take inspiration from The Witness. The smooth 3D exploration through unfamiliar cartoon landscapes we feel will fit our game well. The other game which we will take mechanical inspiration from is Toree 3D. While it is a fairly niche game, the simplistic,

focussed 3D platforming is very similar to what we plan our in-game levels to be like. Some level gimmicks will take inspiration from Neon Beats (). The way this game syncs the level's mechanics with the background music will serve as inspiration for some level gimmicks in our game.

GAMEPLAY

High Concept

A player will explore an overworld, finding levels which house mechanically focussed platforming.

Detailed Overview

Description of player role, player goal, what the player will do to achieve the goal, and explanation of why the game will be fun. Introduce the world setting and artstyle

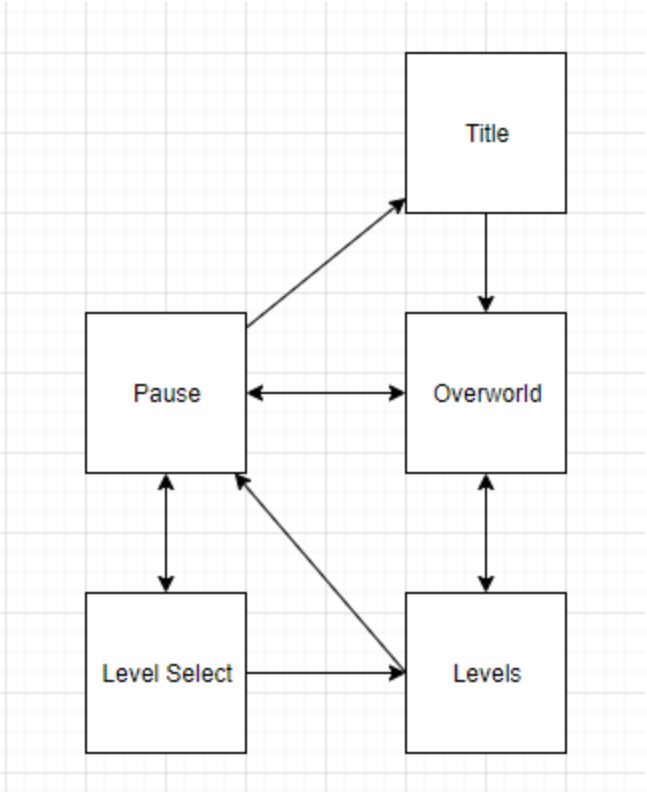
Key Features

- 3rd Person Movement
- Low-Poly Art style
- Mechanically interesting puzzle platforming
- Sense of exploration

First Minute

After clicking play on the title screen, you will be put into the world, where you will walk around and explore the first island in the game.

Gameflow



Game Objectives

Overall: The objective of the game is to gain the 7 orbs by completing puzzles centered around 7 mechanics, in order to complete a major scale and restore music to your islands.

Small Scale: In order to gain a crystal, a player will have to either complete a puzzle-platforming level, or find a “secret” hidden on the overworld. Each island will have one warp point to a level, and one crystal which can be found be exploring.

Game World

Plot

The plot is unimportant to our game, as we intend it to be a mechanical and level design showcase rather than a cohesive narrative. The player will control a small robot traversing human environments unfamiliar to it.

Characters

The player controlled character is the only one present.

ART

Graphics

The game will have a predominantly low-poly look. This is done for 3 reasons. Firstly, the fewer polygons a 3d model has, the faster a computer is able to render it, so optimising performance won't take as long. Secondly, it's easy to develop. Thirdly, when done right, low poly without textures can look highly stylised and effective. The graphics take primary inspiration from The Witness.

Sound

The music for the game will be made using Musescore, an open source music notation programme. It will feature a rhythmic base, and simple melodies layered over the top. This will be easy to implement, and allows for variety between pieces while having a recognisable structure.

HUDs

The HUD in the game will be minimal, only occasionally requiring information on what object is currently locked on, and how many crystals have been collected.

Technical Requirements

Controls

Our player will require movement - through WASD, jumping - through spacebar, and interacting with objects - done with the mouse. Pausing the game will be using escape.

Chosen Software

Godot 3.1.2. Joseph has used godot extensively before, and therefore it is his engine of choice. However, at the time development started, there were 2 versions accessible. The stable, 3.1.2, with fewer features, and 3.2, which is in late beta, with more features. As of the start of development, we are unsure as

to what features will have compatibility issues if we decide to port over from 3.1.2 to 3.2 on release. But we decided to begin development in Godot 3.1.2, and take a look at the known compatibility issues upon the stable release and decide whether or not to port over.
Upon testing versions 3.2 and beyond, one particular feature (stopping a character on slopes) was removed, so we decided to stick with 3.1.2

THEME

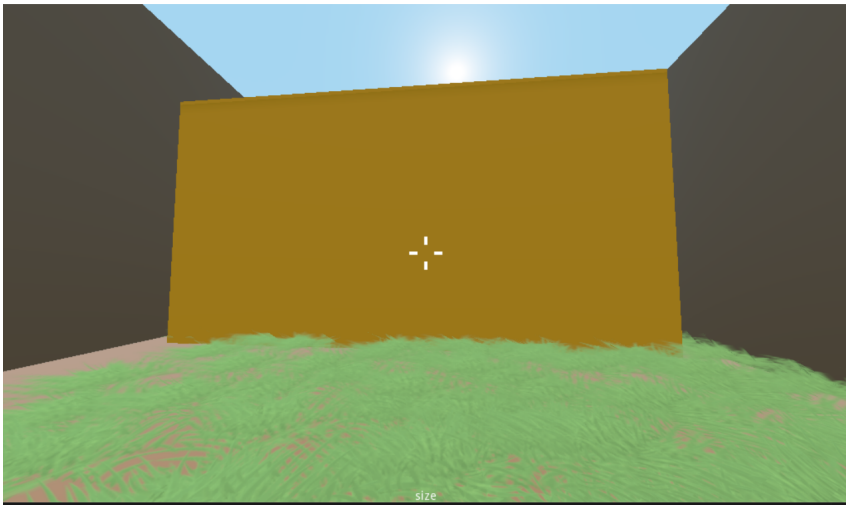
Our game strives to meet the theme by experimenting with the idea of a musical scale. The objective of the game is to collect all the notes of the major scale. Music also plays a major role in the platforming levels, as many of the level elements are synced with the music. A representation of the lack of music in the world will be in the music for the overworld theme, which will be barren and uninteresting until the orbs have been collected, when it will transform into a more musically interesting soundtrack.

Programming Development:

Note: Due to a lost Hard Drive midway through development, no in-engine photos could be taken for early prototypes

Version 1 - First Person Puzzle

The initial prototype of the game was in first person, with the main mechanics focussing around moving blocks, changing their size, and moving around in the world. This culminated in 4 main challenges:

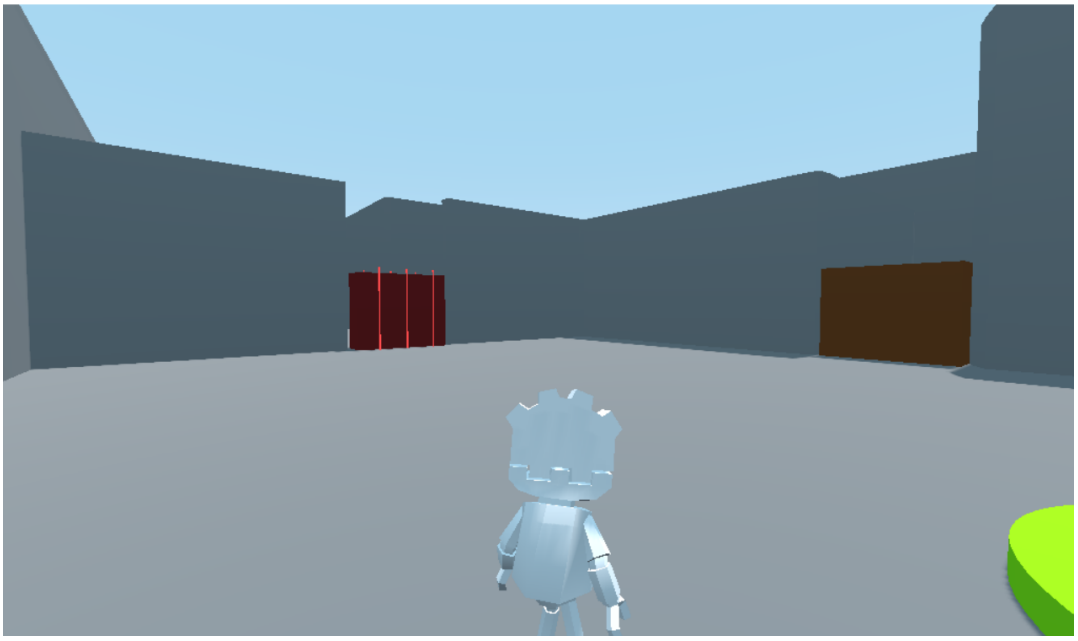
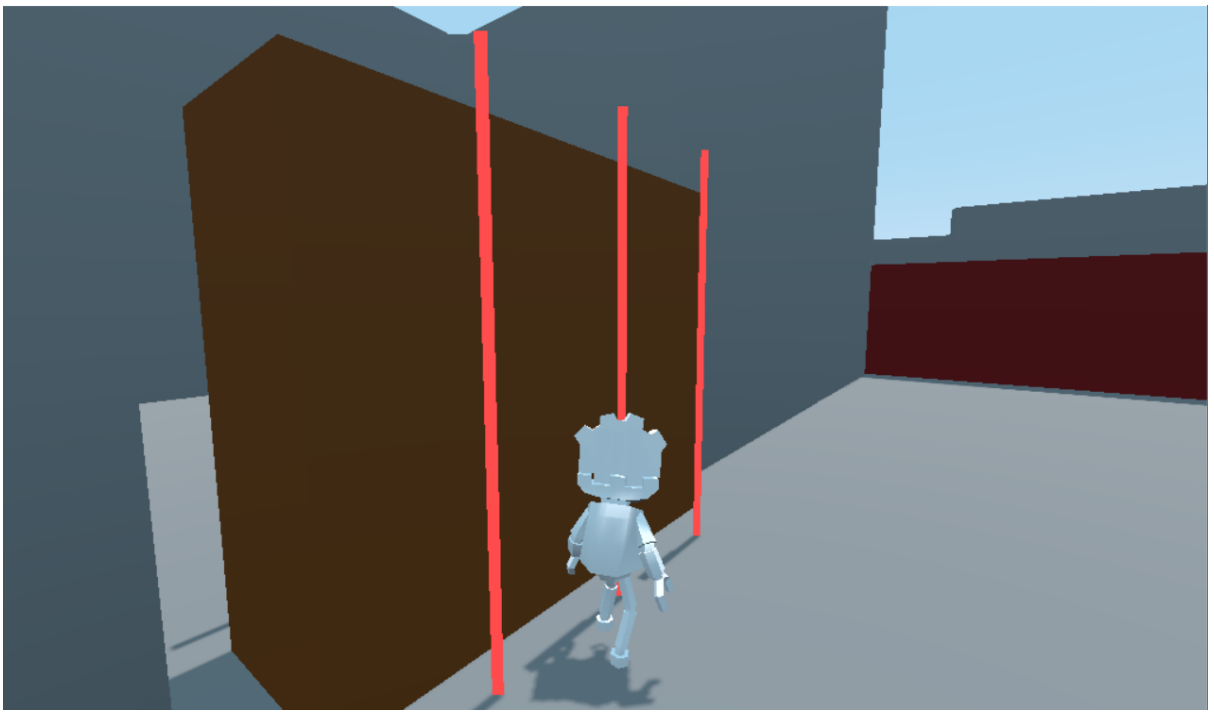
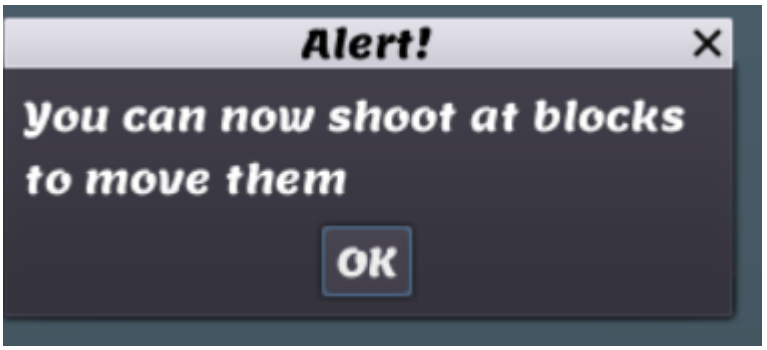
- Developing the Pushing and Pulling mechanics on the blocks went through 2 main iterations, one with rigid bodies and godot's in-built physics, the other with self-programmed kinematic physics. While rigid bodies were simpler to implement, they reacted very inconsistently, particularly with the player itself. As such, a kinematic body was used
- Various other physics issues cropped up along the way. One of particular concern was the player being able to brute force and push kinematic bodies when they weren't supposed to move, effectively skipping puzzles. While this was solved by adding a static body with limited physics layers on top of the kinematic body, this solution isn't necessarily trustworthy.
- The major issue came with the built in move_and_slide function on the player, and its interaction with slopes. When a player tries to move into a not perfectly vertical wall, they will go up it for one frame, then slide down again, leading to a "jitter". Much development time went into trying to find a fix and while the exact reason for the issue occurring was found, I could never find a reliable solution.
- 
- **First person camera as well as example of an in game block**

- 
- **Example of wall which caused “jitter” when walking into it**

Version 2 - Puzzle Box

Inspired by a GMTK Youtube Video (https://www.youtube.com/watch?v=pwHqY_4nsJ4), a second prototype was made to try and design levels within this style. While I found this style didn’t mesh well with our idea for the game, there were a number of elements that came from it.

- When trying to create lasers, I found the strengths and limitations for inbuilt VFX techniques in godot. When trying to add a “glow” to the laser, I ended up messing around with the World Environment node, and finding how oddly specific the glow values needed to be before even non-light emitting meshes began to bloom. I also found that when creating particle effects, the easiest way was to always use quad meshes with billboard enabled, and work on simple size and colour changes from there.
- Another big change was a switch to a third person camera. This was primarily done to reduce the visible impact of the slopes issue from the previous prototype, but it did end up leaving to a new challenge: character animation. I had to learn how to work with animation trees to create smooth-blending animation states - as well as design these animations myself, which I used Mixamo for. My skills I developed through this process helped when making the final character controller
- Some minor UI and HUDs were added to this prototype, which let me know of the constraints of using certain UI nodes - such as popups - and the immense benefits of others - particularly H and VBoxContainers when designing menu systems. These skills were furthered in a personal project unrelated to this competition.

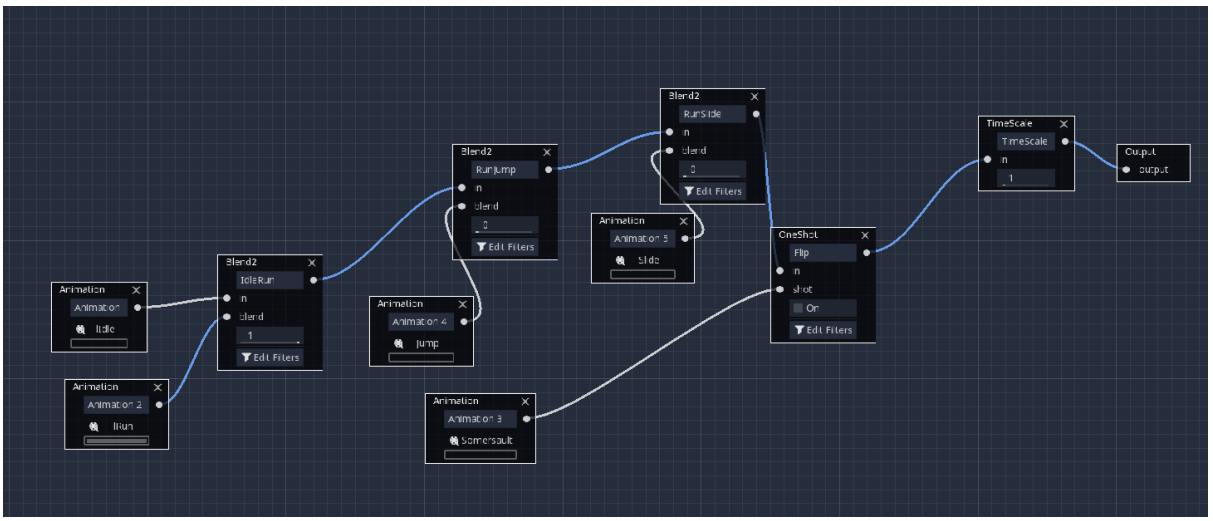
- 
- First attempt at a third person controller
- 
- First Attempt as Lasers (without glow)
- 

- **Early attempts at a Popup UI Alert**

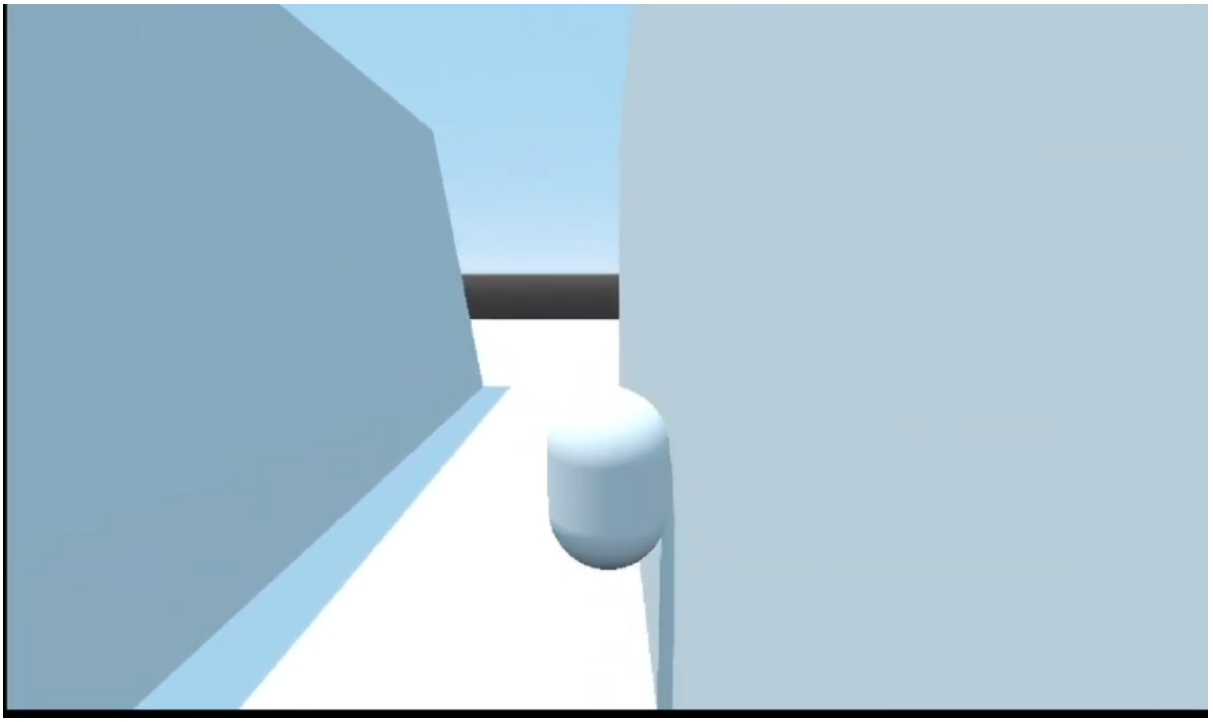
Final Version:

Unfortunately, after these prototypes had been developed, my old computer spontaneously stopped working, causing me to lose all my previous code. This placed time constraints on the final project, and this combined with our team's limited puzzle design capability, led to the final version. This version had more open areas, as well as music synchronisation - to better meet the theme as well as reduce development while keeping mechanical interest in the game.

- The character controller was completely re-written, and so got some major upgrades, particularly with regard to the movability of the character. Thanks to some increased knowledge of 3D vector maths, I was able to make a well-functioning double and wall jump system for the character, which felt much better to control than the previously rigid system
- The final character used in the game was inspired by devlogs by Robert Thomson (https://www.youtube.com/watch?v=yV2YAwmg6yk). This technique allowed me to make simple animations by hand, which meant they blended much more easily in game. As an added bonus this character's scale fits in nicely with the world, unlike some of the other characters.
- The music synchronisation posed some issues. The system used was to use a rigid BPM, and manually set a looping timer to notify an autoloaded script when a beat had occurred. The main issue came about when trying to loop this music, as the ogg file from musescore added a delay at the end of the audio file. In hindsight, I should have used Audacity to manually remove this delay, but the system in place - which waits for a beat to finish before restarting the loop - works well enough most of the time, even if it does remove some immersion due to pauses in the music.
- Finally, a major issue faced by me was the import of 3D objects (and materials) from blender. While initially using obj files, I found that the exports sometimes added faces in unwanted places (such as over the hole of a well), and overall the materials were dark, bland, and unsaturated, despite being noticeably more vivid in blender. The solution was to replace every model in the game with a gltf file, which had a much larger file size, was much more reliable. This was a significant time investment, as this change was implemented late in the process.

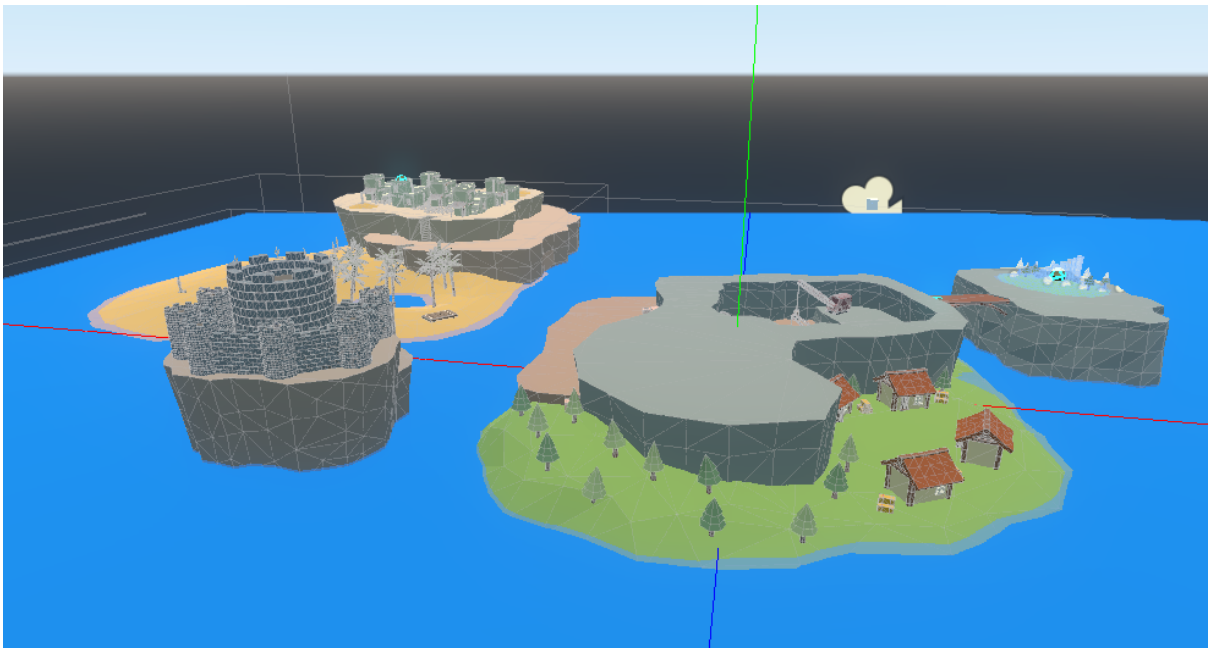


Final Animation Tree for the Player Character

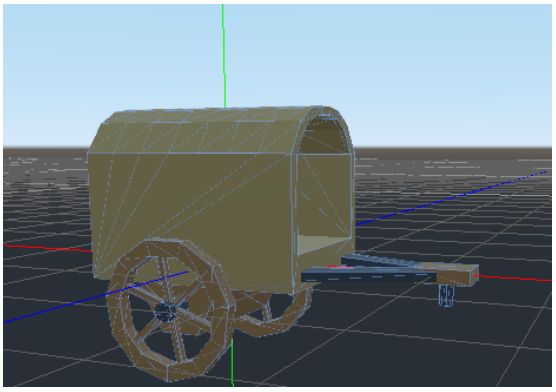


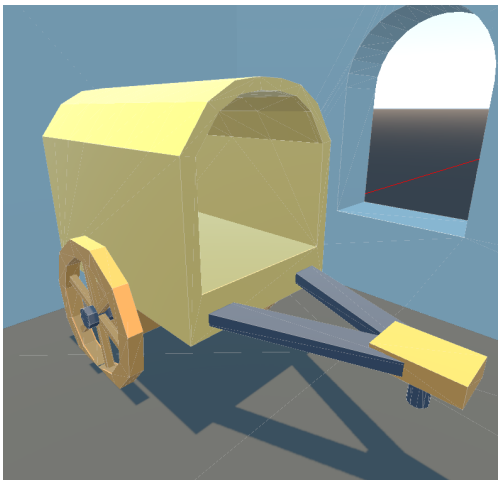
Early Wall Jumping Test

Music Synchronisation Setup

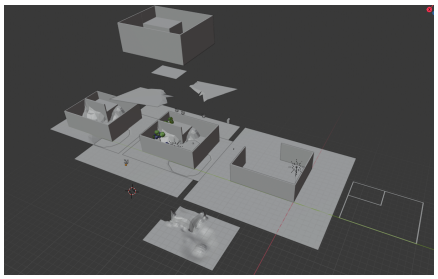


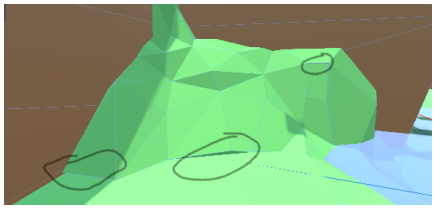
Final Layout of Overworld





OBJ vs GLTF Differences





Phase 1 of Art Development

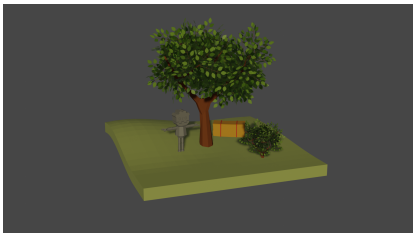
Blender was chosen to be the primary software for 3D art creation throughout the development process, as it is free, open source and has a plethora of resources for learning how to use it. I started the development process with limited knowledge of the tools within Blender, and early level models were slow to produce, and had many imperfections which created problems with lighting and player interaction with surfaces (such as normals for meshes being reversed for some models, which would make shaders not work in Godot). Shown to the left are some of the models used in Phase 1.

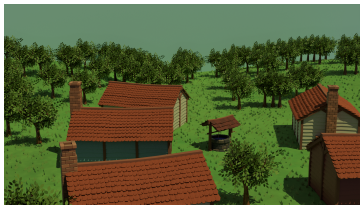
Phase 2

Over time however, I became more confident with the program, and how to set out a scene in blender to organise models and their components. This was Phase 2 of the at development, where I experimented with detail and making more complex shapes with the software. It was then that I encountered the most trouble with the skill of keeping a consistent art style across all the art for the game, which I was finding especially difficult when dealing with low-poly models:

Low Poly

Low poly is a style of 3D art which refers to models comprised of a low number of polygons. The technique of reducing polygon count is often used as a tool to optimise larger games, where it is imperative to reduce the amount of processing which must be done on objects that aren't in immediate view of the player (ie. for objects in the distance the game will reduce the resolution of shapes to decrease the amount of processing needed). For this game, we intended to use Low Poly as an art style. This has been done on many occasions, notably on games such as The Witness (Thekla, Inc.) or Abzû (Giant Squid Studios). *Images from these games, used as inspiration, can be found at the end of this report.* The use of stylised Low Poly raises the issue of having a consistent amount of detail across models, such that the style feels consistent.



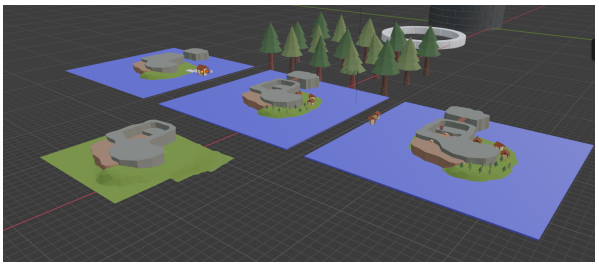


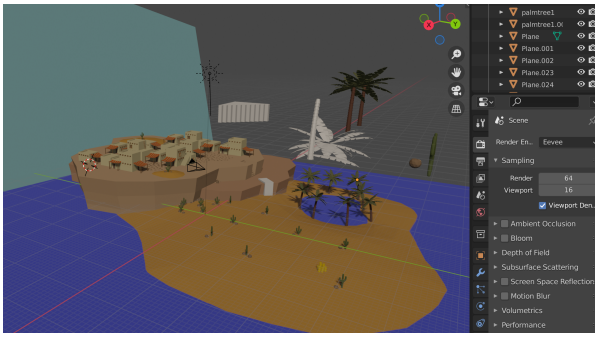
Phase 2 was characterised by models with a relatively high amount of detail compared to other phases of development, such as the well and tree seen to the left. This art style was ultimately unsustainable for the entire development, as I lacked the expertise and time to complete enough models (with that level of detail) to use on the entire game.

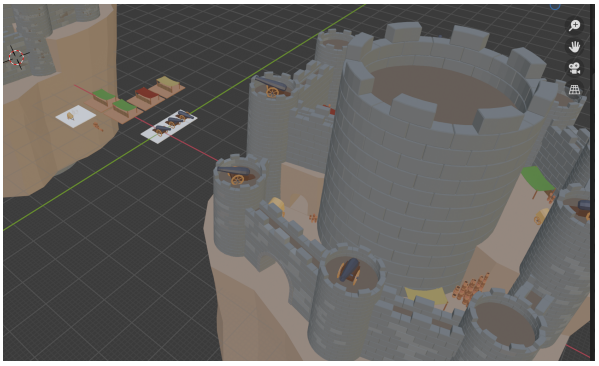
(Note: the well model was created following a tutorial by Grant Abbitt (https://www.youtube.com/watch?v=bWHhx4uos84)).

Phase 3 (Final Iteration)

This was the final phase of art development, where I applied the learning from previous phases to develop an art style which was relatively consistent and time-effective. It was at this point that the gameplay structure of a central overworld, with separate sub-levels, was finalised. Art development was thus split between assets for each of the three islands, as well as layout and population of the islands and the sub-levels. In this phase, I made a model for each island with unique landforms and structures for each area, then made assets that served both as aesthetic interest, and as areas for coins and orbs to be hidden. I worked separately on the sub levels, but still recycled assets between areas to add interest to each level, and link them to the theme of the area. As part of my design process, I would create renders of each scene using a background (as seen in the picture of the desert island), to ensure that the colours and layout of the areas would look natural and consistent in game.

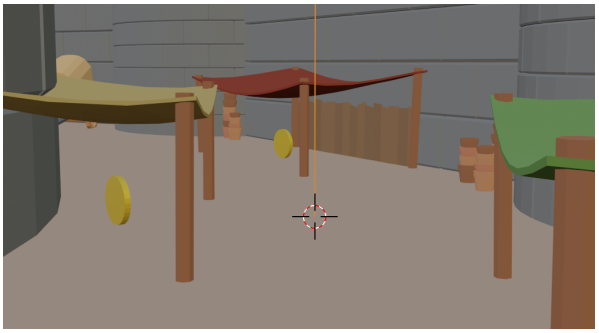




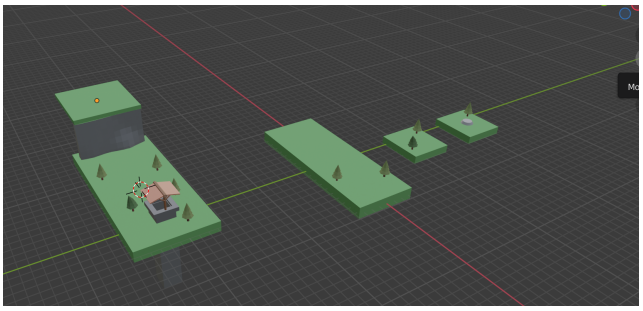


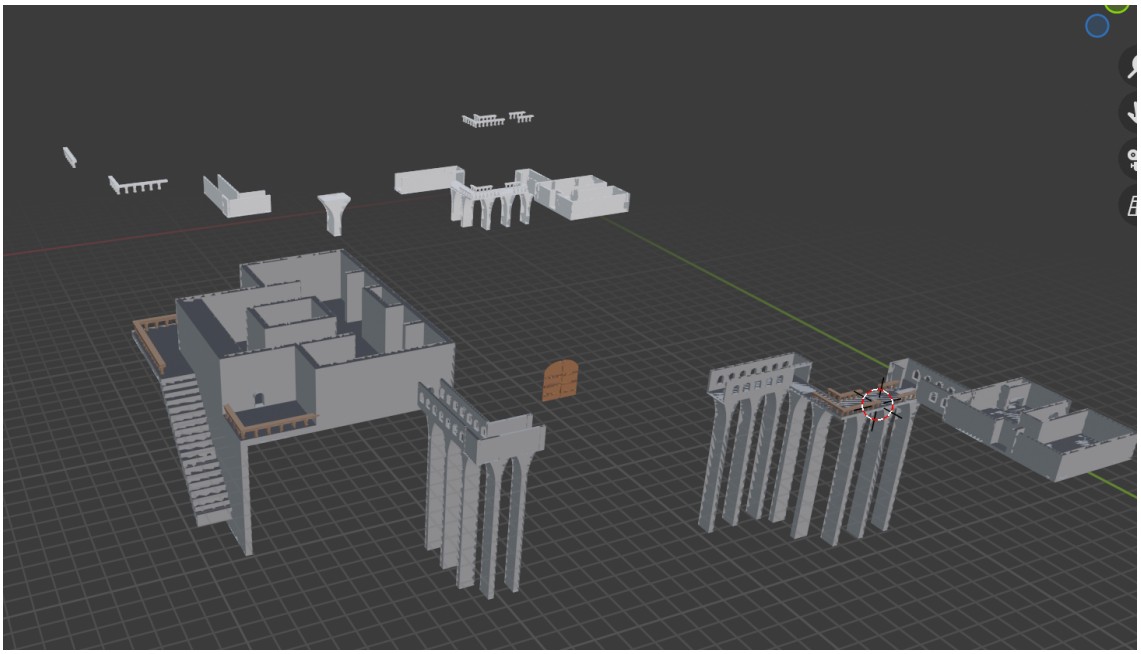
I employed multiple measures to ensure that the style remained consistent throughout the world. These included using the Decimate modifier in blender to roughly match the polygon resolution of each of the assets, and designing shapes to have similar parent polygons (such as cylinders, which I mostly modelled as heptagonal prisms). The intended result is a world which, although it is obviously designed to be an artistic representation of real world objects and landscapes, is consistent enough to allow the player to be immersed into the world.

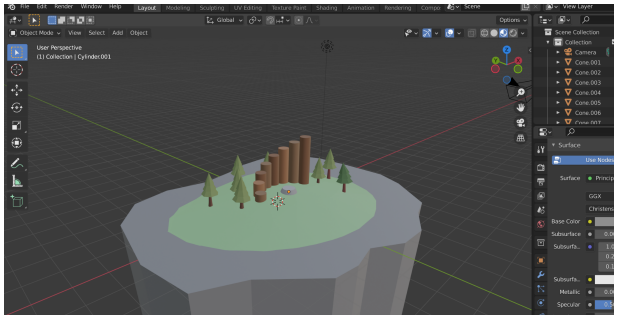
The sub levels were designed in two different ways. In the desert and castle levels, I made a complete model of the level in Blender, which was then imported into godot and populated with mechanics. For the village/plains level, I created platforms and assets to fit a level that had already been designed; these platforms were then substituted into the final level. Some examples of the sub level design

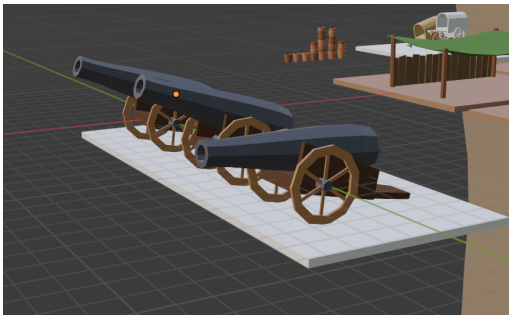


process, and models for several assets, can be found below:





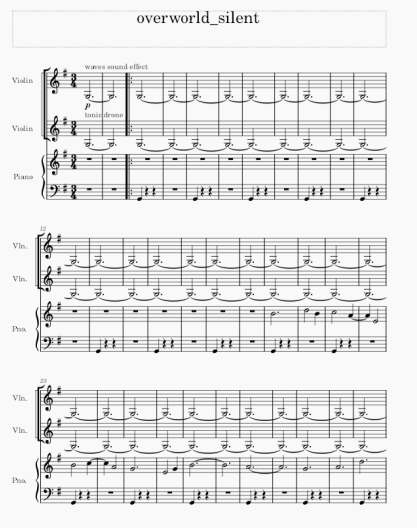




The images to the left and below are an example of the inspiration to the art style in Global Scale. (Source: Abzû (Giant Squid Studios)).

Music

The design of the music for Global Scale was important to the overall design of the game, as we were interpreting the theme “scale” to refer to a major scale within music terms. We used the concept of a shrine and collectible orbs as notes in a scale, so as the player journeys across the world, they are restoring the notes to the scale and ultimately bringing harmony and music back to the world. All the music in the final game was created using musescore, a free open source music notation software. This allowed us to create the music in a way that was more natural to us, as we study music at school and have a relative fluency with music notation, while being inexperienced in using electronic music production software. The music was designed to feature a repeated ostinato pattern, with harmony and melody lines layered over the top gradually. Each sub level has its own music with unique rhythms and melodies, such that the music helps immerse the player in each area. Additionally, the music was designed to loop, such that it could provide a continuous sound to the player to enhance the gameplay experience. Some images of the scores of the music are shown below. Once the music had been finalised, it was exported to .wav format and incorporated into the game.



The music for the overworld took special consideration. We wanted to give the effect of a world devoid of harmony, until the orbs were found to restore music to the world. This was done using a thin instrumentation for the first half of the game, including only a drone on the note G (the tonic for the overworld music). This was populated with some simple melodies which added some interest to the music. Once four of the eight orbs had been found, this sparse music was changed to a music richer sound, with instruments providing a constant harmonic base, while the same melodies from before were layered over the harmony and rhythmic ostinato, thus being reminiscent of the world before. This gave the effect of music returning to the world, as well as refreshing the experience for the player mid-way through the game.