Documentation.

The Player object has 5 MonoBehaviours which are:

- Player: Has a Mover class that controls the player movement
- PlayerAnimationManager: Controls and change the parameters of the player animations.
- PlayerClothesManager: Stores the player Clothes and starts the Shop negotiation.
- PlayerDialogManager: Starts the Available Dialogs in the scene
- PlayerInteractionManager: Starts the Available Interactions in the Scene

In a separate Object with a Canvas resides the UiInventory witch receives the signals to change clothes and send them to PlayerAnimationManager and PlayerClothesManager.
Also, in the same objects, there is the MenuManager, which controls the volume of the scene and has an Exit Button.

The Outfits of the Player stored on ClothingPiece ScriptableObjects that stores the outfit Icon and RuntimeAnimationController that will be used by the PlayerAnimationManager.

To interact with the ShopKeeper and the UiInventory they listen to the UiClothingPieces signal when their buttons are clicked.

All of the inputs that the player has are being controlled and managed by the InputManager with static methods and events.

The shopkeeper has a ShopKeeper class that controls and shows the Shop Ui. Also contains a Speaker class witch will interact with the PlayerDialogManager.

Every Dialog is a ScriptableObject stored in the Dialogs folder. Each Dialog contains a List of DialogBits witch contains an Icon, and the string to be shown.

All Dialogs are shown on the screen by the UiDialog a ***Singleton*** that manages the Ui and key presses to advance the Dialog.

The IIntractable and IShop are interfaces, made to interact with PlayerInteractionManager, and PlayerClothesManager respectively.

The SoundManager witch is a ***Singleton*** that stores and plays all the Sounds in the scene, by any object with the ID of the desired sound.

At last, the GenerateAnimations witch is the *only* piece of code that was *based* on an old tool I've made to automate animations creation. This is used to create the AnimatorOverrideControllers witch are part of the outfits.

Thought Process.

During the making of these systems and Scene, I was, at first, thinking “I got to make this game as efficient, decoupled, maintainable, and expandable as humanly possible!” this mindset led to me getting confused and making some mistakes on the go. Later on the development with some of the core mechanics ready I started to think “Wait, your code can be as clean and efficient as you want… If you don’t finish, that won’t matter!” so I was trying to finish everything I planned in the time I got, also maintaining a minimum of code cleanness.

My Impressions.

I think I well, considering it was my first time implementing most of the specific systems.
But, the first thing I would do to this code is, refactor it, and that demands time, and time to think. Also, to help the designers, I would do some custom interfaces.
One other thing that bothered me was my naming conventions, which are consistent at most, but when it came to the clothes and outfits, I got confused.
Overall I think I did a good job, and with other people around to talk to, and some more time, I’ll fix my flaws, and further improve my qualities.