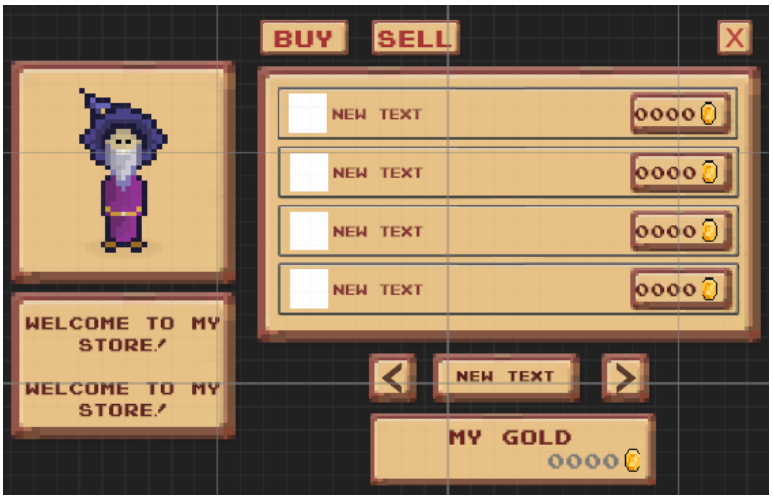
Clothe Shop Simulator

Mechanics

- movement : wasd or arrow keys
- interaction (Merchant) : Get in front of the merchant. then an interaction mark will appear above him. then you can press “space” to interact with him.
- open inventory : Click on the bag button (left-down corner) to open it:

UI panels

1. **Merchant Panel (singleton)** : in this panel you can buy or sell your inventory items. “Buy Button” shows the products that the merchant sells. “Sell Button” Shows your inventory items. You can sell them. On the bottom side shows the money you have. Additionally, you can browse many pages (all items) both in buying and selling.



2. **Inventory Panel (singleton)** : Show all equippable items divided by class(helmet, armor, weapon). Just click on one of them (bottom part) to equip the item

Development Process

your thought process during the interview, and a personal assessment of your performance.

1. I configured my machine for 2d development (install engine, assets, repositories). Then, I analyze all the assets I own at the moment. I see the distribution of these (animations, sprites, etc.)
2. I collect information from similar projects to understand the mechanics I will use.
3. I create a mental image of what the finished product will look like (art, mechanics and sketch).
4. I create a list of small tasks (user cases). Then, I sort them by priority and dependency (split or merge too).
5. I design each task before coding. How it will interact with other components. Sometimes a refactor is necessary. For example, the merchants had a “merchant panel” each one. but I converted it into a singleton and abstracted the game items' data into scriptable objects.
6. Finally I created the missing icons and other details.

Personal Assessment of Performance

There are many aspects to improve, such as selling an equipped object (default items for unequipped items). But it was my first 2D game and I finished it in a calculated time (taking necessary breaks). Also, I needed to comment better on my code. And as always, it is necessary to invest more time in the design to avoid refactors.