Documentation for Interview Task: Functional Clothes Shop in Simulation Game

System Overview:

I have created a functional shop within a simulation game inspired by 'Stardew Valley'. Here it is on Github. To be honest I've never played 'Stardew Valley', I only watched gameplay videos. The game employs a top-down view. It features a player character capable of walking and interacting with the in-game world, a shopkeeper for item transactions, and a variety of items that can be bought, sold, and equipped. These equipped outfits are visually represented on the character.

Thought Process:

During the interview, my thought process primarily revolved around creating an engaging and interactive game environment. I have made a small block diagram in Miro for code architecture.

The main features are:

Control system: I have implemented the Unity Input System where I have assigned several keys for walking (WASD or arrows), interacting with other objects (E) and opening inventory window (I). I consider this system quite flexible even for applying this game on other platforms.
Player Character: I implemented player character movement using Unity's built-in character controller, ensuring smooth navigation.

Characters system: for both player and NPCs I created a base class UnitBase where I store the main logic of character which includes movement and obtaining the items. By using subclass UnitPlayer I apply controls and some other functions for the player's avatar.

Item System: I designed a simple inventory system with item icons, their quantities, and prices. This involved creating item data structures and implementing buy/sell functions. I have added two scriptable objects: ItemHolder and LoadOutHolder. The first one is for storing all possible item types in this game, taking into account their parameters. The second one stores the loadouts for NPCs, which can be used to easily equip NPCs at the start of the game.

Outfit Equipping: The key feature was making purchased outfits visible on the character. This involved creating a character customization system and syncing it with the inventory items. You can equip the helmet or the uniform in the Inventory window.

UI Design: I designed a user-friendly UI that displayed the player's inventory, equipped outfits, and shop interface.

I used the Event Bus and Mediator design patterns for improved and convenient communication between entities at runtime.

For this prototype, I used pre-made art assets to expedite development, but all the code was written from scratch during the interview period. I ensured modularity in the codebase, making it easy to expand upon in the future.

Performance Assessment:

Overall, I am satisfied with my performance on this task. However, It took more than 48 hours. I implemented all the required features, and the game is functional and playable. The codebase is flexible. If I were to revisit this project, I would focus on adding more clothing options and improving the visual aesthetics of the game. I have left some TODO comments with my suggestions for extensions.

I think this project demonstrates my ability to design and implement game systems efficiently and to create an engaging player experience.I am excited about the potential for further development in this direction and would welcome the opportunity to discuss any potential improvements or extensions to this prototype. I'd be happy to get any feedback from your esteemed team.