

# PROJECT 2 GAME DESIGN

# FANTASIA

By Kang.S

**Demo:** [Click here to download the demo](#)

**Video:** <https://youtu.be/6XTIUDXIYCE>

**Genre:** RPG, Strategy, Adventure

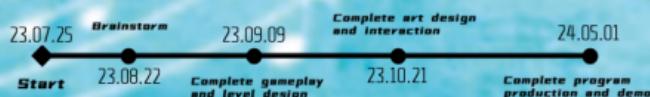
**Platform:** PC

**Develop Engine:** Unity 2021.3.34f1cl

**Control:** Mouse + Keyboard

**Created by** KANG SHUO

## Schedule



## 1 Introduction

This game is a suspenseful detective experience set in China around 2008, tackling the challenging social issue of child trafficking. Players take on the role of a young police officer partnered with a colleague to conduct "Operation Fantasia," a mission to expose and dismantle a child trafficking ring. As they solve a series of complex cases, players must navigate interwoven story lines, where each discovery brings new puzzles and dangers. Through this journey, players experience the pressures and sacrifices of those on the front lines.

With immersive storytelling, the game seeks to convey the reality of child trafficking—a pervasive threat even in modern society—while emphasizing the bravery and dedication of those who fight to protect the vulnerable. It invites players to reflect on crisis awareness and community safety, aiming to leave a lasting impact and inspire vigilance.

This narrative-driven approach, combined with atmospheric design and engaging gameplay, seeks to immerse players not only in the thrill of detective work but also in a profound social message.

## 2 Story Background

In 2008, amidst prosperity and underlying darkness, the world witnessed grand events, natural disasters, and harrowing human tragedies side by side. One seemingly ordinary day, a call of united desperation reached the Shangcheng City Police—a joint report from parents whose children had vanished. These parents had nowhere else to turn, and their pleas would soon unravel a chilling web of corruption and despair. The investigation begins with a single question: could these kidnappings be linked to a recent medical scandal involving tainted medication and child fatalities? As the detectives dig deeper, they are met with layers of deceit and horror, realizing that beneath the surface, a far-reaching darkness has been lurking all along.

## 3 Inspiration

This game draws inspiration from the atmospheric storytelling and emotional depth found in Chinese games like *Senfu* and *Firework*, as well as the open-world immersion and complex social dynamics seen in *GTA*. The narrative richness and tension in *Senfu* and *Firework* deeply influenced my approach to world-building and character-driven storytelling. I aim to create a similarly immersive environment that weaves cultural elements with a suspenseful, socially resonant story, reflecting the contrasting realities of China in 2008, where prosperity and societal darkness coexist.

From *GTA*, I was inspired by the concept of an open, dynamic world where players can engage with both action-driven gameplay and deeper social commentary. The freedom to explore and interact with the world around them mirrors my desire to provide players with opportunities to not only solve crimes but also reflect on the societal issues underlying the narrative, such as child trafficking and systemic corruption.

## 4 Reference

For structural and gameplay references, I looked to *Senfu* and *Firework* for their ability to blend cultural storytelling with gameplay mechanics that serve the narrative. These games provide a framework for balancing mystery-solving with emotionally impactful moments. In terms of gameplay mechanics and open-world exploration, *GTA* provided valuable insights into creating an interactive environment where every corner of the world offers something meaningful to the player's experience.

# PROJECT 2 FANTASIA

## 5 Game Dynamics

Players alternate between a police officer and an undercover agent, each uncovering clues from distinct perspectives. As the officer, players gather evidence through formal investigations, while as the agent, they gain covert intel from inside the trafficking ring, balancing risk with discovery. Each role influences the other, with choices made by one character shaping the other's investigation. This dynamic creates a layered, suspenseful experience, requiring strategic decisions to connect clues and uncover the truth.

## 6 Core Experience

Players navigate a high-stakes investigation by alternating between two characters: a police officer and an undercover agent within a child trafficking ring. Each character offers unique insights—while the officer uncovers clues through formal investigations, the undercover agent gathers intelligence from inside the criminal network. Switching perspectives allows players to piece together clues from contrasting worlds, making strategic choices to solve cases and unravel the truth. This dual-role approach provides a tense, immersive experience, blending detective work with the risks of undercover infiltration.

## 7 Game Mechanism

### Control



Move Left and Right



Jump



Pick up items, talk to characters and interact with interactive messages



Press V to switch perspective and the protagonist

The game features a dual-character mechanic where players can switch between a police officer and an undercover agent to progress the investigation. Each character has unique gameplay mechanics tailored to their roles:

### Police Officer:

Players engage in evidence collection, witness interrogation, and analysis of case files. A time management system challenges players to prioritize leads before they go cold, adding tension to the investigative process.

### Undercover Agent:

Players perform covert tasks such as gaining the trust of key members within the trafficking network, decoding hidden messages, and navigating dangerous situations without blowing their cover. Actions as the agent are constrained by limited resources and the constant threat of exposure.

The core mechanic lies in the interaction between the two roles. Information gathered by one character unlocks new possibilities for the other, requiring players to think strategically about when to switch roles. Additionally, a branching decision system ensures that player choices impact the story, leading to multiple possible outcomes.

## B Level Design

### Story

#### Prologue

1. Xia Zhi Learns about the child trafficking case and joins the task force
2. Zhu Lijiu becomes an informant for the task force and tries to get into the criminal gang

#### Chapter 1

1. Zhu Lijiu abducts children to gain the trust of the criminal gang
2. Xia Zhi helps with the abduction
3. Xia Zhi investigates the case

#### Chapter 2

1. Zhu Lijiu initially gets the criminal gang's trust
2. Xia Zhi connects the medical dispute with child trafficking

#### Chapter 3

1. Zhu Lijiu thought he was trusted, but the criminal gang suspected him and killed him
2. Xia Zhi led the team to the scene, but the criminal gang had already evacuated, leaving only the disabled child and Zhu Lijiu's body

#### Chapter 4

- Plot: Xia Zhi came to the abandoned factory and accidentally found the secret message left by Zhu Lijiu before he was killed. The rescued children said that there was a sister who was imprisoned here and took out candy from her pocket for them to eat.

### Main Characters

Policemen Xia Zhi  
Informant Zhu Lijiu  
Task force Leader Wu Xin  
Criminal investigation team leader  
Tian Yu

Zhu Lijiu  
Xia Zhi  
Criminal gang  
Wu Xin

Zhu Lijiu  
Xia Zhi  
Criminal gang  
Wang Youtian

Zhu Lijiu  
Xia Zhi  
Criminal gang  
Wang Youtian

Xia Zhi

# PROJECT 2 FANTASIA

## 9 Game Code

### # Backpack Item GUI (Mono Script)

```
Assembly Information
Filename: BackpackItemGUI.cs
Assembly: CSharp.dll

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class BackpackItemGUI : MonoBehaviour
{
    public Interactable interactable;
    public Dictionary<int, Info> foodBackpack = new Dictionary<int, Info>();
    public Dictionary<int, Info> equipmentBackpack = new Dictionary<int, Info>();
    public Dictionary<int, Info> implementBackpack = new Dictionary<int, Info>();

    private void Awake()
    {
        instance = this;
    }

    // Update is called once per frame
    void Update()
    {
        if (Start() == true)
        {
            Update();
        }
    }

    public void OnClick()
    {
        GUIManager.instance.itemName.text = GUIManager.instance.itemDescription;
    }
}
```

**Backpack Item GUI**

### # Interactable Manager (Mono Script)

```
Assembly Information
Filename: InteractableManager.cs
Assembly: CSharp.dll

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class InteractableManager : MonoBehaviour
{
    public static InteractableManager instance;
    public Dictionary<int, Info> foodBackpack = new Dictionary<int, Info>();
    public Dictionary<int, Info> equipmentBackpack = new Dictionary<int, Info>();
    public Dictionary<int, Info> implementBackpack = new Dictionary<int, Info>();

    private void Awake()
    {
        instance = this;
    }

    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        if (Start() == true)
        {
            Update();
        }
    }

    public void OnClick()
    {
        Dictionary<int, Interactable> itemDic = GUIManager.instance.itemDictionary;
        itemDic.Add(AddInteractableID, InteractableType.type);
    }
}

Dictionary<int, Backpack> switchInfo = new Dictionary<int, Backpack>()
{
    case InteractableType.Food:
        Backpack = FoodBackpack;
        break;
    case InteractableType.Information:
        Backpack = InformationBackpack;
}
```

**Interactable Manager**

### # Talk (Mono Script)

```
Assembly Information
Filename: Talk.cs
Assembly: CSharp.dll

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.Component;
using UnityEngine.SceneManagement;
using UnityEngine.SceneManagement;

public class Talk : MonoBehaviour
{
    public Transform talkPanel;
    public Text talkText;
    public GameObject talkPanel;
    public Text talkText;
    public void Start()
    {
        if (Start() == true)
        {
            Update();
        }
    }

    // Update is called once per frame
    void Update()
    {
        if (Start() == true)
        {
            Update();
        }
    }

    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Player"))
        {
            talkPanel.SetActive(true);
            talkText.text = "Hello Player";
        }
    }

    private void OnTriggerExit(Collider other)
    {
        if (other.CompareTag("Player"))
        {
            talkPanel.SetActive(false);
        }
    }
}
```

**Talk**

### # Third Person Movement (Mono Script)

```
Assembly Information
Filename: ThirdPersonMovement.cs
Assembly: CSharp.dll

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Component;
using UnityEngine.SceneManagement;
using UnityEngine.SceneManagement;

public class ThirdPersonMovement : MonoBehaviour
{
    public CharacterController controller;
    public float speed;
    public float runningSpeed;
    public float runningTime;
    public Transform mainCamTransform;
    private Animator animator;
    public GameObject endFootStep;
    private bool isRunning;
    private bool isWalking;
    public void Start()
    {
        if (Start() == true)
        {
            Update();
        }
    }

    // Update is called once per frame
    void Update()
    {
        if (Start() == true)
        {
            Update();
        }
    }

    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Player"))
        {
            if (gameObject.GetComponent<Interactable>().isInteractable)
            {
                gameObject.GetComponent<Interactable>().OnInteract();
            }
        }
    }

    private void OnTriggerExit(Collider other)
    {
        if (other.CompareTag("Player"))
        {
            if (gameObject.GetComponent<Interactable>().isInteractable)
            {
                gameObject.GetComponent<Interactable>().OnInteract();
            }
        }
    }
}

Influence = Input.GetAxis("Cross Axis");
if (influence > 0)
{
    speed = influence * walkingSpeed + walkingSpeed;
    running = influence * runningSpeed + runningSpeed;
    if (running > walkingSpeed)
        running = walkingSpeed;
    else
        running = walkingSpeed;
}
else
{
    if (influence < 0)
        influence = -influence;
    speed = influence * walkingSpeed + walkingSpeed;
    running = influence * runningSpeed + runningSpeed;
    if (running < walkingSpeed)
        running = walkingSpeed;
    else
        running = walkingSpeed;
}
```

**Third Person Movement**

### # Interactable Manager (Mono Script)

```
Assembly Information
Filename: InteractableManager.cs
Assembly: CSharp.dll

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Component;
using UnityEngine.SceneManagement;
using UnityEngine.SceneManagement;

public class InteractableManager : MonoBehaviour
{
    public static InteractableManager instance;
    public Dictionary<int, Info> foodBackpack = new Dictionary<int, Info>();
    public Dictionary<int, Info> equipmentBackpack = new Dictionary<int, Info>();
    public Dictionary<int, Info> implementBackpack = new Dictionary<int, Info>();

    private void Awake()
    {
        instance = this;
    }

    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
    }

    public void AddInteractableID(Info info)
    {
        Dictionary<int, Backpack> switchInfo = new Dictionary<int, Backpack>()
        {
            case InteractableType.Food:
                Backpack = FoodBackpack;
                break;
            case InteractableType.Information:
                Backpack = InformationBackpack;
                break;
            case InteractableType.Equipment:
                Backpack = EquipmentBackpack;
                break;
        };
    }
}
```

**ESC Controller**

## 10 Game UI



**SETTING**

### Screen

Resolution

V-Sync

Music

SFX

Control Guide

**TASK**

### EXCERPTABLE TASK

|           |                  |
|-----------|------------------|
| Task Name | Task Description |
| COMPLETED |                  |

### Task Information

|                  |                  |
|------------------|------------------|
| Task Description | Task Description |

**BACKPACK**

|         |         |         |
|---------|---------|---------|
| Slot 1  | Slot 2  | Slot 3  |
| Slot 4  | Slot 5  | Slot 6  |
| Slot 7  | Slot 8  | Slot 9  |
| Slot 10 | Slot 11 | Slot 12 |

|           |                  |                  |
|-----------|------------------|------------------|
| ITEM NAME | Task Description | Task Description |
| ITEM NAME | Task Description | Task Description |
| ITEM NAME | Task Description | Task Description |
| ITEM NAME | Task Description | Task Description |

Leave

# PROJECT 2 FANTASIA

## 11 Game Demo

