

# CSE 566 – Virtual Reality Spring 2019

## Assignment 2: Basic 3D User Interface

Jinghuan Shang – ID# 112076155

March 26, 2019

### 1 Important Links

Unity Project File: <https://drive.google.com/open?id=1IgBVmBblhr1zDzPy6L4gL79sDaK8gUUL>

Video: <https://drive.google.com/open?id=12knJkptBW7b8Ug00D9PzLDi80ERmpobt>

### 2 Environment Set Up

Software used:

- Unity: 2017.4.21f1
- Google VR SDK for Unity v1.190.1<sup>[1]</sup>

Hardware used:

- VR Headset: VR SHINECON 3D VR Glasses
- Mobile Phone: HUAWEI P10 Plus with Android 8.0
- Controller: VR SHINECON Glass Remote Control

### 3 Directory Hierarchy

Only important files are mentioned.

1	VRLab2\	Unity project directory
2	Assets\	Assets used in the project
3	Lab2Supermarket.unity	Main scene file
4	*.cs-	C# script files
5	[ other directories]\-----	Imported assets files

## 4 Implementation

### 4.1 Supermarket

The supermarket mainly consists four parts: the house, shelves, items and a shopping cart. In order to let the camera for mini map work functionally, the roofs are removed.

The house is from the asset store<sup>[2]</sup>. The shelf is from an online resource<sup>[3]</sup>. The items are from the asset store<sup>[4]</sup>. The shopping cart is from an online resource<sup>[?]</sup>.



Figure 1: Overview of the supermarket. We can see a house without roof, shelves, and items. The main camera is associated with the location of the cart. A mini map is displayed on the upper left part on the screen.

### 4.2 Mini Map

The implementation of mini map is adopted from<sup>[5]</sup>. Roughly, a mini map contains two major parts: A dedicated camera and a raw image to visualize the mini map.

The intrinsic rationale of a mini map is quite simple: it's just an aerial view from above of the player's head. Therefore, I place a camera right above the player(+10 in Y-axis). This is a dedicated camera which is different from the *Main Camera*. By following the method from<sup>[5]</sup>, the world captured by the dedicated camera is shown on a small region on the canvas.

The original way from<sup>[5]</sup> can't serve correctly when starting moving the player. By placing the camera outside the player object and applying same translation to the player synchronously once moving the player, I resolve the problem.

Also, to let the mini map small enough, the roofs of the house is removed to ensure they won't block the view of the camera.

### 4.3 Shopping List

The shopping list is implemented by putting texts and a button on a panel contained in a canvas. The shopping list is bound with button *A* on the joystick to toggle the show it.

There are two important issues here. First, due to the continuous signal emitted by a button, simply binding *A* to both show and hide will cause the shopping list flashing multiple times on the interface. That means the event bound to *A* will be consecutively triggered random times. In this situation, the list will become Schrodinger's cat – you can't predict whether it will successfully show up or hide after pressing *A*. Hence, button *A* is only to show the list. To close the menu, I use another button on the panel with gazing interaction. After gazing 2 seconds, the list will hide. A variable is updated per frame to serve as a timer to implement this gazing interaction.

The second issue is how to trace and show the progress of shopping on the list. To fulfill this functionality, I build a Dictionary, *Current Item in the Cart* to save items in the cart and their quantities. Once an item is added to or removed from the cart, the dictionary is immediately modified. In detail, if an item is newly added to the cart, it firstly creates a key in the dictionary with corresponding value 0. Then add this value by 1. When removing, decrease this value by 1. If the quantity of this item is 0, remove this key from the Dictionary. By this method, the shopping list can trace how many items we have got. Another dictionary is pre-defined in the code to serve as the real *Shopping List*. Its values never change.

The shopping list shows the progress by adding texts dynamically on the canvas panel. The texts are generated by *Current Item in the Cart* and *Shopping List* jointly. The items waiting to buy are always shown on the list while the unwanted items are only shown if the player adds them by mistake.

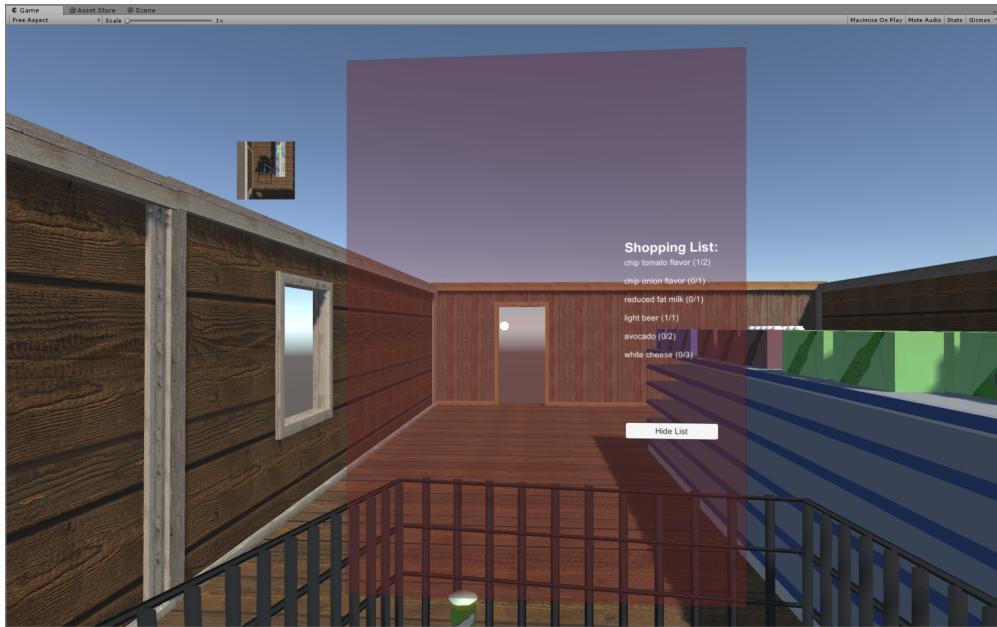


Figure 2: Shopping List

#### 4.4 Proximity Path

The proximity threshold is set 10 in the virtual world. The distance is calculated directly by the locations of the cart and the original place where the item should be. If the distance is greater than the threshold, the line won't be shown.

The path is displayed by an object attaching *Line Renderer*. The start and the end of the path are set when the player is gazing some item on the shelf or in the cart.

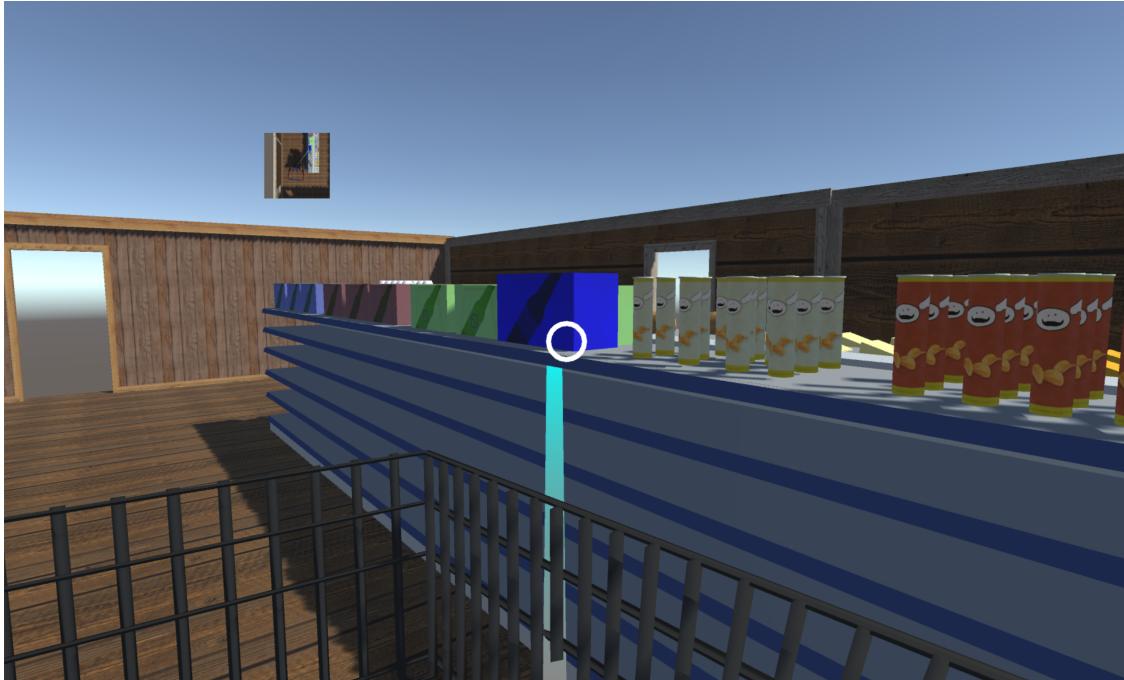


Figure 3: Enlarging the size of the item gazed by the player

#### 4.5 Buy, Return and Move the Cart

Gaze interaction is adopted from<sup>[6]</sup>. An item is bind by two event triggers: PointerEnter and PointerExit. These two events ensure the item to act when you are gazing. It is important to mention that the Google Cardboard SDK<sup>[1]</sup> has already made the work that set your gaze point as a pointer.

The item will be enlarged and change its color to stress the selection.

If the player is gazing at an item, when pressing *B* on the joystick, the item will automatically move into the cart. The return can be performed by the same procedure by gazing the item in the cart.

The cart is binded by 2 another keys *X* and *Y* on the joystick to move forward and right. Hence, in this project I use 4 keys in total to perform all the interaction features.

#### 4.6 Items

Three kinds of items are implemented into 3 different flavors/types respectively.

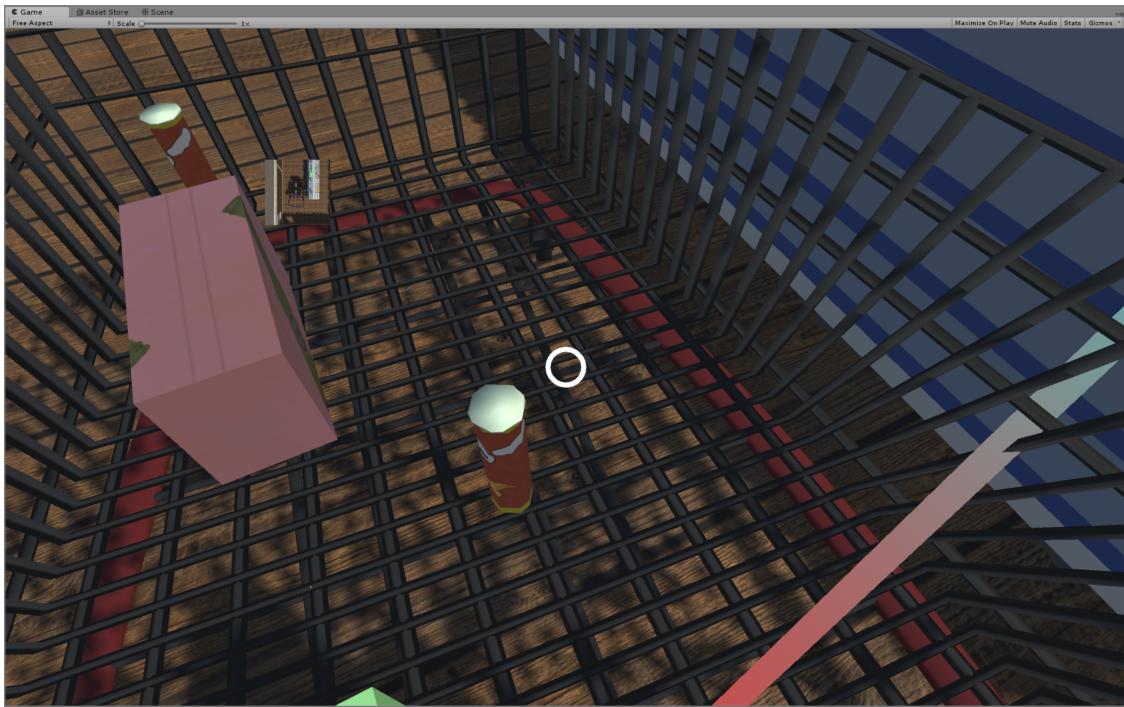


Figure 4: Buy Items



Figure 5: Different flavors/types of items

## 5 Acknowledgment

I would like to thank Xueying Bai and Ying Lu for discussions during this assignment. I also appreciate all the authors of free assets available in the Unity Asset Store.

## References

- [1] Google vr sdk for unity v1.190.1. [Online]. Available: <https://github.com/googlevr/gvr-unity-sdk/releases>
- [2] Mega fantasy props pack. [Online]. Available: <https://assetstore.unity.com/packages/3d/environments/fantasy/mega-fantasy-props-pack-87811>
- [3] Toy gun with shop sheaves 3d model. [Online]. Available: <https://www.turbosquid.com/FullPreview/Index.cfm/ID/1212723>
- [4] Food and grocery items - low poly. [Online]. Available: <https://assetstore.unity.com/packages/3d/props/food/food-grocery-items-low-poly-75494>
- [5] [tutorial] implementing a minimap in unity. [Online]. Available: <https://blog.theknightsofunity.com/implementing-minimap-unity/>
- [6] Unity vr tutorial - gaze timer interaction + teleport. [Online]. Available: <https://www.youtube.com/watch?v=bmMaVTv8UqY>