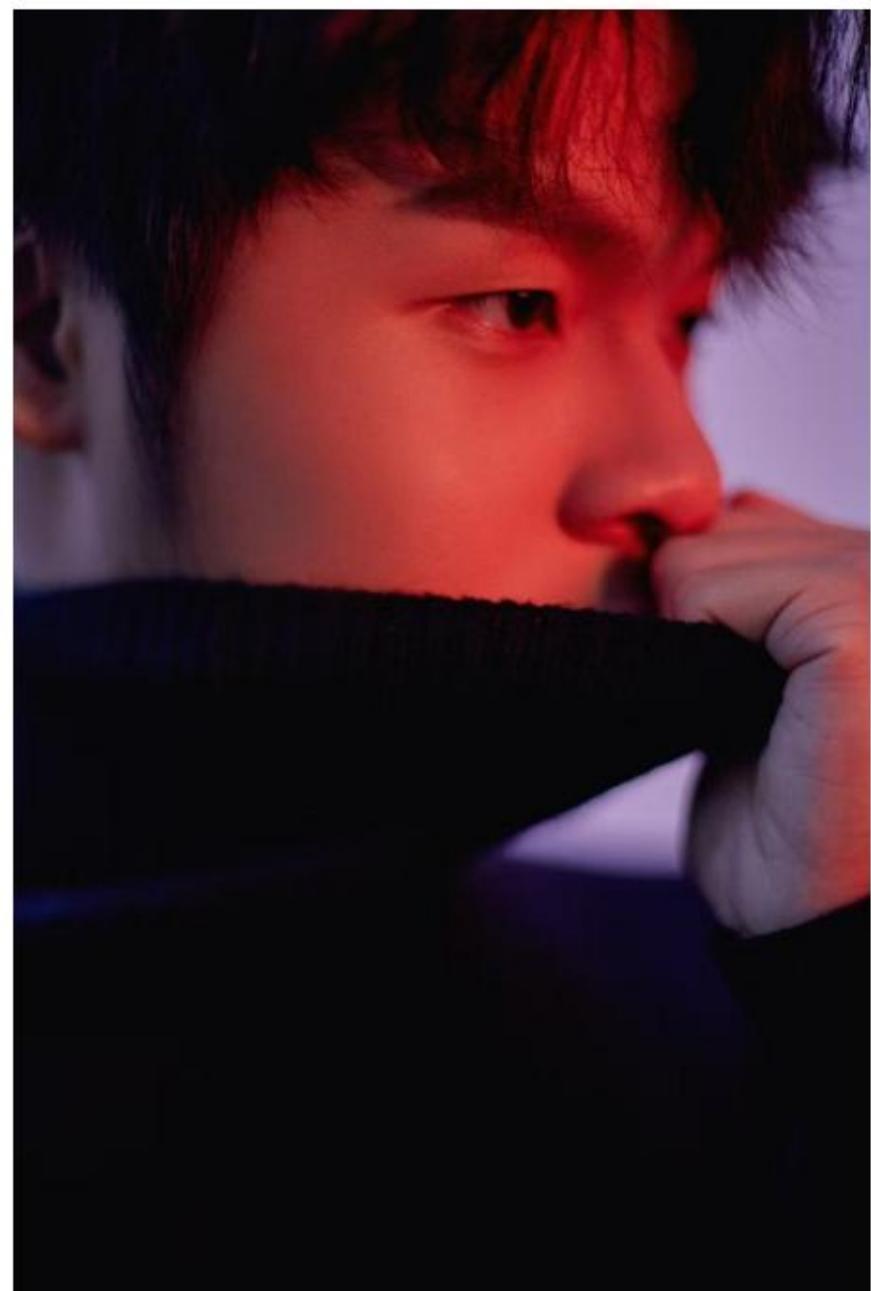
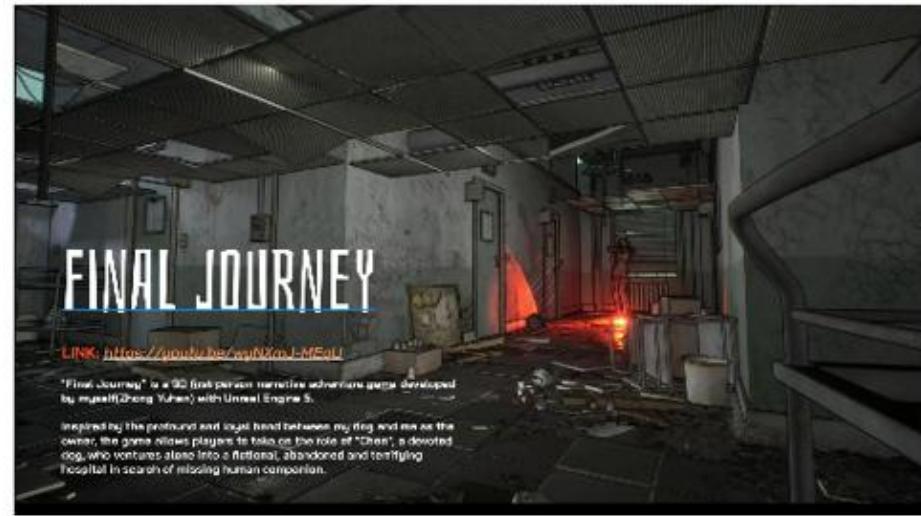


Portfolio

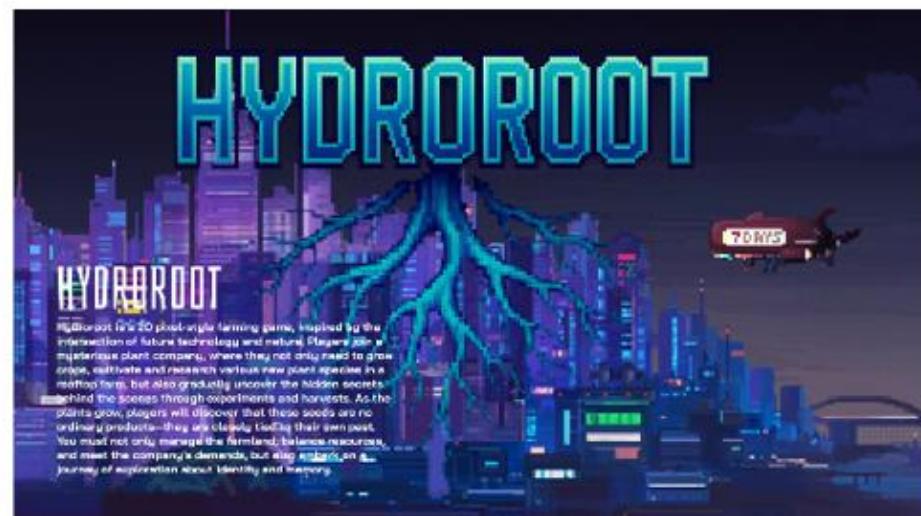


Yuhan Zhong

Proficient in Unreal Engine and Unity Engine, with a focus on narrative-driven game design and treating games as cultural vehicles. Currently working on two personal projects and two collaborative projects.



Final Journey
PC. Horror Puzzle Game



Hydroroot
PC. Farm Simulation Game



Synchronizer
PC. Rhythm Shooter Game



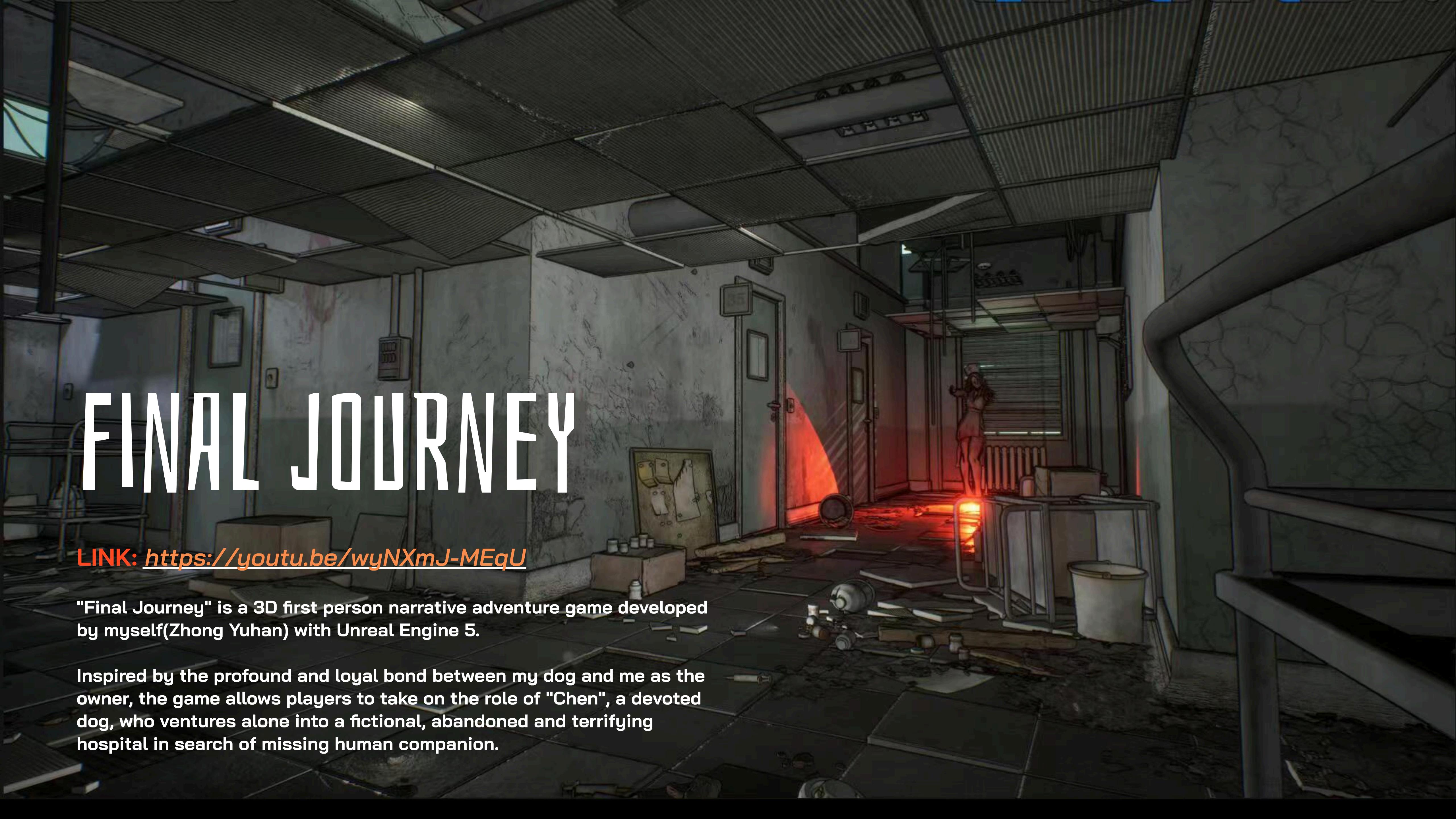
Echoes of Canvas
VR. Narrative Art Experience Game

FINAL JOURNEY

LINK: <https://youtu.be/wyNXmJ-MEqU>

"Final Journey" is a 3D first person narrative adventure game developed by myself(Zhong Yuhan) with Unreal Engine 5.

Inspired by the profound and loyal bond between my dog and me as the owner, the game allows players to take on the role of "Chen", a devoted dog, who ventures alone into a fictional, abandoned and terrifying hospital in search of missing human companion.



Inspiration



The inspiration for this game mainly stems from my deep affection for dogs—I currently have two dogs in the United States, and they are incredibly important companions in my life. Even when I'm at home, they become extremely anxious and restless if they can't see me, as they desperately want to find me. After traveling with me and returning home, they'll rush to my side without hesitation. This deeply touches me and makes me realize how much people value dogs' loyalty. I want to create a game that tells a story from a dog's perspective—a game that celebrates their loyalty, attentiveness, emotions, and unconditional companionship.

Research

In the research on Hachiko: A Dog's Story:

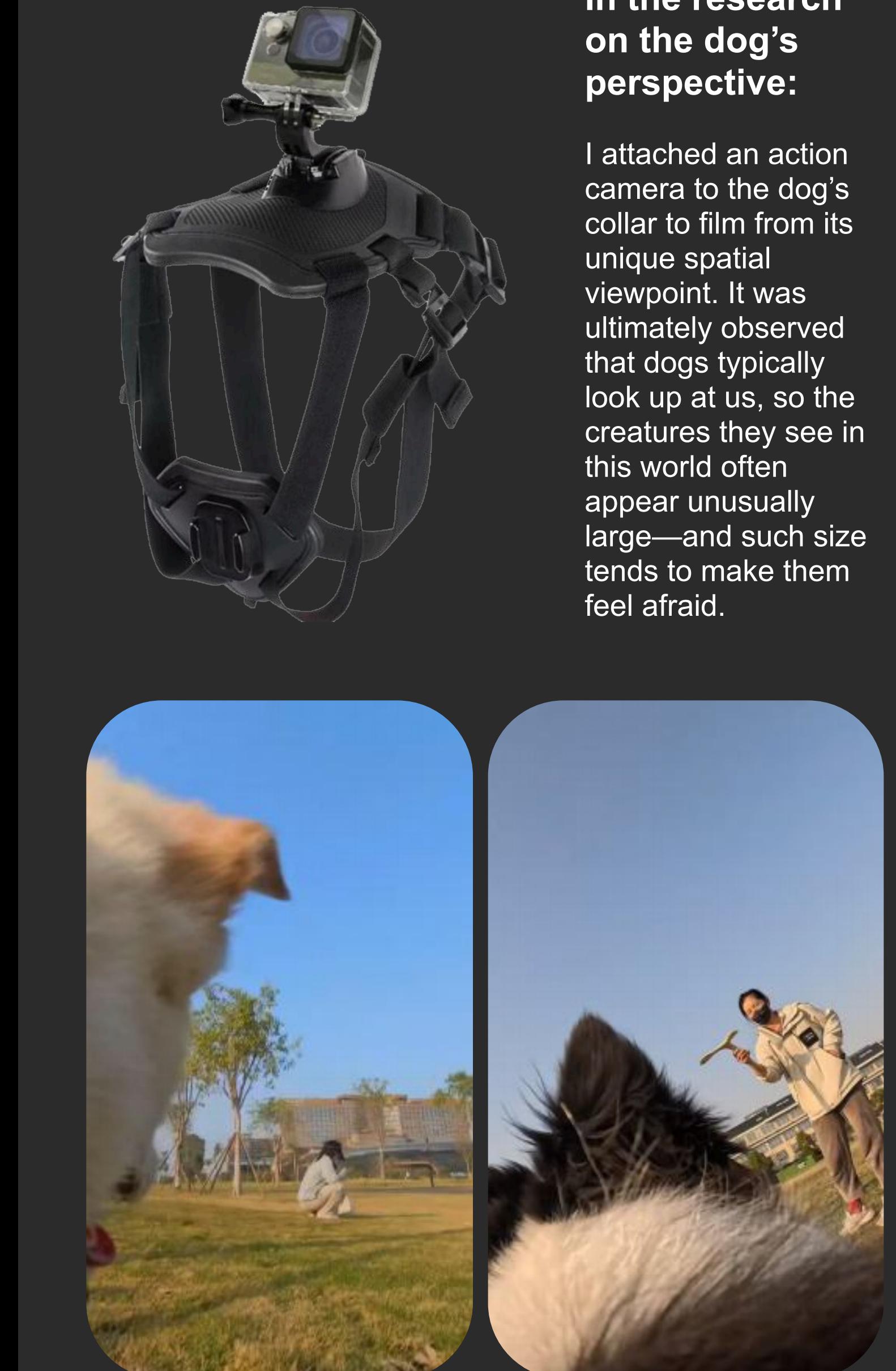
Olfactory Guidance Mechanism: Inspired by the scene in the film where the dog relies on its sense of smell to track, I propose using "scents" as clues in the game. This emphasizes presenting the game from a dog's perspective, aligning perfectly with the canine viewpoint.

Non-Verbal Narrative Structure: Drawing inspiration from the emotional storytelling and behavioral expressions in the film, the game will convey emotional conflicts through a combination of environmental sound effects, lighting, scenes, and interactions to build an emotional landscape.



For the ending design, I will integrate the tragic conclusion of Hachiko:

A Dog's Story with the O. Henry-style narrative structure. This aims to deliver a thought-provoking ending—though emotionally sorrowful, it will also bring players unexpected feelings.



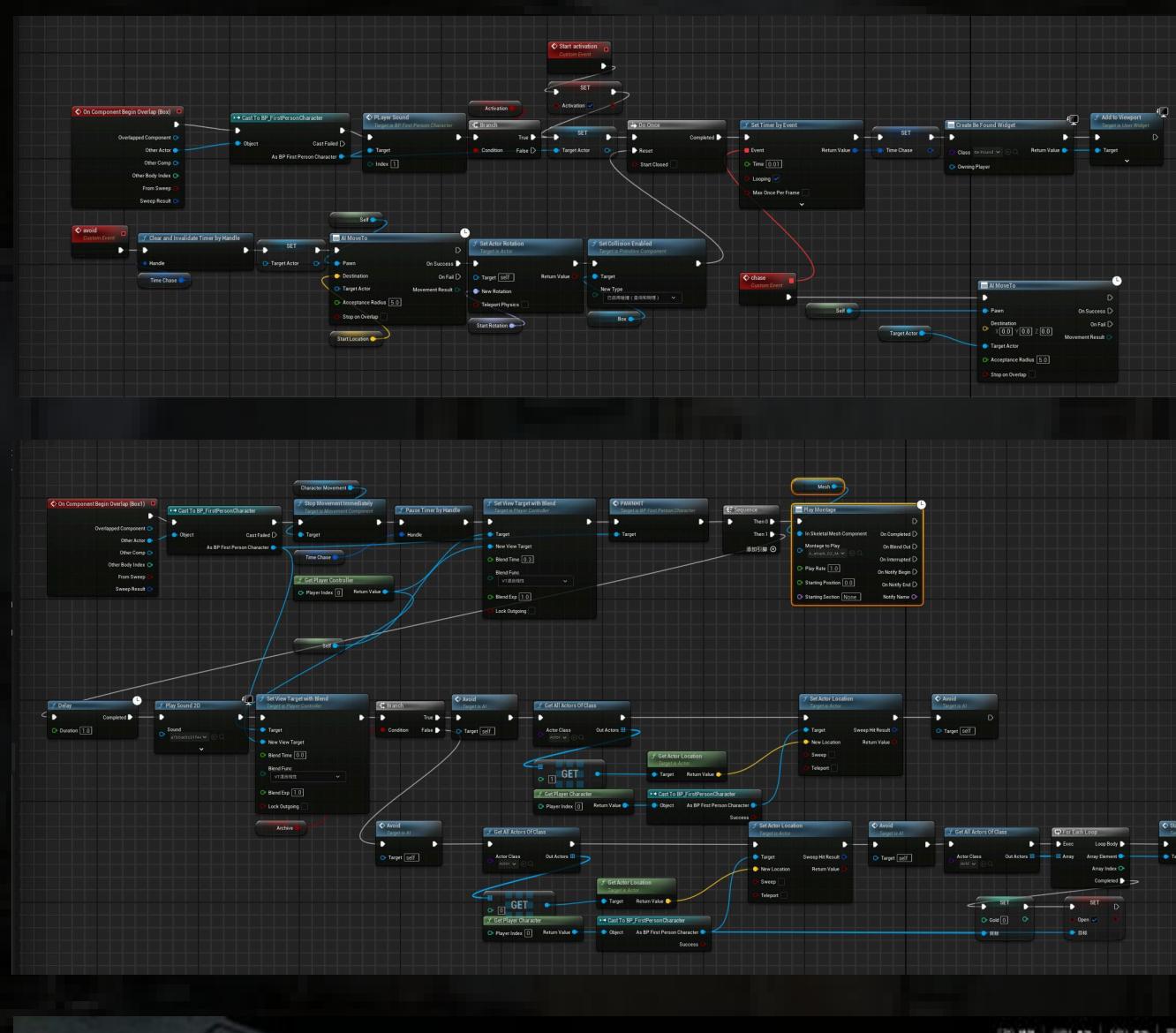
In the research on the dog's perspective:

I attached an action camera to the dog's collar to film from its unique spatial viewpoint. It was ultimately observed that dogs typically look up at us, so the creatures they see in this world often appear unusually large—and such size tends to make them feel afraid.

Technical Implementation

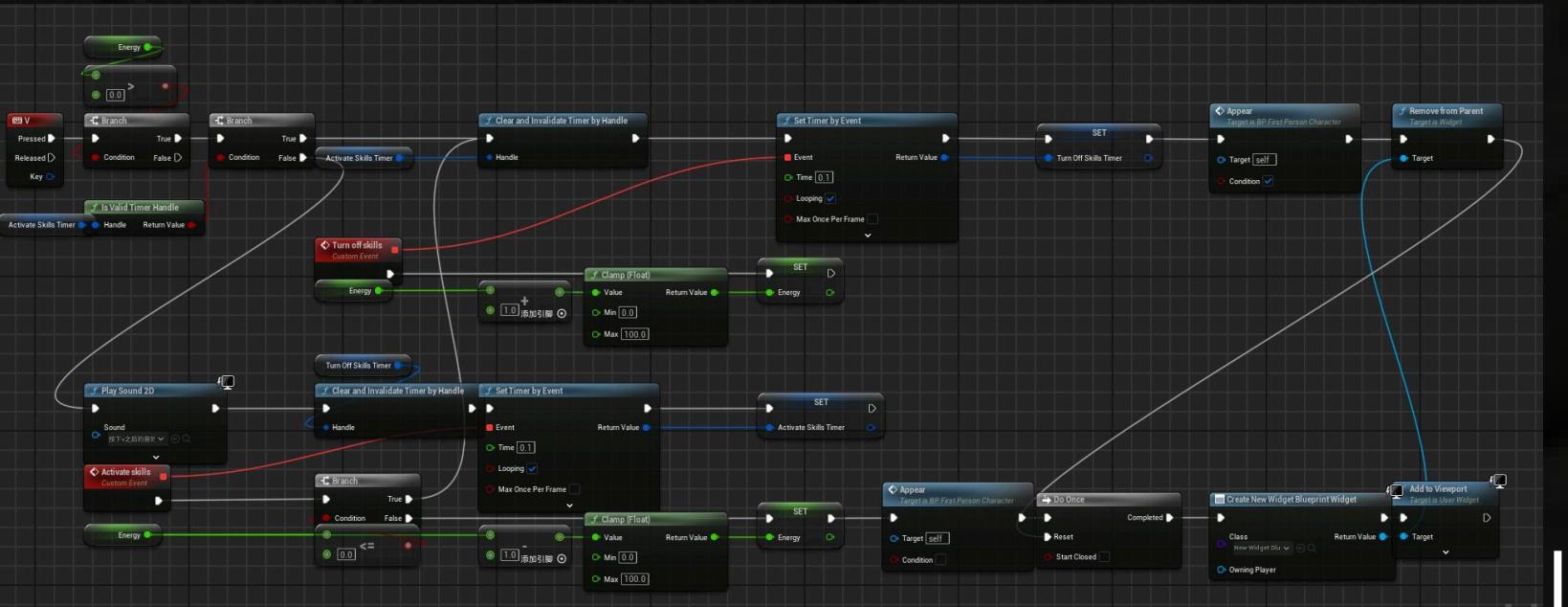
AI Targeting

When AI identifies a player target via its perception system (e.g., visual perception), it will first trigger a series of operations through the On Target Perception Updated node.



Scent Search

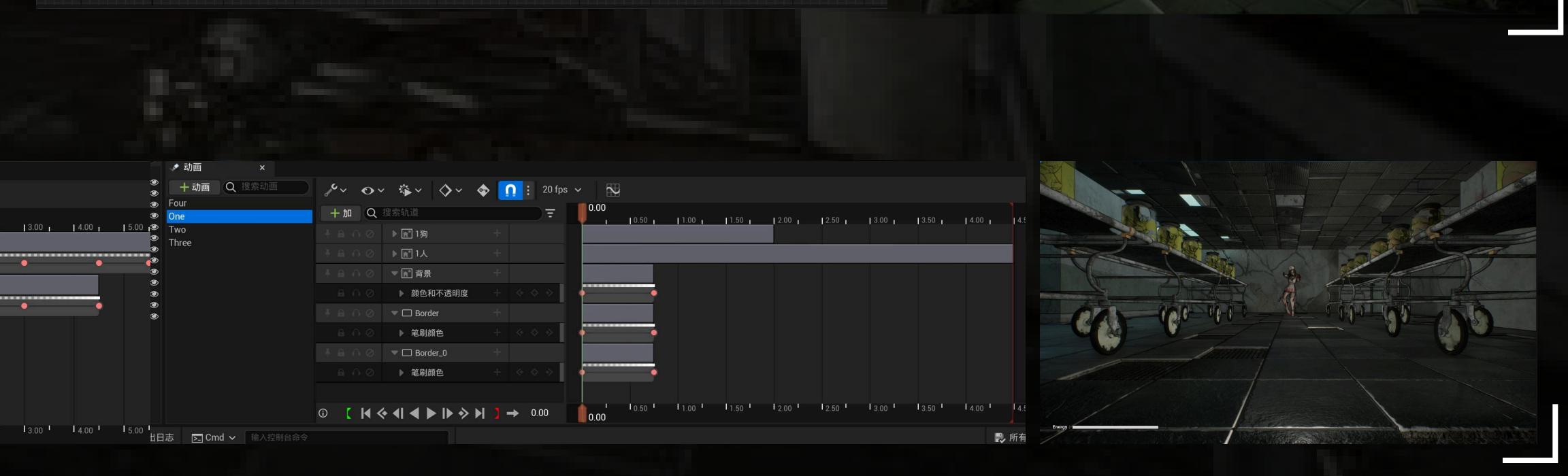
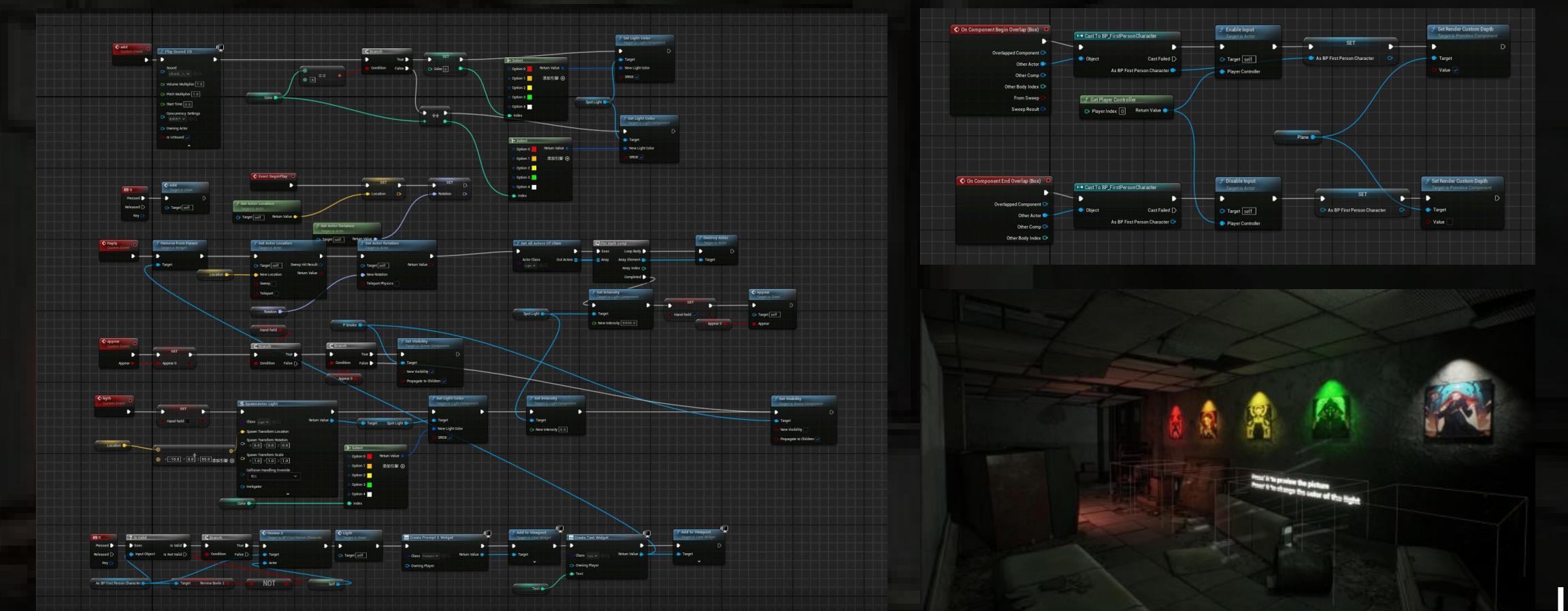
The AI's eligibility to attack at this moment is ensured by a boolean variable and whether the current animation state machine is in the Idle state. If the initial conditions are met, the next step is to determine if the AI has sufficient stamina, which serves as a prerequisite for attacking.



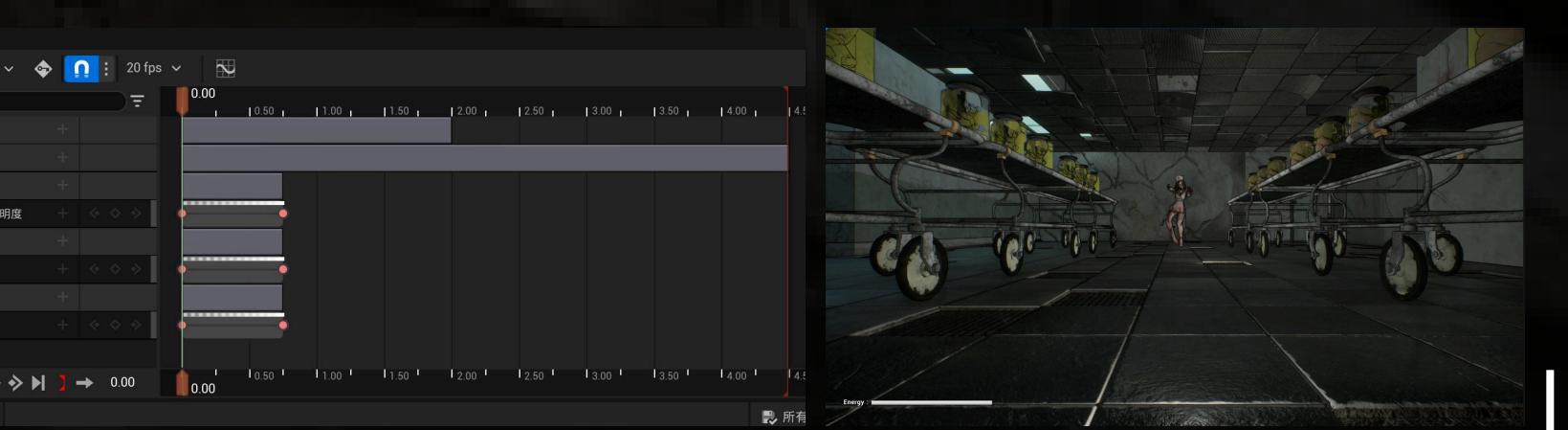
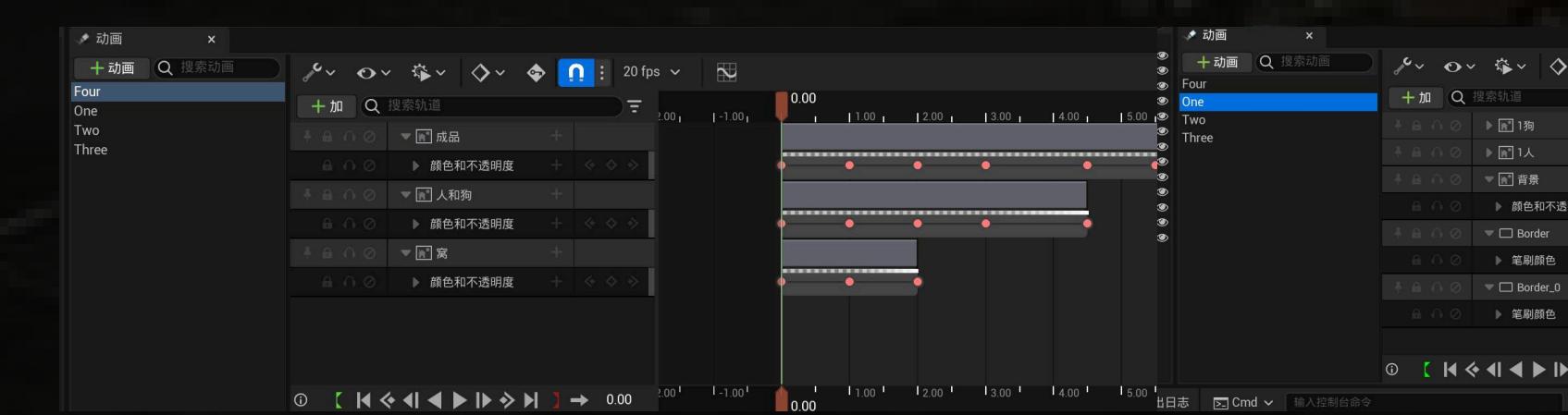
Light Signal Puzzle

When the player enters a specific Trigger Box, the system first verifies if the object is the player character. Upon confirmation, it calls the Enable Input node to activate input and uses Set Render Custom Depth to add a stroke highlight to the target object—guiding the player's attention and marking the interactive object.

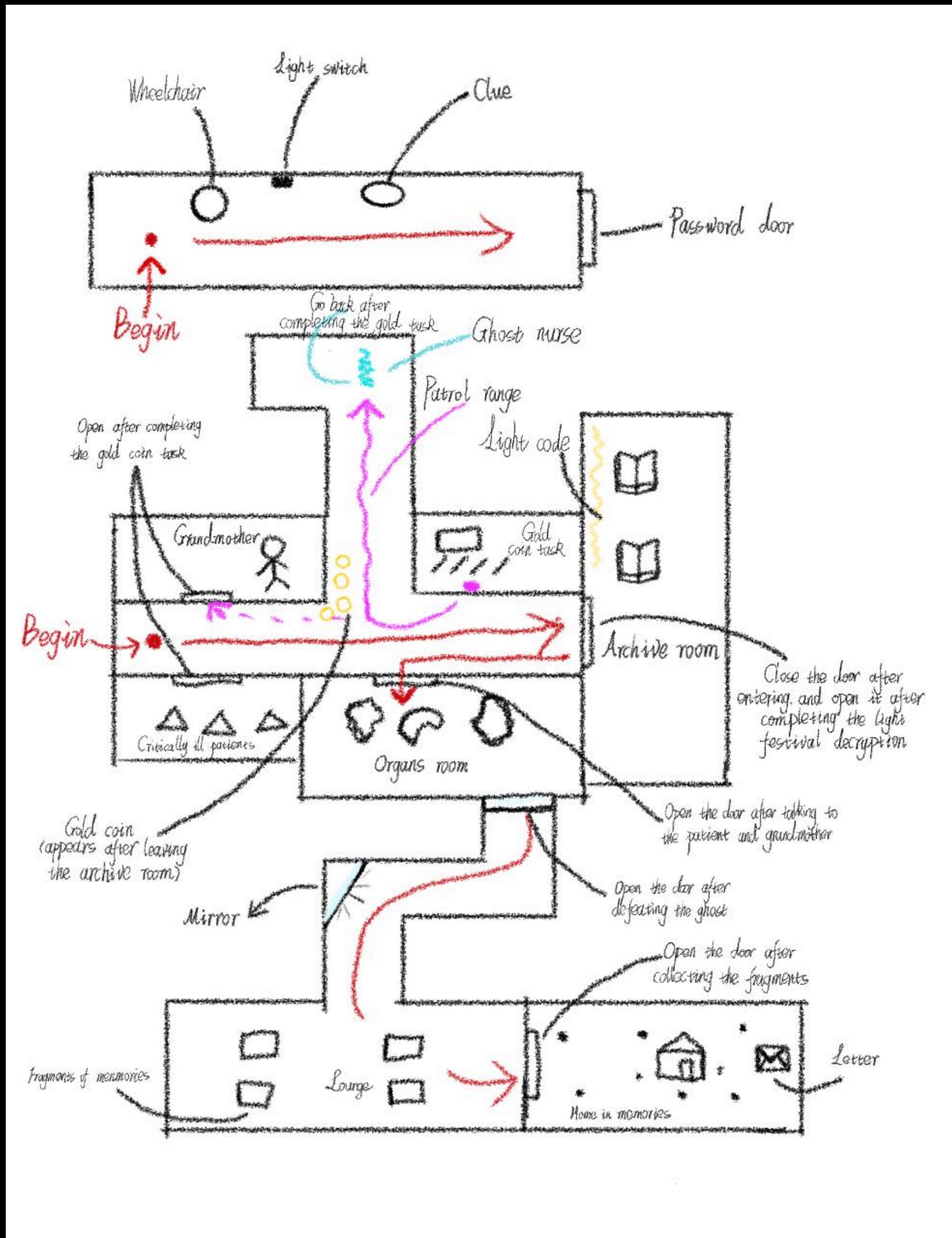
When the player leaves the area, the system disables control via the Disable Input node and removes the highlight, ensuring that interactive actions are only available within a reasonable range.



Interactive Animation



Map Design



Character Design



─ A dying patient in bed, but from the dog's perspective, he sees a rather terrifying scene.



─ Xiao Yu in Xiao Chen's memories.



─ Xiao Yu after falling ill.



─ A nurse in the corridor—on the left is the form seen from the dog's perspective.



─ The player-controlled character—Xiao Chen, who is Xiao Yu's best friend. Initially, the player has no idea that they are controlling a dog, and the puzzle is gradually uncovered later on.



Game Flow



Entering the Hospital (Puzzle Solving Phase)

- The player, as Dust the dog, **sneaks** into the mysterious building through the basement.
- Solve environmental puzzles, unlock doors, and **uncover hidden paths**.
- **Slowly build tension** through eerie visuals and passive ghostly figures.



Uncovering the Truth (Discovery Phase)

- Explore rooms filled with files, diaries, and photographs.
- Realize the “monsters” are former patients and staff.
- Discover the doctor's twisted obsession and Xiaoyu's fate through emotional storytelling.



Escaping Pursuit (Chase & Stealth Phase)

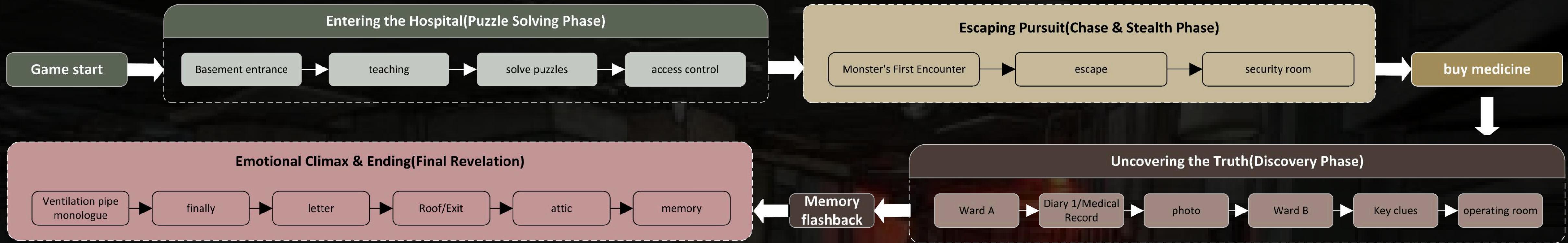
- Hostile creatures begin appearing—monsters with weapons, disturbing figures.
- The player must **hide**, run, and evade using stealth mechanics.
- Critical items and clues are collected during close escapes.



Emotional Climax & Ending (Final Revelation)

- Inside the ventilation shaft, Dust **realizes he is a dog** and Xiaoyu was his owner.
- Find Xiaoyu's final letter, memories, and her empty room.
- Attend her funeral in the final scene—concluding the journey with heartbreak, love, and closure.

Game Flow



Narrative Featured Screenshots

- Explore rooms filled with files, diaries, and photographs.
- Realize the “monsters” are former patients and staff.
- Discover the doctor's twisted obsession and Xiaoyu's fate through emotional storytelling.



HYDRO ROOT

A dark, pixelated cityscape background featuring a futuristic, glowing structure in the center. A small, floating 7DAYS logo is visible in the upper right corner.

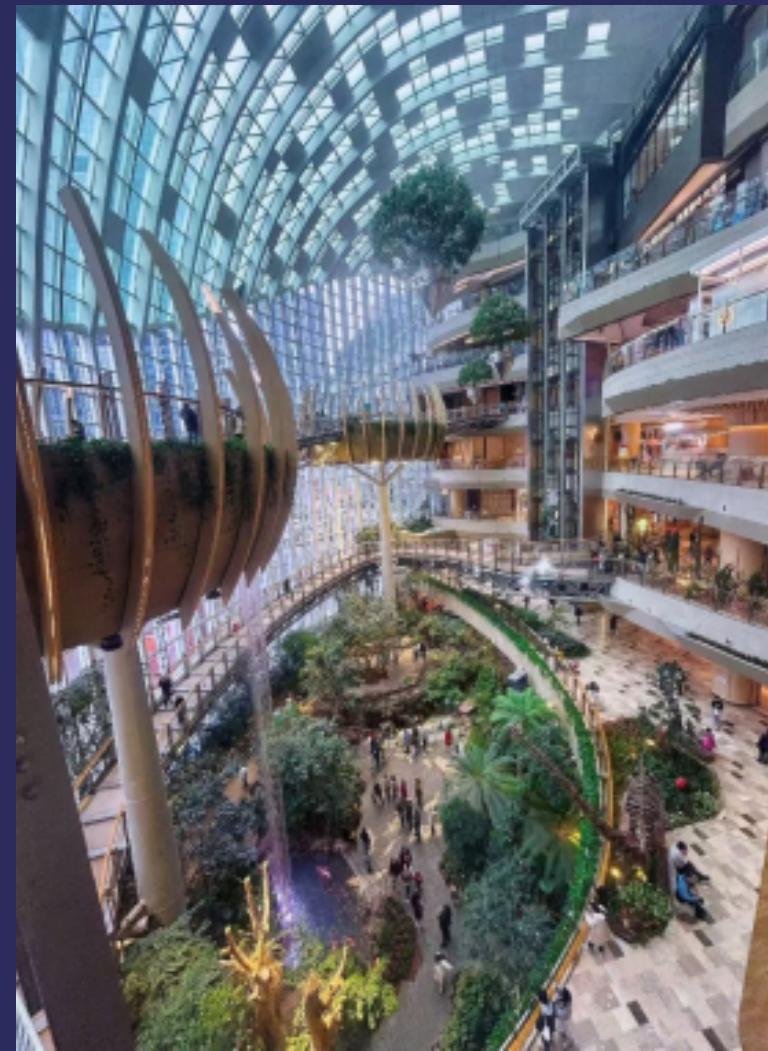
HYDRO ROOT

Gameplay Video: <https://youtu.be/fipWxpgMxaY>

Promotional Video: <https://youtu.be/vANtIVkrXKo>

Hydroroot is a 2D pixel-style farming game, inspired by the intersection of future technology and nature. Players join a mysterious plant company, where they not only need to grow crops, cultivate and research various new plant species in a rooftop farm, but also gradually uncover the hidden secrets behind the scenes through experiments and harvests. As the plants grow, players will discover that these seeds are no ordinary products—they are closely tied to their own past. You must not only manage the farmland, balance resources, and meet the company's demands, but also embark on a journey of exploration about identity and memory.

Inspiration



The inspiration for this game came from the thoughts I had when I saw the "cyber forest" built in Chongqing during a trip there. Two elements that seem contradictory—plants and future technology—can actually coexist.

As for the game's story, it draws inspiration from the film *The Ultimate Gift*. In the movie, to inherit the legacy left by his grandfather, the protagonist must complete the tasks assigned by his grandfather step by step. Through completing these tasks, he gradually gains profound insights and acquires gifts beyond money, such as friendship and love.

Similarly, in my game, players need to finish the tasks given by the plant company step by step to slowly unravel the mystery of their own origins, and finally understand the truth that nature and technology can coexist.

Research

Stardew Valley

Non-Mandatory Quest System:

Drawing inspiration from the design philosophy of *Stardew Valley*, the game emphasizes allowing players to gradually unravel the mystery of their own origins and ultimately comprehend their unique "gift" through free exploration and quest progression. Therefore, a non-mandatory quest system is adopted, enabling players to enjoy a more natural and immersive gaming experience without excessive constraints.

Dave the Diver

Dual-Cycle System:

Taking cues from the design concept of *Dave the Diver*, the game not only lets players focus on growing and researching new plant species but also allows them to send specific cultivated crops into the food processing stage, converting them into a variety of dishes and products. Through the cyclic mechanism of "planting—processing—reinvestment," the game further enriches the depth of gameplay and the diversity of experience.



Technical Implementation

Farming System

Each plant requires three conditions to be met: water, sunlight, and fertilizer. Sunlight and water need to be generated by consuming electricity, while fertilizer must be purchased with in-game currency.

```
public class FarmlandController : MonoBehaviour
{
    public int Index;
    private bool hightLightMode = false;
    3 个引用
    public bool isHighLight
    {
        get
        {
            return hightLightMode;
        }
        set
        {
            if(value != hightLightMode)
            {
                SetHighLight(value);
                hightLightMode = value;
            }
        }
    }
    public GameObject highLightObj;
    public SpriteRenderer PlantsSR;
    public SpriteRenderer m_renderer;
    public Transform m_ItemRoot;
    public GameObject m_WaterIcon;
    public GameObject m_SunIcon;
    public GameObject m_FertilizerIcon;
    public GameObject m_WarningIcon;
    public SpriteRenderer m_BoxStateIcon;
    public GameObject Arrow;
    public GameObject Harvest;

    public ConsumableConfig m_Seed;

    public bool havePlantUpgrade1=false;
    public bool havePlantUpgrade2=false;
    public bool haveProcessMachine=false;
    public bool isWarning;
```



```

public int currentDay = 0;
@UnityEditorOnly 10 个引用
private void Start()
{
    Arrow = transform.Find("Arrow").gameObject;
    Arrow.SetActive(false);
    Harvest = transform.Find("Harvest").gameObject;
    Harvest.SetActive(false);
    highlightObj = transform.Find("Highlight").gameObject;
    m_WaterIcon = transform.Find("WaterIcon").gameObject;
    m_WarningIcon = transform.Find("Warning").gameObject;
    m_BoxStateIcon = transform.Find("BoxState").gameObject.GetComponent();
    PlantSR = transform.Find("Crop.Seed").Find("GrowState").GetComponent();
    m_WaterIcon.SetActive(false);
    m_SunIcon = transform.Find("SunIcon").gameObject;
    m_SunIcon.SetActive(false);
    m_FertilizerIcon = transform.Find("FertilizerIcon").gameObject;
    m_FertilizerIcon.SetActive(false);
    m_IconRoot = transform.Find("Crop.Seed");
    m_Renderer = transform.GetComponent();
    m_renderer.UseColor = m_renderer.color;
    WaterCount = 0;
}

1 个引用
public void SetHighLight(bool ishighlight)
{
    if (ishighlight)
    {
        highlightObj.SetActive(true);
    }
    else
    {
        highlightObj.SetActive(false);
    }
}

1 个引用
public void UseSeed(ConsumableConfig config)
{
    if (config.consumableType == ConsumableConfig.ConsumableType.Seed)
    {
        m_Seed = Instantiate(config);
        if (haveProcessMachine)
        {
            m_Seed.UseSun = Mathf.CeilToInt(m_Seed.UseSun * 1.5f);
            m_Seed.UseFertilizer = Mathf.CeilToInt(m_Seed.UseFertilizer * 1.5f);
        }
        currentDay = 0;
        SetBoxState();
        PlantsSR.sprite = config.GrowSprites[currentDay];
        //m_0b = GameObject.Instantiate(m_Seed.SeedObj), m_IconRoot;
    }
}

5 个引用
public void SetBoxState()
{
    if (m_Seed != null)
    {
        m_BoxStateIcon.gameObject.SetActive(true);
        if (currentDay == m_Seed.GrowTime)
        {
            if (FertilizerCount >= m_Seed.UseFertilizer && WaterCount >= m_Seed.UseWater && SunCount >= m_Seed.UseSun)
            {
                m_BoxStateIcon.sprite = FarmlandsManager.Instance.boxState[m_Seed.GrowTime - 1].EnoughSprite[currentDay];
                m_WarningIcon.SetActive(false);
            }
            else
            {
                m_BoxStateIcon.sprite = FarmlandsManager.Instance.boxState[m_Seed.GrowTime - 1].NoEnoughSprite[currentDay];
                m_WarningIcon.SetActive(true);
            }
        }
        else
        {
            m_BoxStateIcon.gameObject.SetActive(false);
            m_WarningIcon.SetActive(false);
        }
    }
}

1 个引用
public void RemoveCrop()
{
    m_Seed = null;
    isUseBox = false;
    isRipe = false;
    PlantSR.sprite = null;
    currentDay = 0;
    Arrow.SetActive(false);
    Harvest.SetActive(false);
}

1 个引用
public void UseBox()
{
    isUseBox = true;
    //m_renderer.color = AfterUseBoxColor;
}

```

a plant's daily resource requirements are met, it will grow; otherwise, it will not.

ding System



Technical Implementation

Game Systems and Ending Determination

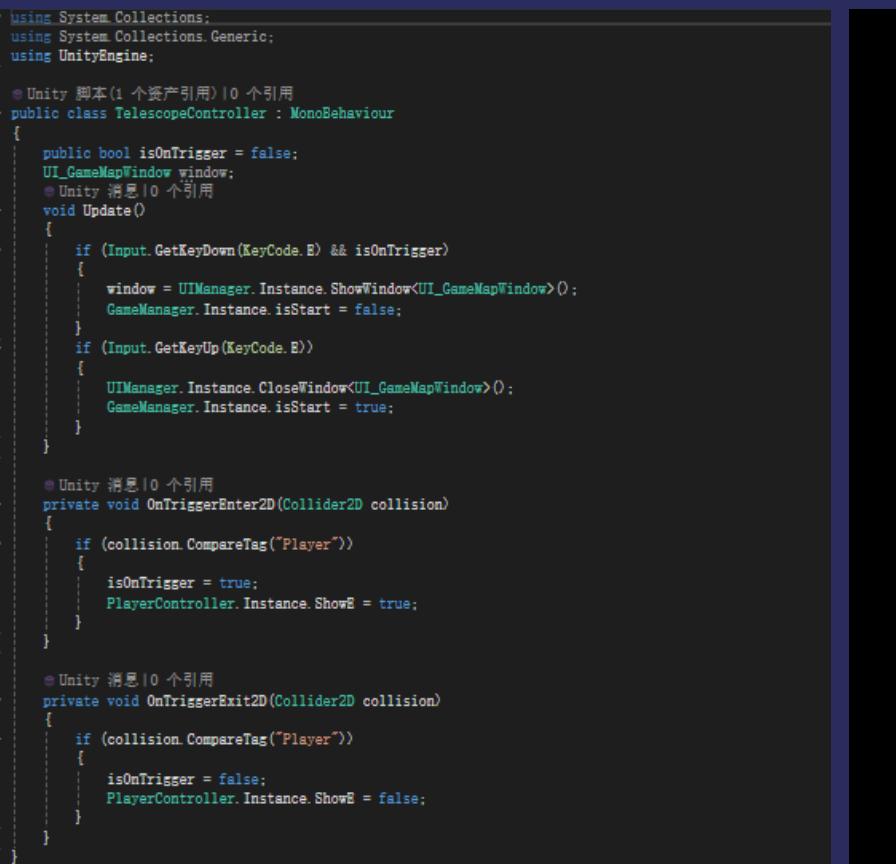
This section includes day switching, the tutorial, and ending determination.



Equipment Upgrade System



Telescope System



Game flow



Processing



Repayment



Ending Determination



Scene Animations

Cutscene 1

As the city rapidly expanded, my grandfather founded HydroRoot—the largest plant-supply company at the time. Thanks to him, I spent my childhood making memories as part of one of the city's wealthiest families.

Cutscene 2

But after my grandfather passed away, my father's reckless investments quickly drained our fortune, plunging us into debt. Eventually, even HydroRoot was bought out. Then, my mother left without a word.

Cutscene 3

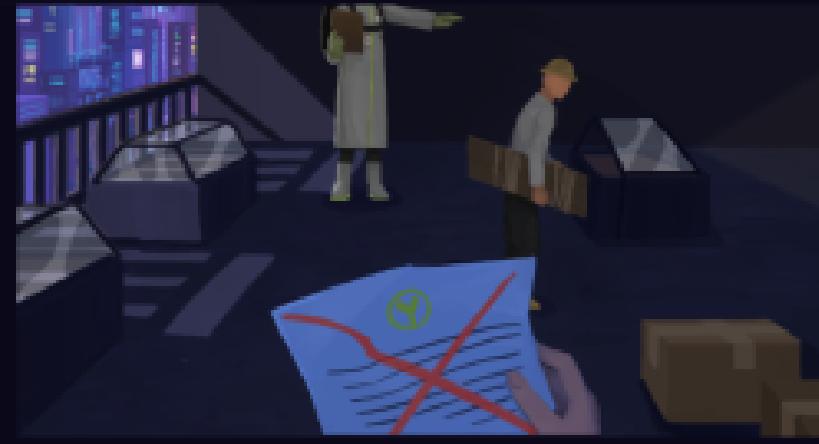
After my father killed himself from alcohol, I moved alone to a small house on the outskirts of the city, burdened with repaying the debts he left behind. But one day, a man from HydroRoot found me and made me an offer I couldn't refuse.

Cutscene 4

"This is a challenge, kid. Prove that you're not as useless as your father. The equipment will be sent to your apartment once the contract has been signed. By the end of Day 42, we'll review your results. Meet the goal, and your debts will be cleared — This is our last favor out of respect for your grandfather."



Ending Determination



1. BAD ONE - Fail to meet goals in any round:

The company man comes to the apartment.

"You failed, kid - just as incapable as your father. We have nothing left to say on this situation. Still, we know that the debts are your father's fault, not yours. From now on, those collectors will not trouble you. Go and live on your own, but never come back to HydroRoot."



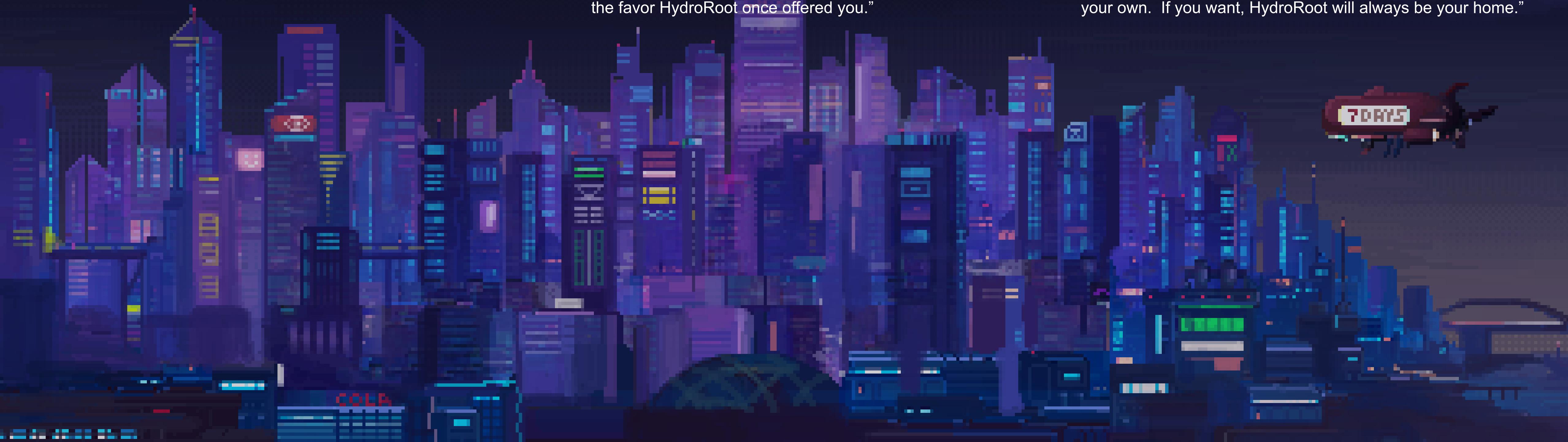
2. NORMAL ONE - Meet goals in the range:

"You kind of made it, kid. Congratulations, even though this level is not even as good as your grandfather's worst apprentice. As the contract says, we've cleared all of your father's debts; those debt collectors will never trouble you. Now go back to your normal life and live on your own. Don't forget the favor HydroRoot once offered you."



3. EXCELLENT ONE - Excellently meet goals and earn much more:

"You made it, kid, and you made it perfectly. Congratulations, your grandfather would be gratified if he were alive. As the contract says, we've cleared all of your father's debts, and those debt collectors will never trouble you. Now you can go back to your normal life and live on your own. If you want, HydroRoot will always be your home."



Team Members and Roles

(Me)Zhong Yuhu(Lawrence): Game Programming & Develop
Xie Yankai(Yankai Xie): Gameplay & Narrative & Level Design
Jiang Haoyi(Daffodil): Game Art & Design



SYNCHRONIZER

3D Action & Puzzle Game made with UnrealEngine

Synchronizer is a game that combines music rhythm with shooting elements. Its inspiration came from the ideas that popped into my mind when I listened to music in airport waiting halls during each trip back to my home country.

Players take on the role of a shooter, attacking by shooting enemies, activating "Rhythm Time," or hitting musical notes floating in the air.

Eventually, they embark on a journey to return to the place where their dream began.

Start

Exit

LINK: https://youtu.be/daQ_xsM1fRw

Inspiration

The inspiration for this game comes from my life as an international student. Every time I waited for my flight back home at Los Angeles International Airport (LAX), I would sit in the waiting hall listening to music and daydreaming about various aspects of life after returning. Therefore, I want to design the levels in the game as obstacles encountered during the journey, such as customs queues, flight delays, lost luggage, and other issues. Players need to stay calm like a marksman and deal with these challenges accurately. Meanwhile, music, as the most important comfort in my journey, provides strength and solace.

Research

The Combination of Shooting and Music

Currently, there is no ideal game that combines shooting and music elements on the market. Therefore, in terms of gameplay, I plan to draw inspiration from the mechanism of Iso—a character in Valorant—who targets orbs, and on this basis, incorporate music trajectory elements. This will allow players to click at the right rhythm nodes, thereby creating an integrated experience of shooting and music.



Music Sheets

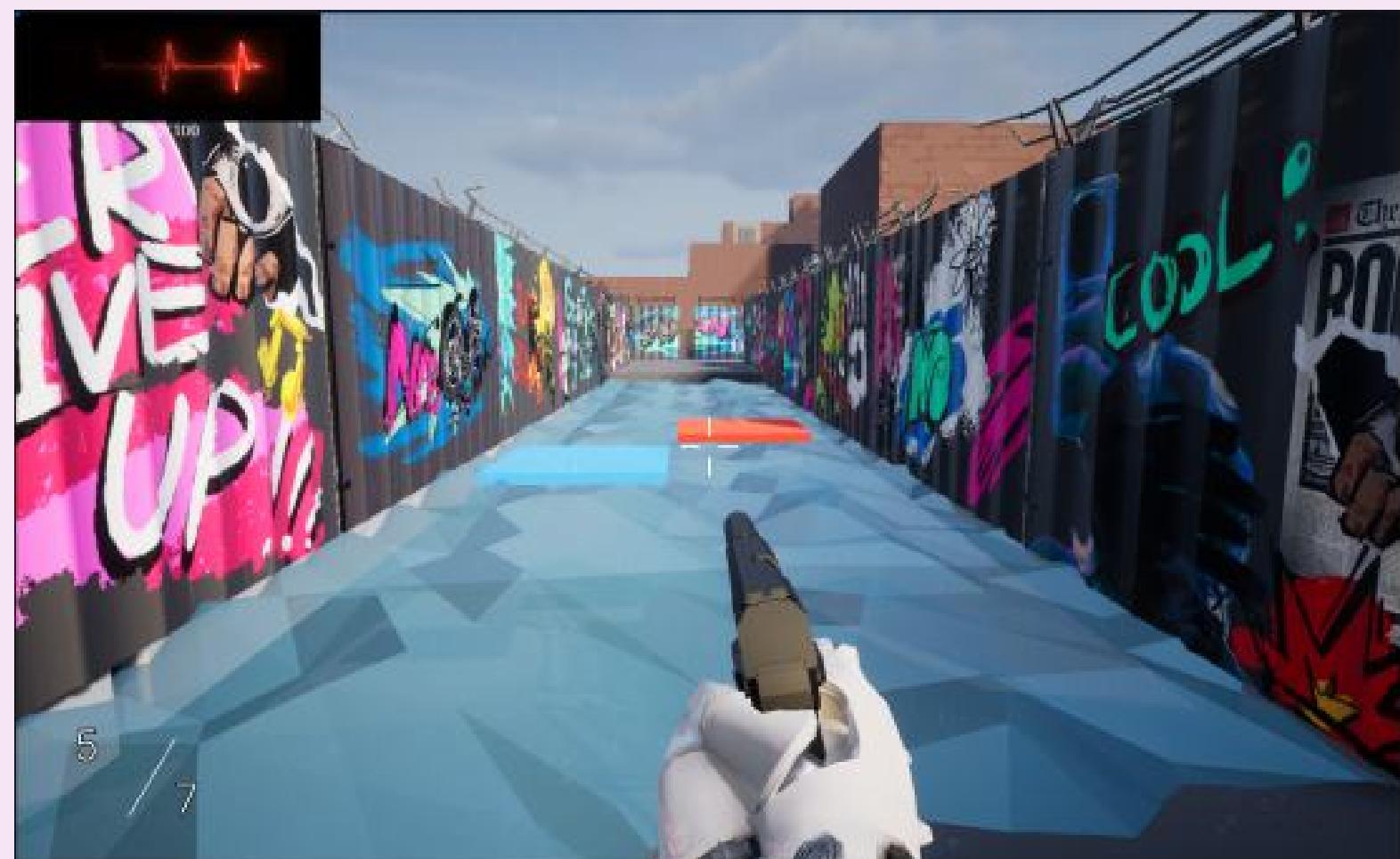
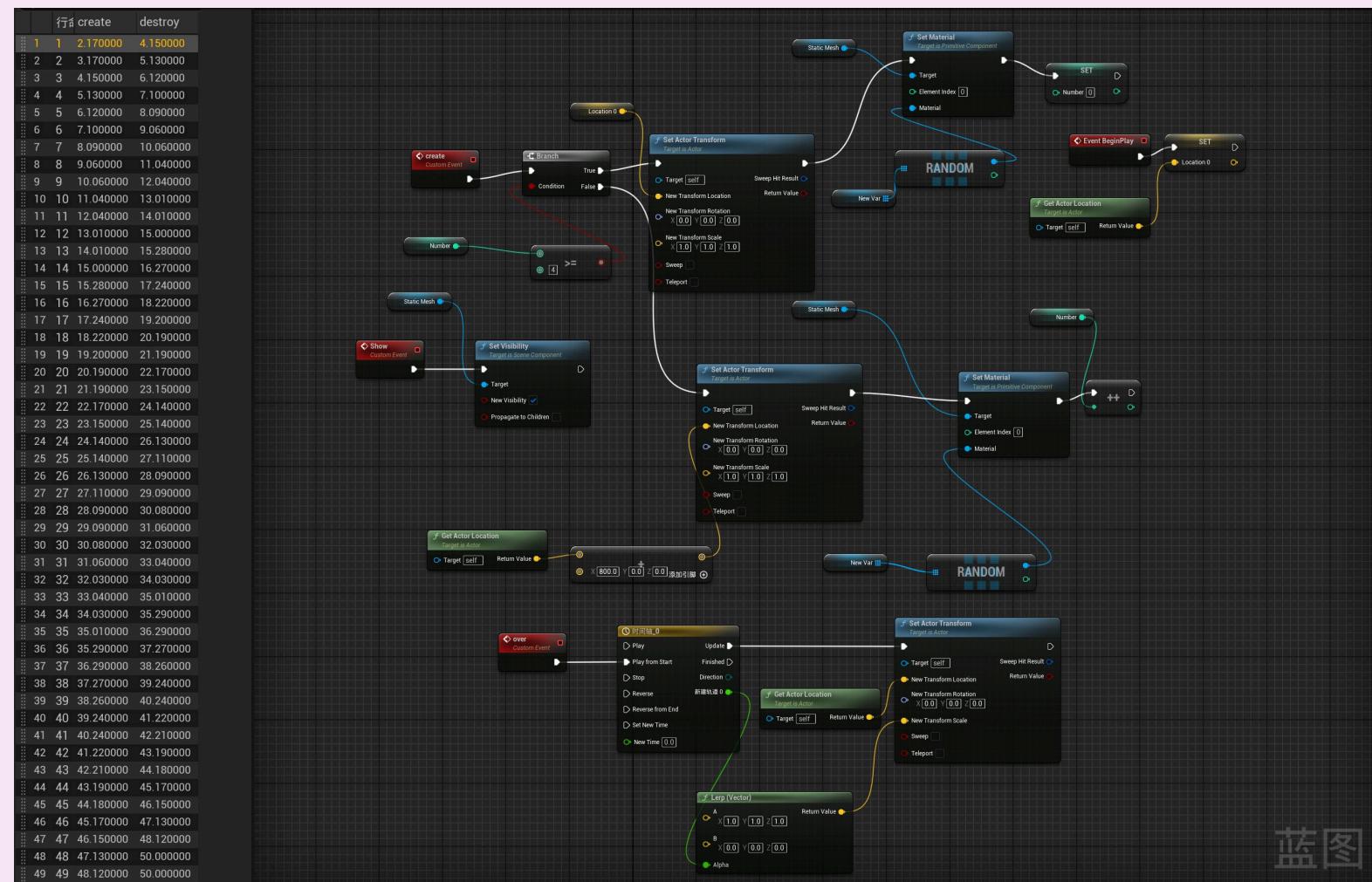
Before officially developing this game, I specifically consulted my older sister about relevant music knowledge. She is currently studying at Berklee College of Music, majoring in Composition and Music Production, so she has a professional understanding of rhythm, harmony, and melody arrangement. I asked her for advice on how to combine melodies and beats with players' operational actions—such as how to make drumbeats echo the feedback of shooting, and how to match the ups and downs of melodies with the rhythm of levels. Under her guidance, we finally co-created the music sheet designs for three levels, ensuring that the shooting operations in the game closely align with the rhythmic changes of the music.

第一关	生成时间	消失时间	第二关	生成时间	消失时间	第三关	生成时间	消失时间	第四关	生成时间	消失时间
00:02:17	00:04:15	00:05:18	00:06:19	00:06:12	00:06:15	00:06:10	00:06:12	00:06:15	00:10:14	00:11:00	00:11:00
00:04:15	00:06:12	00:06:15	00:06:08	00:06:08	00:06:08	00:12:22	00:12:27	00:12:28	00:13:18	00:14:16	00:16:02
00:05:13	00:07:10	00:07:10	00:07:08	00:07:08	00:07:08	00:15:29	00:18:12	00:18:12	00:17:02	00:18:18	00:19:18
00:06:12	00:08:09	00:08:09	00:11:28	00:11:28	00:11:28	00:19:07	00:19:25	00:19:25	00:20:09	00:20:25	00:20:25
00:07:10	00:09:06	00:09:06	00:12:14	00:12:14	00:12:14	00:20:09	00:20:22	00:20:22	00:20:07	00:21:02	00:21:02
00:08:09	00:10:06	00:10:06	00:13:22	00:13:22	00:13:22	00:20:25	00:21:15	00:21:15	00:22:09	00:23:21	00:23:21
00:09:06	00:11:04	00:11:04	00:13:22	00:13:22	00:13:22	00:21:25	00:22:10	00:22:10	00:22:09	00:23:21	00:23:21
00:10:06	00:12:04	00:12:04	00:14:18	00:14:18	00:14:18	00:22:14	00:23:03	00:23:03	00:24:19	00:26:05	00:26:05
00:11:04	00:13:01	00:13:01	00:17:19	00:17:19	00:17:19	00:23:07	00:23:28	00:23:28	00:27:06	00:28:04	00:28:04
00:12:04	00:14:01	00:14:01	00:18:17	00:18:17	00:18:17	00:24:02	00:24:21	00:24:21	00:29:21	00:30:06	00:30:06
00:13:01	00:15:00	00:15:00	00:19:18	00:19:18	00:19:18	00:24:26	00:25:15	00:25:15	00:30:19	00:31:10	00:31:10
00:14:01	00:15:28	00:15:28	00:20:14	00:20:14	00:20:14	00:25:29	00:26:10	00:26:10	00:31:45	00:32:18	00:32:18
00:15:00	00:16:27	00:16:27	00:21:09	00:21:09	00:21:09	00:26:34	00:27:05	00:27:05	00:33:05	00:35:04	00:35:04
00:15:28	00:17:24	00:17:24	00:23:10	00:23:10	00:23:10	00:27:07	00:27:28	00:27:28	00:35:12	00:36:11	00:36:11
00:16:27	00:18:22	00:18:22	00:23:08	00:23:08	00:23:08	00:28:02	00:28:22	00:28:22	00:36:20	00:37:18	00:37:18
00:17:24	00:19:20	00:19:20	00:24:06	00:24:06	00:24:06	00:28:27	00:29:15	00:29:15	00:38:07	00:40:03	00:40:03
00:18:22	00:20:19	00:20:19	00:25:06	00:25:06	00:25:06	00:29:23	00:30:10	00:30:10	00:40:14	00:41:11	00:41:11
00:19:20	00:21:19	00:21:19	00:26:29	00:26:29	00:26:29	00:30:14	00:31:04	00:31:04	00:41:20	00:42:00	00:42:00
00:20:19	00:22:17	00:22:17	00:27:11	00:27:11	00:27:11	00:31:04	00:32:02	00:32:02	00:42:29	00:43:26	00:43:26
00:21:19	00:23:15	00:23:15	00:27:21	00:27:21	00:27:21	00:32:04	00:32:22	00:32:22	00:45:15	00:46:12	00:46:12
00:22:17	00:24:14	00:24:14	00:30:21	00:30:21	00:30:21	00:35:10	00:36:29	00:36:29	00:47:12	00:48:29	00:48:29
00:23:15	00:25:14	00:25:14	00:32:19	00:32:19	00:32:19	00:38:15	00:40:01	00:40:01	00:50:26	00:52:08	00:52:08
00:24:14	00:26:13	00:26:13	00:33:17	00:33:17	00:33:17	00:41:22	00:43:11	00:43:11	00:53:12	00:54:21	00:54:21
00:25:14	00:27:09	00:27:09	00:34:15	00:34:15	00:34:15	00:44:22	00:45:00	00:45:00	00:55:00	00:56:00	00:56:00
00:26:13	00:28:09	00:28:09	00:35:19	00:35:19	00:35:19	00:45:12	00:46:09	00:46:09	00:58:14	00:59:21	00:59:21
00:27:09	00:29:09	00:29:09	00:36:12	00:36:12	00:36:12	00:46:18	00:47:25	00:47:25	00:00:29	00:02:09	00:02:09
00:28:09	00:30:08	00:30:08	00:37:09	00:37:09	00:37:09	00:48:07	00:49:14	00:49:14	01:02:16	01:04:23	01:04:23
00:29:09	00:31:06	00:31:06	00:38:07	00:38:07	00:38:07	00:49:28	00:51:01	00:51:01	01:06:01	01:07:08	01:07:08
00:30:08	00:32:03	00:32:03	00:39:08	00:39:08	00:39:08	00:51:13	00:52:01	00:52:01	01:08:16	01:09:13	01:09:13
00:31:06	00:33:04	00:33:04	00:39:15	00:39:15	00:39:15	00:53:00	00:54:09	00:54:09	01:11:29	01:12:29	01:12:29
00:32:03	00:34:02	00:34:02	00:39:20	00:39:20	00:39:20	00:54:20	00:55:27	00:55:27	01:12:10	01:13:09	01:13:09
00:33:04	00:35:01	00:35:01	00:39:24	00:39:24	00:39:24	00:54:19	00:56:09	00:56:09	01:13:18	01:14:16	01:14:16
00:34:03	00:35:29	00:35:29	00:39:26	00:39:26	00:39:26	00:56:07	00:57:14	00:57:14	01:14:27	01:15:25	01:15:25
00:35:01	00:36:29	00:36:29	00:39:26	00:39:26	00:39:26	00:57:26	00:58:22	00:58:22	01:15:25	01:17:03	01:17:03
00:35:29	00:37:27	00:37:27	00:39:29	00:39:29	00:39:29	00:57:41	00:58:24	00:58:24	01:17:11	01:18:08	01:18:08
00:37:27	00:39:24	00:39:24	00:39:34	00:39:34	00:39:34	00:57:44	00:58:24	00:58:24	01:19:10	01:20:06	01:20:06
00:38:26	00:40:24	00:40:24	00:39:42	00:39:42	00:39:42	01:02:22	01:11:13	01:11:13	01:20:28	01:22:04	01:22:04
00:39:24	00:41:22	00:41:22	00:39:47	00:39:47	00:39:47	01:11:17	01:12:06	01:12:06	01:22:13	01:23:13	01:23:13
00:40:24	00:42:21	00:42:21	00:41:22	00:41:22	00:41:22	01:12:11	01:13:00	01:13:00	01:23:21	01:24:21	01:24:21
00:41:22	00:43:19	00:43:19	00:42:21	00:42:21	00:42:21	01:13:29	01:14:19	01:14:19	01:25:08	01:27:23	01:28:15
00:42:21	00:44:18	00:44:18	00:43:19	00:43:19	00:43:19	01:14:19	01:15:15	01:15:15	01:27:23	01:29:15	01:29:15
00:43:											

Technical Implementation

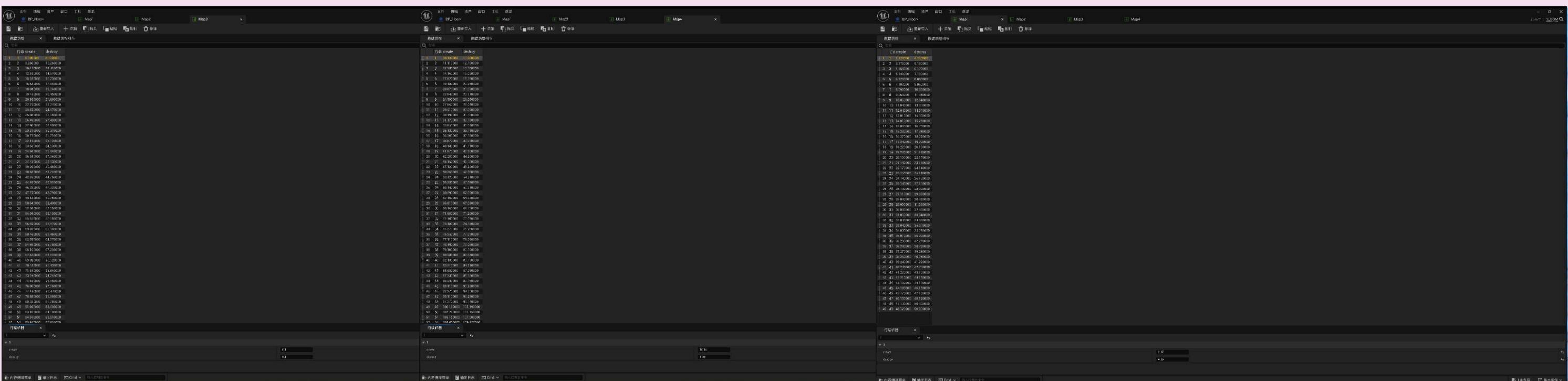
Platform Spawning in Level 1

For Level 1, I designed a simple platforming section. These platforms spawn in sync with the music's rhythm—some appearing quickly and others slowly—requiring players to react in time with the music's beat.



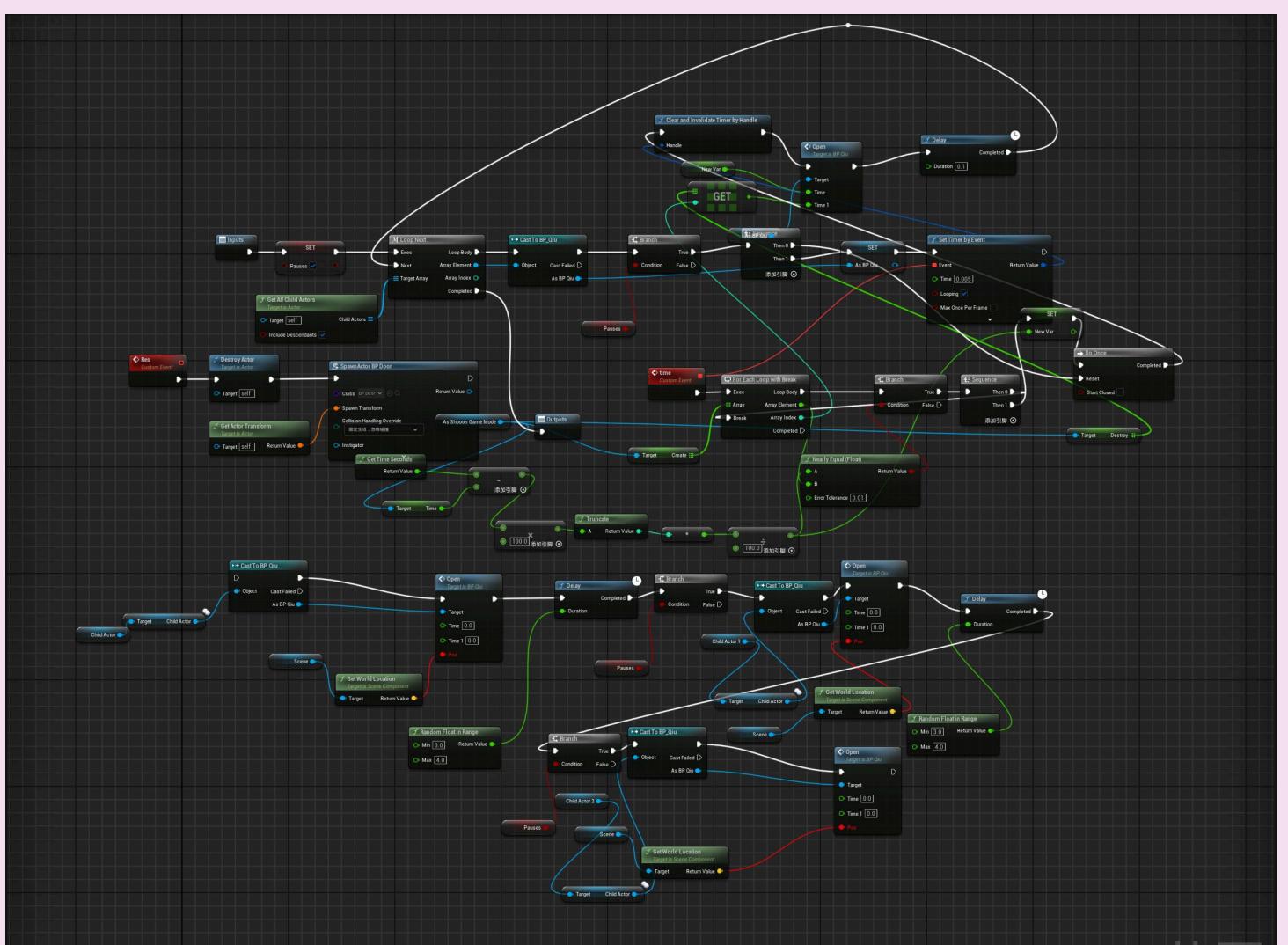
Music Sheets

There are three levels in total. Each level features distinct music and corresponds to a unique rhythm.



Kicking Down Doors

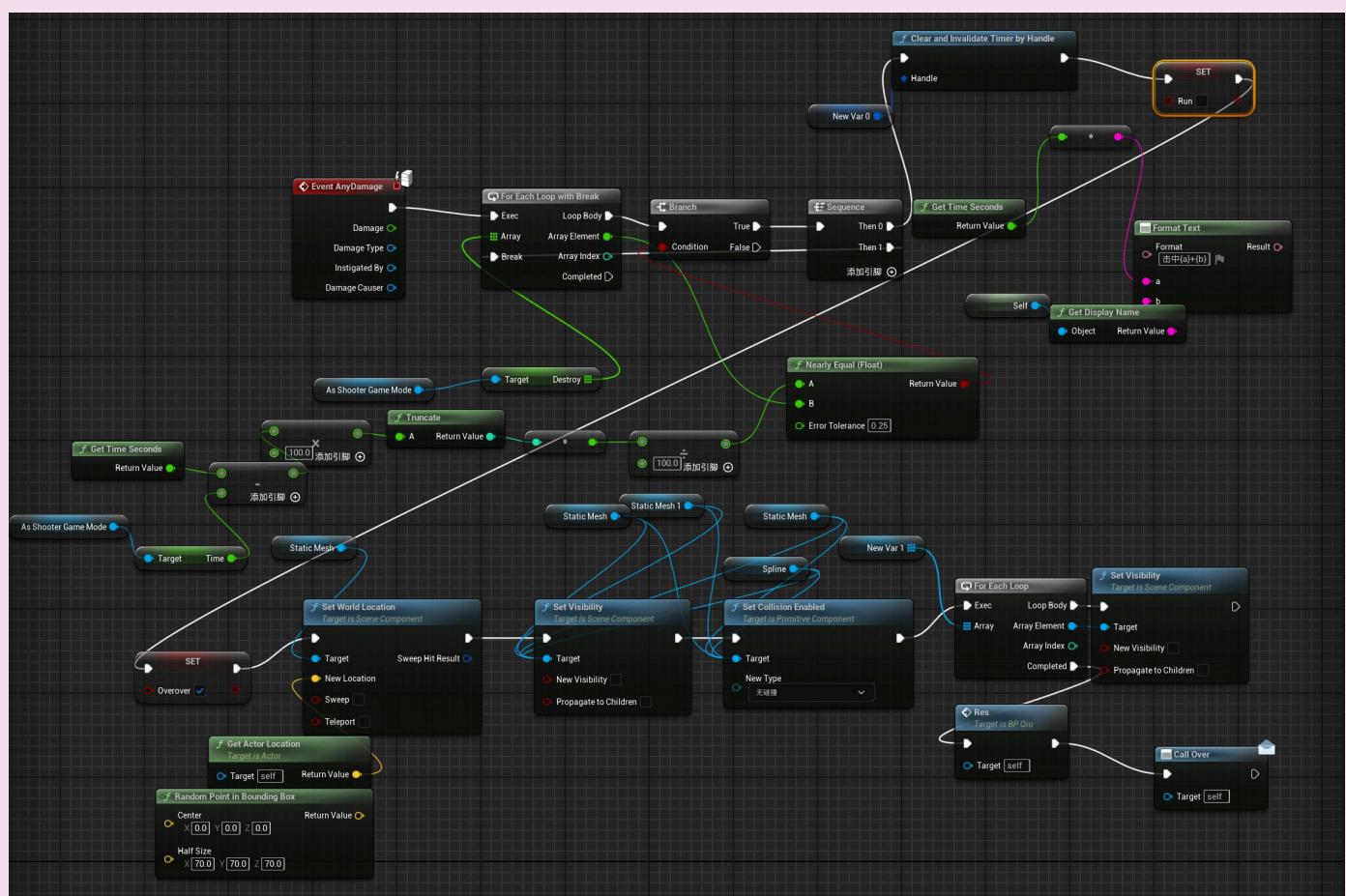
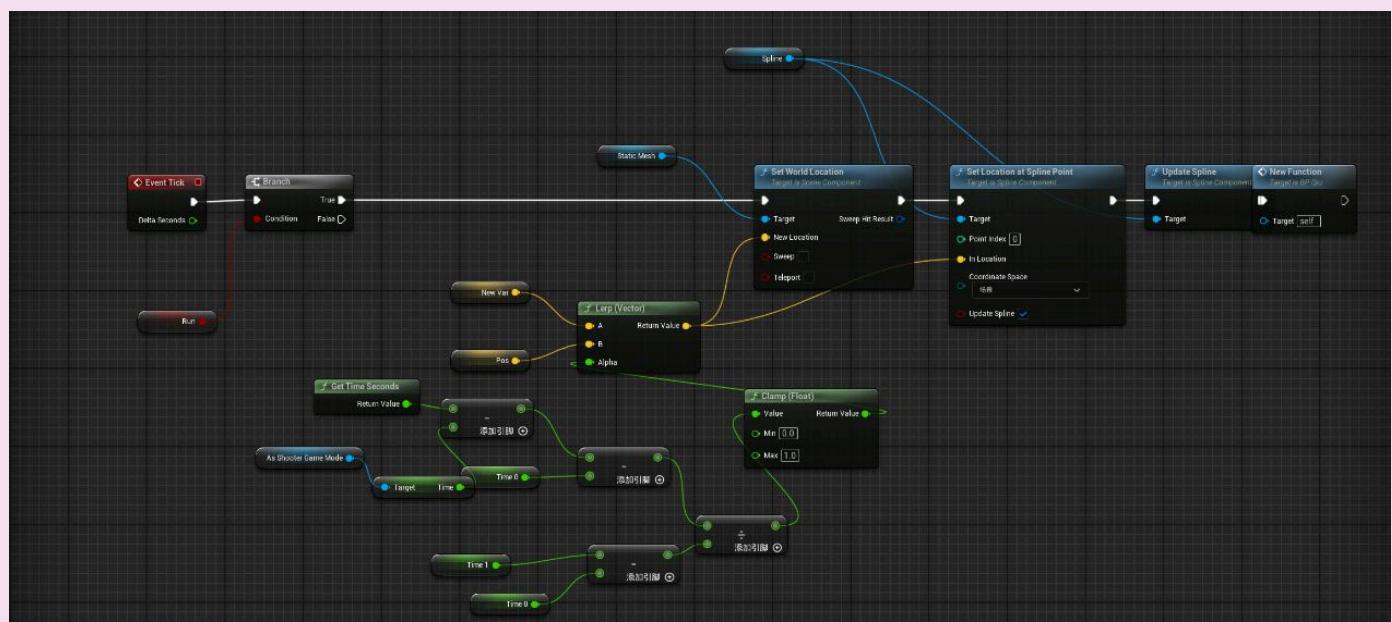
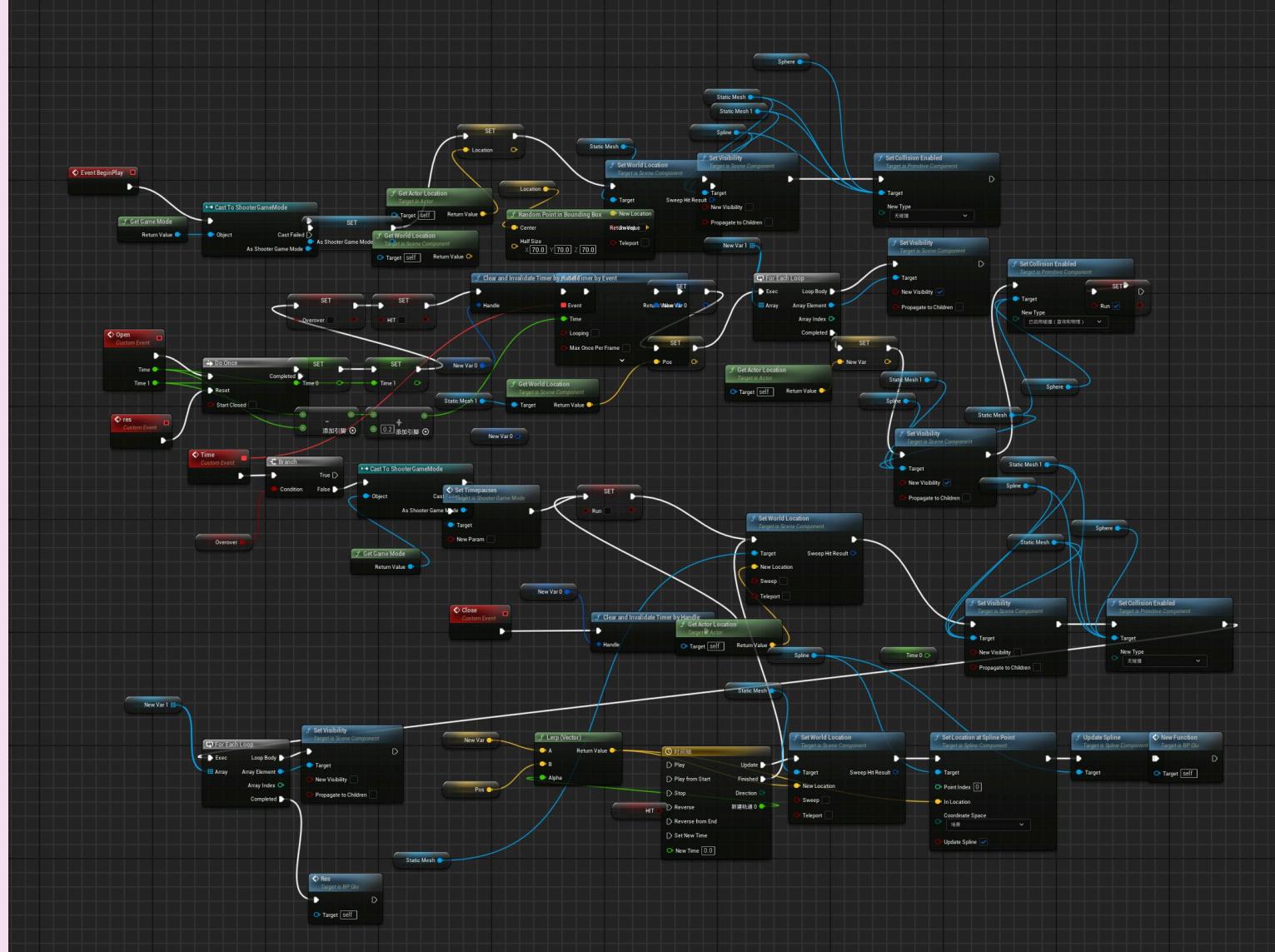
Some doors may have enemies hiding behind them—if players open these doors directly, they will be instantly defeated. Therefore, players need to enter the Music Space to kick down the doors.



Technical Implementation

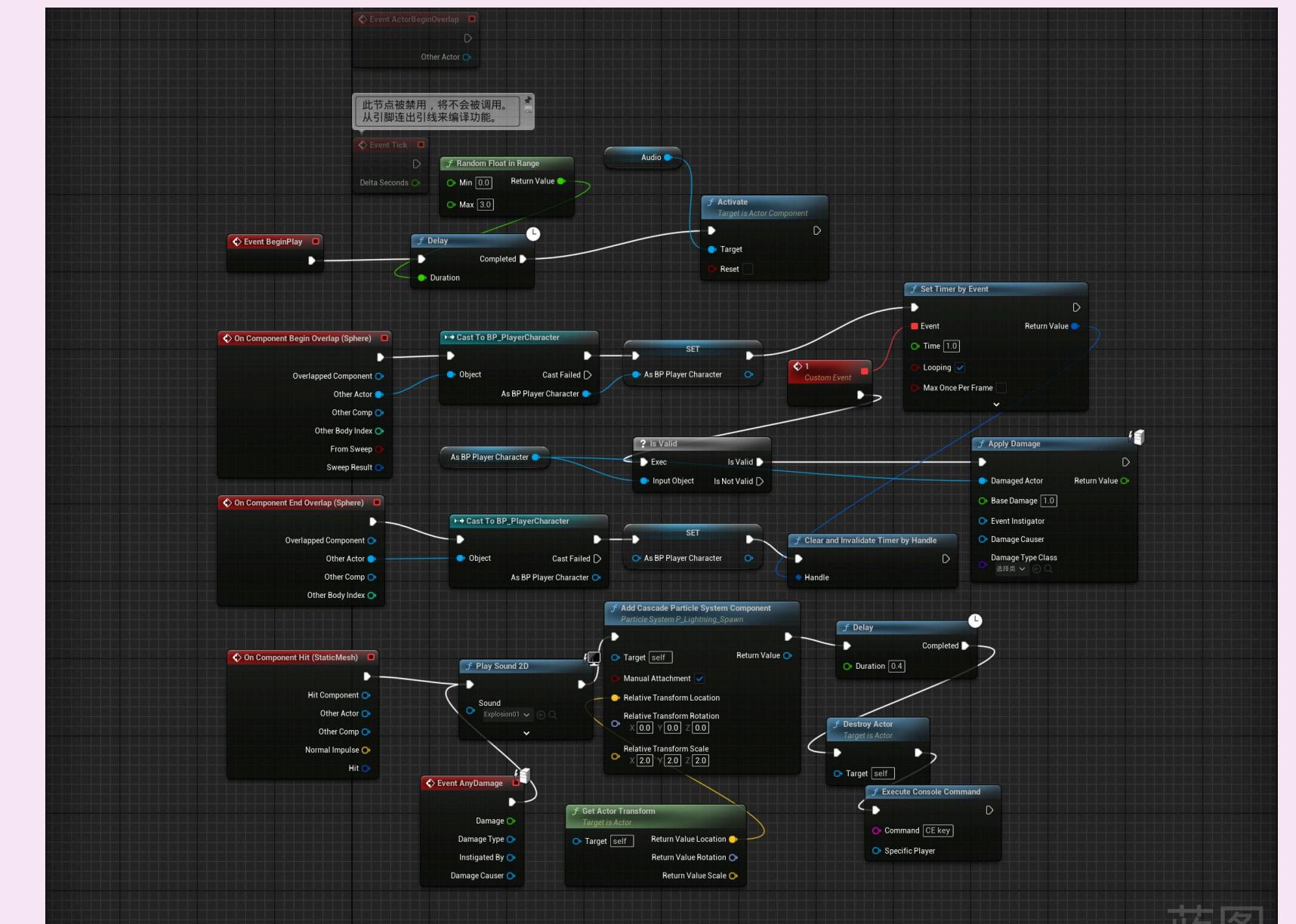
Switching to Music Space

Players can switch to the music space, where they can defeat enemies by clicking on musical notes.



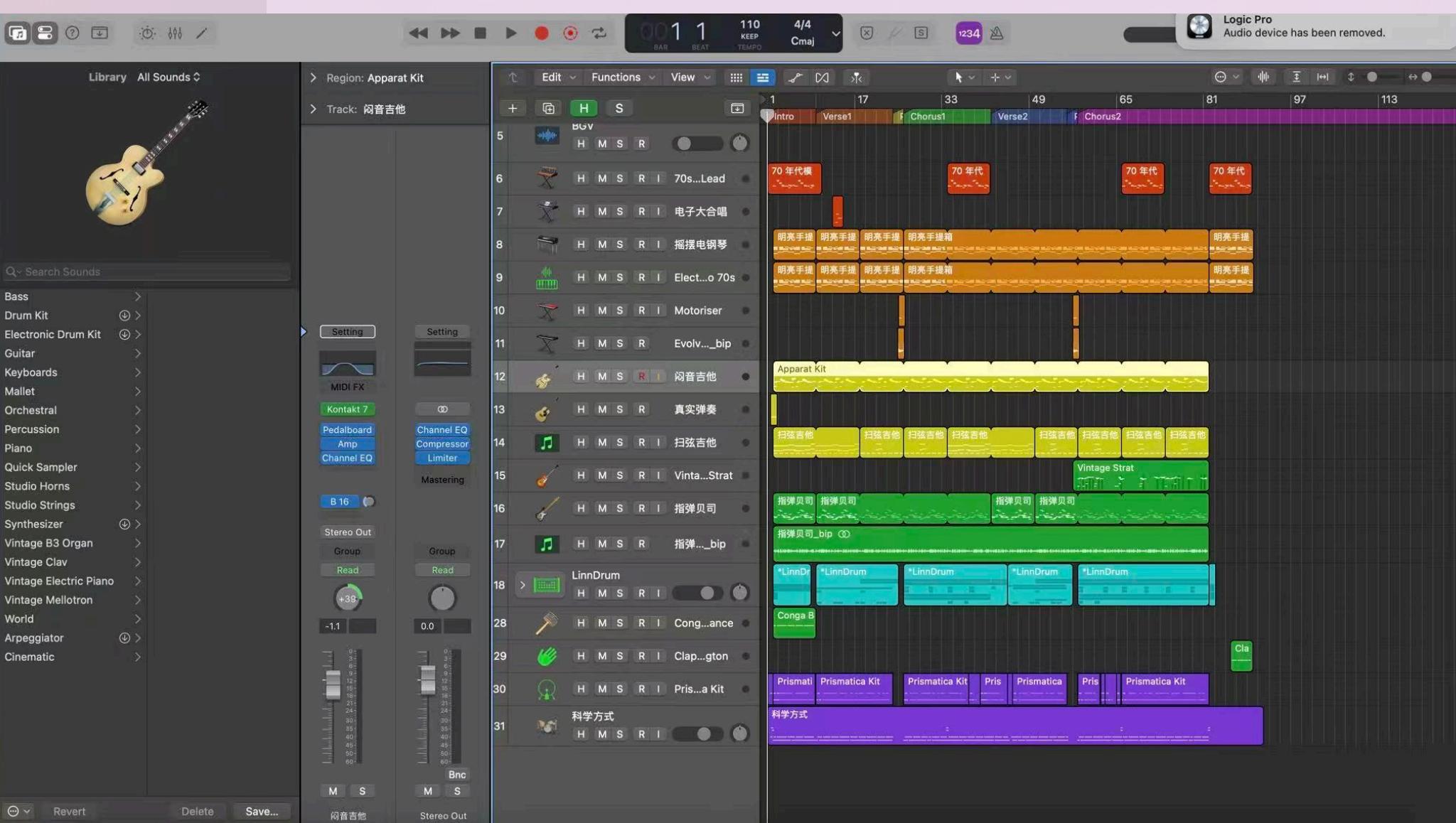
Noise

If players stay within the noise range, their health will continuously decrease.

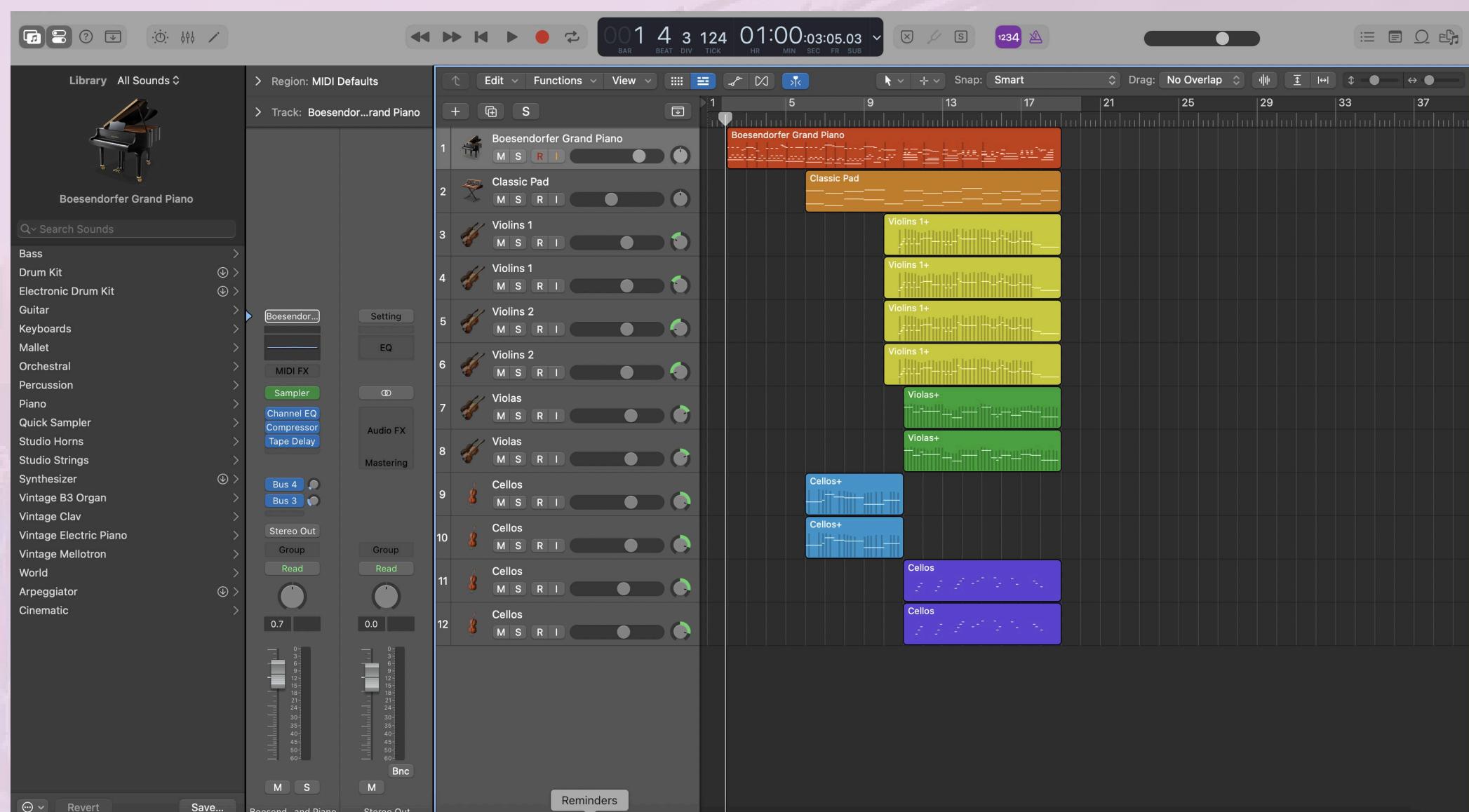


Remix

The original music didn't have such a strong sense of rhythm. Its rhythm was enhanced by using music production software to emphasize the accents.



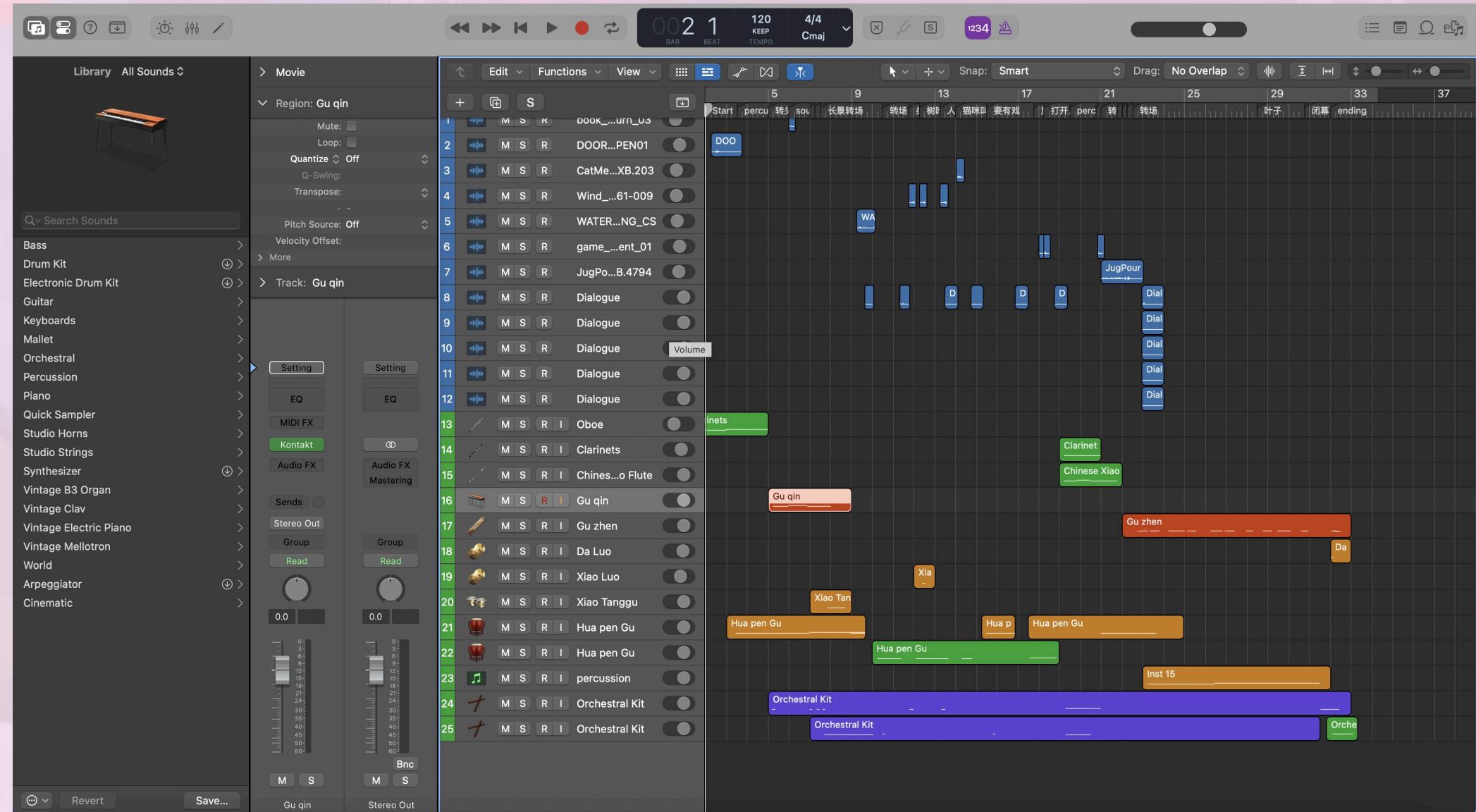
This screenshot shows a Logic Pro session titled 'Apparatus Kit'. The arrangement consists of 31 tracks, including various guitars, drums, and electronic sounds. The tracks are color-coded and show complex patterns. The left panel displays the 'Apparatus Kit' region, and the right panel shows the 'MIDI Defaults' region. The session is set to 4/4 time, 110 BPM, and C major.



This screenshot shows a Logic Pro session titled 'Boesendorfer Grand Piano'. The arrangement features 12 tracks of piano and various string instruments. The left panel displays the 'Boesendorfer Grand Piano' region, and the right panel shows the 'MIDI Defaults' region. The session is set to 4/4 time, 124 BPM, and C major.

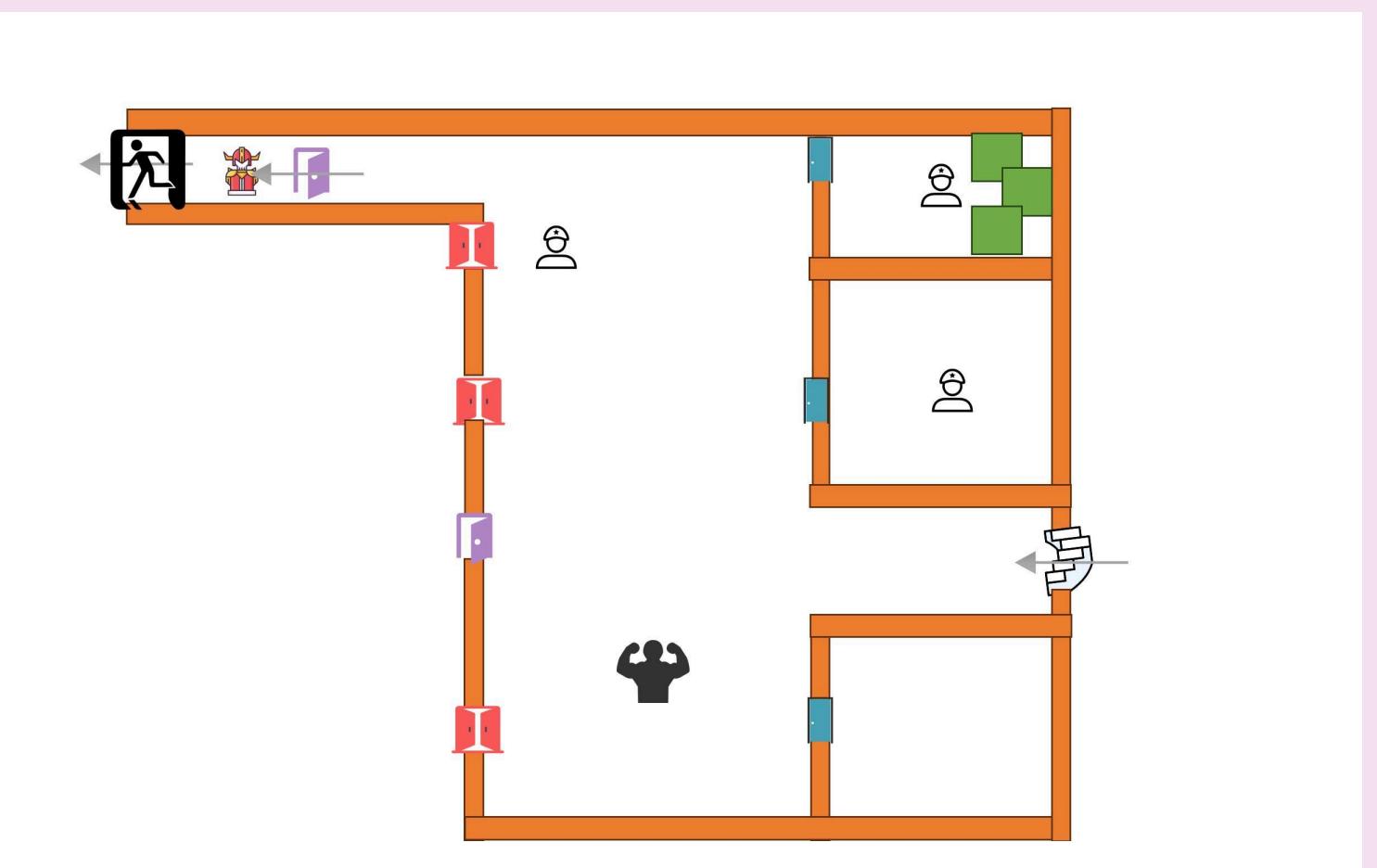
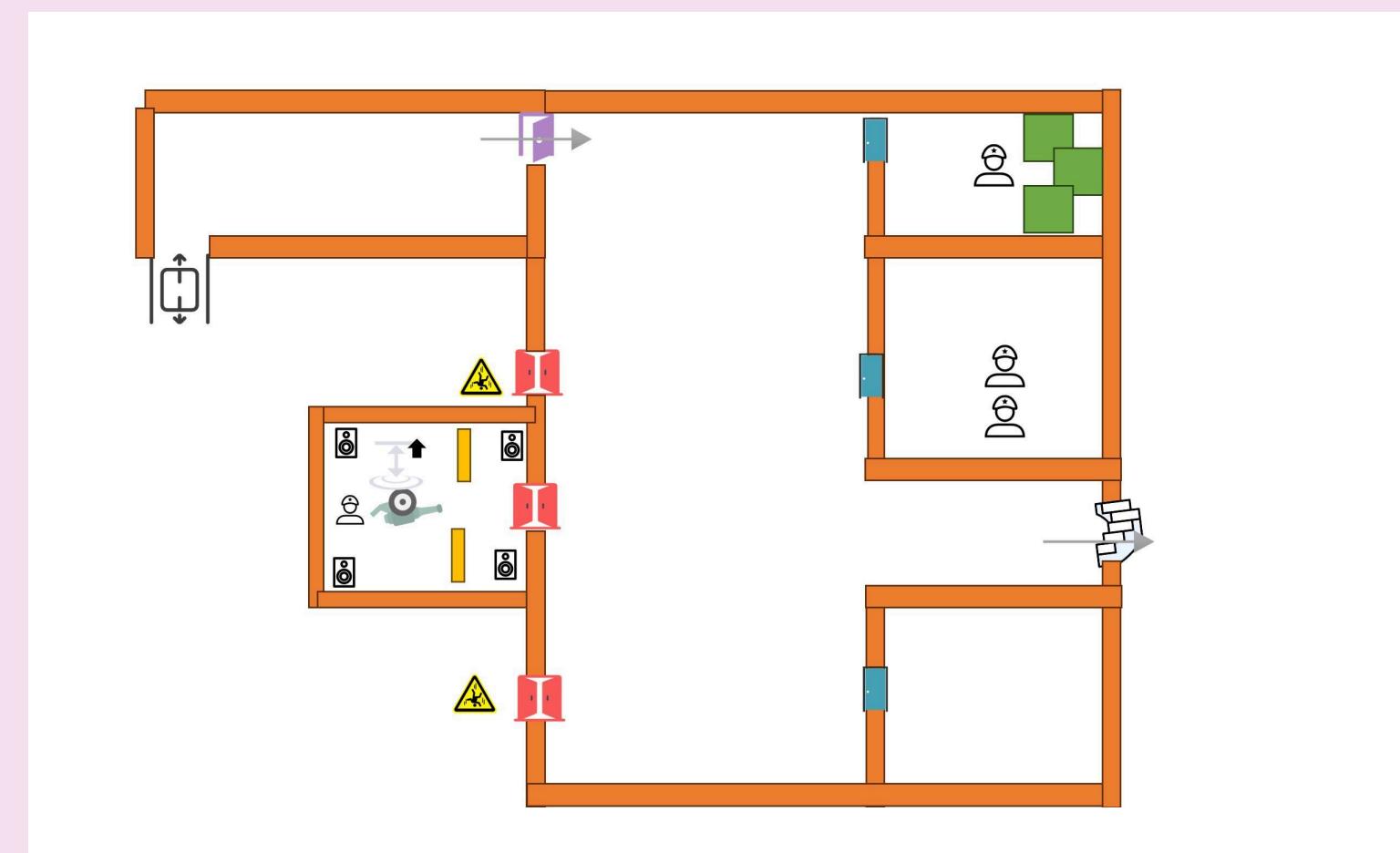
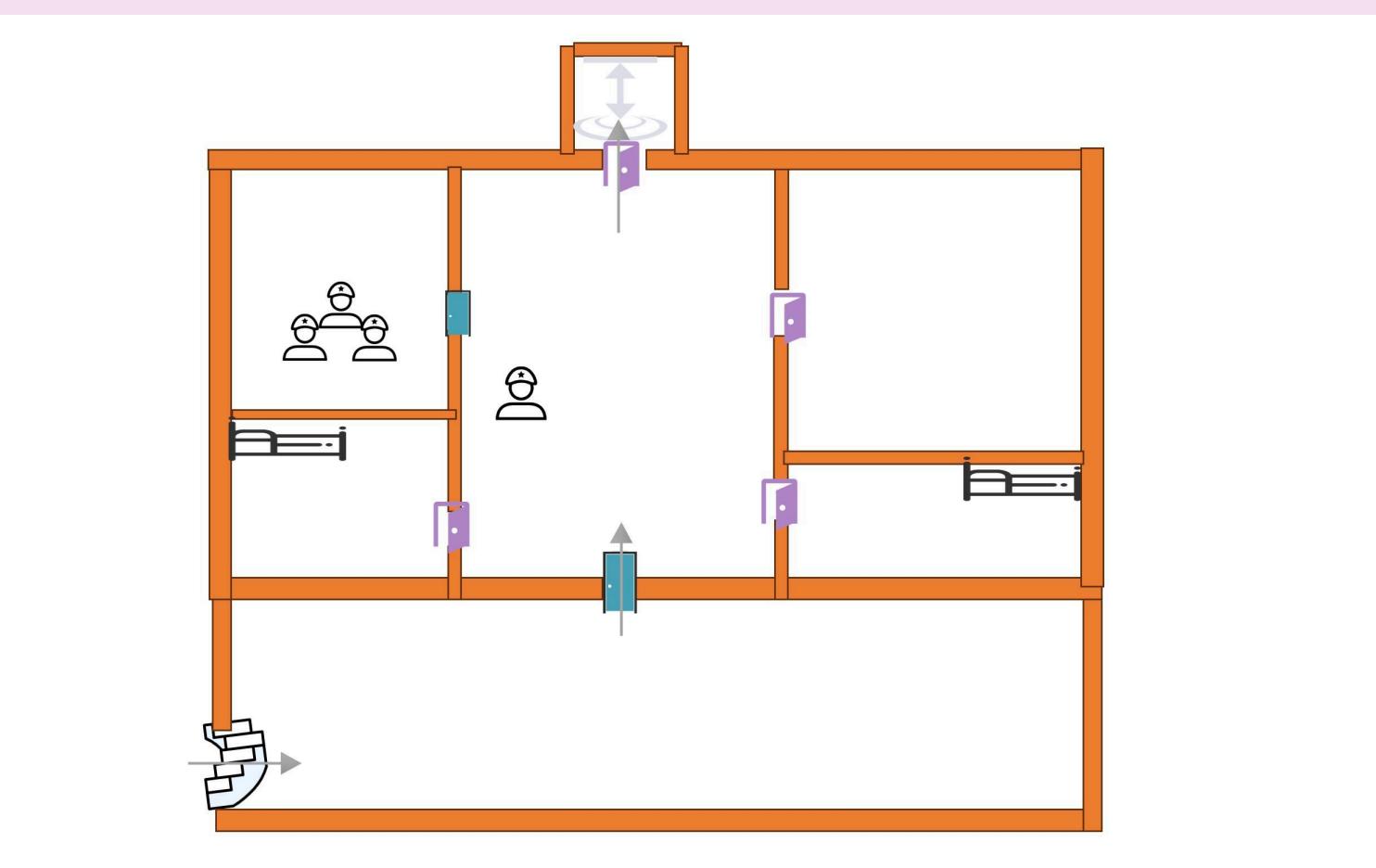
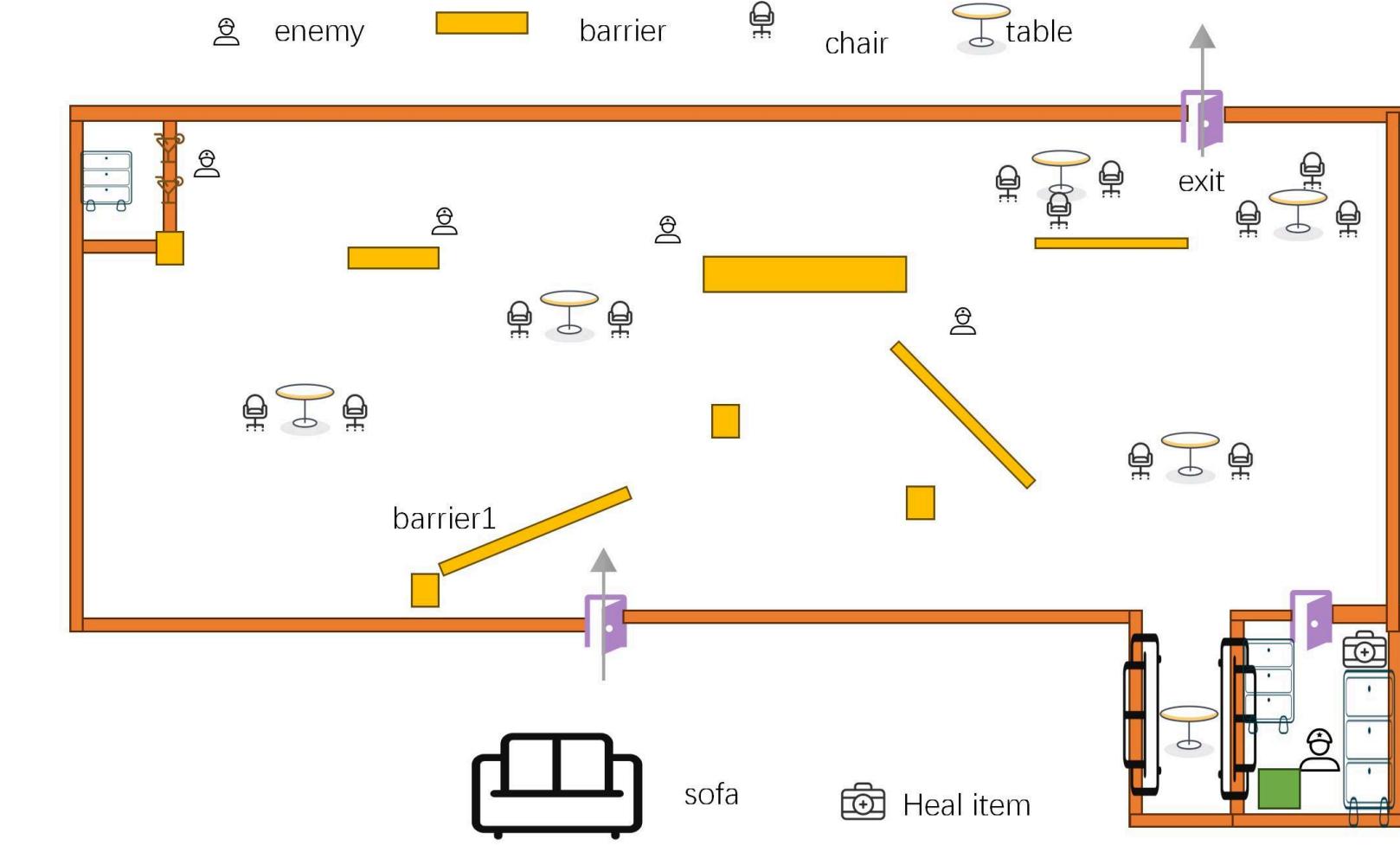
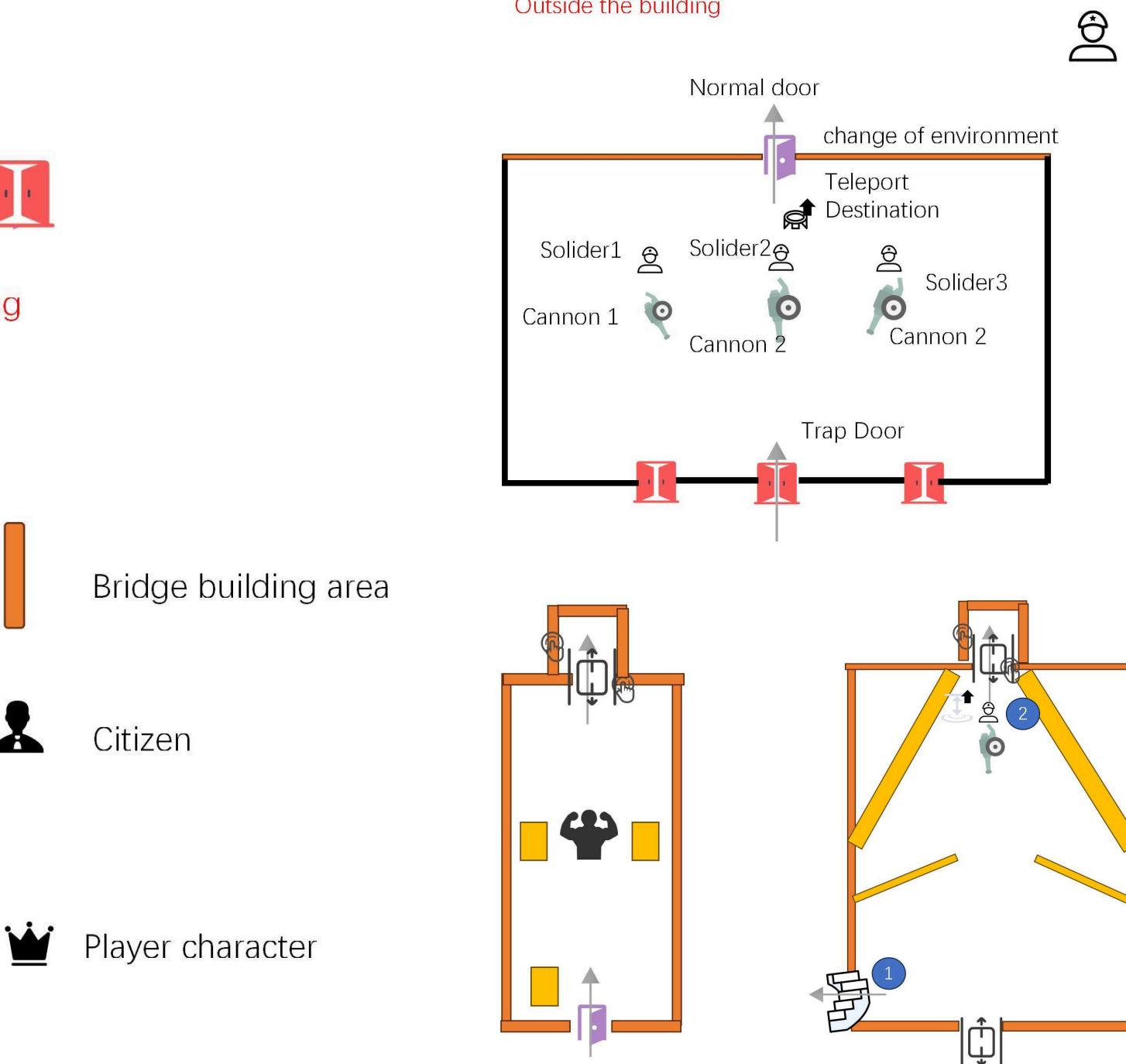
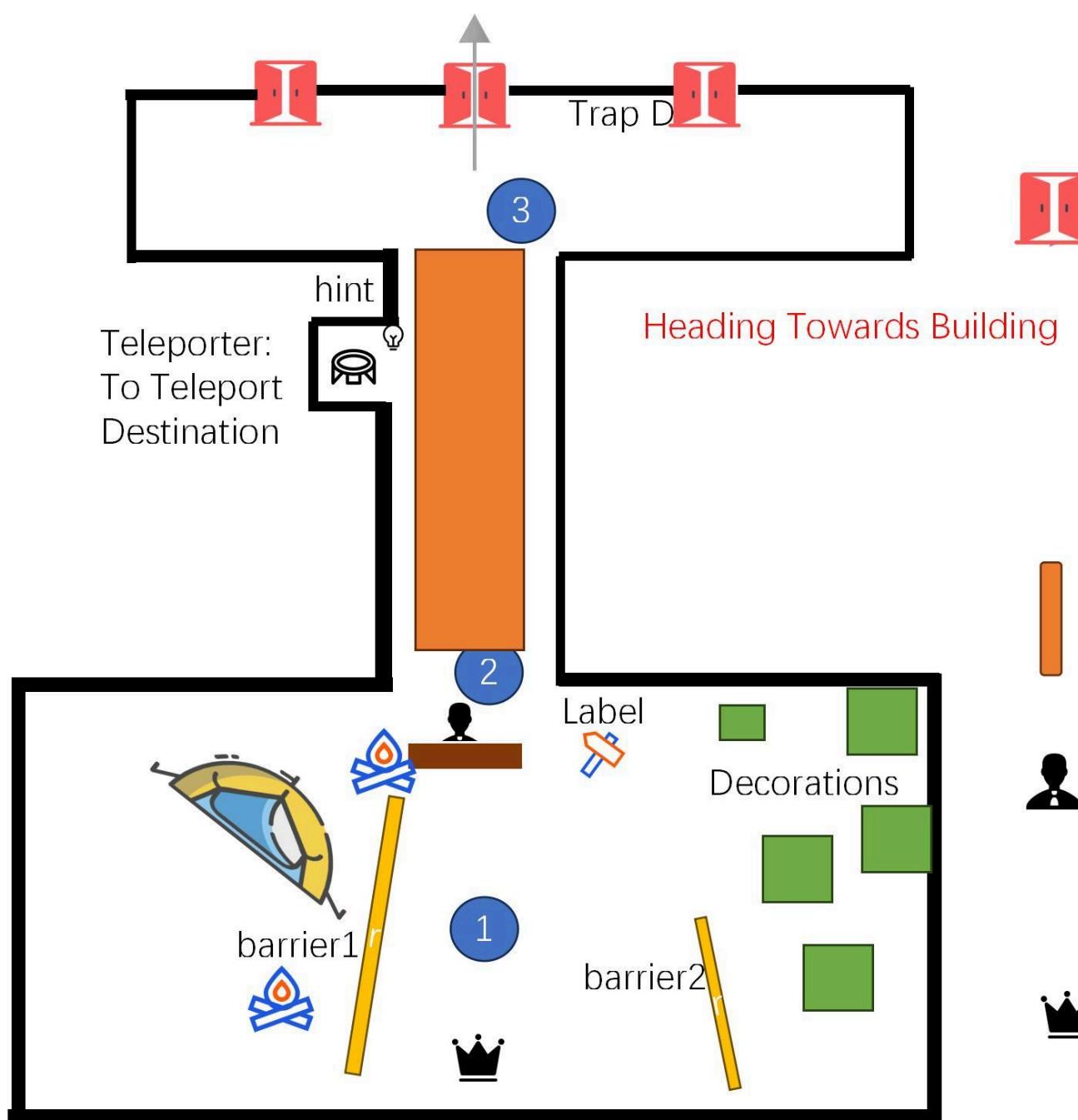


This screenshot shows a Logic Pro session titled 'Acoustic Guitar'. The arrangement features 44 tracks, primarily acoustic guitars and various effects. The left panel displays the 'Acoustic Guitar' region, and the right panel shows the 'MIDI Defaults' region. The session is set to 4/4 time, 74 BPM, and A major.



This screenshot shows a Logic Pro session titled 'Gu qin'. The arrangement features 25 tracks of traditional Chinese instruments like Gu qin, Gu zhen, and Da. The left panel displays the 'Gu qin' region, and the right panel shows the 'Movie' region. The session is set to 4/4 time, 120 BPM, and C major.

Map Design



ECHOES OF CANVAS

LINK: <https://youtu.be/mBUkks1LNik>

Echoes of Canvas is a VR game that immerses players in the world of paintings. Instead of being mere onlookers, they become travelers wandering between colors and memories.

Every brushstroke is a path, and every shadow hides a story. As you explore classic paintings reimagined as 3D spaces, the line between art and reality begins to blur. Once static images now awaken with breath, whispers, and memories.

Through light, texture, and sound, players will gradually piece together fragments of the painter's soul—and perhaps even find a reflection of themselves within the canvas.



Inspiration

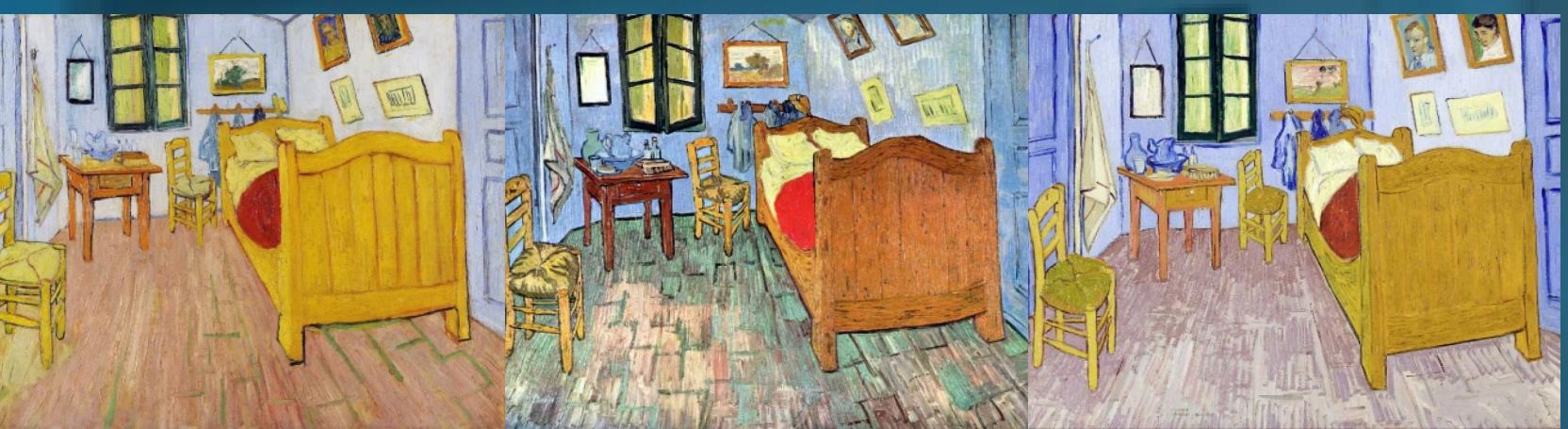


The inspiration for this game came from a serendipitous discovery during my summer trip—a Van Gogh-themed art museum. By constructing a three-dimensional space, it made visitors feel as if they had stepped into the world of Van Gogh's paintings. This immersive experience left me deeply inspired.

That's when I began to wonder: Could I use a similar approach to let players personally experience the works of art I love? I believe VR is the perfect medium to achieve this sense of immersion.

I have a particular fondness for the works of Dalí and Van Gogh, and both of their lives are closely tied to the imagery of "wine." Therefore, I used "wine" as a thread connecting the two artists, and built the game's narrative inspired by their life stories. This allows players to experience the passion and contradictions in their hearts within the virtual world, truly "stepping into" the artists' souls.

Research



The inspiration for this game stems from my research on Van Gogh's The Yellow House series. Throughout his life, Van Gogh depicted the same scene—the yellow house that symbolized "home"—three times. Each creation emerged at a different stage of his life and reflected profound shifts in his state of mind.

The First Yellow House (September 1888)

This piece was painted shortly after Van Gogh arrived in Arles and moved into the yellow house. At the time, he was filled with ideals and hope, envisioning a "Studio of the South" with his friend Gauguin—a place to find spiritual belonging through art and friendship. This marked the happiest period of his life, as well as the most brilliant phase of his artistic career.

The Second Yellow House (1889)

This version was created during Van Gogh's stay at the Saint-Rémy Asylum. His mental state was unstable then, oscillating frequently between lucidity and confusion. In this painting, yellow no longer symbolized warmth; instead, it became a cry of despair against reality. Yet even amid his suffering, his creative power remained extraordinary, radiating a tragic yet powerful vitality.

The Third Yellow House (Late 1889)

This version was created during Van Gogh's stay at the Saint-Rémy Asylum. His mental state was unstable then, oscillating frequently between lucidity and confusion. In this painting, yellow no longer symbolized warmth; instead, it became a cry of despair against reality. Yet even amid his suffering, his creative power remained extraordinary, radiating a tragic yet powerful vitality.

These three Yellow House paintings are not only a microcosm of Van Gogh's artistic career, but also a portrait of his life's journey—from hope to fragmentation, from warmth to loneliness, and from an ideal home to a spiritual collapse.

In my game, I aim to let players step into these three "yellow houses" in person, experience the emotional ups and downs and mental changes Van Gogh went through at different stages, and feel how he used his brush to depict the deepest struggles and yearnings of the human soul.



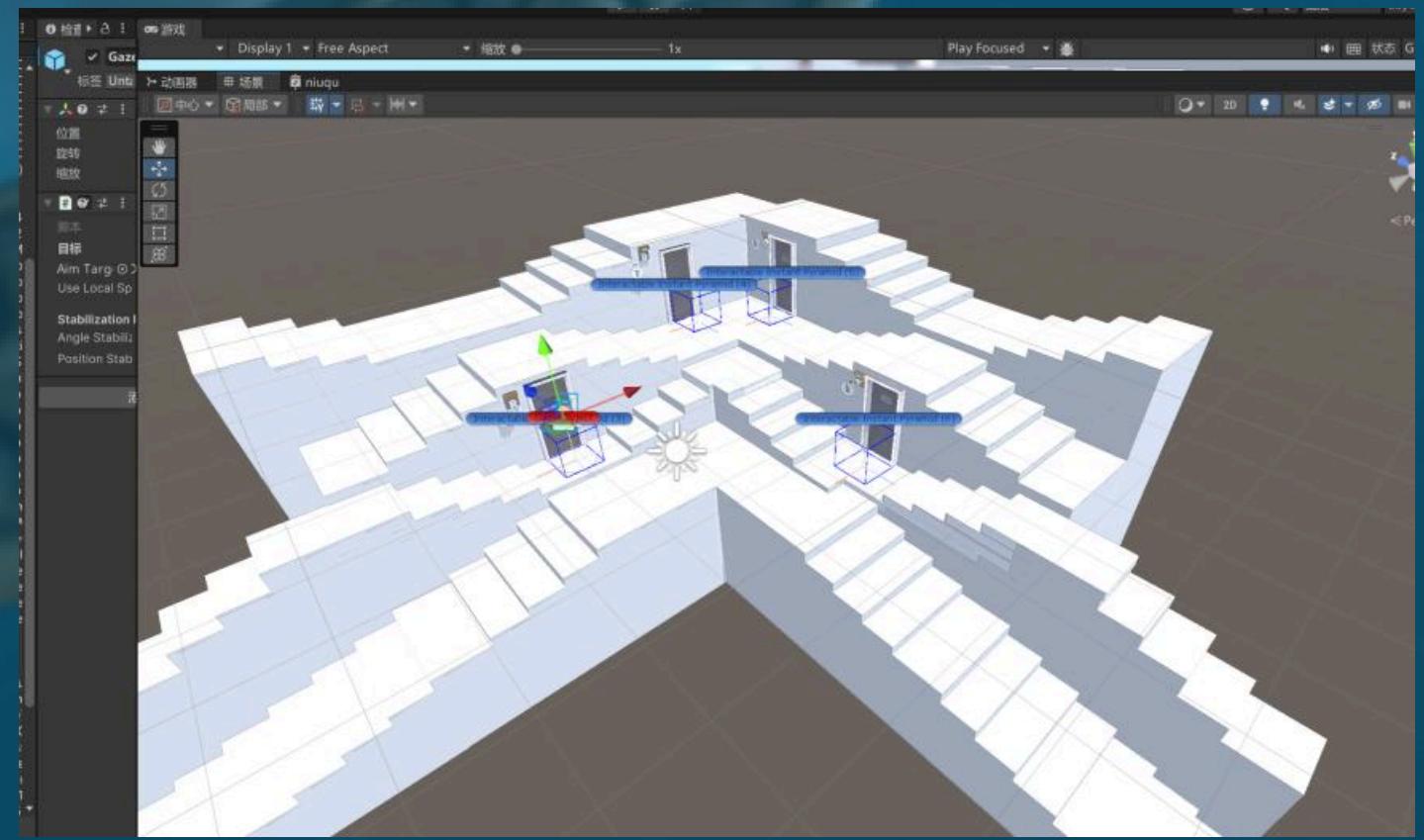
During the period when he created this painting, Dalí had not yet indulged in a life of luxury and pleasure. He focused more on exploring reason and the subconscious, rather than relying on substances (such as alcohol) to escape reality.

The emptiness, silence, and cool-toned colors in the painting perfectly reflect the restraint and clarity in his mind—a state of mind where the artist examined dreams with reason and gazed at chaos with calmness.

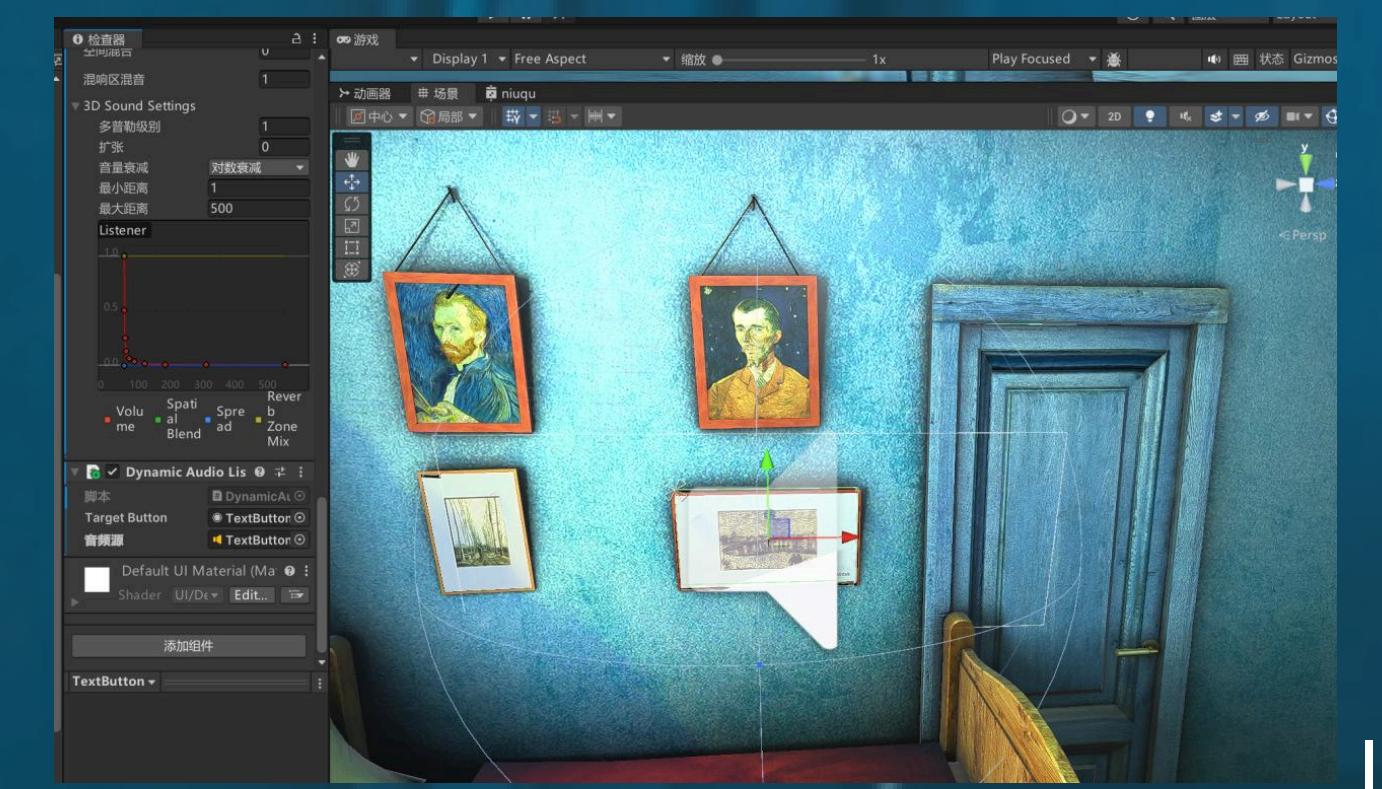
This work not only showcases the purity and depth of Dalí's early thoughts, but can also be said to be the creation that best represents the spiritual core of his entire life.

Technical Implementation

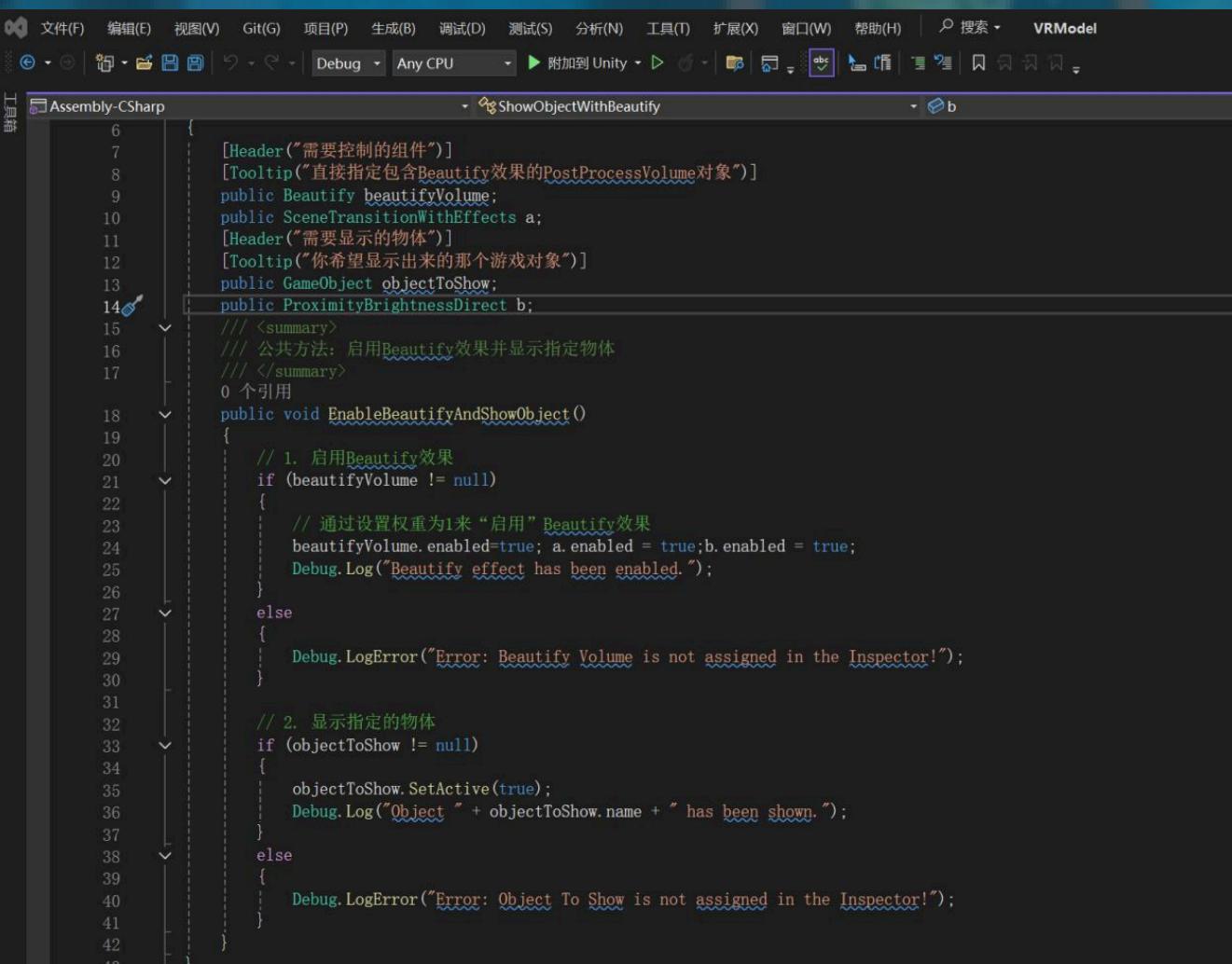
Scene Transition Management



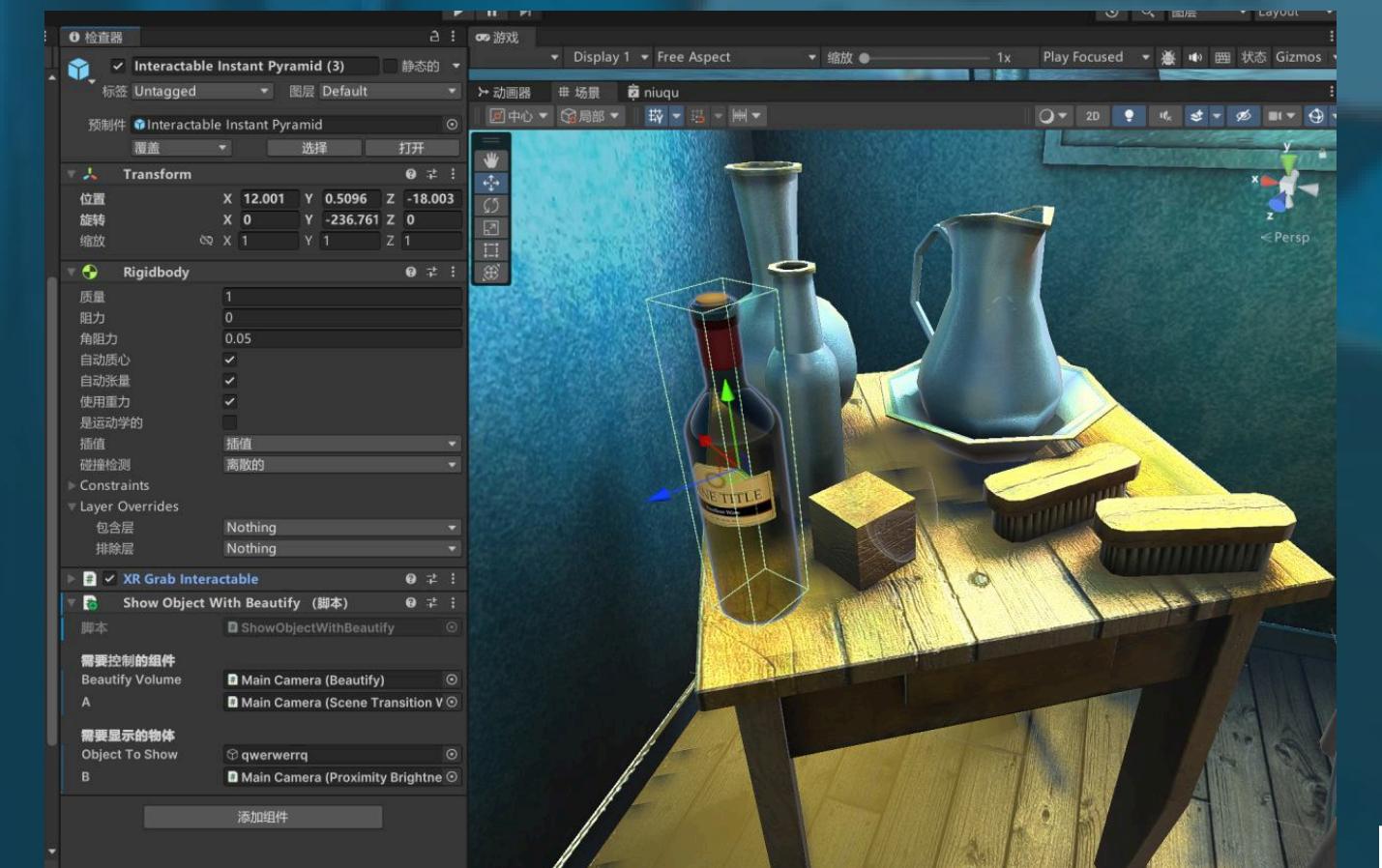
Audio Playback Logic



| **Scene Blur Effect:** The subsequent light stretching and color tone adjustments follow the same logic (e.g., for interactions like picking up wine glasses/bottles and alarm clocks).

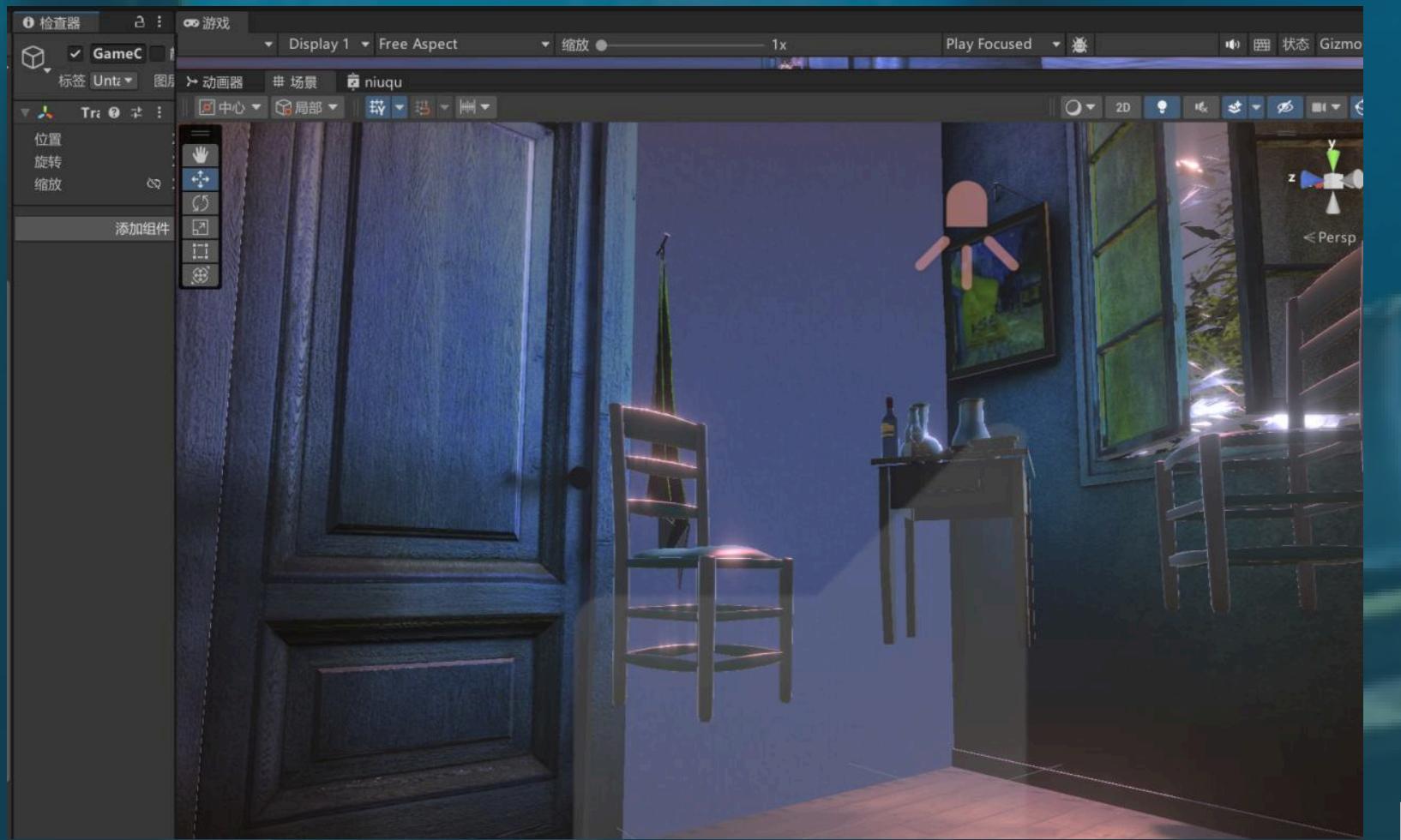


```
6  {
7      [Header("需要控制的组件")]
8      [Tooltip("直接指定包含Beautify效果的PostProcessVolume对象")]
9      public Beautify beautifyVolume;
10     public SceneTransitionWithEffects a;
11     [Header("需要显示的物体")]
12     [Tooltip("你希望显示出来的那个游戏对象")]
13     public GameObject objectToShow;
14     public ProximityBrightnessDirect b;
15     /// <summary>
16     /// 公共方法: 启用Beautify效果并显示指定物体
17     /// </summary>
18     0 个引用
19     public void EnableBeautifyAndShowObject()
20     {
21         // 1. 启用Beautify效果
22         if (beautifyVolume != null)
23         {
24             // 通过设置权重为1来“启用”Beautify效果
25             beautifyVolume.enabled=true; a.enabled = true;b.enabled = true;
26             Debug.Log("Beautify effect has been enabled.");
27         }
28         else
29         {
30             Debug.LogError("Error: Beautify Volume is not assigned in the Inspector!");
31         }
32         // 2. 显示指定的物体
33         if (objectToShow != null)
34         {
35             objectToShow.SetActive(true);
36             Debug.Log("Object " + objectToShow.name + " has been shown.");
37         }
38         else
39         {
40             Debug.LogError("Error: Object To Show is not assigned in the Inspector!");
41         }
42     }
43 }
```

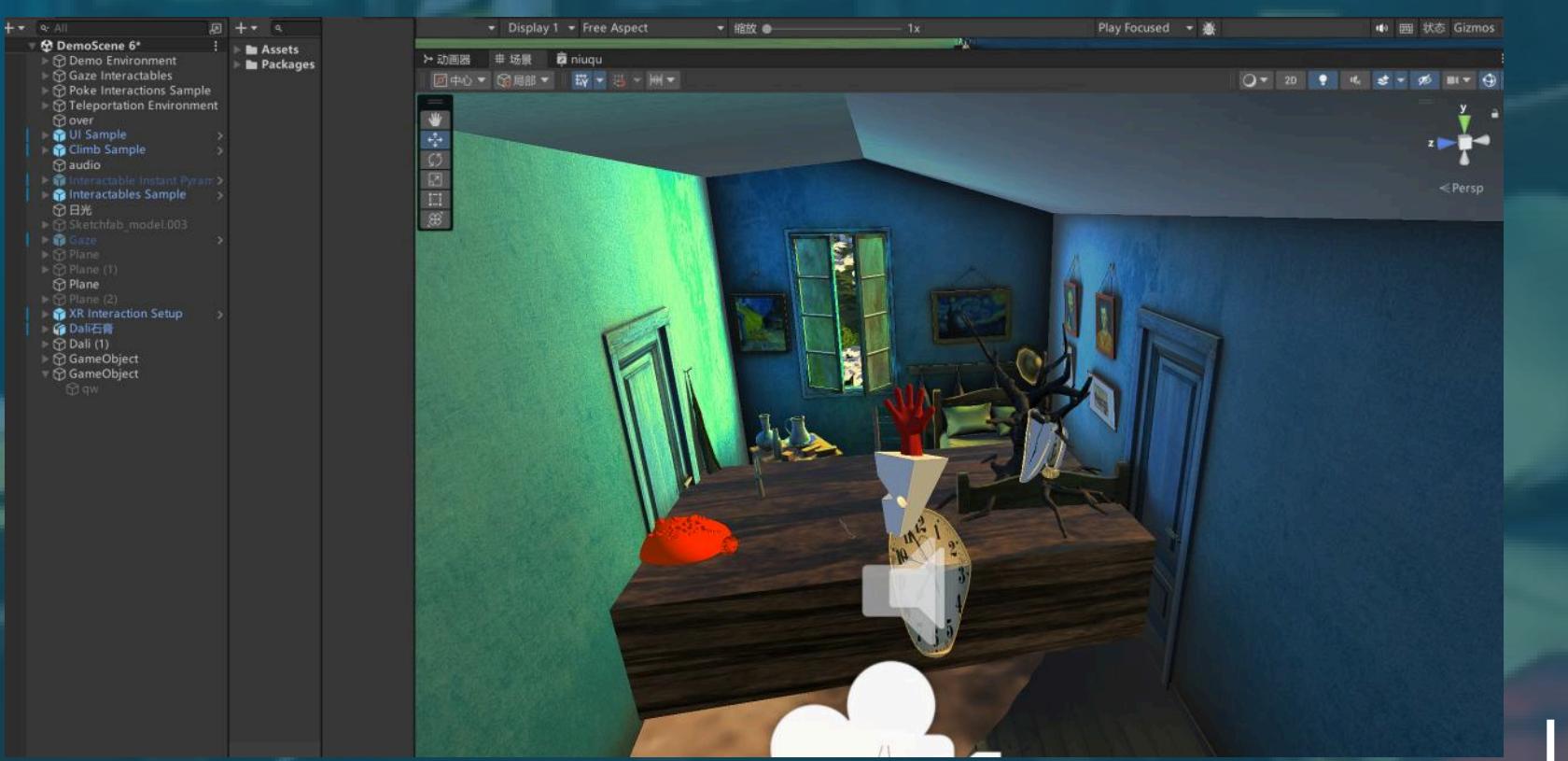


Technical Implementation

Floating-in-Air Effect

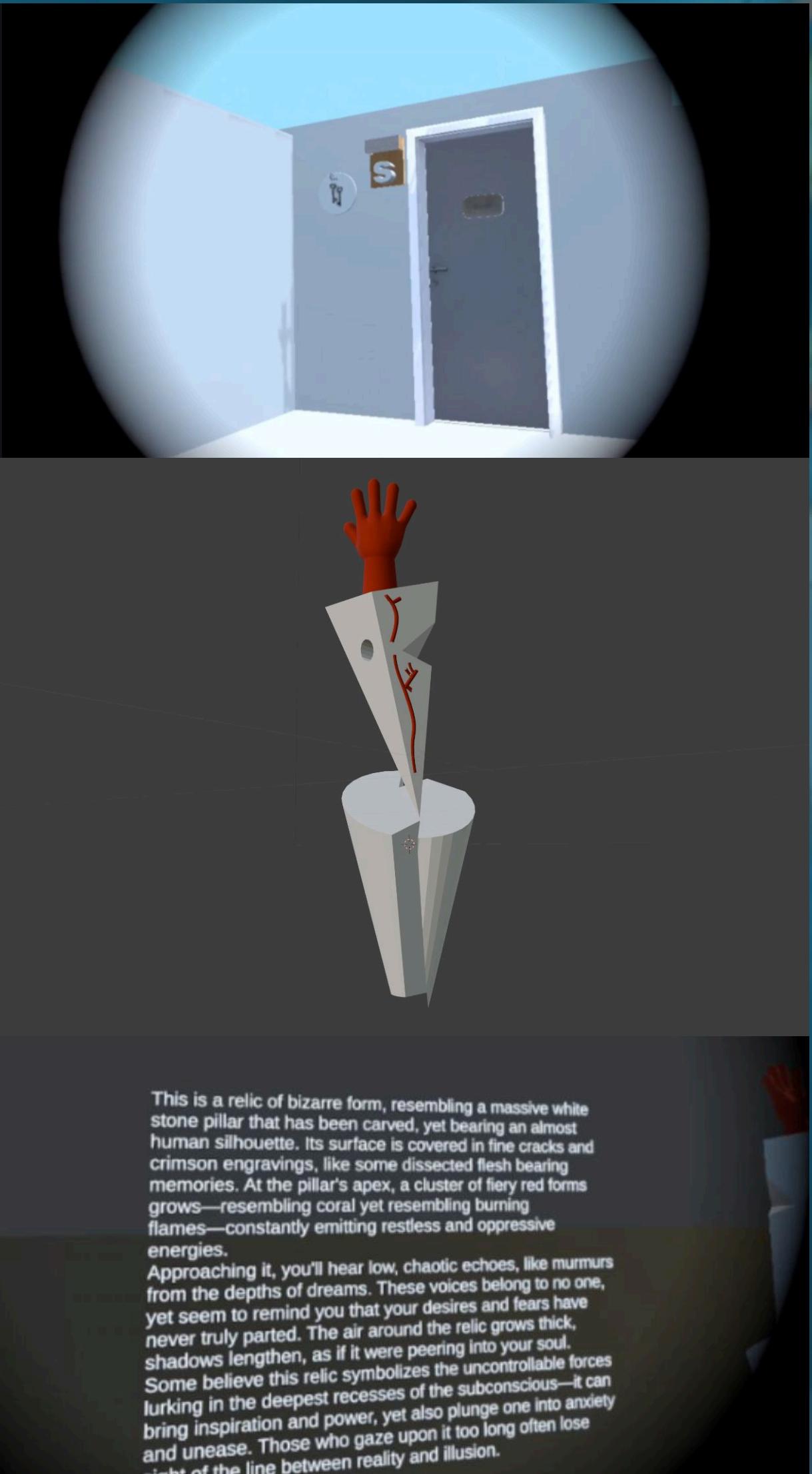


Three-Scene Flicker Function

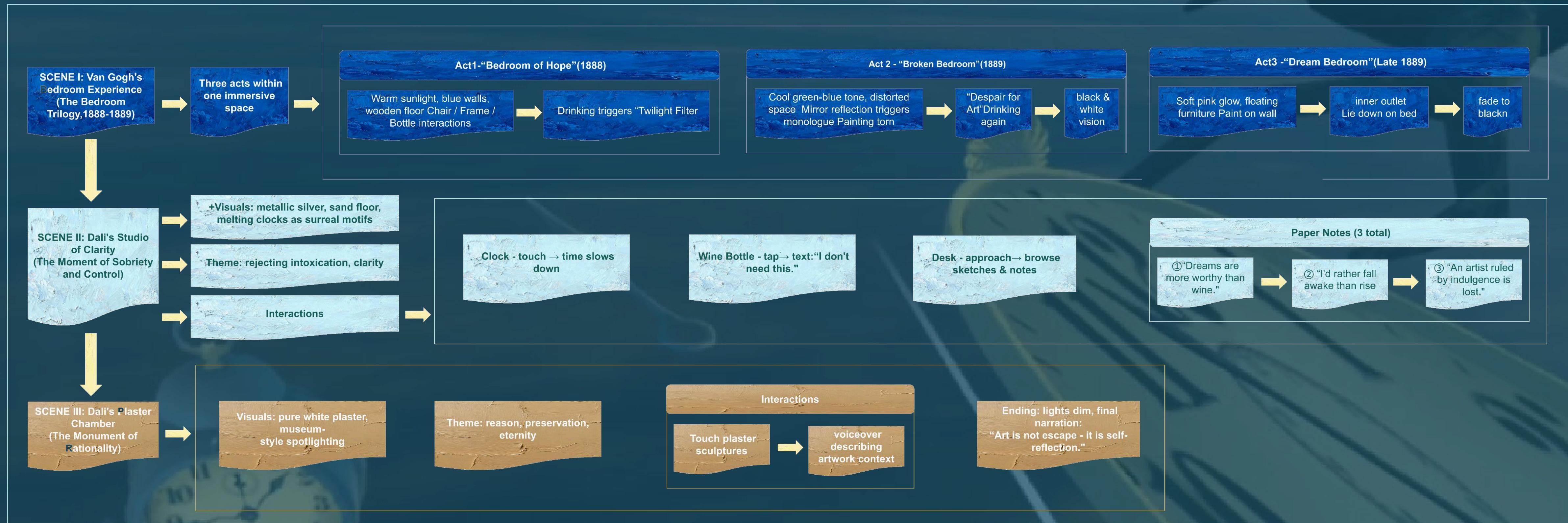


Game Flow

Dalí's Plaster Columns



Game Flow



Dali's Scene



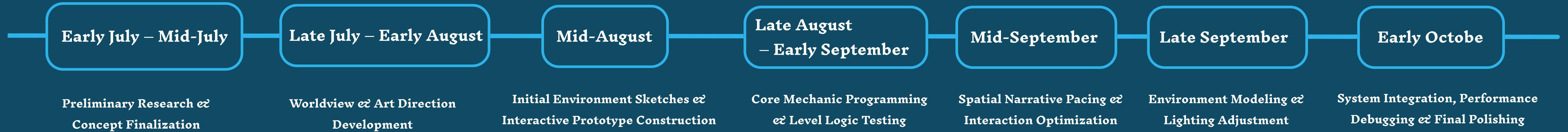
Game Flow

Van Gogh's Scene



HOW WE WORK TOGETHER

Timeline



Programmer / Level Designer



Yuhan Zhong

As the programmer and level designer of the team, I was primarily responsible for building the entire VR interaction system and level flow logic using Unity. I translated design concepts into immersive spatial experiences, ensuring players could interact in a natural and intuitive manner within the virtual environment. Throughout the project development process, I implemented core interaction mechanisms, spatial trigger logic, and performance optimization, while continuously testing the smoothness of interactions and player comfort. In terms of level design, I created an immersive atmosphere with narrative tension through precise control of spatial layout, pacing, and event triggers—allowing players to gradually perceive changes in emotion and story as they explore. Additionally, I focused on technical integration for team collaboration, efficiently linking program structures with art assets. This ensured the project was successfully completed within a short timeframe and achieved strong immersive expressiveness.

Artist / Researcher / Environment Designer



Zixian Yang

As the project's artist, environment designer, and lead researcher for the preliminary phase, Zixian took on extensive work in data collection and project direction-setting during the early stages. She conducted in-depth research on psychology cases, spatial narrative techniques, and VR visual presentation methods related to the project's theme, laying a solid theoretical foundation for the game's overall style and emotional tone. In the art production phase, Zixian was responsible for defining the project's visual style and spatial atmosphere, enhancing players' immersion through color, lighting, and composition. She designed and built core environments, ensuring each space was not only artistically expressive but also aligned with the interaction logic. She collaborated closely with programmers to continuously optimize art assets and performance, making the environments both realistic and expressive in VR. By integrating research findings with artistic creation, Zixian achieved a high degree of unity in the project's visual presentation and psychological narrative.