Third-Person Character Controller Project

Overview

This Unity project implements a third-person character controller with camera management, item collection, and health management features. The project follows SOLID principles to ensure code maintainability and readability.

Features

1. Player Movement

- **Player Controller Script**: Handles character movement, including forward, backward, left, right, and jumping. Movement is smooth and responsive, providing a polished gameplay experience.

2. Camera Control

- **Cinemachine**: Integrated with Cinemachine to provide third-person camera control. The camera follows the player, and the mouse input allows the player to rotate the camera view. The camera control script is included within the "Player Controller" script.

3. Item Collection System

- **Scriptable Object for Item Collection**: A `Float Scriptable Object` is used to store the number of collected items.
- **Collectible Script**: Attached to collectible items in the environment, allowing the player to pick them up. This script updates the "Collected" Scriptable Object upon item collection.
- **CollectedUI Script**: Updates the UI to reflect the number of collected items dynamically as the "Collected" Scriptable Object changes.

4. Health Management System

- **Scriptable Object for Health**: Similar to item collection, a `Float Scriptable Object` is used to store and manage the player's health.
- **Health Script**: Manages the player's health, applying damage, tracking the health state, and triggering death when health reaches zero.
- **UISlider Script**: Updates the health bar UI in real time, based on the value in the health Scriptable Object.
- **Throwable Weapon Script**: Implements a turret that shoots bullets, dealing damage to the player.

5. Game Over and UI Management

- **Game Over Logic**: Displays "Game Over" when the player's health reaches zero. Other UI elements are managed as per the task requirements.

SOLID Principles Applied

1. **Single Responsibility Principle (SRP)**: Each script has a single responsibility. For example, the `Player Controller` script handles movement, while the `Health` script manages health-related logic.
2. **Open/Closed Principle (OCP)**: The use of Scriptable Objects allows the game to be easily extended with new items, health conditions, or UI changes without modifying existing code.
3. **Liskov Substitution Principle (LSP)**: Wherever interfaces or base classes are used, derived classes can substitute for their base classes without altering the correctness of the program.
4. **Interface Segregation Principle (ISP)**: Interfaces (if any) are designed to ensure that no client is forced to implement methods it doesn't use.
5. **Dependency Inversion Principle (DIP)**: The project relies on abstractions, such as Scriptable Objects, for managing game data rather than concrete implementations.

Installation and Setup

1. Clone this repository to your local machine.
2. Open the project in Unity (version 2021.3 or higher recommended).
3. Play the scene to test the character controller, item collection, and health management systems.

Conclusion

This project demonstrates a third-person character controller with integrated camera control, item collection, and health management, adhering to SOLID principles. The code is designed to be maintainable, readable, and extendable.