`

*Who's Refuge?*

## Team Graph Giraffes

### Designed by Adam Gazdecki, Cole Lewis, and Kora Loudermilk

# Table of Contents

**1.0 Chapter One**…………………………………………………………………………………………3

1.1 Overview……………………………………………………………………………………..3

1.2 Gameplay…………………………………………………………………………………....3

1.3 Detailed Features and Mechanics……………….…………………………...…………..4

1.3.1 Interactions and Rules…………………………………………………………..4

1.3.2 Objectives……………………………………………………………………...…4

1.3.3 Conflict…………………………………………………………………………....4

1.3.4 Menus and Interfaces…………………………………………………………...5

1.4 Story………………………………………………………………………………………….8

1.5 Level Design………………………………………………………………………………...8

1.5.1 Area 1 - Halcyon Tower…………………………………….…………...……….9

1.5.2 Area 2 - Sonorous Swamp……………………………………………………..11

1.5.3 Area 3 - Fairbright Maze………………………………………..………………11

**2.0 Chapter Two**………………………………………………………………………………………..14

2.1 Game Object and Components…………………………………………………………..14

2.2 Assets……………………………………………………………………………………….16

2.2.1 Visual……………………………………………………………………………..16

2.2.2 Audio……………………………………………………………………………..17

2.2.3 Narrative…………………………………………………………………………17

**3.0 Chapter Three**……………………………………………………………………………………..19

3.1 Dev Update 1 Milestone………………………………………………………………….19

3.2 Dev Update 2 Milestone………………………………………………………………….19

3.3 Dev Update 3 Milestone………………………………………………………………….19

3.4 Proof of Concept Milestone………………………………………………………………20

3.5 November 25th Milestone………………………………………………………………..20

3.6 December 2nd Milestone…………………………………………………………………20

3.7 Final Exam Milestone……………………………………………………………………..20

3.8 QA Testing Strategy and Schedule………………………………………………………20

3.8.1 Strategy………………………………………………………………………….20

3.8.2 Schedule………………………………………………………………………...21

# 1.0 Chapter One - Game Design Document

## 1.1 Overview

*Who's Refuge?* is an exploration-adventure game in which the player navigates an environment and solves puzzles around the world in order to progress. This game draws inspiration from games such as The Legend of Zelda and Myst. The core mechanics include utilizing visual hints from differing environments; exploring the island in the pursuit of answers to mysteries; solving puzzles to access new areas and items; and seeking and thinking about helpful items.

After waking on a strange island, you emerge to find yourself alone, surrounded by mysterious structures and items left behind from people long gone. What was left points to a strange door on the island with three key slots that leads to nowhere. Follow the signs to collect all three keys and solve the mystery of the barren island!

## 1.2 Gameplay

The player will need to complete three separate puzzle chains in different areas. Each chain gives the player access to a key, in order to finish the game. Each area will feature a separate and unique environment that requires the player to observe their surroundings for information and clues, utilize different movement mechanics, recognize new patterns that arise from each area, and experiment with interactable objects. The completion and gain of each key can be done in any order, although some key quest areas may require the player to have found an item located elsewhere on the island. We want the player to have non-linear access to the puzzle chains so that they are open to picking and choosing their challenges. The world’s non-linearity encourages the player to take their own unique approach to traversing the world and optimization over multiple playthroughs.

In the course of each puzzle, the player will be asked to manipulate the environment in the course of some puzzles by means of switches, matching items, or pressing buttons in a certain sequence. Puzzles may also include enemy interactions in which you’re required to defeat them in a special sequence or otherwise.

The gameplay loop will consist of

1. The player explores the map.
2. A puzzle is found by the player from visual clues and the environment.
3. They formulate a solution and attempt it. They succeed, retry the puzzle, or decide to go back to Step 1.
4. Player is presented with a reward and returns back to Step 1.

And so on until all three keys are collected.

On the map will also be hidden collectables that will reward the player for exploration. This will encourage the player to take their time within step 1 of the gameplay loop. The collectables will not offer a tangible reward to the player other than the satisfaction of their collection and the drive to fully complete the game.

Alongside collectable items present in the world are pieces of equipment that some puzzles may rely upon, remaining unsolvable unless the corresponding item is equipped. Equipment can be purchased using coins collected by the player throughout the world.

## 1.3 Detailed Features and Mechanics

### 1.3.1 Interactions and Rules

Player will move along three axes of movement using the W, A, S, and D keys to move forward, left, backwards, and right respectively, then using the spacebar to jump. Using the Ctrl key, the player can toggle running and will walk by default. The player will interact with items and specific environmental objects with the Q key. Items in the inventory are selected using the arrow keys, the selected item in inventory can be used with the E key. The camera view will be manipulated using the mouse and will view the player from a third-person perspective.

The player can walk, run, jump, collect objects, and equip one item from their inventory at any given time.

The player can not teleport or fast travel, remove objects from inventory until opening the final door, or swim in the water beyond the island.

### 1.3.2 Objectives

*Who’s Refuge?* has only one overall objective: collect all three keys across the island to unlock the door in the middle and find where everybody has gone. Each key can constitute its own objective in the area that contains it.

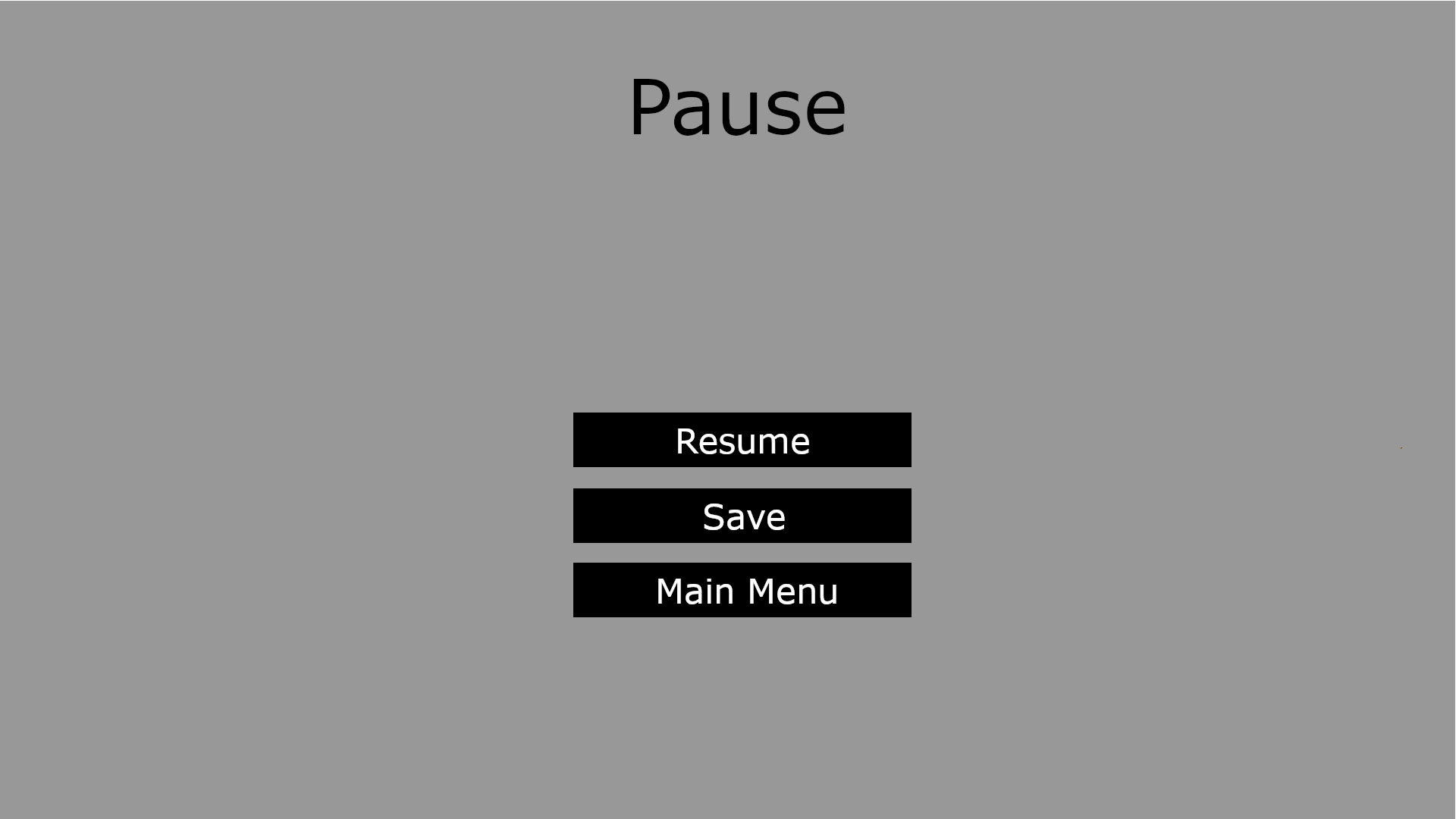
### 1.3.3 Conflict

The player encounters conflict between them and reaching the final goal through the puzzles and impediments they need to solve, within those are enemies which the player needs to contend with.

### 1.3.4 Menus and Interfaces

#### Pause Menu

Includes buttons to resume the game, save the game, and quit to the menu. In game, the background will be a low-opacity blur over the background of the game. During pause, all movement will cease until resumed.



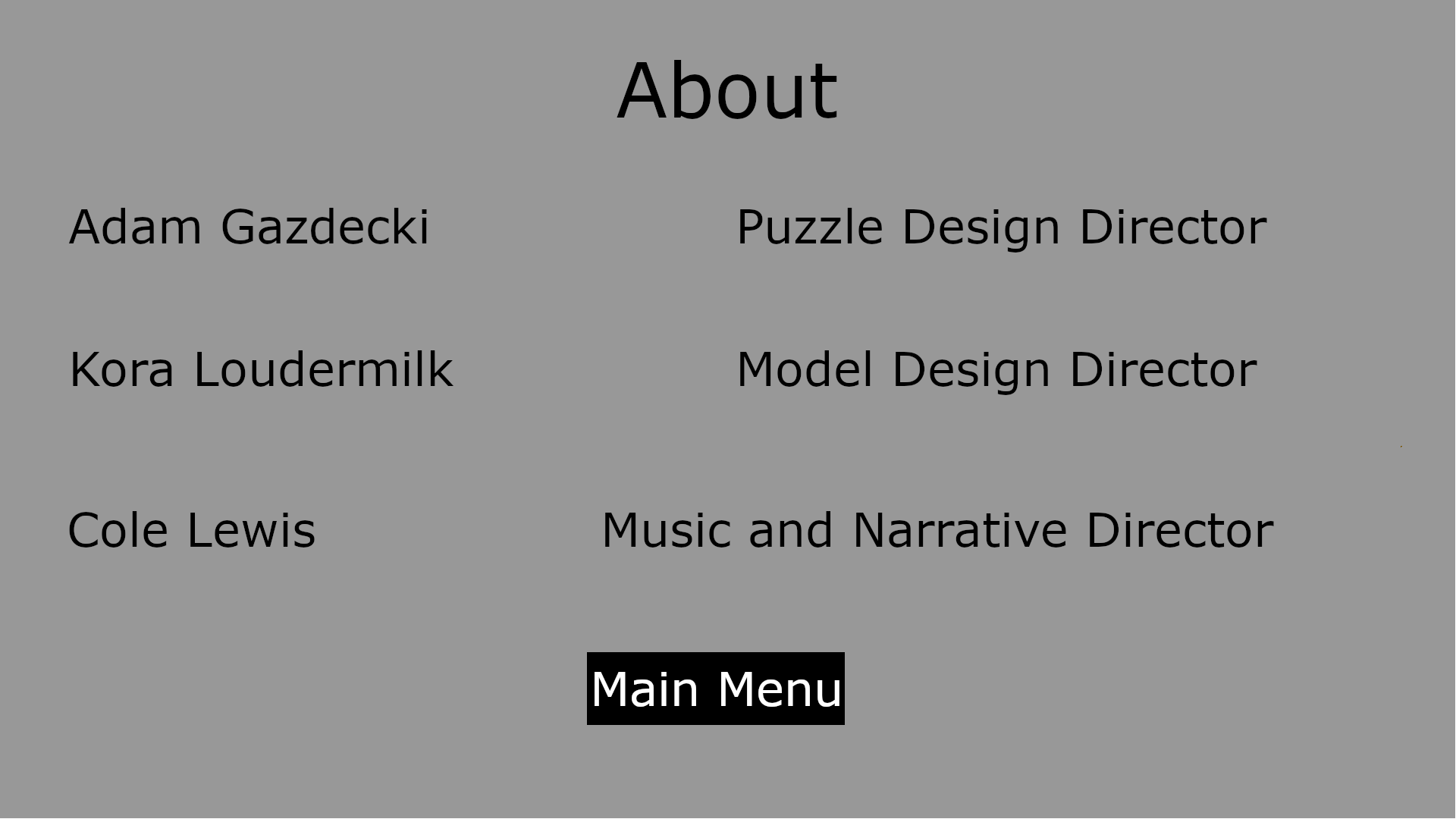
#### Title Screen/Main Menu

Includes buttons to play the game, quit to desktop, open the about menu, and open the how to play menu, and load a saved game. While the below image shows a gray background, we plan to loop a video of the island from a distance, see [here](https://tcrf.net/images/b/ba/WiiSportsResort-title.png) for an example of how this may look. Buttons will be translucent and rounded over the shot of the island upon implementation.



#### About Menu

Includes a button that allows the player to return back to the title screen. Menu contains the game name, the team member’s names, and their roles. The background will be a low-opacity, translucent layer over looped footage of the island as described in the title screen, the button to return will be styled similarly.



#### How to Play Menu

Includes a button that allows the player to return back to the title screen. Informs the player of the controls and the objective of the game.



#### Game HUD

Active during all gameplay except for when superseded by the pause screen, will display health, collected items, selected active item, and how many keys the player has, an empty outline indicating that key is not collected. The background is gray as a stand in for game content. The currently selected item will have golden corners around the item frame, the items shown below are placeholders and may or may not reflect actual in-game items.



### 1.4 Story

The narrative in *Who’s Refuge?* is scattered throughout the world and left to the player to learn as they explore the island; the level of immersion in the narrative is dependent upon the player’s level of exploration within the world.

The player awakes on an island scattered with the belongings and structures of people who had once inhabited the island as a refuge from an unknown calamity. As the player explores the island, they find whispers of the island’s history such as stray pages from a journal or scrawled warnings left ominously on the walls of decrepit buildings. The player comes to learn that those on the island had escaped a deadly mutagen that had spread across the Earth and turned people into monsters who did nothing but infect and destroy. Eventually, the island was not even safe anymore as they were closed in on by the monsters, enabled by constant evolution to find them even in the ocean. The door in the middle of the island leads to the final refuge where the monsters can not reach them, however, it was left only able to be opened by three keys and solving the puzzles keeping them locked, to safeguard against the monsters attaining intelligence.

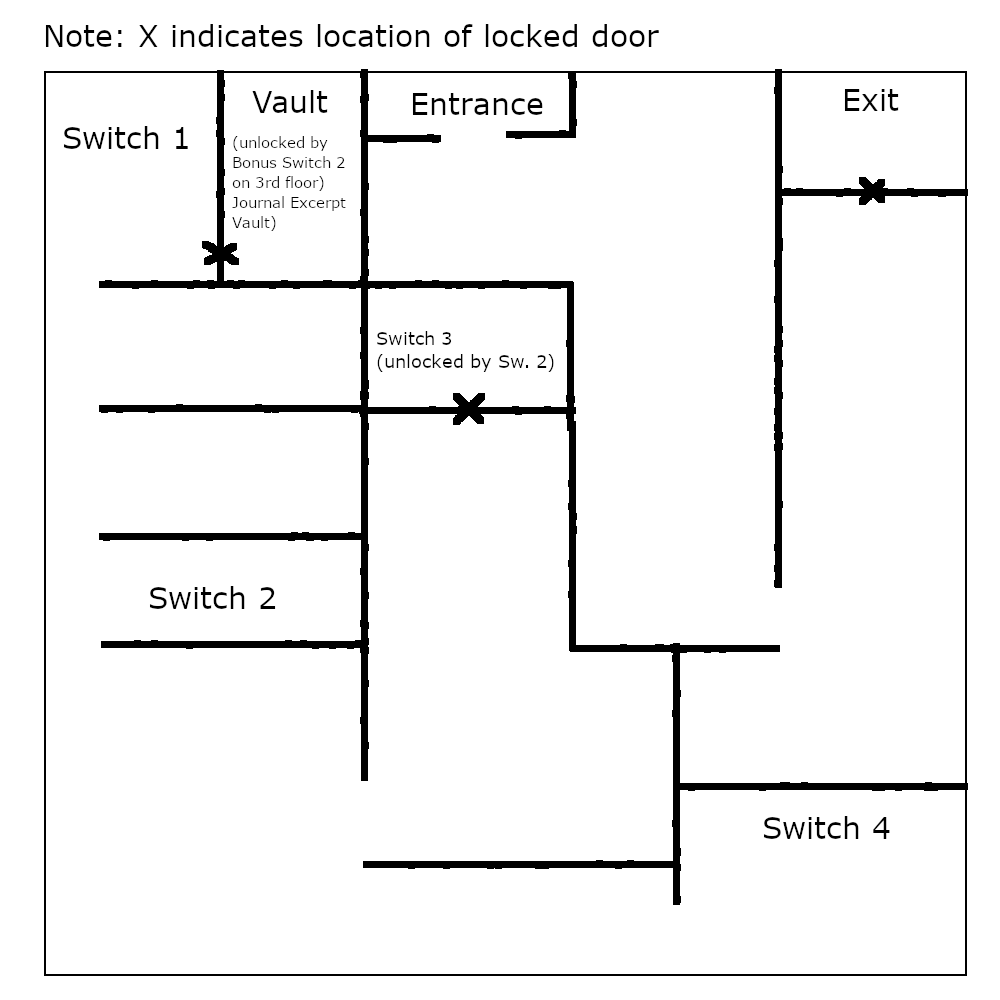
### 1.5 Level Design

The game prominently features three areas of interest, one for each key. These consist of a distinct puzzle chain the player is required to traverse and complete in order to gain possession of each area’s respective key. Each area will have a unique theme or motif that distinguishes it from other areas and the island at large. These can be considered “dungeons” or “levels”.

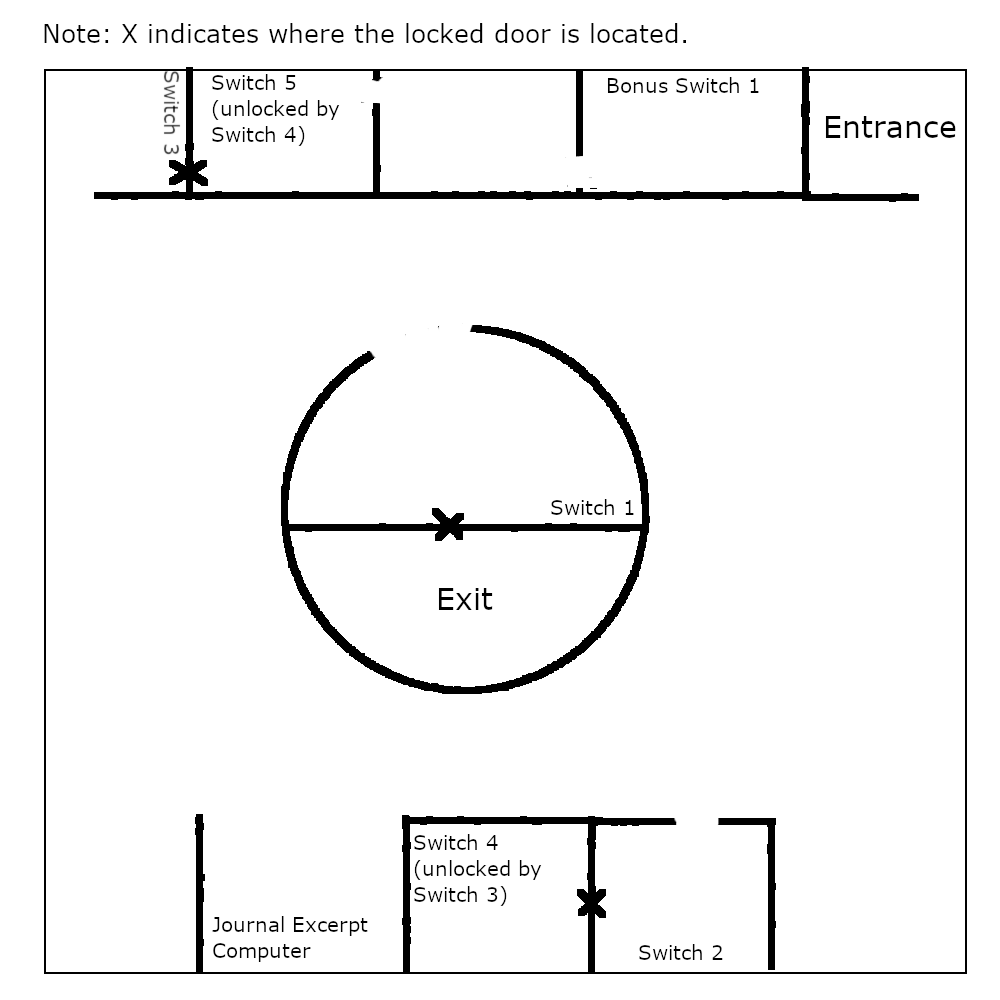
#### 1.5.1 Area 1 - Halcyon Tower

Halcyon Tower will have three floors the player is required to traverse. In order to complete each floor, the player must toggle a series of switches in the correct order, at which point they can advance. Below is the floor plan for each level of the tower with switches, numbered in sequential order, and the entrance and exits. The closed rooms indicate locked doors, the exit door always opens once the highest numbered switch is toggled on and always leads to the next floor.

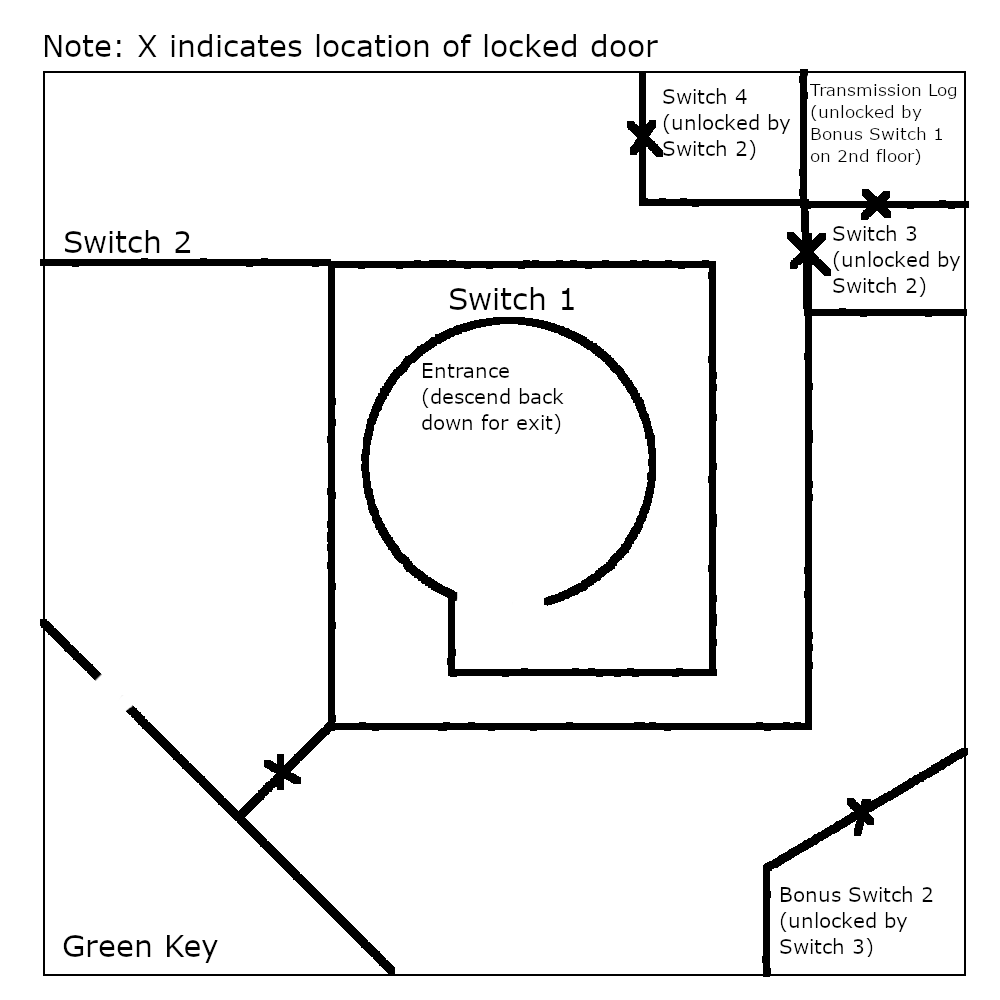
##### Floor 1:



##### Floor 2:

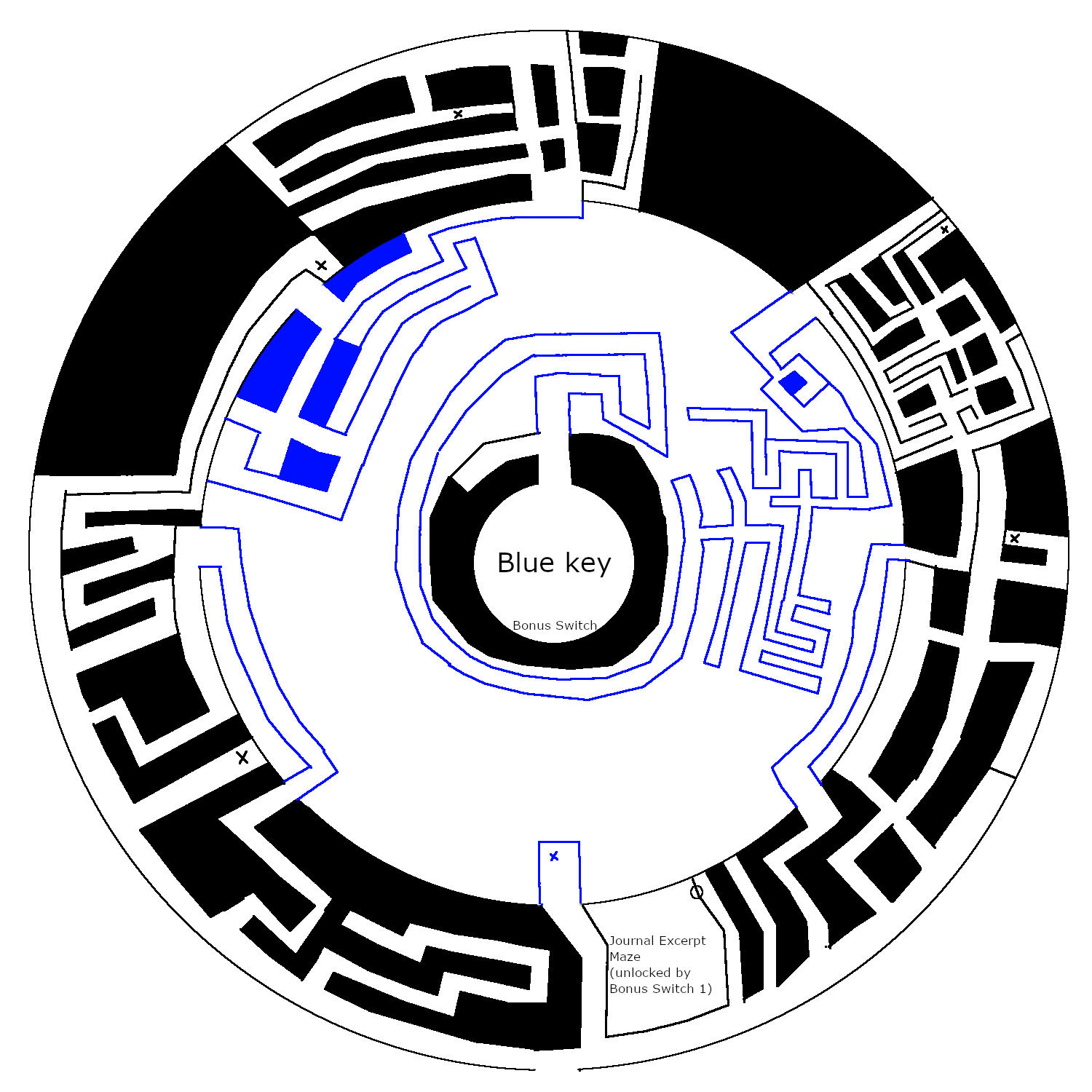


##### Floor 3:



#### 1.5.2 Area 2 - Fairbright Maze (Name Pending)

The central idea of this puzzle area is a maze made of two concentric circles. The outer ring remains static while the inner ring rotates 45 degrees clockwise on each activation of a switch, marked by an X. The solid colored areas represent empty space. Unfortunately the top-down diagram below is crude and non-proportional, which doesn’t do the idea justice until implemented in game.



#### 1.5.3 Area 3 - Perilous Plain (Name Pending)

Perilous Plain (Name Pending) will consist of an environment designed around a puzzle chain. A progressing set of puzzles that requires the player to question their assumptions and combine obvious mechanics into non-obvious combinational mechanics in order to progress. This chain creates the foundation for three dimensional parkour challenges. We may remove or add steps depending on the underlying mechanics’ functionality and how challenging it is for the player to figure out. This chain will remain independent from all items and knowledge gained in other areas of the island. A final key challenging will be laying these out and restricting access to a semi-linearly fashion.

##### Puzzle One

Obvious Mechanic: There’s a stick on the ground and the player can “stab” or “swing” with it. This low level mechanic should be explained in both the “How to Play” and in the tutorial. Or even further this should simply be told to the player in some way.

Obvious Mechanic: There is a giant crab. This crab will be presented to the player and other visual clues will attempt to make the player curious or at least well aware of this crab.

Non-Obvious Mechanic: This is the new ability and creative thinking that the player learns on their own. It should answer the internal question of “what do I do with this?”. Using visual clues from the environment like having the crab appear when the stick is picked up, or laying objects out in vicinity of each other, we will attempt to encourage the player to poke the crab. This is the solution to the puzzle. Rather than telling the player they are able to do this, we want to create a sense of outside the box thinking and discovery of this solution. Furthermore, this should give the player a new ability, some new piece of knowledge they can hold on to. The player can now effectively move the crabs wherever they want using their stick.

##### Puzzle Two

Obvious Mechanic: The giant crab will pinch the player and knock them back. This will use a visual cue like a large or sharp claw, or maybe an idle animation that would make the player weary of approaching.

Obvious Mechanic: The player is able to jump onto things in the environment. This is a fundamental that will be simply told to the player in the tutorial.

Non-Obvious Mechanic: If the player falls directly on top of the crab, they will not only be knocked back, but also be boosted upward in a controllable arc. Now they are able to reach higher places than with their usual jumping.

##### Puzzle Three

Obvious Mechanics: The player sees square holes and square rocks. Maybe instead this will be a key and a keyhole, but it should be some cue that makes the player want to retrieve an item. The player is also able to pick stuff up and put stuff down just like their stick.

Wrong-Assumption: We want the player to make an assumption based on their knowledge from the previous puzzles. We want them to have a clear objective and immediate idea of how to reach that goal. They should think to themselves “I’ll crab jump up there to get that square rock. Easy.”  
The Catch, A Non-Obvious Mechanic: The crab doesn’t want to go where the player tries to poke it. It refuses, fights back, or some other animation to show that the player should give up trying this. There should also be a visual cue as to why. The ground will be obviously different, whether it’s grass instead of sand, or perhaps a bush or spiky bush is in the way. There has to be some explanation to the player and here visual cues are more important than logic.

Non-Obvious Mechanic: Different colors or types of crab only walk on certain ground. You have to use the farther away and less convenient crab to solve this puzzle. This teaches the player a new fundamental rule, crab-ground-exclusivity.

##### Puzzle Four

Obvious Mechanic: The player wants to crab jump somewhere to retrieve an item.

The Catch: The player can’t move the crab exactly where they would like to because of the ground restriction they learned in the previous puzzle.

Non-Obvious Mechanic: The player can bridge larger gaps by falling onto a crab from a greater height. This solution builds the framework for a three dimensional spatial movement puzzle chain. e.g. Parkour. The player has figured out a set of tools and combinational mechanics from a set of simpler obvious mechanics.

##### Puzzle Chain Reflection

This puzzle chain is completely open to criticism and change. We have been trying to learn and plan good design of puzzle games and what makes them fun. The technical challenge of implementation may change some of our baseline or combinational mechanics, but we’re confident in our ability to create new puzzles and steps from the same set of simpler mechanics.

# 

# 2 Chapter Two - Technical Design Document

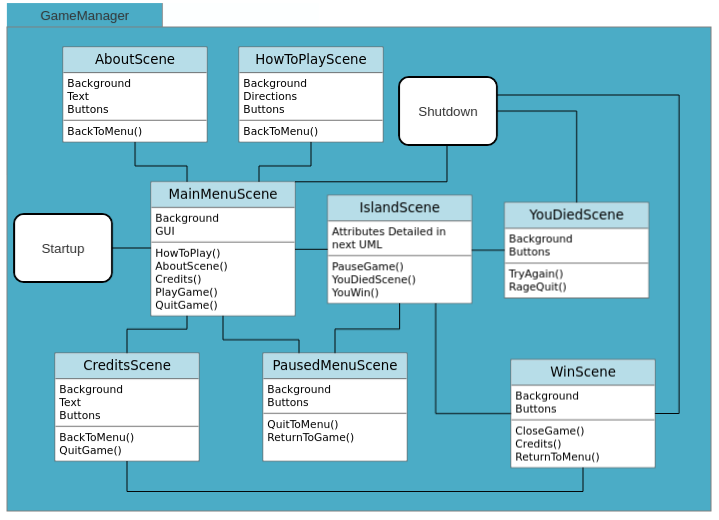
Our team will use the Unity game engine to construct the game and Blender to create a majority, if not all, of the 3D model assets. We’re going to use Unity’s terrain building features to create the island and paint the foliage onto it. Each terrain game object will have foliage painted or placed. Our planned environment and skillset is not large enough to merit procedural generation.

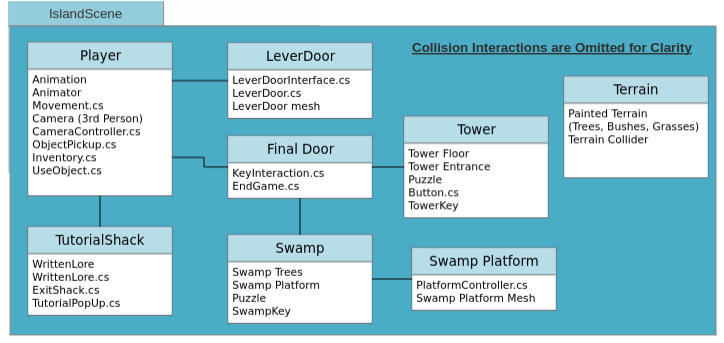
## 2.1 Game Objects and Components

| Player GO | Terrain GO | Tower GO | Swamp GO |
| --- | --- | --- | --- |
| Animation C  Animator GO  Movement.cs C  Camera GO (3rd Person)  CameraController.cs C  ObjectPickup.cs C  Inventory.cs C  UseObject.cs C | Painted Terrain GO  (Trees, Bushes, Grasses)  Terrain Collider C | Tower Floor GO  Tower Entrance GO  Etc (modular pieces)  Puzzle GO  Button.cs C  TowerKey GO | Swamp Trees GO  Swamp Platform GO  Puzzle GO  SwampKey GO |

| LeverDoor GO | Swamp Platform GO | Final Door GO | GameManager GO |
| --- | --- | --- | --- |
| LeverDoorInterface.cs LeverDoor.cs C  LeverDoor mesh GO | PlatformController.cs C  Swamp Platform Mesh GO | KeyInteraction.cs C  EndGame.cs C | MainMenuScene GO  AboutScene GO  HowToPlayScene GO  IslandScene GO  PauseMenuScene GO  YouDiedGUI GO  CreditsScene GO  WinScene GO |

| TutorialShack GO |
| --- |
| WrittenLore GO  WrittenLore.cs C  ExitShack.cs C  TutorialPopUp.cs C |





## 2.2 Assets

### 2.2.1 Visual

* + Trees
  + Swamp trees
  + Swamp shrubs
  + Swamp platforms
  + Grass Texture
  + Sand Texture
  + Shrubs + Bushes
  + Tower wall
  + Tower floor
  + Tower window
  + Tower entrance
  + Picture frames
  + Central Interactable Door
  + Retrievable and Usable Red Key
  + Retrievable and Usable Blue key
  + Retrievable and Usable Green key
  + Desk
  + Chair
  + Decrepit house
  + Interactive Environment Lever
  + Interactive Environment Button
  + Staircase
  + Rubble pile
  + Piece of Paper
    - Paper texture
  + Kettle
  + Pacifier
  + Crib
  + Knife
  + Crab
    - Pinching animation
    - Movement animation
    - Poked by stick animation
  + Stick
    - Poking animation
  + Player
    - Player run animation
    - Player jump animation
    - Player interact animation
    - Player equip animation

### 2.2.2 Audio

* + Ambient tower sound
  + Ambient cave sound
  + Wind sound effect
  + Landing sound
  + Water or wave sound effect

### 2.2.3 Narrative

* Journal pages will be scattered through the world to help tell the story of what had come to happen both before and during the people’s time on the island:
  + Journal Excerpt 1:
    - “Even from a distance we could tell the land was clean. There weren’t the tell-tale signs of corruption: no translucent slick patches on the water or stench in the air. From the moment the island became even a speck on the horizon our whole small fleet erupted in tearful cheers, I grabbed my son close and pointed at the island, telling him ‘That’s our new home!’ He smiled and jumped up and down smiling, even if he didn’t understand yet, because he never knew a world without this evil.”
  + Journal Excerpt 2:
    - “How fortunate we were to land someplace so accommodating and isolated, I can’t think of anywhere more fitting for us to hide while the world crumbles. We’ve been here about a week now, and already split into different teams for fishing, building, gathering, everything we need to do to survive. I’d never seen the stars so bright before, the light pollution in the old city always put a glare over the sky. I think I’ll come to like this place more than anything that had come before.”
  + Journal Excerpt 3:
    - “Day 68 of construction, the Halcyon Tower is nearly half built at this point, even from the second floor we can see the entire island. My hope is that we’ll be done before the rainy season begins. Once we’re finished building, the last thing to do will be hoisting the satellite dish to the roof. Lots of people are saying it’s hopeless, trying to find other survivors, but enough of us have hope left to try, even if it means talking into the void for as long as it takes.”
  + Journal Excerpt Computer:
    - “Just a couple of minutes ago I sent out another transmission across the sea that separates us from the rest of the world. It went the same as any other: I turned on the microphone and told them my name is Lorelei, that I and many others are taking refuge on an island surrounded on all sides by dozens of miles of water. I read out the coordinates of the tower three times in a slow, clear voice before pressing the button on the microphone once more to turn it off. Then I wait, we all wait, for anything. From this height, we know the radio signals reach land, but just barely. I repeat this process four times every hour for six hours every day, three other volunteers cycle through every six hours so that we’re constantly broadcasting. I’ve been doing this for eight months now, taking weeks off here and there to help around the island. Everyone on the island is always so hopeful that we’ll find somebody, anybody. They pat us on the back and congratulate us, but they don’t know the dread that comes with each silence between the transmissions. After long enough behind the microphone, even the most hopeful will start to lose faith that there’s anybody out there at all.”

# 3 Chapter Three - Task List

## 3.1 Due by Dev Update 1, 10/28

1. Design terrain for the island
   1. Place water around the island
   2. Set player boundaries
   3. Place hills, mountains, etc.
2. Handle rendering into the distance
3. Design preliminary environmental assets
4. Begin designing puzzle areas, beginning with the tower
   1. Logic between doors and switches/buttons
   2. Animating doors
   3. Constructing tower prefabricated assets

## 3.2 Due by Dev Update 2, 11/4

1. Implement player model
   1. Rig
   2. Animate
   3. Basic movement scripting
2. Implement player inventory
3. Design player items
4. Create Main Menu with menu options

## 3.3 Due by Dev Update 3, 11/11

1. Continue working on tower puzzle area, start fleshing out other puzzle areas
2. Begin implementing player item functionality
3. Place narrative items
4. Begin designing interactivity system (i.e., player is prompted to interact with an item when in range)

## 3.4 Proof of Concept, 11/18

### Three features due!

1. Finish designing player items
2. Finish concept tower design, continue working on other puzzle areas
3. Flesh out the island by adding contextual structures

## 3.5 Due by 11/25

1. Continue work on puzzle areas, start finishing second puzzle area
2. Begin work on game state serialization
   1. Player position
   2. Inventory
   3. Puzzle state
3. Begin work on third puzzle area

## 3.6 Due by 12/2

1. Continue working on third puzzle area
2. Start working on final door area
3. Compose song
4. Make sound effects

## 3.7 Due by Final Exam, 12/9

1. Make game trailer
2. Write post mortem
3. Complete Component testing
4. Build executable
5. “Last looks” over the entire game, strenuously play test

## 3.8 QA Testing strategy and schedule

#### 3.8.1 Strategy

Each milestone in the task list will go through three phases, pre-alpha, alpha, and beta. The pre-alpha stage will be defined as the period where every major code feature is still being developed. During this stage, minimal testing will be done, with each developer simply testing features to make sure they work correctly, bugs either being fixed as they come up (if necessary for the completion of the feature) or documented in the issues section of the github. Inside the github, each glitch will be rated by order of severity. High Priority glitches being ones that have 1) a high probability of occurring and/or 2) high obstruction to gameplay. Low Priority glitches will be glitches that 1) have a low probability of occurring and 2) cause minor obstructions to gameplay. Once the features have finished development, the milestone will enter the alpha period. During this period, the features will be polished, tested, and any bugs documented in github issues will be fixed, with High Priority glitches being focused on first. For features where it’s feasible, unit tests will be written to help with regression testing in the future. If there are previous milestones that have been completed, regression testing will take place in this stage. Once the due date for the milestone has been reached, and all High Priority/known glitches have been patched, the milestone moves into the beta stage. The beta stage will be the point where all functionality should be working as intended, and primary testing effort should be placed on the next milestone. However, if bugs originating from the milestone are still found later, then they will be documented and fixed whenever possible.

#### 3.8.2 Schedule

Each milestone will start in pre-alpha, transition into alpha once primary feature development has completed, and will be expected to reach beta by or at the due date of the milestone. During the last milestone before the final game is due, extensive testing will take place. Every test will be run and more tests created if possible, and every bug that has been documented but not yet fixed will be fixed in this milestone. Extensive play testing will occur as well, with each developer playing the game in its entirety multiple times, trying to play the game as it was not intended to be played to try to find more bugs.