I made a paperdoll system. So i have the base of the character with animations and other two components. One is the body that will hold the body clothes and the head that will hold the hair and the hats.

All the clothes has it own animations, so i made a sync in the Player class to make sure all the animations will play together.

After that i made the shop, to do that i created the scriptables objects to hold the necessary data to swap my animators and sprites during the runtime. The shop loads the scriptables and instantiates ui elements so the player can interact with them. In the shop is possible to equip an item too. I made the inventory after so i can use the shop methods to play events in my InventoryManager class to make everything easier for me. In the case of need more similar implementations in the future ShopManager and InventoryManager both are an inheritance from the CharacterCustomization class.

To have more items i decided to make colors, so i can have more scriptables and items for the gameplay.

With all these created i started to decide about the background of the game, so i searched for some ruins assets and made a tilemap to apply in the world. Used some assets with rigidBody so the player can push some around just for fun.

The interactions are all handled by the Interactle.cs file. So when the player enters in the trigger collider is possible to see the text in the player and pressing E on the keyboard will call the method Interact which is different along all the interactions. Some of them make dialogs, open the shop and give the player coins to keep buying clothes in the shop.

With all these setted was just a matter of sync some animations for some npcs, phrases and cool sounds to make everything more alive.

The dialogue is handled by the class DialogueManager, so if any instance of the game needs to use the dialogue it’s easy just call the PlayDialogue() method from the DialogueManager.cs and it will spawn the text and after some time it will erase the text from the screen. The time is a parameter, so it’s easy to change that too.

I made a quit panel just to be more organized to exit from the game.

It was very fun to create these systems and all these gameplay elements.