The movement was made using a Rigidbody2D with the function .MovePosition(Vector2 position), with a Vector2 created for the player input, it handles the animation and the direction of the player.

The character is divided into four parts (Body, Torso, Hair, Legs), each part has its own Sprite Render and its own layer in the Animator with basically every same Blend Trees, the only difference is the body part.

Every interaction is made by a box collider that's on trigger to detect the player and pressing the letter "E".

Available interactions:

- NPC on the bar, building on the left side of the garden, there is a small dialogue with him.
- The green tent, in front of the bar, place where you can change your clothes.
- NPC on the white house, top left side of the map, there is a small dialogue with him and is there you can buy some clothes.

The shop system:

Uses a Singleton pattern (can be removed); a List<> for the items that are going to be sold/bought; a GameObject used for the template that's used to display the products; a GameObject in the Canvas to be used to store the items; a Button that is defined to be buy/sell; and a GameObject that's contains every object of the shop.

On the Start() is created a variable to storage the amount of items on the store. Uses a for() with the variable create, on the for() is instantiated the template in the GameObject that contains the items of the shop, using the same GameObject instantiated the information about the item and check if the player already has the item to define the button to buy or sell.

There are two functions for the button, both receive a value for the item index: the buy and sell functions. For the buy function, check if the player has the money in case, so it is spent subtracting the value of the item and setting the Boolean to true that the player owns the item, and the button is changed to the sell function. For the sell function, add the item value to the player money, remove the item to the player unchecking the Boolean to false, and change the button to buy the item.

That's basically how the shop works.

The clothes system:

There are two scripts for the clothes: one on the player, and other where you can change the clothes.

The manager script:

There are 4 public variables for the script, ScriptableObject CaracterBody. The next three are arrays and all of them are string. They define the types of body, the character states and the direction. The private variables are Animator, AnimationClip, AnimatorOverrideController and a class that I made to override the AnimationClip.

This script only manages the animation of the body to be determined by the clothes that the players has equipped. The main and only function on the script override the default animation clips with the character body parts. Works in this way: Uses a for() with the body types array and get the current body

part and its respective id. After that there is another for() inside of the first with the character states, inside this second for() is get the character state and another for() that gets the player direction and storage the direction in a string inside the for(). I used a Resources.Load<> to get the current animation from the player body and override the default animation with the correct body. In the end of the function, below all the for() is used the AnimatorOverrideController to apply the correct AnimationClip.

The script to change the clothes:

There are two serialized variables: the ScriptableObject CharacterBody and the class that I made for to storage the body part name; List<> of body part options; a Text for the name of the body part component, and the current index. this class is called BodyPartSelection.

Every function receives an int for the part index, with the only exception of the add/remove functions.

There two functions that's used in the shop script to add/remove the cloth that's available clothes to the player.

One function is a bool function that return false in case the part index is bigger than BodyPartSelection variable, otherwise return true.

Two other functions are used in the UI to change the clothes: both are basically the same, the difference is a variable, one adds one and the other subtract it. First, there is an if to check the part index case is true, check the current body part index. In case is bigger than the body option from the BodyPartSelection, increase the current index from the body part otherwise set the current index to 0, and call the UpdateCurrentPart(partIndex).

This UpdateCurrentPart(int partIndex) is the last function of the script. The only utility of this function is to update the name text and the character body part. To update the name uses the BodyPartSelection with the partIndex and get the text to set to the body part name from the body part options. The body part that is updated is basically the same place of the name.

That's how every system in the mini game works.

During the process I tried to make the shop system very dynamic in case it was going to be used in the Little Sim World to be very adaptable in case that's required to add more option to buy or sell, the resolution was to use ScriptableObjects because would be easy to store the items variables and easy to add in the shop.

The cloth system was more difficult because it was necessary to separate every part of the character in different Sprite Renders. I made with the same thought of the shop system to be dynamic in case of using in the Little Sim World.

I think I did well. It was a good game to do in such amount of time and I would like to have more interaction with the game world, however I learned a lot during the whole process.