

ENGR -AD 1000, Fall 2020.

Computer Programming for Engineers

Easelt DECOR

Final Project Report

December 12, 2020

Aman Assali , NYU NET ID: aa8230

Maisha Mahrin , NYU NET ID: mm10294

Eric Asare , NYU NET ID: ea2525

1 .Introduction

Online shopping has made our lives easier but no one can deny the fact that it comes with a lot of disappointments. In many cases, specifically when it comes to home decorating products such as furniture, the delivered item usually does not meet our expectations in fitting to the interior of our home. This is because we tend to struggle to visualise how the specific colour, size and shape of a product will look in our space and make thoughtful changes at the right time before the item is purchased. The fact that online transactions are mostly non-refundable leaves us with no option other than to buy new items to replace the ones we bought hours or weeks ago online. Moreover, realizing that the decision of buying a particular product was not right frustrates many of us. Wanting to reduce such disappointments and frustrations while online shopping, we came up with the idea of creating Easelt Decor.

EaseIt Decor AR application is a smartphone application, functionable on iOS devices, that gives the user a realistic understanding of the item he or she is buying and whether it suits their space. The application comes with a local library similar to an online store which houses assorted 3D models of home decorating artifacts such as furniture. The user is able to select an image of the item he/she wants to buy from the library . The selected 3D model image will be projected onto the real room of the user. The app has functionalities such as positioning, changing colouring of the item and scaling the object up and down to give the user a realistic feel of the actual size or colour she or he would love to buy for his or her room.

2. Project Development

We developed our project on unity game engine platform. We downloaded assets from the Unity Asset Store to get started with the implementation. For the first phase of the application development, we limited the objects to basic furniture needed in every home such as sofas, tables, chairs, beds etc. The first four features of the app that we decided to work on are placement of objects, position, color, scale and rotation.

Plane Tracking:

The project requires the surface of the user to be tracked; hence, we used AR default plane to track the surface from the AR Foundation's Package library, which was attached to the AR Session Origin as a prefab.

Note: *This section makes references to the code used. To see the code, please refer to the project file submitted alongside the report. The position, placement, scaling and rotation enabling of the furniture codes are all under the script "Object Placement". Below the brief description of the specific codes are given:*

Placement and Position:

This feature relies on the user's camera and space captured by the camera. In order to project furniture to the scene the user clicks on the preferred furniture button from the upper InfoPanel. This activates the object and now the user has to touch any point on the plane on his screen to place the object in that position. The user can change the object's position by touching another point within the trackable plane. To stop the object from moving completely, the user can click on the same furniture type button with the pause symbol.

Position code:

The placement and positioning of the objects rely on raycasting. Then a touch on the screen must be detected, if there is a touch the 'GetTouchCommand' is used to retrieve the value, if non the touch is set to default.

The touch position is detected and the ray is casted depending on the position hit by the user's finger. The code checks first if there is a furniture object already present, if not, then it instantiates a placabel prefab on the position touched. A name is given to the

object so that it can be found later when the color of the object's components (children) need to be changed. If there is an object already present then the position of that object will be transformed to the hit position or the area the user touches on the screen. Note: for the object to be placed the touch position has to be on the trackable plane, otherwise the function won't be activated.

InfoPanel (Menu) (USER INTERFACE):

The menu (panel) comes with a local library housing 3D model images. A menu is originally enabled but can be hidden using the 'X' button on the top right corner and reopened by clicking on the '+' button on the top right corner. After the user chooses the furniture models he/she wants to place, a new panel will be displayed on the main Info Panel that has the color option, scale option, and rotate option. The user can then choose the desired feature to edit.

Menu Code

A canvas was created in which a panel was attached to the top section of it. A 'furniture Button' was created with an image that displays the furniture the button places. Another button was created to manage the 'pausing' of the object (so that it stays in place when adding another furniture). Then a panel with the color buttons, rotation slider, and scale slider was created to be displayed as the 'furniture Button' is clicked. The same process was used for every furniture button. OpenPanel and ClosePanel codes are used to set active and deactivate the panels required.

Scale:

To scale the object chosen, the user must first click on the scale text button on the InfoPanel. Then, the user can drag the handle of the slider beneath the scale text to rescale the object to the desired size. The object can get scaled up or down in all axes x, y, and z equally, i.e the length, width, and height of the object will increase at the same rate.

Rescale code:

The scale code transforms the 'localScale' of the furniture to the scale read in the slider. The scale is equated to all three dimensions.

Rotate:

The same mechanism as the scale will be used for rotating the object. The user clicks on a text button with 'rotate' written on and then he/she can drag the handle of the slider attached beneath it to change the angle. The object can only be rotated along the y-axis.

Rotate code:

The rotate code transforms the 'rotation' of the furniture to the rotation read in the slider. The rotation is only applied to all y-dimension.

Color:

The user is able to change the color of the furniture parts separately. For example, a chair furniture would have a base component, a seat component, a back (seat) component, and a metal component (for the legs). The app allows the user to change the color of the most important or prominent features of the furniture in a chair's case the user can change the seat, back (seat), and base colors. If the user wants to change the base's color, he/she clicks on the base button under Color text on the InfoPanel. Then a color slider panel shows on the InfoPanel with RGB sliders. The user can then drag the sliders to try the different colors. Note: positioning all the RGB sliders to the far most left will change the furniture component to a black color, while positioning all the RGB sliders to the far most right will revert the component to its original color. After the user is done changing the colors, he/she can disable the RGB sliders' panel by clicking on the 'Color' text button.

Color code:

The function first gets the children of the parent furniture displayed in the scene by utilizing the 'gameObjectName' and the childIndex. Then the color variable changes the color material of the object by utilizing 3 sliders with red, green, and blue values. The sliders read those values and set the color of the object accordingly as the user moves the slider handler.

App Library

For the current app, we have two pieces of furniture in the app library which comes along in the zip file with the Unity project. We have an arm chair and a bed with pillows, blanket and mattresses. It is important to mention that one furniture is treated

specifically as one during rescaling and rotating. During color changing, the furniture's materials change color separately and can be decided by the user.

3.Results and Evaluation

Several errors and challenges were faced during the development of the project, many of which we have successfully overcome. For example, the color code was not working for the separate children of the prefab furniture, example bed. The bed base, the pillow cover, the blanket, none was changing color. We then realized that because the object was getting instantiated we had to find this instantiated gameobject. It was a bit difficult to figure out the optimal method to do so, but we arrived at giving a name to the object instantiated in the code and using that name to refer to it in the colorModifier code.

Then, our move code was not being a swift drag, it was dragging the objects, but the drag was not smooth. So, we had to add and edit the move code from other code examples while making sure the original code does not get affected. Now, our objects get dragged smoothly without a slight jerk that happened before.

4.Conclusion and Future Work

The project was successful, we achieved our goal of giving the user a real feel of the object to be purchased. We added a small app library on the panel from which the user can choose an object to spawn and to visualize in their space. We added the feature of allowing the user to change the colour of the object chosen giving the user more options in terms of colour choice for the object he or she is buying. The user can also position

the object within the space his camera shows using a touch-screen system. We managed to add the feature where the user on a click of button and then from a slider could scale the object up or down . To avoid covering the screen of the user, the user can tap on a button to show the panel of all the buttons containing the various functionalities.

We want to make our app an excellent one so we have proposed other features we want to develop in the future. These future work would give the user enhanced realistic view and more options to inform his or her decision on the right goods to buy whenever shopping. We have already done thorough research including learning Vuforia Engine to facilitate the implementation of the functionalities described below

The main reason behind our not implementing some of these functionalities such as importing images was due to our decision to implement the more essential customizations first.

We want to add more 3D objects into the apps library so that the user would have a variety of home decor products to use to visualize in their space. We will add options where the user will be able to hide the object by clicking on the specific furniture button. The user can also choose to reset his/her space so that all objects projected are removed by clicking on a 'reset' button. For now, we have separate codes for the scaling, rotation and movement of specific objects. In the future, we want one code for scaling, rotation and movement respectively to work on as many objects the user decides to bring on screen. We would like to also give the user the ability not only position the object but to rotate the object in 360° in both horizontal and vertical directions. We want to even give the user more options by allowing the user to import images from his/her photos and project them into certain objects. For example, if the user wants to buy a photo canvas, he or she can select the image she wants to print on the canvas and

project it onto the canvas to give him or her a visual feel of how the photo canvas would look like when printed. We also want the user to be able to import 3D models or images from websites that support them and project the models to his/her space using the app. This feature involves relying on home decor e-commerce websites that provide 3D models of their products possibly Amazon or IKEA. The user can choose the desired furniture, download it and import it into the app, which will project it into his/her screen, so that he/she can visualize how it will fit. However, we are unsure of how feasible this is yet.

5. Reflection On Learning

The term project seemed daunting in the beginning. We had bright ideas but were clueless on how to bring our ideas into life. Having come this far in our term project means we have learnt a lot.

We have acquired a lot of knowledge and skills. We learnt how to use the Unity game Engine tool which we had no experience in. We learnt how to use the knowledge and algorithm skills developed from our C++ experience to implement on C# for this project. We learnt how to break complex systems down into smaller and simpler levels to achieve the final complex solution in small but collaborative steps. For instance we were able to break down our project into achievable tasks .

Some of the skills we developed were how to research, evaluate and assess the quality and feasibility of information acquired from various sources such as the unity forum, stackoverflow and youtube . We learnt how to use those information to solve the challenges we were facing in our project. For example , we wanted to make the object ,spawn into place, to be moved by touch. The information we found could help us move only spheres but not furniture so we had to work through other sources of information to make it work for our assets. This also taught us code efficiency

Living thousands of miles away from one another while developing a single project,teamwork and communication skills are undoubtedly the major skills we have learnt in this project.We were always communicating our progress and challenges to one another to ensure that we are all on track. For instance when the unity collaboration tool was not working as we wanted it to , we went back to the drawing board, divided the work in such a way that every progress made would be communicated.We also acquired decision making skills. Determining which feature should be added now and which feature should be left for future work. We had to consider the consequences of each of our actions and decisions all the time. And this is a skill that we deem transferable even to our daily activities.We developed other communications skills such as public speaking skills and presentation skills because we had to work hard on our presentation skills to be able to deliver efficiently . We also learnt the bitter lesson that

we should always have a substitute in case something fails at the end but more significantly , we learnt to never give up but to persevere . We nearly gave up when the collaboration tool failed us and we have to start all over again. We persevered encouraging one another and we made it this far.