



This is a simple store where the player can move around, exit the store, interact with the shopkeeper, buy items/clothes/accessories and wear them.

Controls:

WASD/Arrow Keys: Move Character

E: Interact

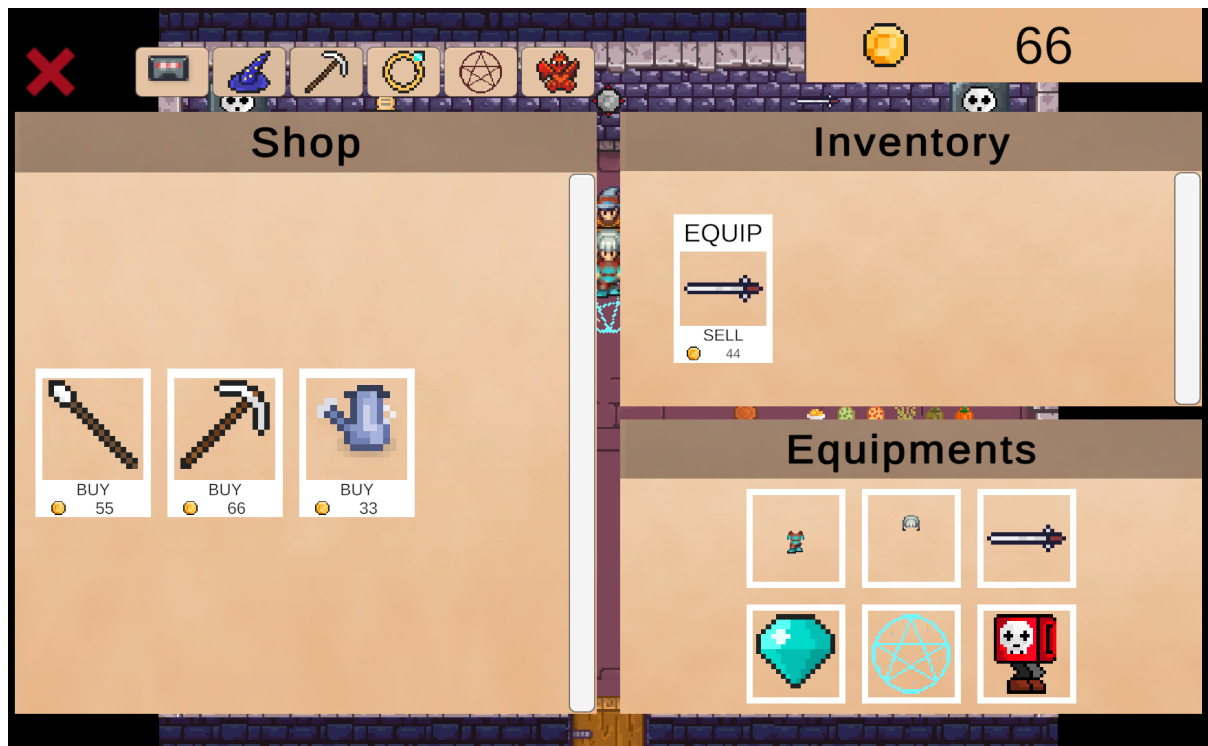
Mouse/Left-Click: Interact with Store Interface

Esc: Exit Store Interface

Interactables:

Door: Quit game

Shopkeeper: Open Store/Inventory



Movimentation of the Player: Using an Interface for Unity components to make possible to migrate to new types of input and movimentation easily, It uses Unity `GetAxis()` to know the direction to move.

Interaction of the Player: The player only identifies the possible interaction, and when the player press E, the other object will send an event or call a local method as a trigger. It also can disable the player controls (in this example the player should not move while shopping)

Buying/Selling items: It was used only the panels from Unity interface and TextMeshPro for texts. Most of the data comes from ScriptableObjects that holds the item information and saves/loads from PlayerPrefs in the local machine. The interaction from Shop and Inventory is made mostly by Delegates/Events making the classes not knowing each other, but sending messages either way.

Equipping items: the Items can be equipped from the Inventory after you bought an item from the Shop. Only the Clothes, Hat and Magic Circle is visible after equipping it to the player. The way used to change the player clothes was using the SpriteLibrary, making a single animation and using multiple sprite sheets.

Currency: The player needs to have the money to buy an item at the Store, at the Top-Right of the screen is the player's currency, for the test, increases the money value each second. Selling items will also retrieve money to the player.

Git: Was used Gitflow, mostly of the work is presented in the develop branch.

Cheats/Debugs:

- Key 0 : Reset all data
- Key 1 : Add 100 of Currency
- Key 2 : Remove 100 of Currency

Process during development:

The development of this project was ideally to make it the most presentable, from feature to feature, following the description in the briefing.

I was worried about the modularity of the features, but turns out that was not the problem. Some simple problems like collision and interactions of the player in the scene took more time than I expected. Breaking down the tasks helped about that.

The idea was to break down the tasks from macro to micro while keeping in the scope of the project and using the Brackeys and Selier the Shaper files/props/sprites helped to just build the necessary to put the ideas to the game. And delegates/events, interface and abstract classes were my preference, so I could separate or change something on the fly and keep the game running.

Personal assessment of performance:

In general, it could be better, there are some things that I wanted to finish, but I needed to stay in the scope of the project. The Store could be better, there are some items that the player equips, but do not show in the model. There's no confirmation of purchase of an item, there's not a link of what the player is wearing and selling that item (selling the item and keep using the item is possible).

But as a test, this is simple and have all the features that I felt essential to prove a concept, this is a good starting point so can be presented to Designers to determine what's the next steps.