Valentin Gaudio Task - Unity Developer

Hi! I'm going to explain how i've made this task:

Gameplay:
I started with the player's movement, so i've created a **PlayerController** script that handles the player Input with the UnityInputSystem, and it also handles the movement of the player and its animations.
I wanted to make a state machine for the player so I made one, but it wasn't working so well so I didn't want to waste more time on that. But a state machine is a good option if the player gets more animations.

Inventory:
Then we have the **PlayerInventory** script, it handles the logic for the inventory, and it has a reference for the **InventoryUI**, so we update the items in the UI using methods from the **InventoryUI** script.
PlayerInventory is a singleton, we use it in a few scripts like **ItemController**, that handles the itemPrefab so the player can pick up the item and **PlayerInventory** can add it to the inventory.

For the inventory slots we use the **InventorySlot** class, and we make a list of this in the **PlayerInventory**, with a maximum number.

In this Inventory we can add items, swap items, use items, delete items, and if we have 2 of the same type, we can add the quantity between them when we swap them.
Each Item has a stack number, it's the maximum number of quantity that the item can have.

For the Items we use scriptable objects, and we also have an ItemDataBase scriptableObject, where we put the items in an array and we can get the item data from there. **PlayerInventory** has a reference for ItemDataBase since we use it to load the items using the item ID.

When we use an item, it has effects on the player, like JumpForce +1 , Speed +1, etc.

For the Save and Load System:
There is a script **InventoryPersistence**, it has 2 classes inside to handle the json structure.
When the game starts, the call **InventoryPersistence.**LoadData() , and if we want to save the inventory items, we use **InventoryPersistence.**SaveData().

Extra features:
Added a camera movement that follows the player obj.
Added an AudioManager for the sounds.

My personal thought:
I think I can make it better, like, replacing the InventoryUI reference inside the PlayerInventory for events, and making a state machine for the player, adding some particles and enemies and npcs…
It can be a good roguelike/metroidvania game.
It took me a long time with the inventory, i solved some errors like dragging an item and closing the inventory and so on.

I'm going to continue with this project because I've never done an inventory like this, and I like how it looks.

Have a great day!