

Game Design / Technical Artist

Project\_1: Memory Odyssey P02-10

# PORTFOLIO

Project\_2: Nightmare P11-19

Project\_3: DayDream P20 - 25

Jian Junren (Devil\_ovo)

Personal Website: <https://devilovo.art/gamework>

Let it fill your soul — *Dead Poets Society*



# Memory Odyssey

Tag:

Puzzle

Muti-artstyles

Emotion

## Brief Introduction

An adventure to immerse in your lost memory and complete the information of your life.

## Video(Quick Look around 3mins)

<https://youtu.be/yBFNp4kTemA>

## Video(Whole Process around 20mins)

<https://youtu.be/fRinl960KGs?si=jXg6jdJvC1G9z3Ni>







# Memory Odyssey

**Game Producer :** Jian Junren (Devil\_ovo)

**Art & Code :** Jian Junren (Devil\_ovo)

**Render Pipeline :** Built-in



## Overview

### • Introduction

Memory Odyssey is an adventure with different games and a focus on solving puzzle and exploration for the answer of who you are.

### • How to play

Player discovers more informations and clues through playing different games of a playstation in the sickroom.

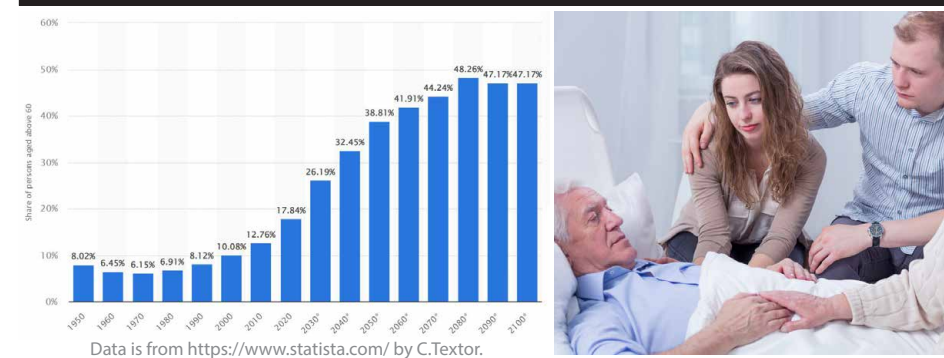
### • Story Outline

Actually you are an old man suffering from senile dementia and today it is coming on you. You only see a game machine in front of you. Through playing different game cassettes, you go through your memory.

Game Cassettes are actually made up of three parts, which indicates the old, middle age and childhood respectively. Player has to figure out the puzzles to explore more areas and get clues.

Everytime player finds the clue it would be marked in your room. At the end of the game, you finally realize it is not a game but the memory palace of your own. Then you remind of who you are and leave the "room".

## Society Background



### • Aging problem in China

With more aging population in China, by 2035, an estimated 400 million people in China will be aged 60 and over, representing 30% of the population.

### • Lack of Terminal care

Hospice becomes a realistic and significant problem. Because of high pressure from work, young generation have no time to spend with their olds, which introduces the game character - a lonely old man staying in a nursing home.

## Reference & Inspiration

### • Art Style

dark and low saturation scene, which is suitable to represent death.



### • Art Style

Full of npcs town where player could receive mission to push forward the progress of the story



### • Story Structure

Story plots

Story process

Game mechanic(To the moon)





# Game Mechanics

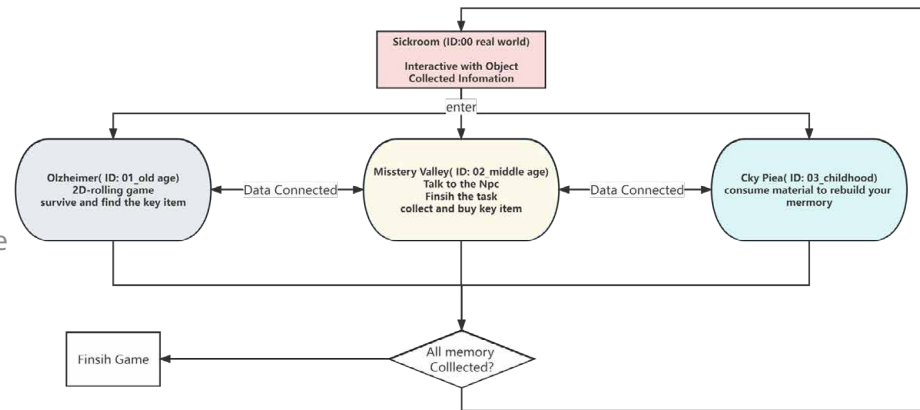
## Levels Outline & Play Loop

Scene00(Sickroom): Indicate the real world. Switch camera position to interact with different objects. It is also the only way to go to other scenes in game station.

Scene01(Olzheimer): Indicate the memory in old. It is a traditional 2D-rolling game.

Scene02(Misstery Village): Indicate the memory of middle age. Finish tasks of npcs, remind who they are and earn money.

Scene03(Cky Piea): Consume materials bought in scene02, build the memory of the childhood.



## Critical Objects Design(plan & fact)

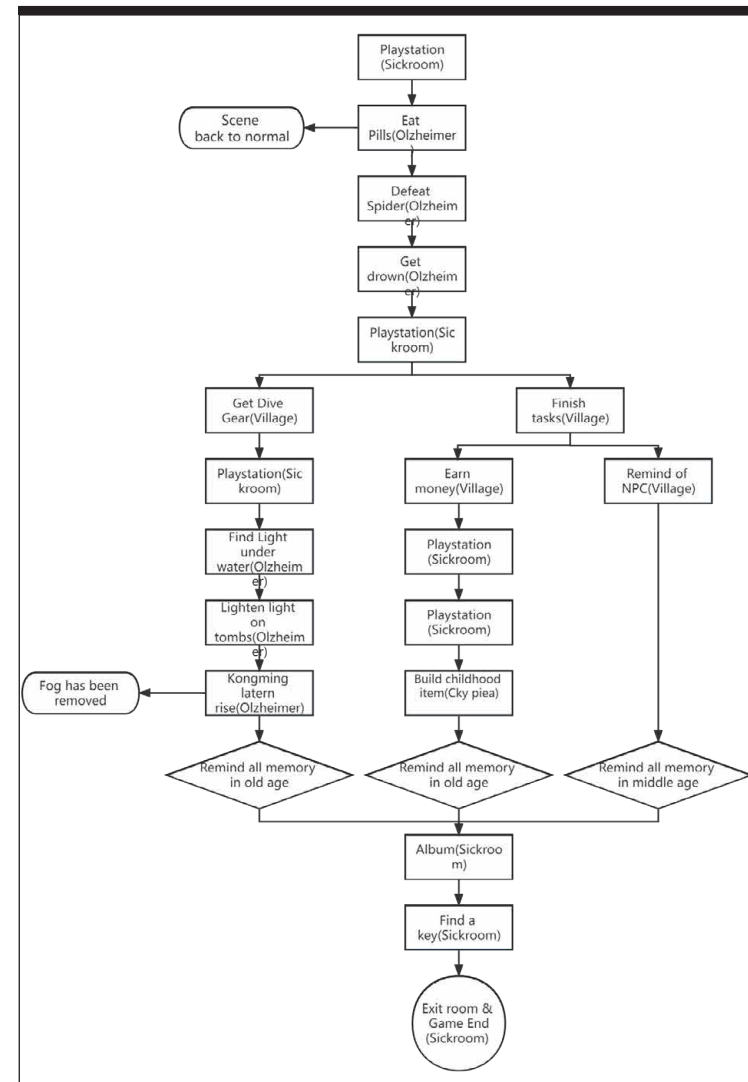
Critical Objects (Initial Plan)

Name	SceneID	Type	Attribute	How to use	Story behind
pill	00	Critical	scene appearance change	/	to cure heart disease
lamp	01	Critical	unlock new area	cross foggy area in scene02	symbol of light, remove the dark
wheelchair	02	Critical	unlock new area	can run in scene01	always sit after a stroke
glasses	02	Critical	scene appearance change	/	glasses help to read
breath mask	02	Critical	unlock new area	can dive in scene01	on a surgery
photo	02	Critical	unlock new area	/	photo of the love
flower seed	02	Collection	construct	consume in scene03	favorite flower
tree seed	01	Collection	construct	consume in scene04	tree in hometown
rope	02	Collection	construct	consume in scene05	build some toys
branch	02	Collection	construct	consume in scene06	the habit of picking branches as a child
ballon	02	Collection	construct	consume in scene07	really love it when he was a child
paper ring	03	Critical	finish game	get tips in scene01	an item to play house
key	01	Critical	finish game	/	key of the door in the real world

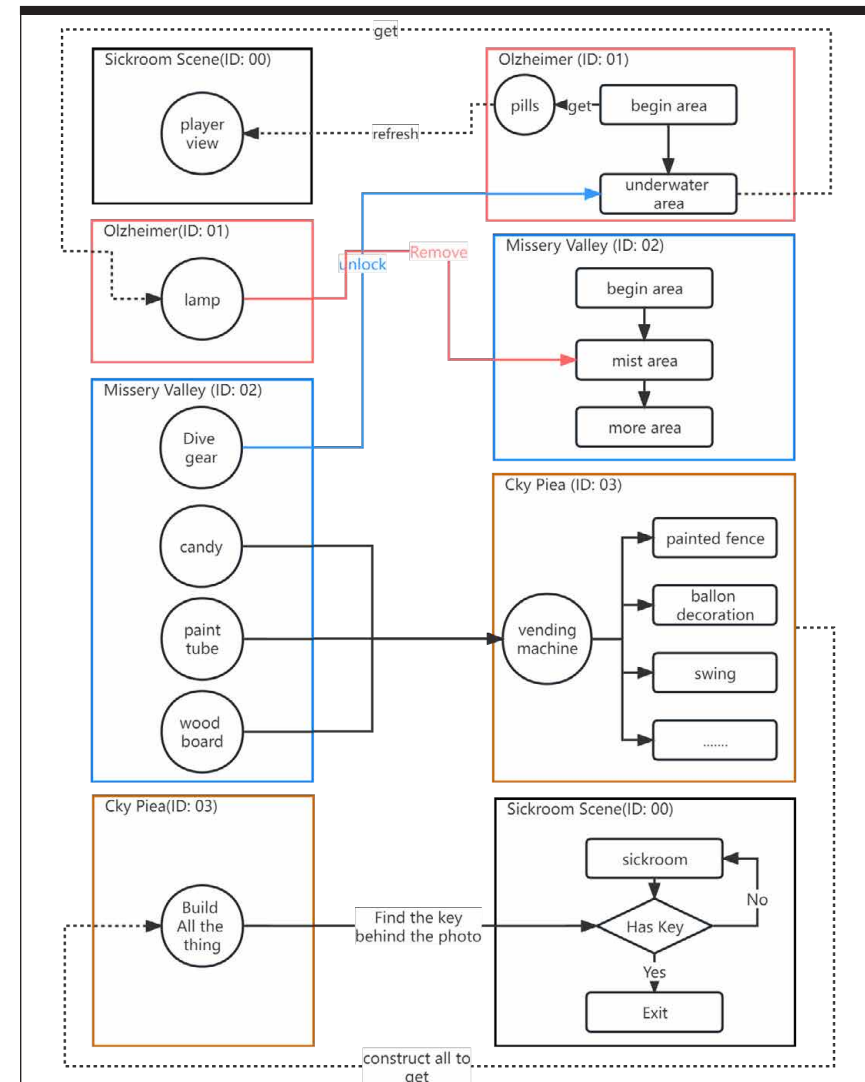
Critical Objects (Fact)

Name	SceneID	Type	Attribute	How to use	Story behind
pill	00	Critical	scene appearance change	/	to cure heart disease
lamp	01	Critical	unlock new area	cross foggy area in scene02	symbol of light, remove the dark
wheelchair	02	Critical	unlock new area	can run in scene01	always sit after a stroke
glasses	02	Critical	scene appearance change	/	glasses help to read
breath mask	02	Critical	unlock new area	can dive in scene01	on a surgery
photo	02	Critical	unlock new area	/	photo of the love
flower seed	02	Collection	construct	consume in scene03	favorite flower
tree seed	01	Collection	construct	consume in scene03	tree in hometown
Candy	02	Collection	construct	consume in scene03	build some toys from other kids
Wooden board	02	Collection	construct	consume in scene03	abandoned woodboard
PaintTub	02	Collection	construct	consume in scene03	really love it when he was a child
paper ring	03	Critical	finish game	get tips in scene01	an item to play house
key	01	Critical	finish game	/	key of the door in the real world

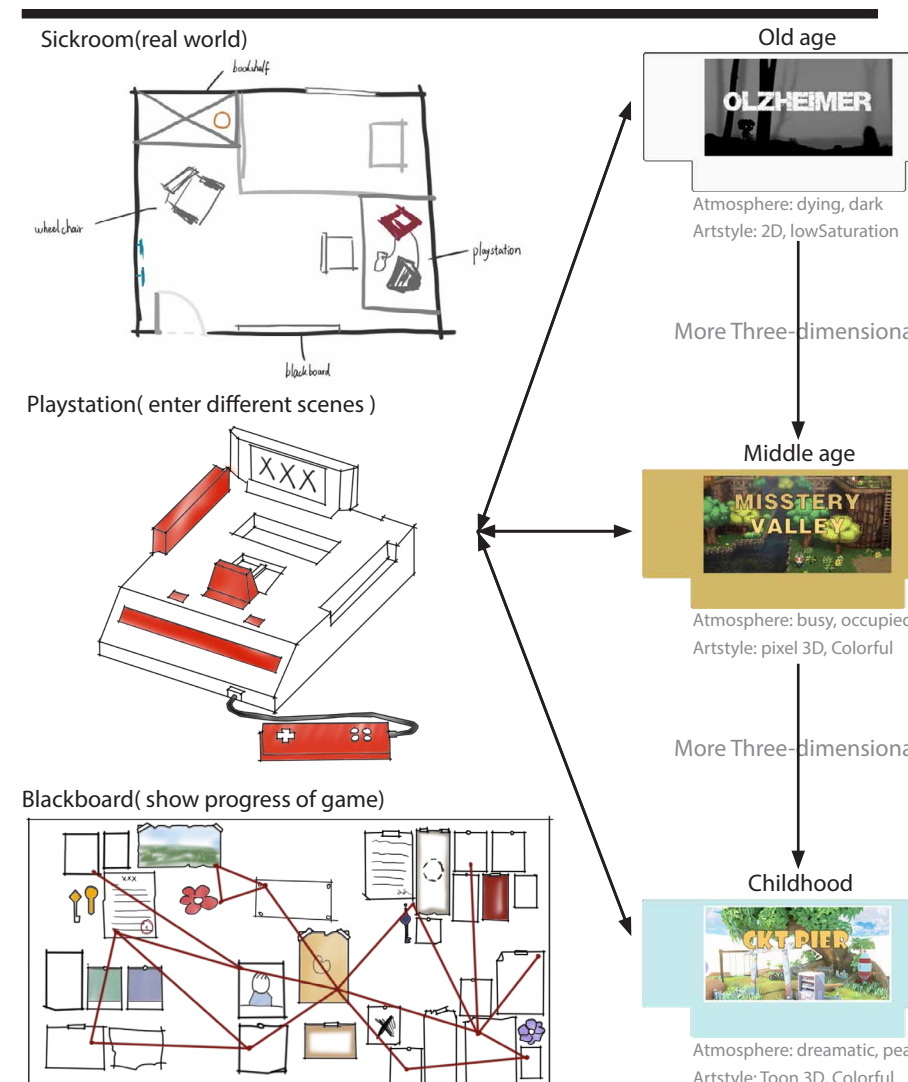
## Levels Progress



## Critical Items Logic



## Levels arrangement & sketch





# Level Design

## Sickroom (real world)

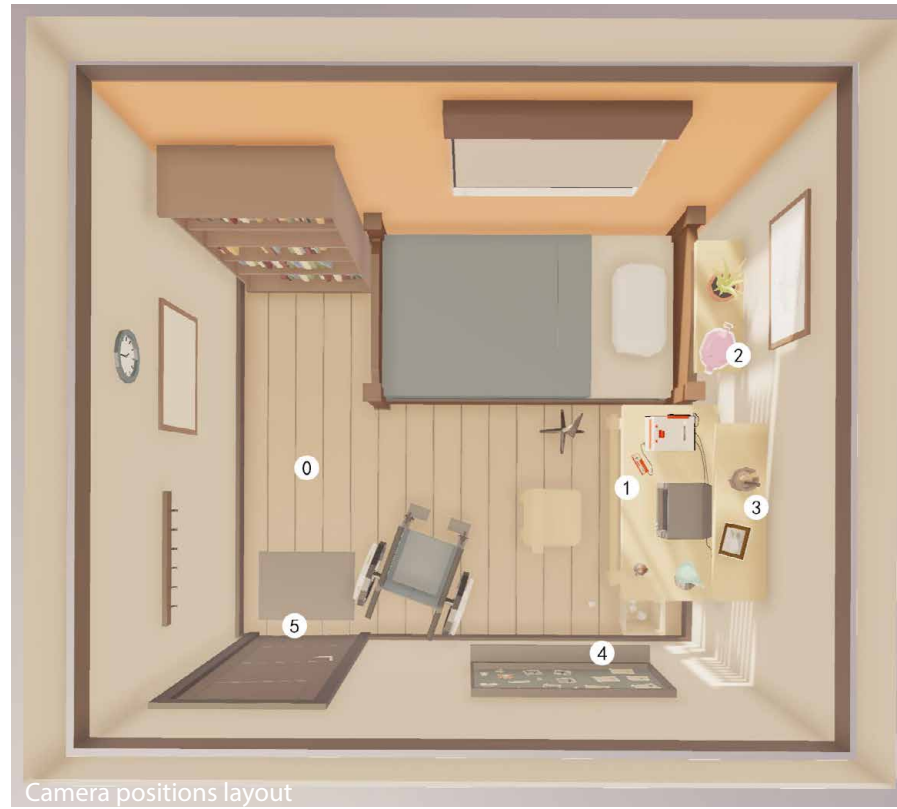
### • Overview

Playing method:

"W" & "S" to switch the position of view.  
Interactive with objects by mouse clicking.

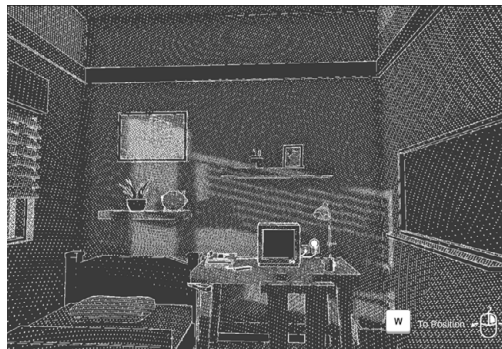


Camera Position (In game)

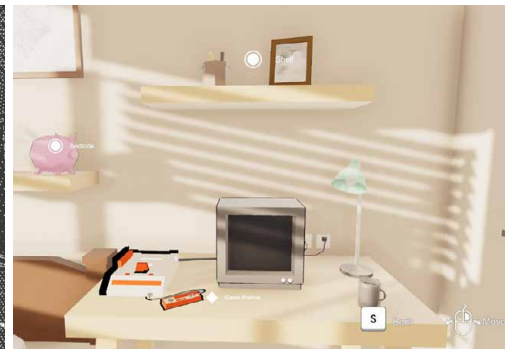


Camera positions layout

### • Game Event Details



At first, the player's view is dark. After eating the **pills**, it will return to normal.



Player could interact with highlight objects to get more informations.



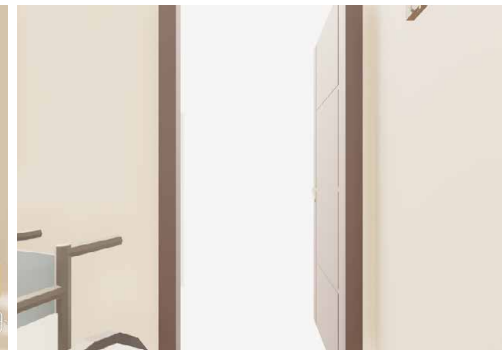
Player could go to different memory region through play station on the table.



Everytime player get a new information, it would be shown on the blackboard.



When player collect all memory piece, player will find a **key** behind album.

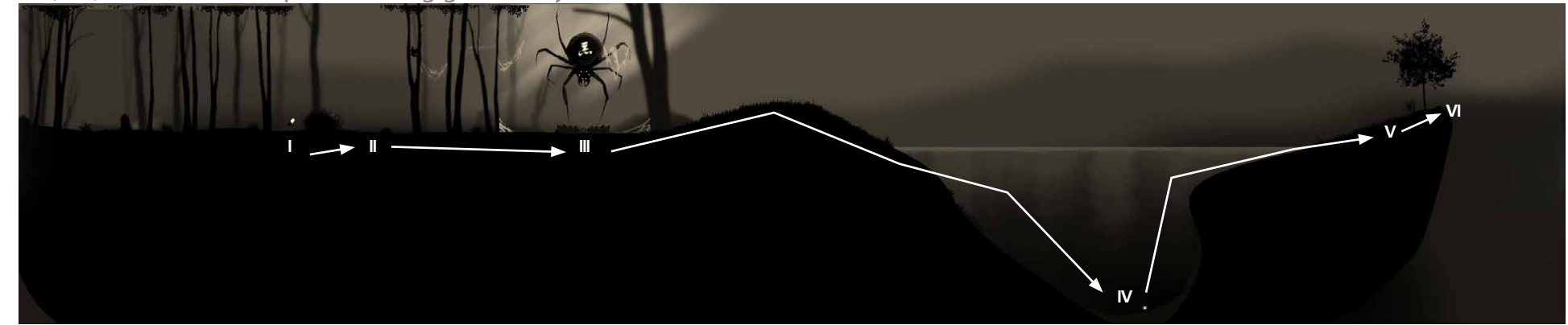


Finally, player opens the door and go to the heaven. The game is finished.

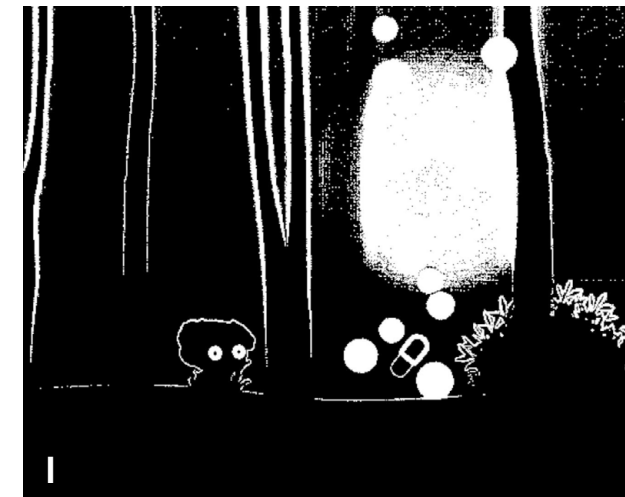
## Olzheimer(old age)

### • Overview

Playing method: a simple 2d rolling game. Play with "WASD".



### • Game Event



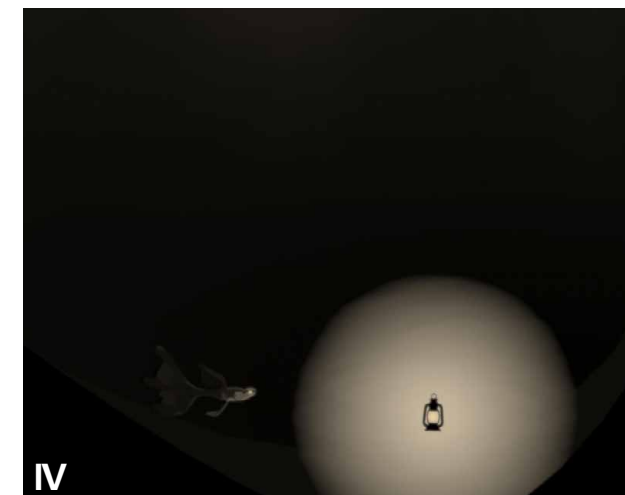
Firstly, player has to eat **pills** to recover from the dying state, which also affects the scene of sickroom in the real world.



A crow will pass by player, which indicates the death of god.



A giant spider who loves to eat memory of people will block the way. After breaking its two legs, it will flee away.



(After getting the **dive gear** in the Mystery Village) Player could become fish and dive into water and get the **lamp**.



At the end of the game, player encounters a grave, and needs to put down the lamp to see the name on it.



After that, player reaches the end of his life. And kongming lanterns are lightened.







# Level Design

## Misstery Valley(middle age)

### Overview

Playing method: "WASD" Move, Trade, Talk to npc, Finish tasks

**Coin**  
 The coins earned in this level can be used to buy materials which construct the memory in the next scene. Also player can check it in the pig pocket.

**Player**  
 Lots of missions are needed to finish to recover the memory, which indicates the busy and simple middle age.



Orthographic view of the whole scene

### Special Location on scene

- |  |   |   |                       |
|--|---|---|-----------------------|
| <b>1</b> Born point<br>The beginning of the game.      | <b>2</b> Outdoors Market<br>A place for player to trade.                  | <b>3</b> Bridge opposite<br>A mystery girl is waiting for player. Player can get a dive gear. | ← Main way for player |
| <b>4</b> other scene(forest)<br>A path to other scene. | <b>5</b> River side<br>A boy wants to go the opposite to catch butterfly. | <b>6</b> Fog area(locked)<br>Only after player lighten the lamp, the fog would fade.          | ← Way to other scene  |



### Game Event Details



Player find a weird **dive gear** at the corner of the house. After picking it up, player can dive in the old age.

At the bridge, player will encounter a **shadow(broken memory)**. After finishing the task ,it would recover.

Player will meet a young boy right by the river. He will tell player how to repair the bridge to go to **new area**.

Merchant is selling items such as **candy, painttubes and woodboard** which can be used in **Cky Pica**.

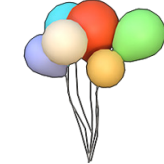







## Cky piea(childhood)



### Overview

Playing method: Comsume items from middle age to build facilities through UI

### Build Object Details

- |  |   |
|--|---|
| <b>1 Ballons</b><br><br>Lots of colorful ballons, which are children most favourite things.<br>2 1 1 1              | <b>2 Rocket</b><br><br>A giant rocker, maybe could take me to the space!<br>3 1 2 2                      |
| <b>3 Building Block</b><br><br>A castle made up of colorful building blocks<br>0 1 2 2                              | <b>4 Work bench</b><br><br>A bench for drawing and painting, child loves it well.<br>2 0 1 1             |
| <b>5 Bicycle</b><br><br>A small red dirty bicycle. It seems to have gone through many situations.<br>1 0 2 2        | <b>6 Swing</b><br><br>A swing for two children. Always used in the leisure time.<br>1 0 3 3              |
| <b>7 Colorful fence</b><br><br>Colorful painting on the fence, including the height scales carved on it.<br>2 1 1 1 | <b>8 Grass &amp; Flowers</b><br><br>Variety of plants. It must be comfortable to sleep on it!<br>2 1 0 0 |

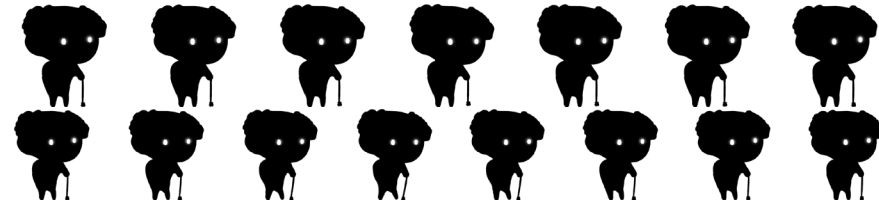




# Character Design

## Character design(Player)

### • Old age(Sequence Animation)



### • Middle age(Sequence Animation)



### • PlayerInput

```
private void PlayerInput()
{
    InputX = - Input.GetAxisRaw("Horizontal");
    InputY = Input.GetAxisRaw("Vertical");
    InputY *= 1.2f;
    movementInput = new Vector3(InputX, 0, InputY) * moveSpeed;
    isMoving = movementInput != Vector3.zero;
    if(Input.GetKeyDown(KeyCode.Escape))
    {
        EventHandler.CallSettingPanel(true);
    }
}
```

### • PlayerMovement

```
private void Movement()
{
    float horizontalMove;
    horizontalMove = Input.GetAxis("Horizontal");
    float faceDirection = Input.GetAxisRaw("Horizontal");

    if(!isFish)
    {
        verticalMove = 0;
        moveSpeed = baseSpeed;
    }
    else
    {
        verticalMove = 0;
        moveSpeed = 2 * baseSpeed;
    }
    rb.velocity = new Vector3(horizontalMove * moveSpeed * Time.fixedDeltaTime, verticalMove * speed * Time.deltaTime);

    if (horizontalMove != 0)
    {
        anim.SetBool("isWalking", true);
        anim.SetFloat("isSwimming", true);
    }
    else
    {
        anim.SetBool("isWalking", false);
        anim.SetFloat("isSwimming", false);
    }

    if (faceDirection != 0)
    {
        transform.localScale = new Vector3(faceDirection, 1, 1);
    }
}
```

## Character design(NPCs)

### • Character background design



Hidden ID: nephew  
A young boy who loves catching insects especially butterfly.



Hidden ID: wife  
A mysterious girl who you always find familiar with. But you forget who she is.



Hidden ID: colleague  
An honest merchant sells different kinds of merchandise.



Hidden ID: brother  
A brave man who likes to adventure.

### • Dialogue design

Name	DialoguePack_Index	Dialogue_content	Name	DialoguePack_Index	Dialogue_content
Nina	1-1	Excuse me?	Ashy	3-2	Oh, what happen? Ah... hehehe. Thank you for the work. By the way, there are something new merchants are selling. I think you should take a look.
Sunny	2-1	Eldert brother, let's catch the butterfly. I heard that there are lots of butterflies across the river. But for the thick fog, it is inaccessible now.	Cloe	4-1	.....
	2-2	I wonder when we could dispel that fog. Maybe a light? Huh? You lighten the light? You did it! how we just need some woodboards to repair the bridge. Oh! Now we are able to go to butterfly valley to catch them!	Merchant Hoang	6-1	Young man... is waiting for you across the bridge, are you? Do you want to sell or buy?
Ashy	3-1	Wow, I have been waiting for you for a long time. How have you been? It is surprising going well! Do you remember the task I gave you? There's the most I want! Oh! Thank you. Here's your reward, some wood boards. You may use it to build something.	Merchant Wang	7-1	Today is a good day! have you seen Ashy? She is waiting for you. Do you want to buy or sell something? Oh, see you.
			Old John	8-1	Fishing is beautiful for every thing. You're free, do you want to fish?

### • How to realize

#### View of Inspector

#### Data Storage

```
[System.Serializable]
public class DialoguePiece
{
    [TextArea]
    public string dialogueText;
    public bool hasToPause;
    [HideInInspector] public bool isDone;
    public UnityEvent afterTalkEvent;
}
```

#### DialogueFunction

```
private IEnumerator DialogueRoutine()
{
    isTalking = true;
    if(dialogueStack.TryPop(out DialoguePiece result))
    {
        // Call UI Show Dialogue
        EventHandler.CallShowDialogueEvent(result, npc.face, npc.npcName);
        EventHandler.CallShowEvent(result, npc.face, npc.npcName);
        npc.isTalking = true;
        yield return new WaitForSeconds(1) == result.isDone;
        if(result.afterTalkEvent != null)
        {
            result.afterTalkEvent.Invoke();
        }
        isTalking = false;
    }
    else
    {
        npc.isTalking = false;
        EventHandler.CallGameStop(false);
        EventHandler.CallShowDialogueEvent(null, null, null);
        if(OnFinishEvent != null)
        {
            OnFinishEvent.Invoke();
        }
        canTalk = false;
        FillDialogueStack();
        isTalking = false;
    }
}
```

#### Main Function

```
private void Update()
{
    uiSign.SetActive(canTalk);
    if(canTalk & Input.GetKeyDown(KeyCode.Space) && !isTalking)
    {
        StartCoroutine(DialogueRoutine());
    }
}
```

#### FillDialogueStack

```
/// Summary: Build Dialogue Stack
/// 2个引用
private void FillDialogueStack()
{
    dialogueStack = new Stack<DialoguePiece>();
    for (int i = 0; i <= dialogueList.Count - 1; i++)
    {
        dialogueList[i].isDone = false;
        dialogueStack.Push(dialogueList[i]);
    }
}
```

### • Daily routine design

#### Navigation mesh

#### Routine design on plan

#### Routine sheet

Name	Duration/Each State	MovementPos	StateAtStop	IsInteractable
Nina	15	1:1	Stand Standing	Yes
Nina	25	1:2	Stand Standing	Yes
Sunny	15	2:1	LeftLight Standing	Yes
Sunny	15	2:2	LeftLight Standing	Yes
Ashy	15	2:3	LeftLight Standing	Yes
Ashy	15	2:4	LeftLight Standing	Yes
Cloe	20	4:1	Stand Standing	Yes
Cloe	20	4:2	LeftLight Standing	Yes
Ella	20	5:1	Stand Standing	Yes
Ella	20	5:2	Forward Backward Standing	Yes
Merchant Hoang	20	6:1	Forward Backward Standing	Yes
Merchant Wang	15	6:2	LeftLight Standing	Yes
Merchant Wang	15	6:3	LeftLight Standing	Yes
Old John	20	8:1	Fishing	Yes
Old John	20	8:2	Forward Backward Standing	Yes
Old John	20	8:3	Forward Backward Standing	Yes

### • How to realize

#### Data Storage(In ScriptableObject)

```
[System.Serializable]
public class ScheduleDetails
{
    public int durationTime;
    public Vector3 targetPos;
    public E_NPCState stateAtStop;
    public bool interactable;
}
```

#### Main Function

```
private void UpdateNPCNextDestination()
{
    if(time >= nowNPCData.durationTime)
    {
        scheduleDataIndex++;
        if(scheduleDataIndex > data.scheduleList.Count - 1)
            scheduleDataIndex = 0;
        UpdateNPCData(scheduleDataIndex);
        time = 0;
    }
}
```

#### UpdateData(for moving)

```
private void UpdateNPCData(int index)
{
    isMoving = true;
    interactable = false;
    nowNPCData = data.scheduleList[index];
    targetPos = nowNPCData.targetPos;
    NavMeshPath path = new NavMeshPath();
    nav.ResetPath();
    nav.CalculatePath(targetPos, path);
    nav.SetPath(path);
    anim.SetBool("isFishing", false);
    anim.SetBool("isMoving", true);
}
```

#### Animation Control Function

```
private void SwitchAnimDirection()
{
    anim.SetFloat("DirX", - Vector3.Normalize(nav.steeringTarget - this.transform.position).x);
    anim.SetFloat("DirY", - Vector3.Normalize(nav.steeringTarget - this.transform.position).z);
}
```

#### Stop What to do Function

```
private IEnumerator Wandering(Vector3 v)
{
    Vector3 target = originalPos + v;
    yield return new WaitForSeconds(timeSpan);
    target = originalPos - v;
    UpdateNPCData(target);
    yield return new WaitForSeconds(timeSpan);
    isPlayingAnim = false;
}
```

#### ArriveDestination Function

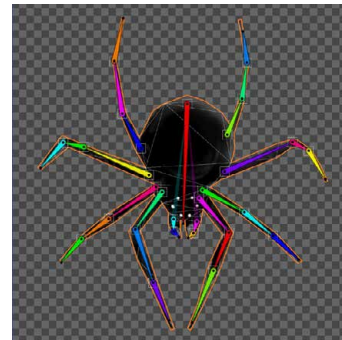
```
private void ArriveDestination()
{
    if(Vector3.Distance(this.transform.position, targetPos) <= stopDistance)
    {
        isMoving = false;
        interactable = nowNPCData.interactable;
        anim.SetBool("isMoving", false);
        nav.IsStopped = true;
        if(isTalking)
            return;
        time += Time.deltaTime;
    }
    if(!isPlayingAnim || nowNPCData == null || !isMoving)
        return;
    NPCBehavior(nowNPCData.stateAtStop);
}
```

## Enemy design(Spider)

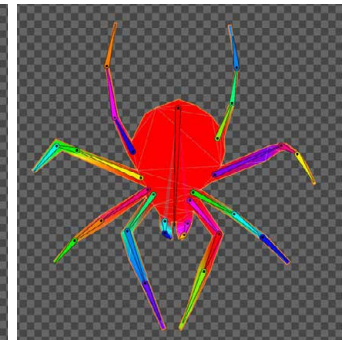
### • Spider



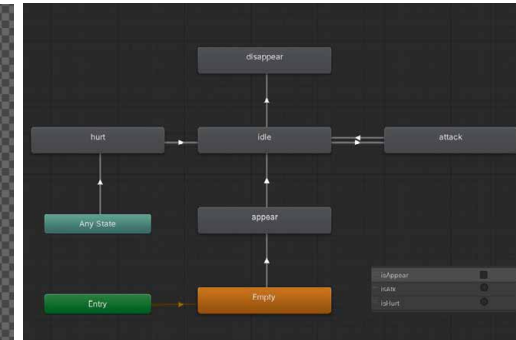
### • Spider (Rig & IK)



### • Spider (Weight of Rig)



### • Animator Logic



### • Script Spider(Tell legs to attack)

```
private void CalculateIsAppear()
{
    if (hp <= 0)
        return;
    distance = Vector3.Distance(playerCtrl.position, this.transform.position);
    if (distance <= appearRange)
    {
        isAppear = true;
        anim.SetBool("isAppear", true);
        if (distance <= attackRange)
        {
            if (!isAttacking)
            {
                isAttacking = true;
                anim.SetTrigger("isAtk");
                LogBeginToAtk();
            }
        }
    }
    else
    {
        isAppear = false;
        anim.SetBool("isAppear", false);
    }
}
```

### • Script Leg(attack core codes)

```
private void Start()
{
    playerTransform = GameObject.FindWithTag("Player").transform;
    targetPos = Ik.position;
}

private void OnTriggerEnter2D(Collider2D other)
{
    if (other.gameObject.CompareTag("Player"))
    {
        EventHandler.CallTransitionEvent("scene01", Vector3.zero);
    }
    else if (other.gameObject.CompareTag("Trap"))
    {
        if (isAtk)
        {
            if (isAlive)
            {
                spider.GetHurt();
                isAlive = false;
                trap.trap = other.gameObject.GetComponent<Trap>();
                trap.OnTrap();
            }
        }
    }
}

private IEnumerator AtkPlayer()
{
    isAtk = true;
    startPos = Ik.position;
    targetPos = Ik.position + Vector3.up * 1;

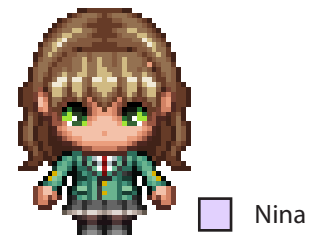
    // Before Atk Stretch its leg
    playerLastPos = playerTransform.position;
    Ik.transform.DOMove(targetPos, 3);
    yield return new WaitForSeconds(2.5f);

    // Attack Player
    Ik.transform.DOMove(playerLastPos, 2);
    // Spider Sound Effect
    EventHandler.CallPlayerSoundEvent(E_SoundName.SpiderAttack);

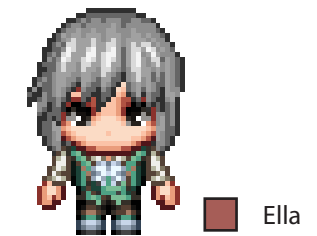
    yield return new WaitForSeconds(2f);

    // Back to original place
    Ik.transform.DOMove(startPos, 3);
    yield return new WaitForSeconds(2.5f);

    isAtk = false;
}
```



Hidden ID: sister  
A cool girl who seldom talks to other people.



Hidden ID: aunt  
A old woman who always cherishes hope towards life.



Hidden ID: friend  
An honest merchant sells merchandise and also provide important infos.



Hidden ID: uncle  
An old man who is extremely addicted to fishing.



# Visual Effect Design

## One-bit artstyle

need one C# script and two shaders

### Effect Comparison



### Capture RT(C#)

```

// Capture RT C#
using UnityEngine;
using UnityEngine.Rendering;

public class CaptureRT : MonoBehaviour
{
    // ... other codes ...
}
    
```

### Dither shader critical codes

```

// Dither Shader Critical Codes
float rand01(float x) {
    return frac(sin(x * 12.9898 * 3.1415926535997964) * 43758.5453121);
}

// ... other codes ...
    
```

### Threshold shader critical codes

```

// Threshold Shader Critical Codes
float4 _MainTex;
float4 _Color;

// ... other codes ...
    
```

## Gaussian-blur

need one C# script and one shader

### Effect Comparison



### Blur shader critical codes

```

// Blur Shader Critical Codes
// Vertex Part
v2f vert(appdata v)
{
    // ... other codes ...
}

// Fragment Part
fixed4 frag(v2f i) : SV_Target
{
    // ... other codes ...
}
    
```

### Blur C# critical codes

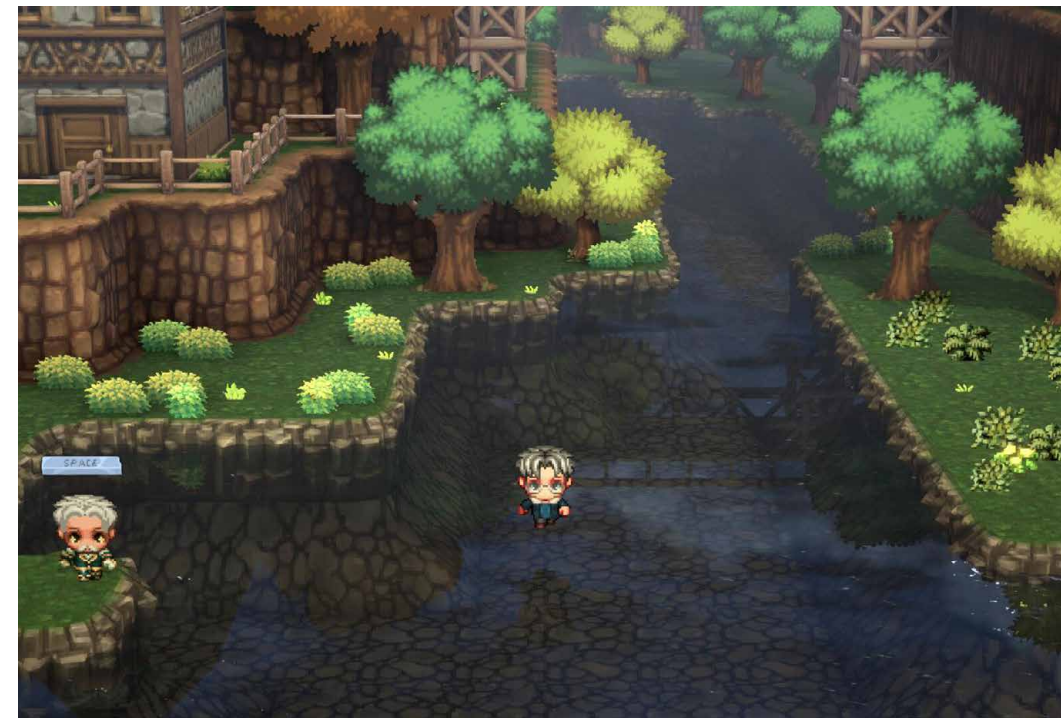
```

// Blur C# Critical Codes
// Region Blur
private void OnRenderImage(RenderTexture src, RenderTexture dest)
{
    // ... other codes ...
}
    
```

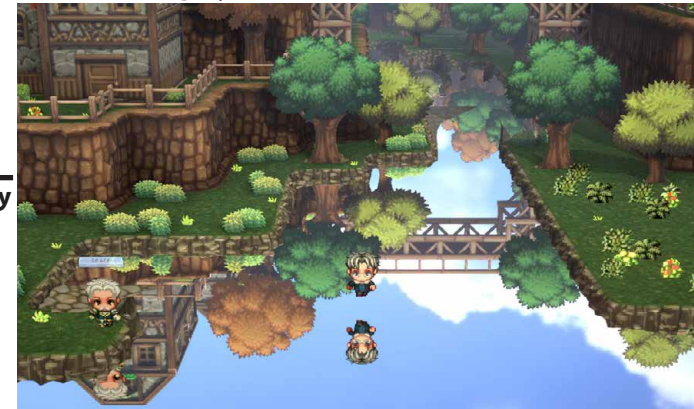
## Real time reflection water(Planar Reflection)

need one C# script and one shader

### Complete Effect Display



### Reflection Display



### Transform Matrix(C#)

```

// Transform Matrix C#
Matrix4x4 CalculateReflectMatrix(Vector3 normal, Vector3 positionOnPlane)
{
    // ... other codes ...
}
    
```

### Vertex Part

```

// Vertex Part
v2f vert (appdata_full v)
{
    // ... other codes ...
}
    
```

### Fragment Part

```

// Fragment Part
fixed4 frag (v2f i) : SV_Target
{
    // ... other codes ...
}
    
```

### Normal Water Display



### Vertex Part

```

// Vertex Part
v2f vert (appdata_full v)
{
    // ... other codes ...
}
    
```

### Fragment Part

```

// Fragment Part
fixed4 frag (v2f i) : SV_Target
{
    // ... other codes ...
}
    
```

Multiply

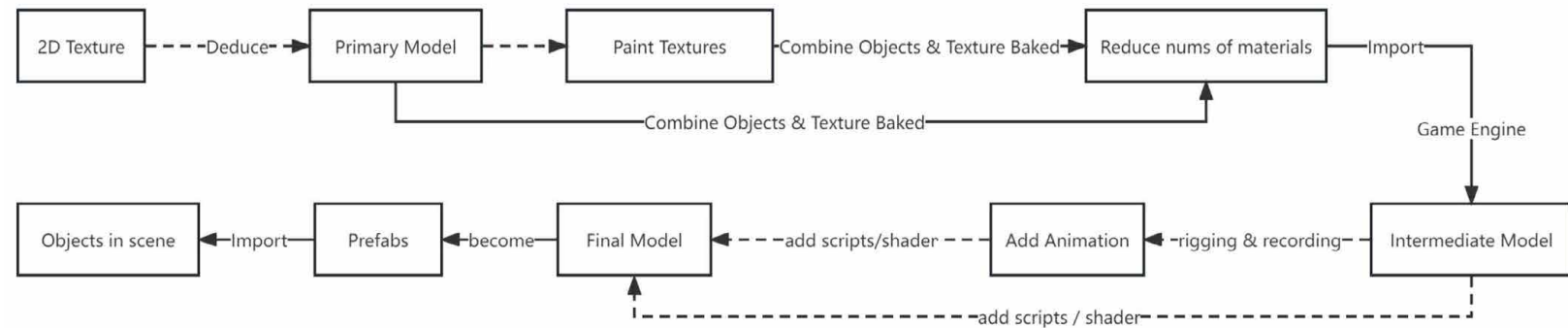
Multiply



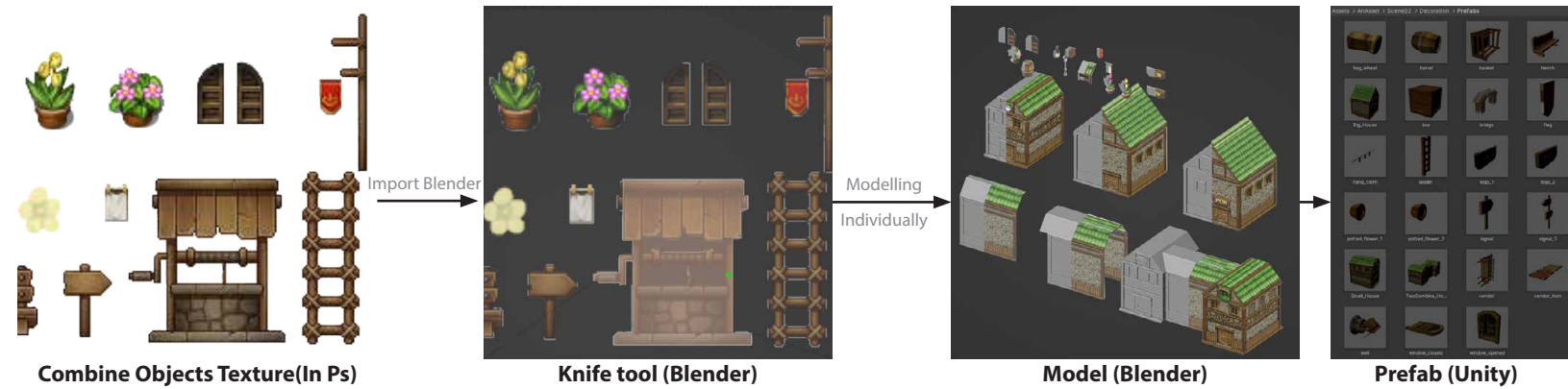
# Workflow Design

## Art Assets WorkFlow

### Overview



### Example 1: 3D Pixel Model(One Texture One Material for different objects)



### Example 2: Floating Ballons(vertex color and vertex animation)

**Effect Display in game**

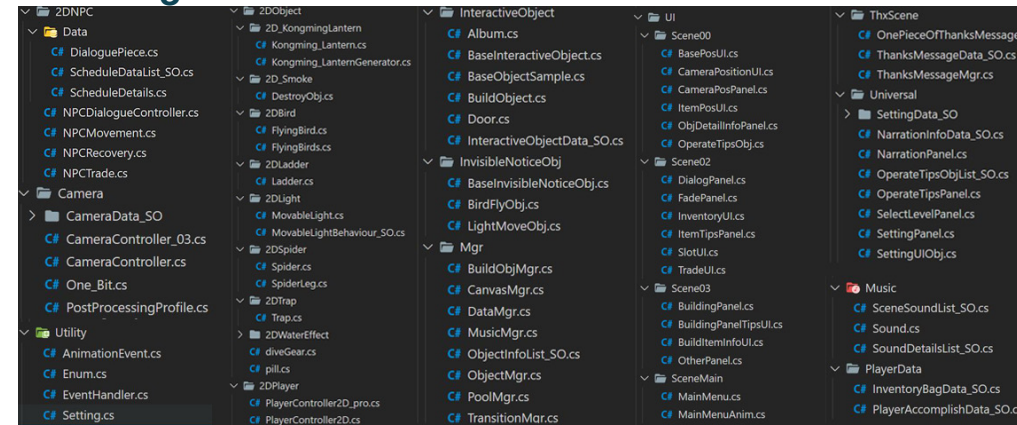
**Ballons Floating shaderLab**

**Vertex Color in Blender**

**Vertex Offset and Mask**

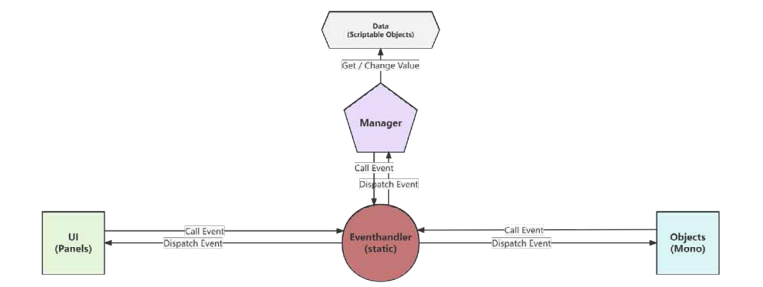
## Programming Logic

### Coding Files Overview

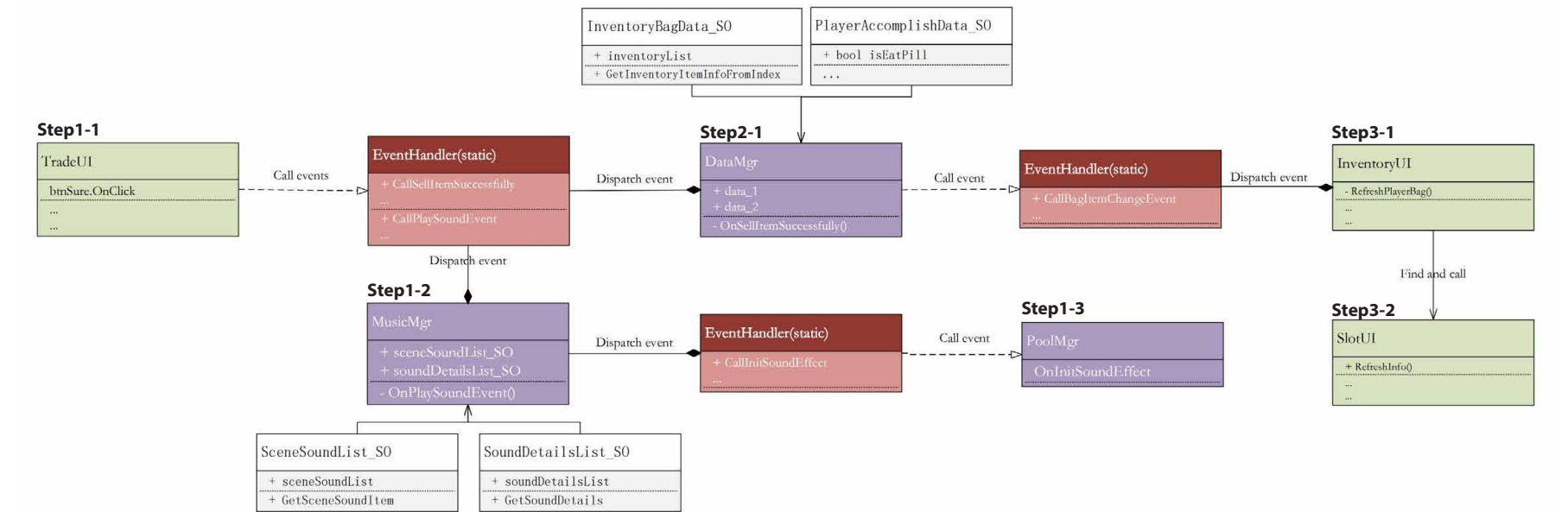


### Communication between Scripts

Basic elements: Mgrs, Datas(ScriptableObjects), Objects, EventHandler, UI(Panels)  
 Coding : Object-oriented programming



### Example 1: Buy item event



#### Step1-1: TradeUI (BtnListener + CallEvent)

```

    btnSure.onClick.AddListener(() =>
    {
        if(!canTransact || !isTrading)
            return;
        if(!isSell)
        {
            EventHandler.CallSellItemSuccessfully(info,info.Parse(inputField.text) * info.priceToSell, info.Parse(inputField.text));
            EventHandler.CallPlayerSoundEvent(E_SoundName.SellThing);
        }
        else
        {
            EventHandler.CallBuyItemSuccessfully(npcBag.inventoryList[0],info.Parse(inputField.text) * info.priceToBuy,info.Parse(inputField.text));
            EventHandler.CallPlayerSoundEvent(E_SoundName.BuyThing);
        }
        OnShowTradePanel(false,false,null);
        isTrading = false;
    });
  
```

#### Step1-2: MusicMgr (EventListener)

```

    private void OnPlayerSoundEvent(E_SoundName soundName)
    {
        var soundDetails = soundDetailsData.GetSoundDetails(soundName);
        if(soundDetails != null)
            EventHandler.CallInitSoundEffect(soundDetails);
    }
  
```

#### Step1-3: PoolMgr (EventListener)

```

    private void OnInitSoundEffect(SoundDetails soundDetails)
    {
        var obj = GetPoolObject();
        obj.GetComponent<Sound>().SetSound(soundDetails);
        obj.SetActive(true);
        StartCoroutine(DisableSound(obj, soundDetails.soundClip.length));
    }
  
```

#### Step2-1: DataMgr ( EventListener+CallEvent)

```

    private void OnBuyItemSuccessfully(InventoryItemInfo info, int money, int num)
    {
        for (int i = 0; i < bagData.inventoryList.Count; i++)
        {
            // same items merge
            if(bagData.inventoryList[i].index == info.index)
            {
                bagData.inventoryList[i].num += num;
                // change Data
                universalData.money -= money;
                // Refresh UI
                EventHandler.CallBagItemChangeEvent();
                return;
            }
            // nope find empty slot to storage
            if(bagData.inventoryList[i].index == -1)
            {
                bagData.inventoryList[i] = info;
                bagData.inventoryList[i].num = num;
                // change data
                universalData.money -= money;
                // Refresh UI
                EventHandler.CallBagItemChangeEvent();
                return;
            }
            // synchronization Scene02 and Scene03's data
            universalData.candy = bagData.GetInventoryItemInfoFromIndex(6).num;
            universalData.paintTube = bagData.GetInventoryItemInfoFromIndex(7).num;
            universalData.woodBoard = bagData.GetInventoryItemInfoFromIndex(8).num;
        }
    }
  
```

#### Step3-1: Inventory UI (EventListener)

```

    private void OnBagItemChangeEvent()
    {
        RefreshInventory();
    }
    public void RefreshInventory()
    {
        for (int i = 0; i < slotUIS.Count; i++)
        {
            slotUIS[i].Init(inventoryData.inventoryList[i]);
        }
    }
  
```

#### Step3-2: Slot UI

```

    public void Init(InventoryItemInfo itemInfo)
    {
        if(itemInfo == null || itemInfo.index == -1)
        {
            imgIcon.sprite = null;
            imgIcon.color = new Color(0, 0, 0, 0);
            txtNum.text = "";
        }
        else
        {
            info = itemInfo;
            imgIcon.sprite = itemInfo.icon;
            txtNum.text = itemInfo.num.ToString();
        }
    }
  
```



# Summary

## Plan VS Reality

### • Difference

As we can see from the differences between the two tables below, I spent more 5 weeks on Project\_1 development and the progress in fact did not seem as organized as my planning.

Most of the time especially in the second half of the development period, I worked on different aspect of development at the same time.

### • Reasons

#### School Work

Before July, I was still working on my graduation design at college. There were around 3-4 hours per day for me to develop my this project.

#### WorkLoad Misjudgement

As my first indie game(demo), I misjudged the workload I would take. So you may find there are lots of initial design I removed from my plan.(e.g. *page 2 Critical Objects Table*)

### • Schedule(15 weeks)

Task	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11	Week12	Week13	Week14	Week15
	06/03/2023	15/03/2023	22/03/2023	29/03/2023	05/04/2023	12/04/2023	19/04/2023	26/04/2023	03/05/2023	10/05/2023	17/05/2023	24/05/2023	31/05/2023	07/06/2023	14/06/2023
Game Design Document															
Technical Design Document															
Create Levels															
Create Character(Sprite)															
Create Objects															
Create Trigger & Events															
Create UI															
Shader Development															
Guide Development															
Testing															
Evaluation															
Create journal of design and implementation ideas															
Create Manual															

### • In Fact(20 weeks)

Task	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11	Week12	Week13	Week14	Week15	Week16	Week17	Week18	Week19	Week20	Week20
	06/03/2023	15/03/2023	22/03/2023	29/03/2023	05/04/2023	12/04/2023	19/04/2023	26/04/2023	03/05/2023	10/05/2023	17/05/2023	24/05/2023	31/05/2023	07/06/2023	14/06/2023	21/06/2023	28/06/2023	05/07/2023	12/07/2023	19/07/2023	26/07/2023
Game Design Document																					
Technical Design Document																					
Create Levels																					
Create Character(Sprite)																					
Create Objects																					
Create Trigger & Events																					
Create UI																					
Shader Development																					
Guide Development																					
Testing																					
Evaluation																					
Music & Sound Effect																					
Journal design & implemented ideas																					
Create Manual																					
Remark									(4.29-5.2) Luobe Dare 92		(5.10-5.18) Graduation design								(7.08-7.10) BMK 2023		

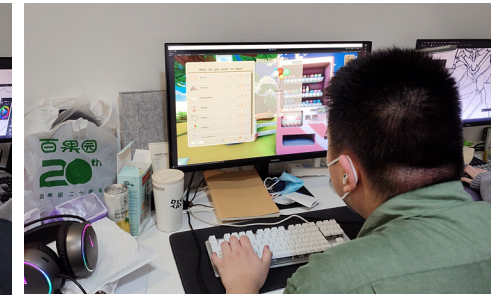
## Feedback from others



**Name:** Ma Sile  
**Occupation:** Game Designer  
**Feedback:**

Though the pace of game is a kind of slow-type, I still think it could be an interesting and excellent game after adding a complete and touching story.

Anyway, as a demo, it is acceptable.



**Name:** Chen Tianlang  
**Occupation:** Game Programmer  
**Feedback:**

I really love the different artstyles insides the game. However, I think guide in game are weak.

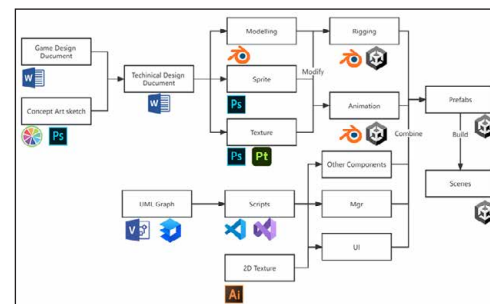
I also suggest that the difference between the playing modes of four scenes could distinguish from each other more.

## What I learn from this project

### • Workflow Improvement

I develop and improve my own workflow about art assets and codings.

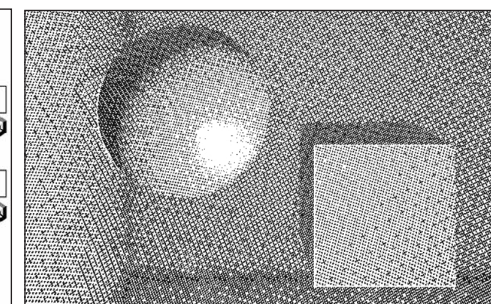
Also I realized the importance to build a coding structure in advance.



### • Shader Development

I have learnt how to use shader to create interesting visual effects through *OnRenderImage*.

Such as planar water reflection and one-bit art effect.



## Other Problems in projects

### • Incomplete Story

My whole point of my game is to narrate a story of an old man. But due to the limited time, I only made a small part of it. And this makes the game less attractive and fun.(e.g. *page 2 Critical Objects Table*)

### • Coding Management

I did not plan a basic structure of my codes, which causes extra repeated codes insides my project. Some could be avoided by inheritance or encapsulation.

### • Rendering error

Because I did not give myself deep insight into the built-in render pipeline, there are some rendering error in my project and I have no idea to deal with it.(e.g. shadow of 2D Sprite, 2D Sprite Postprocessing)



### • Performance optimization

In this project, I did not use the LOD system. In scene2(misstery village), because of the extra camera for *water planar reflection*, the number of batches and drawcalls just almost triple.

Statistics	Statistics	Statistics
Audio: Level: -74.8 db Clipping: 0.0% DSP load: 0.1% Stream load: 0.0%	Audio: Level: -74.8 db Clipping: 0.0% DSP load: 0.1% Stream load: 0.0%	SetPass Calls: 133 (Dynamic Batching) Batched Draw Calls: 2 (Static Batching) (Instancing) Batched Draw Calls: 2.5k
Graphics: 250.6 FPS (3.9ms) CPU: main 3.8ms, render thread 2.4ms Batches: 304 Saved by batching: 228 Tri: 151.9k Verts: 214.7k Screen: 1983x1116 - 25.3 MB SetPass calls: 133 Shadow casters: 1500 Visible skinned meshes: 0 Animation components playing: 0 Animator components playing: 12	Graphics: 147.1 FPS (6.8ms) CPU: main 8.8ms, render thread 5.3ms Batches: 907 Saved by batching: 620 Tri: 454.4k Verts: 544.9k Screen: 1983x1116 - 25.3 MB SetPass calls: 133 Shadow casters: 4502 Visible skinned meshes: 0 Animation components playing: 0 Animator components playing: 12	Triangles: 151.9k Vertices: 214.6k Time: 0.00ms
Render Textures: 54 / 447.1 MB	Render Textures: 54 / 447.1 MB	Used Textures: 52 / 18.7 MB
Off	Off	Off
SetPass Calls: 299 (Dynamic Batching) Batched Draw Calls: 6 (Static Batching) (Instancing) Batched Draw Calls: 7.1k	SetPass Calls: 893 (Dynamic Batching) Batched Draw Calls: 6 (Static Batching) (Instancing) Batched Draw Calls: 7.1k	Triangles: 404.3k Vertices: 541.8k Time: 0.00ms
Used Textures: 61 / 19.0 MB	Used Textures: 61 / 19.0 MB	Render Textures: 61 / 0.53 GB
On	On	On

## Future & Hope

### • Story Board

When I have spare time in the future, I would like to write a complete story about this character, including born, where he grows, who he meets.

Then make it a story board first and give it to my classmates or friends to criticize. Anyway, I need to create a complete and touching script.

### • Indie game in future

Maybe some days in the future, after I have a complete and touching story as well as a vivid character, I would make it as a indie game projects after work or study.

I would explore more possible playmodes in games insides the game. I also tend to invite some of liked-minded fellows to join me.





# Nightmare



Tag:

Adventure

Action

Atmospheric

## Brief Introduction :

You enter the nightmare and you have to face your heart demon.

## Video(Quick look around 2 mins) :

<https://youtu.be/5oUK0401B1k?si=8AZ1rPY7hw2BuNw5>

## Video(Whole process around 9 mins) :

<https://youtu.be/26c-BbQJDxM?si=DXX1ugFYUTLxTj82>

## Game Downloaded in itch.io (around 160MB):

<https://devil-ovo.itch.io/nightmare>





# Overview

## • Basic Information

Producer : Jian Junren (Devil\_ovo)

Art & Code : Jian Junren (Devil\_ovo)

Render Pipeline : HDRP

## • Introduction

You enter the nightmare and you have to face your heart demon. You have to conquer the enemy to leave this bad dream.

## • Story Outline

Actually, the surrounding around you in dream is the reflection of the obstacles and depression in reality.

You are just a girl who dislikes physics, sports and teachers. And what you are the most afraid of is the beetle near your house.

You have this dream after a big fight with your mother about the proposal that you wanted to learn piano at weekends. You fell in love with piano at the first sight, and it was your dream to become a pianist to perform at the ceremony of the school.



# Inspiration

I got this idea from my experience of my childhood as well as one of my friends. I still remember that I was extremely reluctant to go to school because of the strict teachers and exams.

And my friend, she hated the sport and physics lessons. Also like most of the girls, she was afraid of insects, especially beetles.

And children have lots of dreams when they sleep. So I developed an idea to use a nightmare to express the experience I and my friend as well as other children shared together in our childhood.

# Reference

## • Art style



Control



Little Nightmares II

## • Story outline



What Remains of Edith Finch



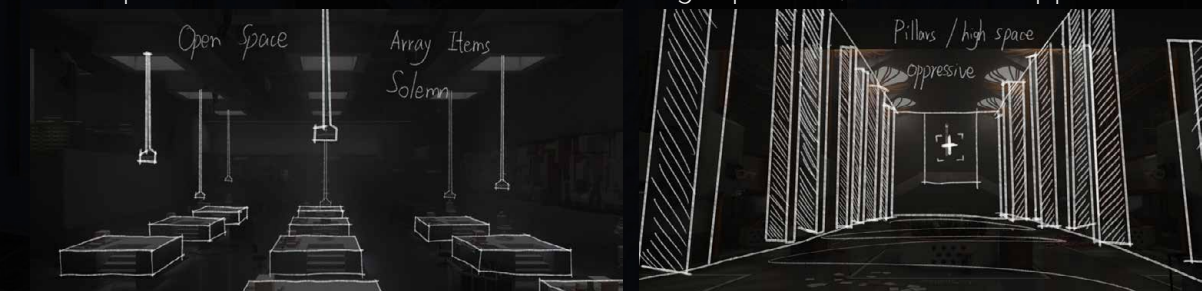
Gris

# Pre-phase Research

## • Research on Control

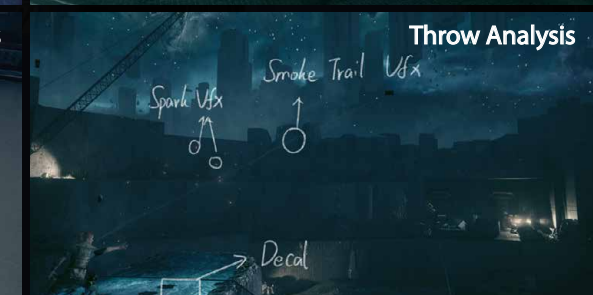
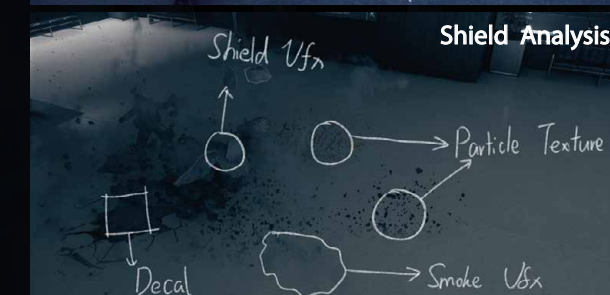
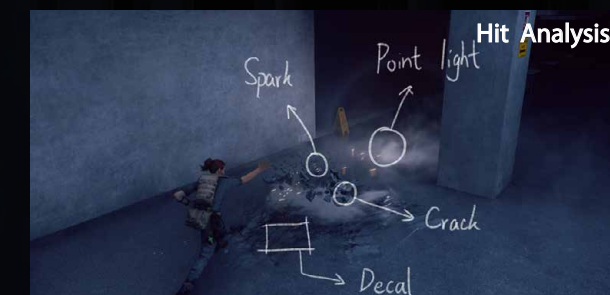
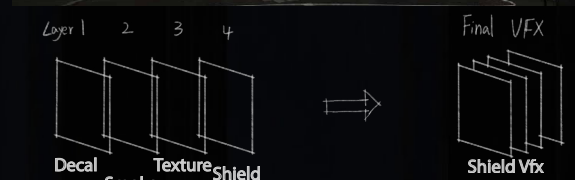
### Scene Analysis :

The empty space and the array of items could bring tense and solemn atmosphere in the classroom. The first battle scene of boss in *Control*, it is constructed by huge pillars, which is oppressive.



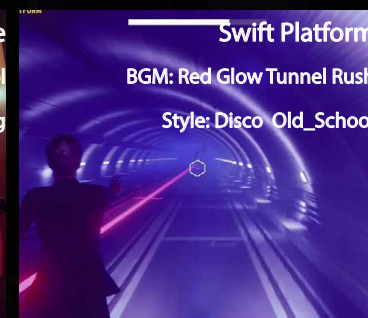
### VFX Analysis :

Actually, many vfxs could be made up of various effects.



### Music & Ambient Analysis :

There are many music in *Control* helping the scene become more atmospheric and attractive.





# Ambition

## • Introduction

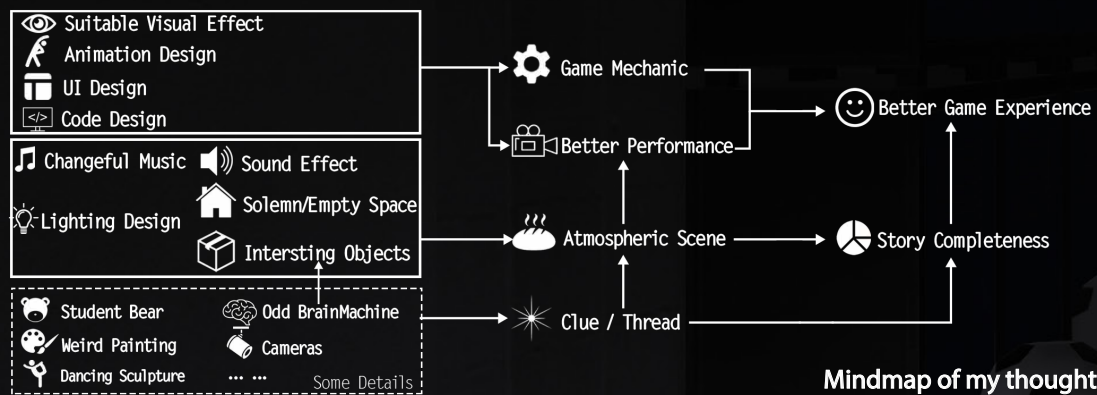
Among the reference of games, My most favourite one is *Control*. Not only because it has a special taste of architecture space, but also its combat, visual effects, scenery, music and other things all combine together closely, which creates a fantastic artwork as well as a scene telling a story.

## • Ambition of this project

I want to make a game which could inspire players to imagine what happened to the girl and immerse themselves in the scenery I created.

After researching and analysing other games, I suppose the key point is the details. So you would find I focus on making different kinds of details to help players soak themselves in this atmospheric game. That's why I make so much efforts on these aspects in order to build a convincing scene.

## • What I do in this game



## Aspect of main level design :

The iteration between the initial thought and final design took me quite a long time. In fact, I made several projects of the level design. (recommended to browse my website)



## Aspect of other level design :

I also design the MainMenu Scene and EndGameScene to make the whole story complete.



## Aspect of objects detailed design :

There are lots of objects in game that players seldom notice. But they still could help build atmospheric scene.



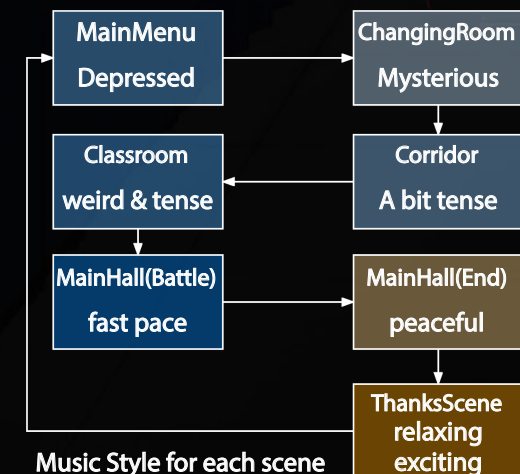
Notice: all paintings on wall in game are drawn by my girl friend Joiz.

## Aspect of VFX design :

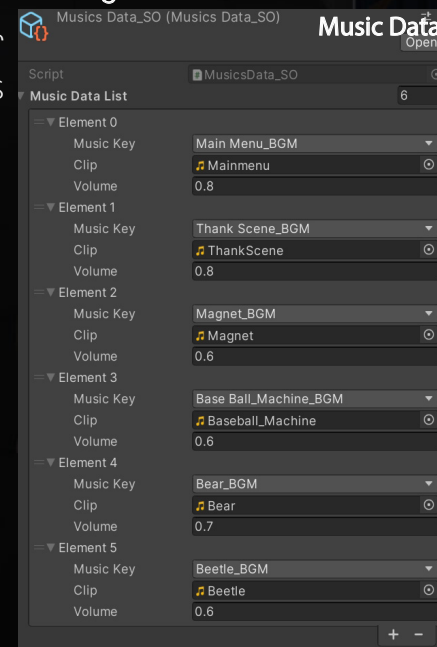
I made countless VFX in this game, which are around 16 Prefabs and 26 variants. More information is in the rest of the pages.

## Aspect of Music or SoundEffect design :

I prepare background music for all scenes, each of which is different from each other.



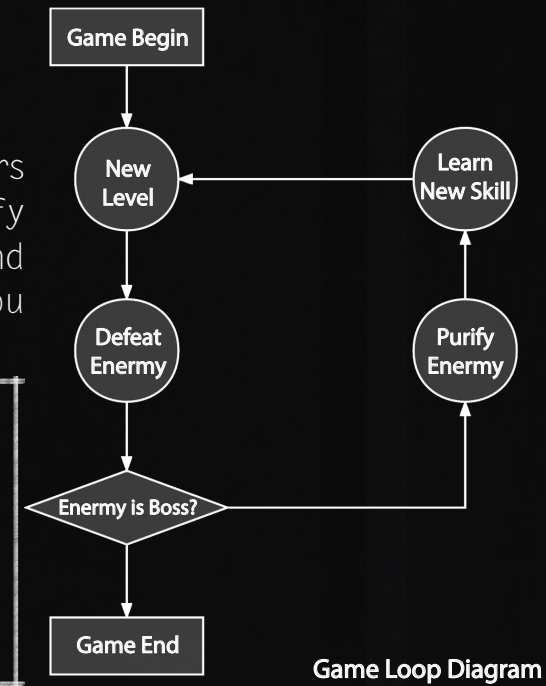
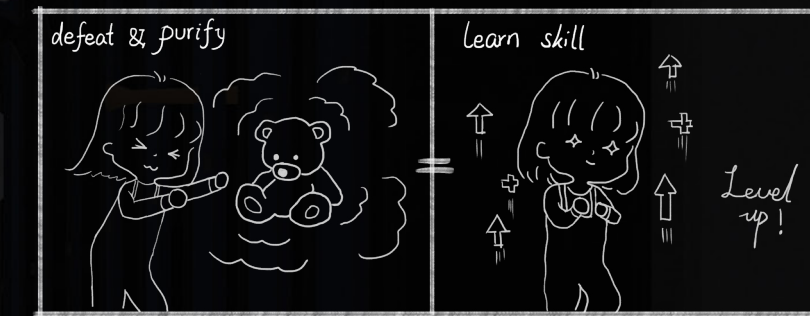
Notice: It is highly recommended to watch the whole process video or play the game.



# Game Mechanics

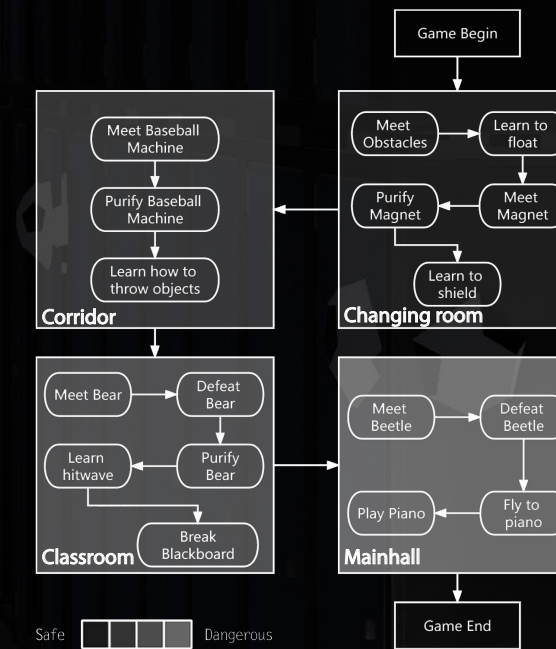
## • Play Loop

The game playing loop is simple. Player enters a new scene to defeat the enemy and purify it. After that, you got the ability of it and becomes stronger. The game ends after you defeat the boss.



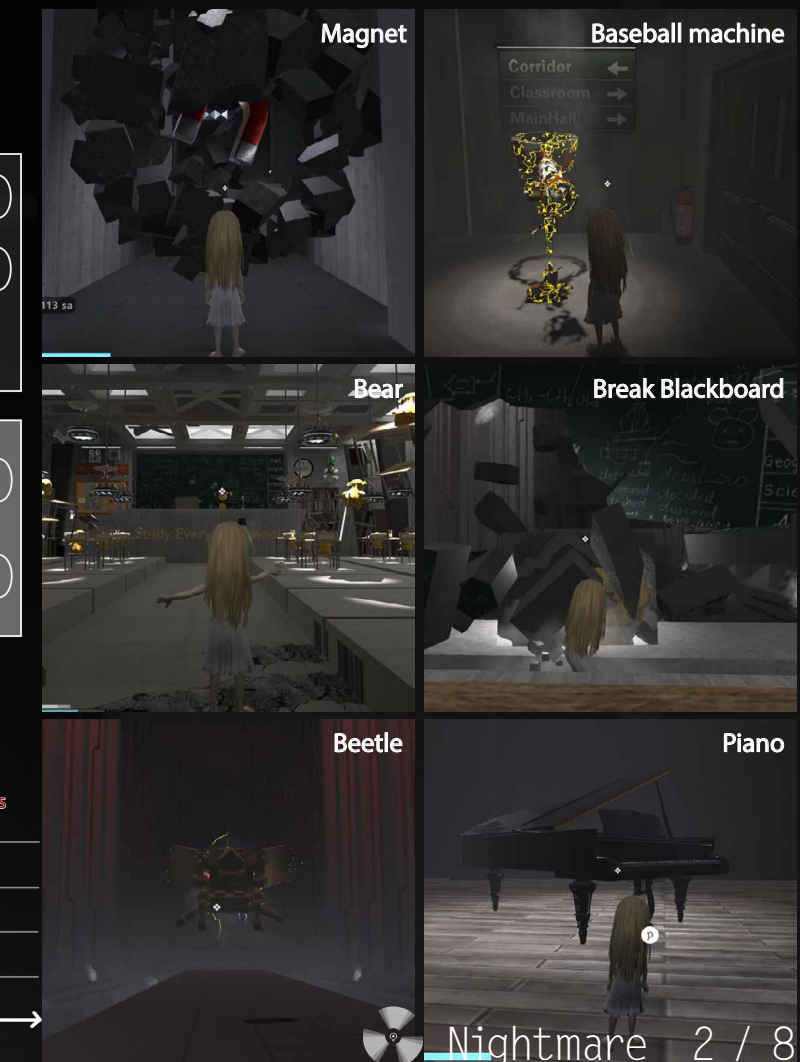
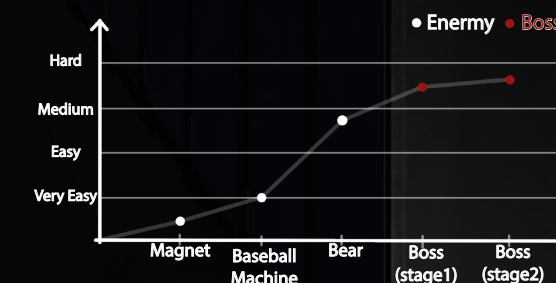
## • Game Progress

### Game Progress Diagram



Safe    Dangerous

### Difficulty Curve

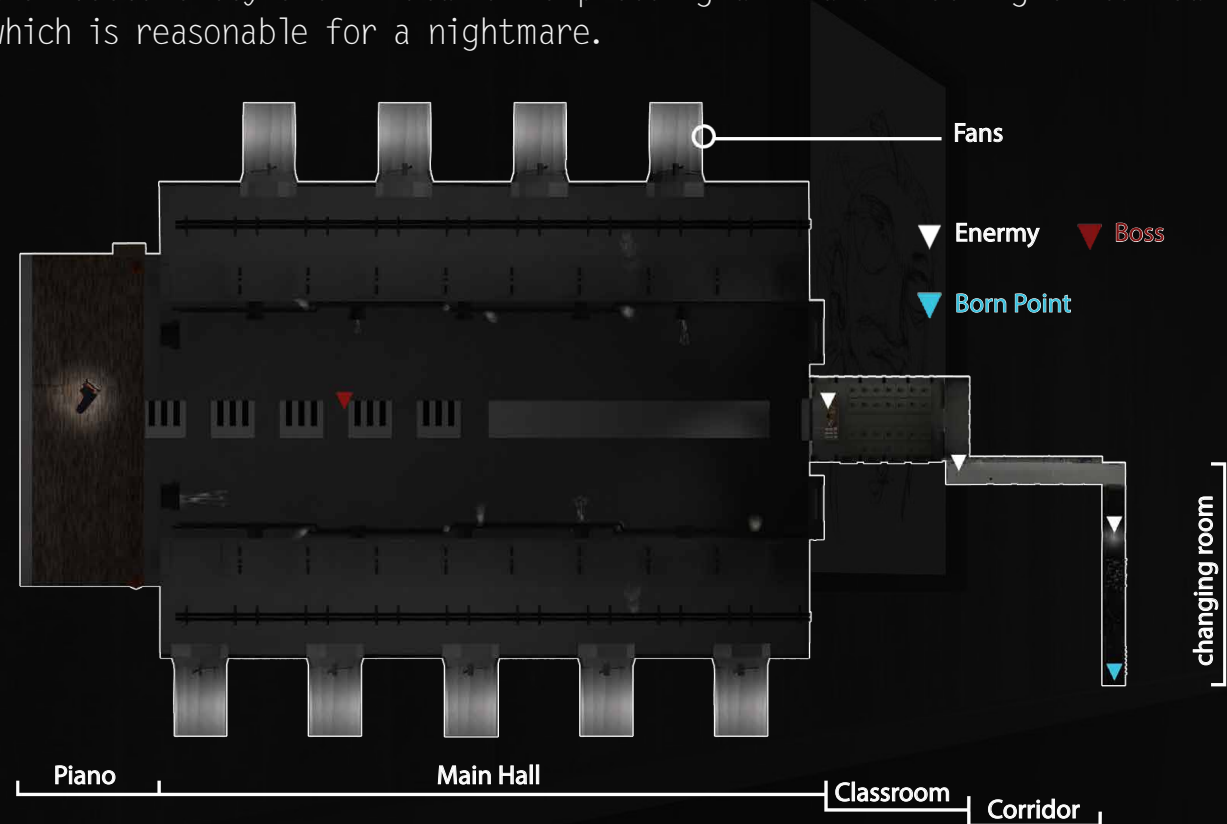




# Level design

## • Final Plan & Style

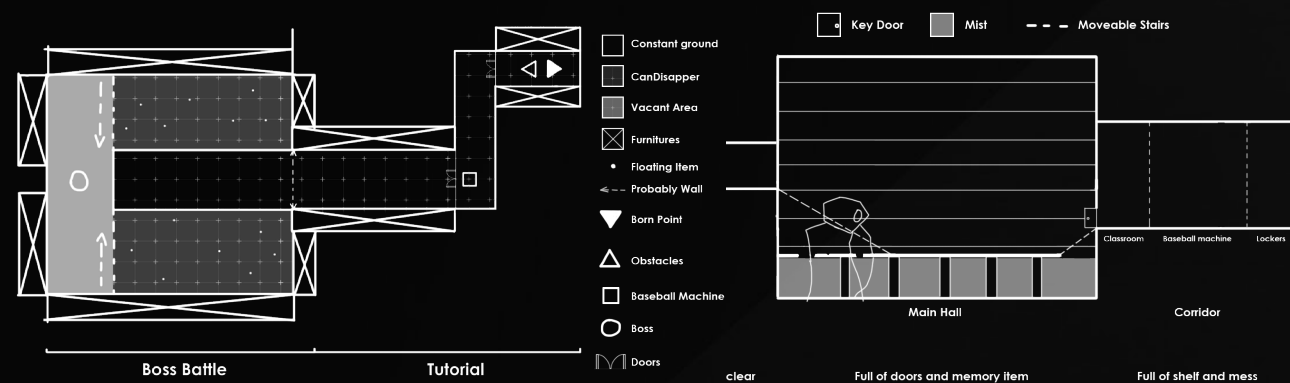
The final level design is made up of four parts, which are changing room, corridor, classroom and Huge Hall(including piano). All scenes are in the architecture style of Brutalism expressing a kind of feeling of surrealism which is reasonable for a nightmare.



## • Level Final Design Sketch

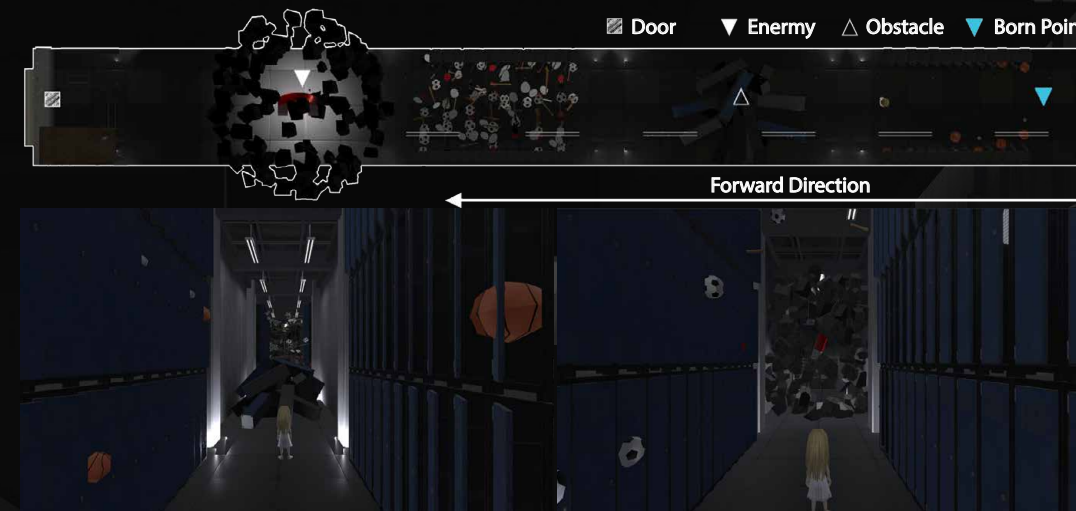
The whole level is divided into two parts : tutorial part where player learn how to use skill and boss battle part.

The size of space of the main hall is the biggest and the floor level is the lowest.



## • Changing room

Changing room is full of lockers containing sports equipments. There is also a magnet which blocks your way.



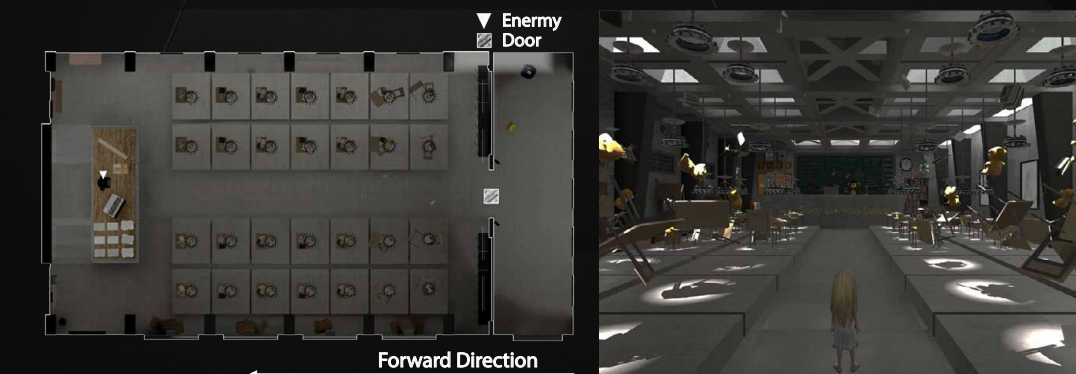
## • Corridor

A corridor is occupied with different pipes emitting steam. What' worse, there is a dangerous baseball machine at the exit.



## • Classroom

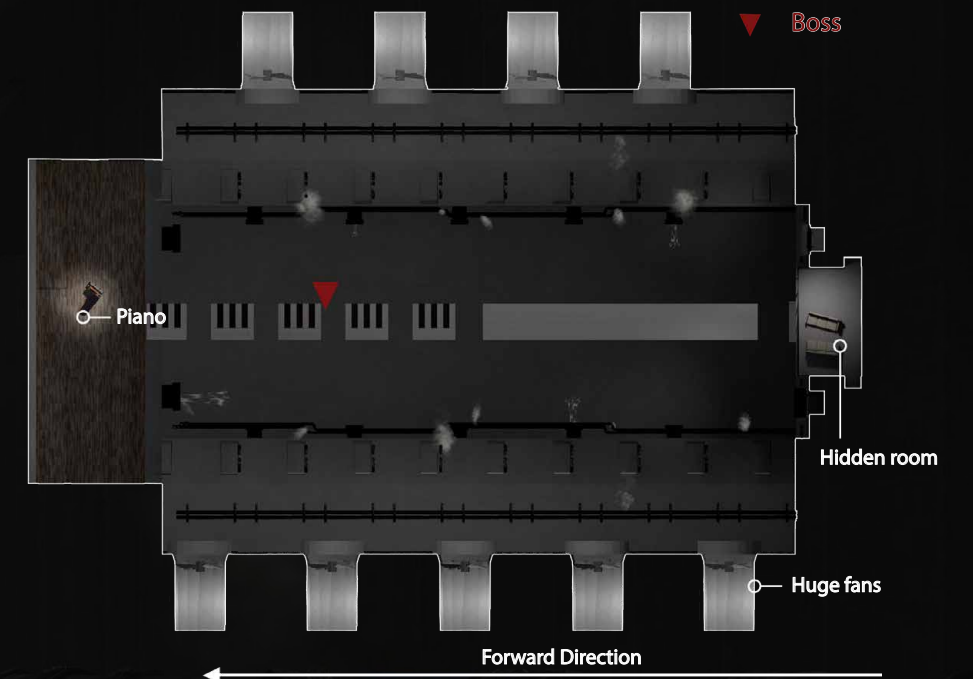
Classroom is controlled by a huge bear. The brain machine seems like that it wants to eat the conciousness I left.



## • MainHall

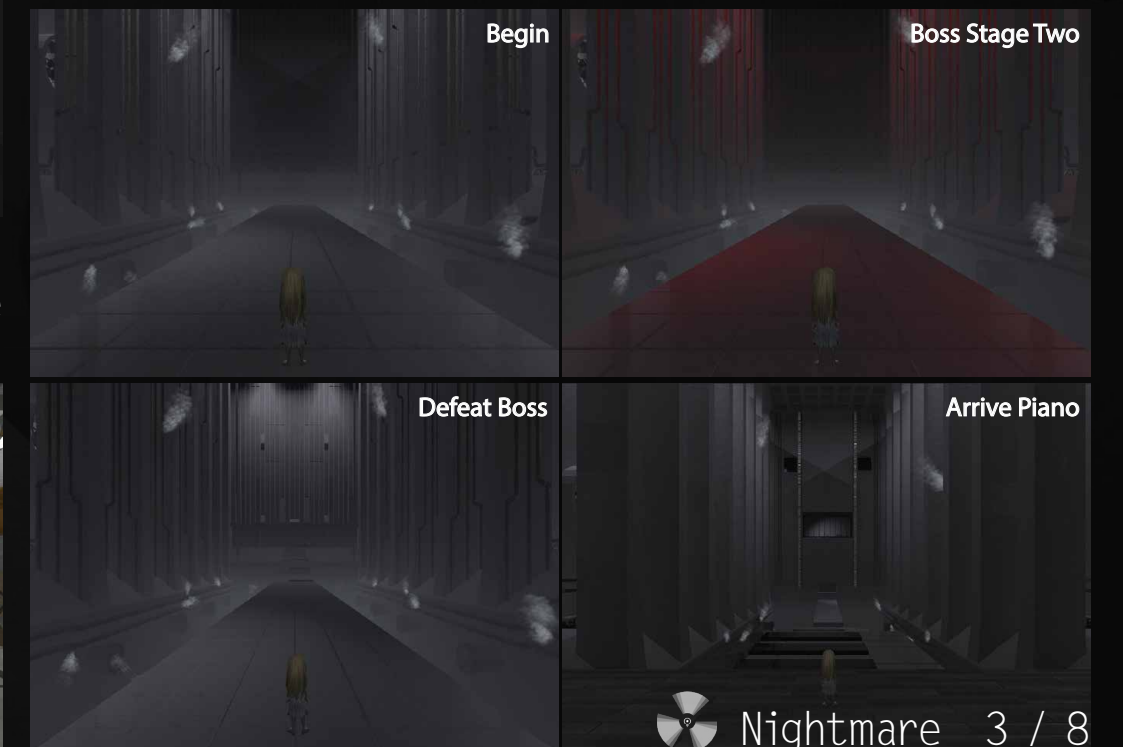
MainHall is like a giant sewer. It is full of different pipes as well. The atmosphere here is dark and scary.

But when the lights are on, it also seems like a mainHall for sacred or important events.



## Different Stages of Scene :

The appearance of the mainhall could be changed after some events such as defeating the boss.





# Player Design

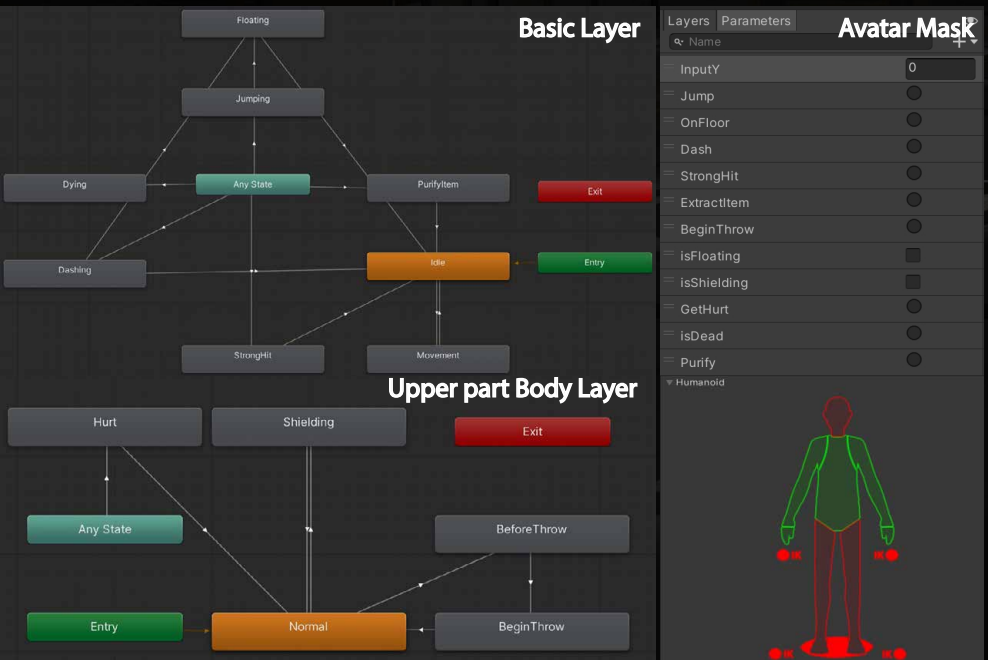
## Player Ability Design

Player Energy Limit : 20  
 Player Energy Recovery Speed : 1/s  
 When Energy Exhausts, Skills are Locked Until  
 Energy Arrive Ratio : 35%

<b>Flashing</b> Energy Cost : 3  Use to avoid injury	<b>Shielding</b> Energy Cost : 5  Use to avoid injury
<b>Floating</b> Energy Cost : 0  Use to avoid injury, pass through obstacles	<b>Throwing</b> Energy Cost : 3 Atk : 3  Use to attack enemy
<b>Purifying</b> Energy Cost : 0  Steal Enemy's Ability	<b>Strong Hit</b> Energy Cost : 5 Atk : 6  Use to attack enemy, Reverse bullet, Break wall

## Player Animator State Machine Design

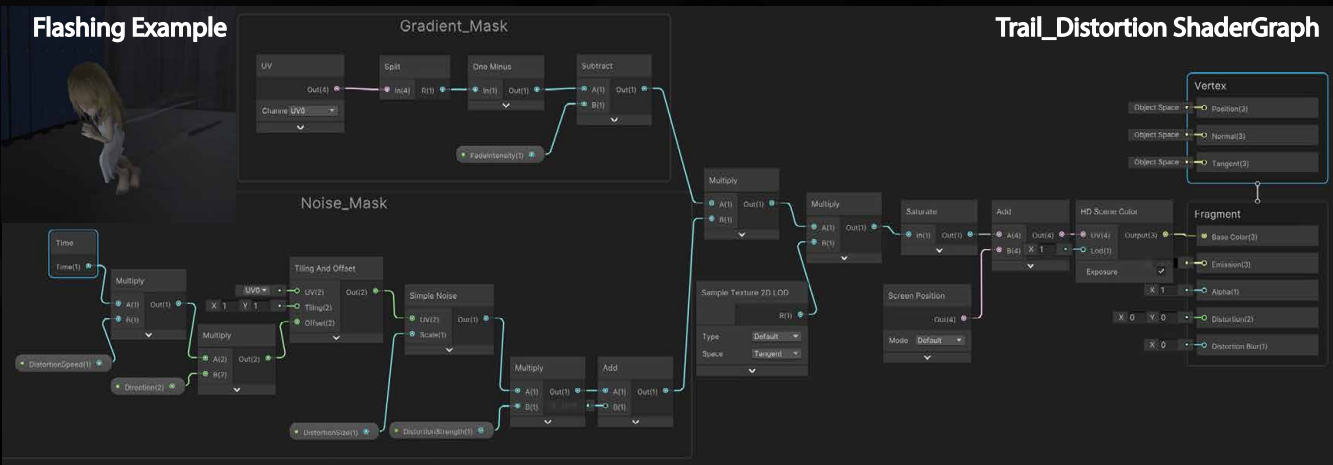
Player sometimes may have multiple actions, such as throwing objects when she is floating. So I use a avatar mask to accomplish it.



## Flashing

Player flashes forwards to avoid injury.

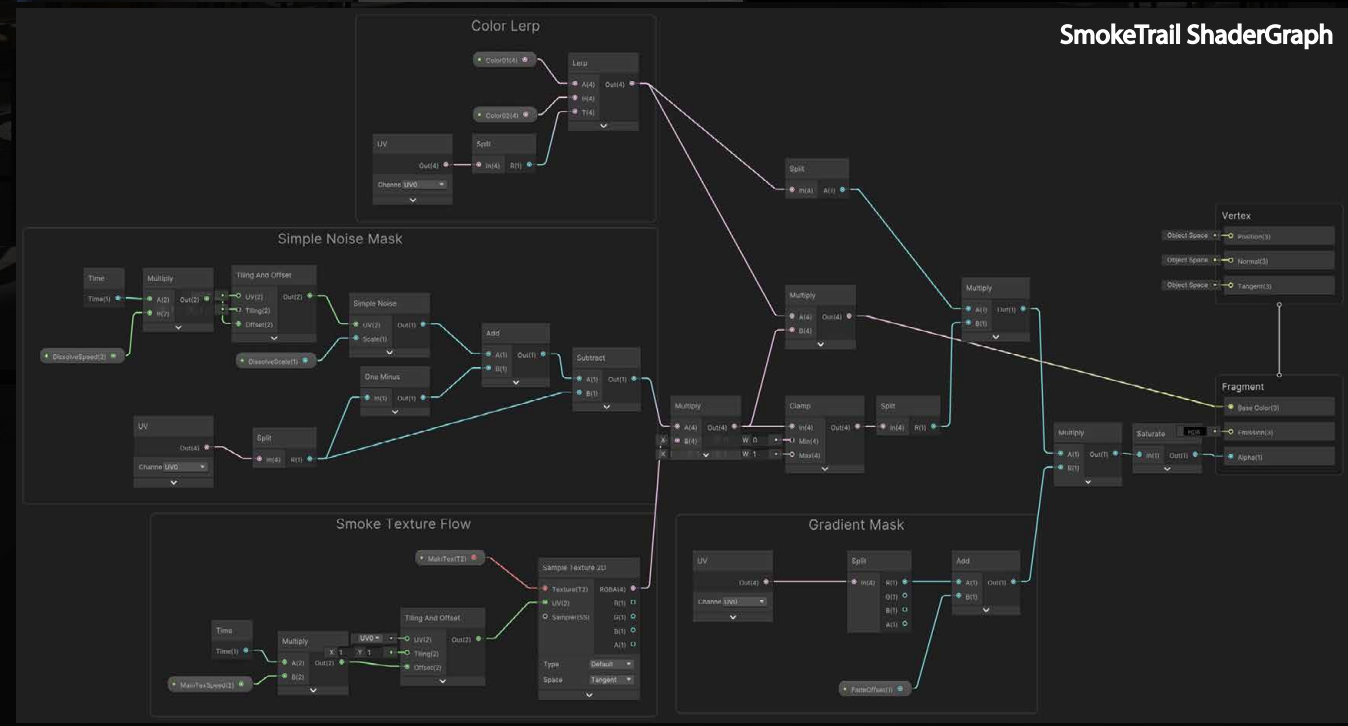
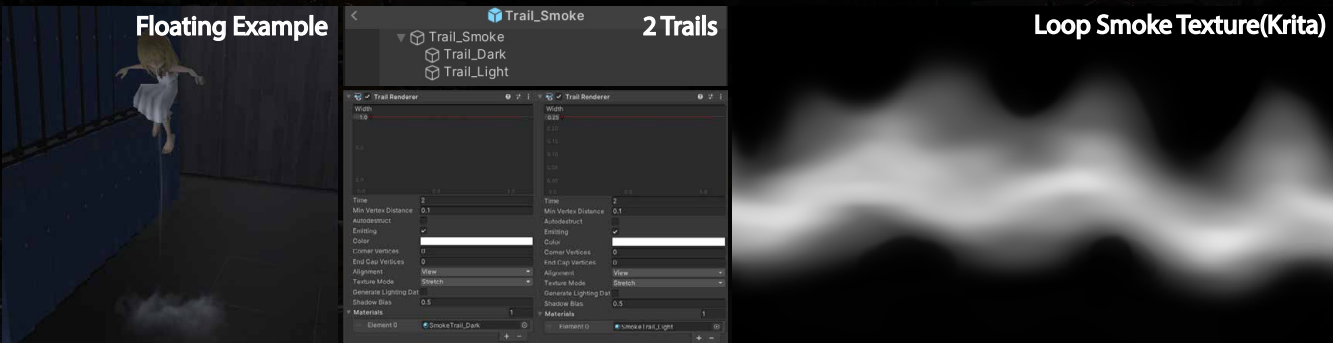
How to achieve : Trail\_Distortion\_Shader + 1 TrailRender + SmokeVFX(in page 7 Other Design)



## Floating

Player floats in the air to avoid injury and pass through obstacles.

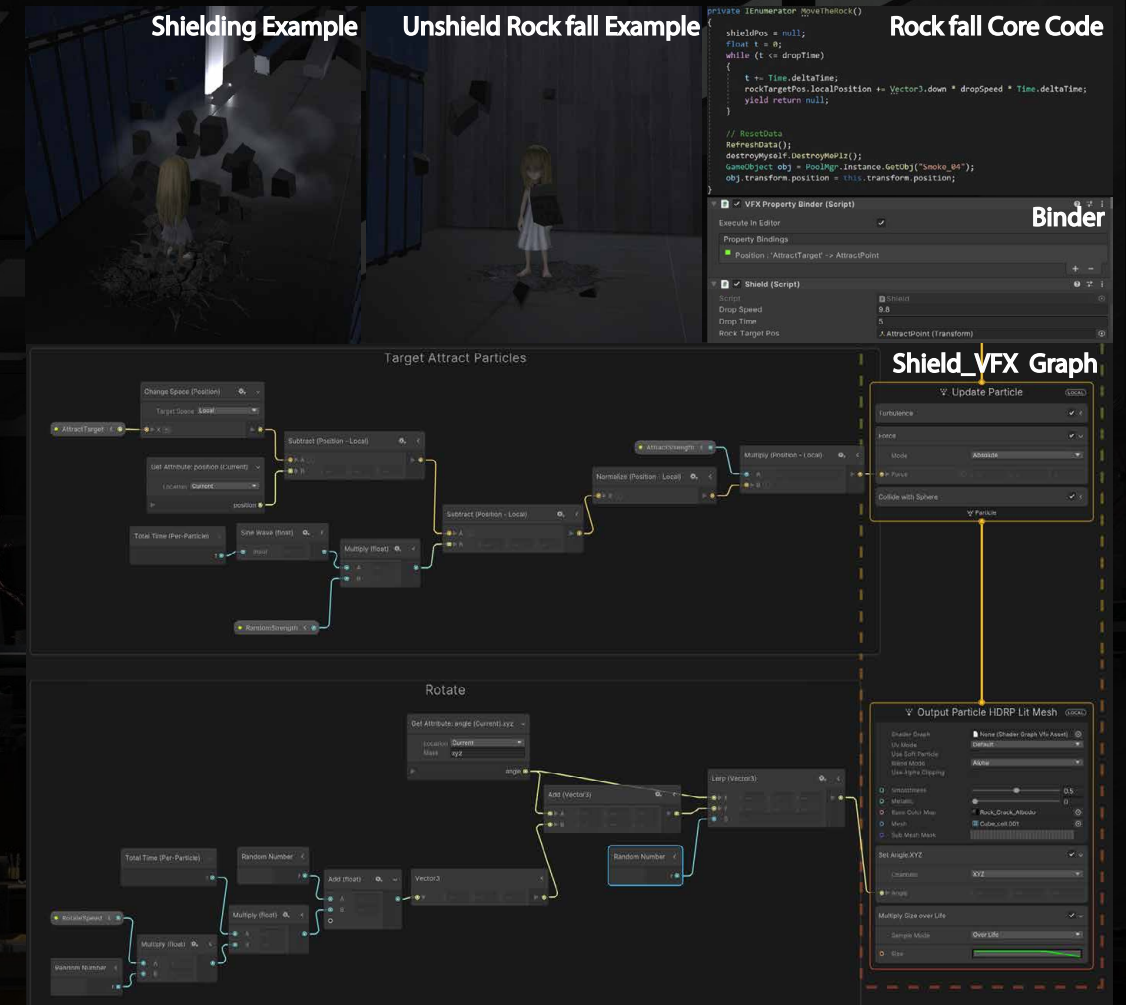
How to achieve : SmokeTrail\_Shader + 2 TrailRender + SmokeVFX(in page 7 Other Design)



## Shielding

Player forms a shield to avoid injury.

How to achieve : Shield\_VFX + Decal + SmokeVFX(in page 7 Other Design)



## Throwing Objects

Player throws a piece of wall crack to attack enemy.

How to achieve : Decal + SmokeTrail\_Shader + TrailRender + SmokeVFX(in page 7)

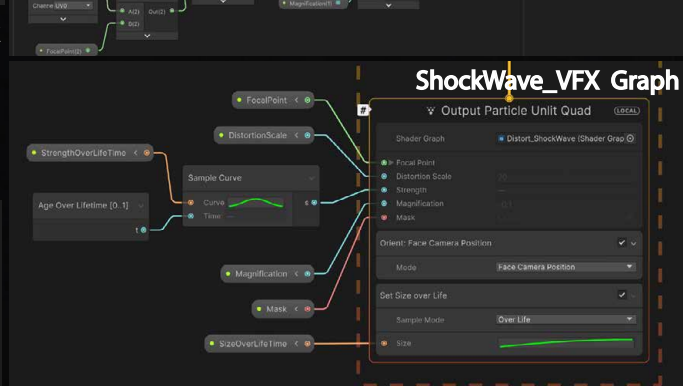
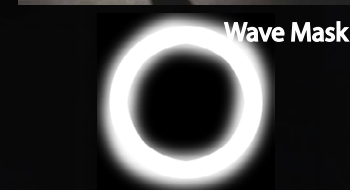
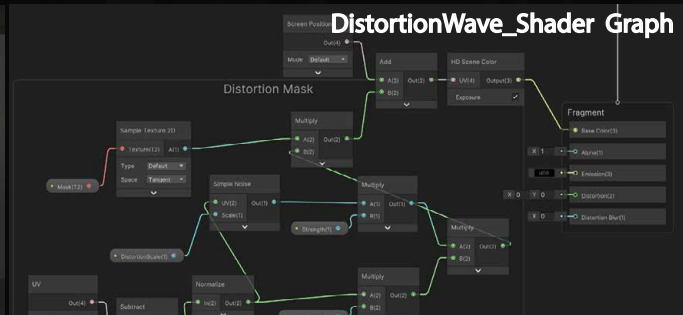




## •Purifying

Player purifies a object and gain its ability.

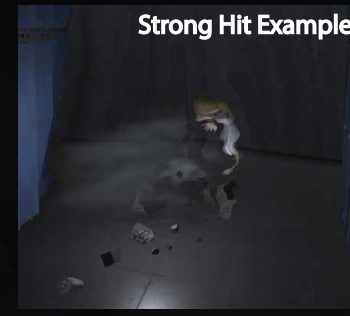
How to achieve :DistortionWave\_Shader + ShockWave\_VFX



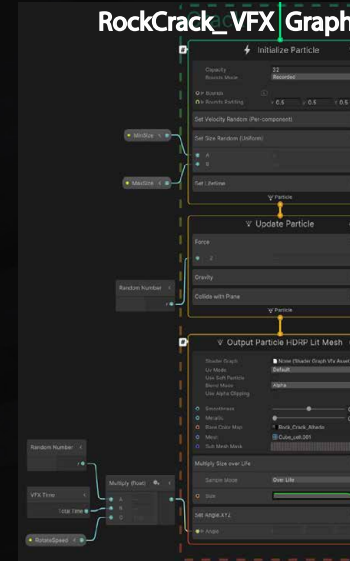
```
Core Code
// Summary: When Purifying, Miss Distortion
private IEnumerator PurifyCoroutine()
{
    if (!IsGeneratingVFX)
    {
        IsGeneratingVFX = true;
        PoolMgr.Instance.GetObject("ShockWave", target.position);
        // Generate VFX
        EventHandler.CallSoundTrack(E_SoundEffect.UnderPurify);
        yield return new WaitForSeconds(0.5f);
        IsGeneratingVFX = false;
    }
}
```

## •Strong Hit

How to achieve :DistortionWave\_Shader + Decal + ShockWave\_VFX + RockCrack\_VFX + PointLight + SmokeVFX(in page 7 Other Design)



```
StrongHit Core Code
public IEnumerator StrongHit()
{
    if(!DataMgr.Instance.isOutEnergy)
    {
        DataMgr.Instance.EnergyCost(strongHitCost);
        // Anim
        anim.SetTrigger("StrongHit");
        // SFX
        EventHandler.CallSoundTrack(E_SoundEffect.StrongHit);
        // Smoke Effect
        float offset;
        for (int i = 0; i < 3; i++)
        {
            offset = Random.Range(-0.2f, 0.1f);
            GenerateViusalEffect("Smoke_Forward", this.transform.position + this.transform.forward * (smokeForwardOffset + offset), this.transform.rotation);
        }
        GenerateViusalEffect("Smoke_01", this.transform.position);
        yield return new WaitForSeconds(0.7f);
        // Particles Effect
        GenerateViusalEffect("Rock_Crack_01", this.transform.position + this.transform.forward * particalSystemOffset, this.transform.rotation);
        // PointLight
        GenerateViusalEffect("Point_Light", this.transform.position + this.transform.forward * pointLightOffset.x + Vector3.up * pointLightOffset.y);
        // ShockWave Effect
        GenerateViusalEffect("ShockWave", rightHand.position + this.transform.forward * shockWaveOffset);
        // Decal
        for (int i = 0; i < 2; i++)
        {
            int tmp = Random.Range(1, 4);
            GenerateViusalEffect("Decal Crack_0" + tmp.ToString(), this.transform.position + this.transform.forward * (decalOffset + tmp * 0.1f));
        }
    }
}
```



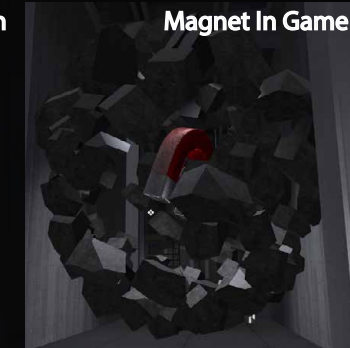
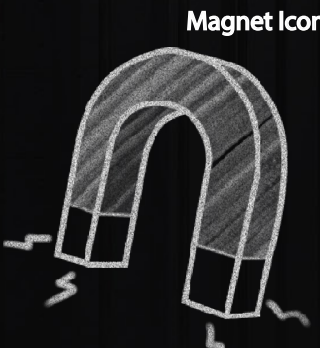
# Enemy Design

## •Magnet

Introduction : A magnet wants to protect itself, attracting lots of iron stones.

Ability : Form a shield to avoid being attacked.

How to achieve :Shield\_VFX(page 4 player shield) + Dissolve\_Shader(in Bear)



```
Core Code
private IEnumerator OpenAndCloseShield()
{
    if(!IsPlayerEnterShield && !IsShield && !IsDead)
    {
        EventHandler.CallSoundTrack(E_SoundEffect.Shield_Magnet);
        isShield = true;
        vfx.gameObject.SetActive(true);
        vfx.Play();
        coll.SetActive(true);
        // equal to vfx Life time
        yield return new WaitForSeconds(15f);

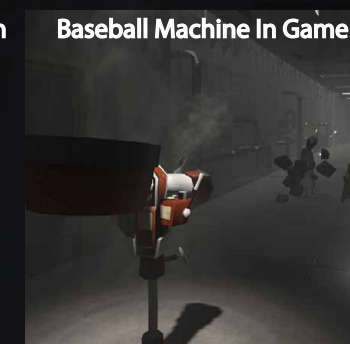
        coll.SetActive(false);
        vfx.gameObject.SetActive(false);
        yield return new WaitForSeconds(2f);
        isShield = false;
    }
}
```

## •Baseball\_Machine

Introduction : A baseball machine which loses its control. It could be painful to get hurt from it.

Ability : Fire a baseball

How to achieve :BulletSpark\_VFX + Dissolve\_Shader(in Bear)

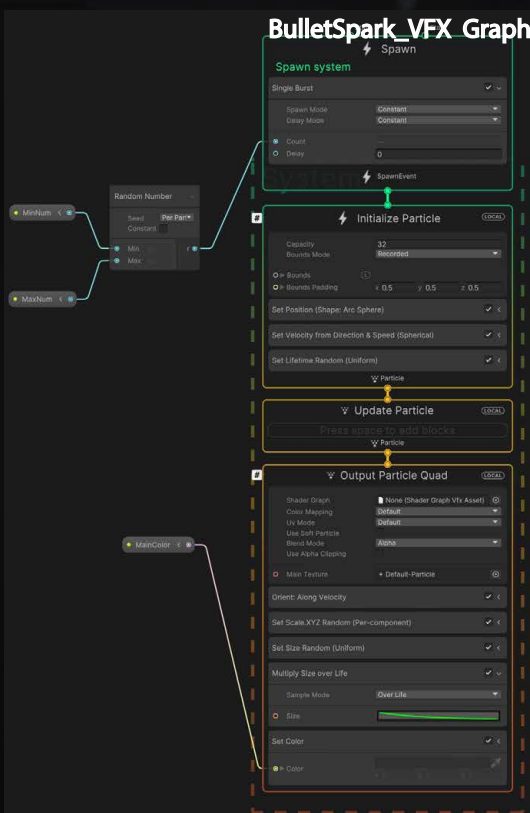


```
Baseball Machine Core Code
private void TargetPlayer()
{
    if (player == null)
        return;
    if(canFire)
    {
        this.transform.rotation = Quaternion.LookRotation(new Vector3(player.transform.position.x - transform.position.x, 0, player.transform.position.z - transform.position.z));
    }
}

private void OnTriggerEnter(Collider other)
{
    // Bullet Core Code
    if (other.gameObject.CompareTag("Shield"))
    {
        // SoundEffect
        EventHandler.CallSoundTrack(E_SoundEffect.Baseball_HitShield);

        isHitThing = true;
        rb.useGravity = true;
        rb.velocity = Vector3.zero;

        // VFX(Spark)
        PoolMgr.Instance.GetObject("Spark_01", this.transform.position);
        StartCoroutine("DestroyMePlz", 0);
    }
}
```

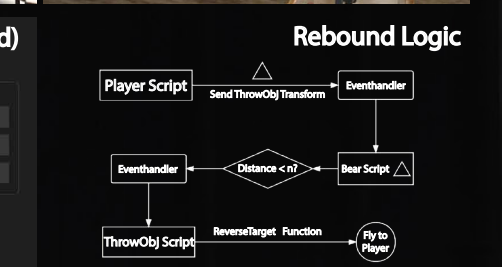
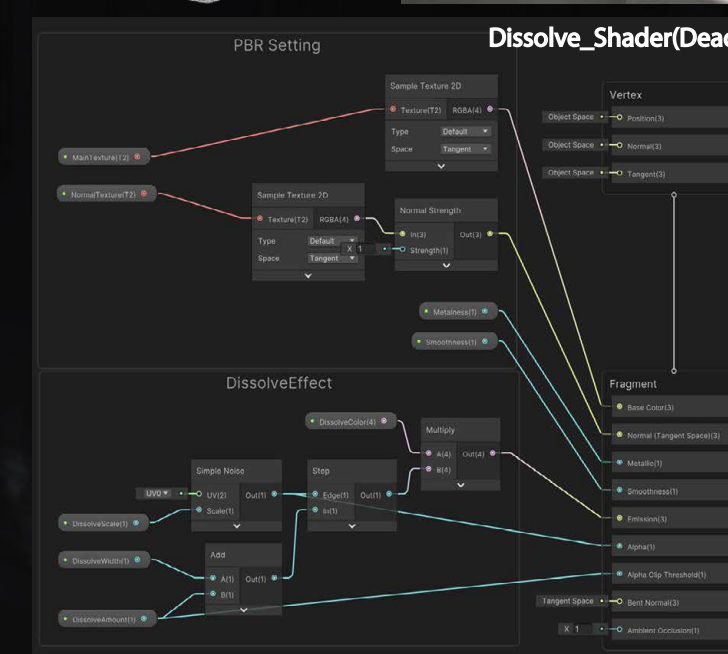
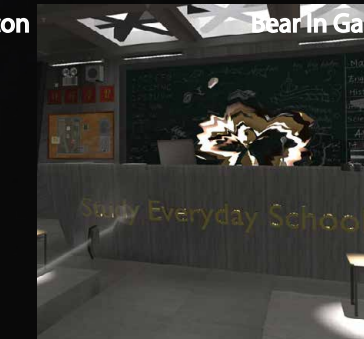


## •Bear

Introduction : A giant bear who looks like my teacher loves throwing chalk eraser to students.

Ability : Throw Chalk Eraser, Rebound Bullets

How to achieve :DistortionWave\_Shader + ShockWave\_VFX + Dissolve\_Shader(Dead) + DeadSpark\_VFX



```
Rebound Code
// Bear Script
private void OnPlayerExtractObj(bool isExtract, Transform playerThrowObj)
{
    if (isExtract)
        this.playerThrowObj = playerThrowObj;
}

private void OnPlayerThrowObjDestroy()
{
    this.playerThrowObj = null;
}

private void DecidedHatToObj()
{
    reverseTime += Time.deltaTime;
    skillTime += Time.deltaTime;
    if (playerThrowObj != null && reverseTime == reverseCOTime)
    {
        if (NextBolt.Distance(this.transform.position, playerThrowObj.position) <= 8f)
        {
            EventHandler.CallReversePlayerThrowObj(this.transform.position);
            StartCoroutine("ReverseObj");
            reverseTime = 0;
            return;
        }
    }
    if (skillTime == betweenSkillCOTime)
    {
        StartCoroutine("ThrowObj");
        skillTime = 0;
    }
}
```

```
Throw Obj Script
private void OnReversePlayerThrowObj(Vector3 bearPos)
{
    if (Vector3.Distance(bearPos, this.transform.position) <= 8f)
    {
        SetTarget(playerCtrl.transform.position);
        isReverse = true;
        Destroy(this.gameObject, 15f);
    }
}

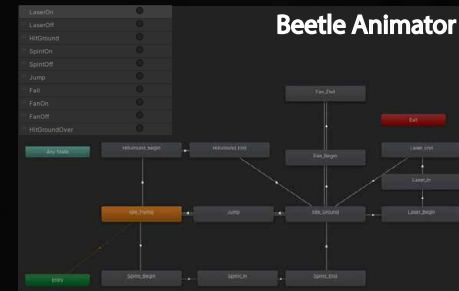
DeadEffect Code
protected virtual IEnumerator DeadEffect()
{
    // DissolveEffect
    float t = 0;
    EventHandler.CallSoundTrack(E_SoundEffect.Dissolve_Energy);
    if (vfx != null)
    {
        vfx.gameObject.SetActive(true);
        while (t < 1)
        {
            t += Time.deltaTime / dissolveTime;
            mat.SetFloat("DissolveAmount", t);
            yield return null;
        }
        PoolMgr.Instance.GenerateObj("DeadSpark", this.transform.position + Vector3.up * healthObjGeneratorOffset);
        Destroy(this.gameObject, 1f);
    }
}
```



# Boss Design

## •Boss Ability Design

Beetle Hp : 70  
Stage 2 Begin Hp : 75%



Ability	Atk	Condition	Player Strategy
HitGround	4	-	Floating
MeeleAtk	4	-	Floating, Flashing
Laser	1/s	Only stage 2	Shielding
ThrowTrunks	3	Only stage 2	StrongHit, Flashing
FanAtk	1.5/s	Only stage 2	Flashing
Recovery	-	Hp <= 66%	Throwing

## •HitGround

Hit ground with great force. Make a cube wave to injure player.

How to achieve :

ShockWave\_VFX(in page 5)  
SmokeVFX(in page 7)

### HitGround Example

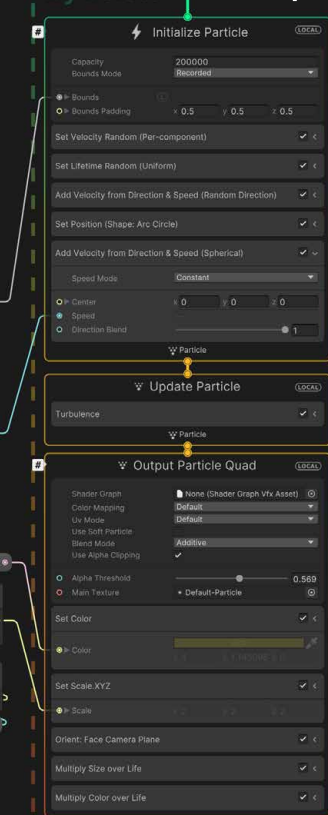


### CubeMgr

```
private void MoveCubes(float rowIdx, float colIdx)
{
    float top;
    for (int i = 0; i < rowIdx.Count; i++)
    {
        top = 1 / (Mathf.Abs(rowIdx[i] - 1) + 1);
        if (top < 0.156f)
            top = 0;
        rowIdx[i].MoveCubes(top);
    }
}

public void MoveCubes(float height)
{
    for (int i = 0; i < cubeList.Count; i++)
    {
        if (height == 0)
            iSkip = iSkip + 1;
        else
        {
            iSkip = 0;
            cubeList[i].Move(height * Mathf.Pow(1.5f, Mathf.Abs(i - 2.5f)));
        }
        if (iSkip >= thisSound)
        {
            EventDispatcher.CallSoundTrack(1, SoundEffect.HitGround_Cubes);
            hasSound = true;
        }
        else if (iSkip >= thisSound)
            hasSound = false;
    }
}
```

### Meele Attack\_VFX Graph

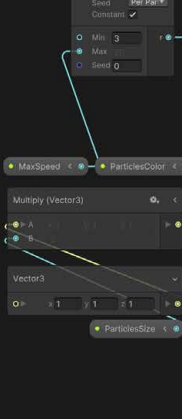
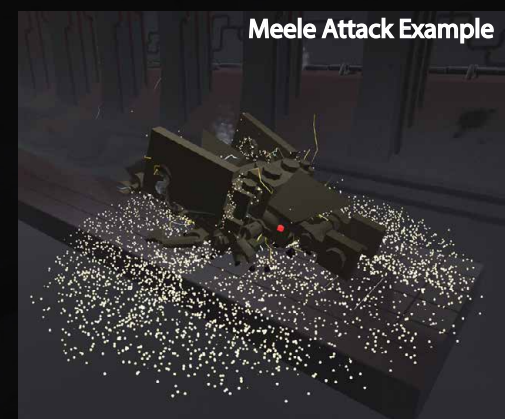


## •Meele Attack

Fly close to the player and attack.

How to achieve :

MeeleAttack\_VFX  
Decal  
RockCrack\_VFX (in page 5)  
SmokeVFX(in page 7)



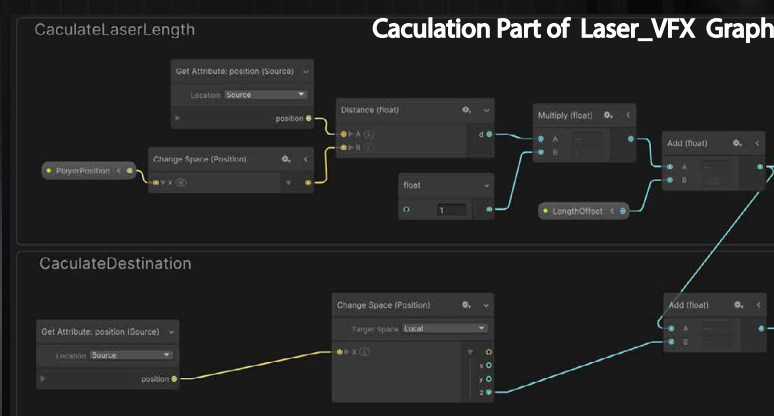
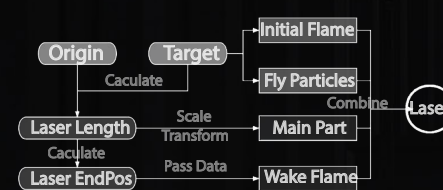
## •Laser

A powerful laser beam, only can be avoided by shield.

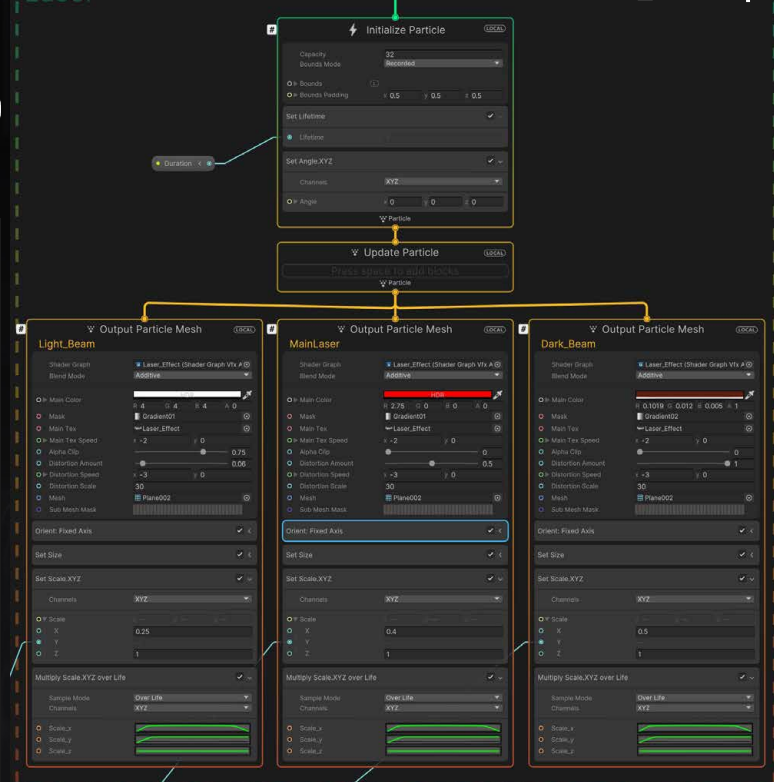
How to achieve : Laser\_VFX + Lightning\_VFX + SmokeVFX(in page 7)



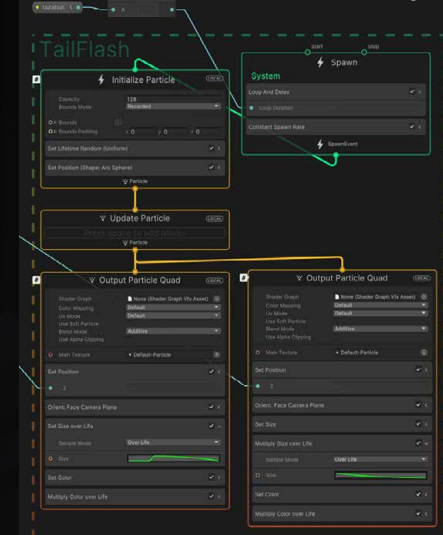
### Mindmap of Laser\_VFX



### Main Part of Laser\_VFX Graph



### Wake Flame Part of Laser\_VFX Graph



## •Throw Trunks

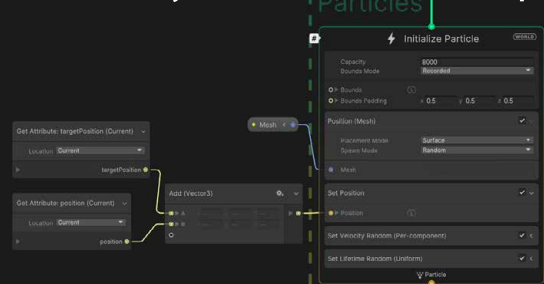
Beetle throws three trunks to the player, causing area of attack effect.

How to achieve : ParticlesTrail\_VFX + DistortionWave\_Shader(in page 5)



### Throw Trunks Example

### Key Part of Particles Trail\_VFX Graph

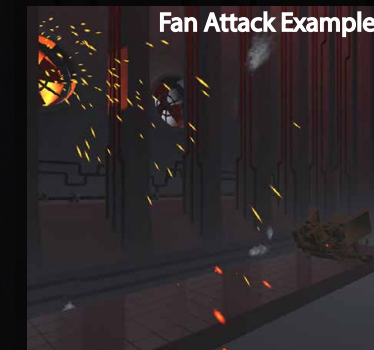


## •Fan Attack

Control one of the fans, let it overload to generate sparks.

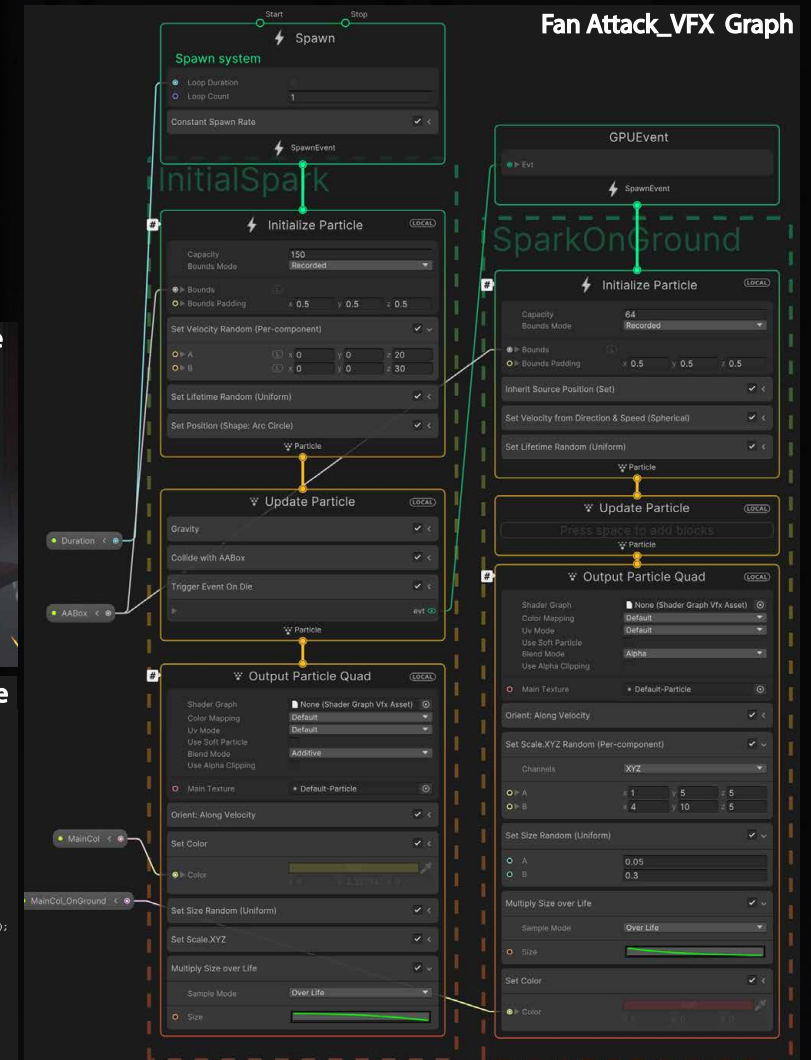
How to achieve :

FanSpark\_VFX  
SmokeVFX(in page 7)



### Fan Attack Core Code

```
private IEnumerator FanAtk()
{
    // Eyes become red
    yield return StartCoroutine("EyesChangeColor", true);
    // Call Fan to work
    int tap = GetPlayerFanNum();
    EventDispatcher.CallFanAccelerate(tap, true);
    // SFX
    EventDispatcher.CallSoundTrack(1, SoundEffect.Fan);
    // Constantly create smoke_vfx
    for (int i = 0; i < 2; i++)
    {
        PostMgr.Instance.GetObject("SmokeBoss", this.transform.position);
        yield return new WaitForSeconds(1f);
    }
    // Eyes become white again
    yield return StartCoroutine("EyesChangeColor", false);
}
```

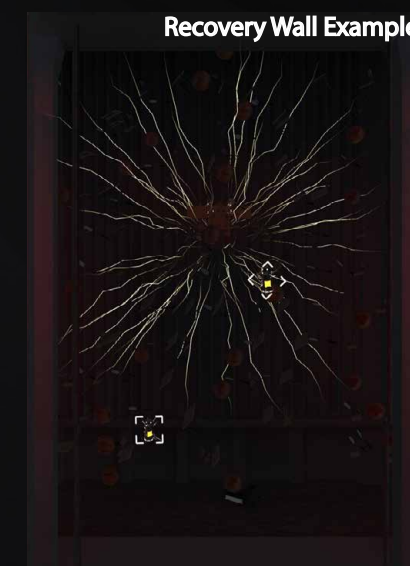


## •Recovery Wall

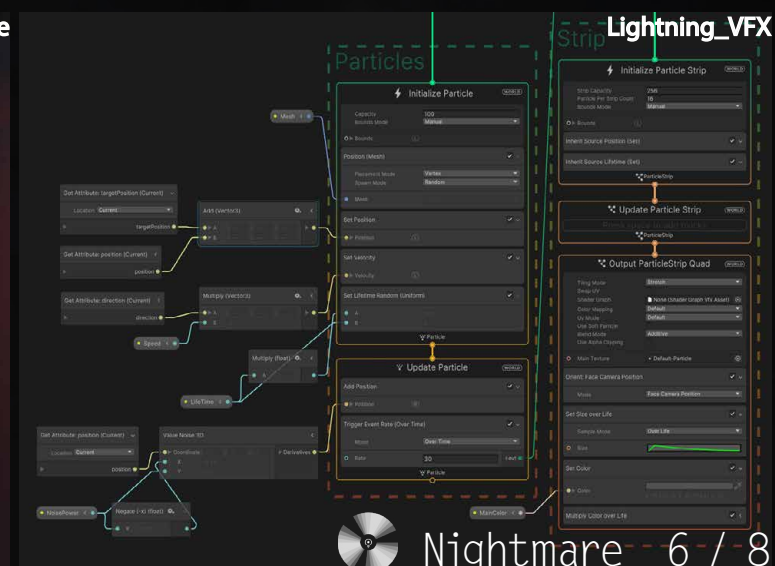
Beetle hides behind the wall to recover, which is made of various items.

Player has to destroy energy tanks to stop that.

How to achieve : Lightning\_VFX + ObjectMoveWithTarget\_VFX(in page 7)



### Recovery Wall Example



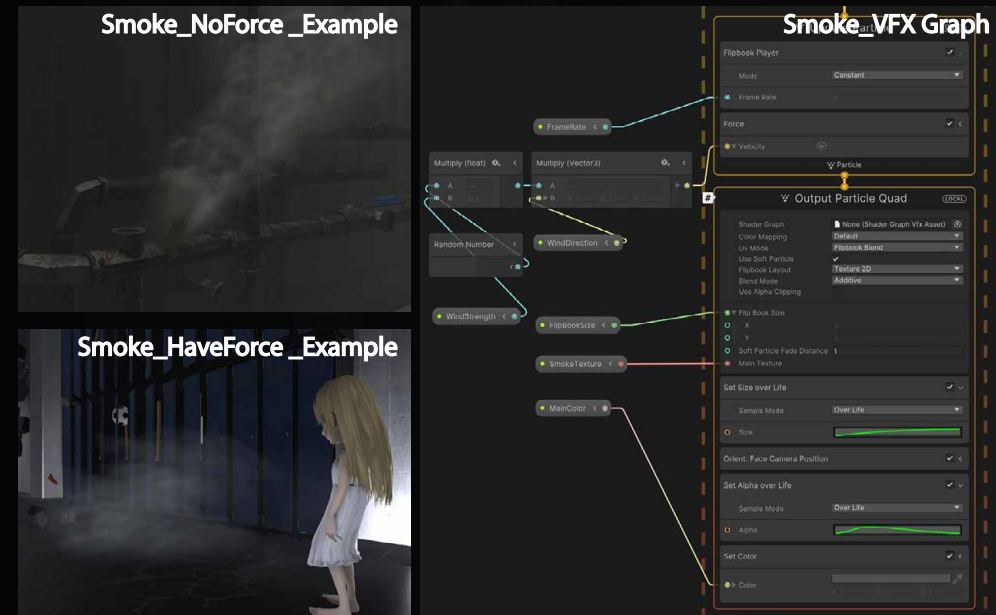
### Lightning\_VFX



# Other Design

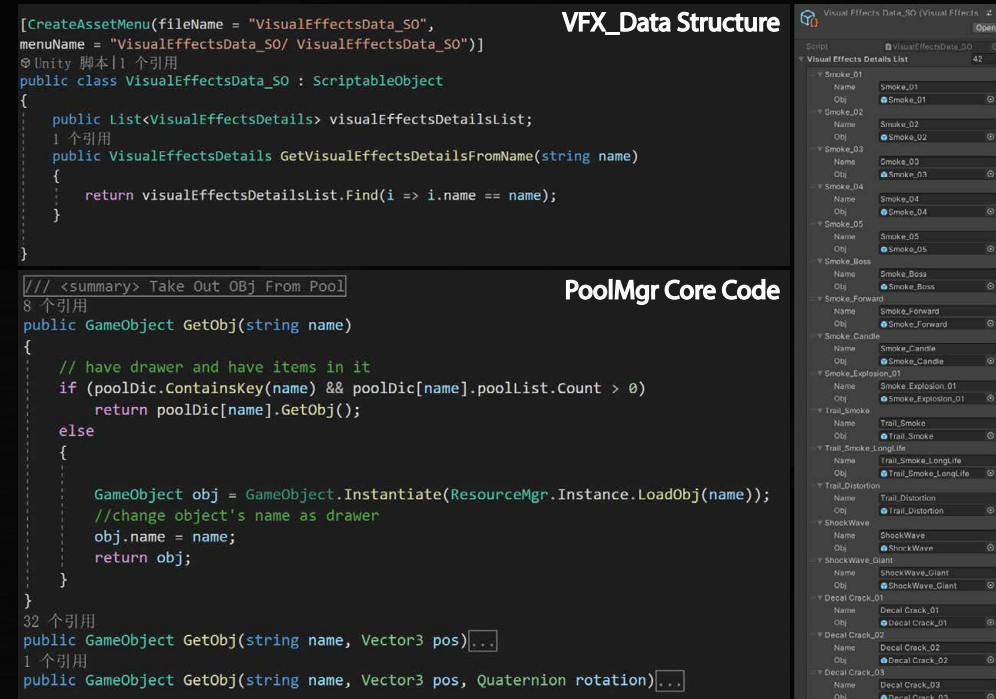
## Smoke\_VFX(flipBook)

Here uses the flipbook to make smoke animation. Smoke could be added force to look more realistic.



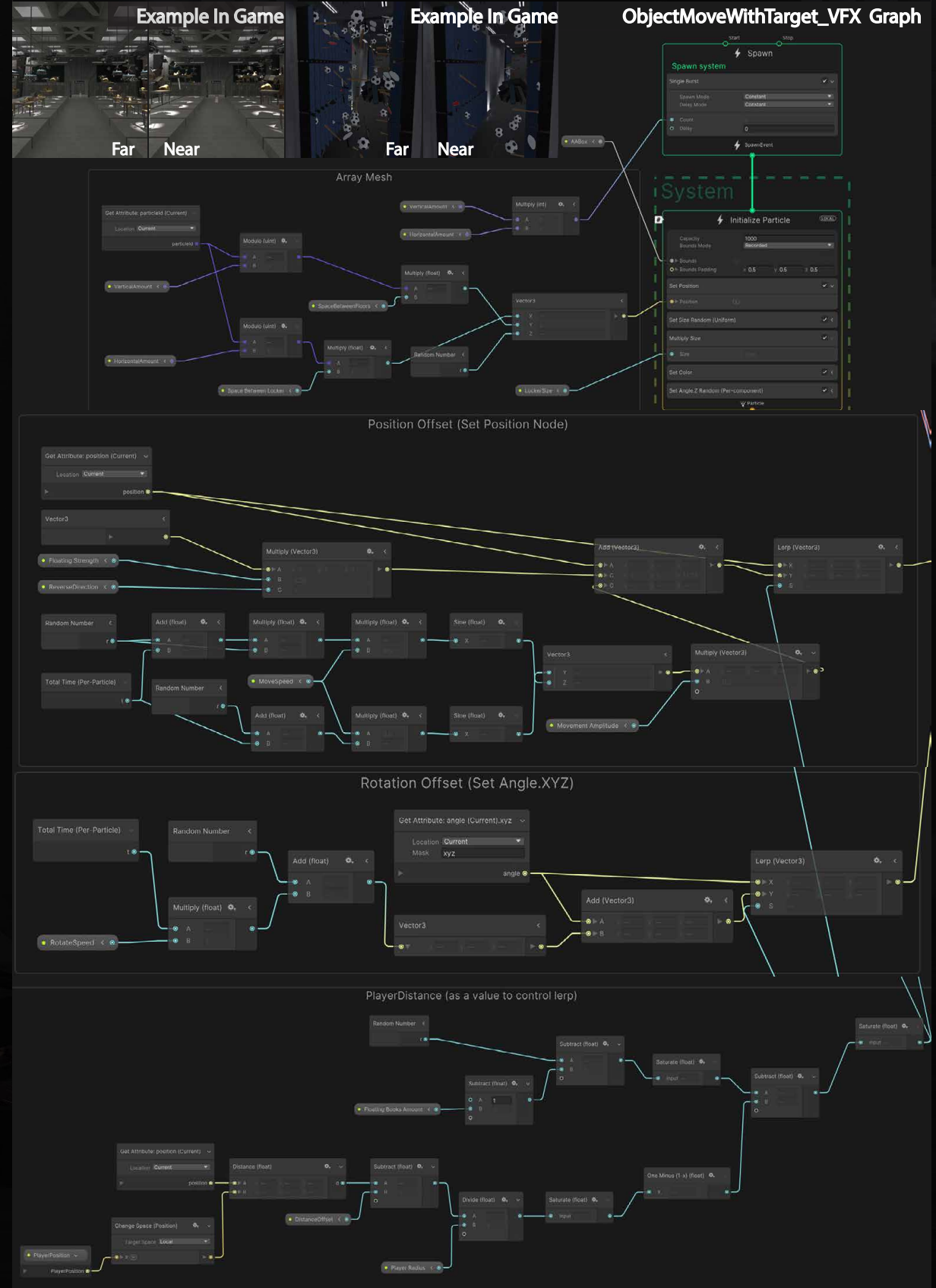
## VFX Management

Because there are countless VFX used in the project, I use ScriptableObject to store the prefabs' data, also use a PoolMgr to manage their generation to decrease the expense.



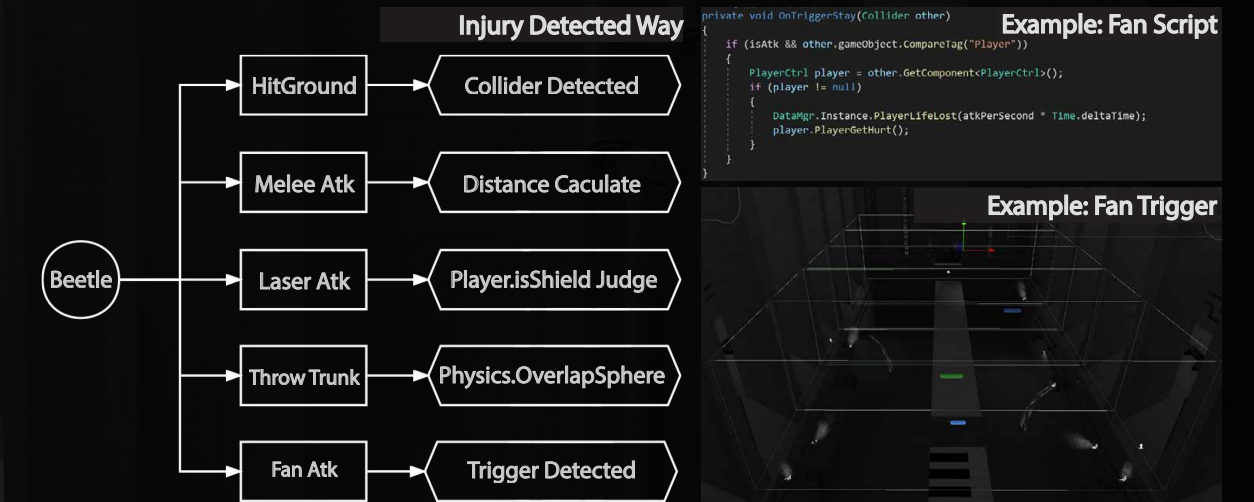
## ObjectMoveWithTarget\_VFX

It is important for an atmospheric scene that it could be interactive with player. Items in the scene will move with the change of player location.



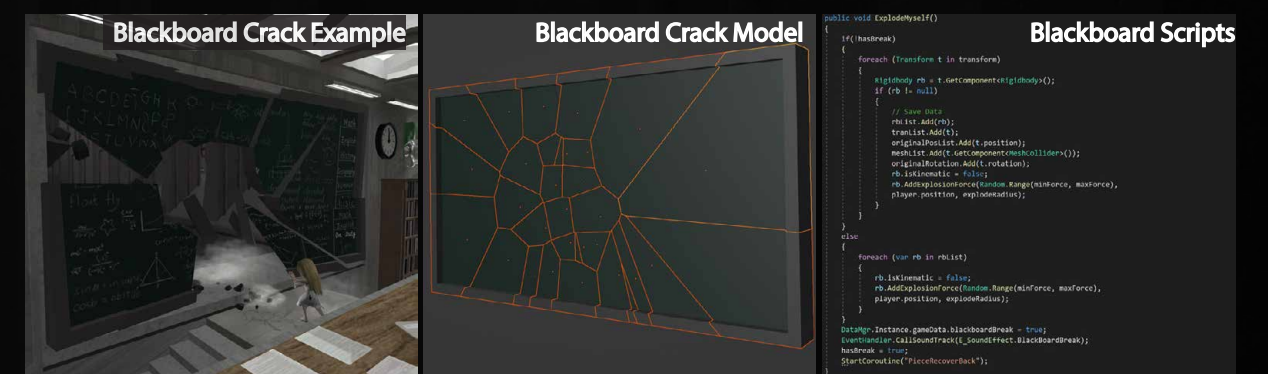
## Injury Detected System Design

Because VFX\_Graph could not judge the injury, I build an independent injury detected system.



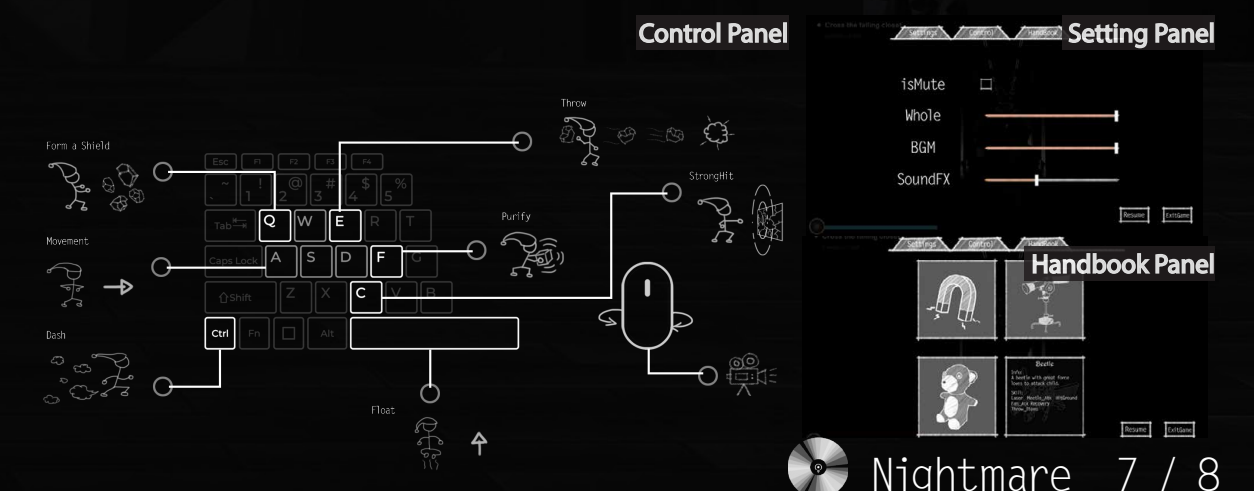
## Object Crack Design (Blackboard)

The scene will become more interesting and interative if the player could break the wall. So I make a destroyable blackboard in game(can recover after few seconds).



## UI Design

I also design the whole UI to suit the style of my theme - nightmare. It seems like chalk painting drawn by a little girl.





# Summary

## •Plan VS Reality

### Difference :

The project\_2 is over 2 weeks ahead of my schedule, which is really surprising. It is mainly because I spend less time in codes development. However, it took me more time in VFX design and shader development.

### Reason :

As my second project, I know how to use MVC structure to construct my codes, which makes my codes neat and tidy.

Because it was my time to get in touch with VFX\_Graph in Unity, it took me more time to understand its mechanic as well as HDRP's.

### Schedule : 14 weeks

Task	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11	Week12	Week13	Week14
Game Design Document	01/08/2023	08/08/2023	15/08/2023	22/08/2023	29/08/2023	05/09/2023	12/09/2023	19/09/2023	26/09/2023	02/10/2023	09/10/2023	16/10/2023	23/10/2023	30/10/2023
Technical Design Document														
Create Levels														
Create Character & Boss														
Create Objects														
Create Animation														
VFX Development														
Create UI														
Shader Development														
Code Development														
Music & SoundEffect														
Testing														
Evaluation														
Create journal of design and implementation ideas														
Create Manual														

### In Fact : 12 weeks

Task	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11	Week12
Game Design Document	01/08/2023	08/08/2023	15/08/2023	22/08/2023	29/08/2023	05/09/2023	12/09/2023	19/09/2023	26/09/2023	02/10/2023	09/10/2023	14/10/2023
Technical Design Document												
Create Levels												
Create Character & Boss												
Create Objects												
Create Animation												
VFX Development												
Create UI												
Shader Development												
Code Development												
Music & SoundEffect												
Testing												
Evaluation												
Create journal of design and implementation ideas												
Create Manual												
Remark						8.38-8.29 11 hours				9.30-10.2 Lunar date 04		

## •Feedback

I published this game on my itch.io page and other social medias, so this time I not only received feedback from my friends, but also received some feedbacks from the game lover from all over the world.

### Feedback from my friends :



Name : Carpe

Occupation : Game Designer

Feedback :

Although there are some small bugs in game, it is no denying that I have fun with this demo.

I love action design of the boss which is powerful and frighthening. I also like the scene design and the light effects he creates, accompanied by music, which brings the scene alive and atmospheric.



Name : Liu Jinghong (Joiz)

Occupation : Game Artist

Feedback :

The scene is comfortable and stylish. The mechanic of the boss is various and interesting. I am really impressed by the scene classroom. The interactive objects which could go up when player gets close to it is amazing. Also I think the idea of using bear to refer teacher and students is brilliant. I love this game.

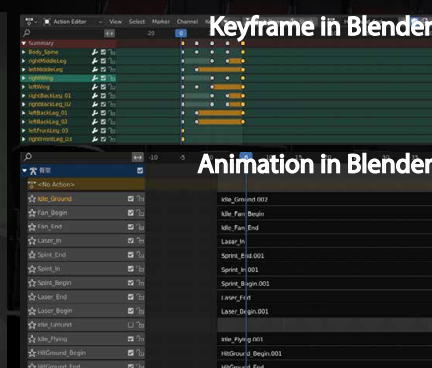
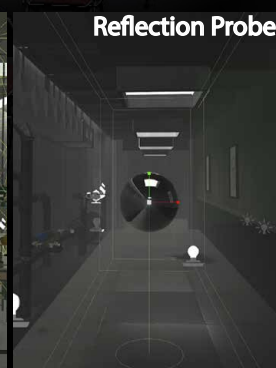
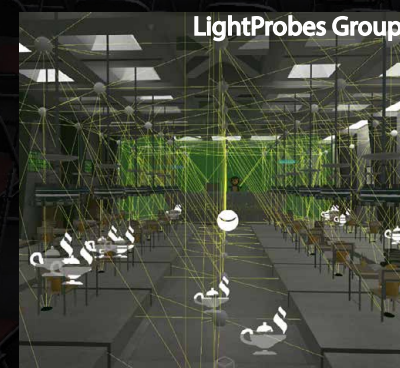
## •What I learn from this project

### VFX Graph & Lighting in HDRP :

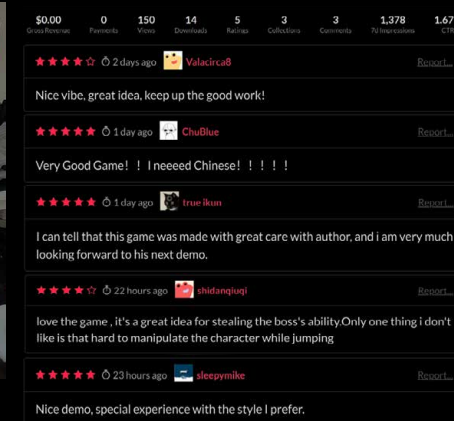
The project\_2 helps me get in touch with VFX\_Graph and lighting of HDRP. I learn a lot about light baking, probes and Vfx\_graph working with shader graph.

### Workflow Improvement :

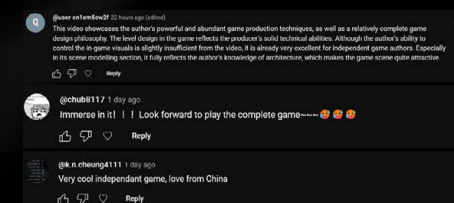
The project\_2 mainly requires knowledge of animation, rigging and Vfx\_Graph. After finishing this project, I know how to make animation in Blender with rigging and import it into Unity seamlessly.



### Feedback from itch.io :



### Feedback from youtube :



### Feedback from bilibili :



## •Problems in Game

### Nonstandard Modeling:

My modeling is not very standard. There are lots of faces overlapping with each other, which causes extra cost in the game engine. I spent plenty of time deleting the unnecessary faces of my models.

### Blender Statistics

[locker\_door.004 | Verts:118,320 | Faces:100,855 | Tris:191,943 | Objects:0/202 | 3.3.1

### Unity Inspector Statistics

138 Meshes  
138 meshes selected, 216744 total vertices, 566835 total indices

### Unity Game Statistics

Play Focused Mute Audio Stats Gizmos  
Statistics  
Audio (suspended):  
Level: -74.8 dB  
Clipping: 0.0%  
DSP load: 0.1%  
Stream load: 0.0%  
Graphics: - FPS (Playmode Off)  
CPU: main 31.2ms render thread 10.6ms  
Batches: 1444 Saved by batching: 0  
Tris: 863.9k Verts: 1.4M  
Screen: 1920x1080 - 23.7 MB  
SetPass calls: 288 Shadow casters: 1026  
Visible skinned meshes: 0  
Animation components playing: 0  
Animator components playing: 0

### Blender Statistics (Optimized)

Collection | Verts:65,160 | Faces:54,296 | Tris:101,012 | Objects:0/180 | 3.3.1

### Unity Inspector Statistics (Optimized)

90 Meshes  
90 meshes selected, 117219 total vertices, 194739 total indices

### Unity Game Statistics

Play Focused Mute Audio Stats Gizmos  
Statistics  
Audio (suspended):  
Level: -74.8 dB  
Clipping: 0.0%  
DSP load: 0.1%  
Stream load: 0.0%  
Graphics: - FPS (Playmode Off)  
CPU: main 17.0ms render thread 7.8ms  
Batches: 1146 Saved by batching: 0  
Tris: 355.0k Verts: 621.6k  
Screen: 1920x1080 - 23.7 MB  
SetPass calls: 322 Shadow casters: 779  
Visible skinned meshes: 0  
Animation components playing: 0  
Animator components playing: 0

### Some bugs on Movement :

There are still some small bugs in game. For example, if you flash too often, you would fly to the sky. This is because of the rigidbody.

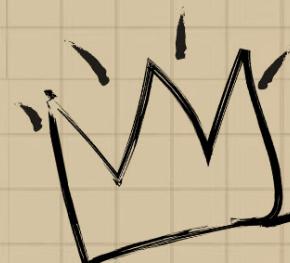
## •Conclusion & Future hope

I am quite satisfied with this project, because it is really a game that could be played and reach my initial expectation. Compared with the time I started to make my first project, I feel I had made a great progress.

I am also inspired by the feedback of game lovers from all over the world, which makes me happy create games.

In the future, I would definitely continue this game's development in my spare time. Because it still has a lot to dig out. For example, the destroyable wall(also could recover) has great potential possibility for level design, and it is also reasonable under the theme of nightmare.



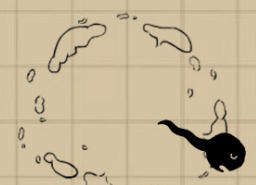
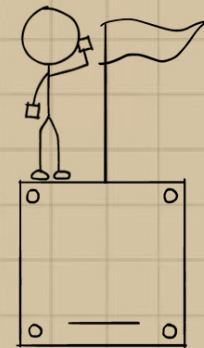


# DayDream,

Platformer

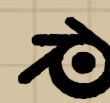
Atmospheric

Adventure



Developed by Devil\_ovo  
Available at: <https://devil-ovo.itch.io/daydream>  
Promotional Video(1min) at: <https://youtu.be/2cjd17iNVL0>  
Whole Process Video(4min) at: <https://youtu.be/FA1j6GXz9LU>

Made by





# Overview

## Basic Information

**Producer:** Jian Junren (Devil\_ovo)

**2DArt & Code:** Jian Junren (Devil\_ovo)

**Render Pipeline:** URP

**Duration:** 2023/11/01 – 2023/11/18

## Introduction

DayDream is a traditional platformer game. What player can do is to move, jump and avoid the adversary.

## Ambition

Though the game mechanics is simple and traditional, I still want to make some innovations on this game.

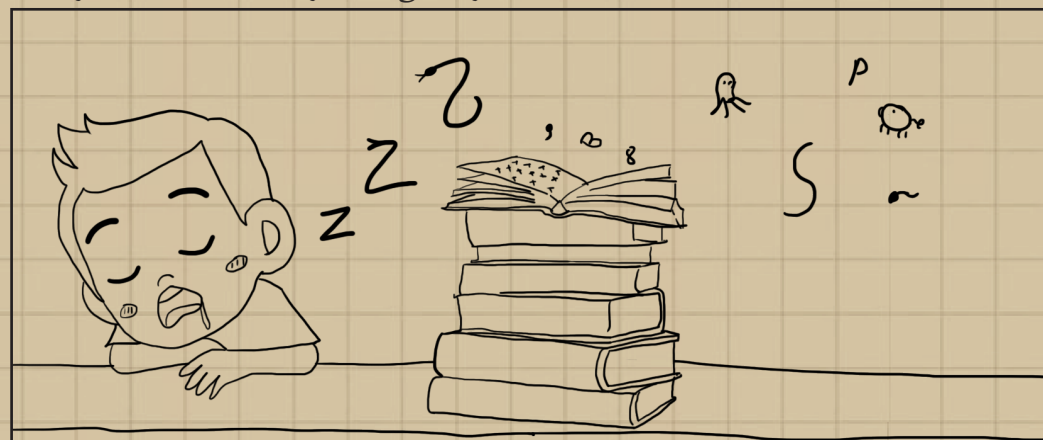
Finally, I decide to enhance the game by storytelling. I focus on 2DArt(especially animation design) and turn this game into 3D to give surprise to the players.

## Story Outline

You are a kid who dislikes study and have great imagination.

During the class, you feel tired and bored. You sit in your seat in a trance.

You made a yawn and fell asleep immediately. In your daydream, everything in your notebook comes to life.



# Inspiration

I believe that most of people felt dizzy and bored when they were in primary school. I often killed my time in class by doing sketch on my textbook.

Sometime when I stare at my textbook for a long time, I felt all the alphabet came to life with my wild imagination. Sometimes a letter H can be considered as a ladder or a train track.

I also get inspired by the movie Taare Zameen Par below which reminded me of the experience in my childhood.

# Reference

## Story

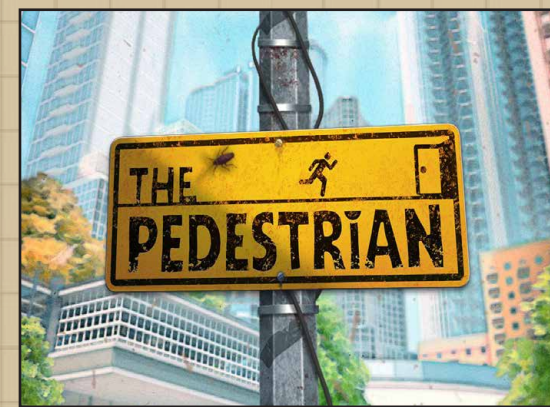


Taare Zameen Par



Superliminal

## Art Style



Pedestrian



Word Game

# Game Mechanics

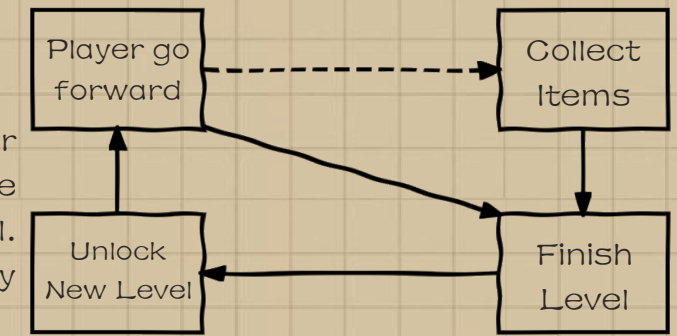
## Game Control & Loop

**Control:**

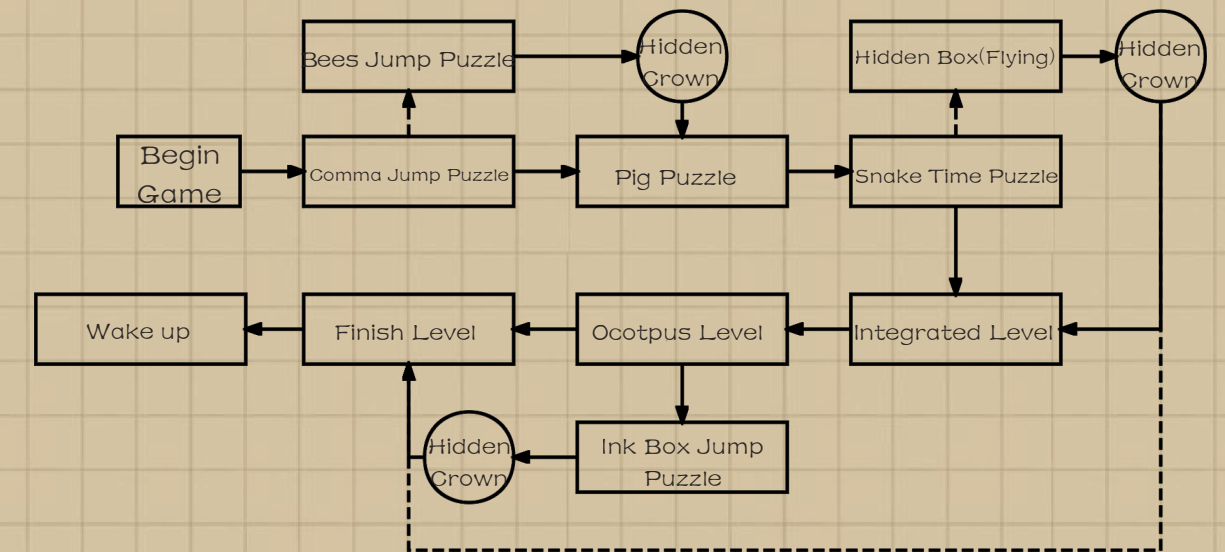
AD—Move Space—Jump

**Loop:**

The game loop is simple. Player try their best to arrive the destination to finish the level. Of course, if they want, they could collect all the crowns.

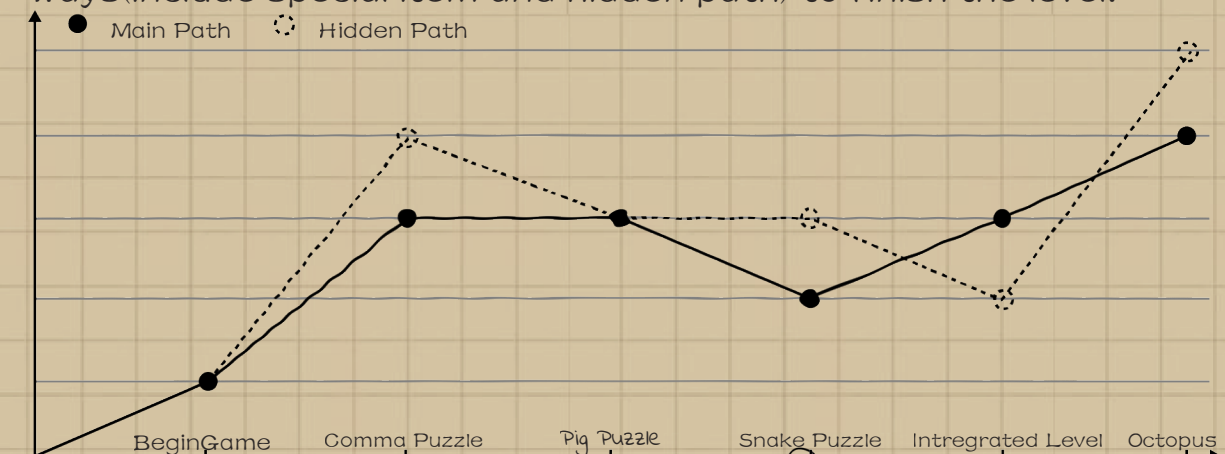


## Game Progress



## Difficulty Curve

The difficulty curve could be different if player choose different ways(include special item and hidden path) to finish the level.







# Sketches

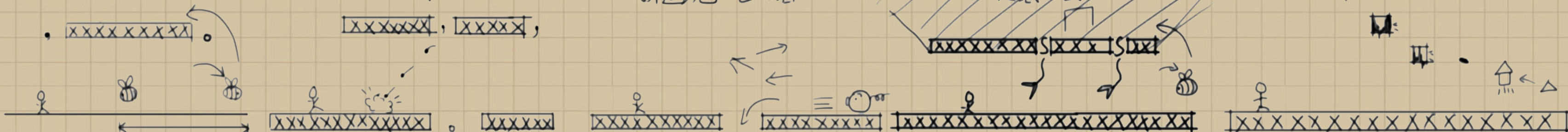
Bee

蝌蚪 (边跳边游)

猪 折返跳 甩开猪

蛇 (时间+隐藏地图)

贼 (隐藏关卡)

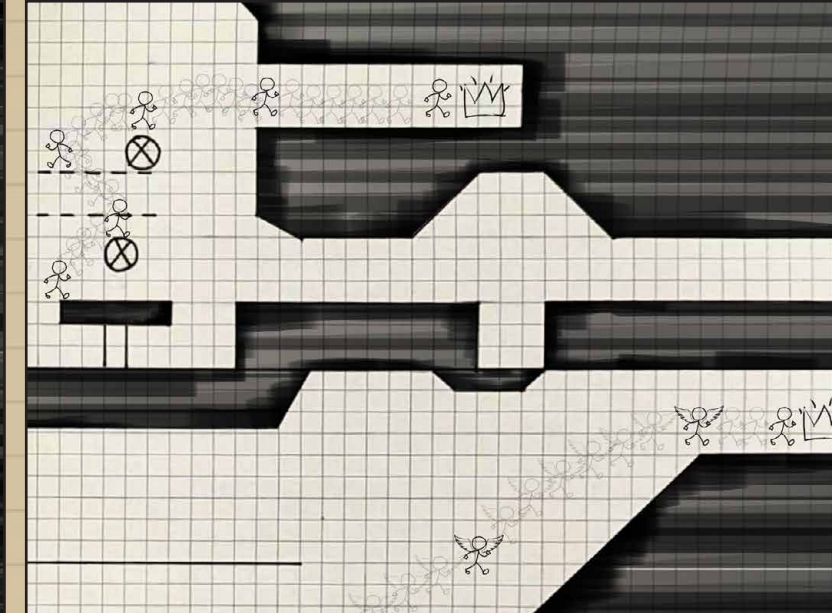
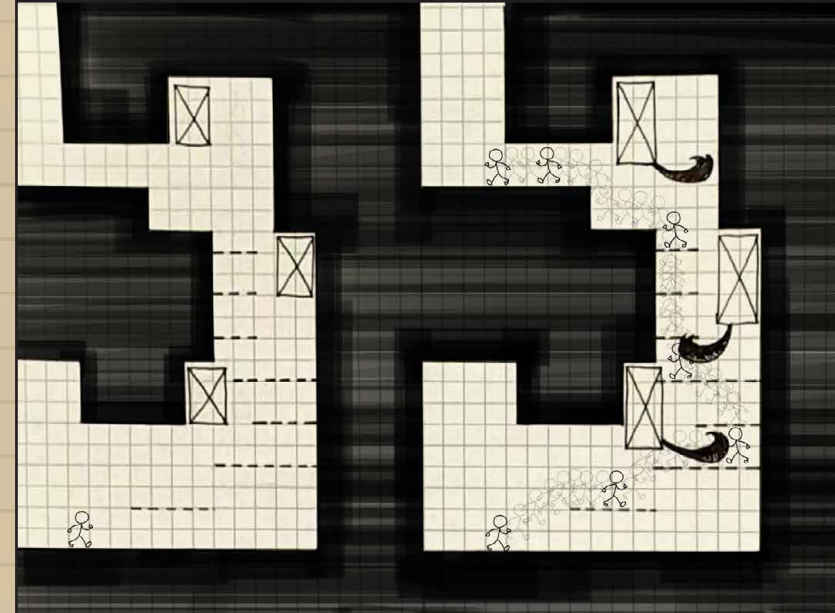
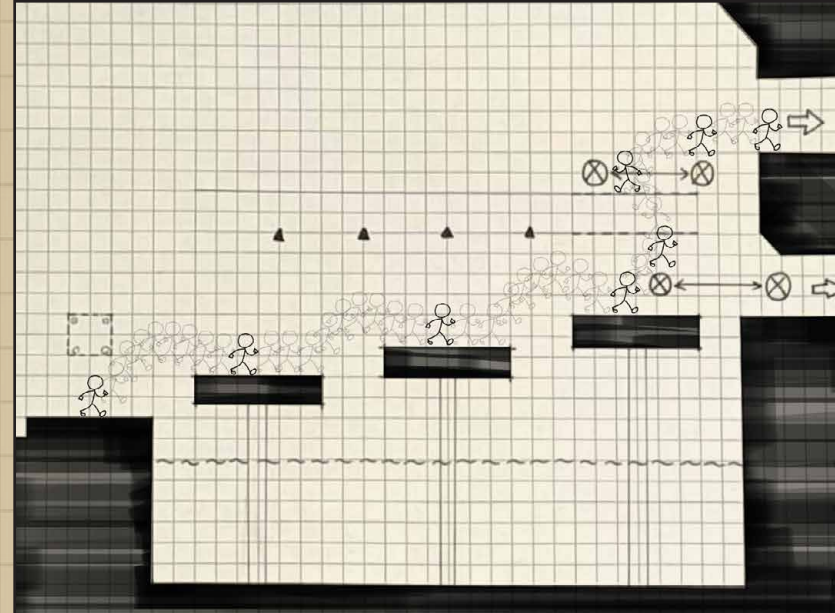
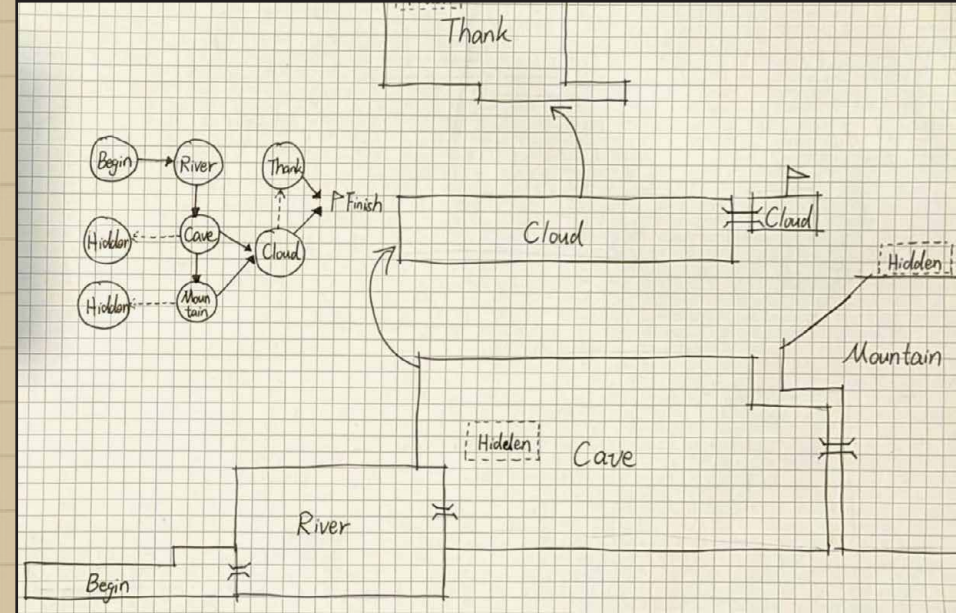


Level mindmap

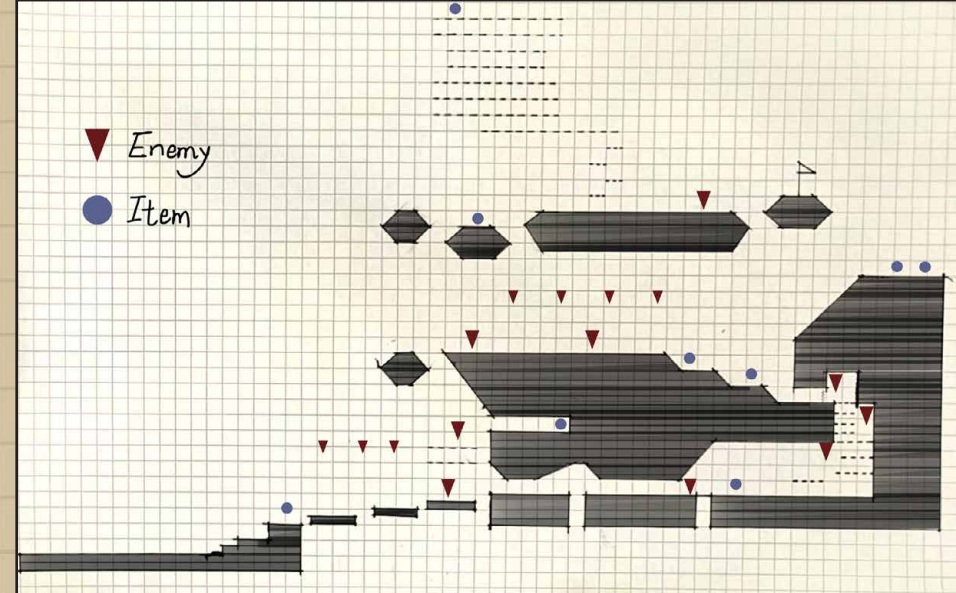
Level Unit

Level Unit

Hidden Space Unit



Enemy & Items distribution map





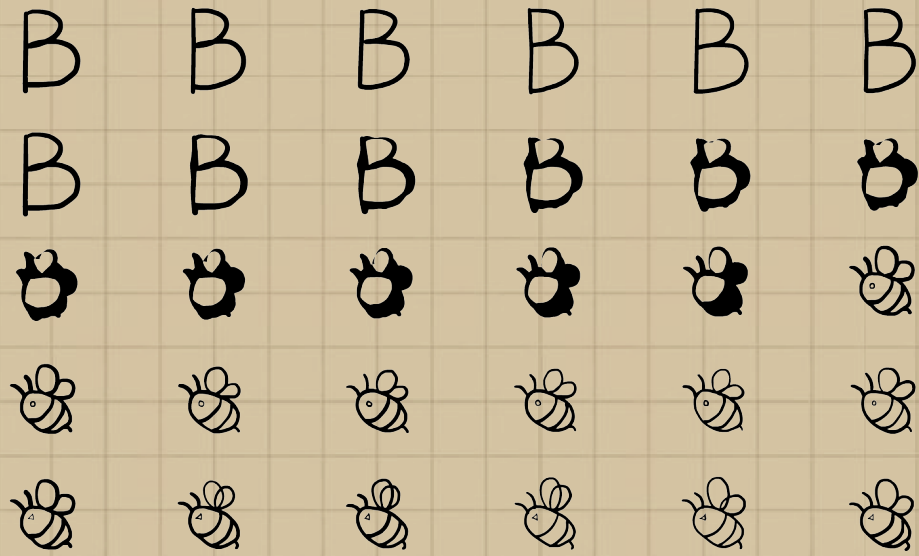




# Enemy Design

## • Bee Danger: ★

A dumb enemy that only patrol between two locations.



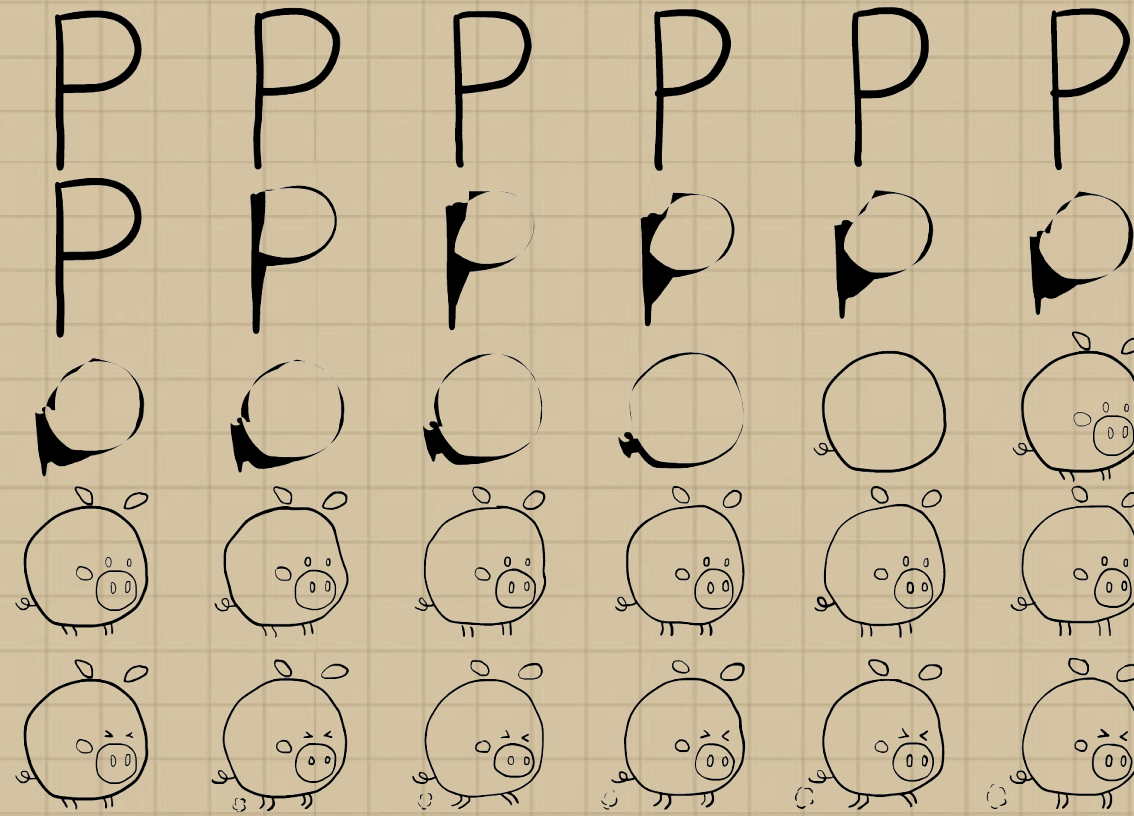
## • Comma Danger: ★★

A kind of enemy that only appears and attack when player comes close. After attacking, it would die.



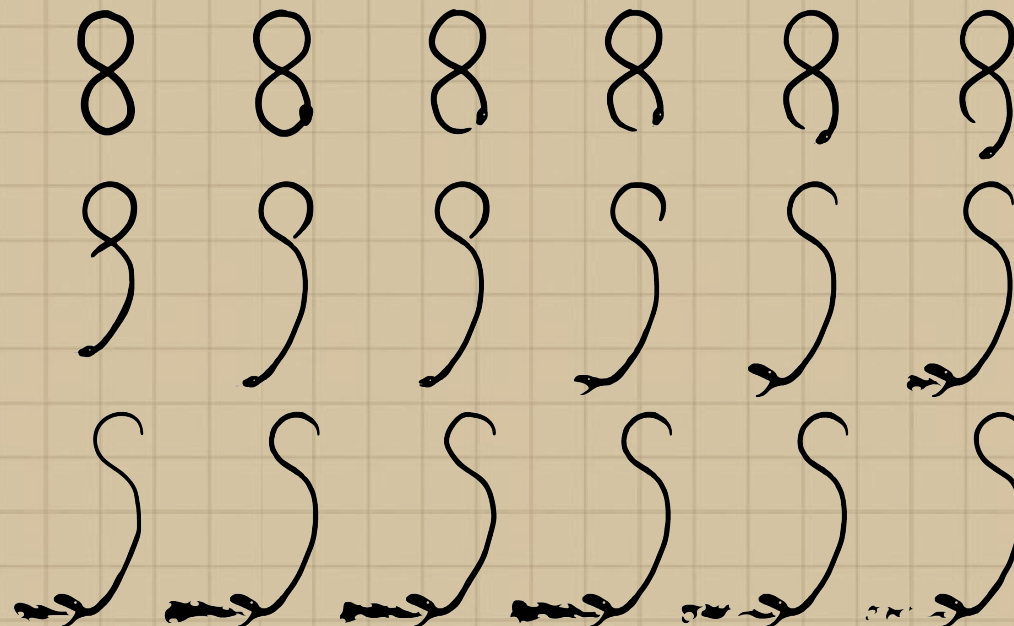
## • Pig Danger: ★★★

A kind of enemy that only appears and attack when player comes close. After its appearance, it would chase player to the end of the time.



## • Snake Danger: ★★

A kind of enemy that only appears and emits fire when player comes close. It is invincible.



## • Octopus Danger: ★★★

A kind of enemy that only appears and emits dangerous ink bubbles when player comes close.

Ps: Bubble could make the hidden block become visible.

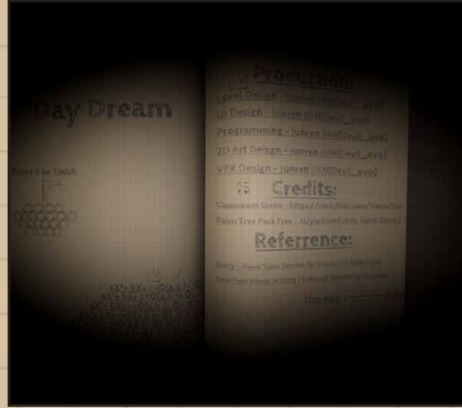




# + Visual Effect Design

## • Blinking Effect(Post-processing)

Vignette Effect



Blur Effect

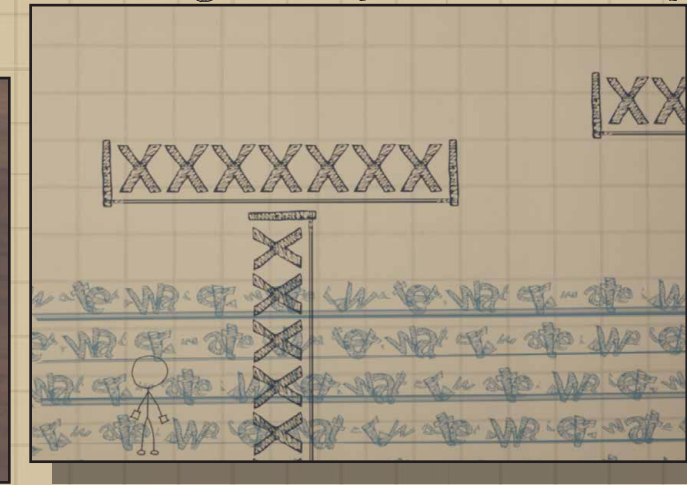


### Core Code :

```
private IEnumerator BlinkAndDark()
{
    float t = 0;
    while (t <= 1)
    {
        _vignette.intensity.Override(t);
        t += Time.deltaTime;
        yield return null;
    }
    t = 1;
    while (t >= 0)
    {
        _vignette.intensity.Override(t);
        t -= Time.deltaTime;
        yield return null;
    }
}
```

```
private IEnumerator Blurring()
{
    float t = 1;
    while (t >= 0)
    {
        _depthOfField.gaussianEnd.Override(t * 3);
        t -= Time.deltaTime;
        yield return null;
    }
    t = 0;
    _depthOfField.gaussianEnd.Override(0);
    while (t <= 1)
    {
        _depthOfField.gaussianEnd.Override(t * 3);
        t += Time.deltaTime;
        yield return null;
    }
    _depthOfField.gaussianEnd.Override(3);
}
```

## • Moving Texts(TextMeshPro)



### Core Code :

```
public class MovingFonts : MonoBehaviour
{
    public Vector2 speed;
    public Vector2 strength;
    private TMP_Text textMesh;
    private Mesh mesh;
    private Vector3[] vertices;

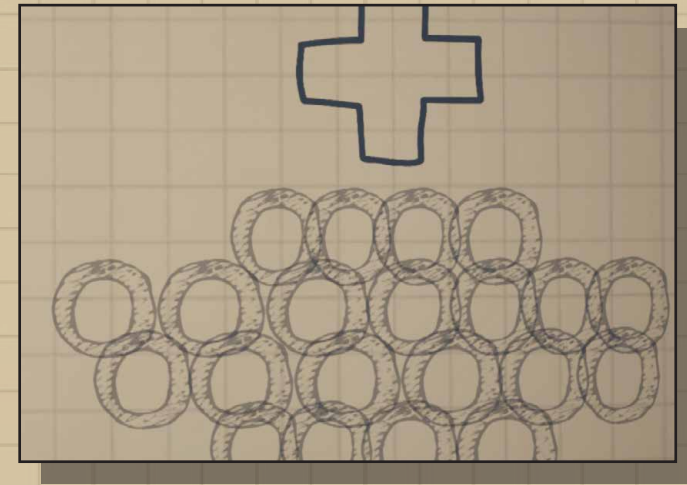
    void Start()
    {
        textMesh = this.GetComponent<TMP_Text>();
    }

    void Update()
    {
        textMesh.ForceMeshUpdate();
        mesh = textMesh.mesh;
        vertices = mesh.vertices;

        for (int i = 0; i < vertices.Length; i++)
        {
            Vector3 offset = Moving(Time.time + i);
            vertices[i] = vertices[i] + offset;
        }

        mesh.vertices = vertices;
        textMesh.canvasRenderer.SetMesh(mesh);
    }

    1 个引用
    private Vector2 Moving(float time)
    {
        return new Vector2(Mathf.Sin(time * speed.x) * strength.x, Mathf.Cos(time * speed.y) * strength.y);
    }
}
```



## Feedback

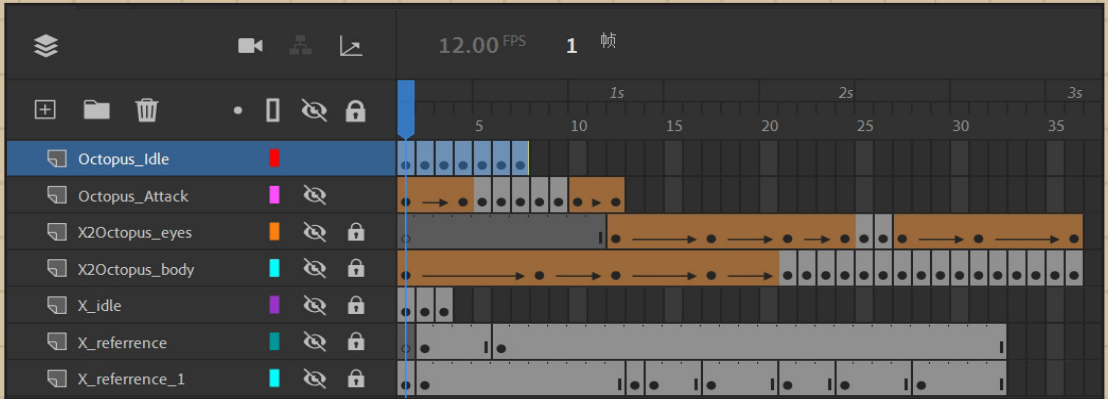


**Name :** Huang Yuxuan (Joiz)  
**Occupation :** Technical Artist  
**Feedback :**  
 The topic of the game is fantastic! And the 2D Art is quite which is suitable for the daydream. I'm also interested in the text visual effects that bring life to the scene. What impressed me most is the last scene that players realizes that they actually play the game on the notebook. This plot design is my favorite.

## What I learn from it

### • 2D Animation(Adobe Animate)

From this project, I learn how to use shape interpolation(in Adobe Animate) to create sequence animation.



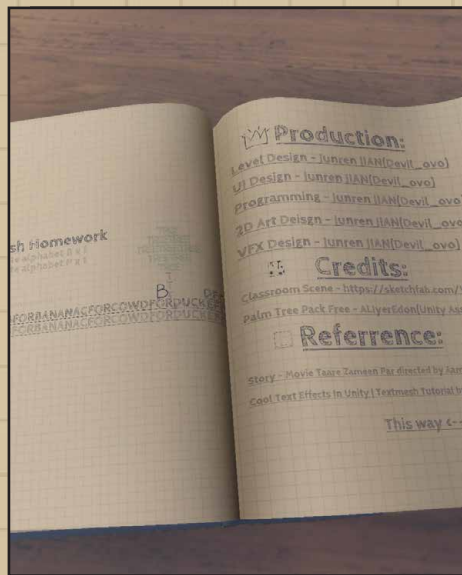
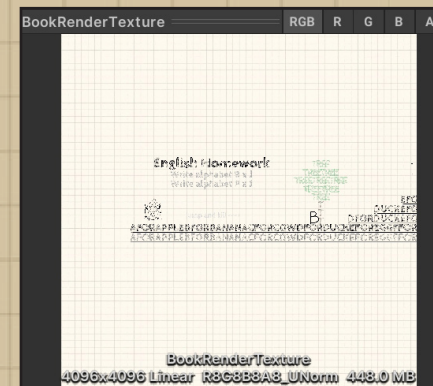
## • 2D --> 3D Scene(RenderTexture)

3DBook ----> 2DScene

### Method:

Extra Camera ----> RenderTexture

RenderTexture ----> MainTexture



## Shortages & Future

DayDream as the topic still has lots of possibility for me to discover and develop, such as from the real world elements, which include children use their shadow of hands to imitate animals that could be a boss. This can bring possibility on mechanics and make interaction between the real world(3D) and the dream world(2D). I'm sure that it could provide players with the brand-new experience and raise their imagination.

I also want to create cool effects such as a vortex made up of words.



# GameJams !!!

## HERE'S UR PACKAGE

### BASIC INFORMATION

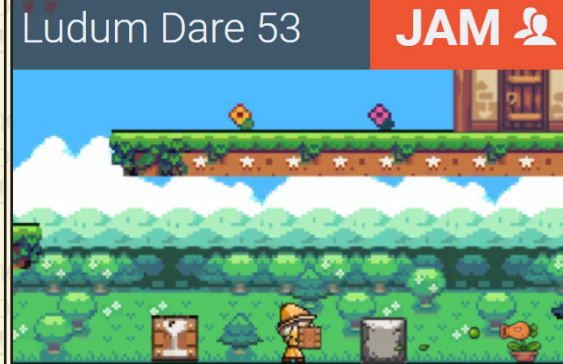
Date : April 29th – May 2nd

Event : Ludum Dare 53

Team Members :

Jian Junren (Devil_ovo) <b>PROGRAMMER</b>	Wang Peiran (sansen54) <b>LEVEL DESIGN</b> <b>ART DESIGN</b>
---	---

Available to Play (Web):  
<https://sansen54.itch.io/heres-ur-package>



Here's ur Package

### OVERVIEW

Tag : Platformer

Intro : In this game, your only goal is to deliver packages to the guests, though there are many difficulties you need to face on the roads.

### MECHANICS DESIGN

<b>CARGO</b>	<b>DIFFERENT ATTRIBUTE</b>	Help player go through enemy and obstacle
	<b>POINT</b>	Gain more point for achievement
made of rock, indestructible.	⚡	you need to deliver it in time.
when on fire or broken, it will explode and destroy boxes nearby.	+	use it to heal yourself
a wooden box with nothing in it, it's a bit sturdy.	🍄	use it to trap enemies
very fragile, 1 damage can destroy it, take care of it.	🔮	it can help you jump much higher or rebound bullets from enemies.

### FEEDBACK FROM OTHERS

by [LegoProjects](#) (submitted 5 months ago)

Cute game! It was weird that I was required to drop everything before the finish level button would work and some of the hitboxes were a bit sticky but the art style was really cute! I really enjoyed playing this game and the idea of picking up and being able to drop items in front of you has some cool implications

by [Kara](#) (submitted 3 months ago)

Really cool game idea! Took me some time until I figured out how to place the parcel so that I can enter the house, though. But neverthless, really cool game!

For me it's not quite clear which graphics come from you and which not. I think I recognize at least two more assets besides the one that you linked in the description. I think it would be good to make that more clear. But regardless of what comes from you and what doesn't the world looks very nice. I really like the pixel art and your little character fits perfectly into that world.

by [NerdCat](#) (submitted 3 months ago)

Great jumping puzzle game! Found many easter eggs in it. It's a clever idea to give different boxes different attributes. This game has a lot of potential!

by [Saidness](#) (submitted 5 months ago)

The artwork and music was great. I loved playing with the theme. It was an interesting take for the gameplay.

**GAMEJAM 1 / 3**

# If you are UI

### Basic Information

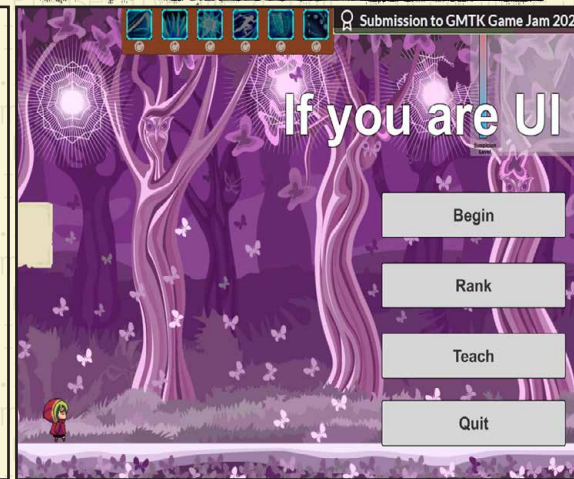
Date : July 8th – July 10th

Event : GMTK 2023

Team Members :

Jian Junren (Devil_ovo) <b>Planning</b> <b>Programmer</b>	Wang Peiran (sansen54) <b>Planning</b> <b>Monster Design</b> <b>Skill Design</b>
--	--

Available to Play (Web):  
<https://sansen54.itch.io/if-you-are-ui>



### Overview

Tag : Platformer

Intro : In this game, you will not play the role of "Player", instead, you will control the "UI" of the "Game", and your aim is to help the "Player" to beat the enemies and get higher goal.

### Mechanics Design

**Control UI**

**Help Player**

**Player Dead?** Yes/No

**Player Suspect?** Yes/No

**Game Over**

**What you can do? — Press Button Of UI**

You could highlight the skill to tell "player" to use it. Also you could refresh the CD of skill.

When "player" is close to death you could add HP to him. And you can add point secretly, too.

Each enemy has the corresponding skill to cope with.

**How to get feedback from "Player"**

Player(UI) could see the input of Player(Game), and his suspicion level. If suspicion level is too high, game is over. So try to be careful to cheat.

Player(UI) could see what player(Game) is thinking when you do something. For example, if you give correct tips, player(Game) will be thankful.

### Feedback From Others

by [TeamCarrot](#) (submitted 94 days ago)

Amazing artwork! Even if not selfmade it is impressive. Nice idea but I found the game a little bit complicated and confusing. Managed some levels but I still wasn't all sure about what I was doing. Still I enjoyed it.

by [LosenArc](#) (submitted 97 days ago)

Creative game!  
I like how the player talks to you and gets suspicious!  
I was a bit confused with the skills but I still had fun!

by [HolySpiritFire](#) (submitted 90 days ago)

Fun game, wish I could have helped the player more to block those arrows. Dang, couldn't make it past the Rock Monster though and I tried a few times!

by [Sailing Rocks](#) (submitted 99 days ago)

Good job! I really like the idea and the way you put it to action. Never seen nothing like that employed it.

by [LIISova](#) (submitted 99 days ago)

My team had the same idea as you, though our game requires you to help the "player" by taking charge of all of the games' systems instead of cheating in favour of the "player". You've created a very fun spin of the same general idea, it's very neat, thou a little overwhelming at the start with all the buttons

**GameJam 2 / 3**

# Animal Merchant

### Basic Information

Date : September 30th – October 2nd

Event : Ludum Dare 54

Team Members :

Jian Junren (Devil_ovo) <b>Planning</b> <b>Programmer</b> <b>Animation Design</b>	Wang Peiran (sansen54) <b>Planning</b> <b>Art Design</b> <b>Level Design</b>
---	--

Available to Play (Web):  
<https://devil-ovo.itch.io/animal-merchant>



### Overview

Tag : Puzzle Strategy

Intro : In this game, you are an animal merchant selling animals. You have to pay attention to the number of animals which keep reproducing to make sure your raft will not sink.

### Mechanics Design

**Transfer Animals** → **Control Numbers** → **Raft sink?** (Yes/No) → **Mission Failed**

**Number Enough?** (Yes/No) → **Mission Failed**

**Manupulation**

**Food Chain Design**

Animals will breed when there're two, but a single mushroom can also breed.

Bear and Wolf will die in Mushroom raft because of poisoning.

Animals and mushroom will reproduce every 15 seconds, while chicken breed every 20 seconds.

You can try to take one's natural enemies to their raft to reduce the number.

### Feedback

by [LegoProjects](#) (submitted 5 months ago)

Takes not so much time to understand how to play, good game.

by [Miles](#) (submitted 2 months ago)

Cool graphics! The idea is simple but makes sense. I fell into the water at some point and there was no way to get out. (Other than that pretty cool game!)

by [Miles](#) (submitted 2 months ago)

Nice game! Maybe the main character should walk faster to make it a bit easier, or there should be another way to transport animals anyways good take on the theme!

by [LegoProjects](#) (submitted 5 months ago)

The game was too difficult for me (not in a fun way), I wasn't able to finish any mission successfully. Maybe it's just me, but maybe the balance is a little bit off.

Nice game! Maybe the main character should walk faster to make it a bit easier, or there should be another way to transport animals anyways good take on the theme!

by [LegoProjects](#) (submitted 5 months ago)

I liked the low-poly graphics!

**GameJam 3 / 3**