Technical Test Report

Submission Instructions

Movement: Use WASD keys.
Open Inventory: Press the Tab key.
Interact: Press the E key.

(Gamepad is supported)

System Overview

For the technical test, I built a shopkeeper interaction system in Unity. This system includes several features: interacting with the shopkeeper, buying and selling items, displaying item icons, showing item prices, and equipping purchased outfits. Here's a brief explanation of each feature:

Shopkeeper Interaction:

I created a **C# interface** called IInteractable to handle interactions. This interface makes it easier to manage how players interact with the shopkeeper.

Buying/Selling Items:

I developed a **basic inventory system using ScriptableObject** classes: SO_ItemData for item details and SO_InventoryData for inventory management. The TransactionManager singleton **manages all the buying and selling logic**, ensuring that adding and removing items is handled correctly.

Item Icons:

Each item has an icon (Sprite) that is shown in the UI, making it easy for players to see what items they have.

Item Prices:

Every item has a set price displayed in the shop, so players know how much each item costs before they buy it.

Equipping Purchased Outfits:

Players can buy and equip items like hats, tops, and pants. Once purchased, these items can be worn by the player, and the changes are visible on their character.

Thought Process

During the task, I took a systematic approach. I started by defining the main components of the system, aiming for modular and reusable code. Using ScriptableObject for items and inventory made managing assets in Unity simpler. Implementing a singleton for the TransactionManager helped centralize and streamline transaction processes, avoiding conflicts and ensuring consistency.

To improve:

The system can be improved using more efficient data structure for the inventory, also, creating a more complex system for transactions between inventories, that don't rely on singletons. During the development I notice that a POO design can be applied. I use it on the slot logic, but I think it can be explored more to make the system more efficient.

UX can be improved too, with options for drag and drop, gamepad support on UI and overall a better flow, i take inspiration on the stardew valley UI, but the UX is not there.