JOÃO PAULO MANFREDINI BARROS

DOCUMENTATION ON “TEST TASK” PROJECT
FOR BLUE GRAVITY STUDIOS

GAME SUMMARY

For this project I created a really simple environment containing only the requested necessary elements. An npc with a functioning shop system and dialogue, and a player with a functioning inventory, equipment and movement system.
The game has a top-down view and is set on a small island.

CODE STRUCTURE

The main systems of this prototype (excluding player movement) can be explained as follows:

Player Inventory: The system behind the player’s inventory is quite simple, it’s merely a list that can hold objects of type ITEM, a scriptable object that holds the following data:
string slotType (head, chest, or feet, this data will be used by the equipment system)
string itemName (name to be displayed on all UI systems)
sprite itemImage (image used to represent the item)
int itemPrice (buying price of the item)
int itemSellPrice (selling price of the item)
GameObject itemObj (GameObject that will be instantiated on the player by the Equipment system)

Player Equipment: Simply a secondary List to the Inventory one, with a limit of three items per slot (head, chest, feet). When the player clicks an item from the inventory panel, the game will check what slot that item represents, let’s say I click on a shirt, in that case the slot will be of type “chest”, then the game will check via three boleans if the chest slot is already full, if it’s not, it will move the Shirt from the inventory list to the Equipment list, and spawn the corresponding ItemObj as a child of the player.

Shopkeep Npc Inventory: Same list-based system used by the player.

Item icons: There are four kinds of item icons that can appear in the game, those that represent items on the shop, on the player’s selling list, on the player’s inventory and on the player’s equipment tab.
Each has a different functionality:
If you click on a shop icon, you are trying to buy an item.
If you click on a sell list icon, you are trying to sell an item.
if you click on an Inventory icon, you are trying to equip an item.
if you click on an equipment icon, you are trying to un-equip an item.
So each icon-type has it’s own script, but they all share the Create() funcion, that requires a reference to the ITEM Scriptable Object that slot represents.

FINAL THOUGHTS

If I were to remake it from zero, I'd change the code architecture in a way so that I wouldn't require a different script for each slot type. and probably spend a little more time plotting out how I wanted the code organization to turn out, as I ended up changing lots of things during development that made the code harder to customize.

I'd say the game performs well on the front-end, but the back-end could use a little more work, if this system were to be implemented on a full game.

Having said that I'd say that overall I'm satisfied with how the project turned out.