WWU - Computer Science Department
CSCI 497M/597M Virtual Worlds
Milestone 2 Report – Spring 2020

Marketplace

Julianne Sem, Terence Tan

Introduction

The goal of this project is to implement a marketplace where users can buy or sell each other's items. A marketplace is a common feature in Virtual Worlds. It's similar to eBay. In a marketplace, users can open up a window and view what people are selling or they can sell their own items. It is a very useful feature for users who want to grind money by selling items and for users to buy items that are rare or tedious to collect. Our approach will be to use the Unity game engine, Unity assets, and a database. After our project has been implemented, we will focus on integrating it with the Virtual World Server.

1. Introduction

Motivation

The marketplace is an important feature in many virtual worlds. It helps increase user interactivity in the world. This feature will give users another option to buy or sell items besides going to vendors. Users will be able to list their items for sale, allowing them to have control over the value of items. Having a marketplace will help motivate users to create and craft items that other users can buy.

Challenges

1. Connecting database to the project
2. Updating the list of items in the database without changing what is currently displayed on the window. For example:
    - if user 1 currently has the marketplace open, and user 2 adds an item to the marketplace, user 1 shouldn't be able to see that item without hitting a refresh button. Another related issue is if user 1 buys an item from the marketplace, user 2 shouldn't be able to buy that same item, even though it might still appear on their screen

3. If user 1 wants to buy an item user 2 is selling, setting up a fast and reliable way to notify user 2 that user 1 wants to buy from them may prove challenging.
4. Writing code that can easily be built on
5. Integration

Reasons:

1. The idea of connecting a database to the project is a new concept and will probably take a lot of time to understand. We will also most likely have to communicate with the server group to setup a database, which might be difficult to communicate
2. This is mainly due to the fact that we are unsure of the exact behavior that will occur when the database gets updated. We also have to make sure that the item gets pulled from the server if a player is trying to put it in their inventory as opposed to the client, or we will run into duplication errors.
3. There is currently no communication system in place as far as we know, so that means that users cannot message each other if they are interested in trading in the marketplace. In other words, we have to build our own notification system from scratch which may prove a difficult task to do.
4. This project will most likely be accessed and modified in future quarters, so making sure that the code written is both easy to read and modify will be important. The code should be written in a way that allows future students to build off of the current code without having to modify too much of it. This means we will have to think about as many errors as possible, and plan around those.
5. Since we are starting off doing our own project that will be integrated later, we don't know what features other groups will add to their individual project preintegration. This means there is a high probability that we will have overlapping features like inventory systems, and figuring out how to integrate those inventory systems while still making sure all projects work will be difficult

2. Related Work

Key Technologies

Databases:

We used SQLite for our database. Initially we were going to implement a MySQL database however, when we started this project, we added the files from the Crafting System which had an SQLite database already created. In the future, the database commands may have to be converted to SQL in order to be integrated with the server.

Assets:

Basic Bandit:

This asset is so we have a model that we can use as the marketplace NPC.

Related Class Projects

Our project on Marketplace relates to these other projects:

- Virtual Economy 1 - This group is focusing on user-user and user-npc trading. Our projects are both related to a virtual economy. More specifically, we will need to communicate with them on how to standardize currency and any other trade rules.
- Virtual World Server - Since we plan on using a database, we will have to figure out how to connect our database to the server
- Resource Generation, Virtual Chef, Dynamic Ecosystem - These groups while different fall under the same category of how our project relates to theirs. Any components they have/create will have to be allowed on the marketplace.
- Avatar Controls - This group is focusing on standardizing the way avatars move and all the controls needed for avatars.

3. Architecture

Requirements

- Create NPC for users to interact with (Julianne)
  - Add simple idle animation to NPC
  - Add trigger when clicked on
  - Lock rotation (if time permits)
- Create Player (Terence)
  - Allow player to move
  - Allow player to look around in all directions
- Build marketplace GUI (both)
  - Receive data from the database when the GUI starts
  - Create two sections: Buy and Sell
  - Allow panel to be scaled
  - Buy
    - Display items for sale
    - User can drag item into their inventory to buy
    - create popup window to confirm purchase
  - Sell
    - Allow users to drag items from inventory to marketplace GUI
    - Create popup window requesting price of item
  - Receive data from the database when refresh button is pressed
  - Display errors to the user when necessary
  - Add search bar (if time permits)
- Create the database (Julianne)
  - Connect database to the project
  - Create queries to update the database when user interacts with marketplace GUI
- Create user GUI (Terence)
  - Add cursor in center of screen so user knows where they are looking
  - Create user inventory system
    - Implement removing and adding items to the inventory

    - Notify seller if another user wants to buy their item
    - Lock other options (i.e. crafting) if trade is occuring
- Integrate the marketplace to the virtual world server (both)
    - Export project as an asset

Architecture or Design Space

So far, our project contains four main components:

- Marketplace GUI

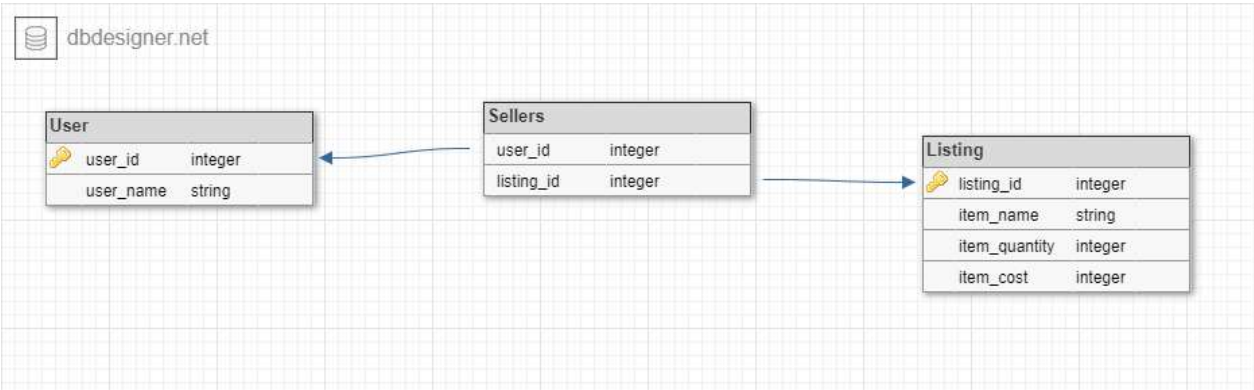
  We haven't implemented the Marketplace GUI, but we do have a plan and a mockup (Figure 1 below). The Marketplace GUI has two sections, buy and sell. The buy section will have the items available to buy listed, as well as the cost. The sell section will display a list of items that the user is selling. There is also a refresh button that will refresh the Marketplace GUI and display updated items that the user can buy. To implement this we will need to create the database and connect it to the Marketplace GUI.

- User GUI

  For the user GUI we are using the Crafting System's GUI to save on time, and allows us to focus on implementing the features we need for a functioning marketplace. If time permits, we will also add a way to notify users that their item has been bought, so they don't have to constantly check the marketplace to see if their item sold. We will also be extending the current inventory panel by adding a currency display so the user knows how much money they currently have. This currency value will be updated via a database whenever user B buys an item from user A on the marketplace. When the user is in range to open the marketplace, a message will pop up letting the user know they can right click to open the marketplace. Figure 3 and 4 shows what we currently have for our user GUI.

- Database

  We still need to finalize a design and create the database. Our project relies on other projects such as the crafting projects and virtual server, which have their own schema and databases. Therefore, we need to talk to other groups and get a clear idea on how everyone else's databases look before we can fully create our own. Below we have designed a simple schema with three relations. This is a draft design and will most likely change in the future.



- Items

  We haven't added the items yet, but we will probably use the same items from the Crafting System project, so we don't have to worry about fixing any errors that occur from trying to use a different set of imported items. By using the items from the Crafting project all we need to do is allow users to drag and drop items into our marketplace panel, which should significantly reduce implementation time of our project.s

- Crafting System Project

  We started building our project off of the Crafting System Project, as this project was closest to what we needed in terms of GUI. We thought this would be a good starting point for our project because by building off of an already existing project, we can design our project in a way that easily works with a past project, which is one of our goals.

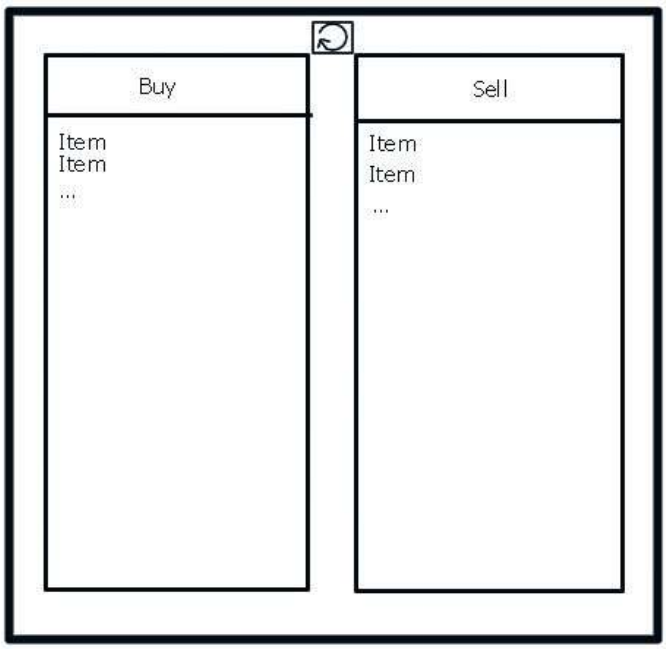


Figure 1 - Marketplace GUI Mockup

There will be two sections titled buy and sell. The little icon at the top is the refresh button in the event new items get added after a player opens the marketplace.

Milestone 2 Progress

Currently we have finished creating the player, implementing player and npc interaction, and importing the inventory system from the crafting project. This is about 30% because we have the basic user functionality features almost finished. We still need to implement adding/removing items from inventory, but other than that the user functionality is pretty much done. For the next milestone, we will start working on the marketplace GUI and database for the marketplace.

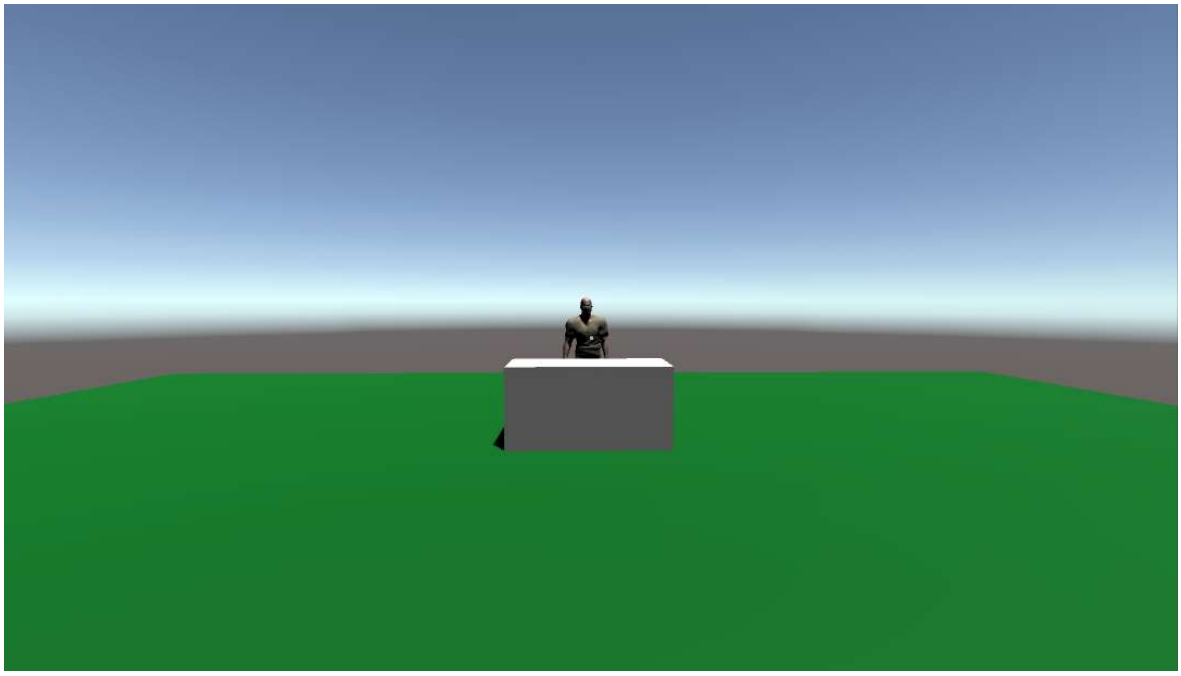


Figure 2 - Player away from NPC

In this figure, the user is away from the NPC and isn't close enough to interact with the NPC.

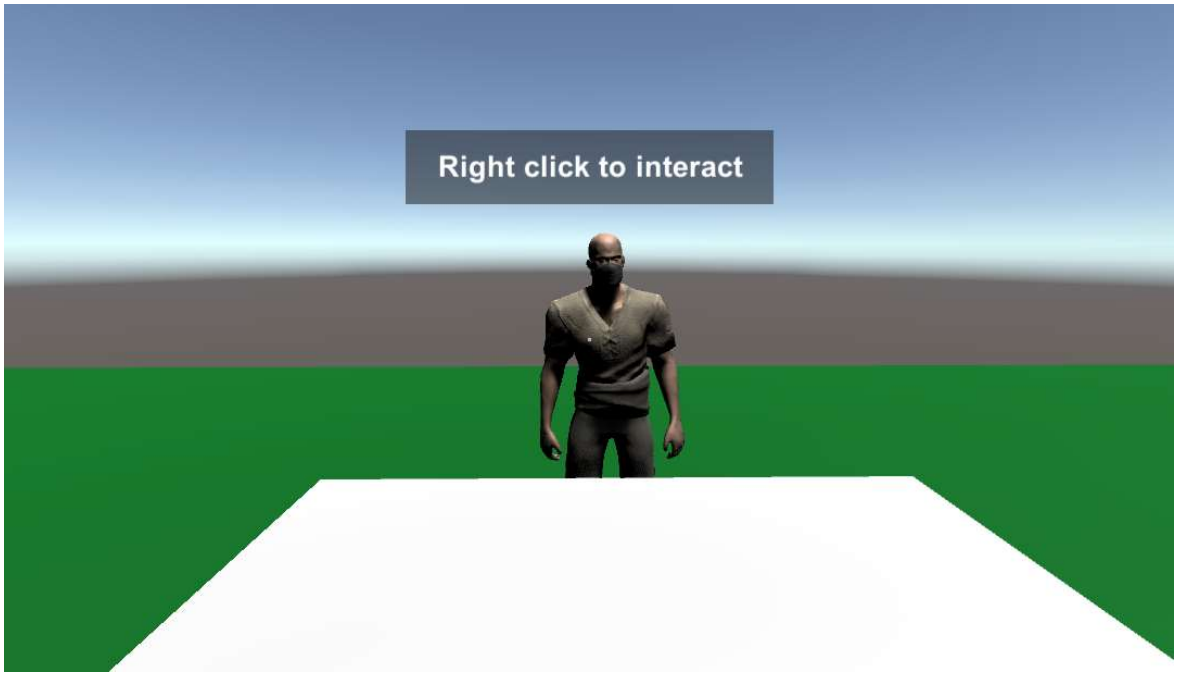


Figure 3 - Player close to NPC

In this figure, the user is close to the NPC. The textfield is activated notifying the user that they can interact with the NPC..

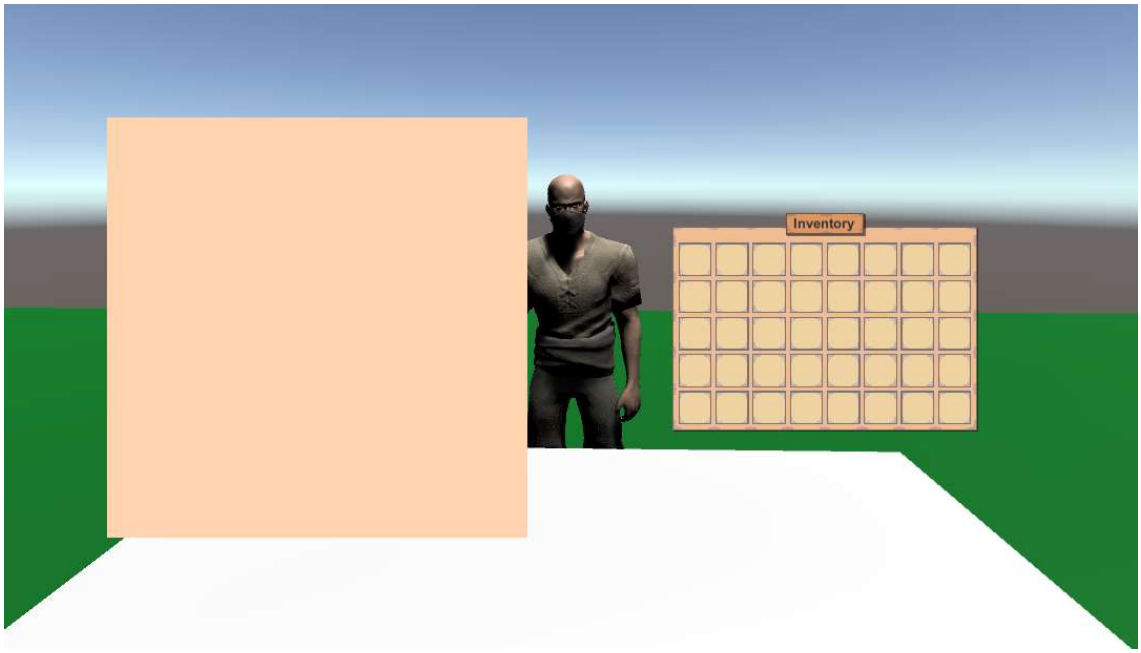


Figure 4 - Player triggered interaction with NPC

In this figure, the user has right clicked on the NPC and the Marketplace GUI and users inventory is displayed. Currently the Marketplace GUI is a blank image, but by next milestone it will have information displayed to the player.

4. Integration Plan

Virtual Economy 1

We've been communicating with Virtual Economy 1. The main issue with integration with this project is coming up with a standardized currency. We are letting them design the currency and they will notify us as soon as they are finished. Another issue is with the database. We need to make sure that we have similar databases to access the currency that the player has. This will help to avoid execution errors and inaccurate currency values.

Virtual World Server

We have spoken with the server about integration when we were planning out our project. They requested us to send our database schema. Currently our database is incomplete, but will be finished by next week. After finishing the project, we will export it as an asset and send it to the server team to import it.

Resource Generation

We haven't spoken with Resource Generation yet. However, before integrating with the server we will need to speak to them about their database. Currently our database is using the same database from crafting system to pull items to sell. When we integrate with Resource Generation, we need to make sure that we can access those resources and allow the user to sell or buy it to the marketplace.

Virtual Chef

We still need to speak with the Virtual Chef group. Before integrating with the server, we will need to make sure that our database can access items from the Virtual Chef's database. Users should be able to cook a meal and sell it on the marketplace. They should also be able to buy or sell ingredients.

Dynamic Ecosystem

Similar to the previous projects, we need to make sure our database for items matches with theirs if they are generating items.

Avatar Controls

We haven't spoken with Avatar Controls yet. However, before integration we will need to discuss which key to use to interact with the NPC and make sure it doesn't interfere with other controls. Currently we have the user right click on the NPC to interact.

Milestone 3 Progress

Since the last milestone, we have added a database, a buy and sell section in our marketplace, items to the inventory, and the ability to drag items into the sell section from the inventory. If a user drags an item from the inventory to sell, a window pops up requesting for the price. If they input an integer, the item gets destroyed. If they input something else, an error window pops up. At this point there are only a couple of things left to implement, but apart from those things we really just need to link all of our components together (i.e. making sure the database is properly manipulating the data, making sure items that are sold are put into the buy section of the marketplace), and then make sure that we can integrate the project.

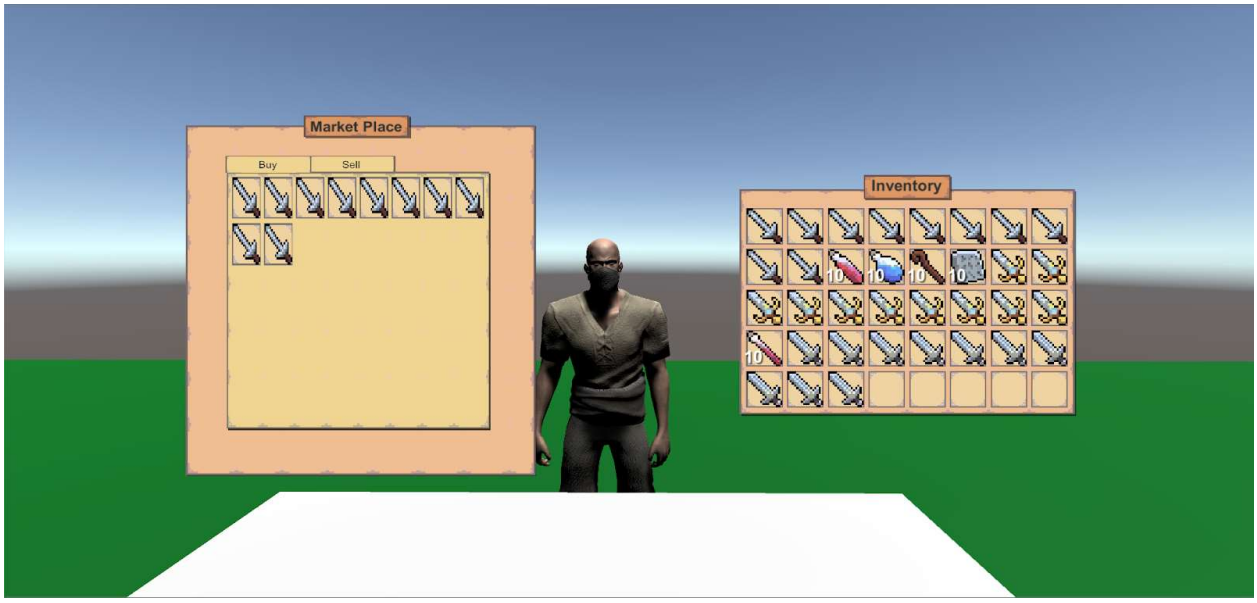


Figure 5 - Buy tab and updated inventory

Here we added a buy and sell section to our marketplace. We decided to use tabs instead of our original design of two columns for a slightly cleaner look. We added some temporary items to the buy section for testing. Items have been added to the inventory as well.

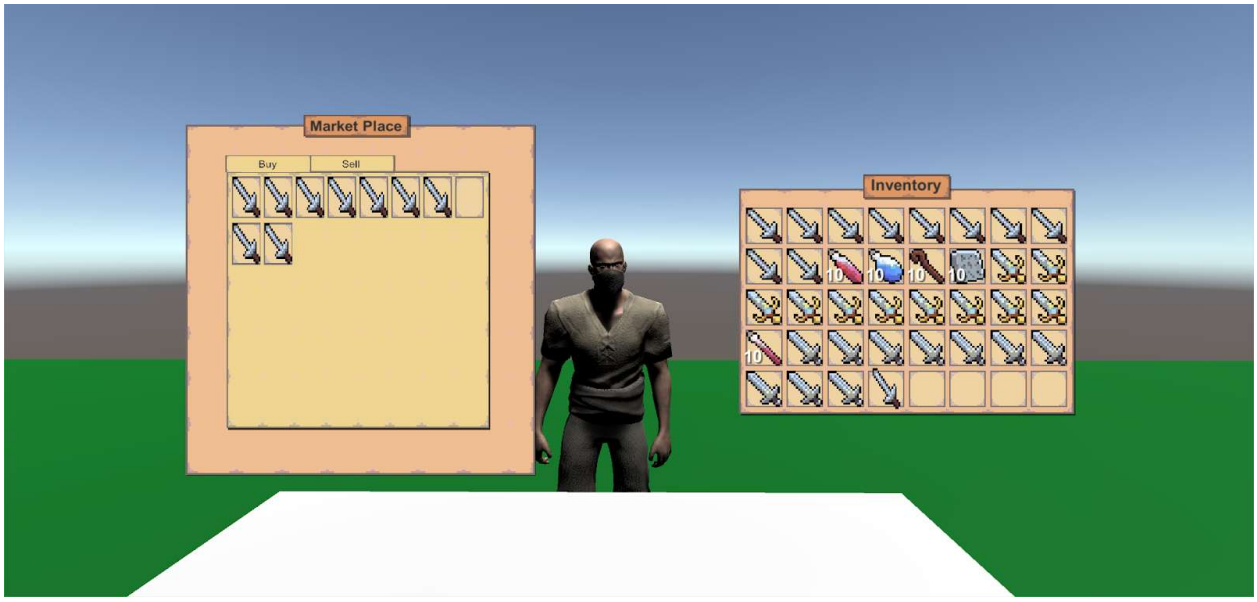


Figure 6 - Adding items from buy

Users can drag items from buy to their inventory. In the figure above, we can see that the item in the empty slot in the buy section has been moved to the inventory.

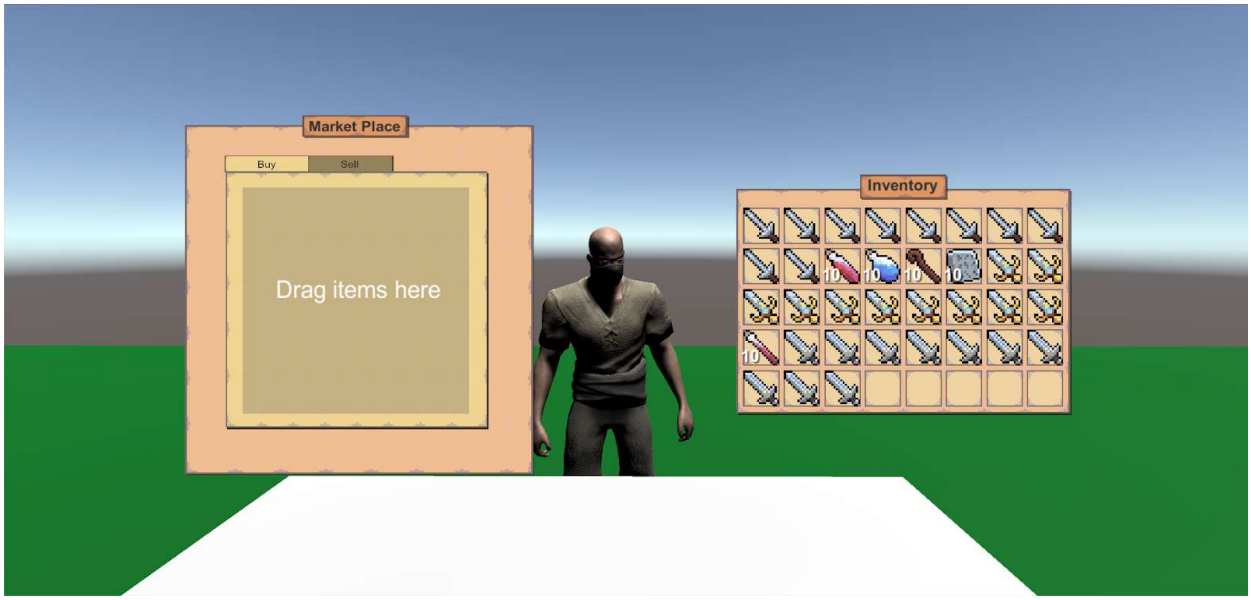


Figure 7 - Sell tab

Our sell tab allows users to drag items to be sold in the area shown.

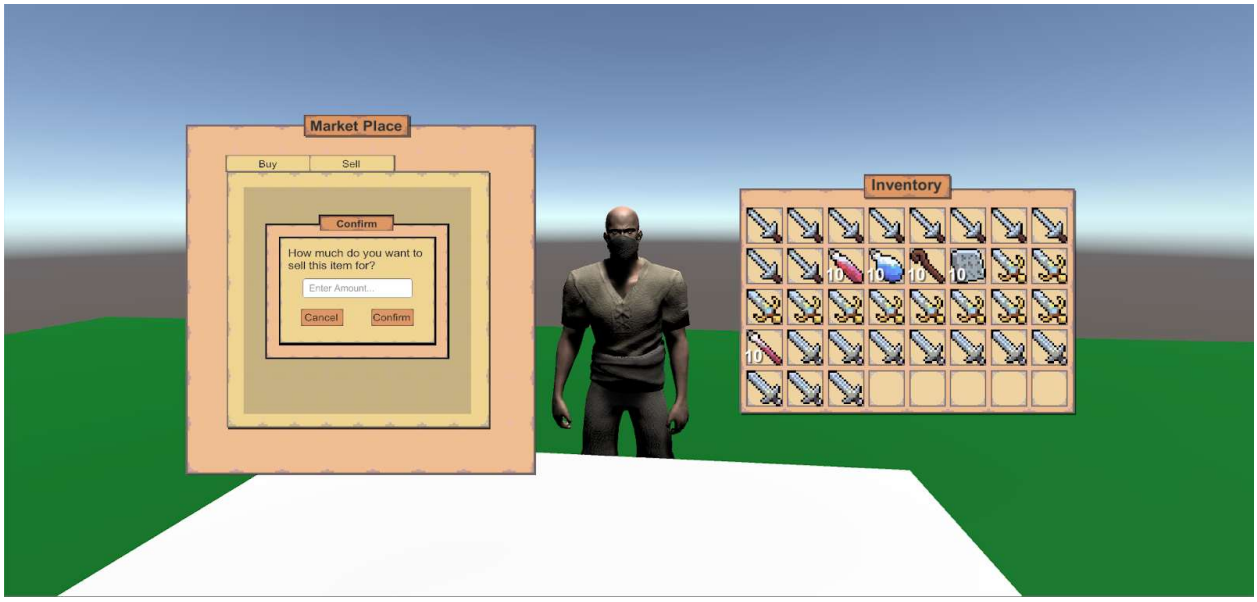


Figure 8 - Confirmation

After dragging an item to the sell section, a popup window appears

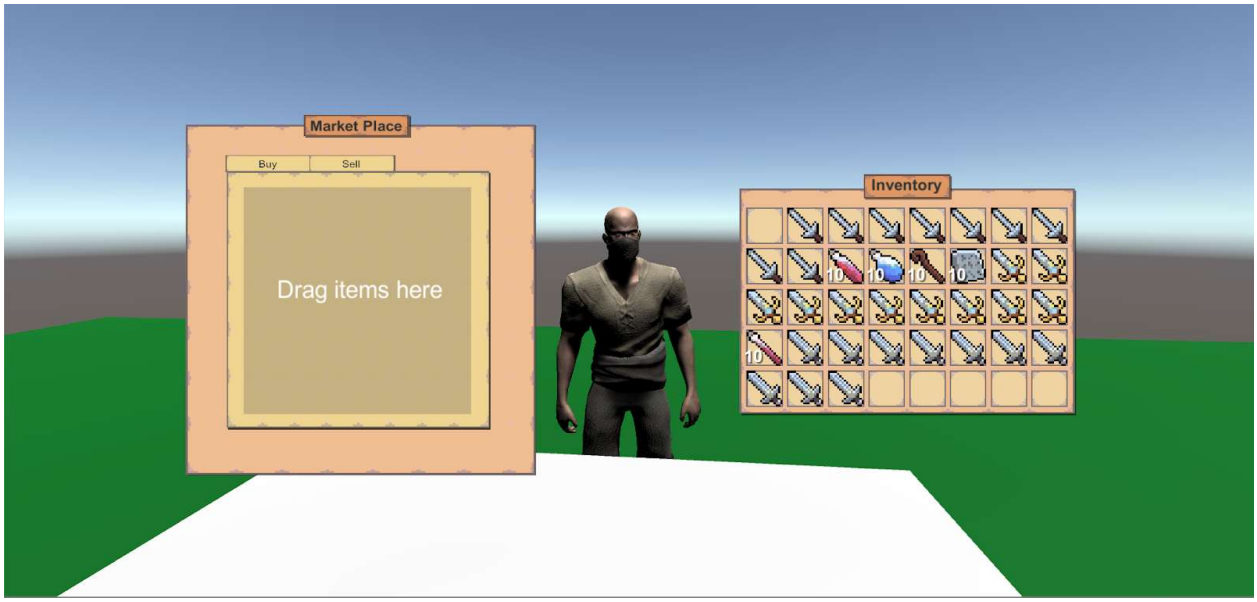


Figure 9 - Selling

Once the user inputs an integer value, the item is sold.

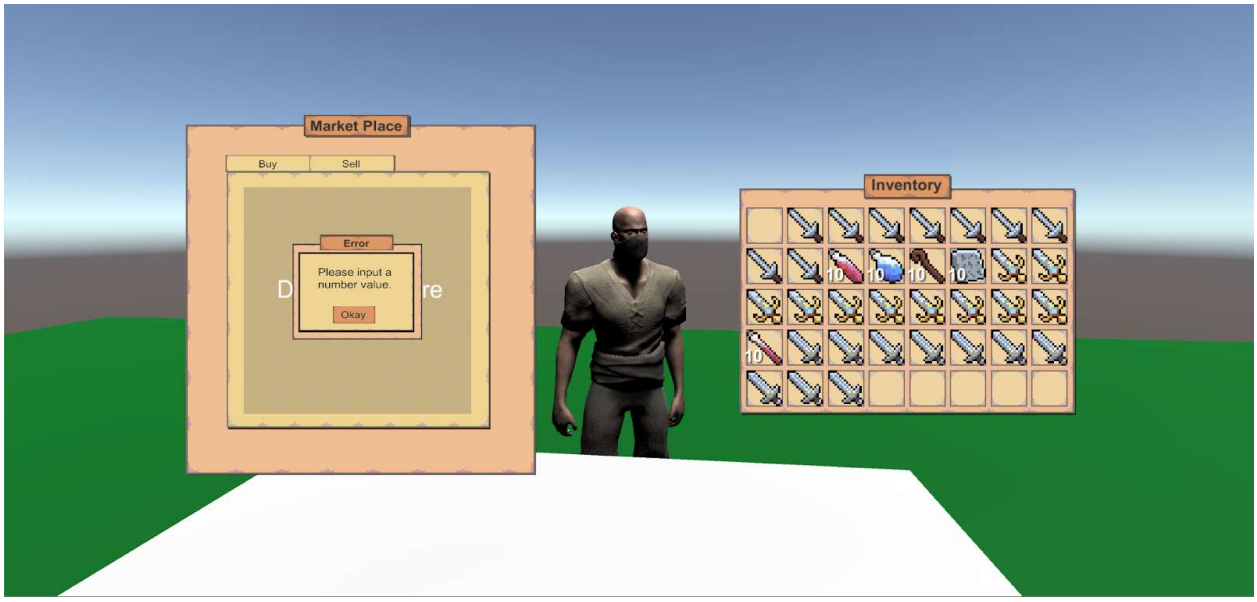


Figure 10 - Error

If a user inputs a non-integer value, an error window pops up instead of selling the item.

Current Progress

Since the last milestone we've worked on the buy tab of the marketplace and started integrating with the Virtual Economy 1 group. Now when the user drags an item to sell in the sell area it will immediately be added to the buy tab with the correct price. Users can now also drag items from the buy section to their inventory and purchase the items. When a user drags an item from the buy tab to their inventory, a pop up window will appear asking the user to confirm their purchase. If they confirm the purchase, the users money will be decremented by the price of the item and another pop up window will appear asking them to move the purchased item to their inventory. Due to time constraints, we weren't able to finish the searching and connecting the database. Currently the database gets updated when a new listing is added.

We've also started integration with Virtual Economy 1 (Vendors). Currently we have both of our projects in the same Unity project and have started creating a scene with a Vendor NPC and a marketplace NPC. However, the scene is still not working yet.

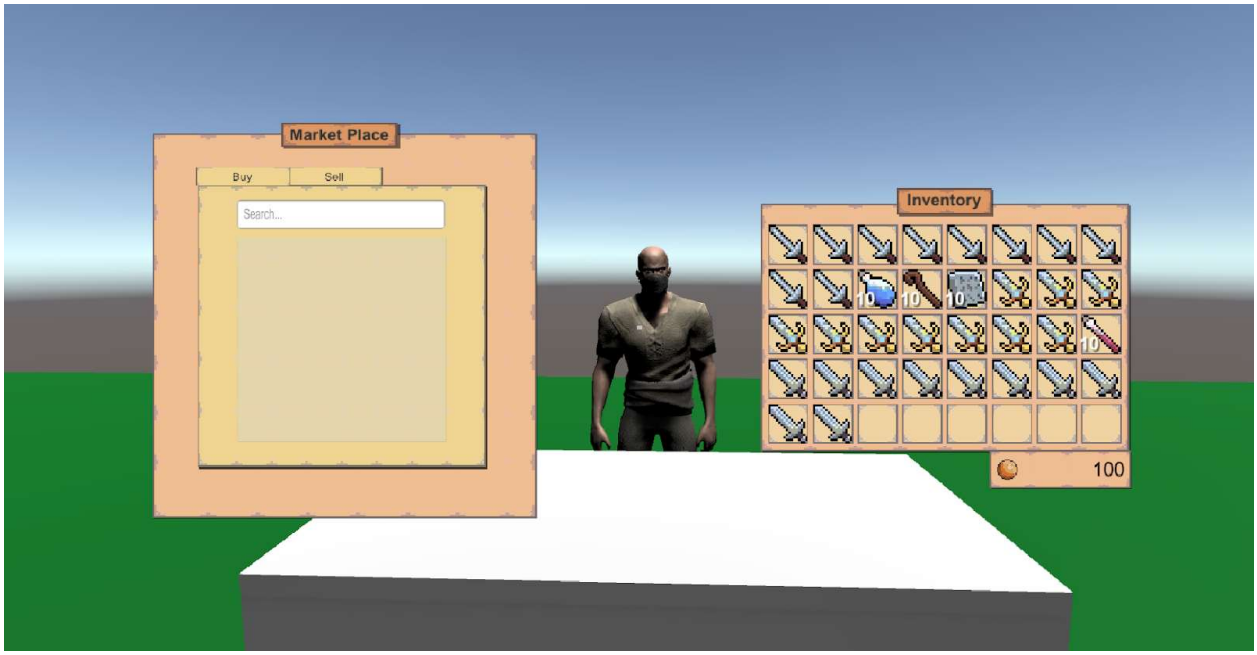


Figure 11 - Buy tab with no listings

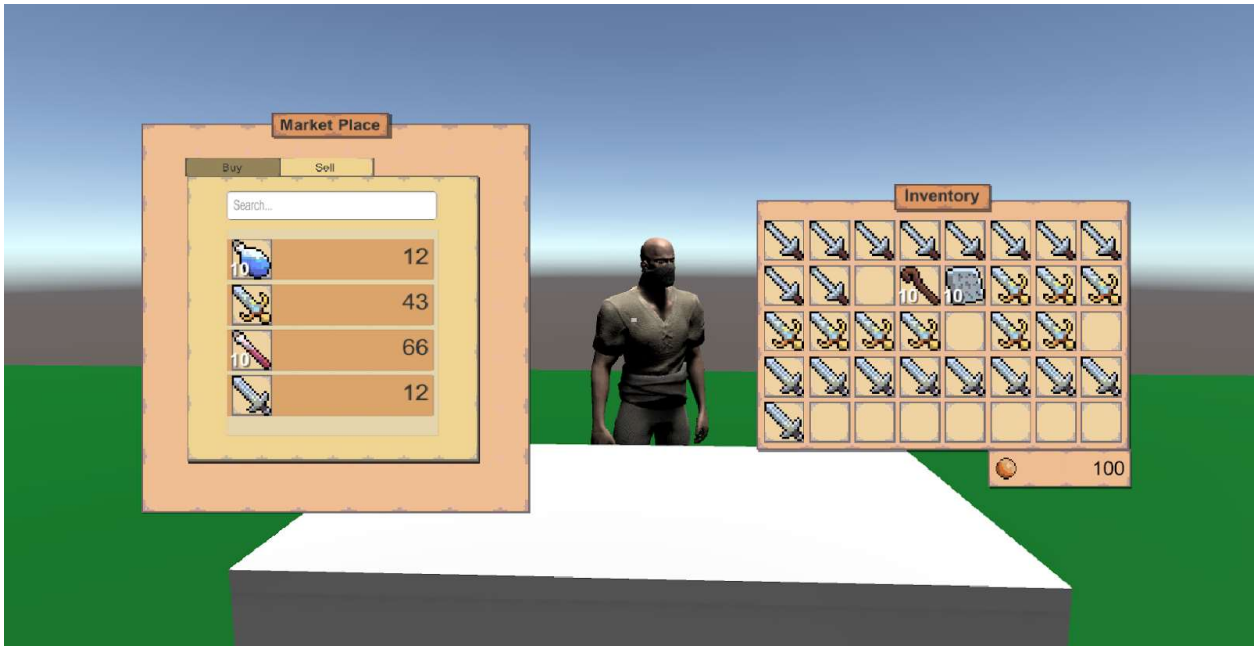


Figure 12 - Items added to buy section

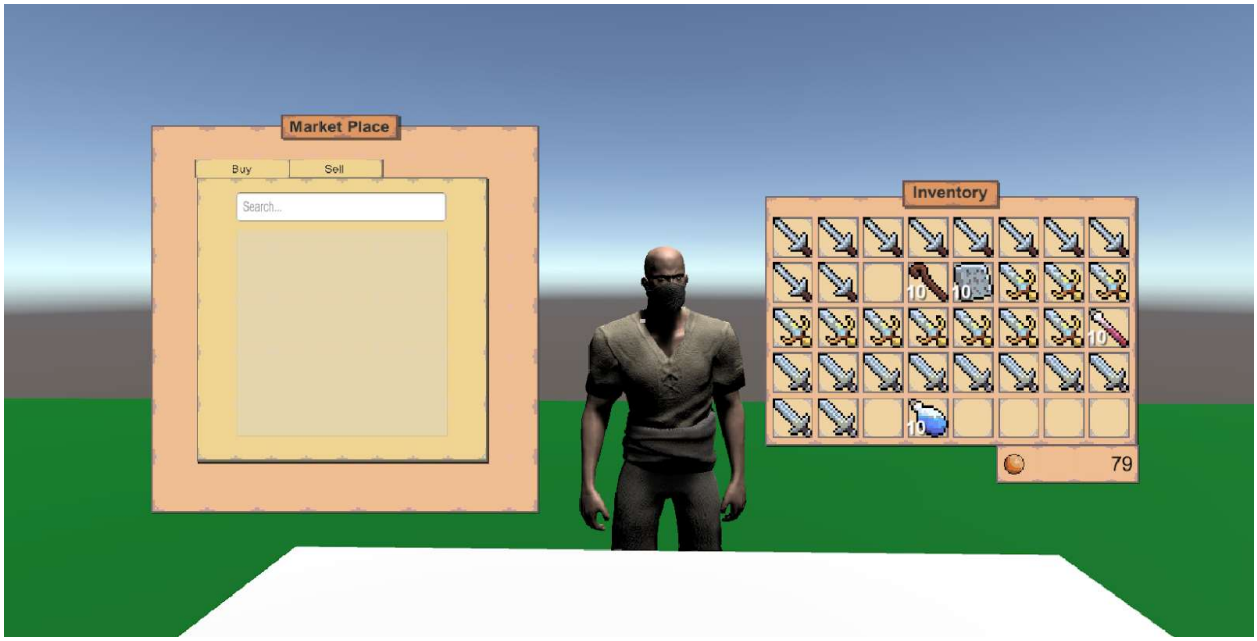


Figure 13 - Item was bought and player's currency decreased

6. Evaluation / Testing

Use Cases

Title: Buy Items

Actor: User

Precondition: The user has enough currency to purchase items.

Trigger: User right clicks NPC and views available listings in the buying section.

Basic Flow:

1. User right clicks on NPC and opens marketplace
2. User goes to buying section
3. User looks through current listings for sale
4. User drags desired item from the marketplace GUI to their inventory to purchase

Alternative Flows:

Description: User doesn't have enough currency

1A1. User receives an error message notifying him that he doesn't have enough money.

Description: User's inventory is full

1B1. User receives an error message notifying him that he doesn't have room in his inventory.

Description: User wants to refresh the page and get updated listings.

1C1. User presses the refresh button on the marketplace GUI.

Description: User tries to buy item that has already been sold.

1D1. User receives an error message notifying him that the item has been sold already.

Post Condition: User has the bought item in their inventory and the cost of item has been taken out of their currency.

Title: Sell Items

Actor: User

Precondition: User has items that they want to and can sell

Trigger: User right clicks NPC and goes to the selling section

Basic Flow:

1. User right clicks NPC and opens marketplace.
2. User goes to selling section.
3. User finds the item they want to sell in their inventory.
4. User drags item from inventory to the marketplace GUI.
5. User types in desired selling price or starting bid price.
6. User clicks confirm to add the listing.

Alternative Flows:

Description: User wants to sell item that can not be sold in the marketplace.

2A1. User receives an error message notifying him that the item he wants to sell, can't be sold in the marketplace.

Post Condition: The item is removed from User inventory and the listing is viewable in the buying section of the marketplace.

Title: Seller receives payment

Actor: User

Precondition: The user has an Item for sale on the marketplace.

Trigger: The user's item has been bought.

Basic Flow:

1. Seller receives notification in chat that item has been sold and the amount of currency they have received from the transaction.

Alternative Flow:

Description: Seller is offline

3A1. Seller receives notification next time they are online.

Post Condition: The item is removed from the marketplace and the seller receives the revenue.

7. Conclusion

Summary

The goal of this project was to implement a marketplace where users can list their items for sale and buy other users items. We were able to get the basic functionality for the marketplace done. In order to create the marketplace, we used the inventory and the database from the Crafting System project. We also used an asset called Basic Bandit to create the NPC.

Even though the basic functionality is implemented, there are still a few bugs and features left to implement. One of the issues is that the user isn't able to move purchased items to certain slots after confirming their purchase. Another issue is that users can get charged twice if they drag an item into two different slots.

Project Value

When implementing this project we've learned a lot of new things. One of the things we've learned was how to use SQLite. Before this project we only knew how to use MySQL. We also learned more about how to use Unity and Unity Collab. Setting up the Unity Collab was a bit difficult, but it was very easy to use once it was working. Besides learning new languages and game engines, it was really interesting learning how to implement a marketplace. There are many small functions that need to work to implement it. It was especially difficult since we were using code from previous projects, we spent most of our time trying to understand their code. We've learned many important lessons from this project that will be important for coding projects that we do in the future.

Future Work

There are a few features that can be added to improve the marketplace. One feature that we did not have time to finish was fully integrating the database to the project. There are two scripts MarketDB and PlayerDB that have functions to access and modify the database. Functions from those two scripts still need to be added to Slot.cs and SellItem.cs. Currently one of the functions to add a listing to the database is located in SellItem.cs, but there needs to be a way to get the playerid into the function. Another feature that can be finished is the search bar. The buy panel currently has a textfield for a search bar, but the functionality hasn't been implemented yet. Another feature that can be added is editing the quantity to sell. Currently the user drags the item that they want to sell in the sell area and it would sell all the given quantities. Instead of selling all of the quantities, another text field can be added where the user input their price and it can ask what quantity they would like to sell. A few other possible features that we didn't implement were allowing users to auction off items, allow users to remove their listings, and implementing categories to make finding certain items easier. Lastly, the project still needs to be integrated with other projects such as the server and resource generation. We have started integrating with Virtual Economy 1 (Vendors), but it is still not complete.