

Embers of Beyond

Embers of Beyond is a puzzle platformer set in a mystical ghost realm inspired by Asian folklore. Players control a lantern-bearing girl who transforms into various characters along her journey to solve intricate environmental puzzles. As she navigates this eerie and enchanting world, she unravels mysteries and uncovers hidden truths.

Design · Code · Concept Art

Erin Hao

3D Art

Erin Hao, Synty®

Video: https://www.youtube.com/watch?v=yLQnM-1F8_g

Game: <https://ninetail9.itch.io/embers-of-beyond>



3D PUZZLER · NARRATIVE-DRIVEN PLATFORMER · AGE 6+



Mechanics

Resource & Ability

◆ Emberlight

Slowly **burns away**, player must **keep the light** to ward off spirits, and travel in the dark.



◆ Transformation

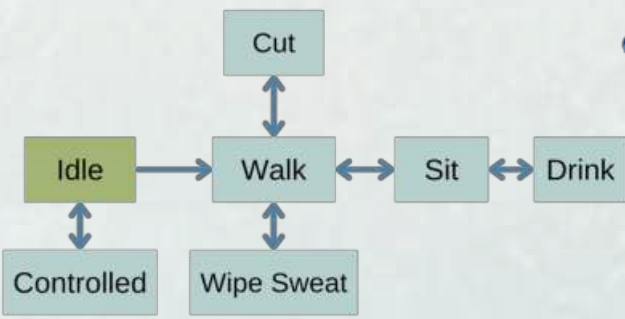
Use **Emberlight** to briefly **posses** nearby characters, and use their **abilities**.



Transformable Characters

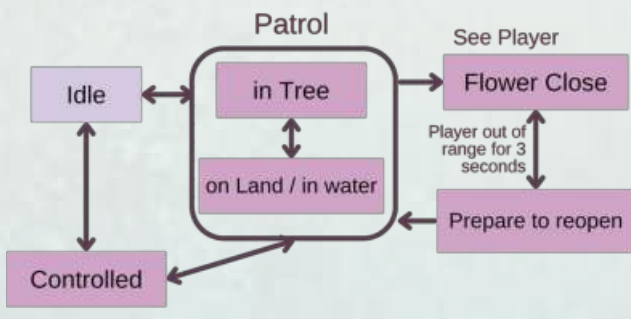
◆ Lumberjack

A ghostly labourer doomed to endlessly chop wood. Transforming into the Lumberjack allows you to **cut down bamboo**.



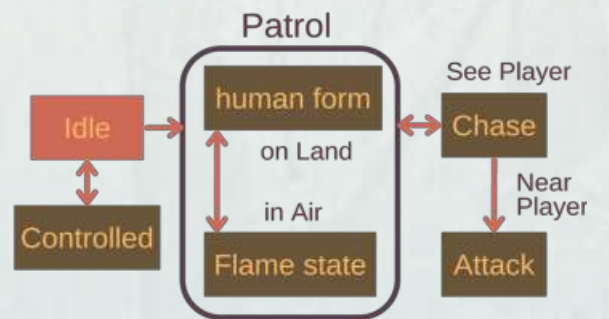
◆ Bloom Spirit

A spirit born from withered flowers. Transforming into the Bloom Spirit lets you **glide through trees (peach) or across water (lotus)**.



◆ Will-o'-the-Wisp Guard

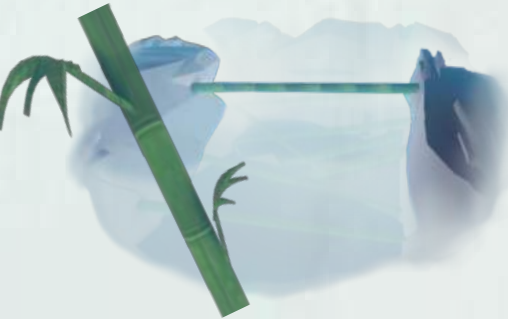
A spectral guardian of the ghost realm. Transforming into the Guard enables **light sources teleportation between**.



Interactables

◆ Bamboo/Poles

Chop them down as Lumberjack to create **new paths**.



◆ Breakables

Obstacles that could be chopped by Lumberjack.



◆ Runestone

Checkpoint/sanctuary gradually restores Emberlight, and **foreshadows** what lies ahead.



◆ Flickering Embers

Collect to replenish **Emberlight**.



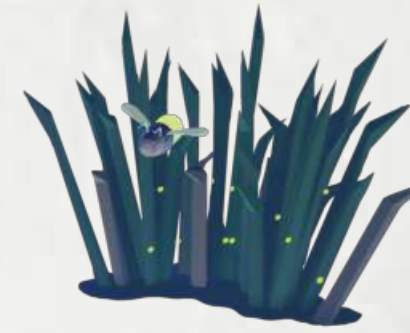
◆ Ghostly Wind

Alters speed, either hastening or hindering movement.



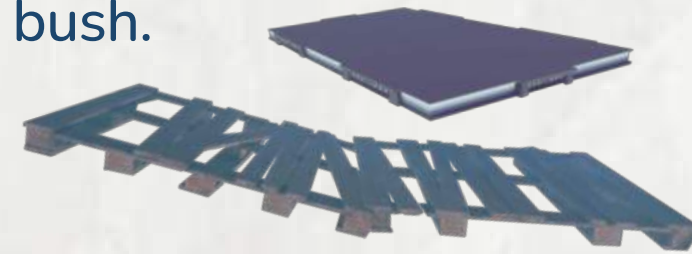
◆ Hazards

Deal damage. Beware of bamboo shoots, will-o'-wisps, winds, chasms, and other dangers.



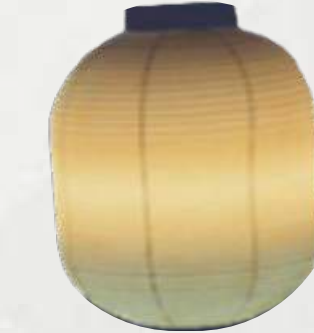
◆ Special Terrain

Travel in stealth through water/tree bush.



◆ Will-o'-the-Wisp Lights

Teleport using these as the Guard.



Level Design

Tutorial ★

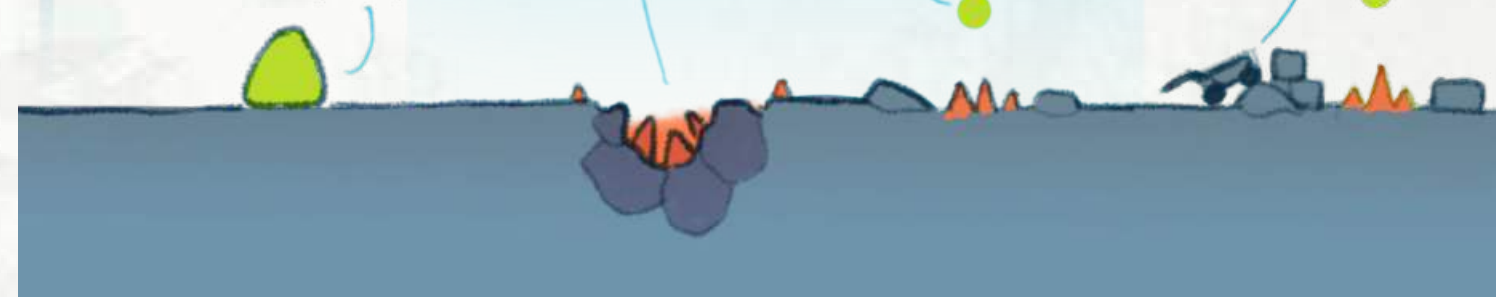
foggy, mysterious, unknown, unsettling, spacey

Encounter: hazard - bamboo spike
1st time - in a pit
2nd time - on flat ground

Encounter: collectible resource
Put over a hazard to indicate its property

Cart, foreshadows traces of living

Encounter: respawn point



Save 1 ★★★★★

Strangely attractive, complex bamboo, confined spaces

Encounter: Spirit - Cutter (default state cutting bamboo as clue)

Goal: travel across cliff
learn to control cutter and use it's skill to cut bamboo (save point allows exploration time)

Encounter: hazard - cliff

resource as clue

Encounter: other cuttables

Encounter: architecture
Both the ground and roof are walkable, alternative path of different elevation

Goal: top of roof
Has to cut in the right order, or the objects block each other

Let player think roof as right path again, but lower floor is the actual way



Save 2 ★★★★★ Pace-breaker Level

Serene, relaxed, safe, smooth

Encounter: wide river, can't jump to cross

1st time - given bridge(indicate safe way)
2nd time - cut bamboo

Fast Pace
(Semi safe area with no fatal hazard, for player to boost through)

Encounter: speed change mechanics

Encounter: sudden event
Bamboo falls when triggered, creates temporary slow region

Speeds up

Slows down

Speeds up

Slows down

Sudden Stop



Save 3 ★★★★★

Bare rocky Area, foreshadows difficulty and danger ahead

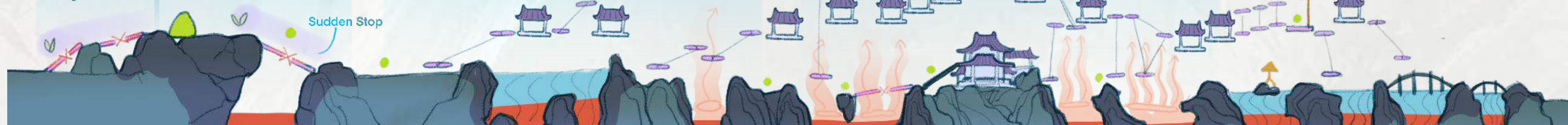
Encounter: one-time platform

1st time - over elevated ground (safe)
2nd time - over cliff

Sudden Stop

Encounter: moving platform
1st time - 1 platform, ground >> over water
2nd time - 1 platform over water
3rd time - 2 platforms over timed hazard

Encounter: Timed Hazard



Alternative Path

High Path - Safer travel, less resource
Ground Path - High risk, high reward

Goal: get to roof
Has to go pass lamp post to get down to the cutter



Experience Goals

Immerse players in a world of **lurking dangers and haunting beauty**. Explore the ghostly city, uncover its secrets, and question why the sister may no longer wish to return to the living world.

Story

Bound by a family pact, the protagonist must venture into the **ghost realm** to **find her lost sister**, who vanished years ago while fulfilling duty as a spirit judge. Haunted by dreams of a desolate landscape, the protagonist recognizes it as the ghost realm and seizes the chance to infiltrate it during the Ghost Festival, determined to bring her sister back before closes after the festival.

Characters

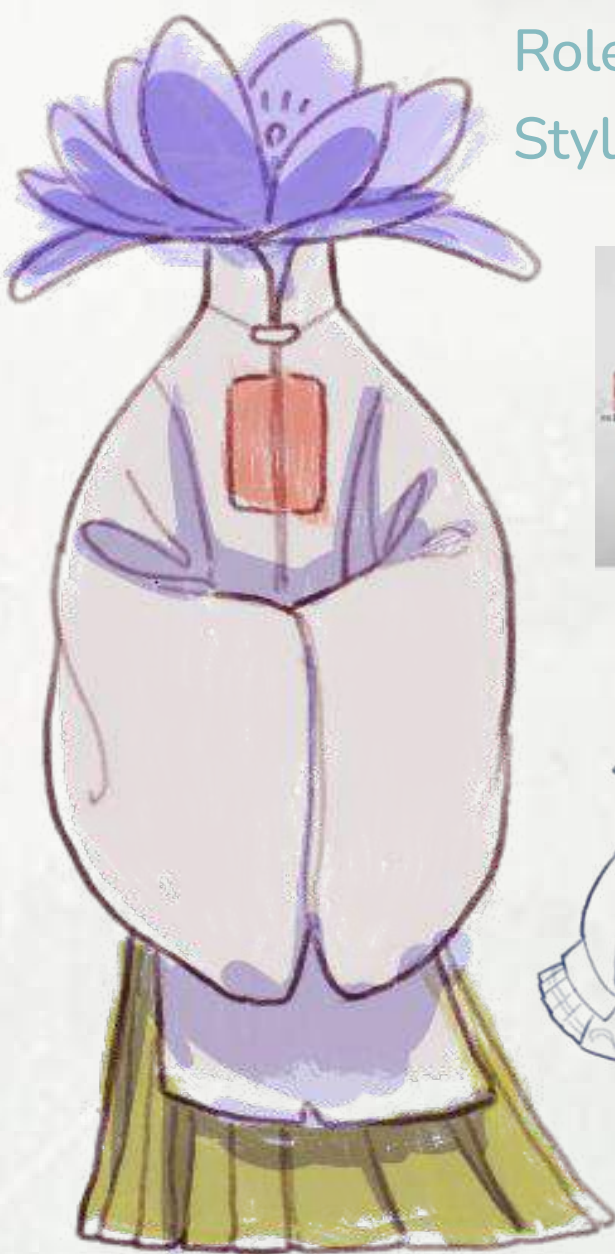
Art Style

Inspired by Asian folklore, emphasizing elegance and mystery.



Main Character

Role: Lantern-bearer, Spirit Judge
Style: Soft colors, traditional attire.
Lantern symbolizes health.



Bloom Spirit

Role: Guardian of withered flowers
Style: Petal-like layers, ethereal forms.



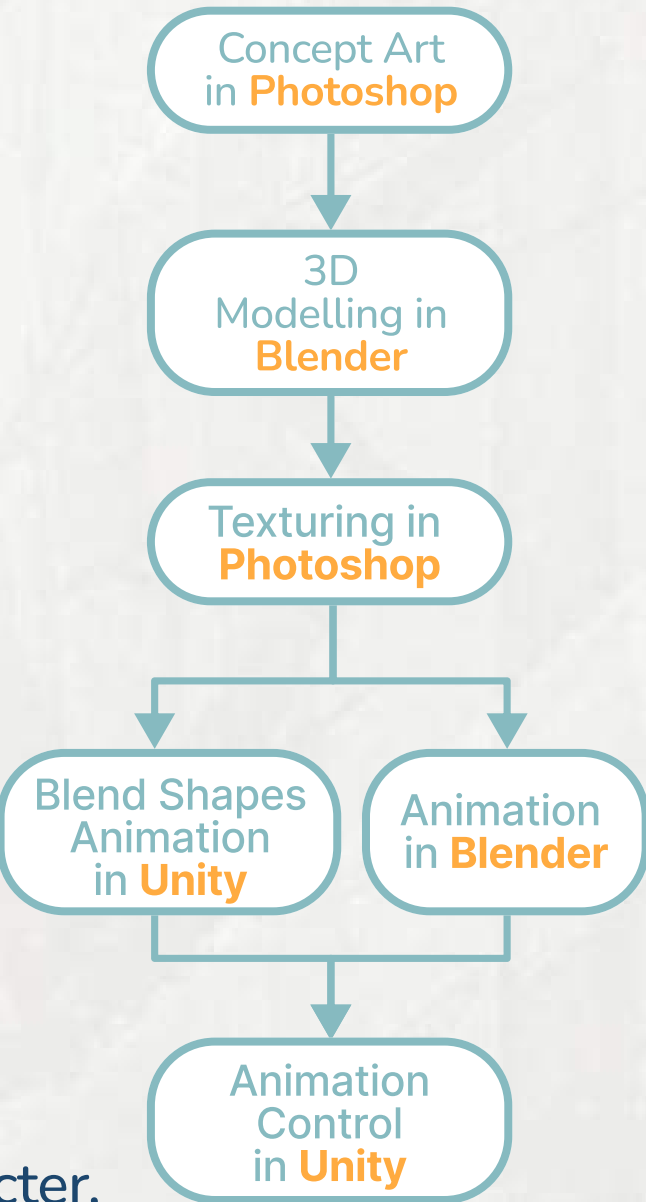
Will-o'-the-Wisp Guard

Role: Ghost realm protector
Style: Lantern head, dark tones



DCC Workflow

I created the designs, 3D models, and animations of the main character, bloom spirits, and guards to maintain a cohesive style and storytelling vision.



Environment

Storytelling

The environment tells the story of a place suspended between life and death. Overgrown bamboo forests and abandoned structures symbolize a world lost to time. Lantern-lit paths offer safety, while crumbling ruins and remnants hint at a deeper history. Color and lighting reflect the protagonist’s journey—from hope to the chilling realization of what's ahead.



Design

Unity's HDRP creates a mystical, otherworldly realm, using advanced lighting and atmospheric controls to balance beauty and dread.

- ◆ **Lighting & Shadow:** Soft lighting and enhanced shadows illuminate lanterns’ ethereal glow, while dynamic light conveys a serene yet haunting atmosphere.
- ◆ **Volumetric Fog:** Calibrated to evoke isolation and mystery, it emphasizes distant silhouettes and layers the landscape.
- ◆ **Post-Processing:** Subtle bloom, ambient occlusion, and color grading evoke the spectral, dreamlike essence of a forgotten realm.

Props

Handcrafted 3D props like runestones, bamboo fences, stone lanterns, and broken carts enrich the narrative, reflecting the realm's decay and forgotten past.

Carved props with inscriptions were created using SVG-to-mesh conversion and optimized with Boolean operations, serving as mystical checkpoints and markers throughout the world.



UI

UI elements are inspired by traditional Asian patterns, with shapes and colors designed to blend harmoniously into the world’s aesthetics. The minimalist icons and lantern-based HUD maintain visual clarity while preserving the cultural essence of the game’s setting.



Music & Sounds

Traditional Asian instruments like the pipa, bamboo flute, and guqin are used for interactive SFX, creating an authentic, immersive soundscape. Ambient sounds like rustling bamboo, distant streams, and soft footsteps enhance the atmosphere, adding realism to the environment.

Programming

AI with FSM

FSM (Finite State Machine) is used to define AI behaviors for multiple characters, breaking actions into distinct states like **Idle**, **Cut**, **Walk**, etc. Character's state transitions are handled based on timers, conditions, or random choices.

Taking a Lumberjack as an example:

- ◆ **State Enum:** Defines each character's possible states (e.g., Idle, Walk, Sit).
- ◆ **State Functions:** Each function (e.g., StateIdle, StateCut) manages a specific behavior.
- ◆ **Transitions:** Controlled in FSM(), enabling smooth shifts between states based on predefined triggers.

This approach ensures modular and scalable AI behavior across different characters.



```
Cutter.cs (Lumberjack)

void FSM()
{
    switch (CurrentState)
    {
        case State.Idle:
            StateIdle();
            break;

        case State.Cut:
            StateCut();
            break;

        case State.Walk:
            StateWalk();
            break;

        case State.WipeSweat:
            StateWipeSweat();

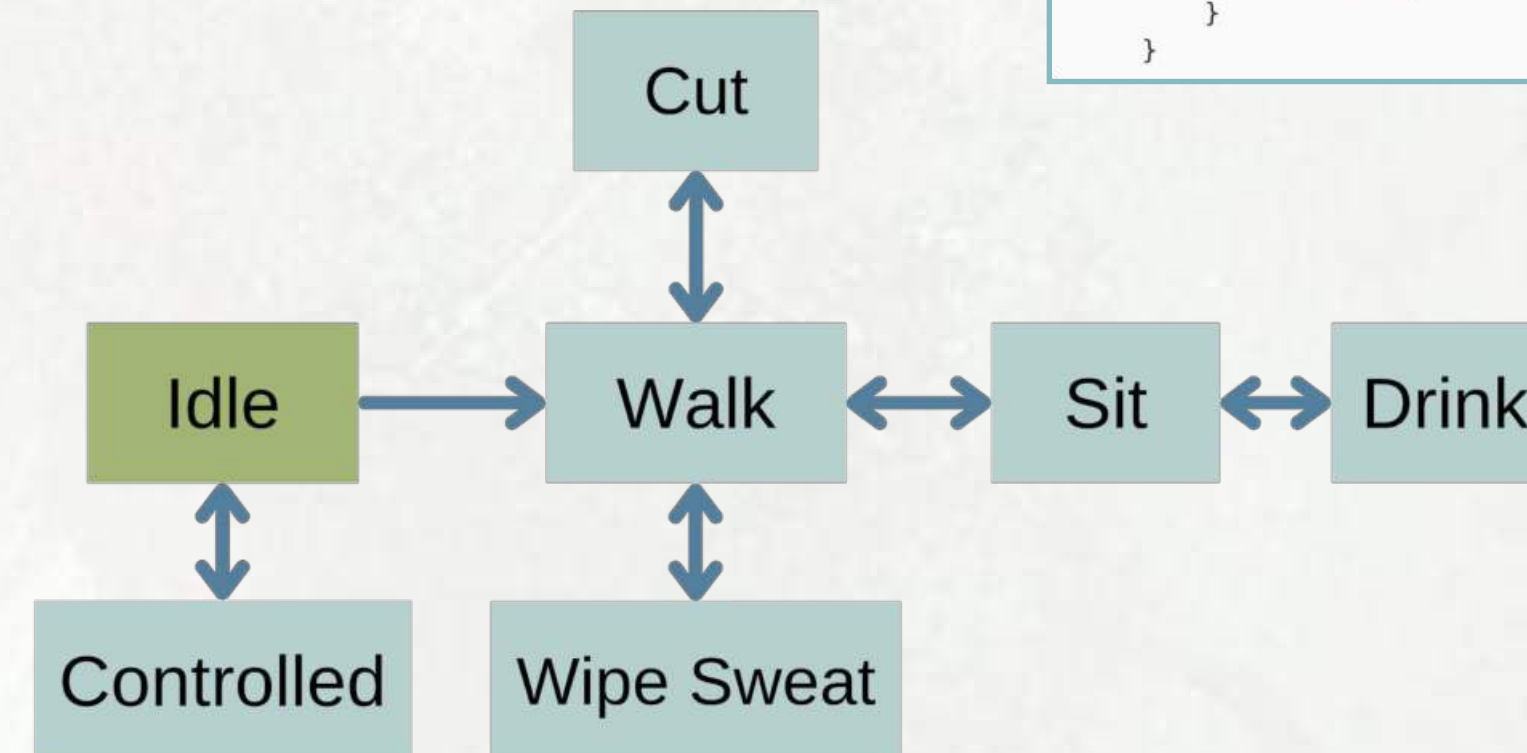
            break;

        case State.Sit:
            StateSit();
            break;
    }
}
```

```
Cutter.cs (Lumberjack)

void StateIdle()
{
    FSMTimer -= Time.deltaTime;
    if (FSMTimer <= 0)
    {
        FSMTimer = 0;
        State randomBehaviorState = (State)Random.Range(1, 4);
        if (randomBehaviorState == State.Cut) ToCut();
        else if (randomBehaviorState == State.WipeSweat) ToWipeSweat();
        else if (randomBehaviorState == State.Walk && wayPoints.Count > 0)
            ToWalk();
    }
}

int cutCount = 0;
void StateCut()
{
    FSMTimer += Time.deltaTime;
    if (FSMTimer >= 3)
    {
        animator.SetTrigger("CutAmbient");
        FSMTimer = 0;
        cutCount++;
        if (cutCount % 4 == 0)
            bamboo.Fall();
        else
            bamboo.Hit();
    }
}
```



```
PlayerInteractableDetector.cs

public void Detect() //detect interactables
{
    Collider[] colliders = Physics.OverlapSphere(DetectionPoint.position,
DetectionRadius);
    Interactable detected = null;
    foreach (Collider collider in colliders)
    {
        Interactable interactable =
collider.GetComponentInParent<Interactable>();
        if (interactable != null)
        {
            detected = interactable;
            break;
        }
    }
    Detected = detected;
    if (detected != null)
        Detected.OnDetect.Invoke();
}

void Interact() //interact with detected
{
    if (Input.GetKeyUp(InteractKey) && Detected != null)
        Detected.OnInteract.Invoke();
}

void UpdateIndicator()
{
    if (Detected && Detected.interactable == true)
    {
        InteractableIndicator.gameObject.SetActive(Detected);
        InteractableIndicator.transform.position =
Detected.transform.position;
    }
    else InteractableIndicator.gameObject.SetActive(false);
}
```

Scripts

Player

Scripts related to controlling player movements, interactions, and state management.

Character Abilities & Interactions

Scripts focusing on special abilities, character behaviors, and interactions.

Interactables

Scripts that handle interactions with the game environment, including breakable items and dynamic elements.

Utilities

Helper scripts and tools for managing various game utilities, configurations, and editor-related functions.

Development & Iteration

Conceptualization

The player explores personal disconnection from cultural origins through a surreal journey in a ghost realm, symbolizing the search for identity and belonging. This experience reflects my personal complex feelings shared by many younger generations from immigrant families.

Cultural Disconnection

Draws on the theories of Homi K. Bhabha and his concept of Hybridity, hich examines how cultural identities are formed in the liminal space between different traditions and influences.

Memory Reconstruction

Informed by Pierre Nora’s notion of Lieux de Mémoire (Sites of Memory), suggesting that fragmented memories are reconstructed to create a sense of shared history and identity.

Lost Familiarity

Inspired by Edward Said’s ideas of Exile and Estrangement, which discuss the emotional complexities of being both part of and separate from one’s cultural heritage.



Iteration & Drafts

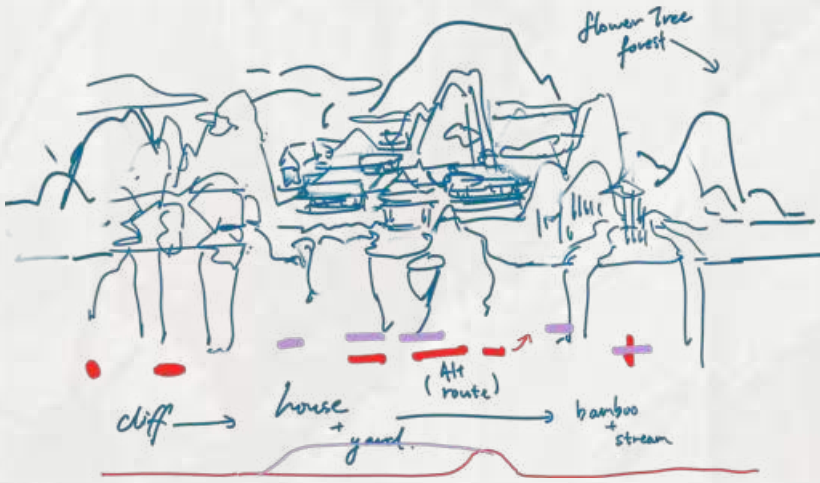
Various iterations and level drafts were developed to refine gameplay flow, visual composition, and narrative pacing, ensuring the experience aligns with the conceptual themes.



Development Workflow

Pre-Production (Oct 8th - Oct 22nd)	Design documentation, initial game mechanics, art style setup, and reference gathering.
Production (Oct 22nd - Dec 3rd)	Level design, asset list creation, prop modeling, and character rigging/animation.
Implementation (Nov 19th - Dec 17th)	Level building, VFX, sound assets, and UI integration.
Polishing & Testing (Dec 24th - Jan 14th)	UI and sound installation, post-processing, and playtesting.

To do \ Weeks	Oct 8th	Oct 15th	Oct 22nd	Oct 29th	Nov 5th	Nov 12th	Nov 19th	Nov 26th	Dec 3th	Dec 10th	Dec 17th	Dec 24th	Jan 7th	Jan 14th
Design Doc														
major Game Mechanic (small functional demo)														
Find References														
Set Artstyle														
Level Design														
Concept sketches														
Asset list														
gather prop models														
Player Model + Rigging														
Spirit Model+ Animation														
State Machines + Animation														
Level demo														
Build levels with assets														
Particles + vfx														
gather sound asset														
Design UI														
install UI and Sounds														
Post Process Camera														



Screenshots

