Test Project Process

Bailey Bruketta

My first step was to draw out all the core classes I would be using for this test. During production, I utilized half of what I drew up. Some I deemed unnecessary, and others, such as the day and night cycle/ clock, I decided to forego as I had decided pursuance would be superfluous.

As i did not currently have the 2020.1f.2 editor, I went ahead and built initially inside of a 2019 version. I began by setting up a capsule and mesh to act as character, and mapped input through the gamemanager class. I am not quite happy with the movement, but I decided not to mess with it too much as there are many ways to set up movement with a camera, and they all vary in goal and style.

After this was done, I began making the classes to manage the store's inventory, which I decided would remain static due to time constraints and a desire to return work promptly. Would the requirements had been "thrift store," this would have been accounted for, and I built some functionality into the cart class to account for this. This is, should an item be successfully added to the cart, the cart class returns an integer to the UImanager class. This would indicate that an item could then be removed from the store inventory.

In this way, the current state of the store is akin to an infinite vending machine. I would have liked to set up a system to populate the "racks" with meshes representing the available items. If I were to do this, I would utilize the methods used in the "refreshStoreMenu" to fill an array of meshes at the start of the game, and remove said items after a player takes an item into their cart. Items such as shirts could be stacked, and I accounted for this by adding a second column in the "ItemsAvailableInStore.txt"

I used CSV formatted text files to set up both the types of items in existence, and those present in the store. I like this method as it makes the means by which one can change things very simple. Items can be added to the store by simply writing in a line of text. The UIMaster is able to pull these text files into structs to populate both the player cart, and the store inventories.

After I had finished implementing everything I was to implement, I upgraded the version to 2020.1.2f.1, as it had finished installing. I noticed immediately that my performance during runtime was a fraction of that before. I discerned this to be the fault of the lighting settings, and switched modes to the deprecated "enlighten," away from "progressive cpu." This returned the full performance that I was enjoying previously.

Additionally, I noticed the UI anchoring was different than it was previously. To remedy this, I fixed anchors that had not been set up properly. I also amended a single line in the purchasing algorithm to account for a player purchasing something that costs the total amount of their available money.

Once I was happy with the way things were running, I built and tested a few times, changing a few minor things, before zipping and shipping.