

Design Architecture Overview

During the technical interview for the ClothesShopSimulation game, I really put some thought into the design architecture. I wanted to make sure it was organized, scalable, and followed good OOP principles. It was important to me that the player character could walk and interact smoothly with the game world.

To handle interactions with the shopkeeper, buying and selling items, and displaying item icons and prices, I used Singleton components like the EconomyManager. It made things easier by keeping track of the in-game currency and handling all the economic operations.

I also made use of interfaces to keep things flexible. For example, I implemented the IInteractable interface to allow the player character to interact with different objects in the game world. This way, if we wanted to add more interactions later, it wouldn't be a hassle to integrate them.

To keep the UI clean and organized, I separated the functionality into different classes like ShopPanel and InventoryPanel. It made it easier to work on specific UI elements without messing up the whole interface.

For managing the items, I used ScriptableObjects. They allowed me to create and modify items without writing a ton of code. The ItemData ScriptableObject was especially handy for managing item properties and making changes without breaking everything.

In the end, I was pretty satisfied with the solution. The architecture seemed to work well and met all the required features. Of course, it's not perfect, and there's always room for improvement, but considering the time and resources available, I think it turned out pretty solid. It sets a good foundation for future development and expansions in the game.