



University of Toronto
Faculty of Applied Science & Engineering

ECE1778: Creative Applications for Mobile Devices

Final Project Report

April 14th, 2020

Team Place-It:

Yixiao Hong 1001311145
Hongzheng Xu 1001772904
Liren Pan 1005807183

Word count: 2421

Table of Content

1. Introduction	2
2. Statement of Functionality & Screen Shots from App	2
2.1 Place It	3
2.2 Measure It	4
2.3 Import It	5
2.4 Draw It	6
3. Overall Design	7
3.1 Overall System	7
3.2 Workflow of the App by Four Main Functions	8
3.2.1 Place It	8
3.2.2 Measure It	8
3.2.3 Import It	8
3.2.4 Draw It	9
3.3 ARCore SDK	9
4. Reflective Thinking	10
4.1 Underestimating the Workloads	10
4.2 Difficulty of Developing a Cross-platform App in Unity	10
5. Contributions	12
6. Future Works	13
6.1 UI interface improvement	13
6.2 Technical Improvement to Improve User Experience	13
7. Authorization	14

1. Introduction

The goal of our project is to build a mobile app that allows users to import or construct 3D object models and render these virtual objects in the real world. The user will also be able to measure dimensions of physical objects with the app.

Sometimes, we want to see if an object can fit into the surroundings, like adding a new piece of furniture, and how it looks. However, it is difficult to visualize without the actual object. As Engineers, we can use 3D modeling software like AutoCAD or SolidWorks, but not everyone has access to them. Also, 3D modeling is time-consuming and we may still need to do some physical measurement. We all have smartphones nowadays, so we want to resolve this by developing an app. In addition, there is currently no such kind of application in the market where the user is allowed to import their own 3D model into the AR scene and directly draw 3D models in the application for AR displaying.

2. Statement of Functionality & Screen Shots from App

There are four main functions in our app, namely “Place It”, “Measure it”, “Import it” and “Draw it”. The team has successfully implemented all the functions as proposed. All the functions can be easily accessed from the Home Page.

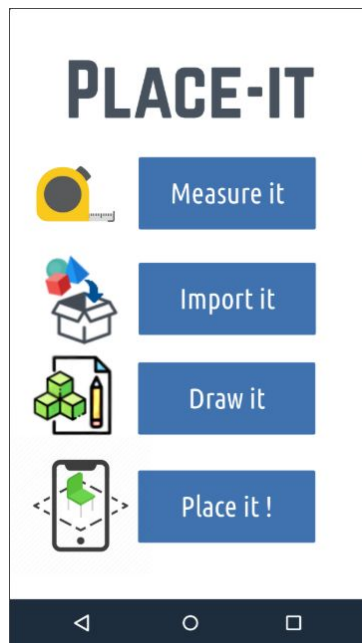


Fig 1. Home Page of “Place-It”

2.1 Place It

This is the core function of the App, which allows the user to place and rearrange virtual objects in the real world through the smartphone's camera. The logic flow for the user is intuitive but this page also provides a few quick instructions. Once the user enters this page, a quick instruction will teach the user to scan the surface. Then the user can select the desired objects from the inventory bar and place them on the scanned surface. Also, the user can rearrange and slightly rescale the placed objects by gestures like dragging, rotating and pinching.

To achieve this function, an algorithm is developed with the help of Google ARCore SDK. Also, the inventory bar is integrated with other functions like “Import It” and “Draw It” to load resources.

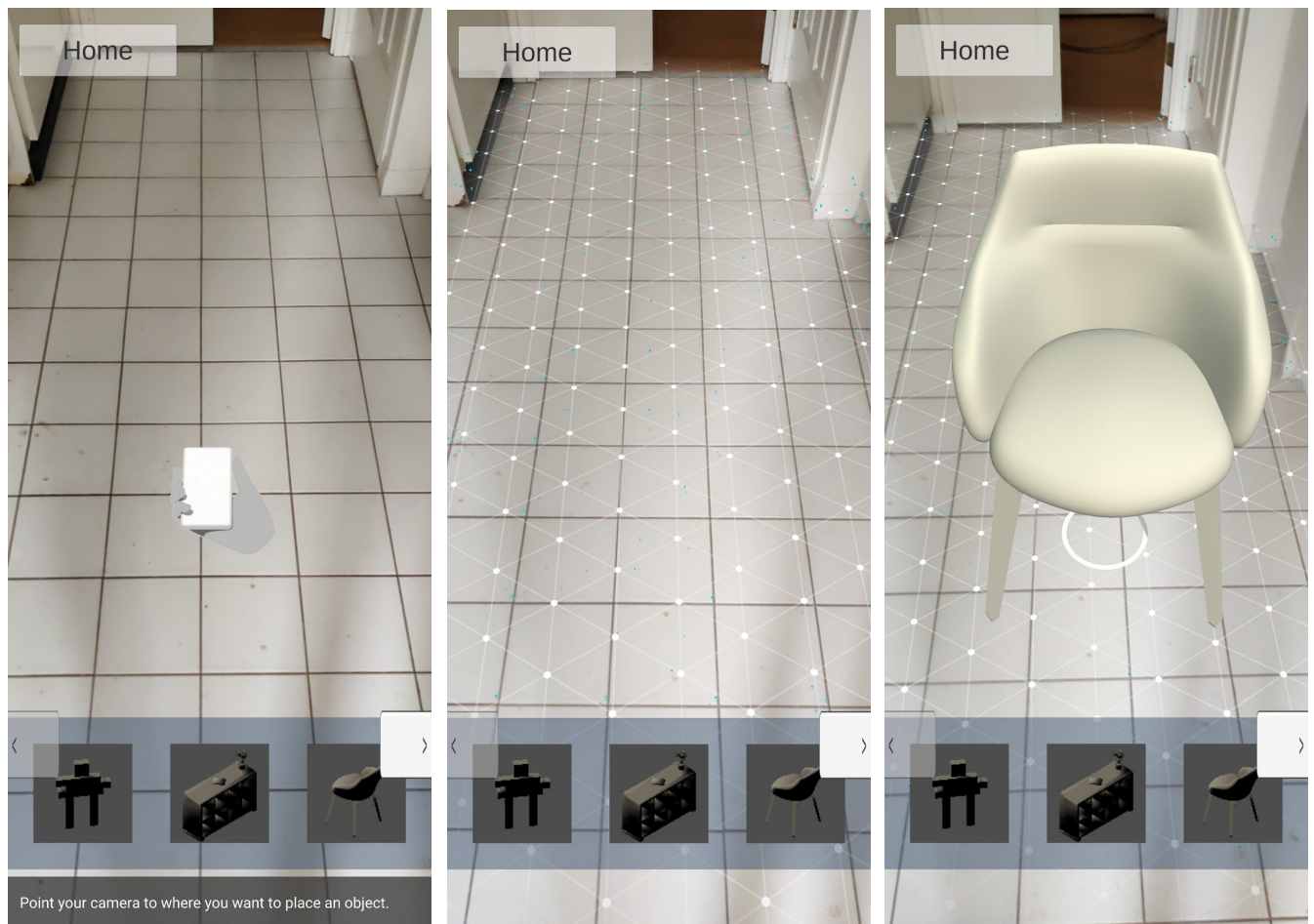


Fig 2. “Place It” Function Page

2.2 Measure It

In this function, the user is able to measure in the real world through the smartphone's camera. Similar to the previous "Place It" function, the user needs to scan the desired surface first. Then the user can place multiple anchor points on the scanned surface. The app will draw a line between any two selected anchor points and return the distance between them.

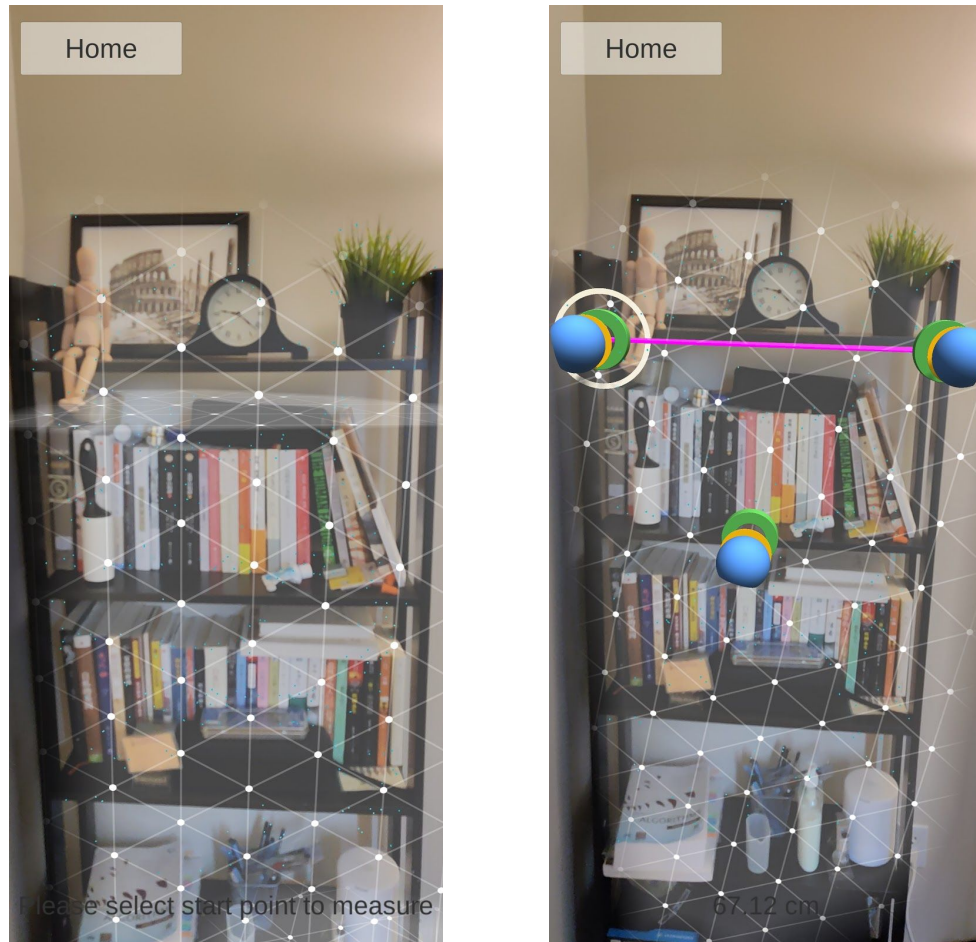


Fig 3. "Measure It" Function Page

2.3 Import It

In this function, the user is able to import any .obj file stored in the external storage into the app. Once the user clicks on the “Import Model” button, the user will be directed to the smartphone’s file browser, then the user can select any .obj file (also highlighted) to import. The imported file is now in the app and can be selected in the “Place It” function.

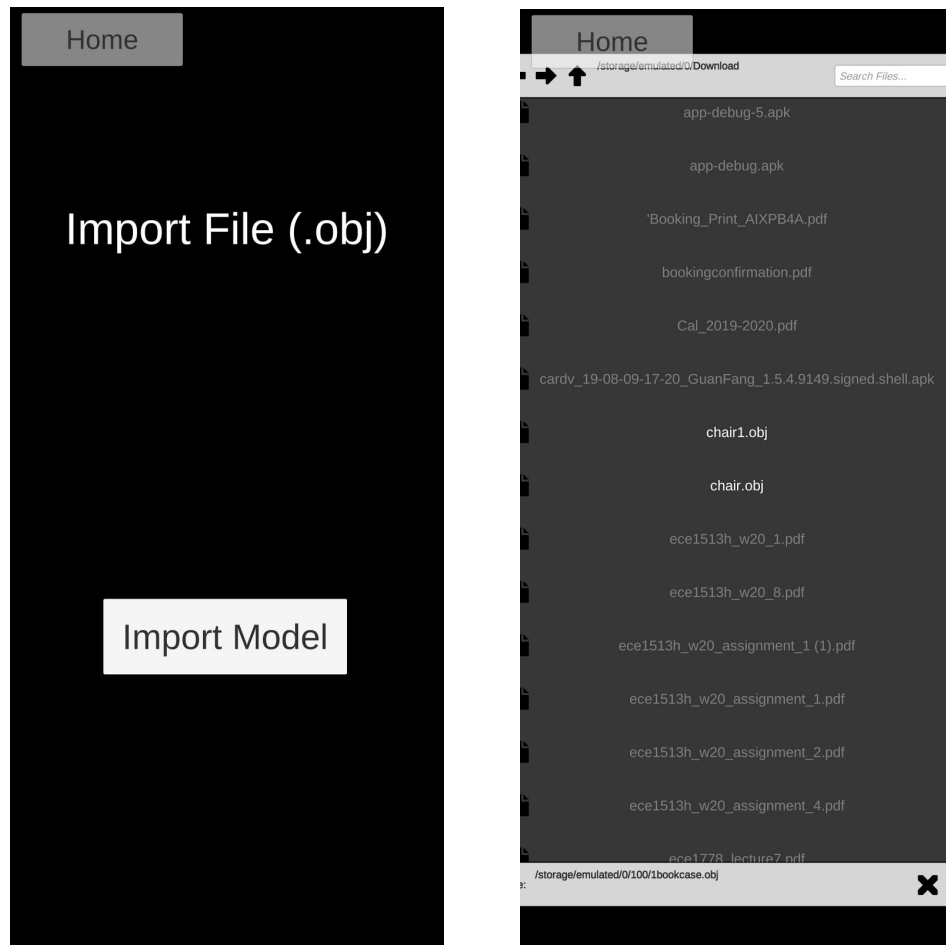


Fig 4. “Import It” Function Page

2.4 Draw It

In this function, the user is able to draw 3D objects by stacking simple geometries. These geometries can be relocated and rescaled by the user. The user can also rotate the camera view and reset it in order to facilitate the drawing process. Once the user is done with the drawing, he/she can click on the “Place It !” button to be redirected to the “Place It” function page.

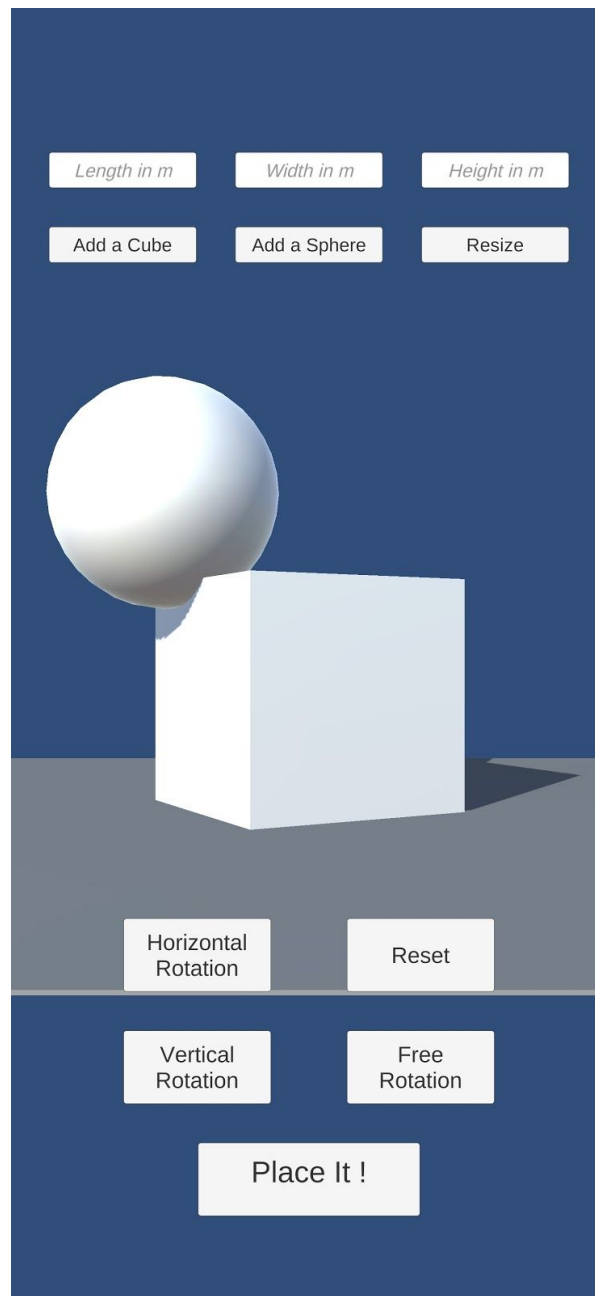


Fig 5. “Draw It” Function Page

3. Overall Design

3.1 Overall System

Augmented Reality (AR) is an interactive technique that allows users to reside virtual objects in a real-world environment. To achieve the basic AR function of the application, the team uses Unity as the 3D modeling engine and ARCore as the augmented reality engine.

To achieve all the functions mentioned in section 2, some other functional components than the ARCore need to be implemented by the team, which are displayed as blocks as shown in the block diagram below.

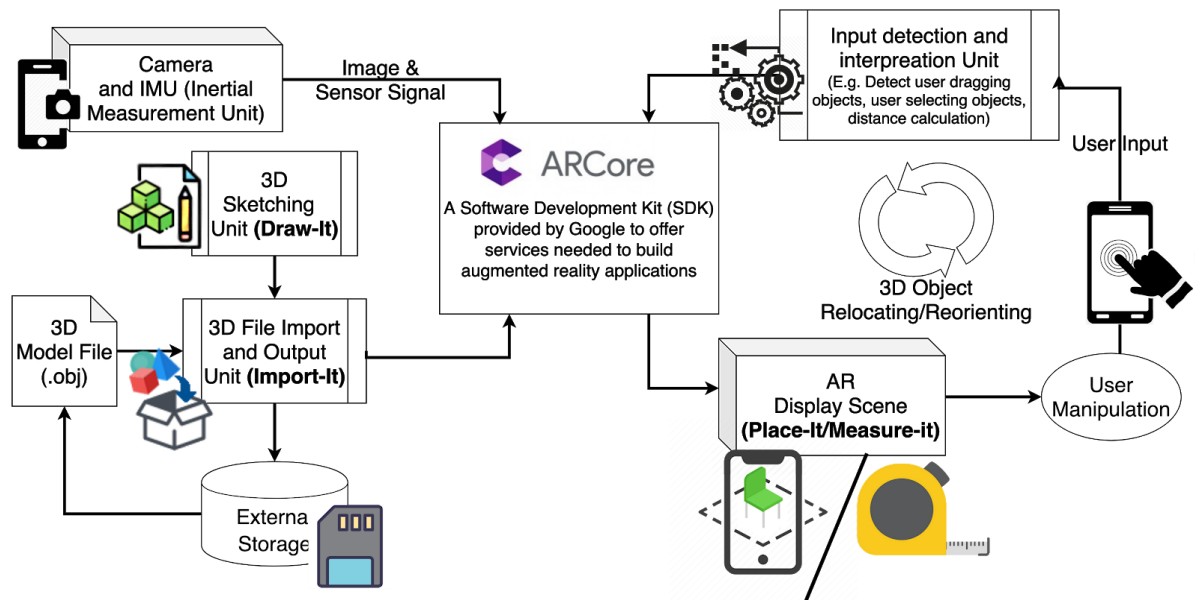


Figure 6. Block Diagram of the System

3.2 Workflow of the App by Four Main Functions

3.2.1 Place It

- 1) ARCore generates the augmented reality virtual environment based on the image and position inputs from the camera and the Inertial Measurement Unit (IMU).
- 2) The user can select objects and place them in augmented reality.
- 3) To recognize the user's selected point, a ray is released at where the user is pointing at, from the screen coordinate and project to the augmented reality's coordinate. This is used to determine if the user has selected the objects and get the object location.
- 4) The "Input detection and interpretation unit" monitors the user input and performs the coordinate transformation of the selected 3D virtual objects in the Unity built 3D coordinates systems.
- 5) The ARCore takes the new coordinates and orientation of the 3D virtual objects and re-render the 3D objects in the AR scene

3.2.2 Measure It

- 1) ARCore generates the augmented reality virtual environment with detected planes based on the image and position inputs from camera and IMU
- 2) The user hit a point on the scanned plane, a 3D marker placed on the hitting point
- 3) The "Input detection and interpretation unit" calculates the distance between the selected two 3D markers according to the relative distance of the two 3D markers in the Unity 3D coordinate system. Since the Unity 3D coordinate system has the same scale as the actual word as generated by the ARCore, the calculated relative distance versus the actual distance is 1:1.
- 4) The corresponding actual distance between the two 3D markers is displayed on the UI.

3.2.3 Import It

- 1) The user enters the "Import It" user interface with a file browser showing all the files in the external storage of the mobile phone
- 2) The user selects the file with acceptable format (.obj file) and the "3D file import and output unit" performs runtime mesh build of the OBJ file and converts it into a Unity 3D object format (GameObject).
- 3) The converted 3D model in the form of GameObject can be processed by ARCore and displayed in the Unity scene.

3.2.4 Draw It

- 1) The user enters the “Draw It” user interface where the system renders a Unity 3D coordinate system
- 2) The user add and adjust the simple 3D objects in the current scene
- 3) When the user chooses to export the drawn 3D object, the current 3D objects in the scene will be exported as a .obj file to the temporary folder which can be imported to “Place-It”.

3.3 ARCore SDK

The main SDK of our App that allows us to build an augmented reality, which contains 4 major parts:

- 1) **Environmental Understanding** is the part that looks for clusters of feature points captured by the camera, then it will recognize edges and planes. This is the very fundamental of augmented reality.
- 2) **Motion Tracking** is the part that ensures a correct perspective as the smartphone moves. It facilitates concurrent odometry and mapping technique and device’s inertial measurement to where the smartphone is located in the real world as well as its position and orientation.
- 3) **Oriented Point** allows us to place a virtual object onto a surface, ARCore will look at the feature points and attempt to place the object with consideration of angles, then it will return the object’s pose.
- 4) **Anchors and Trackables** ensure the accuracy of the placed virtual object’s pose when the surrounding environment updates. It is achieved by tracking the anchor we defined as we place an object.

4. Reflective Thinking

4.1 Underestimating the Workloads

During the project planning stage, the team underestimated the difficulty of implementing importing and rendering external 3D model files features.

Since reading/storing files from/to external storage are very easy in Android, we thought it should be not that difficult in the Unity as Unity project can be finally compiled to run on the Android devices. However, as Unity is a game development framework, and almost no game requires I/O features, it has very little support in Unity to deal with loading/writing files from/to external storage during run time. We even need to implement the UI for the IO operation.

In addition, rendering imported files at runtime is another challenge. Unity has its own compiling process to process assets like OBJ files during the compilation. To transform the runtime imported file to a form that has the same properties as compiled by Unity, we have to simulate Unity's compiling process to make sure the imported obj files can be correctly presented in Unity. For example, the app needs to calculate appropriate collision volume for objects at runtime.

Since the team didn't do enough research on the ARCore's APIs, we found that ARCore does not have APIs for constructing or scanning 3D models. So, we implemented the DrawIt feature from scratch. We used pure Unity to implement the 3D model constructing feature. We manually set the scene, such as the Camera position, and implemented all the interactive logics. The team spent time on developing the DrawIt feature is more than double the estimated.

4.2 Difficulty of Developing a Cross-platform App in Unity

At the very beginning, since some of our group members use Android and some of our group members use IOS, we decided to use Unity to develop the cross-platform app so that all the members can enjoy our app. However, during the implementation, we found that using Unity to develop a Cross-platform app is not as easy as we thought. Certain frameworks, such as Flutter, could be a better choice.

Firstly, Unity is an engine that is optimized for game development. Developing mobile apps in Unity requires extra effort. For instance, Unity requires using Canvas to develop UI. However, adding listeners to Canvas is way more difficult than in Android/Swift.

Secondly, running Unity on IOS requires us to recompile the unity project in XCode. As we used ARCore in our Unity project, we have to convert the ARCore APIs to the corresponding APIs in Apple's ARKit. Also, because of the mismatched version between ARCore and ARKit, we have to downgrade our ARCore to make it work in IOS. It results in tons of work for us to do the conversion.

In contrast, Certain cross-platform frameworks, like Flutter, are easy to learn and are easy to compile the program to run on both, even more, platforms. However, one drawback of those cross-platform frameworks is that these frameworks can only be used to implement simple apps. Applications that use complex libraries, like ARCore or ARKit, cannot be implemented by these frameworks.

In conclusion, Unity is not designed for developing mobile apps. Using native developing tools, like Android Studio or XCode, for developing complex applications is better than Unity, using cross-platform frameworks, like Flutter, to develop simple applications is greater than Unity.

5. Contributions

Feature	Spiral No.	Participants		
		Liren Pan	Hongzheng Xu	Yixiao Hong
Writing Report	Not applicable	√	√	√
UI Design	Spiral 1	√	√	√
PlaceIt	Spiral 2	√	X	√
MeasureIt	Spiral 3	√	√	X
ImportIt	Spiral 4	X	√	√
DrawIt	Spiral 5	√	X	√
Review/Debug	Spiral 1-5	X	√	X

Table 1. Contribution Table

All three of the team members participated in writing all the reports/documents. The team also worked together to finish the UI design in Spiral 1. Since some of the features are harder to develop than the others, we often start the implementation before the assigned Spiral starts. To this end, in Spiral 2 and 3, Yixiao Hong and Liren Pan developed the feature PlaceIt. In addition to that, Liren Pan worked with Hongzheng Xu to develop the feature MeasureIt. In the last two Spirals, Yixiao Hong and Hongzheng Xu are responsible for developing the feature ImportIt. At the same time, Yixiao Hong helped Liren Pan to develop the feature DrawIt. It is worth mentioning that, because of the limited Android device resource and the COVID-19 crisis, the team cannot sit together to develop the app. Hongzheng Xu did the debugging/code reviewing throughout the entire project.

6. Future Works

The team has successfully achieved all the mentioned basic functionalities of the application “Place-It”, while there are still some aspects that can be improved to improve the user experience and optimize the performance of the application:

6.1 UI interface improvement

- 1) Add user manual and tips to provide guidance to the user of how to use the App
- 2) Add buttons in the Place-It to clear all the placed items in the AR Scene
- 3) Add buttons in the Measure-It to clear all the placed measurement markers
- 4) Add an interface in “Import-It” where user can easily manage the imported models
- 5) Add animation and visual effect in the UI

6.2 Technical Improvement to Improve User Experience

- 1) Render the texture onto the imported 3D models to improve the visual effect of the models in the AR scene
- 2) Change the current Synchronous model loading process to Asynchronous models loading process to eliminate the wait time
- 3) Support more types of 3D files which can be read by the application besides .obj file
- 4) Achieve multi-platform of the application such as compile the Unity project to IOS devices.

7. Authorization

State if OK or not to post each of the following on the Course Website:

All of the Team members (Yixiao Hong, Liren Pan, Hongzheng Xu) have agreed on giving the authorization of posting all of the following materials:

- i. Video of final presentation
- ii. Report
- iii. Source code