

Player Controller

My thought process at first was to first do the player movement, then with the player walking and moving around I could make the character interact with objects.

This was the first part: build the character controller and not think about animations. The character can be moved using the good old arrow keys or the usual W, A, S and D keys.

Interactions

The player can interact with anything with the Interactable behaviour.

Interactable is an abstract class with a virtual Update, so it needs to be inherited and override the Update method, but calling the base virtual method inside it.

The class PlayerController checks if there is an interactable object nearby (using a circle radius). As soon as an interactable object is detected the player can press the space key to interact with the object.

By default the Interactable class does not prevent player from moving, but it can be made in a more specific way inside the class that inherits from Interactable, which is the case for Seller. It's specific for when the player interacts with npc, it will call the player method to stop movement until it says otherwise.

Inventory System

With the character moving and being able to interact I could worry about the inventory system. The inventory would be to store items that are owned by the player and keep track of them, then I could do a buy-and-sell system as a complete trading system, storing and removing items.

Initially I thought about only storing the item identifier in a list of integers, being as simple as it should be (now I think it was way too simple).

With this inventory we can add items(its identifiers), remove items and check if a specific item exists in the list. Should be enough.

Buy-and-Sell system

With interactions and the inventory system working I had everything I needed to get to business with the trading system. Being straight to the point: I made it using mostly the Unity UI system and built-in input module.

I updated the inventory script and added coins that'll be used to buy items.

The items to buy are displayed in a scroll view component. I had to use a [script](#) found on github to update the scroll view position as long as the player selects an item from the disposal list, the scroll component focuses on the item, so the player doesn't need to use the mouse or move the sidebar of the component, and use just the keyboard.

All the “buy and sell” is handled by the StoreItem script, it'll handle buy-and-sell only for a specific item. This script contains information about UI, price of item and type of item.

With this script a item can:

- be bought and added to inventory if player has enough coins, removing the amount from player inventory if so;
- be sold if player has the item, removing it from the inventory, and the player receives the amount of coins of item's price;
- be used if the player has the item (as long as we'll need to equip the character with the items I thought about making an “use/wear” option that is accessible inside the store UI).

The animations

Now that almost everything was done I felt like I needed to put some sprites and animations because the “equip system” is deeply based on it.

I can't make beautiful art so I searched on the web and found a nice free character to use. I cut the sprite in three parts: head, torso and legs. They will be individually placed to create the feeling of items and different clothes.

Obs.: I am used to using art assets made with Aseprite, because there is a plugin ([Unity Aseprite Importer](#)) that imports the file and automatically organises animations from the file, so it makes it easier to just import an Aseprite file and have the animations without much effort to initially set them.

Clothing system

When I cut the character in three (head, torso and legs) I was thinking about each part being a “wearable part”, then the character can have many torsos (see as infinity colours and sprites for shirts), many legs (pants), and heads (can include hats).

Head, torso and legs are items, each one of its kind. The player has a default head, torso and legs represented by the identifier number 0 in the PlayerController script.

The player game object has three child objects (represent the parts) each one with its own animator controller component. Those wearables are basically animator controllers that can be changed at run time.

The animator controllers are kept and set by the StoreItem script (that must be in the item UI and set via inspector).

How well I think I did

I think I did pretty well. It's a working prototype and delivered with everything that was asked for in the interview task.

I admit that I can even have fun playing this prototype and changing clothes.

I think I probably could have done better and used the inventory better if I had more time this week. But overall, I think I did pretty well on this task with the time I had.

The only thing I'm not fully satisfied with in this prototype is the inventory being too simple. (But hey! It works!)