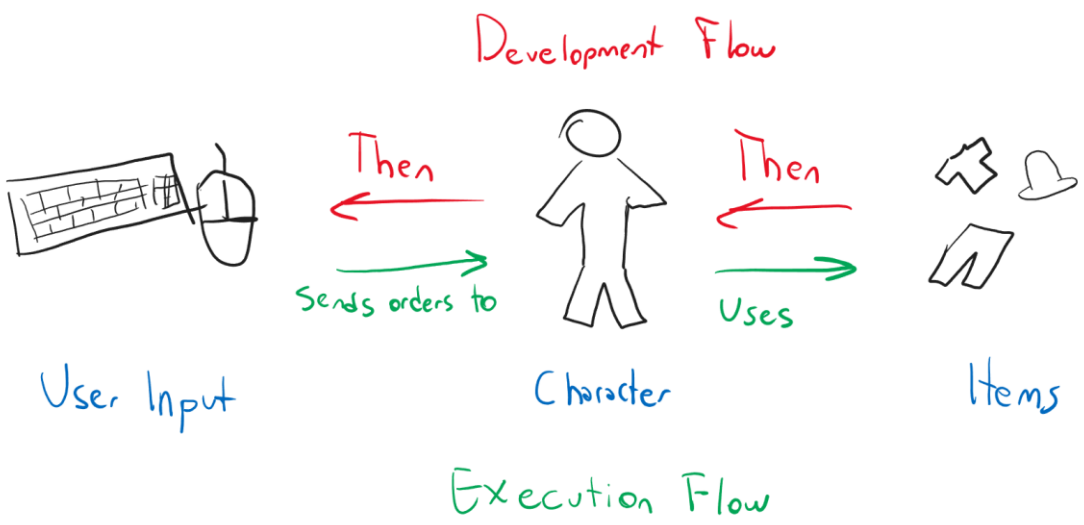
Technical test – Blue Gravity Studios

Lucas Negro

Beginning - Outfit system

Before I started, I **searched for some primordial assets** to create a first prototype, which were the character and the outfits. Luckily, the character base asset mentioned in the task worked well for me. I knew the **outfit system** would be the most complex part for me, so I **began developing** that in a prototype scene. That system should be one that receives orders from other scripts, in other words, **it shouldn't work by itself, ensuring controlled outputs from given inputs**.

Also, to ensure everything worked well, I started from the end, and walked backwards to the source. For example, the end would be the items themselves, and the beginning would be the source of the order to change an outfit. So, I started programming the items in Scriptable Objects, then the character, which uses the items, and then the Selection Handler (user input), which sends commands to the player to use given items.



In red, my development flow, in green would

The Scriptable Objects uses inheritance to define type of items. For the moment, Outfits are the only items in the project, but it is predisposed to create new type of items. Every item has 5 values:

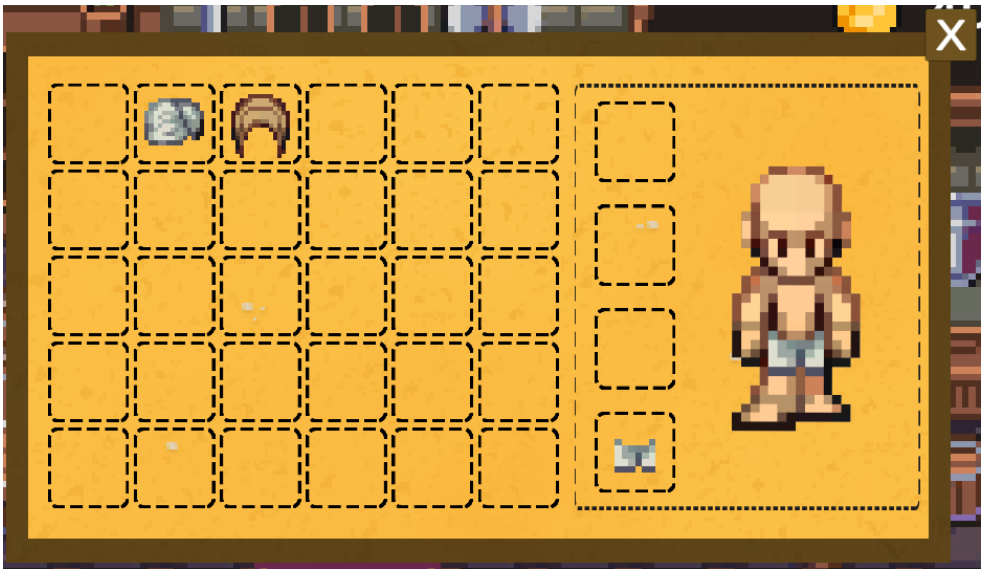
- Buy Value: Used for purchasing the item through the shop.
- Sell Value: Used for selling the item through the shop.
- Title: The item name.
- Item Object: A prefab of the object to instantiate in the world (used only on outfit items).
- Item Sprite: The icon that represents the object. Inventory and shop uses this property to display the item.

Movement and Animations

First, I **prepared the animations** for the base character and the outfits as mentioned in the documentation. Then, I developed the **character movement**, making it possible for the player to control. In this case I used MVC to separate logic, data and visuals. PlayerView controls the character and outfit animations.

Inventory system

For the inventory system, I had to do some research, so part of the code was taken (no copy/paste) from tutorials. Inventory has two sectors, one for unequipped items, and other for items you are wearing (outfits).



One thing that has been added later is a backpack icon, to open the inventory with the mouse (you can also open the inventory with the "I" key).

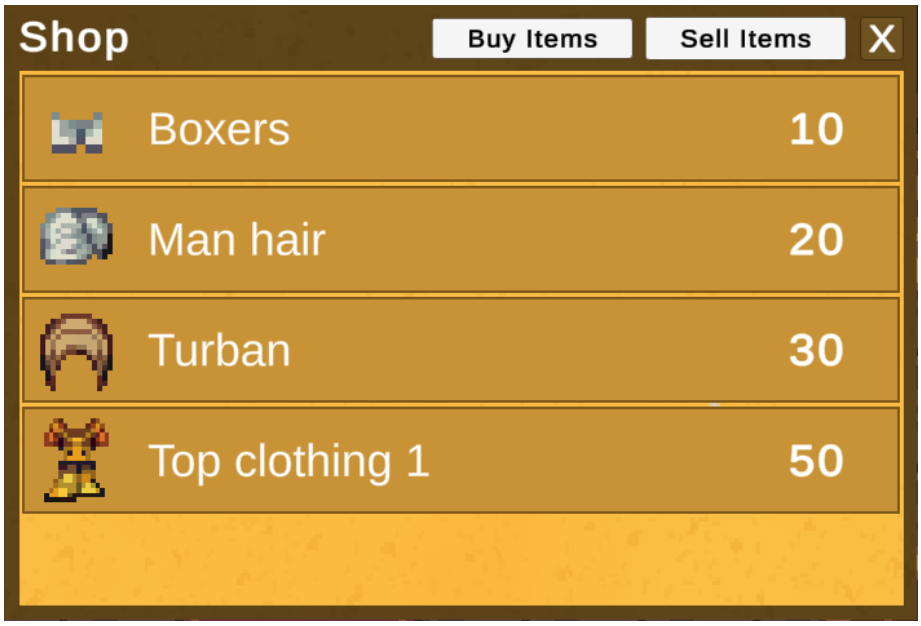
Shopping / buying and selling system

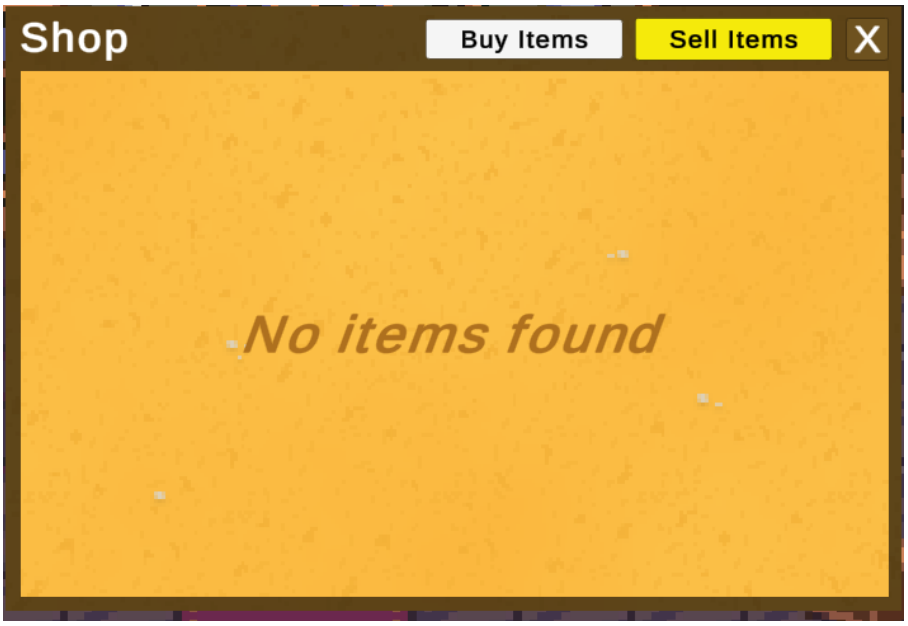
Once finished the inventory, I developed the shopping system. The shop UI is a list-based catalog, and every item displays icon, title, and cost. Also, you have two sections, one for items that you can **buy**, and another for items that you can **sell**. The buy and sell values are taken from the item Scriptable Objects according to what I mentioned in the Outfit System section. The items for the buying list are pre-defined through inspector, while the selling list queries the player's unequipped section of their inventory.

When the player buys anything from the store, it is **removed from the list and added to the player's inventory**, and the coins value is updated. When the player sells anything to the store, **it removes the**

item from the inventory, and updates the coin counter. When there's no item to show, a text appears indicating *"No items found"*.

Lastly, when the player tries to buy an item that costs more than the current number of coins, the buying action is cancelled.





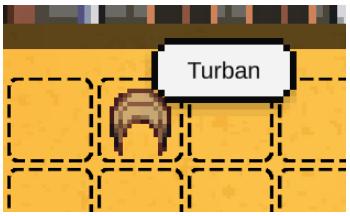
The coins counter functionality is basically two scripts, one in charge of the UI, and other one that handles the logic. The coin value in the UI is updated through event form the logic script, which is also executed every time the player spends or earns coins.

Scene

After finishing the systems, I created the scene. I avoided doing any complex dialog/press-to-interact things to open the shop because of the time, so I made the player to open the shop when getting closer to the shopkeeper (trigger). The scene is a large image covered with colliders.

Tooltip

One thing I added to the game is a tooltip system. Useful to display an item name when the player hovers the pointer over the item (in case the icon isn't enough information for the player).



Fixes

With the final scene, I did some playtesting and found several bugs, mostly related to the inventory or shop.

There was a bug I found was that the player can control the character even with the inventory or shop UI displayed. To solve this problem, I grabbed an Events Manager script that was pretty handy for me in previous projects. I made the player subscribe methods, one for when the inventory/shop appears, and another when the UI closes. Then I made the Input get blocked when a screen is displayed, and unblocked when hidden.

Personal assessment

This project was very interesting. I learned how to make an inventory system and a shop system with almost no knowledge. So, regarding the impact on the interview process, I enjoyed this time.

I think one big mistake I made was overcomplicating the Item hierarchy (the outfit types). I tried to implement inheritance and avoid enums and switch statements to achieve some sort of abstraction, but I ended up depending on them anyways. I think with just enums I could have achieved the same result with less time and less lines of code.

Besides that mistake, I feel that I made a decent job.