Game Design Document

The Last Tide

Overview

Narrative overview

In their travels the protagonist discovers a never mapped **archipelago**, here they will find an **ancient civilization** that was cursed by a deity, the **Protector**, who initially only wanted to help the islanders with **artifacts** that should have brought to peace and prosperity, but then became furious at them because the used said artifacts for different purposes, the worst offender being a “ritual” where the islanders combined all 3 major artifacts to become **ageless**.

This archipelago, that lies outside of time, is composed **mainly of 3 islands**:

- An island that represents the **body**, with lots of flora and inhabitants whose body was distorted by the curse;
- An island that represents the **mind** with more artificial structures and inhabitants whose mind was corrupted by the curse;
- An island that represents the **soul** with almost no vegetation, dilapidated buildings and inhabitants whose soul was removed.

At the center of this archipelago there is a **clock tower** that was initially a place of worship where the protector would “hide” their other artifacts, one of these being the clock that later allows the player to remember things **throughout the loops**.

Gameplay overview

The Last Tide is an **immersive third person adventure** set on a series of **islands** stuck in a time loop.

- The **loop** is made up of **four time slots** that are chosen by the player when he wants to leave the island. When this happens the **time changes** to the next time slot. The player can stay on the same island changing time slots by **resting** on it;
- The player **explores** the archipelago, on its 4 islands, trying to catch its **secrets**;
- **NPCs** allow the player's progress through the story by discovering useful information through **dialogues** or **puzzle** solving. The puzzles are solved through the aid of the artifacts used for the ageless ritual;
- Their goal is, through their own wits, to **stop the loop** and **save everyone** from the final judgment.

Gameplay

Player

Movement

The movement of the player is composed by:

Walk and run

- To navigate the game's environment at a **walking** pace, players will use a set of Movement Controls;
- For a faster pace, players can hold down the run control while pressing the movement buttons. This action shifts the character from walking to **running**.

Jump

- Players can initiate a jump by pressing the Jump button, elevate their character along the vertical axis;
- The height and distance of the jump are directly **influenced** by the player's current movement state, increasing the horizontal distance if the character is running.

Player Controls		
Control name	***Keyboard***	***Controller***
Movement	[W][A][S][D]	LS Axis
Run	[Shift]+[W][A][S][D]	RT
Jump	[Spacebar]	A

Character Variables	
Variable Name	**Value**
`walkingSpeed (Maximum Walking Speed)`	500 cm/s
`runningSpeedMultiplier (Maximum Running Speed)`	800 cm/s
`jumpForce (Maximum Jump Speed)`	620
`sextantSpeedMultiplier`	500 cm/s

Camera

Third-person camera located behind the player. The camera features a camera movement lag and follows the logic below:

Camera behavior		
Status	***Description***	***Notes***
Stationary character - no interaction	Camera can rotate around the player without rotating the character	
Moving character - no interaction	Rotating the camera changes the direction in which the player is going, keeping the front of the character always where the camera is looking	By pressing the Backward movement button, the character runs towards the camera
Stationary character - moving an object	Character rotates according to where the camera is pointing	
Moving character - moving an object	Rotating the camera changes the direction in which the player is going, keeping the front of the character always where the camera is looking	When the Backward movement button is pressed, the character walks backwards with his back to the camera
Stationary character - lantern equipped	Character rotates according to where the camera is pointing	
Moving character - lantern equipped	Rotating the camera changes the direction in which the player is going, keeping the front of the character always where the camera is looking	When the Backward movement button is pressed, the character walks backwards with his back to the camera
Stationary character - nautical sextant equipped	Camera can rotate around the player without rotating the character	
Moving character - nautical sextant equipped	Rotating the camera changes the direction in which the player is going, keeping the front of the character always where the camera is looking	By pressing the Backward movement button, the character runs towards the camera
Nautical sextant interactions	Camera turn in first person, see through the nautical sextant	Movement disabled
Stationary character - hurdy gurdy equipped	Camera can rotate around the player without rotating the character	
Moving character - hurdy gurdy equipped	Rotating the camera changes the direction in which the player is going, keeping the front of the character always where the	By pressing the Backward movement button, the character runs towards the camera

	camera is looking	
Hurdy gurdy interactions	Camera can rotate around the player without rotating the character	

Player Controls		
Control name	**Keyboard**	**Controller**
Camera	[Mouse Movement]	RS Axis

Camera			
	X	Y	Z
Location	0.0	27.0	0.0
Rotation	0.0	0.0	0.0
Scale	1.0	1.0	1.0

Camera boom			
	X	Y	Z
Location	0.0	0.0	65.0
Rotation	0.0	0.0	0.0
Scale	1.0	1.0	1.0

Camera boom variables	
Target arm length	320.0
Camera rotation lag speed	20.0

Boat

The boat floats in the water and reacts to waves. The boat **cannot** capsize.

Loop

- The game operates on a loop structure, comprising **four distinct time slots**. Failure to fulfill the **resolution path** within these slots results in the **end of the loop**;
- Time within the game progresses either when the player **docks** at an island or chooses to **rest** there;
- Upon the end of a loop or through the Pocket Watch, the loop **resets**, and the player is transported back to the Docking menu, retaining the **knowledge** and insights acquired in previous loops.

Docking

- Players can **select** their destination from a UI menu while aboard a boat, triggering two cutscenes that depicts the journey from the current **port** to the selected one;
- As the boat approaches the **new dock**, the time slot **switches to the next time slot**;
- When the player is at the 4th time slot and tries to move to another island, the **cutscene** related to the **final judgment** starts. The loop **restarts**.

Campfire

- The player can rest on a campfire to progress to the **next time slot**;
- When the player is at the 4th time slot and tries to move to the next slot, the **cutscene** related to the **final judgment** starts. The loop **restarts**.

First Loop

- On the first start of the game, without a previous save, begins what we consider to be our tutorial i.e. the **first loop**;
- **Only in this loop** the player starts directly to a **specific island** and not from the Docking menu mentioned just above.

Saving system

- Each time the player progresses to the **next time slot**, either by changing island or resting on the same one, the game is **saved**;
- If the player **reset** the loop, with the Pocket Watch, all that has happened in the current loop is **saved**;
- A **quick load-option** is present, see UI Section (load last checkpoint), that will load the game at the beginning of the current time slot.

Resolution path

- When the player **completes** all Main quests and the **final puzzle** at the Clock Tower, the game transitions to the final cutscene;
- After the final cutscene the game **returns** to the Main Menu;

- At the Main Menu, the player has the option to **load progress** at the **beginning** of the last time slot, allowing them to revisit and complete any remaining content or explore the game world further.

Interaction system

- The interaction system allows players to press the Interaction button to **engage with entities** when they are nearby;
- An **interaction prompt** (Interaction button) appears over the entity and remains as long as the interaction is possible as explained in the UI Section.

Player Controls		
Control name	**Keyboard**	**Controller**
Interaction	[F]	X

Collectable items

Upon interacting with a collectible object, it is incorporated into the player's **inventory**. The nature of the object—whether permanent or temporary—dictates its duration within the inventory, based on the **specific characteristics** of the item collected.

Movable items

- **Classification and interaction criteria**
  - Movable object must meet specific criteria for player interaction:
    - The player must be in contact with the ground;
    - The height difference between the object and the player must be less than `offsetMovableItems`.
- **Preview mode and placement process**
  - **Preview Display:** the movable object is shown in **semi-transparency** up to maximum distance from the player, specified as `previewDistMovableItems`;
  - **Collisions Suspension:** during the preview phase, the object temporarily **loses its physical properties**;
  - **Collision and Support Feedback:** if the previewed object is in a state of overlap with another object, the preview will **change color** to red (`hex #FF0000`). This visual cue signals to the player that the current position is **invalid for placement**;
  - **Rotation Capability:** during the **preview phase**, players can rotate the object by pressing and holding the provided Turn buttons. The object continues to rotate at a speed determined by its `RotationSpeed` for as long as the button is held down;
  - **Placement Confirmation:** placement of the object is **finalized** by pressing the Interaction button again, provided the object **is not** in a state of **overlap** (Collision and Support Feedback) and **is on a flat plane.** Successful placement **reinstates the object's physical properties**;

- **FreeCam Mod:** while holding one of these Movable items, players can **freely move the camera** around their character by holding down the FreeCam button;

- **Placement limitation**
  - All movable object are subject to placement limitations based on their position relative to the player's pivot:
    - If placed **below** the player's pivot point, objects behave as standard movable objects;
    - If placed **above** the player's pivot point, an offset limitation (`offsetPlaceMovableItems`) is enforced, preventing placement if the offset is exceeded;

Object Placement Controls		
Control name	***Keyboard***	***Controller***
Interaction	[F]	X
Turn Clockwise	[Q]	DPAD_right
Turn Counterclockwise	[E]	DPAD_left
FreeCam	[Alt]	LT

Movable items Variables	
Variable Name	**Value**
`previewDistMovableItems`	600 cm
`offsetMovableItems`	-195.0 cm
`offsetPlaceMovableItems`	15.0 cm

Temporary inventory

- Some **items** are placed in a temporary inventory. Items can be collected through environmental interactions, specific puzzles or specific NPCs;
- This inventory is **lost** when the player reaches the **end of the loop**;
- This inventory can be accessed by pressing the relevant UI button within the Journal;
  - When the button is pressed, the items collected up to that time will be displayed as shown in the related UI screen.

Persistent inventory

- Some **items** are placed in a persistent inventory. Items can be collected through environmental interactions, specific puzzles or specific NPCs;
- These items are **saved forever** in the player's inventory;
- The player accesses this inventory by pressing the relevant UI button within the Journal.

Player Controls		
Control name	**Keyboard**	**Controller**
Tool Wheel	[Tab]	LB
Journal	[J]	Back (View)
Camera	[Mouse Movement]	RS Axis

Journal

- The protagonist's journal is a system **outside the loop logics**;
- **Information** in the journal is **maintained** after a loop ends;
- The journal saves all the information that **helps** the player keep **track** of what was **solved**, where, and any **plot/lore** turns;
- Information is added **automatically** by the game, when the player:
  - completes a puzzle;
  - talks to npc and manages to extract information relevant to his purpose;
  - reaches/discovers places useful for the plot/lore;
- Texts related to information are **fixed**;
- This Journal is displayed as explained in the UI Section.

The in-game Journal acts as a **dynamic repository** of the player's **discoveries and experiences**, cataloging a variety of information integral to both the **storyline and gameplay**. Here's a breakdown of how the journal functions, capturing the different nature of the information **it may** contains:

Lore and Worldbuilding Events (Lore)

This section of the journal chronicles the backstory and significant events that shape the game's universe. Entries are derived from the game's worldbuilding bible, with slight modifications for narrative flow. They're presented as written accounts, bringing the player up to speed on key historical moments such as the birth of an ancient civilization, the misuse of mystical artifacts, and cataclysmic events that have left the islands cursed.

Gameplay Progression

Within these pages, the journal tracks the progress of the overarching game narrative, including the machinations of a Protector shrouding the archipelago in darkness and the consequential time loops. The narrative unfolds in a sequential order, resetting with each loop until the player reaches one of the game's conclusions.

NPCs

The NPC section is a detailed database of the characters the player encounters, documenting:

- **Name:** Each NPC's identifier;
- **Role/Job/Title:** The specific function or occupation the NPC holds within the game's world;
- **Backstory:** A brief history of the NPC, adding depth to their character;
- **Routine:** Daily patterns or behaviors of the NPC, which may provide hints for player interaction;
- **Quests:** Notable tasks or missions associated with the NPC.

Quests

As players undertake various quests, the journal logs:

- **Title:** The name of the quest for quick reference;
- **Request:** The initial task or objective given to the player;
- **NPCs Involved:** Characters that are part of the quest narrative;
- **Activity:** A checklist of actions to be taken or that have been completed;
- **Conclusion:** The outcome of the quest;
- **Reward:** What the player receives upon successful completion.

Tutorials (Puzzle)

The journal also contains tutorial sections for game mechanics and items such as Lantern, Hurdy-Gurdy, Sextant, and other interaction tools. Each tutorial includes:

- **Title:** Puzzle name;
- **Instructions:** Guidance on how to use each item or engage in game interactions;
- **Images:** Illustrative visuals to assist with understanding.

Map

- The Map feature evolves in the Journal menu as the player progresses through the game;
- Initially, the map starts with **limited information**, revealing only the areas the player has **already explored**;
- As the player advances, discovers new locations, and completes various objectives, additional details and regions are **gradually unlocked** and displayed on the map as a UI element.

Quest System

The quest system offers diverse activities that can be completed in any order. Upon completion of all required activities within a quest, players receive a reward that can impact the game world and narrative.

Quests

- Composed of one or more **activities**;
- **No** specific order required for activity completion;
- Completion of all activities yields a **reward**.

Activities

Two main types: **persistent** and **non-persistent**:

- Persistent activities **remain completed** across loop resets;
- Non-persistent activities **reset** at the end of each loop.

Activity Types

Activities in quests require players to perform various interactions within the game world, including:

Dialogues:

- Reach a specific line in a conversation with NPCs.

Interaction Areas:

- Overlap with designated interaction zones within the environment.

Game Elements Activation:

- Trigger elements such as doors, buttons and crystals.

Item Collection:

- Gather specific items that contribute to quest solutions.

Rewards

Rewards are granted upon the completion of all quest activities and can include:

Spawn/Despawn Elements:

- Introducing or removing game elements (e.g., mesh objects, NPCs, area collisions, VFX) within **specific** time slots to create a dynamic environment.

NPC Movement:

- Relocating NPCs within the same level in real-time, altering their interactions and the environment around them.

Dialogue Changes:

- Modifying dialogue lines for specific NPCs within certain time slots, impacting the narrative flow and player interactions.

Sound Effects:

- Playing specific sounds.

NPC interaction

NPCs follow the logic of the interaction system, opening a **dialogue box** with **multiple choice** responses when the Interaction button is pressed near them.

Player Controls		
Control name	**Keyboard**	**Controller**
Interaction	[F]	X

Dialogue system

- Choices are navigated using the Navigate buttons and confirmed with the Select dialog option button, leading to outcomes that can either **continue** the conversation or **end** it with the specific dialogue option;
- Certain dialogue choices carry **consequences** that are **remembered** by the NPC involved;
- Dialogue information automatically appear in the **Journal** after the player **learns** about them, streamlining the process of tracking vital information for game progression;
- Moreover, specific choices may **activate events** such as starting or completing quests and influencing other elements in the game world;
- These dialogue boxes are displayed as explained in the UI Section.

Dialog Controls		
Control name	***Keyboard***	***Controller***
Navigate Options	[Mouse Movement]	DPAD_v
Navigate - Option Up	[W]	DPAD_up
Navigate - Option Down	[S]	DPAD_down
Select dialog option	[Enter]|[LMB]	X

Tools

Tools wheel

- By pressing the Tool Wheel button the tool wheel UI (located in the UI Section) and cursor are **displayed**;
- During this time, player interactions and tool usage are **disabled**;
- With the cursor, the player can select one of the tools on the wheel, to **equip** the selected tool;
- While the player is **holding a tool**, if selects via the tool wheel the same tool, it will **unequip**;
- It is also possible to select a single tool by a **shortcut** related only to the keyboard using the provided buttons.

Player Controls		
Control name	**Keyboard**	**Controller**
Tool Wheel	[Tab]	LB
Lantern shortcut	[1]	NA
Pocket Watch/Compass shortcut	[2]	NA
Sextant shortcut	[3]	NA
Hurdy-Gurdy shortcut	[4]	NA

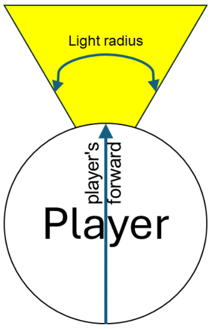
Pocket Watch/Compass

The Pocket Watch serves as a multifaceted gameplay tool with three distinct functions, enhancing both navigation and strategy within the game:

- **Time Tracking:** The Pocket Watch provides a visual representation of the **elapsed time** within the **current loop**. Specifically, the one of the hand indicates the player's **current time slot**, offering insight into the progression through the game's loop structure;
- **Loop Reset Mechanism:** By holding the Primary action button while the pocket watch is equipped, players have the ability to manually **restart the loop**. Allows players to reset their progress and experiment with different approaches or decisions, leveraging the knowledge gained from previous loops to influence the outcome of subsequent attempts.

Player Controls		
Control name	**Keyboard**	**Controller**
Primary action	[LMB]	RB
Tool Wheel	[Tab]	LB

Lantern



- The lantern is a **tool** that the player uses to **illuminate** the surrounding area with a spotlight, colored yellow;
- It is **equipped** through the tool wheel;
- When in use, it directs light **forward from the player's position**, casting light into the environment directly in front of them;
- The character's movement maintains a **constant orientation** relative to the camera.

Lantern Upgrade and Puzzle Interaction

As the player progresses, an upgrade unlocks a new interaction mode for the lantern, distinguished by a green light.
This mode allows the player to interact with specific objects or alter their properties or behavior.
Below are the lantern behaviors related specifically to this upgrade.

Moving large objects

- The lantern allows movement of **planks**, **crates**, and **ramps**;
- This behavior is a child of **Movable items** so it retains its own logic;
- The player can move these objects as long as the lantern upgrade has been obtained, regardless of whether the upgraded lantern is currently equipped:
  - If the player **has equipped** the lantern, it emits a yellow light. When interacting with large objects, the lantern's **light changes** to purple, indicating that the upgrade is active;
  - If the player **has not equipped** the lantern it is **automatically** equipped during interaction with large objects.

Charging Action

- The lantern also allows **interaction with specific objects** that **cannot** be moved by the Movable items system but whose logic can be changed;
- With the lantern equipped, the player can **hold down** the primary action button to charge the lantern's purple light;
- The duration of this charge is `chargeTime`;
- Once fully charged, **releasing** the button will trigger the interaction with the object;
- These objects include:
  - **Walking platform**: When the lantern interacts with the walking platform, it becomes a self-moving platform. The platform moves automatically, changing direction upon hitting walls or edges;
  - **Rotating platform**: These platforms rotate 90° each time the player interacts with them using another tool upgrade. When the lantern interacts with the rotating platforms, this functionality is **disabled**. This allows the player to **lock one and only one** platform in place, ensuring it remains **stationary**;

Lantern items Variables	
Variable Name	**Value**

previewDistMovableItems	600 cm
chargeTime	2 s

Lantern Controls		
Control name	***Keyboard***	***Controller***
Change Logic	[LMB]	[RB]

Nautical Sextant

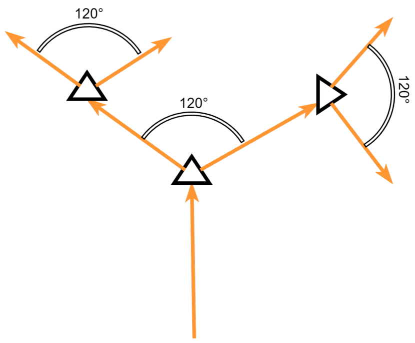
- The nautical sextant is a tool that the player can use to see objects and islands at a distance.
- It can be equipped from the tool wheel, when activated with the Zoom In/Out button, the character moves his arm as if to put it in front of his face.
  - The camera changes, switching to first person and creating a zoom of zoomSextant, by pressing the same button the player will go back to the normal camera setting;
  - When the player is looking through the sextant the player can move around the camera with the mouse and can move with a speed reduction of sextantSpeedMultiplier and cannot run or jump;

Sextant Upgrade and Puzzle Interaction

Upon receiving an upgrade, the sextant becomes instrumental in solving the "Lasers" puzzle, which involves guiding a laser beam through various points to achieve specific goals. The puzzle mechanics include:

- **Laser beam:** The beam needs to be activated by pressing the Interaction button near the **generation point**. The player can activate these generation points regardless of whether the Sextant with the upgrade is equipped.
  After the activation the player's goal is to direct the beam from the mentioned generation point to a designated **endpoint**, illuminating the path it traverses;
- **Mirror:** Mirrors are placed throughout the puzzle area. Players can move these mirrors on all axes allowed and rotate them on the z-axis. The main purpose of these mirrors is to redirect the laser beam in new directions based on the player's adjustments (on how to place down the mirrors see: Movable items);
  - **Specular reflection logic:** The laser beam, upon hitting a mirror, reflects in a manner consistent with the laws of specular reflection. This means the angle at which the beam hits the mirror (relative to the perpendicular of the mirror surface) is equal to the angle at which it reflects away;
  - **Duplicator mirror:** These duplicating mirrors are placed throughout the puzzle area. Players can move these mirrors on all axes allowed and rotate them on the z-axis. The main purpose of these specific mirrors is to duplicate the laser beam in two new directions. This mirror has a triangular shape, the laser beam enters from the base and exits at 120 degrees from the two cathets;

        - The laser beams always exit at the same angle relative to the forward of the mirror, so they are **not affected** by the angle of entry into the base;



    - **Riser mirror:** These mirrors are placed throughout the puzzle area. Players can move these mirrors on all axes allowed and rotate them on the z-axis. The main purpose of these mirrors is to redirect the laser beam 45 degrees upward;
    - **Mirror constraints:** Each puzzle comes with its own set of mirrors. These mirrors can be moved freely by the player but only within the boundaries of the puzzle area;
    - **Real-time reflection preview:** When a player is positioning a mirror, the game shows a preview of the laser's trajectory in real time. This preview helps the player understand the direction the laser will take once the mirror is placed;
- **Darkness:** It is present only in some puzzles involving lasers. If the player gets too far away from the light, generated by the laser or the crystals, they will be swallowed up by it. When the player is swallowed, the reset functionality goes into action.
    - The player can remain in the darkness to swallow Timer when the time's up the puzzle and the player will reset;
    - The area around the laser beam where the player is safe is `laserSafeZone`.
    - The area around the crystal where the player is safe is `crystalSafeZone`;
- **Illumination Crystals:** Certain crystals in the puzzle have the capability to illuminate when struck by the laser beam. This illumination creates a safe zone within the surrounding area, protecting the player from darkness. When the laser stops hitting them, the light fades fairly quickly. These crystals will NOT be moveable by the player;
- **Decaying Crystals:** These crystals are elements that are present in puzzles that have a timer duration. The decay starts when the player interacts with the generation point in the puzzle, which activates the puzzle timer. The decay causes the light emitted by the crystal to fade until turned off. The player can recharge the crystals and restart their timer by hitting them with a laser beam. When even one crystal in the puzzle goes out, the reset functionality goes into action;
    - They all have an identical "charge" `decayTimer` but the beginning said decay is customized per individual crystal `decayStaringTime`, within the puzzle, causes them to expire at different moments;

    - The intensity of the light decreases proportionally with the time remaining on the timer, at `blinkingTime` from the timer's end the crystal begins to blink, when the time is out the crystal turns off;
- **Puzzle reset functionality:** When the player is respawned due to staying in darkness for too long, the puzzle automatically resets. This reset includes the repositioning of mirrors to their original locations and the reinitialization of the laser and crystals to their default states.

Sextant Controls		
Control name	***Keyboard***	***Controller***
Zoom In/Out	[LMB]	RB

Character Variables	
Variable Name	**Value**
`sextantSpeedMultiplier`	500 cm/s

Camera Variables	
Variable Name	**Value**
`zoomSextant`	TBD

Decaying crystals variables	
Variable Name	**Value**
`decayTimer`	8 s
`blinkingTime`	2 s

Hurdy-gurdy

Initially, the hurdy-gurdy allowed players to enjoy the musical aspect for entertainment.

- Players can produce **seven distinct notes**, each mapped to a specific button;
- A new note can only be played once the preceding note's sound has concluded, with the system detecting the end of a note's sound to permit the next note's execution.

Hurdy-Gurdy Upgrade and Puzzle Interaction

Upon receiving an upgrade, the hurdy-gurdy becomes instrumental in solving the "Hurdy-Gurdy" puzzles, which involves using the sequence of notes as a key gameplay mechanic.

The puzzle mechanics include:

- **Musical Note Recognition:** The game incorporates a system capable of recognizing the specific musical notes produced by the player's hurdy-gurdy;
- **Sequence input mechanism:** Interface or control scheme that allows the player to input a sequence of musical notes, with feedback indicating the notes played. These notes and the key they are bound to are displayed as explained in the UI Section;

- **Puzzle Logic for Note Sequences:** Puzzles require the player to discover and play the correct sequence of musical notes;
- **Error handling and incorrect sequence feedback:** implement feedback and error handling for incorrect sequences. When the player plays the wrong sequence, the game should provide appropriate feedback, resetting the sequence once the correct sequence length is reached;
- **Note symbols:** 7 symbols, each representing a musical note, are present, both in the game and in the hurdy-gurdy UI section, to allow the player to understand what is the correct sequence of notes to play;
- The correct sequence allows the player progression in the puzzles:
  - **Doors:** These doors open when their symphony is played and stay open until the start of the next loop;
  - **Pillars:** These pillars are lowered or raised from the floor when their specific symphony is played;
  - **Rotating platforms:** Platforms on the ground that turn (90°) every time the player plays the right code. The platforms in the area of the puzzle must turn all at the same time and if activated by the lantern light upgrade (only one at the time) will not turn.

Puzzles

To see the entire puzzle file: Puzzles.

Specific interactable objects in puzzles:

Laser activated levers

- These levers are **activated** when the laser beam **hits their associated endpoint**, indicated by the light on the lever turning on;
- Once active, the player can pull the lever to **rotate the wheel** (see 2.2.1) as many times as they want;
- If the laser beam **stops hitting the endpoint**, the lever is **deactivated**, and the light on it changes color.

Wheel

- These wheels are stationary objects that **contain four different items** from the props list, with **only one** item usable at a time;
- The wheel can be activated by a **lever** next to it, but only **when powered** by a laser beam;
- Each activation **rotates the wheel** by 90°, **despawning** the currently displayed object and **spawning** the next one in the rotation on the wheel (2.2.1).

Buttons

- These buttons are present in each room of the wheel puzzle (see 2.2.1);
- The player must reach and press these buttons to open the door leading to the next room.

Tutorial

Because of the open world structure, the player is free to manage their experience as they wish. To overcome this problem, the tutorials are handled as below.

- When a **new game** is started from the Main menu, the player begins at the First Loop;
- The player is introduced to:
  - **Mechanics** related to player control;
  - Initial **puzzles** (Temple tutorial);
  - The **dialogue** system used to progress through quests;
- The player can progress through different **time slots** by interacting with the boat or campfire;
- Each island features a **Temple** that serves as a tutorial, helping the player:
  - Become familiar with the puzzle mechanics specific to that temple;
  - Unlock the upgrade related to the **tool** used in that temple;
- Only after completing the tutorial can the newly learned mechanics be used in puzzles **outside** of the tutorial;
- Each puzzle includes specific **triggers**:

  - Highlighting the basic mechanics to make the player aware of;
  - Specifying the end goal of the puzzle.

Development and Debugging Tools

Dialogue tool

To facilitate the creation of dialogues, a specialized tool is provided. This tool offers an interface for drafting dialogue texts and managing response options, including the ability to add or remove choices as needed. It features several key functionalities:

- **Dialogue Option Flags:** Certain responses can trigger specific actions:
  - **"End Dialogue":** This flag concludes the current dialogue and resumes normal gameplay;
  - **"Go to Dialogue X":** This option transitions the conversation to a different set of dialogue text and choices;
- **Actor Interaction Settings:** The tool enables adjustments to the game world in response to dialogue choices, such as:
  - Opening a specific door (Door Y, where Door Y can indicate a specific actor);
  - Spawning an item (Item X, where Item X can indicate a specific actor);
  - Adding an item (Item Z, where Item Z can indicate a specific actor) directly to the player's inventory;
- **Dialogue Option Visibility and Selection:** Each dialogue choice can be flagged to determine its availability. If marked as non-selectable, the option will neither be visible nor accessible to the player.

This comprehensive tool is designed to streamline the dialogue creation process.

Controls

Player Controls		
Control name	***Keyboard***	***Controller***
Movement	[W][A][S][D]	LS Axis
Run	[Shift]+[W][A][S][D]	L3
Jump	[Spacebar]	A
Camera	[Mouse Movement]	RS Axis
Primary action	[LMB]	RB
Interaction	[F]	X
Tool Wheel	[Tab]	LB
Journal	[J]	Back (View)
Pause	[Esc]	Menu (Start)

Lantern shortcut	[1]	NA
Pocket Watch/Compass shortcut	[2]	NA
Sextant shortcut	[3]	NA
Hurdy-Gurdy shortcut	[4]	NA

Movement Controls

Control	***Keyboard***	***Controller***	***Output***
Forward	[W]	LS Axis	Character will move straight according to where the camera is pointing
Left	[A]	LS Axis	Character will move left according to where the camera is pointing
Backward (walking, hurdy gurdy equip., nautical sextant equip.)	[S]	LS Axis	Character will turn with his face to the camera and go straight ahead, without the camera moving and going behind him
Backward (moving object, using lantern)	[S]	LS Axis	Character does not turn around but walks backwards with his back to the camera.
Right	[D]	LS Axis	Character will move right according to where the camera is pointing

Lantern Controls

Control name	***Keyboard***	***Controller***
Change Logic	[LMB]	[RB]

Sextant Controls

Control name	***Keyboard***	***Controller***
Zoom In/Out	[LMB]	RB

Hurdy-Gurdy Controls

Control name	***Keyboard***	***Controller***
C (DO)	[Z]	RB
D (RE)	[X]	RT

E (MI)	[C]	LT
F (FA)	[V]	DPAD_left
G (SOL)	[B]	DPAD_down
A (LA)	[N]	DPAD_right
H (SI)	[M]	DPAD_up

Object Placement Controls		
Control name	***Keyboard***	***Controller***
Interaction	[F]	X
Turn Clockwise	[Q]	DPAD_right
Turn Counterclockwise	[E]	DPAD_left

Dialog Controls		
Control name	***Keyboard***	***Controller***
Navigate Options	[Mouse Movement]	DPAD_v
Navigate - Option Up	[W]	DPAD_up
Navigate - Option Down	[S]	DPAD_down
Select dialog option	[Enter]|[LMB]	X

Level Design

- Regarding world building refer to the specific document:
    - Level Design Document
    - Metrics

Narrative

- Regarding world building refer to the specific document:
    - Worldbuilding bible
- Regarding NPCs refer to the specific document:
    - NPCs
- Regarding Quests refer to the specific document:
    - Quests

UI/UX

Regarding UI/UX refer to the specific document:

- UI/UX

Sound design

Due to the game's player immersion, we avoid using OSTs. Instead, we incorporate spatial audio triggered by player actions and locations.
Below are the types of sounds used, their corresponding triggers, and links to the audio files.

	Sound Effect
Trigger	Reference
Walk	https://freesound.org/people/LittleRobotSoundFactory/sounds/270416/
Door open	https://drive.google.com/file/d/1x06-zew7Bw-iPq0mnHnRP_c2azJ-fjW_/view?usp=drive_link
Wheel rotation	https://drive.google.com/file/d/15AAqNSTv45yfSqj1SkpXAva9sfNvcrGT/view?usp=drive_link
Proposta 2	https://drive.google.com/file/d/1c7sCaZtB2dT--L5I3Ne2RBt83GmT9dpM/view?usp=drive_link
Player interacts with a lever/button	https://freesound.org/people/BenjaminNelan/sounds/410359/
Near the sea wave motion	https://drive.google.com/file/d/1IcJH5ucSxW2A9OIqy1NJHMKanfP6c4Or/view?usp=drive_link
Near the sea wave motion	https://drive.google.com/file/d/1IcJH5ucSxW2A9OIqy1NJHMKanfP6c4Or/view?usp=drive_link
Hurdy-Gurdy note C	https://drive.google.com/drive/folders/1A4FQ4BGn9VurkMCxiFLpH4-wGeV-QWmR
Hurdy-Gurdy note D	https://drive.google.com/drive/folders/1A4FQ4BGn9VurkMCxiFLpH4-wGeV-QWmR
Hurdy-Gurdy note E	https://drive.google.com/drive/folders/1A4FQ4BGn9VurkMCxiFLpH4-wGeV-QWmR
Hurdy-Gurdy note F	https://drive.google.com/drive/folders/1A4FQ4BGn9VurkMCxiFLpH4-wGeV-QWmR
Hurdy-Gurdy note G	https://drive.google.com/drive/folders/1A4FQ4BGn9VurkMCxiFLpH4-wGeV-QWmR
Hurdy-Gurdy note A	https://drive.google.com/drive/folders/1A4FQ4BGn9VurkMCxiFLpH4-wGeV-QWmR
Hurdy-Gurdy note H	https://drive.google.com/drive/folders/1A4FQ4BGn9VurkMCxiFLpH4-wGeV-QWmR
Hurdy-Gurdy puzzle melodies	https://drive.google.com/drive/folders/1dn1OmrBJL74OCZa80ysg1RRgEkzEJIJ4
Hurdy-Gurdy quest melodies	https://drive.google.com/drive/folders/1VRUCgqV1AjyvQ1FWGqinw6L49MXIbS1M
Ambient A	https://drive.google.com/file/d/1yBgq9Tka9wNx_XcPzK0yMxmdEEeh6NMN/view?usp=drive_link
Ambient B	https://drive.google.com/file/d/1H41fsmFjOTraMOW6IE627

	z4I3gsqBnSv/view?usp=drive_link
Ambient C	https://drive.google.com/file/d/1R8nEMTe8l_L5p5kDrcV0L3mVLaRO5h1c/view?usp=drive_link
Ambient D	https://drive.google.com/file/d/1cQG3-BJOXSH4_sR20vyNuudT2PKw9Jow/view?usp=drive_link
Ambient E	https://drive.google.com/file/d/1IrSj1VZ_xNn1rjm_B8YUW7Kh0AqMLQwt/view?usp=drive_link