

The system is basically divided into a player controller, an inventory controller, a coin controller, a user mouse input controller, a game manager for information exchange, and a UI manager with interface state control to manage when each interface should be enabled/disabled or updated.

At the beginning of my process, I started by thinking about the necessary systems for the project. Initially, I identified that I would need to create a script for player movement, a script for player inventory control, a general game control script, and scripts for the interface.

After that, I began looking for the Unity tools I would use. At first, I realized that I would need Cinemachine, the new input system and 2D package. After including these tools, I started creating the player script. I developed the movement code and prepared the character's animation code. I didn't implement the initial animation because I prefer to prototype before adding animations.

After completing the player's movement, I started prototyping the environment, using a visual asset that I had previously downloaded. Once I tested and confirmed the movement, I moved on to creating the inventory system. I began with the item's logic and then the inventory's logic. After finishing both logics, I started creating a UI prototype along with the interface control scripts to test the inventory system. Finally, I updated the scripts to have some integrations between them. I created the mouse input controller to check user interactions and set up the store using the scriptable objects for items created earlier. On the polish stage, I added the door opening and made the store interface more interactive based on the player's money.

Personally, I think I handled the tasks well. The only part that fell a bit short was the equipment slots and wearable visual aspect. I didn't have much time and ended up implementing it in a simpler way. Regarding the wearable visuals, maybe I should have researched paper doll systems a bit more, but overall, I think I did well, and I found the test quite manageable.