LSW Interview Task Documentation

Content:

About

During the development of this prototype, I focused first on outlining and designing the core of the main systems i.e. Equipment, Slots, and Stores. Along with that was finding character art that could fit the framework (assets used are cited in the last section), all the while keeping in mind that this system had to be scalable, so that if time allowed it, I could add new features, which was the case for the color picking and hair styling.

Equipment System

The bulk of the prototype lies in the equipment system. This framework allows for customizable pieces of clothing and hair to be created and visualized on the characters.

Slots

The Clothing.cs file contains the enum for the slots:

```
8 referências
public enum ClothingSlot { Left_Hand, Left_Lower_Arm, Left_Upper_Arm, Left_Foot, Left_Lower_Leg, Left_Upper_Leg,
Right_Hand, Right_Lower_Arm, Right_Upper_Arm, Right_Foot, Right_Lower_Leg, Right_Upper_Leg, Hip, Belt, Body, Chest,
   Headwear, Left_Shoulder, Right_Shoulder, Hair, Facial_Hair}
```

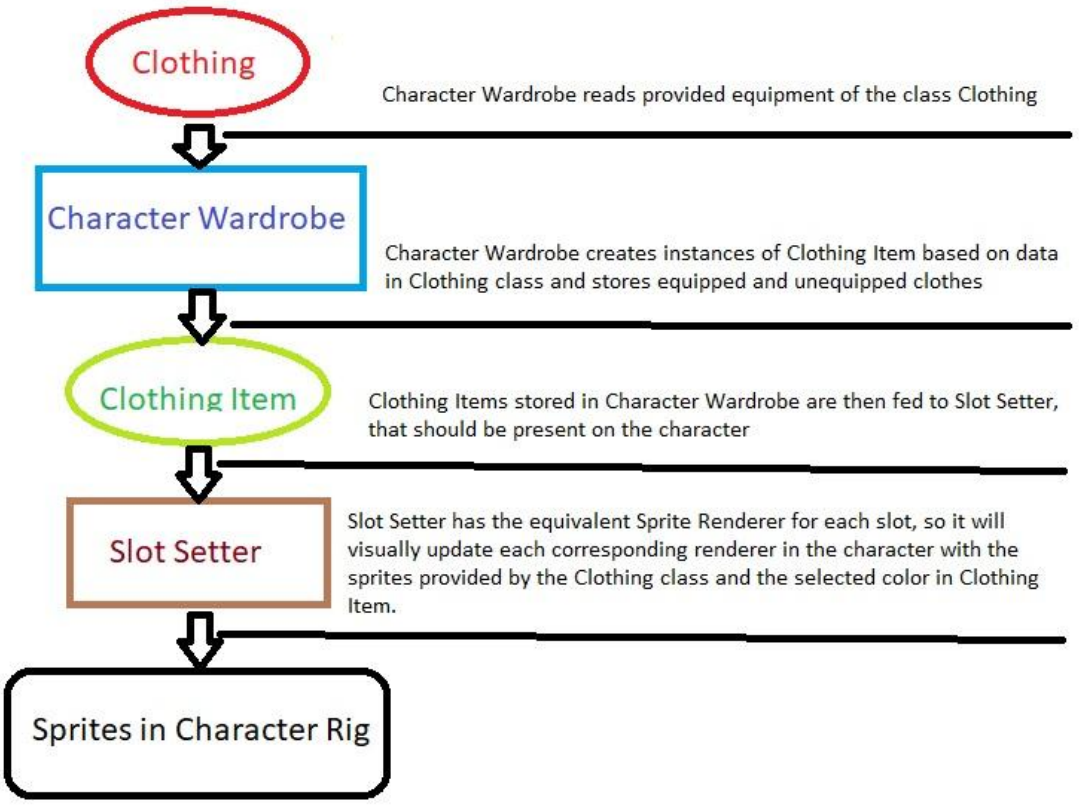
This delineates the possible slots that created equipment will occupy. The classes “CharacterWardrobe” and “SlotSetter” handle the assignment of the equipped clothes to the slots on the character.

Clothing

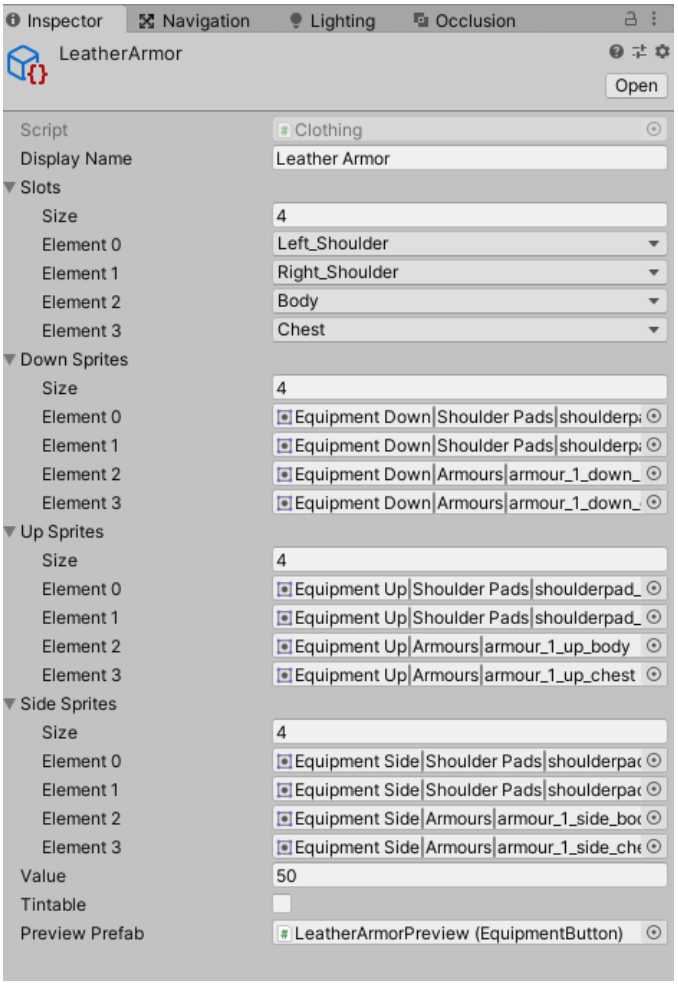
“Clothing” is a class that inherits from the ScriptableObject class, and is intended to serve as data reference for objects of the class “ClothingItem”, which will serve as the instances of clothes that will be accounted for the player’s inventory and for visual representation. New clothing can be created by right-clicking in the “project” tab and selecting “Create -> Clothing”

Guide

Visual representation of the framework:



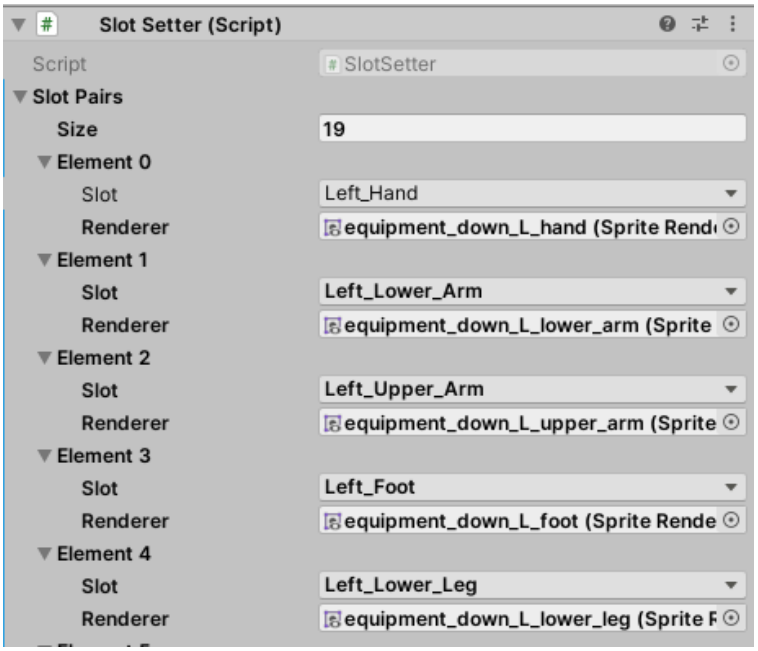
To create new items, first create a new instance of Clothing:



The size of the arrays must be the same, and they must hold the sprites corresponding to the correct slot in each of the different orientations the character will face.

Then, in you character, Add CharacterWardrobe and assign starting clothes and tints, if any piece is tintable:

Then, add SlotSetter to each orientation the character will face and reference them in your CharacterWardrobe:



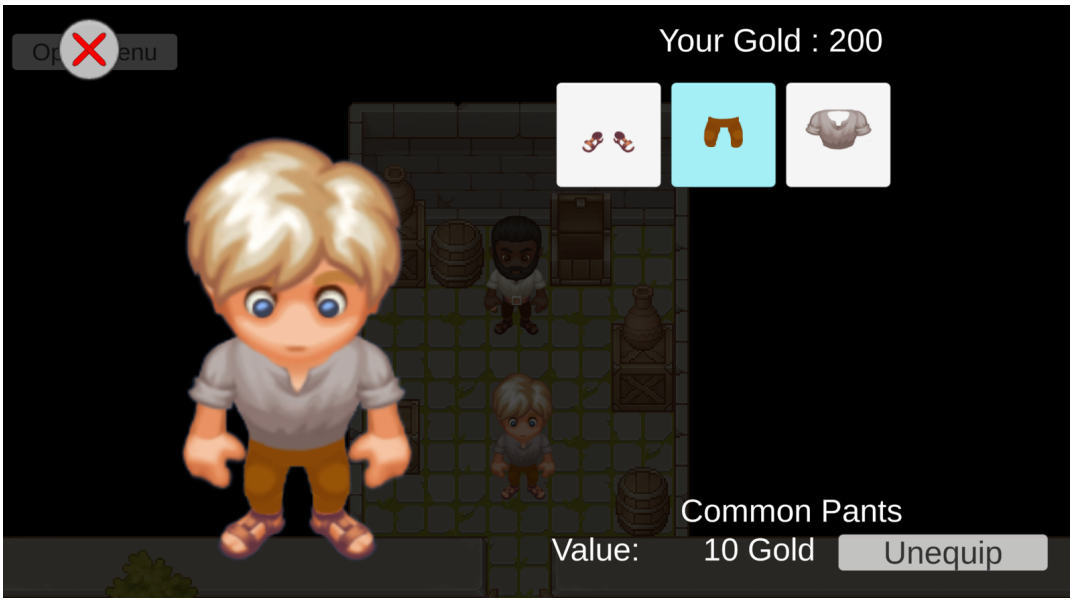
Then, in each SlotSetter, reference each sprite renderer with the corresponding slot, and you're set.

Stores & Inventory

These are the elements of the framework that manage and display inventory management and store interactions:

InventoryMenu

Displays clothes in character's inventory. It shows each item's information, like value and name. It allows the player to equip and unequip individual pieces.

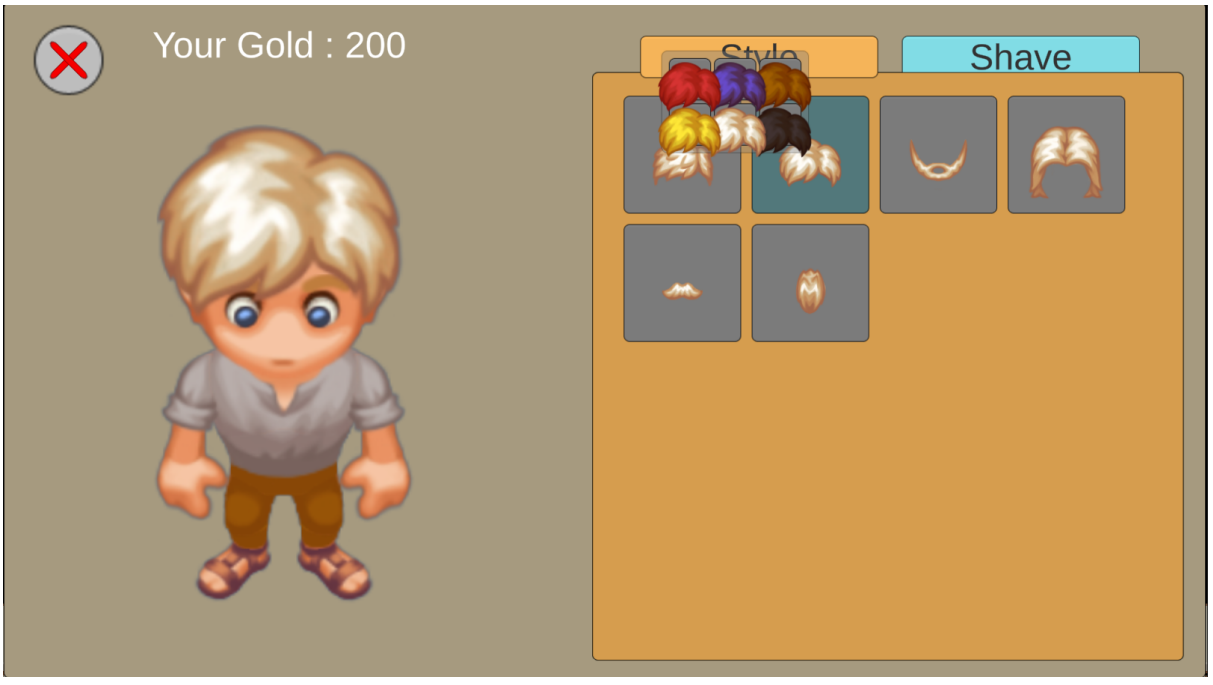


StoreManager

Is a manager for the purchase and selling of equipment. It checks the player's gold and increases or decreases it based on the operation. It also lets the player choose colors for clothing that can be tinted and asks the player if he wishes to equip the recently purchased equipment.

BarberMenu

Inherits from StoreManager, and while similar, it manages transactions differently to reflect the nature of the intended use. It manages the list of hairstyles in CharacterWardrobe instead of the list of equipment. Items purchased here must be instantly equipped, instead of added to an inventory. Also, selling here means excluding the selected hair from the player, which counts as a service, so it also costs money.



Conclusion

I think I achieved most of the objectives set out by the interviewer, I even went the extra mile with things like color customization and the option of choosing hair, which wasn't required of me. Provided more time, I would focus on optimizing these systems, including things like pooling and more effective object usage, and would improve usability in the clothing creation pipeline. That aside, I am pretty happy with the final product.

Assets Used

This prototype used the following Asset packs, and where they were used:

- Cinemachineby Unity Technologies (Camera Behaviour);
- DoTween by Demigiant (Scripted Animation);
- 2D Customizable Character by Daniel Thomas(Character Art & Animation);
- Pixel Art Top-Down Basic by Cainos (Environment Art);
- Ultimate Sound FX Bundle by Sidearm Studios (Music and Sfx);
- Interface Sound Effects by Cafofo (Sfx).