Lucas Mallo - Blue Gravity Studios Test Documentation

How to Play:

- **Movement**: Control the player character using the "W," "A," "S," and "D" keys.

- **Interact**: Interact with the shop NPC by pressing the "E" key.

- **Open Inventory**: To access and manage your items and clothes in the inventory, press the "I" key.

- **Change Clothes**: Change clothes by opening the inventory and dragging a piece of clothing that you own to the equipped slot, and vice versa.

Code Documentation:

- **Player**:The player character prefab employs a basic MVC (Model-View-Controller) system. The Controller class manages inputs through the Player Input system, with System Actions for each input that allows callbacks to subscribe to them. The Model class handles the physics and logic of the player, including movement and interactions with objects in the environment. This class subscribes its functions to the input events of the Controller. Player stats, such as speed and interaction radius, are set using a scriptable object named "PlayerData." The View class manages animations and controls the four character animators, differentiated by an Enum to facilitate changing clothes.

- **Shopping and Selling**: The Shop System consists of two primary classes:

ShopController Class: This class is inherited from the "Interactable" class, which allows players to interact with it using inputs to open the shop UI. It also utilizes a scriptable object to define the items available in the shop. Each item is represented as a separate scriptable object, containing the item's sprite, sell price, and purchase price.

ShopUiController Class: This class handles the creation of buttons for purchasing items and selling items from the player's inventory.

- **Inventory**: The PlayerInventory class stores references to purchased items and equipped clothing. The PlayerInventoryUi class retrieves this data and populates slots for both clothing and items. This class manages clothing changes for the character: when you drag a clothing item from one slot to another, it swaps the data between them and triggers a callback to the View class to update the character's animator based on the Enum representing the clothing data.