Documentation

For the assigned task, I began by planning the project's structure using an MVC (Model-View-Controller) framework. In this structure, the Model represents the game's data, the View represents all the visible assets of the game, including in-game elements and UI, and finally, the Controller contains the logic needed to execute the necessary functionalities.

Let's start by explaining the contents of each area:

Model (Data): Before beginning, I want to clarify that I used ScriptableObjects to store information and provide easy access to it, which also helps in organizing variables. The Model consists of two elements. The first one is PlayerData, which contains information about the player, such as their current clothing with Player_Cloth and the number of coins with Player_coins. The second element is ShopData, which refers to all the clothing items that will be available in the store, categorized by section (Up, Mid, or Down).

View: To construct the View, various libraries of free assets were used to complement those provided in the task. They don't have complicated logic. It's worth noting that the environments are created using TileSets.

Controller: For implementing the store, two controllers were created, namely PlayerController and ShopController. PlayerController manages three different scripts with specific functionalities: Movement, Coin Management, and Clothing Management. On the other hand, ShopController contains the core functionality for making purchases and sales.

In general, it wasn't a completely easy task, as creating the items for each clothing piece was a bit challenging due to the system I created, which adapts the purchased item with animations. (This was partly difficult because I was unfamiliar with the asset library used).

Regarding the development of logic, the use of object-oriented programming and script modularity greatly facilitated achieving the objective.

If there are some variables named in Spanish, please excuse any inconveniences.