

Design Document for:

Sound Slingers

Julia Krawiec, Artist, krawij@rpi.edu
Tatum Hobby, Artist, hobbyt@rpi.edu
Andrew Sirkoch, Programmer, sirkoa@rpi.edu
Xavier Marshall, Programmer, marshx@rpi.edu
Hibiki Takaku, Programmer, takakh@rpi.edu
Greg Sirkoch, Musician, sirkog@rpi.edu

All work Copyright ©2020 by
Julia Krawiec, Tatum Hobby, Andrew Sirkoch, Xavier Marshall, Hibiki Takaku, Greg Sirkoch

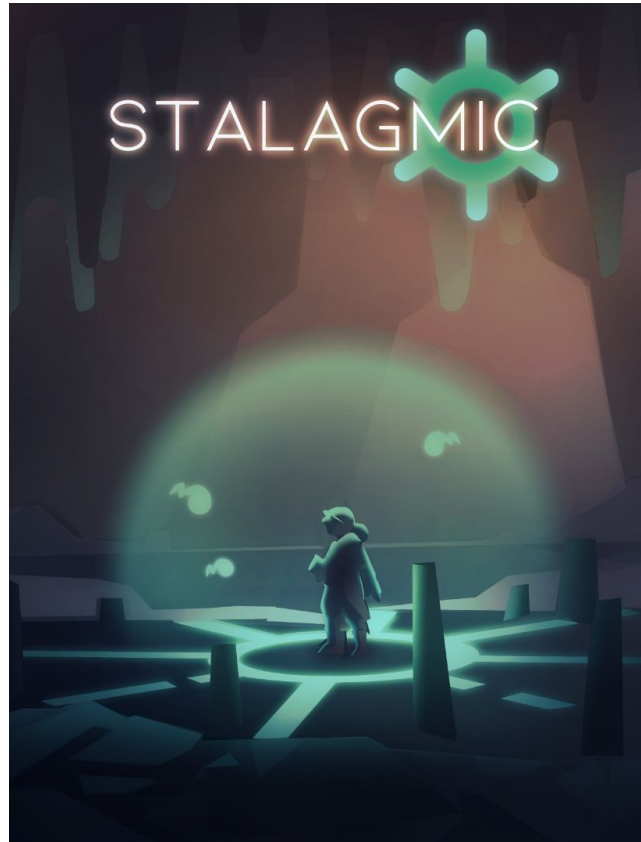
Table of Contents

Introduction & Story

SOUND SLINGERS	1
GAME OVERVIEW	5
COMMON QUESTIONS	5
WHAT IS THE GAME?	5
WHERE DOES THE GAME TAKE PLACE?	5
WHAT DO THE PLAYERS DO?	6
HOW MANY CHARACTERS ARE INVOLVED?	6
WHAT IS THE MAIN FOCUS?	6
DESIGN HISTORY	6
VERSION 1.00	6
VERSION 2.00	6
FEATURE SET	7
GENERAL FEATURES	7
GAMEPLAY	8
THE GAME WORLD	9
OVERVIEW	9
PREHISTORIC SECTION	9
THE PHYSICAL WORLD	10
KEY LOCATIONS	10
TRAVEL	10
SCALE	10
OBJECTS	10
ASSET LIST	11
RENDERING SYSTEM	11
OVERVIEW	12
2D/3D RENDERING	12
CAMERA	12
OVERVIEW	12
TARGET SWITCHING	12
ZOOM OUT TO ENCOMPASS ALL ACTIVE LIGHT FAIRIES	12
GAME ENGINE	12
OVERVIEW	12
GAME ENGINE DETAIL #1	12
COLLISION DETECTION	12
LIGHTING MODELS	13
OVERVIEW	13
THE PLAYER'S SINGING	13
THE WORLD LAYOUT	13
OVERVIEW	13

GAME CHARACTERS	13
OVERVIEW	13
ENEMIES AND MONSTERS	13
USER INTERFACE / EXPERIENCE	15
OVERVIEW	15
USER INTERFACE DETAIL #1	15
MUSICAL SCORES AND SOUND EFFECTS	16
OVERVIEW	16
SINGLE-PLAYER GAME	16
STORY	16
HOURS OF GAMEPLAY	16
VICTORY CONDITIONS	16
SINGLE-PLAYER GAME	17
STORY	17
HOURS OF GAMEPLAY	17
VICTORY CONDITIONS	17
CHARACTER RENDERING	18
OVERVIEW	18
CHARACTER RENDERING DETAIL #1	18
CHARACTER RENDERING DETAIL #2	18
CHARACTER RENDERING DETAIL #3	19

Game Overview Sheet:



Stalagmic is a singing based platformer that provides a musical experience like no other. Players are tasked with singing desired notes while balancing platformer mechanics, in order to advance through an immersive cave-themed environment. Through the use of color symbolism and dynamic music accompaniment, players will finish Stalagmic with a stronger grasp of ear training, learning in an entirely gamified manner!

Developed in Unreal Engine

All work Copyright ©2020 by

Julia Krawiec, Artist, krawij@rpi.edu

Tatum Hobby, Artist, hobbyt@rpi.edu

Andrew Sirkoch, Programmer, sirkoa@rpi.edu

Xavier Marshall, Programmer, marshx@rpi.edu

Hibiki Takaku, Programmer, takakh@rpi.edu

Greg Sirkoch, Musician, sirkog@rpi.edu

Game Philosophy

Why create this game?

Our game provides the player with a one of a kind pitch-based audio experience like no other. With fun puzzle design and stunning graphics, Stalagmic manages to subconsciously ear train its players through the means of color symbolism and subtle audible cues.

Immediate and Long Term Projected Socio-Cultural Impact?

This game has the potential to completely shake up the audio-dependant video game market. By using audio detection algorithms in a new and unique way, Stalagmic is redefining the limits of volume and pitch detection as a gameplay controller.

Predecessor / Competitors?

Wandersong is a 2D platformer where the character can sing to navigate their environment. The character could sing by dragging the mouse towards a color on the color ring of the user interface. The main difference is that the singing aspect was controlled by a mouse. What's unique about our game is that we actually use the microphone input to control the color that we produce. Our group has also done research on historical rhythm games like many Harmonix titles and some Japanese exclusive games. We've also looked at historical karaoke games and other mic input games like Hey You Pikachu! as further reference.

Reference Links:

<https://artsandculture.google.com/story/the-sound-of-colours/0ALymHuhPI1jLg>

<https://arstechnica.com/gaming/2009/03/ne-music-game-feature/>

<https://store.steampowered.com/app/530320/Wandersong/>

https://store.steampowered.com/app/893720/One_Hand_Clapping/

<https://www.pokemon.com/us/pokemon-video-games/hey-you-pikachu/>

Target Audience?

The game is being made for ages 10 and above and generally family-friendly. We plan on targeting players that at least have the ability to retain a pitch to trained singers.

Game Overview

Common Questions

What is the game?

In the game, the player solves a variety of platforming puzzles through voice input. The player will need to match a pitch to a corresponding object to activate and/or move it to progress.

Where does the game take place?

A cave system in a planet.

What do the players do?

The player has the ability to move their character freely within the bounds of their environment and jump on platforms. When the player sings into their microphone, the pitch of their audio will be determined and affect objects in the environment within a given range.

How many characters are involved?

A player character and their travel companions, light fairies.

What is the main focus?

The main focus is to allow sounds from the player to become an interactable medium within the game.

Design History

This is a brief explanation of the history of this document.

In this paragraph describe to the reader what you are trying to achieve with the design history. It is possible that they don't know what this is for and you need to explain it to them.

Version 1.00

Version 1.0 is the original version of the advanced game design document. Everything was filled out as a rough draft. Initial core mechanic ideas and environments were brainstormed.



Mission: Light Up The Labyrinth



Make noises to activate potential light sources

Narrative Idea:

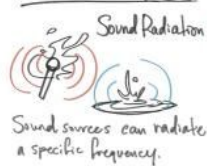
- Player is locked away from the outside.
- You're given a chance for freedom if you complete the trials aka puzzles/tasks successfully.
- ↳ Main goal: Explore and light up all areas

Mechanic Idea: Negative sound kills you

- Anything with a sound source emits light.
- Negative sound source is generated by certain objects/areas/enemies
- ↳ Stay in there too long and you die.
- Light does not matter, only the sound source
- ↳ Puzzles that rely only on mic input sounds as a temporary light source.



Puzzle Mechanic Ideas



Aesthetic Ideas



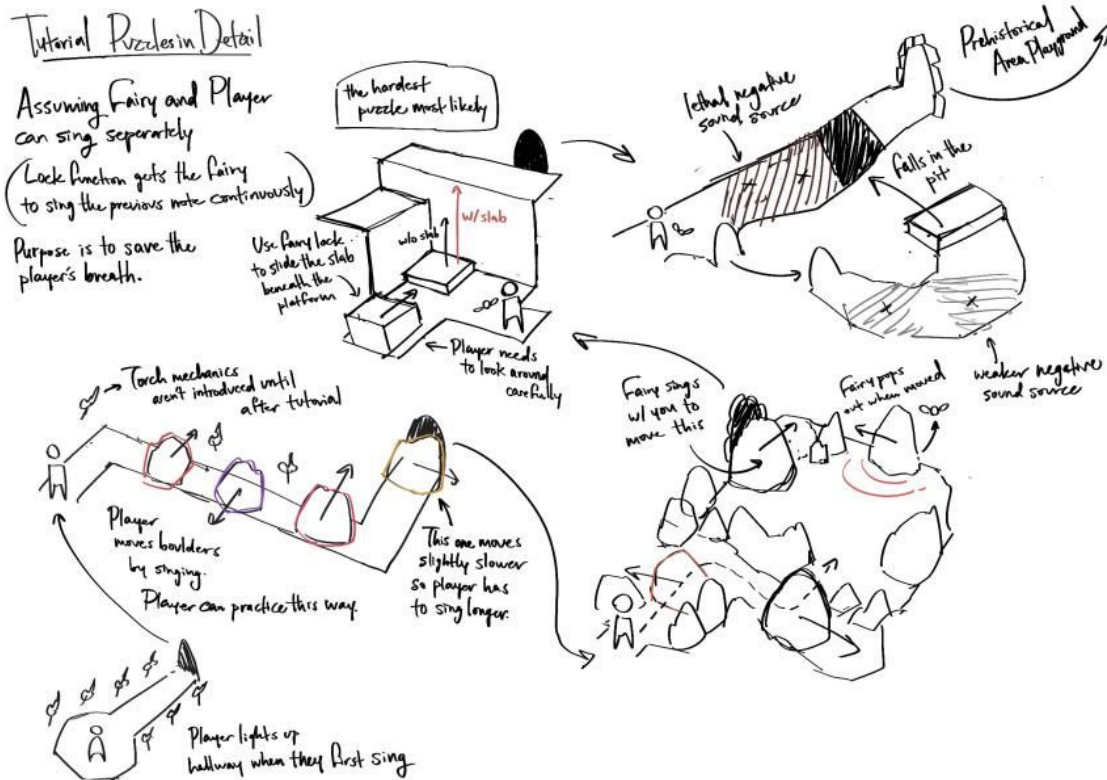
Version 2.00

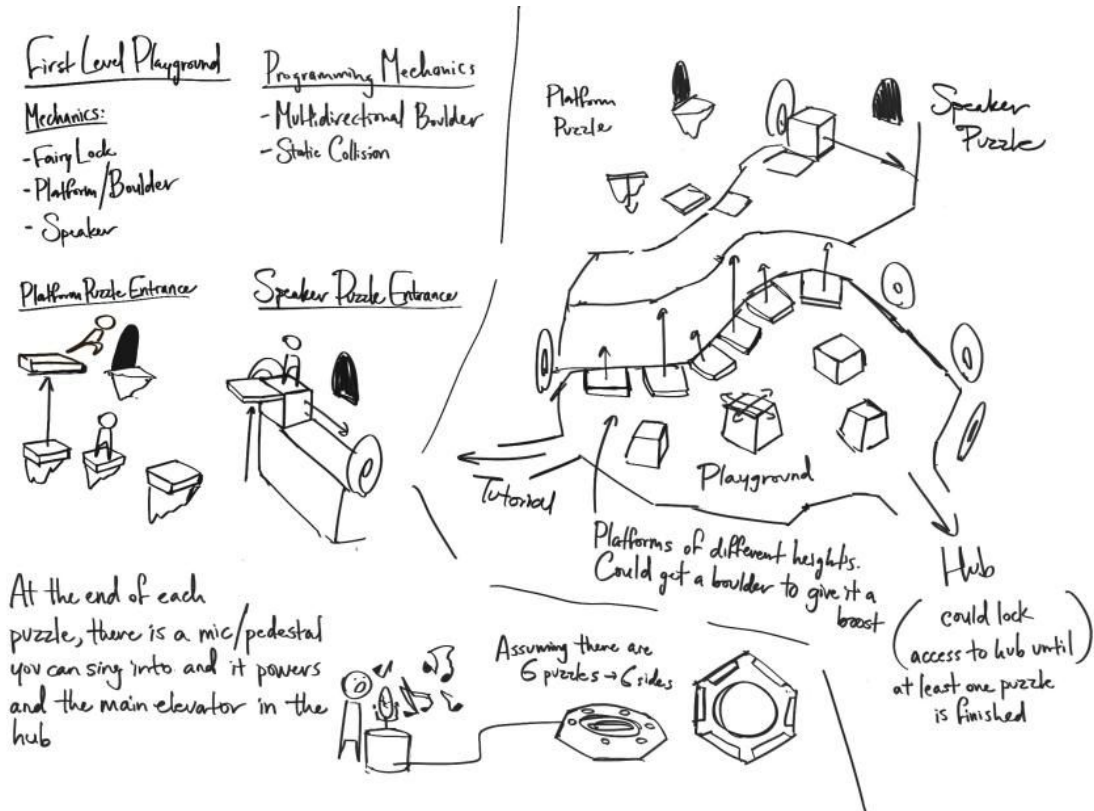
Version 2.00 has more in depth information and screenshots based on our 1st playable prototype

Included in the changes are:

1. Scope brought down to include 2 level environments
2. Screenshot of the prototype included
3. Story restructured to fit gameplay
4. Added a little bit on research for predecessor games

Level design concepts:





Version 2.30

Version 2.30 has had some more refined gameplay added and a concrete vision of the game's art style.

Included in the changes are:

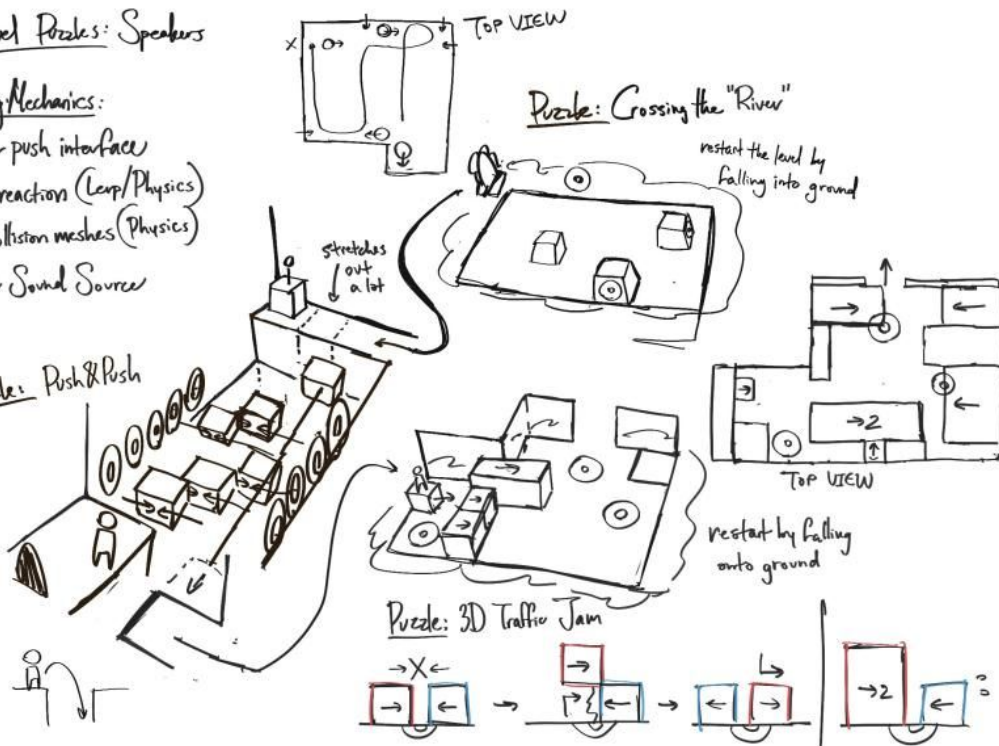
1. More concept pieces to describe gameplay and art style
2. Some edits on the descriptions of the puzzle pieces involved in gameplay
3. Current UI screenshot and description added

First Level Puzzles: Speakers

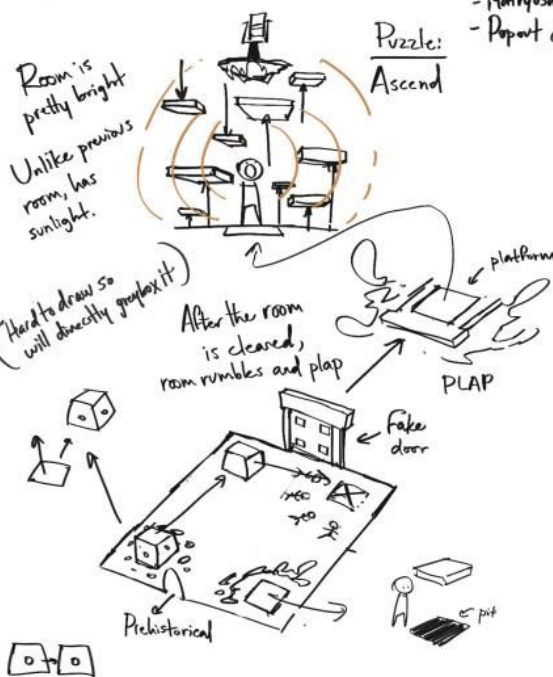
Programming Mechanics:

- Speaker push interface
- Gravity reaction (Leap/Physics)
- Static collision meshes (Physics)
- Negative Sound Source

Puzzle: Push & Push



First Level Puzzles: Platform



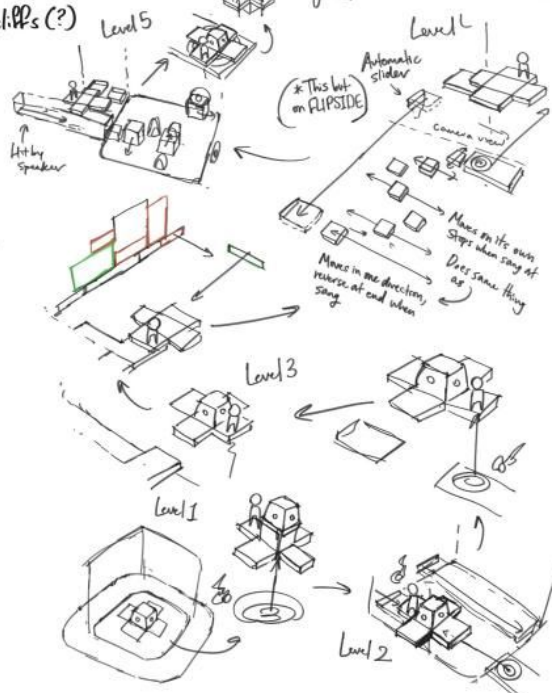
Prog. Mechanics

- Matryoshka columns
- Popout clips (?)

Puzzle: Descend

No torches, dark room

Objective: Bring block up to the top by singing to the surrounding objects.



Version 2.50

Version 2.50 has had some updates to the character design, some updates to gameplay descriptions

Included in the changes are:

1. Updated character concepts
2. Minor gameplay description edits
3. Added a title!!!!

Version 2.75

Version 2.75 has an updated game overview sheet and some new screenshots from updated game assets

Included in the changes are:

1. New screenshot and edited description in the game overview sheets
2. Title changed from “Cave Rock” to “Stalagmic”
3. Added light fairy model screenshot
4. Changes were made to the level design according to playtesting feedback

Version 3.00

Version 3.00 has all final updates to the game overview sheet that reflect the final version of the game before official release

Included in the changes are:

1. Updated screenshot of UI to have the new environment assets instead of placeholder geometry
2. Added screenshot of the player character in engine to reflect the finished model, textures, and animations
3. Added a third character detail section to talk about the fairy character and updated the screenshot(s) of the fairy model in engine to show the dynamic color changing texture and animations
4. Replaced the picture in representing the world layout with a description of the current game layout/level design
5. Added an asset list for all environment pieces and included pictures for reference

Feature Set

General Features

- Isometric Map
- 3D graphics
- Variety of puzzles
- Multiple Cave Themed Environments
- Stylized Art
- Voice Control [pitch and volume]

Gameplay

- User Voice Input
- User Pitch Detection
- User Volume Detection

The Game World

Overview

A lone cave explorer finds themselves lost in a cave system. It is dark and hard to see, but they stumble upon light fairies in the cave that help them traverse the area through the power of song.

This cave system is sprawling with mysterious artifacts that activate according to what pitch you sing. Beware, though, there are also corrupted artifacts that can hurt you throughout your journey. You must now find the source of the corruption and bring an end to it.

Prehistoric Section

Overview

The player's starting location and the first section that the player visits. This is where the player starts their journey and meets their fairy ally.

Objects

Torches, boulders, platforms





The Physical World

Key Locations

Primordial/Prehistoric Section

Travel

The player will move around through keyboard or controller input. The world can be somewhat manipulated through voice input

Scale

The character will be relatively small on the screen which will allow for a large amount of the environment to be showing. The size of the environment pieces compared to the character will convey the sense of scale to the player.

Puzzle Mechanics

For every puzzle mechanic, the object changes behavior when the player/fairy sings to it. Most objects will move when the player sings.

One directional objects - Objects that can be pushed in one direction for however long the player sings. Boulders are mainly one directional objects.

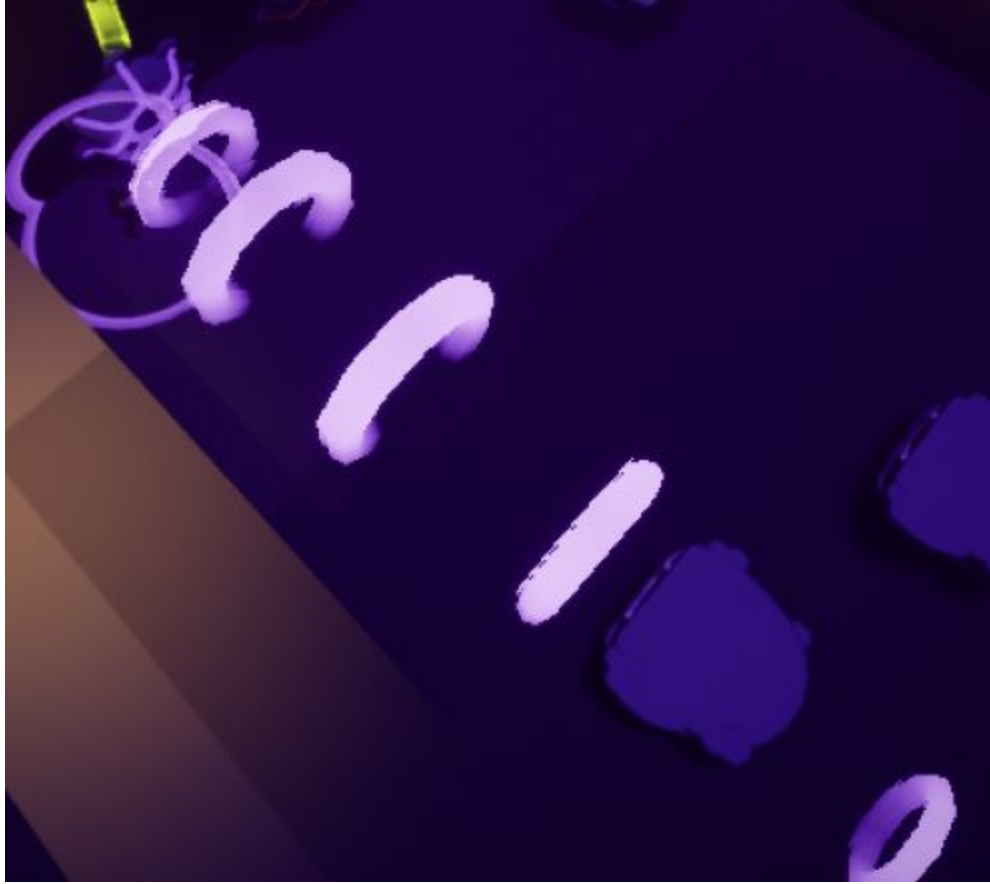


One direction with limit objects - Objects that can be pushed by singing in one direction until they hit a certain distance. These objects are mainly boulders as well.

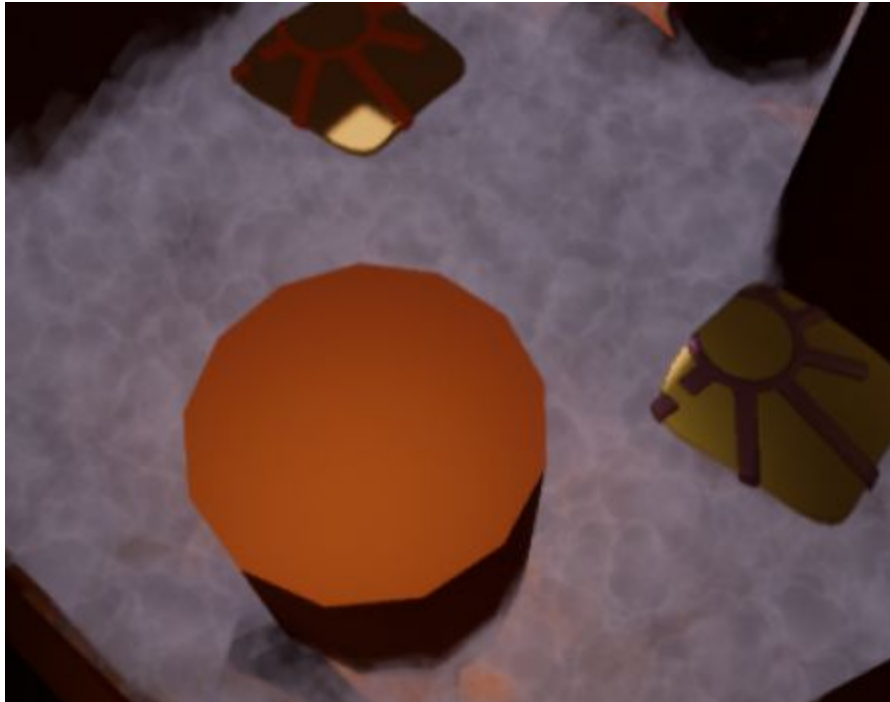
Platform - Objects that will move to a certain position while the player sings to it, but will move back to its original position when idle. These objects are mainly platforms.



Speaker - Objects that will push an object in front of it towards the current direction the speaker itself is facing when the player sings.

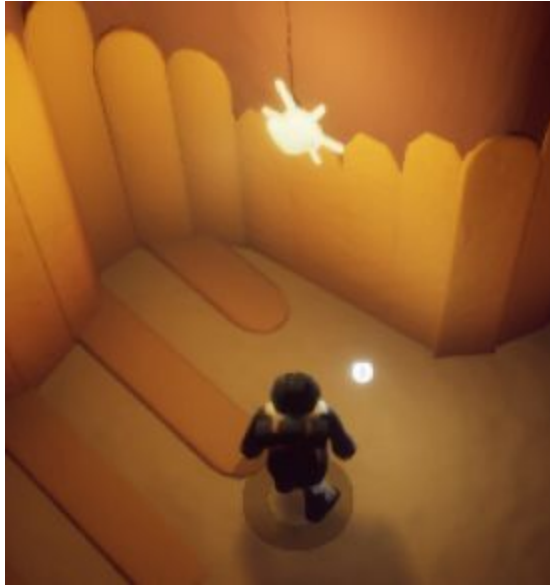


Fast-slow platform - Platforms that automatically move back and forth between locations while idle. While the player sings to it, it slows down.



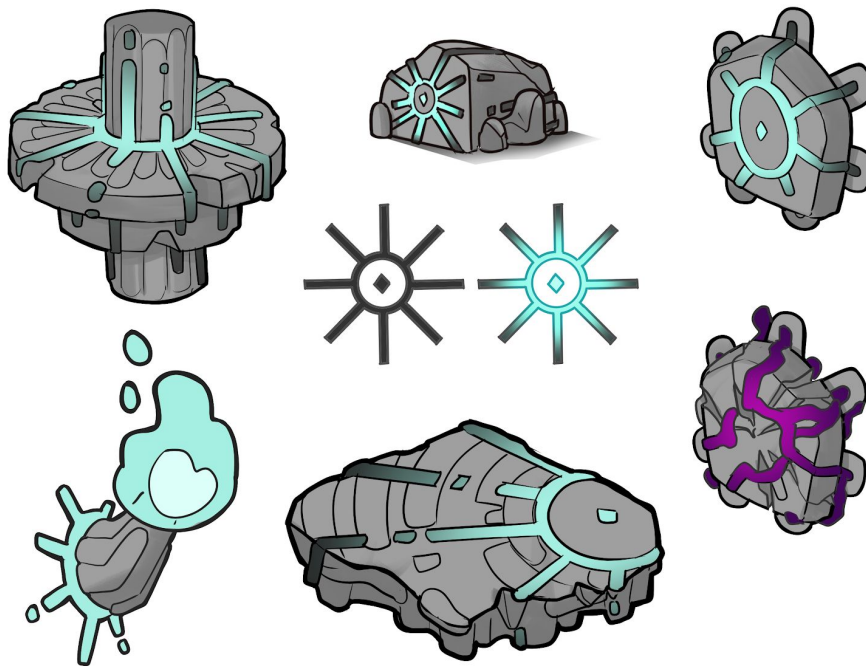
Objects

Torch - An object which will light an area of the environment when the player sings to it.



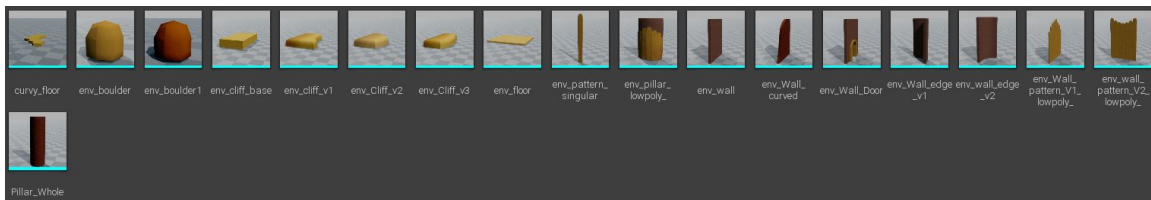
Negative Sound Source - An area of fog where the player will lose health while exposed to it. Depending on the area, the player can disperse the fog by singing.





Asset List

Cave Assets:



Names from Left to Right-

Curvy Floor, Boulder, Cliff Base, Cliff Version 1, Cliff Version 2, Cliff Version 3, Floor, Wall Decor, Pillar Decoration, Wall, Curved Wall, Wall Door, Wall Edge 1, Wall Edge 2, Wall Pattern Version 1, Wall Pattern Version 2, Pillar Platform

Puzzle Assets:



Names from Left to Right-

Column, Floor Glow, Torus (Used for VFX), Gem Stone, Long Platform, Negative Speaker, Regular Platform, Push Boulder, Speaker, Torch

Rendering System

Overview

The game will be a stylized 3D game with strong color palettes. All the environments will be cave themed.

2D/3D Rendering

Unreal Engine 4.25.3

Camera

Overview

Top down 45 degree isometric view

The camera system contains a free floating camera that smoothly follows a target point at a variable zoom but fixed angle. The camera will adjust distance from the target point according to the environment that the camera is in. For example, if the camera is following the player and the player enters a new chamber that is rather large, the camera will be triggered to zoom out to view the entire area. Upon entering a more claustrophobic area, the camera would zoom in.

The camera's target will be the player 99% of the time.

Target Switching

When the player needs to be informed about the existence of an object that is currently too far to see, or too far to zoom out easily, then the camera will switch to monitor that target briefly. This can be imagined as a quick pan to the new target, waiting for a couple seconds, then switching back to the player as the target.

Zoom out to encompass all active light fairies

When the player decides to lock in a light fairies position, and then correspondingly walks out of view of the light fairy, the camera will zoom out/in to always keep the fairy visible on screen.

Game Engine

Overview

We will be using Unreal Engine.

Specifically so we can utilize its robust lighting system and take advantage of its impressive performance when it comes to algorithm heavy games.

Game Engine Detail #1

The game engine will keep track of everything in the world, such as our art assets, lights, VFX, animations, SFX, music, and more.

Collision Detection

Objects in the environment will react to the player's singing when they are within a certain radius of the player.

Various tools equipped with trigger collision zones will be used to manage camera systems and player visibility

Lighting Models

Overview

Given the cave setting we will be making extensive use of Unreal's real time dynamic lighting to produce high contrast effects in dark environments.

The player's singing

As the player sings, their corresponding light fairy will begin to emit a light that corresponds to the pitch that they are singing. This light will contribute to the lighting model of the surrounding environment.

The World Layout

Overview

Central Hub: Player can test out different objects and how they react to sound in a safe environment

Tutorial: Intro to the game where player gets introduced to the basic concepts

Platform Level: A platform heavy level

Speaker Level: A speaker heavy level

Game Characters

Overview

Player is a singer/noise maker with special vocal abilities that influence this cave environment.

Fairy is an ally which assists the player in navigating through levels.

Enemies and Monsters

There are areas of negative influence called negative sound source areas. If the player is exposed to the area for a duration, it kills the player. There are two kinds: regular areas are areas where the player can mitigate exposure by singing. The strong area is where singing will do nothing to prevent the player from being exposed.

User Interface / Experience

Overview

Most of the user's experience will be informed by physical world visuals rather than heads up display graphics. The game will however contain minimal HUD graphics for communicating which note they are singing (in typical Western notation) portrayed through a gradient colored slider located at the edge of the screen.

User Interface Detail #1



UI slider indicates what pitch the player is singing at and corresponds the note to the color of the object

Musical Scores and Sound Effects

Overview

Synthesizer effects on game objects that are interactable with singing. Main theme is composed of spacey synths. Button effect is hollow and bell-like. Atmospheric sounds are cavey and wet.

Single-Player Game

Story

The game takes place in caverns as your character explores them. Through their journey, they find light fairies and embark on a quest to return the core light to the center of the cavern, otherwise corrupted artifacts might destroy the caves.

Hours of Gameplay

The game should take about an hour to 2 hours to complete

Victory Conditions

The player completes all the levels and brings the light fairies to the final cavern.

Single-Player Game

Story

The game takes place in caverns as your character explores them. Through their journey, they find light fairies and embark on a quest to return the core light to the center of the cavern, otherwise corrupted artifacts might destroy the caves.

Hours of Gameplay

The game should take about an hour to 2 hours to complete

Victory Conditions

The player completes all the levels and brings the light fairies to the final cavern.

Character Rendering

Overview

The character will be rendered in 3D in Unreal. They will have a light emanating from them when they make sound in the mic.

Character Rendering Detail #1

The character will be animated and have an idle, walk, run, jump, and sing animation

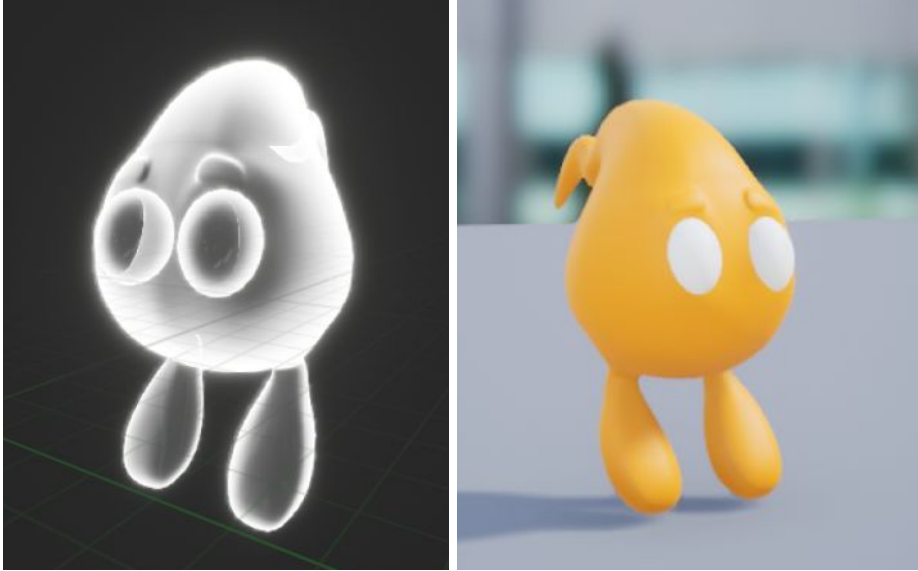
Character Rendering Detail #2

The character will have different material textures that will look fabulous in our lighting



Character Rendering Detail #3

The character will have a fairy companion that will resonate with the audio inputs passed through the player's microphone. The fairy's textures/colors will change dynamically depending on the note frequency it is omitting.



User Testing

Overview

Our most rigorous testing sessions occurred during the title's First Playable stage, where the core mechanics had been largely finished in development but no final art assets were in use. Level designs were rough, but approaching finality. Four different college age (19-21) testers were chosen each with a varied level of prior musical experience. Among the testers with experience, one actively played the Cello and the other had not played since High School. Among the other two, one enjoyed singing casually and the other had poor singing abilities. No testers had formal pitch training, and none were known to be tone deaf. Session length was approximately 45 minutes per person.

Essential Observations and Responses

The majority of gameplay was relatively low action until the final level, however testers reported consistently high levels of engagement in post-session interviews, citing the singing as the biggest contributor. Each tester began with a level of note accuracy matching their relative level of musical experience, but they all managed to more accurately and quickly hit notes the longer they played, which re-affirms our core objective of gamifying pitch training. Most players felt the balance between singing and platforming was well managed in most instances and were able to use the fairy to define their playstyle. The testers who struggled to hold notes for more than a few moments made liberal use of the fairy to help manage their platforming while those who excelled in both used the fairy less. These stylistic differences highlight just how important the fairy was to expressions of player agency.

Among the issues noted was that players often lacked a clear sense of direction in levels beyond the tutorial and final level. This feedback led to us linearizing the structure of the second level and adding lighting and asset indicators for the end of levels. Players also struggled to predict which direction objects would move in when singing to them, which prompted the creation of arrow indicators to convey movement direction. One of the low-experience testers became very quickly frustrated with the isometric platforming as early as the tutorial. In order to be more accommodating to these types of players, we reduced the necessary accuracy to pass the tutorial's platforming challenge. Additionally, we better leveraged checkpoints in the final level to prevent players from getting frustrated due to small platforming mistakes.