

The other side

2 PLAYERS

Programming：Chenyang Xu
Level Design：Shuncheng Yu
Art：Xiale Liu
Story：Xinyan He

Summary

Emma felt lonely and hoped to forget her identity and enter the world of the book to experience another life, perhaps the gods heard her prayer, and the next day she woke up in the fantasy world of the book, but because she missed her cat, she was eager to find a way to return to reality. Kitten Naomi finds Emma unconscious and into a deep sleep, while the room is distorted and alienated rats appear. In the midst of an unknown panic, it is attracted by a glowing book.....

KeyWords

[Reality and fantasy] [pets] [humans] [help each other]

VideoLink: https://youtu.be/YThDNwfK9Y4

Made with

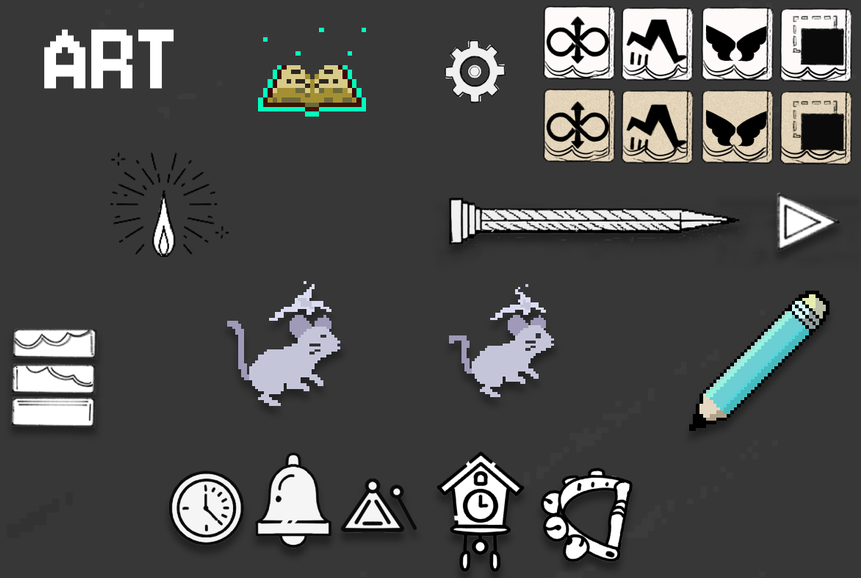
Inspirations





It takes two is the biggest inspiration for our game, our game learns the local split-screen gameplay in It takes two, this gameplay is very challenging, the most important of which is how to ensure that the player cannot play alone, but must rely on the cooperation of two people. The two-person trip gives a good answer, and continuous two-way positive feedback is a necessary element of two-person cooperation. But our game is also a little different from It takes two, it is a 3D game, but our game is 2D, and one is topdown, and the other is 2D horizontal board jumping

ART

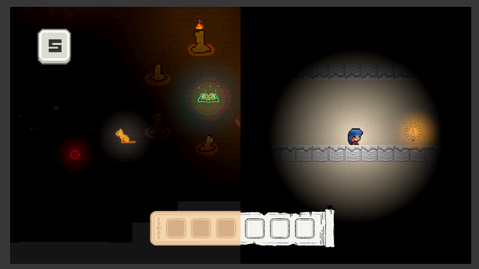


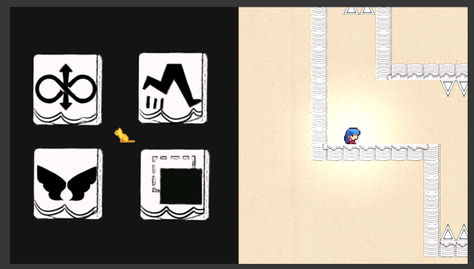
Character showcase

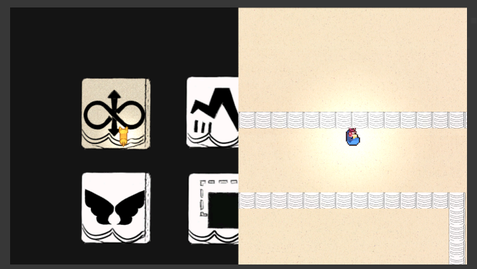




Presentation

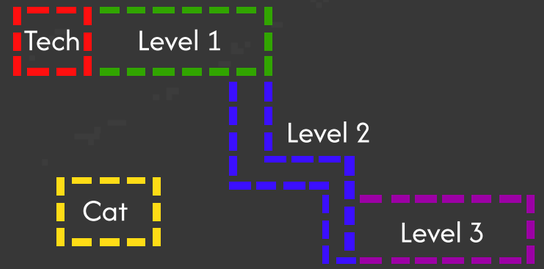








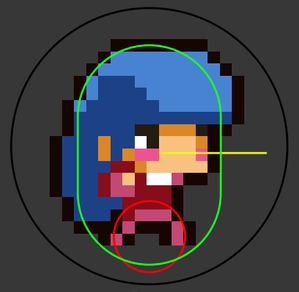
Level Overview



There are four levels here, the first of which is the tutorial level, each level has its own theme, the first level is light, the second level is sound, the third level is physics, each level has puzzles corresponding to its theme, and the cat and the person need to solve the problem together

Input and Actions

Player1

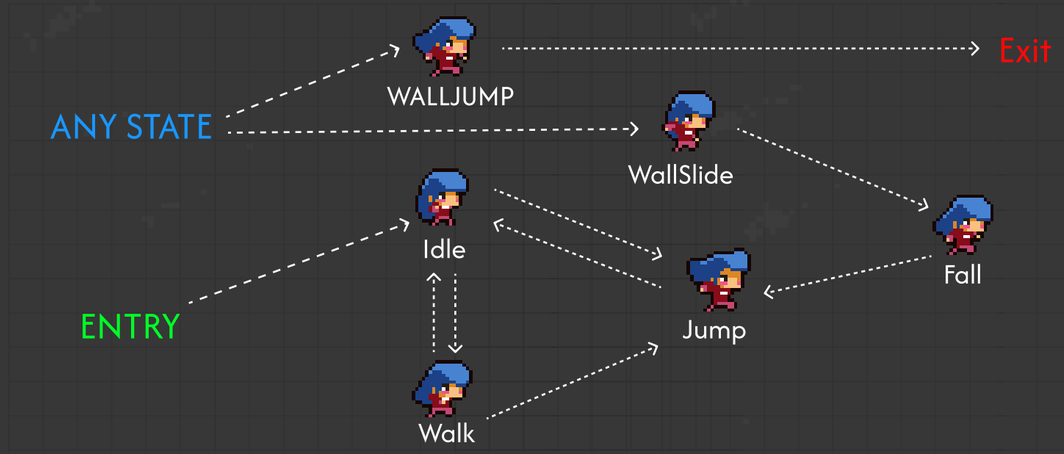


Trigger Colliders

Physical Colliders

WallCheck

GroundCheck

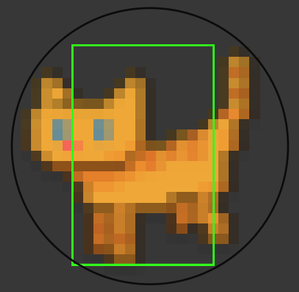


<SCRIPT>

PlatformerController.cs

Includes basic movement system, dash system, wall jump system, animation state management

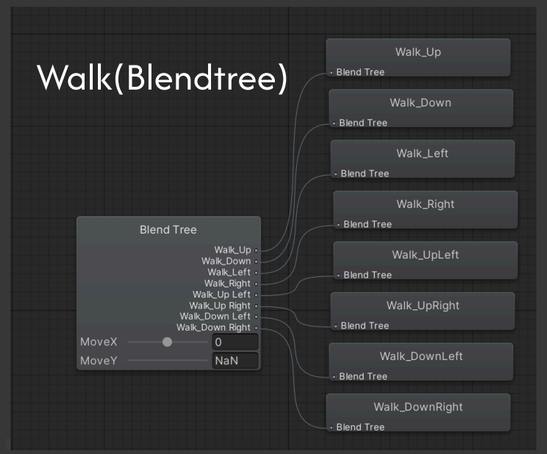
Player2



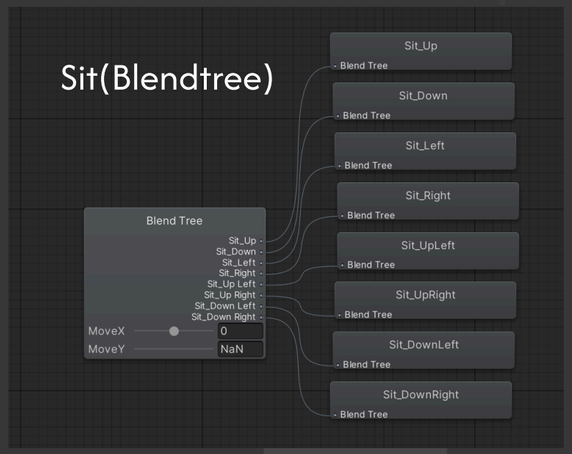
Physical Colliders

Trigger Colliders

Walk(Blendtree)



Sit(Blendtree)



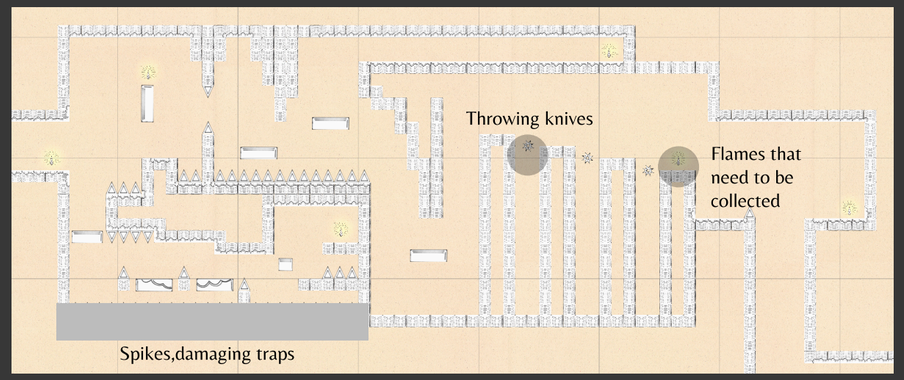
<SCRIPT>

TopDown2DController.cs

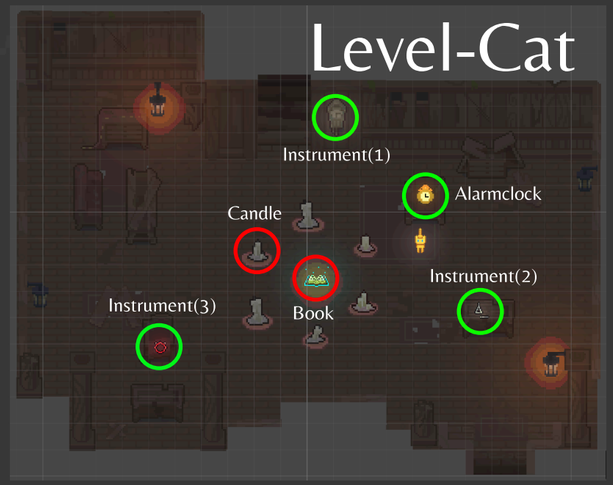
Includes eight-way movement system, animation state management, and parameter design

Level Design

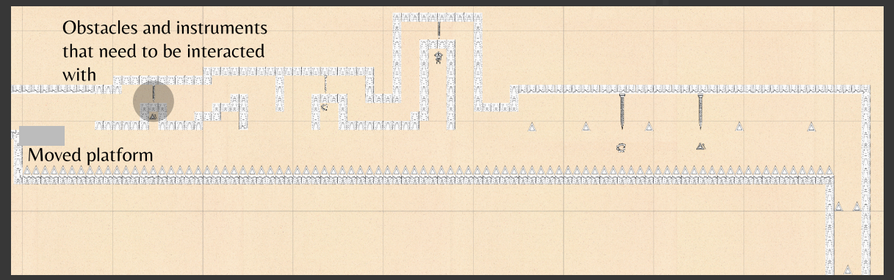
Level1



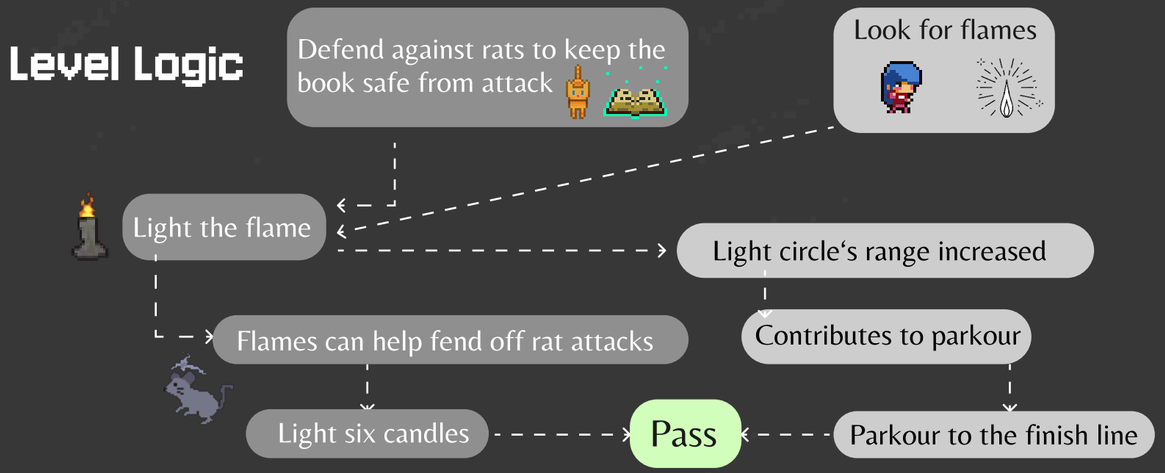
Level-Cat



Level2



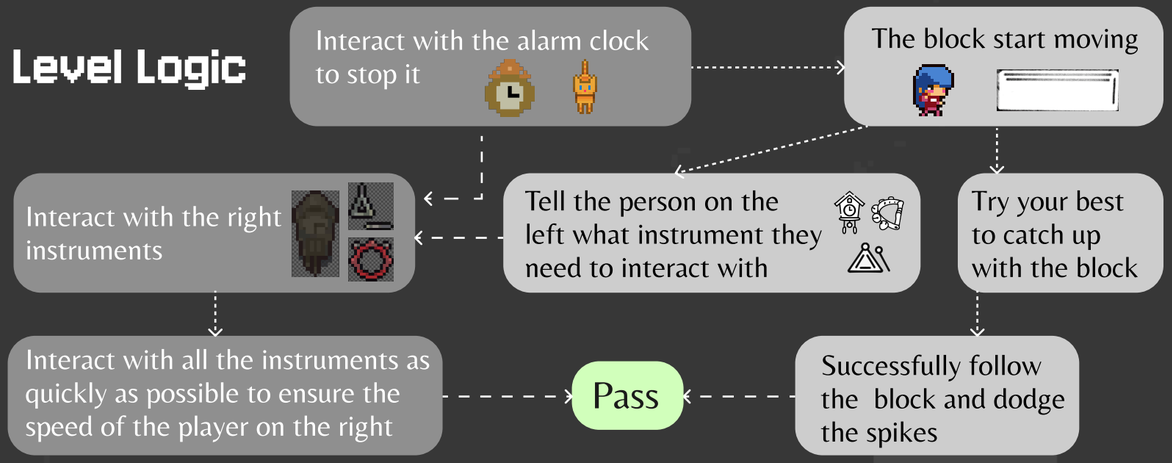
Level Logic



<SCRIPT>

CandleInteraction.cs-Collect the flames
DraggableUIItem.cs-Management drags and drops flames in the UI
FlameIcon.cs-The nature of the flame itself
GameManager.cs-Handle all events
MonsterController.cs-Monster interaction with the book and the player
MonsterManager.cs-Monster spawn
PlayerFlameCollector.cs-Light the flame

Level Logic

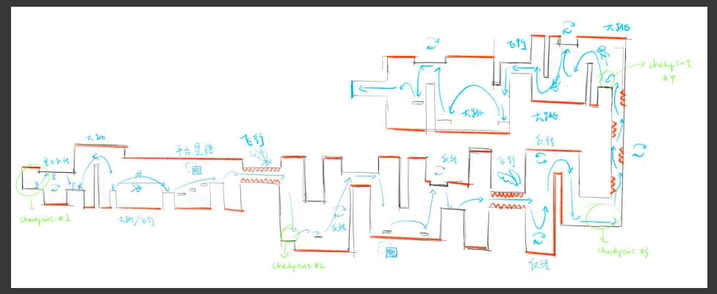


<SCRIPT>

AlarmClock.cs-Control the alarm clock and the activation of the block on the right
CatController.cs-Interaction between cats and musical instruments
CubeController.cs-Control the movement trajectory of the blocks
Instrument.cs-The nature of the instrument itself
SpikePathSystem.cs-Manage instrument type identification and obstacle removal

Level Design

Level3





Level Logic

Player cat: Step on four boards

Player Girl: Gain buff

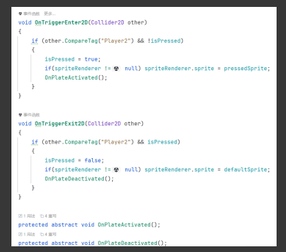
They combine the buff effect to complete parkour

Pass

<SCRIPT>

Base Class: PressurePlateBase.cs
(Inherited from MonoBehaviour)

inheritance



```
void OnTriggerEnter2D(Collider2D other)
{
    if (other.CompareTag("Player2") && !isPressed)
    {
        isPressed = true;
        if(spriteRenderer != null) spriteRenderer.sprite = pressedSprite;
        OnPlateActivated();
    }
}

void OnTriggerExit2D(Collider2D other)
{
    if (other.CompareTag("Player2") && isPressed)
    {
        isPressed = false;
        if(spriteRenderer != null) spriteRenderer.sprite = defaultSprite;
        OnPlateDeactivated();
    }
}

protected abstract void OnPlateActivated();
protected abstract void OnPlateDeactivated();
```

```
protected override void OnPlateActivated()
{
    SetPlatformsVisibility( visible: true);
}

protected override void OnPlateDeactivated()
{
    SetPlatformsVisibility( visible: false);
}
```

PlatformTogglePlate.cs

```
protected override void OnPlateActivated()
{
    playerController.jumpForce = originalJumpForce * jumpForceMultiplier;
}

protected override void OnPlateDeactivated()
{
    playerController.jumpForce = originalJumpForce;
}
```

JumpBoostPlate.cs

```
protected override void OnPlateActivated()
{
    playerRb.gravityScale *= -1;
    player.transform.localScale = new Vector3(
        originalScale.x,
        -originalScale.y,
        originalScale.z
    );
}

protected override void OnPlateDeactivated()
{
    playerRb.gravityScale *= -1;
    player.transform.localScale = originalScale;
}
```

GravityPlate.cs

```
protected override void OnPlateActivated()
{
    playerController.dashDuration = originalDashDuration * dashDurationMultiplier;
}

protected override void OnPlateDeactivated()
{
    playerController.dashDuration = originalDashDuration;
}
```

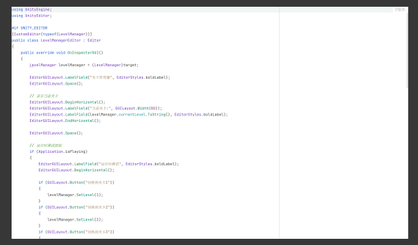
FlyPlate.cs

LevelController

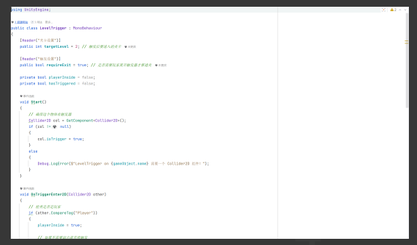
Since my levels are coherent and they are all in the same scene, I need to set up a Manager to manage the enablement and disabling of items in the level

1.LevelSystem

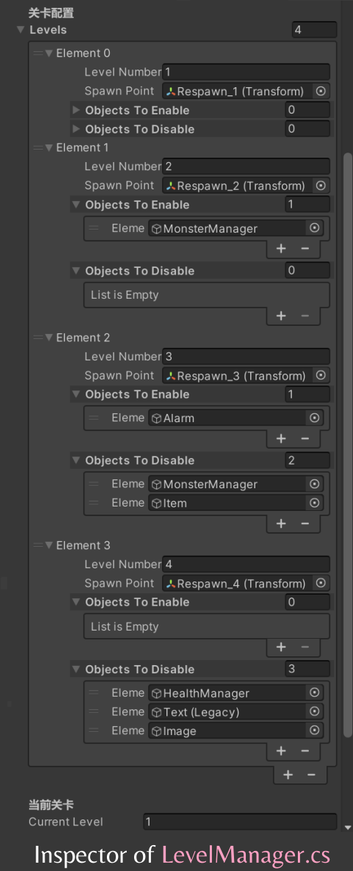
LevelManager.cs and LevelSystemEditor.cs (the Editor script inherited from the Editor), LevelTrigger.cs three scripts interact with each other



LevelSystemEditor.cs



LevelTrigger.cs



Inspector of LevelManager.cs

2.HealthSystem

Only the first and second levels need to manage the amount of blood, and they involve the conditions for switching levels, so the management of the health volume must be used in conjunction with the switching of the first and second levels



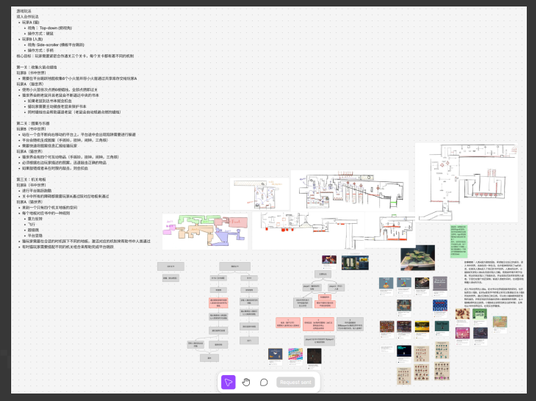
HealthManager.cs

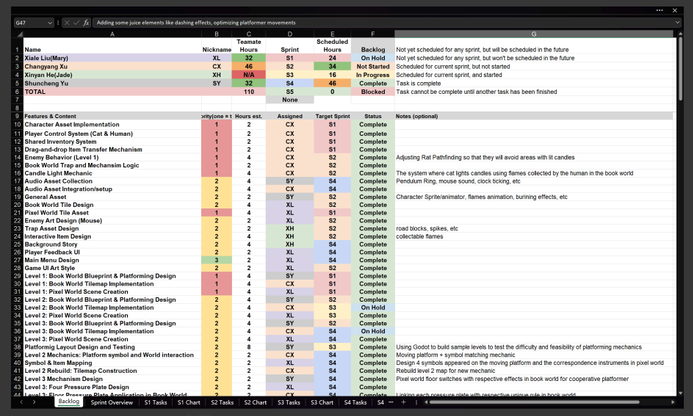


SpecialLevelEditor.cs(Editor file)

Team Work

I was mainly responsible for writing the code and building the entire engineering file on the team, and my teammates helped me with level design and initial testing, art, and backstory





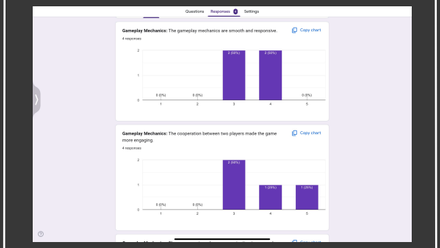
Name	Nickname	Teamate Hours	Sprint	Scheduled Hours	Backlog	Notes (optional)
Xiale Liu(Mary)	XL	32	S1	24	On Hold	Not yet scheduled for any sprint, but won't be scheduled in the future
Changyang Xu	CX	46	S2	34	Not Started	Scheduled for current sprint, but not started
Xinyan He(Jade)	XH	N/A	S3	16	In Progress	Scheduled for current sprint, and started
Shuncheng Yu	SY	32	S4	46	Complete	Task is complete
TOTAL		110	S5	0	Blocked	Task cannot be complete until another task has been finished
			None			

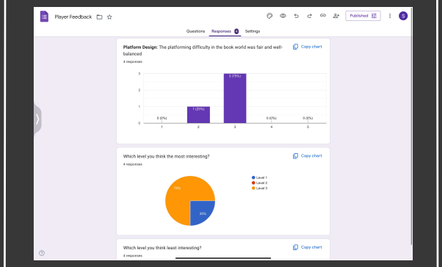
Features & Content	ority(one ★ t	Hours est.	Assigned	Target Sprint	Status	Notes (optional)
Character Asset Implementation	1	2	CX	S1	Complete	
Player Control System (Cat & Human)	1	2	CX	S1	Complete	
Shared Inventory System	1	2	CX	S1	Complete	
Drag-and-drop Item Transfer Mechanism	1	2	CX	S1	Complete	
Enemy Behavior (Level 1)	1	4	CX	S2	Complete	Adjusting Rat Pathfinding so that they will avoid areas with lit candles
Book World Trap and Mechanism Logic	1	2	CX	S2	Complete	
Candle Light Mechanic	1	4	CX	S2	Complete	The system where cat lights candles using flames collected by the human in the book world
Audio Asset Collection	2	4	SY	S4	Complete	Pendulum Ring, mouse sound, clock ticking, etc
Audio Asset Integration/setup	2	4	CX	S4	Complete	
General Asset	2	2	SY	S2	Complete	Character Sprite/animator, flames animation, burining effects, etc
Book World Tile Design	2	4	XL	S2	Complete	
Pixel World Tile Asset	1	4	XL	S1	Complete	
Enemy Art Design (Mouse)	2	2	XL	S2	Complete	
Trap Asset Design	2	2	XH	S2	Complete	road blocks, spikes, etc
Interactive Item Design	2	4	XH	S2	Complete	collectable flames
Background Story	2	4	XH	S4	Complete	
Player Feedback UI	2	2	XL	S4	Complete	
Main Menu Design	3	2	XL	S4	Complete	
Game UI Art Style	2	2	XL	S2	Complete	
Level 1: Book World Blueprint & Platforming Design	1	4	SY	S1	Complete	
Level 1: Book World Tilemap Implementation	1	4	CX	S1	Complete	
Level 1: Pixel World Scene Creation	1	4	XL	S1	Complete	
Level 2: Book World Blueprint & Platforming Design	2	4	SY	S2	Complete	
Level 2: Book World Tilemap Implementation	2	4	CX	S3	On Hold	
Level 2: Pixel World Scene Creation	2	4	XL	S3	Complete	
Level 3: Book World Blueprint & Platforming Design	2	4	SY	S2	Complete	
Level 3: Book World Tilemap Implementation	2	4	CX	S4	On Hold	
Level 3: Pixel World Scene Creation	2	4	XL	S4	Complete	
Platforming Layout Design and Testing	2	8	SY	S3	Complete	Using Godot to build sample levels to test the difficulty and feasibility of platforming mechanics
Level 2 Mechanics: Platform symbol and World interaction	2	2	CX	S4	Complete	Moving platform + symbol matching mechanic
Symbol & Item Mapping	2	2	XL	S4	Complete	Design 4 symbols appeared on the moving platform and the correspondence instruments in pixel world
Level 2 Rebuild: Tilemap Construction	2	2	CX	S4	Complete	Rebuild level 2 map for new mechanic
Level 3 Mechanism Design	2	2	SY	S4	Complete	Pixel world floor switches with respective effects in book world for cooperative platformer
Level 3: Four Pressure Plate Design	2	2	XL	S4	Complete	

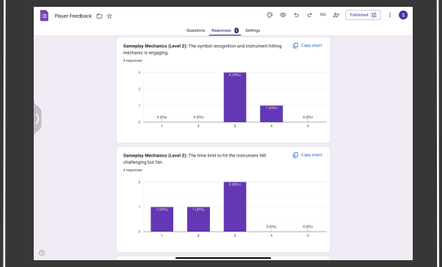
Problem

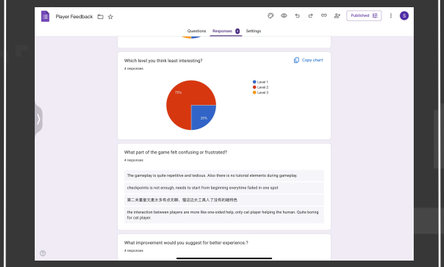
Problem found:
1.Lack of checkpoint
2.movement issues in book world character
3.no falling speed limitations on book world character
4.insufficient player interactions
5.repetitive gameplay for level 2
6.Lack of tutorial
7.the timing restrictions for hitting instrument is too loose

Feedback









Team Work

7.9: The first group discussion to determine the production of a two-player cooperative game
7.13: Referring to a lot of two-player games, we targeted the game Bomb Disposal Expert
7.16: The plan is initially formed
7.19:There was disagreement within the team that our previous game prototype was not perfect
7.22 A new game plan was developed
7.29 Determine all art and mechanism requirements and start development
8.4 The first level is preliminarily completed
8.14 The second and third levels are completed
8.20 sorted out the level switching logic

Future Plan

Future plan:
1.Add more checkpoints
2.Refine character movement
3.enhance player interaction (add book world elements that help the cat player such as switches that unlock abilities or new paths in cat's world)
4.improve feedback (add stronger visual and audio cues for key actions)
balancing difficulty