BGS programmer interview

System explaining

Initially, after analyzing the instructions provided in the interview document, I outlined the ideas for the game development, focusing on creating a more modular and reusable code structure. For the artwork, I chose a top-down Unity Asset model and created custom clothing to align with the 2D model's animations, ensuring synchronized movement when layered on the animator's coordinates.

To implement the inventory system, I created three foundational scripts: one for the inventory itself, another for the inventory UI, and a third one for the UI slots. The inventory can exist independently, serving as a container for storing items. The UI script was designed to receive an inventory instance and dynamically update the slots based on any inventory changes. The slots are represented as buttons, each holding an item from the inventory along with its corresponding icon. When a slot is pressed, it performs a specific action based on the inventory type it belongs to. These scripts were developed with modularity and expandability in mind, allowing for various inventory and UI variations to be easily implemented.

The player's inventory was designed to store the items the player possesses, and its UI enables the player to equip items in a separate equipment inventory. When a player clicks on an inventory slot, the item is equipped, removed from the main inventory, and transferred to the equipment inventory.

The shop's inventory UI functions differently. It has its own inventory and UI elements. When the shop object is interacted with, it receives an instance of the player and can access their inventory. This allows the shop's UI to populate a secondary shop inventory UI, specifically for displaying items available for purchase. Items in the shop can be sold to the player if they have sufficient funds. If a purchase is made, the item is removed from the shop's inventory and added to the player's inventory using the instance obtained during the shop interaction.

Items in the game are created using Scriptable Objects, based on the ItemBaseSO. The ClothesSO is derived from the ItemBaseSO and contains basic item information such as the inventory icon, name, price, and type (weapon or clothing). Additionally, ClothesSO includes specific properties required for equipping the item, such as animations corresponding to different character directions and movements. These animations ensure seamless integration with the Animator when an item is equipped.

Furthermore, I implemented an interaction mechanic by creating a base class called Interactable, which can be extended to handle various interaction types. I developed interactables for objects with two states (open and closed), the shop (which opens the inventory window for buying and selling items), and transport (allowing the player to be taken to another location upon interaction).

This overview covers the main features of the game I developed, presented in a concise and clear manner to showcase the project as a game developer experienced in Unity 2D.