Installers
Installers register the singletons in the game using dependency injection, for example, the UIInstaller will register the UIManager as an available subscription and any other class will be able to inject the same UIManager in them.

Said installers were coded using Extenject extension

Managers
Managers are major classes that, as the name implies, manages their respective part of the game; they can be injected in other classes to access their content using a singleton pattern and they work as a mediator for them. For example the only way to access a screen is by using the UIManager. Some managers require world object references, such as the UIManager. they're given by their respective installers

Items
Items are scriptable objects that can either be the currency or the clothes. An Item has the following parameters: **Name, Type, Quantity**. A Visual Item also has **Sprites Parameters**, and a **Price**. all items are able to be equipped and will appear in the game as long as they're in the right folder. Note that only 1 currency item is accepted, if there are more than one, only 1 will be used when the game loads.

Name: Item name
Type: Item type (enum)
Quantity: Used most for testing, is the quantity the player currently has, can be left as 0.
Sprites: For visual item animating and display, Needs a sprite sheet, a base sprite for mannequins and an Icon for UI visualization
Price: the amount of currency the player needs to have in order to buy said item

Karen's Wig is an item used for testing purposes and it starts in your inventory.

Jobs
jobs are scriptable objects that allow the player to get money in the game, they're translated to a visual "Job View" in the JobScreen (interact with carpet), a job has the following parameters: **Name, Button Text, Reward** and **Item Requirement**.

Name is the job title.
Button Text is what will appear in the work button Eg: "Yodeler Singer"'s button says "Sing!"
Reward is the amount of currency the player will earn by working.
Item Requirement is what the player needs to be equipped in order to be able to work.

Player

The player is able to walk by using WASD keys, and interact with interactable objects in front of him by pressing E, equipped **Visual Items** will appear on the player.

Signals

Signals are used for notifying subscriptions that something happened. Eg: **OnJobExecutedSigna**l will notify the **ItemManager** that the player has worked, and therefore, earned money. the **ItemManager** will then add the reward to the **PlayerInventory**, which will Fire **OnCurrencyChangedSignal**, notifying the **CurrencyDisplay** that it should change its value.

Said signals were coded using Extenject extension

Entities

Entities are world objects that the player is able to interact with, there are 4 entities in total:

Carpet opens the **WorkScreen**, allowing the player to work
Shopkeeper opens **InventoryScreen**, allowing the player to sell items or refresh mannequins
Mannequins opens **BuyItemScreen**, allowing the player to buy a visual item
Closet opens **InventoryScreen**, allowing the player to Equip/Unequip items

This time, I mainly focused on quality of life and bug fixing as the game was basically finished. Added a few animations using Lean Tween, a few more sounds and screen updates, including the brand new **TutorialScreen** and a tooltip when the player can interact with something. One important mention was the pooling from **InventoryScreen;** It wasn't working at all