Proceedings of the 2nd International Conference on Interdisciplinary Humanities and Communication Studies
DOI: 10.54254/2753-7064/20/20231354

Design and Implementation of Puzzle Adventure Game Based on Unity

Weicheng Yuan[1,a,*]

[1]*National Pilot School of Software, Yunnan University, East Outer Ring South Road, University Town, Chenggong District, Kunming City, Yunnan Province, China*
a. lijia_z10j@itc.ynu.edu.cn
**corresponding author*

Abstract: With the progress of Gaming industry, the classic genre Puzzle adventure games, due to their friendly difficulty level, low learning curve, and interesting game mechanics, have been favored by a large number of players. This paper follows the guiding principles of software engineering and game development, designs and implements an adventure puzzle game using the Unity game engine. The game design is carried out through multiple aspects such as game art style, plot, sound effects, and game puzzles to provide players with an interesting gameplay experience. For the implementation part of the game, various plugins and functions of the Unity engine are integrated to complete the development of the game system, such as using Unity's navigation system to bake navigation meshes and implement automatic pathfinding for game characters. The animation state machine is used to arrange and control game character animations, and the Unity event system is used to implement click-based interactive functions in the game. At the same time, optimization was carried out based on the test results, improving the experience of user interface and the stability of game program, further enhancing the player's gameplay experience. Finally, players can experience thrilling adventure in a mysterious world, and achieve a sense of accomplishment by solving puzzles and overcoming obstacles.

Keywords: Unity, puzzle-adventure game, game design, game development

1. Introduction

Puzzle adventure games, a subgenre of adventure games (abbreviated as AVG), place a significant emphasis on logical puzzles within the adventure gaming experience. This genre typically prioritizes independent puzzle challenges through logical puzzles or gameplay. Solving puzzle can unlock more explorable contents, unveiling additional puzzles to solve, and advancing the game's story narrative [1].

Compared to typical adventure games, puzzle adventure games focus on puzzle design and interaction within the thrilling adventure journey, providing players with a sense of accomplishment when they successfully unravel puzzles and overcome obstacles. These kinds of game have gained popularity due to their user-friendly controls, low entry barriers, and engaging game mechanics.

Puzzle adventure games prevailed after the release of Myst and the 7th Guest, both of which involved pre-rendered images and movie clips. As computer technology advanced, puzzle adventure games embraced the capabilities of modern game engines, ushering in fully 3D gaming environments.

Other puzzle adventure games consist of a series of casual adventure games built around puzzles designed for exploration and storytelling, with titles such as “Witness” and the “Professor Layton” series serving as prime examples [2].

Puzzle adventure games contain diverse gaming elements that allows developers to unleash their creativity [3]. They incorporate intricate puzzles and mechanisms, unique level design, captivating background, and distinctive art styles without the support for massive game resources. Consequently, puzzle adventure themes have found favor among both indie and mid-sized game developers [4].

This paper revolves around the theme of adventure and puzzle-solving, using a point-and-click interaction system as the primary mode of player control. Multiple in-game scenes serve as the main means for the game’s narrative, allowing players to freely explore these environments, interact with objects, engage in conversations with non-player characters (NPCs), and ultimately unlock puzzles to progress through the game. The game design outlined above is implemented using game engine Unity, transforming it into a playable video game.

2. Game Design

2.1. Design Overview

The overall game design consists of basic art design, narrative design, and puzzles design in gameplay. The art of the scenes and environments serve as stages for creating ambiance in story. And the narrative of game primarily involves environmental storytelling, with the different scenes serving as the central element of a “three-act” narrative structure. Puzzles design play a role in core game play mechanics when player experience the game.

2.2. Art Style Design

The main artistic design of this game adopts a “low-poly” stylized approach. Emerging as an artistic style within the digital media evolution, low-poly primarily seeks aesthetic balance through abstracted representation [5]. In contrast to high polygon meshes, multiple vertex models, low-poly not only optimizes for performance displayed by a computer but also expresses a unique artistic style, focusing on essential artistic elements like geometric shape, lighting, and texture [6].

2.2.1. Scenes Design

The game comprises a total of six distinct scenes: the room, the parlor, the village, the forest, the graveyard, and the castle. All these scenes are constructed using low-poly style assets. To create an atmosphere conducive to adventure and puzzle-solving gameplay, each scene’s layout and vibe is meticulously integrated into the narrative, allowing players to immerse themselves in a sense of adventure and novelty as they explore each setting.

The indoor scenes are designed with a minimalist aesthetic that aligns with the low-poly style, as depicted in Figure 1. Scenes such as the village and forest vividly exemplify the characteristics of the low-polygon style, featuring flat, soft lighting effects where objects within the scenes have distinct edges, and the overall scenery leans towards abstraction.

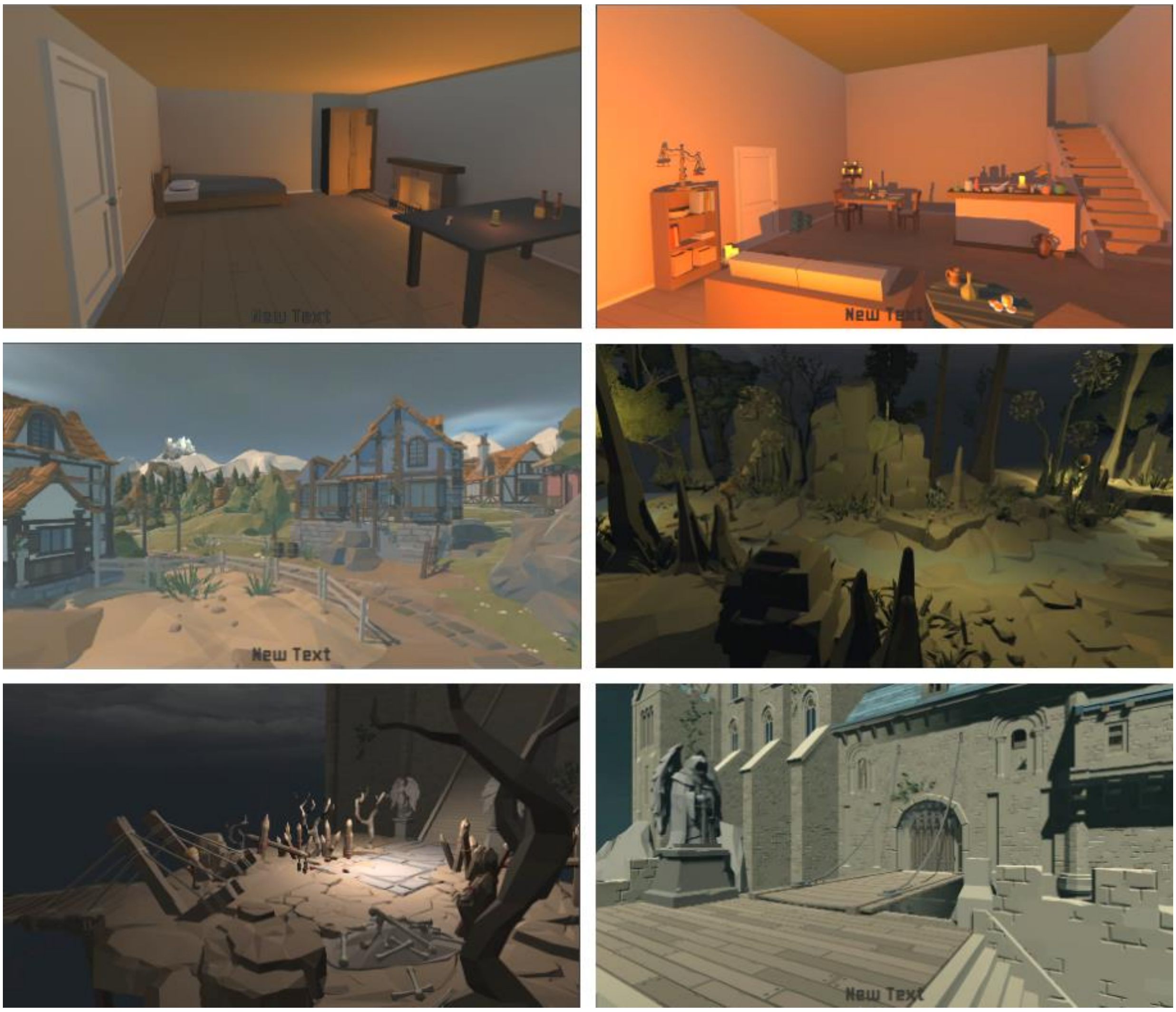


Figure 1: Scene art.

2.3. Narrative Design

The narrative design of a game is of paramount importance as it directly influences player immersion and their ability to empathize with in-game characters. The pacing of gameplay significantly impacts player immersion, and a well-balanced game narrative can effectively guide player emotions and entice them to continue playing [7].

2.3.1. Plot Outline

In this game, players assume the role of a young boy in a small village, who, guided by others, decides to venture beyond the village in a quest to reach the castle and rescue the princess. Starting the journey from his own bedroom, little venturer traverses through the village, forest, and eerie graveyard, engaging with diverse characters along the way, unraveling puzzles within each scene, and ultimately arriving at the castle to uncover its secrets and reveal the truth.

The overall narrative design of the game takes a summarily illustrative approach, in line with the low-poly visual style in the game. The plot emphasizes character development and the overarching storyline, without becoming overly entangled in minutiae. The fable-like narrative design aims for simplicity in story structure while remaining rich in expressiveness, aiming to evoke strong emotional connection from players in a metaphorical narrative way [8].

2.3.2. Three-Act Narrative

In order to tap into player's emotions, a well-structured narrative pace is indispensable. For those desiring a more profound emotional experience, a sequence of emotional fluctuations and accumulations is necessary [9]. The narrative pattern of this game adopts a three-act structure, which divides the storytelling into three distinct phases:

(1). The first act typically comprises one to two small fluctuations, providing context and background to the story and arousing player interest.

(2). The second act introduces a primary obstacle, prompting players to tackle challenges and further immersing them in the narrative.

(3). The third act takes players from the emotional low point to the highest peak, portraying the protagonist's triumph over adversity through courage and determination. By contrasting with the second act, this phase allows players to release the emotions they have accumulated.

Within this game, environmental storytelling serves as the primary narrative mechanism, with the narrative subject being the environmental settings in which the player-controlled protagonist is situated. These settings metaphorically symbolize the protagonist's circumstances and emotions.

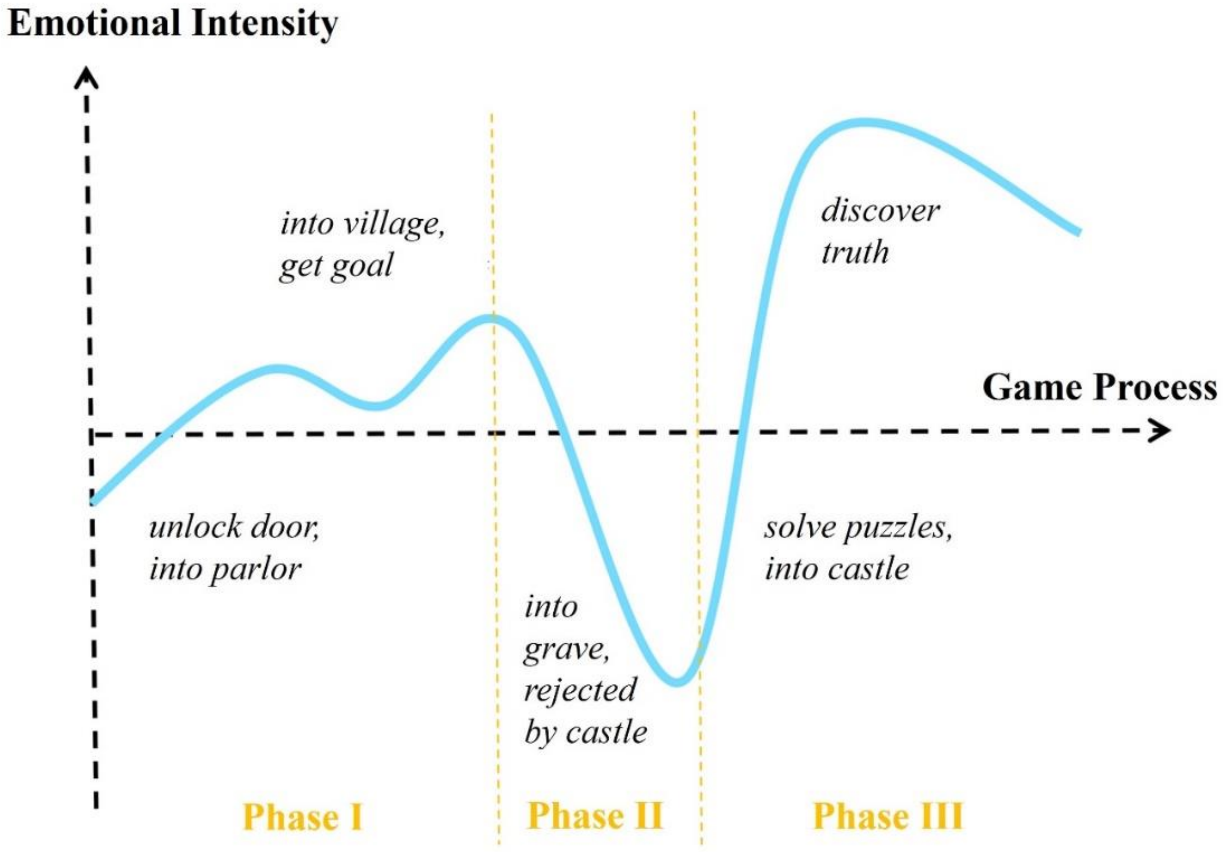


Figure 2: 3 Phases in narrative.

As illustrated in Figure 2, the game initiates with the first phase of the three-act structure. Players resolve relatively uncomplicated puzzles while engaging in dialogues with NPCs, gradually amplifying their emotional engagement. While proceeding into the village, but lost their way in forest after attempting to leave the village, the player's emotional state experiences minor fluctuations. Then after the conversation with the village chief, where the player is assigned the primary and clear goal—to enter the castle—that players gain a substantial understanding of the plot. Successfully navigating out of the village, the emotional intensity reaches a minor climax in phase I. Subsequently, they enter the second phase, venturing into the forest, graveyard, and encountering obstacles while attempting to enter the castle. This phase represents the emotional nadir in the player's journey, constituting the most significant obstacle in the game's progression. Finally, in the third phase, players successfully unravel puzzles, gain access to the castle, uncover hidden truths, and experience a culmination of emotions, reaching the peak of emotional intensity with the direct feedback from both gameplay and story narrative.

2.4. Puzzle Design

The level of difficulty in game puzzles has a direct influence on whether players can sustain their immersion in the game over time. Immersion, in this context, refers to the state in which individuals are fully engaged in the game scenario, filtering out all extraneous perceptions and concentrate on the task at hand [10]. Appropriate difficulty levels are essential to provide a challenging gaming experience. If the challenges exceed a player's capabilities, the tasks become daunting, leading to player anxiety. On the other hand, if the challenges fail to engage players, they quickly lose interest and abandon the game.

2.4.1. Puzzles Solving Method

The core gameplay of puzzles is the utilization of items. Players can obtain these items by exploring and searching the game environment or engaging in dialogues with NPCs. Each item possesses a unique function, requiring players to carefully observe and strategize item usage. Table 1 provides an overview of the puzzle-solving items and their associated functions.

Table 1: Items for puzzle-solving.

Item	Scene	How to obtain	Function
Key	Bedroom	Loot bedroom	Unlock the door to parlor
Coin	Parlor	Loot parlor	Buy Wine
Compass	Village	Talk to chief	Guide way in forest
Wine	Village	Use coin buying from vendor	For skeleton's quest
Mask	Graveyard	Trade with skeleton	As a disguise cheating guard

The ultimate goal of the game is to solve a central puzzle: entering the castle. Players must solve a series of smaller puzzles to progress further. To introduce complexity into the puzzle design, the key items necessary for solving puzzles within a particular scene may need to be acquired from other scenes, which demands players repeatedly traverse between various scenes in search of solutions. A visual representation of the puzzle-solving activities can be found in Figure 3.

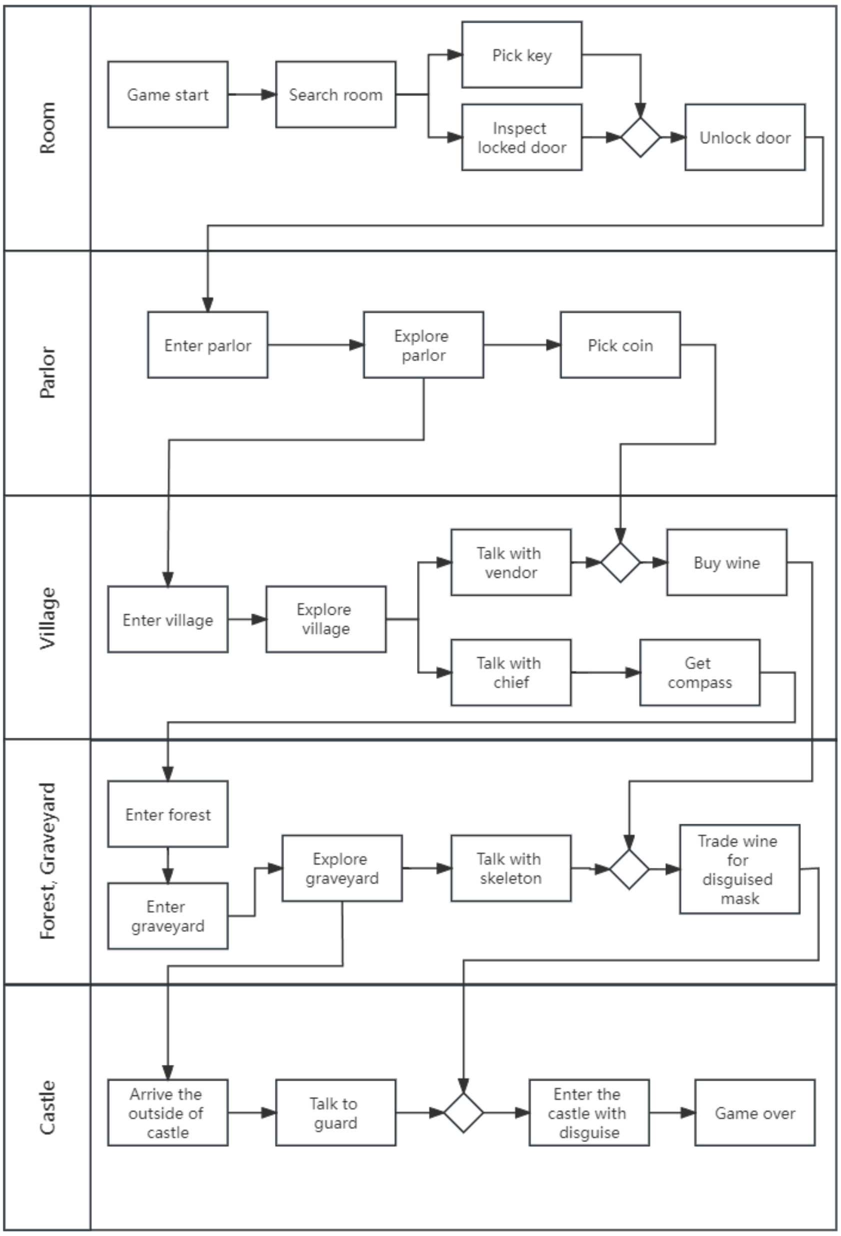


Figure 3: Puzzle-solving diagram.

The general puzzle-solving progression in the game unfolds as follows: The initial bedroom scene serves as a tutorial level, introducing players to the game mechanics. Players wake up in the room, investigate the room’s door, find it locked, and then search for the key to unlock the door, access to next scene. Into the parlor, players can talk with mother sitting on the couch, who hints that the nearby coins are crucial items. Players then collect the coins and exit the house, entering the village scene. Attempting to leave the village directly leads to getting lost and an inability to progress. In the village scene, players converse with the village chief, obtain the compass item, and learn about the destination of the game - the castle. With the compass in hand, players enter the forest, pass through the graveyard towards the castle. Being denied entry to the castle by the guard, players return to the graveyard and negotiate with a skeleton who seek alcohol. Returning to the village, players trade with a vendor, using the coins found in the parlor to purchase wine. Then players offer it to the skeleton, receiving a mask in exchange. Disguise mask enables players to deceive the guard and gain access to the castle, save the princess.

2.4.2. Hints of Puzzles

In order to maintain player immersion throughout the gameplay, game design must offer challenges of the appropriate difficulty of puzzles that corresponds to the player's capabilities, allowing players to become deeply immersed into the gaming experience. Therefore, during the puzzle-solving process, providing adequate hints is crucial, whether conveyed directly through in-game NPCs or indirectly through visual cues, as these hints can provide players with guidance to some extent, preventing them from getting stuck due to overlooking a clue and thus preserving the player's sense of immersion. Specific hint designs and strategies are outlined in Table 2.

Table 2: Designs of puzzles' hint.

How to hint	Details	Related Puzzles
Placing a candle and igniting a light near the key	Light illuminates key as a hint	Unlocking the door
Dialogue from mother in parlor	Reminding player get their pocket money	Buying wine with coin
Dialogue from villager in village	Mentioning the importance of compass	Exploring the forest without getting lost
Empty bottles surround skeleton	Serving as environmental cues for necessary item	Exchanging wine with skeleton to obtain mask

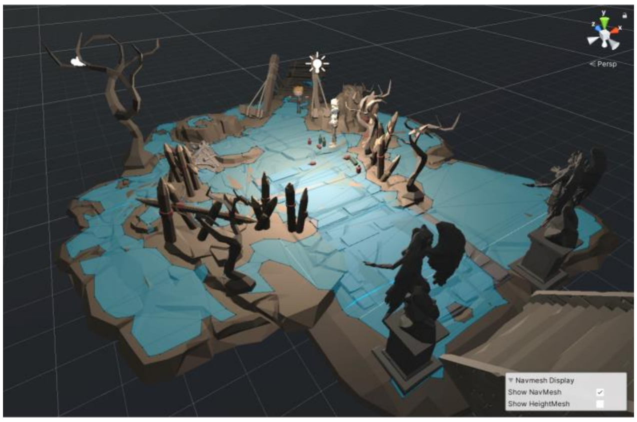
3. Game Implementation Based on Unity

3.1. Player Controller

The function of the player controller is to enable players to navigate the game world by clicking on any walkable area within the scene, allowing the character to move to the corresponding location. To implement player controller, initially, the event system in Unity is configured to listen for player mouse-click inputs to facilitate point-and-click interaction. Subsequently, an animation state machine is established to manage and execute animations corresponding to the player's actions.

3.1.1. Point-And-Click Interaction

To ensure effective navigation, the NavMesh is employed to bake the walkable areas within the scene, ensuring that the character only moves to locations when the player clicks within these baked NavMesh area.



▾ Navmesh Display
Show NavMesh ☑
Show HeightMesh □

Figure 4: NavMesh area.

The baking region is determined based on parameters pertaining to the character model, such as its height, radius, and stride. Figure 4 provides a visual representation of the NavMesh post-baking, with the blue-shaded area denoting the walkable area allocated for the character. Whenever a click event is registered within this blue zone, the character promptly navigates to the point of the click.

Event system in Unity monitors when and where a player clicks the mouse. The “Pointer Click” event within the “Event Trigger” component is then applied to scene assets that can be clicked, including floors, interactable objects, and NPCs. A “Physics Raycaster” is attached to the camera in the scene to emit raycasts from camera when clicking, triggering collision events and facilitating a series of actions to monitor and respond to the events. The activity diagram of click interactions is visualized in Figure 5.

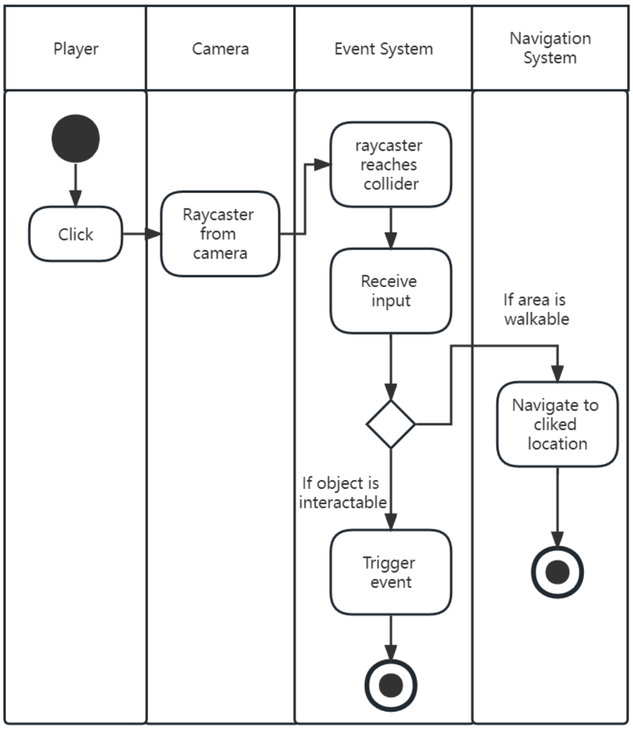


Figure 5: Player controller activity diagram.

3.2. Interaction System

3.2.1. Interactable

The interaction system stands as the critical component within the entirety of the game framework. Interactions between the player and interactable objects (including NPCs and items) rely on the interaction system, including walking, picking up items, using items, talking to others. The reactions generated by these interactable objects under varying conditions are intricately tied to the functioning of the interaction system. The system's ability to assess whether specific conditions are satisfied plays a pivotal role in these interactions.

Within the scene, two game objects exist: one being the tangible, visible model of the key, and the other an empty game object prefab called "Interactable". The latter is equipped with a collider and a script responsible for implementing the interactive functionality, managing all associated logical mechanisms. As illustrated in Figure 6, the green-bordered box represents the collider associated with the Interactable Game Object, which operates independently from the coin model and serves the purpose of detecting click events when an event trigger is added to the Interactable.

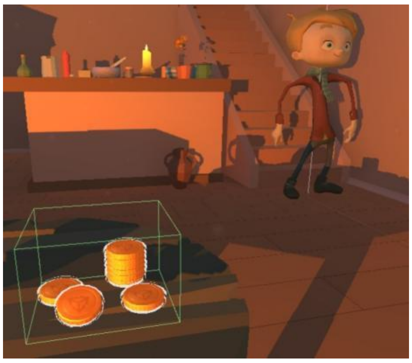
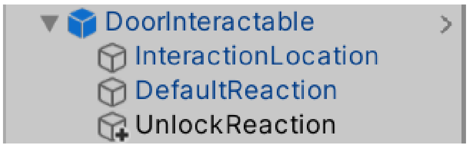


Figure 6: Collider of interactable game object.

Interactable, as a prefab, has two default child objects: "InteractionLocation" and "DefaultReaction". Additional child objects can be added as different reactions based on specific interaction conditions. Figure 7 displays the list of child objects of a Interactable for the door-opening interaction. Each Interactable object has an "InteractionLocation", which specifies the position and orientation to which the player must move when interacting with the object. And child objects suffixed with "Reaction" represent various possible reactions.

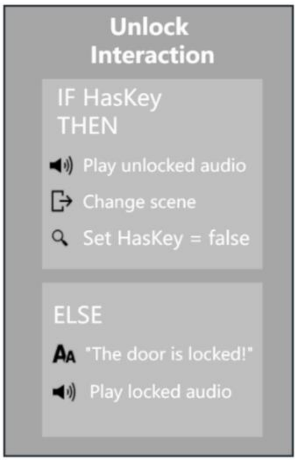


DoorInteractable
InteractionLocation
DefaultReaction
UnlockReaction

Figure 7: Children of interactable.

3.2.2. Conditions

The role of conditions is to trigger different responses when a player interacts with the same object under varying circumstances. Taking the example of using a key to open a door, Figure 8 illustrates how different conditions during interaction lead to distinct reactions. The condition of having a key is logically expressed as the boolean value HasKey=true, which depends on whether the player has picked up the key previously. If the player possesses a key during the interaction, a sequence of events is triggered to open the door. Conversely, if the player lacks a key, a reaction to keep the door locked will be played.



Unlock Interaction

IF HasKey
THEN

- Play unlocked audio
- Change scene
- Set HasKey = false

ELSE

- "The door is locked!"
- Play locked audio

Figure 8: Logic of unlock interaction.

3.2.3. Reactions

Reaction of interaction are implemented using scripts, with the primary function Interact(). This function is invoked when the player reaches the interaction location, enabling the execution of behavioral logic for interactable objects. Within this function, all conditions within the ConditionCollection that have set up in advance are traversed. If any conditions are satisfied, the corresponding reactions are executed. If in the absence of any satisfied conditions, the default response is executed.

Using the act of collecting a coin as an exemplar, Figure 9 provides an illustration of the implementation of fundamental interaction functionality. When the player clicks on a coin using the mouse, this click event is detected by an interactable object equipped with a collider, an event listener, and the script with function Interact(). Subsequently, the player's in-game character is directed to move to the designated interaction location associated with this particular interactable entity. The condition system proceeds to check whether the interaction conditions have been satisfied (in this case, picking up the coin does not require any conditions). If the conditions are satisfied, it triggers a preconfigured set of interaction reactions: playing a character animation for picking up the coin, displaying a text prompt stating "you get a COIN", adding the coin's image to the inventory, and setting the boolean value of the "Has Coin" condition to true.

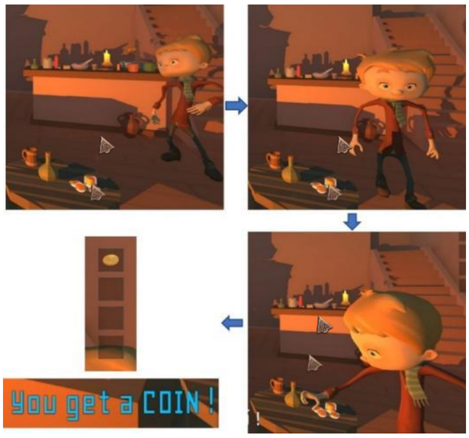


Figure 9: Reactions of collecting coin.

4. Game Test and Optimization

4.1. Puzzles Test

The testing of puzzle design serves several key objectives:

(1). Ensuring the solvability of puzzles.

(2). Ensuring puzzles possess clear and easily understandable goals.

(3). Ensuring no jumping to solve puzzles, including identifying any program errors or design oversights [11].

As a result of the testing process, it was identified that some puzzles may lack adequate player guidance, leaving players uncertain about how to proceed. As a remedial measure, additional hints were subsequently introduced.

4.2. Performance Test

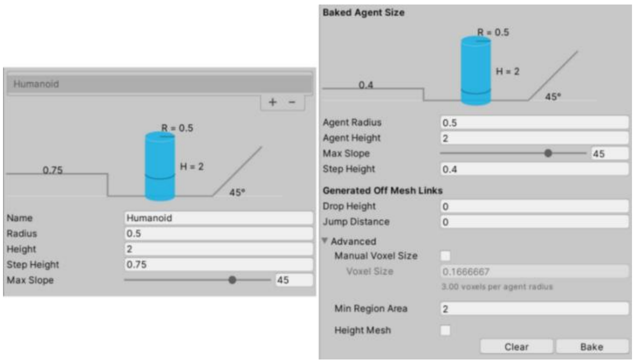
Following comprehensive testing, it is evident that the core functionalities of the game have been successfully implemented and running reliably. However, optimization challenges manifest during gameplay, leading to occasional performance disruptions and frame rate drops after extended sessions of play. The rendering of a substantial volume of models in real-time, particularly under conditions of limited computer graphics capabilities, results in extended rendering times and intermittent lag, thereby affecting the overall immersive experience [12].

Concurrently, minor disparities between select NavMesh-baked areas and the terrain mesh contribute to instances where characters seem to navigate slightly above the ground, resulting in a floating appearance.

4.3. Optimization

In order to mitigate the performance bottlenecks induced by the substantial computational load experienced during gameplay, a reduction in the number of vertex of models within the game's scenes was implemented. Furthermore, the adoption of Multiple Level of Detail (LOD) techniques facilitated the precise management of model complexity, consequently enhancing the efficiency of object rendering operations.

During testing, an issue emerged where the character's walking area became detached from the ground model, leading to characters appearing to walk above the ground. Upon investigation, it was determined that this problem was due to inaccuracies in the NavMesh baking process. Adjustments were made to two parameters within the baking settings in the Navigation panel: the maximum slope and agent step height. These adjustments resolved the floating character issue. Specific parameter settings are depicted in Figure 10.



Name: Humanoid
Radius: 0.5
Height: 2
Step Height: 0.75
Max Slope: 45

Baked Agent Size
Agent Radius: 0.5
Agent Height: 2
Max Slope: 45
Step Height: 0.4

Generated Off Mesh Links
Drop Height: 0
Jump Distance: 0

▼ Advanced
Manual Voxel Size: □
Voxel Size: 0.1666667
3.00 voxels per agent radius
Min Region Area: 2
Height Mesh: □

Figure 10: Optimized configuration.

As illustrated in Figure 11, the problem of characters floating when walking or standing in the village scene has been effectively resolved by modifying the baking settings and re-baking the NavMesh. In contrast to the pre-optimization scenario, there exists an enhanced level of alignment between the NavMesh and the Mesh Renderer of the terrain.

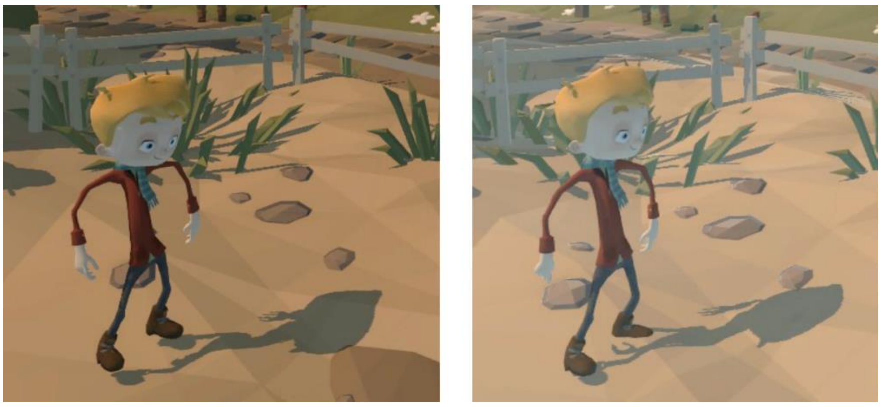


Figure 11: Before-And-After optimization.

Furthermore, improvements have been made to the in-game UI, with enhancements made to both the main menu and the pause menu. A feature has been added to enable players to seamlessly return to the main menu from within the game, providing a more user-friendly interactive experience. As demonstrated in Figure 12, users can initiate this action by clicking the “Back to Menu” button.

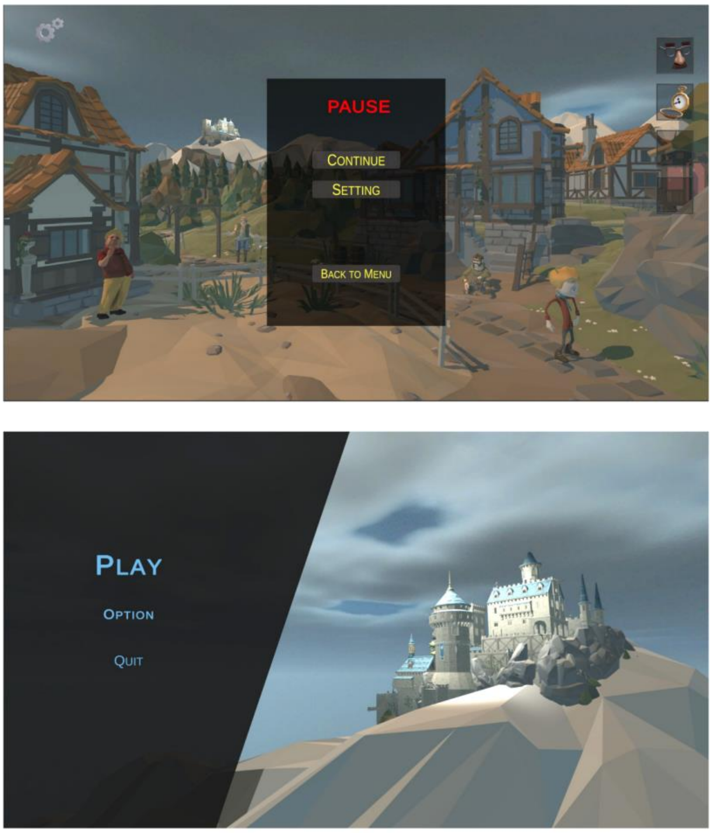


Figure 12: Optimized UI.

4.4. Survey

In order to optimize further gaming experience based on players' feedback, volunteers were asked to fill a questionnaire after playing the whole game. The questionnaire focuses on direct feelings about some indexes which value the completeness of game design and implementation.

According to the result of survey Table 3 showed, it is clear that the guidance of the basic control requires more specific tutorial when players first time playing the game. And the difficulty level needs to be improved, considering the hints of puzzles are adequate. Unique art style design and narrative design including environmental storytelling, 3-act narrative in this game played an important role in creating immersive atmosphere, that is the reason why most volunteers gave the positive respond on this index. Some low scores on storytelling may relate to the limited number of in-game cutscenes and animations, which needs further improvement in this project.

Table 3: Questionnaire on gaming experiences.

Players	Questions (on a scale 1 to 10)		
	How would you rate the controlling and the guidance of game?	How would you rate the difficulty level of these puzzles	How would you rate immersion of the storytelling in this game?
Guo	7	8	9
Leo	8	6	9
David	8	9	8
Eason	7	7	10
Liu	7	9	9
Yin	10	6	10
Zhou	6	9	6
Li	8	5	8

5. Conclusion

This paper primarily investigates the design methods for adventure puzzle games, along with the utilization of various Unity components and techniques, including event system, animation state machines, AI navigation, etc. The research findings hold theoretical significance and practical guidance for the design of the game. Additionally, they have practical implications for the game implementation.

Regarding the game design aspect, the paper encompasses game system design, game art design, and game puzzle design. To ensure player immersion in the game's narrative, a "three-act" game storytelling rhythm design approach is employed.

In the game implementation section of this paper, Unity game engine is utilized to complete the implementation of the game, focusing on three main modules: character controller, interaction mechanism, and game state. A combination of Unity engine plugins and various functionalities is employed to develop the game system. In terms of the game's UI system, enhancements are made to the existing Unity UI components to achieve an aesthetic and streamlined interface, aligning with contemporary UI design standards.

Lastly, the game system is subjected to testing, and optimization measures are applied based on the test results. These optimizations enhance the rendering efficiency of the in-game engine and reduce the computational load on the computer during gameplay.

However, it's worth noting that there are areas for improvement in this project:

(1). The game features relatively few game scenes and puzzle designs, resulting in a relatively small game scale.

(2). The number of in-game cutscenes and animations is limited, leading to a somewhat deficient narrative expression.

References

[1] Rouse, Richard. (2005) Game Design: Theory and Practice (2nd ed.). Worldware Publishing, Inc., 233.

[2] Neves, P.P., Zagalo, N. (2021) Game Design as an Autonomous Research Subject.Information, 12-13.

[3] Jie, Xu. (2023) Research on Puzzle Adventure Video Game Design from the Perspective of Spatial Narratology. Jiangnan University, 25.

[4] Sailer,M.,Hence,J.U.,Mayr,S.K. (2017) How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. Computers in Human Behavior, 371-380.

[5] Jing, Wang. (2016) Research on The Application of Low Poly Style in The Design of Mobile Game Scene. Donghua University, 4.

[6] Qian, M.H., Clark, K.R. (2016) Game-based Learning and 21st century skills: A review of recent research.Computers in Human Behavior, 63, 50-53.

[7] Yan, Hou. (2019) Transmedia Narration-The Narrative Analysis of Video Game. Hainan University,1, 4-5.

[8] Johan Huizinga. (1971) Homo Ludens. Beacon Press, 3, 6-12.

[9] Masahiro Sakurai. (2022) Masahiro Sakurai on Creating Games. Retrieved from https://www.youtube.com/channel/UCv1DvRY5PyHHt3KN9ghunuw/featured.

[10] Jenova, Chen. (2007) Flow in Games (and Everything Else).California: University of Southern California, 8-9.

[11] Deterding, S. (2015) The Lens of Intrinsic Skill Atoms: A Method for Gameful Design.Human-Computer in Interaction, 30, 3-4.

[12] Long, K.H., (2023) Research on Display Technology of Spatial Scene with Large Data Based on Unity3D .Guangxi University, 16.