This is my attempt on the interview task, the system is composed by a Player facade with every component of the player being a isolated class for easy maintenance and easy expansion if ever needed in the future, the PlayerAnimation takes care of the animations and to swap animations for the equip/unequip effect using Paperdolling. The PlayerMovement moves the player using rigidbody physics for interaction with colliders in the world, the PlayerInteraction handles non-movement input and keeps a list with everything Interactable around the player, every interactable has an Approach and an Interact methods for custom behavior. The Shopping system is managed by the ShopManager, a singleton instance that does operations on the PlayerInventory, which holds information about the amount of cash and the items currently owned by the player. Both UIs have systems to instantiate ItemViewers, which display items prices and icons with a single Initialize call.
My thought process to this task was to evaluate the subtasks that would take longer, and then start on the most visible and crucial systems (like the Player Systems), and progressively go to the less visible systems like the interactables and dialog popups for aesthetic purposes, I tested some approaches to movement like using transform or rigidbody and i decided using rigidbodies for easy implementation of collisions with the scenario, i wanted things to be as modular as possible and the scripts to be as small and as independent as i could to follow SOLID standards.
I am very proud of the result, i think it checks all the boxes and i had some time to polish and make it pretty, i also could do in a very chill pace which is always good in tasks like these, i only would point as a improvement my initial reaction to the task, because of the amount of systems i did panicked a bit but after doing some UMLs the whole system became much easier and doable in time.