Project Summary

Overview

This is a 3D RPG style game, built in Unity 6 and using Universal Render Pipeline. The main core functions developed in this project are a slot-based inventory system with persistent saving, an interaction system and a character movement. The main scene “Village” showcases a low poly medieval village and it was found at the Asset Store.

Architecture

Inventory

The inventory main script is called “Inventory” and it holds a list of “Slots”. This script acts as the manager of the slots and initializes them at the beginning. For that, it checks if there is anything saved for using at initialization (I’ll say more about it when I come to Saving). This script uses the Singleton Pattern, in order to make the access easier.

The inventory will send a callback broadcasting that it finished the initialization. The InventoryUI will listen to that and initialize its UI representation of the slots accordingly. It will also start listening and handling slots UI events, like dragging, hovering and selecting. It will keep track of what is being selected and the activation of the corresponding buttons. It will also handle the dragging behaviour and guarantees that the item swapping from a slot to another will occur. The same for deleting and using an item.

A slot holds an Item, which is a Scriptable object with some information. Everytime the item on a Slot changes, it will notify via callback. The slot UI representation will be listening to it, in order to update the visual info of it. Here I used the Observer Pattern approach, due to its flexibility and modularity.

Interaction

This was designed based on the Strategy Pattern and allows it to easily extend to many kinds of interactions that will work well in the same system.

There is a component in the player GameObject that keeps doing a sphere cast to catch the nearby interactables. Those are defined by an interface called “IInteractable”, which has three methods:

- **Hover:** The player will trigger when it comes close to its collider and it is the closest interactable.
- **Unhover:** Will be triggered when the player walk away from its collider or it is not the closest interactable anymore
- **Interact:** Will be triggered when it is hovered and the player press the designated input (the key “E”)

Each of these interactions have an UnityEvent, in order to easily chain different reactions. I used it to implement some feedback to the user when it comes closer to an Item.

Movement
For this one, a simple movement code was implemented. The player input (WASD) was mapped to the world coordinates in order to give the character a direction to move and turn around.

Since the camera is Top-down isometric (it is not at orthogonal, it is at perspective though), I did not have to worry about mapping the input to the camera coordinates, because the camera never rotates. So, the camera will always face the world forward.

I used Cinemachine to set the camera behaviour.

Impressions

It was nice to create those mechanics. I play them all the time in so many games and to decouple how they work it’s amazing.

I think that the real challenge at this task was the UI for the Inventory system. I have already done it before, but not slot based. At the beginning, I started with an infinite inventory approach. Pretty easy to migrate to a slot base. The real difficulty and time consuming was implementing the hovering and drag and drop behaviours. It was not smooth, but I am very satisfied with the results.

Next steps

Of course there are a lot of things that can be done at this project. But there are two things that I think that would be the most relevant to a hypothetical future game.

Firstly, I would like to implement an equipment system. At first, I thought about doing that, given the good assets that I found. But the time was too short for doing it properly. I would love to equip different armors in the character. I would do the same approach that I made for the Consumables: Inherit from Item and try to check its type for calling its own methods.

The second thing is implementing NPCs. The interactable system it’s so easy to extend that makes me eager to implement new kinds of interactions with the world. And no interaction it’s better than with other beings… even if they are not real. It would give life to the village and it could be the entry point for many other systems, like Shops, Dialogues and so on.