Emre Kaplan

08.07.2024

Unity Project Report: BBQ Party Game

Project Overview

This project is a Unity-based roulette spin game where players spin a wheel containing various items. The goal is to create an engaging user experience where items on the roulette wheel are highlighted, spun, and the player is awarded a random item. The game includes features such as animations, a reward system, and a user interface to enhance gameplay.

Objectives

- Develop a roulette spin game using Unity.
- Implement a visual representation of the roulette wheel with various items.
- Create animations for the spinning wheel and reward collection.
- Implement a reward system to add items to the player's wallet.
- Provide user feedback through a win popup and item highlighting.

Key Features

1. Roulette Wheel Animation

- **Highlighting Items:** Items on the roulette wheel are highlighted with a gold square as they are selected during the spin.
- **Spinning Animation:** The wheel spins for a set duration with a specified number of cycles and random stopping points.
- **Blinking Effect:** The final selected item blinks before being permanently highlighted.

2. Reward System

- **Player Wallet:** A dictionary-based wallet system to store collected items.
- **Win Popup:** Displays the rewarded item and a congratulatory message.
- **Item Collection Animation:** Animations to move the rewarded item from the wheel to the player's wallet.

3. User Interface

- **Start Button:** Initiates the spinning of the roulette wheel.
- **Win Popup:** Displays the collected reward and a close button.
- **Tick Symbol:** Indicates the final selected item on the wheel.
- **Wallet Indicator:** Displays the player's wallet contents.

Implementation Details

Scripts

1. RouletteSpin.cs

- **Purpose:** Manages the spinning animation, item highlighting, and reward collection.
- **Key Methods:**
  - `StartSpin()`: Initiates the spin coroutine.
  - `Spin()`: Handles the spinning logic, item highlighting, and blinking effect.
  - `ResetPreviousSquare(int index)`: Resets the previous gold square to its original image after a delay.
  - `ShowReward(string itemName)`: Displays the reward in the win popup.
  - `GetRewardPrefab(string itemName)`: Retrieves the corresponding prefab for the rewarded item.
  - `CloseWinPopup()`: Handles the win popup closure, resets the selected item, and re-enables the spin button.

2. PlayerWallet.cs

- **Purpose:** Manages the player's wallet, adding rewards and updating the UI.
- **Key Methods:**
  - `AddReward(string itemName)`: Adds the rewarded item to the wallet and updates the UI.
  - `UpdateWalletUI()`: Refreshes the wallet display with current items and their quantities.

Animation

- **Item Highlighting:** Items are highlighted with a gold square as they are selected during the spin.
- **Blinking Effect:** The final selected item blinks between gold and its original image before being highlighted with a blue square.
- **Movement to Wallet:** Rewarded items move from the roulette wheel to the wallet indicator on the screen.

User Interface

- **Start Button:** Users click to start the spin.
- **Win Popup:** Displays the reward and a message when an item is won.
- **Tick Symbol:** Indicates the selected item on the roulette wheel.
- **Wallet Indicator:** Shows the player's collected items.

Challenges and Solutions

1. Synchronizing Animations

- **Challenge:** Ensuring smooth transitions and synchronization between item highlighting, spinning, and blinking effects.
- **Solution:** Utilized Unity's coroutine system to manage timed animations and transitions.

2. Reward System Integration

- **Challenge:** Accurately updating the player's wallet with the correct item and quantity.
- **Solution:** Implemented a dictionary-based wallet system and ensured proper prefab instantiation and UI updates.

3. User Feedback and Interactivity

- **Challenge:** Providing clear and engaging feedback to the user during gameplay.
- **Solution:** Added visual indicators like the tick symbol, win popup, and animations to enhance user experience.

Conclusion

This Unity project successfully implements a BBQ Party game with engaging animations, a robust reward system, and an intuitive user interface. The project demonstrates effective use of Unity's features to create a visually appealing and interactive game. Future improvements could include additional gameplay features, enhanced animations, and expanded reward options to further enrich the user experience.

Related Game Links
Game-play video record: https://youtu.be/3LeLnXuBuS0
Unity Project Files: https://github.com/emrekaplannn/Unity_BBQ_Party