

Development Process

The inventory system is primarily composed of the Item and Inventory classes. The Item class contains a reference to the item's owner and a reference to a ScriptableObject that stores the item's data. The Inventory class contains a list of ItemSlots where the items are located.

The equipment system uses SpriteRendererTextureSwap, a script from a plugin called Animancer. It simply swaps a sprite for another in the same position on another sprite sheet, aligning perfectly with the organizational structure of the character and equipment assets I chose.

I used SpriteRendererTextureSwap to avoid creating redundant AnimationClips. However, depending on the game, it might be useful to create a tool that generates AnimationClips from sprite sheets.

The buying/selling system uses a class derived from Inventory to store the items that will be exchanged. In the UI, the idea is that purchases and sales can be made simultaneously.

My idea during development was to create an inventory system flexible enough to reuse parts of the code in the equipment and buying/selling systems for better future maintenance, similar to how it works on Bethesda games (Elders Scrolls and Fallout series).

Because of this, significant refactoring of the inventory system was necessary during the development of other features. Although this initially slowed down development, the flexibility and system integration proved highly useful in developing new features, indicating that my initial goal was achieved.

Except for AudioManager, which I got from an old project, and SpriteRendererTextureSwap, which is from Animancer, all the remaining code was written during the 48 hours.

I believe my performance was positive, although I would have liked more time to improve code readability, create more UI prefabs to facilitate future changes, and add more details such as quality of life features, animations, sounds, HUD information, etc.