



GDD V1.1.0

*They use to mark high water on the side of the tower walls. The etched cluster of dull lines from those days are still visible, drowned under ten feet of water.*



# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Revision History</b>	<b>4</b>
<b>Overview</b>	<b>5</b>
Concept	5
Genre	5
Target Platform	5
Development Tools	5
<b>Story</b>	<b>5</b>
<b>Gameplay</b>	<b>7</b>
1 Resources	7
1.1 Health	7
1.2 Hunger	7
1.3 Warmth	7
1.4 Sickness	9
1.4.1 Food Poisoning	9
1.4.2 Pneumonia	9
1.5 Resource Interface	9
2 Movement	9
2.1 Land Movement	10
2.2 Water Movement	11
2.3 Raft Movement	11
2.4 Interacting with Objects	12
Figure 2.4.4.1	12
2.5 Interacting with Buildings	12
3 Items	13
3.1 Item Construction	13
3.1.1 Category List	13
3.1.2 Type List	14
3.1.3 Item Specification	14
3.1.4 Edible Items	14
3.2 Finding Items	15
3.2.1 Rooftop Items	15

3.2.2 Rooftop Door Items	15
3.2.3 Water Items	16
3.3 Inventory	16
3.3.1 Containers	16
3.4 Crafting	17
3.4.1 Inventory Crafting	17
3.4.2 Item Modification	18
3.4.3 Fires	18
3.4.4 Cooking	19
3.5 Tools	19
3.5.1 Fishing Rod	19
3.5.2 Weather Measurement Tools	20
3.6 Rites	20
Figure 3.6.1 Rite table	21
3.7 Notes	21
3.7.1 Note	21
3.7.2 Note Generation	21
4 Emergency Radio	22
4.1 Weather Station	22
4.2 Music Station	22
4.3 Mystery Station	22
5 Weather System	23
5.2 Water Level	23
5.3 Day and Night Cycle	24
6 City Procedural Generation	24
6.1 District Generation	24
6.2 City Block Generation	24
6.3 Building Generation	25
6.3.1 Building Placement	25
6.3.2 Building Features	25
7 Event System	25
7.1 Radio Changes	25
7.2 Item Generation Changes	26
7.3 Weather Sound Changes	26
8 Saving and Loading	26
8.1 Saving	26
8.2 Loading	26
9 End States	27
9.1 Death State	27
9.2 End Scene	27
10 User Interface	28
10.1 UI Flow	28
10.2 Main Menu	29

10.3 Settings Options	30
10.4 Help Option	31
10.5 Exit/End Credits Option	32
Figure 10.5.1	32
10.6 Game Screen/HUD	32
10.7 Pause Menu	33
10.8 Inventory & Crafting Menus	34
10.9 Radio Menu	35
<b>Non-Functional Requirements</b>	<b>35</b>
System Requirements	35
Minimum System Requirements	35
Recommended System Requirements	36
External Requirements	36
Performance	36
<b>Appendix</b>	<b>37</b>
Glossary	37

# Revision History

## **V 1.0.0 - Completed 09/24/2016**

- Write up from first all group meeting.

## **V 1.0.1 - Completed 09/26/2016**

- Added updated Glass Prison Games logo.

## **V 1.1.0 - Completed 10/20/2016**

- Created more specific, numbered requirements
- Reordered some sections so definitions come first
- Updated format for requirements
- Bolded terms we may want to include in a glossary
- Added content to Resources, Movement, and City Generation sections

## **V 1.1.1 - Completed 10/22/2016**

- Finished remaining gameplay section and added requirements for each section

## **V 1.1.2 - Completed 10/24/2016**

- Updated tense for weather system and raft

## **V. 1.1.3 - Completed 1/22/2017**

- Added story details
- Added event system information
- Updated emergency radio information
- Added note system information
- Updated Table of Contents
- Updated item information
- Updated resource information

## **V. 1.1.4 - Completed 4/16/2017**

- Format edits
- Grammatical edits
- Sections combined

# Overview

## Concept

In a city where the storm never ends and the sea has swallowed the land, a single person still lives. The city teems with fish to hunt, plants to harvest, and materials to scavenge. Survival seems at least possible, but with the storm constantly brewing, it's always uncertain when a sudden flood may drown everything. Your only friend is a small emergency radio.

*Highwater* is a roguelike survival game that places heavy emphasis on its weather and crafting systems. The ongoing storm will cause fluctuations in temperature, wind, air pressure, and water level, creating a constantly changing terrain for the player to navigate. Learning to read the weather patterns and the dangers they pose will be vital to player survival. Players may gather various raw materials to be used in crafting, each with properties that will allow them to be combined in many ways to create more complex tools. By examining the signs that people have left behind, the player may also find guides to creating more useful tools, discover better methods for survival, and learn the past of the city in the sea.

## Genre

Survival/Roguelike

## Target Platform

PC/Mac

## Development Tools

Unity 5.4.0f3  
Modo 801  
Visual Studio  
Maya 2016/2017  
Unity Cloud Build  
Github  
WaffleIO  
FMOD Studio  
RT Voice

## Story

As the player works to survive the increasing water levels and extreme weather conditions, they find remnants of the lives of the city's past inhabitants, who were forced atop the roofs of their homes and businesses in an effort to survive. Leftover camps, notebooks, and scrawlings into

walls throughout the city show that its previous inhabitants largely fell into two belief systems -- science and faith -- when coming to terms with their relative inability to stop the storms from bearing down upon them.

Citizens who believed that faith alone would allow them to ride out the storm and find a way to stop the extreme weather have left behind tokens of their beliefs which the player can collect or craft to aide their safety. Religious inspiration for these individuals was drawn both from their existing spiritual beliefs and new myths created in the face of previously improbable weather conditions. Meanwhile, the pragmatic scientists of the post-flood society opted to record their theories in books and weather-surveying tools that can be crafted or found atop the city's roofs. These citizens opted to put their faith in their own abilities, convinced that there must be a way to stop or at least abate the devastating impact of the city's ever-present floodwaters. Players can use their leftover knowledge to help predict weather patterns in order to increase their chances of survival.

In an effort to survive the increasing water levels and extreme weather conditions, the player's ultimate goal is to reach the top of the city's tallest tower, as this is indicated throughout the game to be their best chance of rescue. Success in reaching this point is dependent on the player's ability to collect and use necessary tools to allow for survival and water travel.

In addition to the last vestiges of the city's past, the player can use an emergency radio to assist them in escaping the flood. This radio, which the player carries with them throughout the game, is their only connection to information outside of what can be found in the city. Real-time weather reports and storm warnings played by an automated system can help the player to make decisions about whether to push forward or take cover in the wake of extreme weather patterns. Meanwhile, the radio's other channels, playing news broadcasts on loop and other cryptic songs and messages, can alert the player to places of safety and resources that the city's past residents came to rely on. While the radio isn't a perfect system -- after all, who's deciding what's getting played on this thing any more? -- it's one of the player's best resources for survival.

# Gameplay

## 1 Resources

The player must manage their **resources** carefully to maintain their health. The player has three resources: Health, Hunger, and Warmth. The player begins the game with full resources. Over time and exposure to the environment, the player is susceptible to increasing hunger and decreasing warmth. If warmth or hunger drop to 0, the player's health begins to be negatively affected. The player dies if their health drops to 0. The status of each resource is displayed to the player in the **HUD**. Food poisoning and illness can also , which affect the player's hunger, warmth, and mobility.

### 1.1 Health

**Health** measures a player's overall well-being. If at any point the player's Health drops to 0, the game is over. Any time the player takes physical damage, such as from falling, their Health will decrease one unit per unit of distance fallen.. Consuming medicine or using medicine boxes will increase the player's Health if it is below 100.

- 1.1.1** Health cannot exceed a maximum value of 100
- 1.1.2** Health cannot fall below a minimum value of 0
- 1.1.3** When a player's Health reaches 0, the game is over
- 1.1.4** If the player consumes medicine or applies a health kit, their Health will be increased by the amount specified in the items section.
- 1.1.5** The player starts the game with a Health value of 100.

### 1.2 Hunger

**Hunger** reflects the player's need for food and will constantly drop over time. It can be improved by consuming food, especially food that has been cooked. Certain food items may be poisonous or infected and have a chance of causing food poisoning, as specified in section 1.4. If hunger drops to 0, the player's health will begin to be negatively affected.

- 1.2.1** Hunger constantly drops over time at a rate of 1 unit every 7 seconds
- 1.2.2** Consuming a food item will improve the player's hunger by the amount specified in the items section
- 1.2.3** Hunger cannot drop below 0
- 1.2.4** While hunger is at 0, the player's health will drop at a rate of 1 unit per 20 seconds
- 1.2.5** Hunger cannot exceed a maximum value of 100
- 1.2.6** The player starts the game at with Hunger at 100

### 1.3 Warmth

**Warmth** measures the player's exposure to the elements and drops when a player is in a cold or wet environment. Taking shelter will stop warmth from decreasing. It can be improved by

proximity to a heat source, such as a fire. Wearing clothing items can reduce the rate of warmth loss. If warmth drops to 0, the player's health will begin to be negatively affected.

**1.3.1** Warmth constantly drops over time at the **warmth reduction rate** which is determined by the current temperature. The warmth reduction rate is inversely proportional to the current temperature:

Temperature (degrees)	Warmth Reduction Rate
$\geq 60$	1 unit every 10 seconds
$\geq 50$	1 unit every 9 seconds
$\geq 40$	1 unit every 8 seconds
$\geq 30$	1 unit every 7 seconds
$\geq 20$	1 unit every 6 seconds
$\geq 10$	1 unit every 5 seconds
$\geq 0$	1 unit every 4 seconds
$< 0$	1 unit every 3 seconds

Figure 1.3.1.1

**1.3.1.1** When temperature is greater than 60 degrees, decrease rate is 1 unit per 10 seconds

**1.3.1.2** When temperature is under 60 degrees but greater than or equal to 50 degrees, decrease rate is 1 unit per 9 seconds

**1.3.1.3** When temperature is under 50 degrees but greater than or equal to 40 degrees, decrease rate is 1 unit per 8 seconds

**1.3.1.4** When temperature is under 40 degrees but greater than or equal to 30 degrees, decrease rate is 1 unit per 7 seconds

**1.3.1.5** When temperature is under 30 degrees but greater than or equal to 20 degrees, decrease rate is 1 unit per 6 seconds

**1.3.1.6** When temperature is under 20 degrees but greater than or equal to 10 degrees, decrease rate is 1 unit per 5 seconds

**1.3.1.7** When temperature is under 10 degrees but greater than or equal to 0 degrees, decrease rate is 1 unit per 4 seconds

**1.3.1.8** When temperature is under 0 degrees decrease rate is 1 unit per 3 seconds

**1.3.2** Warmth drops twice the warmth reduction rate according to current temperature when the player is in the water.

**1.3.3** Standing within a radius of 1 unit of distance to a heat source causes the player's warmth to increase by 4 units every 1 second.

**1.3.4** When a player is wearing an item of clothing, the warmth reduction rate will be modified by the factor specified in the item section.

**1.3.5** When the player is standing within a shelter, warmth reduction rate is 0.

**1.3.6** Warmth cannot drop lower than 0.

**1.3.7** While warmth is at 0, the player's health will drop at a rate of 1 unit every 1 second.

**1.3.8** Warmth cannot exceed a maximum value of 100

**1.3.9** The player begins the game with Warmth at 100

## 1.4 Sickness

Through actions such as eating rotten food, a player can gain a status effect that negatively affects the player's resources and general well-being. This is known as **sickness**. It will affect one resource and slow the movement of the player. A sickness can be cured by taking medicine.

**1.4.0.1** Having a sickness causes the player to move at half speed, both on land and in water.

**1.4.0.2** The player can only have one sickness at a time. Another cannot be gained while one sickness is already active.

**1.4.0.3** When a disease is caught, a message will be displayed in the **HUD**.

**1.4.0.4** While the player has a disease, the affected resource UI bar will display an additional icon to show it is being affected.

### 1.4.1 Food Poisoning

**1.4.1.1** The player has a 30% chance of acquiring food poisoning from eating a "rotten" food item

**1.4.1.2** The player has a 20% chance of acquiring food poisoning from eating raw meat

**1.4.1.3** Having food poisoning causes the player's hunger to drain twice as fast

**1.4.1.4** When the player eats stomach medicine, they are cured of food poisoning.

### 1.4.2 Pneumonia

**1.4.2.1** The player has a 10% chance of catching Pneumonia when their warmth is decreasing and their warmth value is below 25.

**1.4.2.2** Having Pneumonia causes the player's warmth to drain twice as fast

**1.4.2.3** When the player eats Pneumonia medicine, they are cured of Pneumonia.

## 1.5 Resource Interface

The status of a resource is displayed in the **HUD** if its current value is less than the maximum. See the User Interface section for more details on the display.

**1.5.1** When a resource is less than 15, it will generate a new pulse from the resource HUD icon every half second to indicate that the resource is low.

**1.5.2** The warning pulses increase in intensity by growing 1.25 times faster per 5 units dropped from 15.

## 2 Movement

The player moves on solid ground and swims in water.

The movement is relative to the current camera view, with the up key moving the player away from the camera, the down key moving the player towards the camera, the left key moving the player to the left of the screen, and the right key moving the player to the right of the screen.

*The user will be able to interact using the following controls.*

Action	Default Button
Move around	WASD   Arrow keys
Inventory	I
Menu	Esc
Crafting	C
Climb   Jump	Space
Action   Pick up item	F
Rotate Camera	Q   E
Sprint	Shift

## 2.1 Land Movement

The primary movement for the player on land is **running** by using the arrow keys or WASD. The player may also perform a **jump**, or **climb**, depending on their surroundings when using the jump key. The player can also **sprint** by holding down the sprint button while moving.

**2.1.1** On land, the player may run in any direction.

**2.1.1.1** Pressing up moves the player away from the camera

**2.1.1.2** Pressing down moves the player towards the camera

**2.1.1.3** Pressing left moves the player towards the left side of the screen

**2.1.1.4** Pressing right moves the player towards the right side of the screen

**2.1.1.5** Pressing two adjacent keys will move the player at 45 degree angle between the two directions

**2.1.2** The player will always face the direction of movement.

**2.1.3** The player always walks at a speed of 0.01 units per frame on land regardless of direction.

**2.1.4** The player may perform a jump by hitting the jump button. This will run the jump animation and move the player up 1 unit.

**2.1.5** The player may climb by approaching a short ledge and hitting the jump button.

**2.1.6** If the player presses the jump button and is near a ledge, climbing will take precedence over jumping.

**2.1.7** If the player falls off a building, the player will receive damage to their health relative to the height of the fall upon hitting the ground, at 1 damage for every unit the player falls.

**2.1.8** If the player falls off a building into the water, the player will receive damage to their health relative to the height of the fall upon hitting the water, at 0.5 damage for every unit the player falls, unless the height is less than 8 units.

**2.1.9** When moving, the player performs a running animation.

**2.1.10** When not moving, the player performs an idle animation.

- 2.1.11** When jumping, the player performs a jump animation.
- 2.1.12** When climbing, the player performs a climb animation.
- 2.1.13** When falling, the player performs a fall animation.
- 2.1.14** When holding the sprint key, the player runs with a speed of 0.02 units per frame.

## 2.2 Water Movement

When in water, the player will move by **swimming**, which is similar to running but at a slower speed. The player may also perform a **climb** action onto a low surface using the jump key.

- 2.2.1** In water, the player may swim in any direction.
  - 2.2.1.1** Pressing up moves the player away from the camera
  - 2.2.1.2** Pressing down moves the player towards the camera
  - 2.2.1.3** Pressing left moves the player towards camera left
  - 2.2.1.4** Pressing right moves the player towards camera right
  - 2.2.1.5** Pressing two adjacent keys will move the player at 45 degree angle between the two directions
- 2.2.2** The player always faces the direction of movement.
- 2.2.3** In water, the player always moves at a speed of 0.005 units per frame, regardless of direction.
- 2.2.4** The player may climb by approaching a short ledge and hitting the jump button.
- 2.2.5** When moving, the player performs a swimming animation.
- 2.2.6** When not moving, the player performs an idle water animation.
- 2.2.7** When climbing, the player performs a climb animation.

## 2.3 Raft Movement

When on a raft, the player's normal movement stops, and the same controls previously used to control the player instead control the movement of the raft. Instead of directly affecting the velocity, like on water and land, the player moves by adding a force in the direction of the raft's movement. The raft is also subject to outside forces such as current, fish, and obstacles.

- 2.3.1** The player may board a raft using the action button.
- 2.3.2** When on a raft, the player may get off the raft using the action button.
- 2.3.3** When on a raft, the player may use the directional controls to control the raft.
  - 2.3.3.1** Pressing up applies a force to the raft in the direction away from the camera.
  - 2.3.3.2** Pressing down applies a force to the raft in the direction towards the camera.
  - 2.3.3.3** Pressing left applies a force to the raft towards camera left.
  - 2.3.3.4** Pressing right applies a force to the raft towards camera right.
  - 2.3.3.5** Pressing two adjacent keys will apply a force to the raft at 45 degree angle between the two directions
- 2.3.4** When the player is applying a force to the raft, it will always be a force of 0.001 units per frame, regardless of direction.
- 2.3.5** The raft will never come out of the water
- 2.3.6** The raft will never rotate on any axis except around the up axis.

## 2.4 Interacting with Objects

Approaching an object that the player can interact with such as a raft or an inventory item will show text above the object explaining what action can be performed on that object. For instance, approaching a raft will show the text "Board raft". Hitting the action button will perform that action. If there is more than one action available, they will be assigned to different buttons.

**2.4.1** When the player is within a radius of 1 unity units of an interactable object, the action text appears.

**2.4.1.1** The action text matches the action that can be performed for that object.

**2.4.1.2** The object has a visual outline which is displayed when the action text appears.

**2.4.1.3** If multiple actions are available, they are all shown.

**2.4.2** Pressing the specified button performs the intended action.

**2.4.3** When the player exits the radius, the highlight and text disappear.

**2.4.4** Rafts are special objects that can be used to travel the world

Object	Action	Button
Poor Raft	Board raft/Disembark raft	F
Standard Raft	Board raft/Disembark raft	F
	Store items	R
Excellent Raft	Board raft/Disembark rafts	F
	Store items	R

Figure 2.4.4.1

**2.4.4.2** Poor rafts can be used to travel.

**2.4.4.3** Standard rafts can be used travel and store 10 items.

**2.4.4.4** Excellent rafts can be used to travel and store 30 items.

## 2.5 Interacting with Buildings

Buildings are found throughout the city and can have items inside of them. They can provide means of allowing the player to traverse the city and can also provide shelter for the player to keep them out of the elements.

**2.5.1** The player can enter a building if there is a door on its roof.

**2.5.2** If there is an item inside the building, the items are tossed out of the building when its door is opened and the player can interact with these items as usual.

**2.5.3** The player exits if there are no items or all the items have been removed.

**2.5.4** If a door has been opened, it will remain open after the player leaves to signify this.

**2.5.5** The player can climb up and down fire escapes on buildings.

**2.5.6** The player can climb up and down ladders on buildings.

## 3 Items

Items can be found throughout the city. They are necessary to survival and managing the player's resources. The benefit of each item differs, with some alleviating hunger or increasing warmth, while others serving as raw components that can be combined to create new items.

### 3.1 Item Construction

All items in the game are made up of a `Baselitem` class, which controls the entire item and its Item Category class. Item Category classes contain a suite of attributes and actions that an item may have and perform. Items may have as many Item Category classes as necessary.

In addition to being defined by their categories, items are also divided into item types. Item types are what drives the crafting system, which involves combining items with certain tags. An item's types determine which crafting recipes it can be used in. For example, because both a vine and a string can have the "rope" type, they can both be used when a fishing rod recipe calls for a "rope" typed item.

#### 3.1.1 Category List

Category	Description	Attributes	Actions
Solid	This includes all solid items. All items that are an <code>OrganicSolid</code> or <code>MetallicSolid</code>	flexibility, durability, sharpness	Weave, sharpen
Fuel	This includes items that can be used as fuel. Generally oils, fats, or dry plant matter.	lifeGain, burnRate	fuelFire
Ignition	This includes items that can be used to start a fire.	dampness	dry, startFire
Plant	This includes items that came from a plant.	hungerGain, poisoness	cook, eat
Flesh	This includes items that came from an animal.	hungerGain, doneness	cook, eat
Fabric	This includes items that have cloth-like qualities. Can include bandages, sheets, or animal skins.	warmth, size	
Container	This includes items that can be used to increase inventory space.	itemHold	addTo
Medicine	This includes items that can	healthGain,	heal, cureSickness

	increase health or cure illnesses when eaten.	sicknessSpecialization	
--	---	------------------------	--

Figure 3.1.1.1

### 3.1.2 Type List

Tag	Description
Blunt	This is attached to items that are not sharp.
Cloth	This is attached to items that can be sewn together.
Edible	This is attached to items that can be eaten.
Equipable	This is attached to items that can be equipped.
Flammable	This is attached to items that can catch on fire easily.
Floatable	This is attached to items that can float.
Ignitor	This is attached to items that can be used to start fires.
Receptacle	This is attached to items that can hold items.
Rod	This is attached to items that can be used to create tools like a fishing rod.
Rope	This is attached to items that can be used to lift objects or sew.
Sharp	This is attached to items that are sharp.
Fluid	This is attached to items that are liquids.

Figure 3.1.2.1

### 3.1.3 Item Specification

The list of possible items is stored within the **ItemList.yml** file and read in when the game is first loaded. This file is parsed by the serializer. Baselitem classes and their category classes are linked up and hashed into a Dictionary for reference. When given a name, the ItemFactory class will reference the Dictionary and return the item.

- 3.1.3.1 An item generated in game is defined in the ItemList.yml file
- 3.1.3.2 When an item is needed, the information to create it is pulled from the Dictionary
- 3.1.3.3 An item's **ID** is the item's name.
- 3.1.3.4 Only the most base form of items are stored within the ItemList.yml file
- 3.1.3.5 Only base forms of the items are found naturally in the world

### 3.1.4 Edible Items

**3.1.4.1** Edible items can be eaten

**3.1.4.2** When an item is eaten, the player's Hunger is restored according to Figure 3.1.4.2.1

Item	Hunger Restored
Raw Fish Meat	15
Raw Rat Meat	10
Cooked Fish Meat	30
Cooked Rat Meat	20
Lotus Root	5
Water Chestnut	2
River Weed	3
Mushroom	2
Egg	5

Figure 3.1.4.2.1

## 3.2 Finding Items

Most of the resources found in the city are basic items that act as the raw materials for crafting. These materials can be combined into more advanced tools and items.

### 3.2.1 Rooftop Items

Resources can be found throughout the game world. On land, items can be found lying in the open air on rooftops or floors.

**3.2.1.1** Items are randomly placed on rooftops

**3.2.1.2** Items do not spawn at the very edge of buildings

**3.2.1.3** Items do not spawn on steep slopes

**3.2.1.4** Items tend to spawn in a more tightly packed manner when further from doors on the same rooftop

### 3.2.2 Rooftop Door Items

On 30% of rooftops are doors that the player can interact with to spawn supplies.

**3.2.2.1** Doors are randomly spawned on rooftops

**3.2.2.2** Doors do not spawn at the edges of buildings

**3.2.2.3** Doors do not spawn on steep slopes

**3.2.2.4** The number of objects that spawns from doors is calculated by the different between the water height and the top of the building.

**3.2.2.5** When the player interacts with a door, the door opens and items spawn in front of the door.

### 3.2.3 Water Items

In the water, the player may find items floating on the surface. Furthermore, fish can be caught using either a fishing rod.

**3.2.3.1** Items are randomly spawned on the water surface.

**3.2.3.2** Items in the water float.

**3.2.3.3** Fish spawn below the water surface and can only be caught using a fishing rod.

**3.2.3.4** Fish avoid colliding with other fish.

**3.2.3.5** Fish are attracted to a fishing rod's lure.

## 3.3 Inventory

An **inventory** is a structure where the player can store items. Each item is stored in a **slot**. Collected items will be added to the player inventory. Each inventory has a limited amount of space, so the player may also choose to store items in separate storage spaces such as containers.

The player can craft containers to store items in the world. Inventories also appear on rafts and in buildings.

**3.3.0.1** Each inventory has a specified number of slots.

**3.3.0.1.1** The player inventory has **20** slots.

**3.3.0.1.2** A container has the number of slots specified in the figure 3.3.1.3.

**3.3.0.2** A slot may contain up to 99 instances of the same item for crafting materials.

**3.3.0.3** A slot may only contain 1 instance of an item for crafted items.

**3.3.0.4** When an object is collected by the player, and a slot containing that same item already exists and has yet to reach 99 instances, it will be added to that slot.

**3.3.0.5** When an object is collected by the player and a slot containing the same item does not exist, then it will be added to the first empty slot.

**3.3.0.6** If the inventory is full, the player cannot collect another item. A notification is displayed and the item remains in the world.

### 3.3.1 Containers

Containers are objects that have a player accessible inventory.

**3.3.1.1** Containers may appear as standalone items in the world.

**3.3.1.2** Container may be attached to other objects such as a building or a raft.

Container	Storage slots
Building	10
Good Raft	25
Excellent Raft	50

Figure 3.3.1.3. Container list

## 3.4 Crafting

Players may craft items to help improve their chance of survival as the storm worsens. By crafting better items, players can improve hunger, better endure the cold, and predict the weather.

### 3.4.1 Inventory Crafting

Each combination of items that results in a new item is called a recipe. An item is considered to be a valid ingredient in a recipe when it is of the tag specified. For example, if a recipe calls for a “rod” tagged item as a recipe, then an item tagged with “rod” is considered valid. All recipes are stored in the **Recipe.yml** file. All unlocked recipes are stored in the the **UnlockedRecipe.yml** file.

When new items are created by combining existing items, the resulting item first takes the averages of the attributes of the items used to create it. Then, it will check the item’s attributes, and assign it epithets of *Poor*, *Standard*, or *Excellent* depending on the value of the averages. The stats of the resulting item is also changed to reflect whether it is *Poor*, *Standard*, or *Excellent*. Attributes are valued from 0 to 100.

**3.4.1.1** What thresholds constitute Poor, Standard, or Excellent are specified for each crafting recipe in the Recipe.yml file

**3.4.1.2** Contribution of an item’s attribute is dependent on how much of that item is used during crafting

**3.4.1.3** It is possible for lower attribute values to be considered better than higher attribute values, such as lower thickness values on a rope type item being ideal for weaving.

**3.4.1.4** Item type determines whether or not an item is usable in a recipe.

**3.4.1.4.1** Fishing rods are used to catch fish. They are made from 1 rod, 1 rope, and 1 sharp object.

**3.4.1.4.2** Rafts are used to traverse the water. They are composed of 50 floatable objects and 20 rope.

**3.4.1.4.3** Nets are a trap used to catch fish. They are made from 10 ropes.

**3.4.1.4.4** Barometers are used to measure atmospheric pressure. They are composed of 1 container, 1 rod.

**3.4.1.4.5** Thermometers are used to measure the temperature. They are composed of 2 containers, 1 rod

**3.4.1.4.6** Hygrometers are used to measure humidity. They are composed of 1 cloth, 1 sharp object, and 1 rope.

**3.4.1.4.7** Coats are worn to decrease the rate at which warmth falls. They are composed of 10 cloth, and 1 rope.

### 3.4.2 Item Modification

Item modification occurs when new items are created not by combining raw materials, but by applying actions to items. The results do not yield advanced tools, but rather more refined raw materials. After an action is completed, tags and categories may be added or removed, and attributes are changed. Once an action is applied to an item, it cannot be applied again. An action applied may also remove other actions from being an option for that item in the future.

**3.4.2.1** The name of the object changes to reflect the actions that have been applied to it.

Action	Resulting Name
Weave Rope	X Thread/Rope
Sharpen	Sharpened X
Weave Basket	X Basket
Cook	Cooked X

Figure 3.4.2.1. Modifications table

### 3.4.3 Fires

Fires are started by setting down Fire Pit items and lighting them with an Igniter type item in a shelter. The time in which the fire remains is determined by their **burn rate**, which specifies how long a single unit of an item will last. Players can continuously add fuel to fire to keep it burning. When the player chooses to rest by a fire, their warmth increases four times as quickly as resting in a shelter without fire.

**3.4.3.1** When interacting with an unlit Fire Pit, the action menu displays the *Ignite* option.

**3.4.3.2** When the player selects the Ignite option, all *Igniter* items in the player's inventory will be shown as an option.

**3.4.3.3** Upon selecting an *Igniter* item, the Fire Pit becomes lit and becomes a heat source while it is lit.

**3.4.3.4** Fire continues to burn for duration of the sum of the burn rates of items added to the fire.

**3.4.3.5** When interacting with a lit Fire Pit, the action menu displays the *Add Fuel* option.

**3.4.3.6** When an item is added to a fire, the duration of its burn rate is added to the length of the fire for each unit of the item added.

**3.4.3.7** Being within 3 unity units of distance of a fire will cause the player's warmth to increase **4 units per second**

**3.4.3.8** If the player touches a fire, their health will decrease **10 units per second** while remaining in contact with the fire

Item	Burn Rate (seconds per unit)
------	------------------------------

Stick	5
Wood Log	20
Plant Detritus	2
Rotting Flesh	25
Oil	15
Alcohol	3

Figure 3.4.3.6.1. Burn rate table

### 3.4.4 Cooking

The **cooking** action is only available at fires. The player may cook any edible items that they have. When consuming raw meat, the player has a **25%** chance of contracting food poisoning. Cooked food also improves hunger twice as much as uncooked food.

**3.4.4.1** When players interact with fire, Cook appears as an option in the possible action button menu.

**3.4.4.2** Only *edible* items are listed as possible items when the Cook option is selected.

**3.4.4.3** When a cooked item is eaten, it improves hunger at **2** times the base rate for that item

**3.4.4.4** When uncooked meat is consumed, the player has a **25%** chance of contracting food poisoning.

## 3.5 Tools

Tools are items that can be equipped and used by the player. They have a durability that decreases with use. When the durability is 0, the tool breaks.

**3.5.0.1** Tools can be equipped.

**3.5.0.2** Tools can unequipped.

**3.5.0.3** Tools can be used by pressing the use tool button.

**3.5.0.4** Tools have a maximum durability of **100**.

**3.5.0.5** Tools have a current durability that decreases with each use.

**3.5.0.6** When the current durability is **0**, the tool breaks. A broken tool is no longer an item.

**3.5.0.7** Upon breaking, for each component making up the tool, there is a 10% chance of that item being returned.

**3.5.0.8** Upon using the tool, the player cannot move until the use action is complete.

**3.5.0.9** Only one tool can be equipped at a time.

### 3.5.1 Fishing Rod

The fishing rod can be used to fish. When used, it will cast a fishing lure into the water. When used again, the player will reel in the lure. If there was an item like a fish or junk on the line, the player places that item in their inventory.

**3.5.1.1** When not cast, pressing the use button will cast the fishing lure.

**3.5.1.2** When cast, pressing the use button will reel the fishing line back in.

**3.5.1.3** Fish or other items can become attached to the fishing lure.

**3.5.1.4** If an item is attached when reeled in, the player will add it to their inventory.

### 3.5.2 Weather Measurement Tools

When used, weather measurement tools communicate to the player the current weather conditions. Different tools tell the player different aspects of the weather, such as temperature or humidity.

Meteorological Tool	Effect
Barometer	Measures surrounding air pressure
Thermometer	Measures temperature
Hygrometer	Measures humidity

Figure 3.5.2.1. Weather tool table

**3.5.2.1** Barometers measure air pressure

**3.5.2.2** Thermometers measure temperature

**3.5.2.3** Hygrometers measure humidity

## 3.6 Rites

Rites can be performed by the player to provide temporary status effects. Rites consist of combining certain ingredient items and performing an action on the resultant item. The rite items, their ingredients, their necessary action, and the effects are shown in Figure 3.6.1.

Name	Resource Modified	Ritual Type	Ingredients	Effect
Lotus Incense	Item Drop	Burn	Lotus Root, Fire	More Items Spawn
Fire Kabob	Movement	Burn	Red Moss, Stick, Rope, Fire	Land Speed Increase
Drowned Kabob	Movement	Burn, Float	Red Moss, Stick, Rope, Fire	Swim Speed Increase
Dark Plank	Day/Night Cycle	Burn	Wood Log, Knife	Increased Vision Radius at Night
Ultimate Sacrifice	Health	Burn, Float	Any Raft, Flammable Object on Fire	Health Doesn't Go down. Walk on Water. Instant Whale Spawn
Stick In A Bottle	Hunger	Burn, Sink	Stick, Bottle, Fire	Slow Down Hunger Rate
Small Viking Funeral	Raft Movement	Burn, Sink	Wood Log, River Weed, Rope, Fire	Raft Movement Speed
Bermuda Triangle	Fish	Float	River Weed x3, Rope	Higher % of Catching Fish

				Lower Warmth Reduction Rate. Warmth Drop Rate Remains the Same in Water.
Fire Stone	Warmth	Held	Red Rock	
Milwa Stone	Sickness	Held	Stone, Knife	Cure Stomach Illness
Soyah Stone	Sickness	Held	Stone, Knife	Cure Pneumonia
Atma Stone	Rain	Held	Stone, Knife	Water Level Rises Faster
Yintar Stone	Rain	Held	Stone, Knife	Water Level rises Slower

Figure 3.6.1 Rite table

**3.6.1** When the player combines the ingredients and performs the action for a rite specified in Figure 3.6.1, the listed effect occurs.

**3.6.2** Effects of a used rite are no longer active after 5 minutes of real time.

## 3.7 Notes

While traveling through the city, the player can find remnants of the society that existed there before their time. **Notes** found lying around the city offer glimpses into the emotions and opinions of the city's lost residents. These notes can include information that may help the player, such as crafting recipes or weather data, as well as glimpses into city citizens' personal lives.

### 3.7.1 Note

Notes are items that are placed in the world that the player has the ability to read. When read, a **Note UI Panel** displays the note contents.

**3.7.1.1** Note objects exist in the world.

**3.7.1.2** When a player is within 1 unit of the note, the option to read to note is shown.

**3.7.1.3** If the player is within 1 unit of a note and presses the interaction button, the Note UI Panel is shown.

**3.7.1.4** The Note UI Panel displays the text content of the note.

**3.7.1.5** If the text within the note is longer than the panel, the panel has the ability to scroll.

**3.7.1.6** The Note UI Panel contains a close button that when clicked closes the Note UI Panel.

**3.7.1.7** When the Note UI Panel is open, pressing "ESC" closes the note UI panel.

**3.7.1.8** Notes cannot be picked up or moved in the world.

**3.7.1.9** Notes cannot be used in crafting.

### 3.7.2 Note Generation

Notes are placed throughout the world similarly to items, but possess different properties from craftable items.

**3.7.2.1** Notes are generated based on the Notes.yml file.

**3.7.2.2** Certain notes are placed on specific buildings in the city.

**3.7.2.3** General notes are placed randomly throughout the city similarly to items.

## 4 Emergency Radio

The player starts the game with an emergency radio. It serves as a special item in the game. The radio has three stations that the player can listen to, each of which plays tracks from a looping **carousel** of possible sounds that is updated throughout gameplay. The radio acts as the only mode of communication for information about weather and background story to reach the player.

- 4.0.1** The radio may be turned on by pressing the on/off button when the radio is off.
- 4.0.2** The radio may be turned off by pressing the on/off button when the radio is on.
- 4.0.3** The player may switch between three channels on the radio (weather, music, and mystery) by rotating the channel dial.
- 4.0.4** Some radio stations become inaudible during storms.
- 4.0.5** The radio volume may be changed using the volume up and down buttons.
- 4.0.6** The radio UI can be shown by pressing the Radio button in the HUD.
- 4.0.7** The radio UI can be hidden by pressing the "Close" button.
- 4.0.8** The radio UI can be toggled on and off by pressing the "R" key.

### 4.1 Weather Station

The weather station plays a looping announcement stating the current maritime weather conditions.

- 4.1.1** Weather announcements are generated by a text-to-speech library.
- 4.1.2** Weather announcements contain updates on the current state of the weather.
  - 4.1.2.1** The announcement includes the current temperature.
  - 4.1.2.2** The announcement includes the current wind speed.
- 4.1.3** During storms, the weather announcement is replaced with a storm warning.

### 4.2 Music Station

The music station plays tracks from a carousel of music depending on the player's location and the current weather.

- 4.2.1** The music station only plays music clips.
- 4.2.2** The music station can play weather-specific music clips.
- 4.2.3** The music station can play district-specific music clips.

### 4.3 Mystery Station

The mystery station will play audio clips, static, and text-to-speech clips based on the contents of its carousel.

- 4.3.1** The mystery station carousel has one consistent static clip that is played.

**4.3.2** The mystery station carousel can contain any number of additional event-based sound clips.

**4.3.3** The mystery station carousel has clips added to and removed from it based on variables such as player resources, actions, time passed, and/or player location during gameplay.

## 5 Weather System

The weather system will use real-world weather mechanics including wind, rain, humidity, and others. Players can potentially predict weather patterns via “science” or “faith” based methods (see Story). The weather system follows **Lorenzian**-based predictions to create weather system predictions in addition to physics formulas, including the ideal gas law, and quadratic regression based on real world data from Weather Underground.

### 5.1 Weather Status

**5.1.1** The weather system updates the location of pressure systems

**5.1.1.1** The system moves attracting pressure systems toward each other with an attracting force

**5.1.1.2** The system moves opposing pressure systems away from each other with an outward, perpendicular force

**5.1.2** The system calculates the temperature based on pressure given by location

**5.1.3** The system calculates the wind speed

**5.1.3.1** The system calculates the wind speed magnitude from the pressure with coefficients for quadratic regression

**5.1.3.2** The system calculates wind direction based on city center

**5.1.3.3** The system creates a wind vector with magnitude and angles using sin and cos

**5.1.4** The system calculates the humidity based on temperature and pressure with coefficients for quadratic regression

**5.1.5** The system calculates the dew point from temperature and humidity

**5.1.6** The system is subscribed to the day/night cycle (see section 5.3) every 30 minutes of game time

**5.1.7** There are visual indications that the weather is changing described in the *Visuals* section.

Feature 5.1.1 will be run by the day/night cycle with a delegate method. The player class will be calling the weather system to update the weather based on location per a fixed update. This will allow for dynamic weather based on the player's location without a heavy performance impact.

### 5.2 Water Level

The changes in weather create rising and lowering water levels, forcing the player to seek higher ground during times of flood.

**5.2.1** The tide system keeps track of high and low tide

**5.2.2** The tide system is subscribed to the day/night cycle and update every minute of game time

**5.2.3** The water level rises with high tide

- 5.2.4** The water level falls with low tide
- 5.2.5** The water level rises during times of heavy precipitation
- 5.2.6** The water level falls during periods of low precipitation

## 5.3 Day and Night Cycle

Day and night in the city will affect high & low tide where every day is 24 minutes, meaning every minute in real time is one hour in game time.

- 5.3.1** The day night cycle will update every game frame and update the game time
- 5.3.2** The day night cycle will update every minute of game time and call subscribed delegates
- 5.3.3** The day night cycle will update every 30 minutes of game time and call subscribed delegates
- 5.3.4** The day night cycle will update every hour of game time and call subscribed delegates
- 5.3.5** The day night cycle will update the scene to reflect the current time

# 6 City Procedural Generation

The city which the player explores is created using procedural generation, resulting in variation in the layout of the city and the buildings with each playthrough. The city has three distinct layers of generation: **districts** which establish a general area with a unique look, **blocks** which specify smaller groupings of buildings within a district, and **buildings** which populate blocks. Buildings are drawn procedural building meshes to create distinctive buildings.

## 6.1 District Generation

- 6.1.1** The city is composed of distinct districts whose borders are generated using random 2D seed points inside of the city bounds of 20 units as input to a **Voronoi** diagram.
- 6.1.2** One of the following district types will be assigned to each at random: Residential district, Shopping district, and Business district.
- 6.1.3** The vertex adjacent to the most districts will be marked as the city center.
- 6.1.4** The edges of each district are defined by a randomly determined number of vertices.
- 6.1.5** When the game is run using the same initial seed, the districts will always be the same.

## 6.2 City Block Generation

Districts are broken up into more regularly spaced blocks. City blocks are determined by regularly spaced control points in a Voronoi diagram within the district.

- 6.2.1** A district are populated with control points which are regularly spaced 10 units apart in a grid pattern.
- 6.2.2** The control points are used as input for a Voronoi diagram.
- 6.2.3** The output of the Voronoi diagram determines the placement of streets and are 2 units in width.

## 6.3 Building Generation

The height of buildings is determined by weighted **Gaussian** distribution across the entire city; taller buildings are concentrated towards center, but certain districts naturally have taller buildings (such as the business district).

### 6.3.1 Building Placement

Buildings will populate the area of the allocated city blocks. They will be placed close enough that movement between buildings is possible.

**6.3.1.1** Each block is divided up into lots by packing rectangles of random widths and lengths from minimum value of 2 units to maximum value of 4 units into each block's upper left corner.

**6.3.1.2** A building is generated at each rectangular lot using the width and length of the lot.

**6.3.1.3** The height of each building is specified by a 2D Gaussian distribution centered at the center of the city.

### 6.3.2 Building Features

Building features have a chance of generating at a number of attachment points.

**6.3.2.1** At each building attachment point, there is a varying percent chance of generating a building attachment that is district dependent.

**6.3.2.2** If an attachment point generates a feature, a random feature corresponding to that attachment point is selected and attached to the building as in Figure 6.3.2.3.

Attachment Point	Possible Features
Top	Roof, doorway
North, East, South, West	Windows, building wing, doorway, fire escape, window washing platform, pipes

Figure 6.3.2.3

## 7 Event System

Weather and player actions can result in changes to the world that are then reflected in changes to the radio's content, weather sound effects, and the nature of in-game item generation.

### 7.1 Radio Changes

**7.1.1** Radio clips are added to and removed from the radio station carousels based on changes to weather and time.

**7.1.2** Radio clips are added to and removed from the radio station carousels based on changes to player resources and/or location.

**7.1.3** Radio clips are made less audible by the addition of a static overlay track during storms.

**7.1.4** When the music channel on the radio is playing, the game's ambient music volume is lowered.

## 7.2 Item Generation Changes

**7.2.1** Fish generate more frequently during a storm.

**7.2.2** Items generate in the water more frequently after a storm.

## 7.3 Weather Sound Changes

**7.3.1** Thunder sound plays when lightning is spawned by the weather system.

**7.3.2** Weather sounds intensify when weather conditions intensify.

# 8 Saving and Loading

The player's current game will be able to be saved and loaded at a later time. The **game state** is the current state of the game world and will include:

- Player state
  - Resources
    - Health
    - Hunger
    - Warmth
  - Location (X, Y, and Z coordinates)
  - Inventory state
- City state
  - City generation seed
  - Items in the world and their locations that have been altered
- Environment state
  - Current time
  - Current weather state
- Settings state

## 8.1 Saving

**8.1.1** Saving takes the player class and serializes it into a file

**8.1.2** Saving takes the city state and serializes it into a file

**8.1.3** Saving takes the weather class and serializes it into a file

**8.1.4** Saving takes the clock class and serializes it into a file

**8.1.5** Saving takes the settings class and serializes it into a file

## 8.2 Loading

**8.2.1** Loading opens city player class file and deserializes it into the player state class

**8.2.2** Loading opens city state class file and deserializes it into the city state class

**8.2.3** Loading opens weather class file and deserializes it into the weather class

**8.2.4** Loading opens clock class file and deserializes it into the clock class

**8.2.5** Loading opens settings class file and deserializes it into the settings class

## 9 End States

Gameplay comes to an end when the player's health hits 0 and the death screen loads, or when they reach the end of the game.

### 9.1 Death State

Upon death, the death scene is loaded, which includes a menu with options to restart the game or return to the main menu screen.

**9.1.1** When the player's health reaches 0, the death scene is loaded.

**9.1.2** The death scene shows the text "You have died".

**9.1.3** The death scene has a button labeled "Restart", which when clicked restarts the game from the beginning.

**9.1.4** The death scene has a button labeled "Back to Main Menu", which when clicked, loads the main menu screen.

### 9.2 End Scene

If the player reaches the tallest building in the city, the game end scene is triggered. In this scene, the player can read a note on the ground telling them to wait, instructing the player that there is a beacon they can turn on to signal for help. The player can either wait or jump off the edge of the building to their death.

**9.2.1** When the player reaches the tallest building, the end scene is loaded.

**9.2.2** The player loses camera controls during the end scene.

**9.2.3** The camera moves to a cinematic position.

**9.2.4** The player's stat bars disappear.

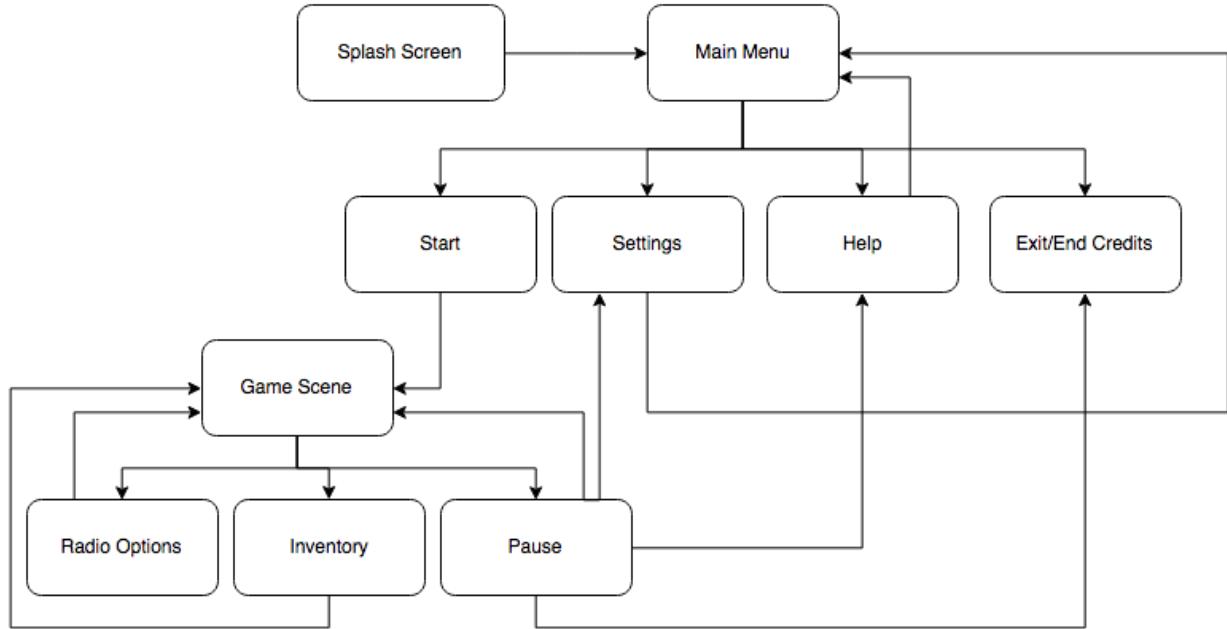
**9.2.5** The player's stats no longer change.

**9.2.6** The rooftop is populated with several notes or items.

**9.2.7** After a random amount of time, spanning from 30 seconds to 60 seconds, a ladder descends from above the player and the game ends.

# 10 User Interface

## 10.1 UI Flow



**10.1.1** The splash screen will transition to the main menu screen after 3 seconds.

**10.1.2** Pressing the **Settings** button will show the various game settings.

**10.1.2.1** Changing the key bindings for actions will set them in game

**10.1.3** Pressing the **Start** button leads to the main game scene

**10.1.4** The player can pause the game by pressing ESC

**10.1.5** When the player's health reaches 0, the game over screen is displayed

## 10.2 Main Menu

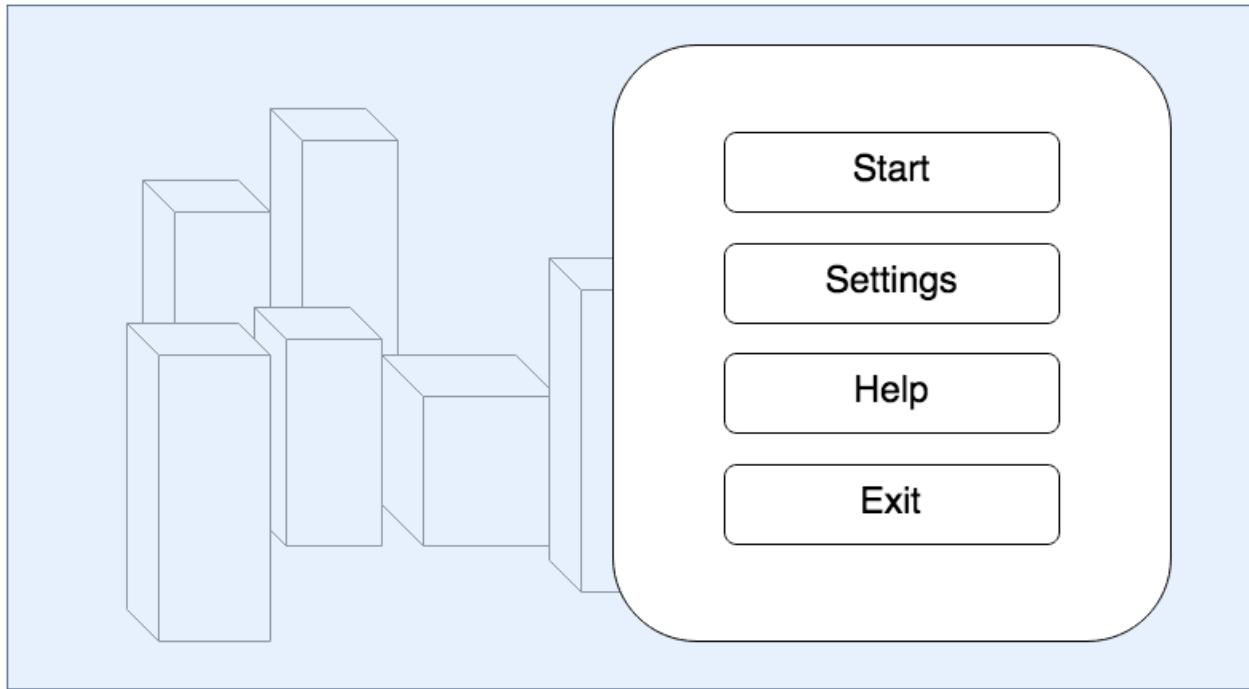


Figure 10.2.1

**10.2.2** Pressing the **Settings** button will show the various game settings.

**10.1.3** Pressing the **Start** button leads to the main game scene

**10.1.4** Pressing the **Help** button shows rules of the game.

**10.1.5** Pressing the **Exit** button closes the game.

### 10.3 Settings Options

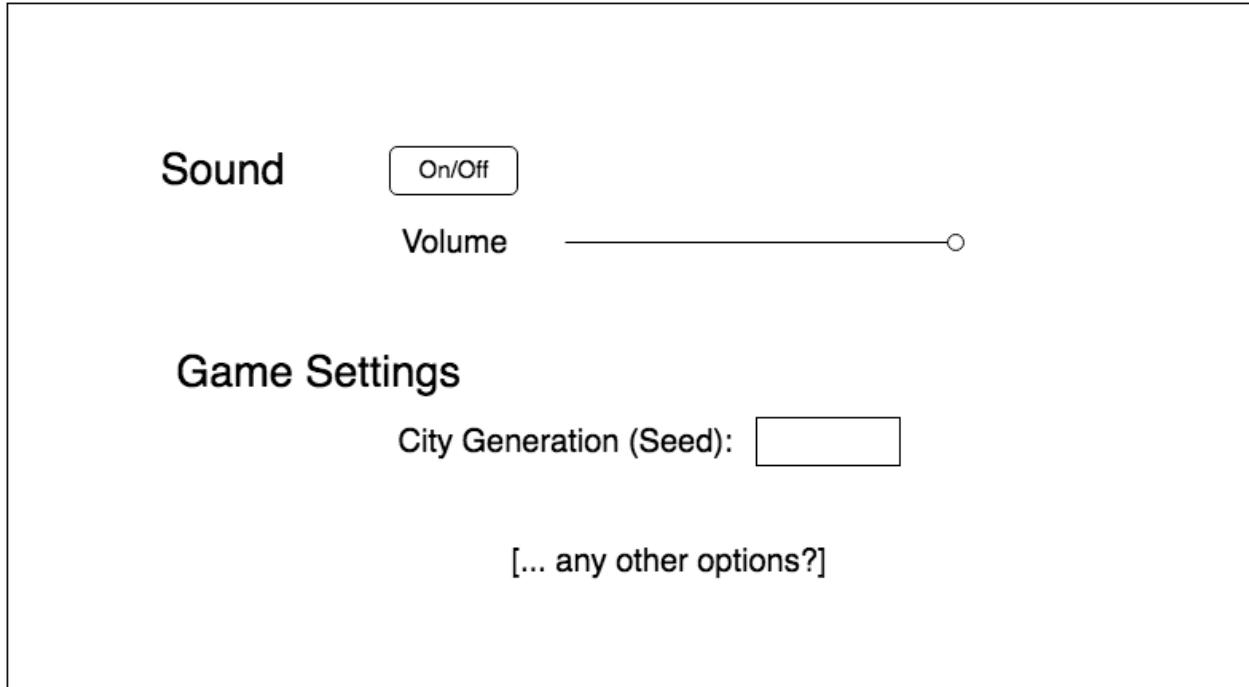


Figure 10.3.1

**10.3.2** The volume of the sound can be adjusted with a slider.

**10.3.3** The player can toggle the game's sound to be turned on and off.

## 10.4 Help Option

### Instructions:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse mattis porttitor fermentum. Ut lobortis accumsan rutrum. Donec ultrices mollis velit. Aliquam erat volutpat. Aliquam ultrices, risus in suscipit feugiat, est odio varius lacus, eget tempus dolor lacus ac odio. Nullam sit amet finibus sapien. Aliquam vehicula lectus et nunc auctor feugiat. Interdum et malesuada fames ac ante ipsum primis in faucibus. Etiam pulvinar efficitur arcu ac hendrerit. Mauris blandit feugiat fermentum. Quisque non dignissim nulla, in porttitor ligula. Integer lacinia sapien id lectus placerat sodales. Vestibulum sit amet ipsum magna.

Figure 10.4.1

This screen displays instructions on how to start playing the game.

## 10.5 Exit/End Credits Option

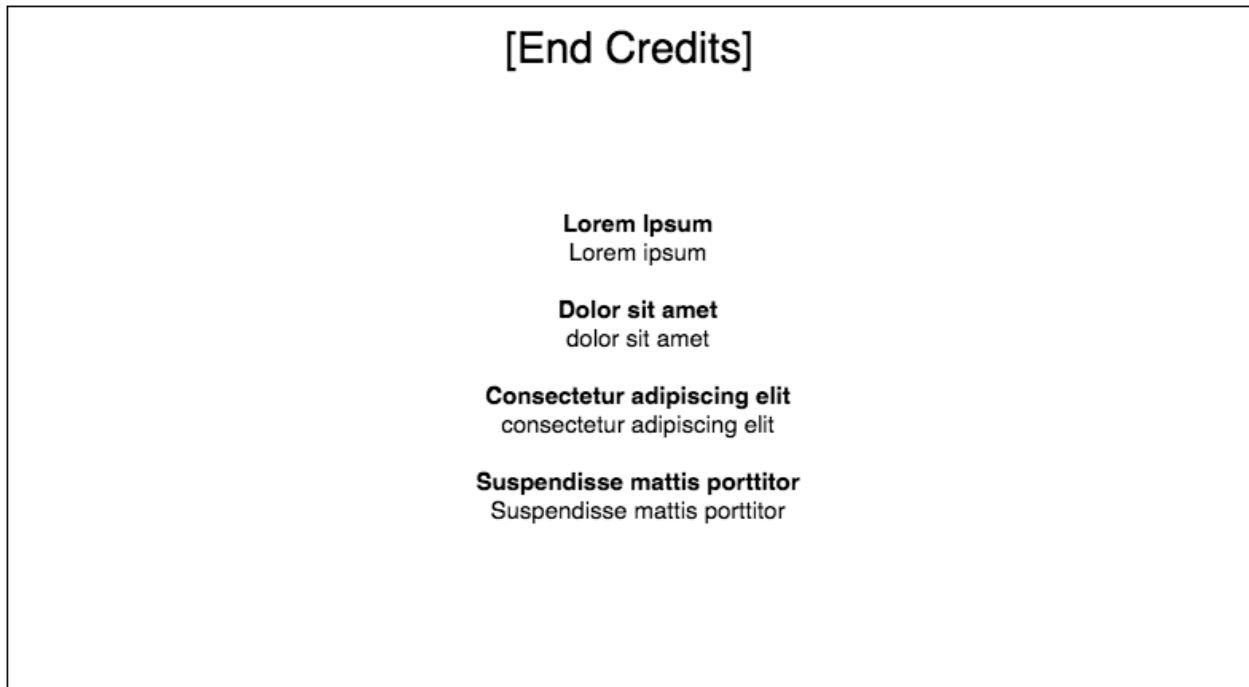
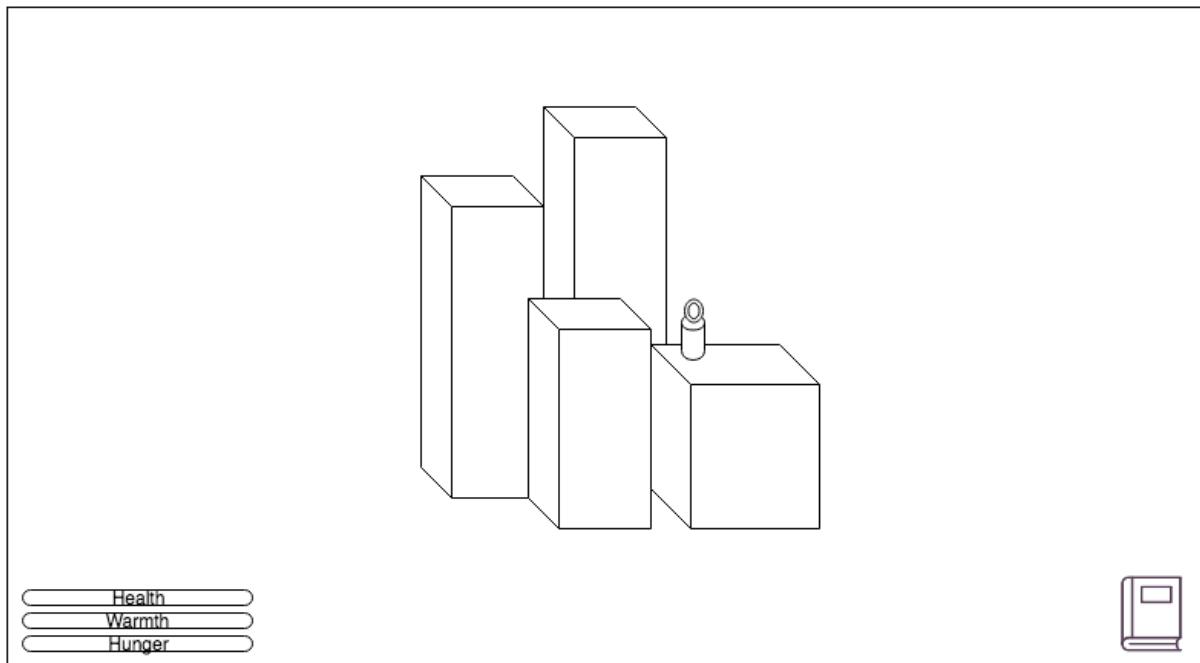


Figure 10.5.1

Displays team information and contribution to the game.

## 10.6 Game Screen/HUD



- 10.6.1** The player's health, hunger, and warmth resources are displayed
- 10.1.2** The **Radio**, **Inventory**, and **Crafting** buttons are displayed.
- 10.1.3** Pressing the **Radio** button displays a 3D radio model.
- 10.1.4** Pressing the **Inventory** button displays the inventory.
- 10.1.5** Pressing the **Crafting** button displays the crafting panel

## 10.7 Pause Menu

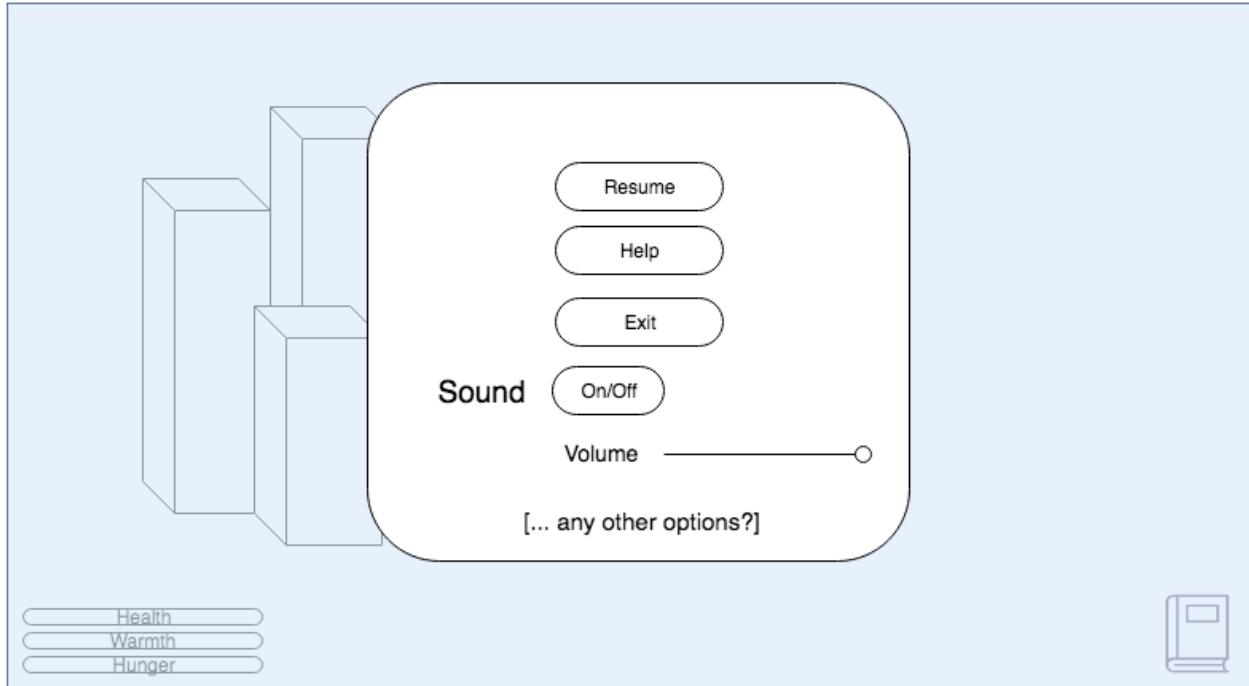


Figure 10.7.1

- 10.7.2** When the game is paused no player stats deplete.
- 10.7.3** When the game is paused the weather does not update.
- 10.7.4** When the game is paused the player can not move.
- 10.7.5** When the player presses Resume, the pause menu closes and stats, weather, and movements again update.

## 10.8 Inventory & Crafting Menus

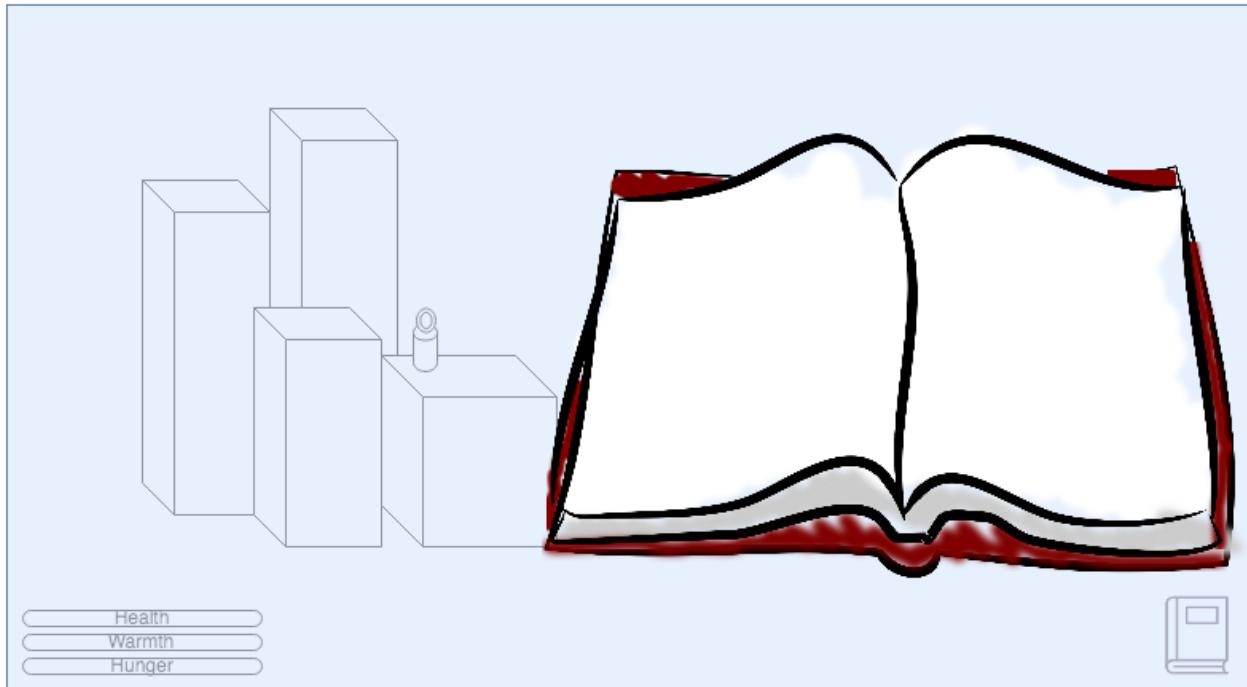


Figure 10.8.1

- 10.8.2** Hovering over an item in the inventory displays its name
- 10.8.3** Pressing on an item in the inventory opens a subpanel displaying information about the item.
- 10.8.4** Pressing on the crafting button on the inventory display opens the crafting panel.
- 10.8.5** Pressing the Back to Main button closes the inventory display.
- 10.8.6** Pressing on a recipe will display the ingredients needed for the recipe
- 10.8.7** If a recipe is craftable, it will be white and the Begin Crafting button will appear
- 10.8.8** If a recipe is uncraftable with the current inventory, it will be grayed out
- 10.8.9** If there is insufficient amounts of an ingredient for a recipe, the ingredient will be grayed out
- 10.8.10** Pressing on the Begin Crafting button will begin the crafting process

## 10.9 Radio Menu

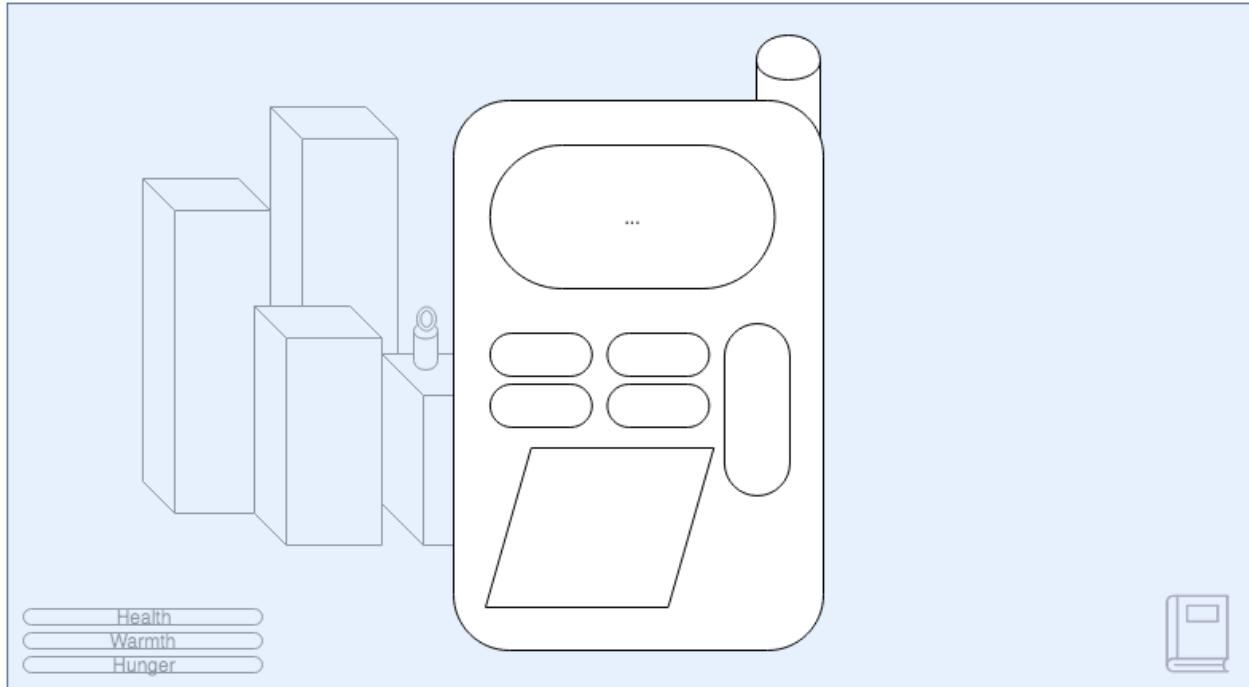


Figure 10.9.1

**10.9.2** Pressing the up volume button increases radio volume

**10.9.3** Pressing the down volume button decreases radio volume

**10.9.4** Pressing the on/off button turns the radio on or off

**10.9.5** Clicking and dragging on the dial will turn the dial and change the channel the radio is playing

**10.9.6** When the dial is turned, the red line showing the channel the radio is on moves as well

## Non-Functional Requirements

### 1. System Requirements

#### a. Minimum System Requirements

OS: Mac OS X 10.9.4 or Window 7

Processor: 2 GHz

Installed RAM: 8GB

Available Hard Disc Space: 2GB

Video Card: Shader version 2.0 capable video card

## b. Recommended System Requirements

OS: Mac OS X or Windows 10

Processor: 3 GHz

Installed RAM: 16GB

Available Hard Disc Space: 2GB

Video Card: nVidia GeForce 8500 / ATI Radeon HD 2600

## 2. External Requirements

- a. Keyboard
- b. Mouse or trackpad

## 3. Performance

- a. The game runs at an average of at least 60 frames per second on a machine with the recommended system requirements
- b. The game runs at an average of at least 40 frames per second on a machine with the minimum system requirements

# Appendix

## Glossary

<b>Animal Crossing: New Leaf</b>	Life simulation game by Nintendo. <a href="#">Wikipedia Link</a>
<b>block</b>	A sub-group of buildings within a district.
<b>building</b>	A structure placed within a district.
<b>carousel</b>	A list of sound files which play in a looping sequence. Tracks are added to and removed from the carousel to be played.
<b>diametric</b>	
<b>district</b>	An area of the city with a distinct look and shape.
<b>HUD</b>	Heads up display. The interface displayed to the user in the main game view.
<b>inventory</b>	A structure for storing items made up of slots.
<b>Lorenzian</b>	Uses a system of ordinary differential equations first studied by Edward Lorenz. It is notable for having chaotic solutions for certain parameter values and initial conditions
<b>note object</b>	Any type of in-game object whose interaction triggers a note GUI (includes items modeled as notebooks, letters, etc.)
<b>Poisson Distribution</b>	Randomly distributed points that utilizes minimum distances during placement
<b>raycast</b>	An infinitely long line that is cast from a point, used to check for collisions in Unity.
<b>resource</b>	A statistic that the player must keep high to survive. Health, Hunger, and Warmth.
<b>roguelike</b>	A game featuring procedurally generated levels. Upon death, it requires the player to start the game over from the beginning.
<b>slot</b>	A unit of storage in an inventory that can contain an item.
<b> Voronoi diagram</b>	Partitioning of a plane into regions based on distance to points in a specific subset of the plane.
<b>Unity3D</b>	The game engine used by Highwater
<b>Unity distance unit</b>	The measure of distance within the Unity Game Engine. 1 meter.