Final Report

Interactive Interior Design with HoloLens

16/4/2017

University of Hong Kong
Department of Computer Science

Supervisor: Loretta Yi-King Choi
Author: Tam Chi Ho
UID: 2013510585
Acknowledgements

I would like to thank my supervisor Dr. Choi for her guidance and support which played a major role in this project.

Abstract

Mixed Reality is a new technology that allows users to perform physical interactions with the digital world in real time. It combines features of Virtual Reality and Augmented Reality. The project aims to explore and display the potential of this technology by creating a Mixed Reality application. The focus of the application is interior design but there are also interactive features involved. This report presents our technology used, application design, methodology, and future development.
# Table of Contents

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Project Overview</td>
<td>4</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>5</td>
</tr>
<tr>
<td>2 Technology</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Tango tablet</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Microsoft HoloLens</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Ovrvision Pro</td>
<td>7</td>
</tr>
<tr>
<td>2.4 Leap Motion</td>
<td>8</td>
</tr>
<tr>
<td>2.5 Unity3D</td>
<td>8</td>
</tr>
<tr>
<td>3 Application Design</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Main Menu</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Preview Mode</td>
<td>10</td>
</tr>
<tr>
<td>3.3 Drag Out</td>
<td>13</td>
</tr>
<tr>
<td>3.4 Additional features</td>
<td>14</td>
</tr>
<tr>
<td>3.4.1 Visual mesh</td>
<td>14</td>
</tr>
<tr>
<td>3.4.2 Spatial mapping</td>
<td>15</td>
</tr>
<tr>
<td>3.4.3 Generate 3D model</td>
<td>15</td>
</tr>
<tr>
<td>4 Methodology</td>
<td>16</td>
</tr>
<tr>
<td>4.1 Tango</td>
<td>16</td>
</tr>
<tr>
<td>4.2 Ovrvision Pro + Oculus + Leap motion</td>
<td>19</td>
</tr>
<tr>
<td>4.3 HoloLens</td>
<td>20</td>
</tr>
<tr>
<td>4.3.1 Menu</td>
<td>20</td>
</tr>
<tr>
<td>4.3.2 Rotation and Scaling</td>
<td>21</td>
</tr>
<tr>
<td>4.3.3 Dragging</td>
<td>22</td>
</tr>
<tr>
<td>4.3.4 Generate model function</td>
<td>24</td>
</tr>
<tr>
<td>4.3.5 Delete function</td>
<td>25</td>
</tr>
<tr>
<td>5 Problems</td>
<td>26</td>
</tr>
<tr>
<td>6 Further Development</td>
<td>27</td>
</tr>
<tr>
<td>7 Conclusion</td>
<td>28</td>
</tr>
</tbody>
</table>
1. Project Overview

1.1 Background

In Virtual Reality, a head-mounted device covering the user’s sight and hearing is used to transmit 3D computer graphics and audio. Users can therefore immerse themselves in the digital world as if they were in the environment. For example, users can enjoy the view of the Grand Canyon while sitting on sofa.

In Augmented Reality, the digital data is overlaid on the physical world capturing by camera to create the illusion that both aspects are in the same reality. This usually appears on smartphones or cars. For example, map instructions or information can be projected on the dashboard of a vehicle to guide the driver. Popular games such as Ingress or Pokémon GO also uses Augmented Reality.

Mixed Reality is rather like Augmented Reality. Both digital and physical data are mixed together to create an illusion. Unlike AR, digital and physical objects can have physical interactions in real time to even enhance the illusion of having both aspects in a same reality. As an example, if you throw a digital ball to a physical wall, the digital ball should be able collide with the wall and bounce back.

The idea of Mixed Reality has been discussed for years, but was not realized until 2016. Google and Microsoft are the only companies which released their devices for Mixed Reality development in early 2016. This is the reason that we are going to use Microsoft HoloLens and Google Tango Tablet Development Kit in the project. And there are no other MR device products on the market beside the 2 mentioned
1.2 Objective

The main objective of the project is to create an application enabling simple interior design using Mixed Reality while exploring the features and limitations of the new technology. In this project, the team would focus on enhancing the user experience of the application to demonstrate the power of Mixed Reality to the public. During development, I have explored multiple VR/AR/MR devices including Tango tablet, Oculus, Leap Motion and Ovrvision Pro and eventually decided that HoloLens best fits the project.
2. Technology

2.1 Tango tablet
The Tango Tablet Development Kit is a device that uses computer vision to learn about the world around it. It is equipped with multiple sensors that enable application developers to perform motion tracking, area learning and depth sensing.

2.2 Microsoft HoloLens
HoloLens is a head-mounted device equipped with various sensors, cameras and has a processor itself. The visor is tinted and inside the visor is a pair of lenses that project graphics to the user.

HoloLens uses the position and orientation of the user’s head to determine their gaze direction. Therefore, if the user looks around the room without moving his/her head the gaze
pointer would not be moved. It is similar to have a ray casting out from the centre of HoloLens and interactions can be made using this gaze ray.

The main sources of input of HoloLens is an air tab gesture and a bloom gesture. However, the bloom gesture is used as a Start menu gesture leaving only the air tab gesture to be use for development. Other than hand gestures, HoloLens can also receive voice input.

2.3 Ovrvision

Ovrvision Pro is a high performance stereo camera that can be mounted on Oculus Rift/HTC Vive to achieve immersive AR.
2.4 Leap Motion

Leap Motion is a small USB motion sensor designed for physical desktops or virtual reality headsets. It has two monochromatic IR cameras and three infrared LESDs. It can detect users’ hands and transform them into a new input source. Gestures such as pinching and holding fingers can be easily recognized by the device.

2.5 Unity3d

Unity is a free cross-platform 3D development tool developed by Unity Technologies. It is mainly used for creating game applications and supports the integration of two Mixed Reality devices currently on the market, Google Tango tablet and Microsoft HoloLens. Other devices such as Leap Motion, Oculus and Ovrvision are also supported.
3. Application Design

3.1 Menu

When the application is first launched, an open menu button would appear in the top left corner of the user’s view. The button would always follow the user’s rotation so it can always be seen by the user.

Clicking the button would open the main menu for the user to choose the type of furniture they want to have a look at. The six types of furniture are chairs, tables, coffee tables, sofas, shelves, and slide boards.
After choosing the type, a selection of pictures of the type of furniture would be shown.

The user can then click on any of the furniture pictures to view the actual 3D model of the item. This is called the Preview Mode. While in Preview Mode, the item and the menu also follows the user and always faces the user.
3.2 Preview Mode

Once a furniture is chosen, the user would enter the preview mode where he/she can perform scaling or rotation on the object. The user must first select the object by performing an air tab gesture while the cursor is on the object. (All similar actions would be referred as selecting the object in the report) Once tapped, the object would be selected and 12 handlers would appear surrounding the object.

The cubes are for modifying the scale of the object and the spheres are for rotation along the y axis. For enlargement, the user can click on any one of the cubes and drag their hand away from the object just like performing an enlargement in normal photo editing software like photoshop or paint, vice versa for contraction.
For rotation, the user can click on any one of the spheres and drag their hand to the direction they want the object to rotate.

The user can also continue traversing the list of furniture by clicking the left right buttons or using the voice commands “next item” and “previous item”.

3.3 Drag & drop

If the user would like to place the object into the environment, he/she can tap on the object and hold the tap to drag the object out of preview mode. Once the object is dragged out, it would be enlarged three times of the size in the preview mode and can be navigated by the user’s hand. If the user wants to drop the object, he/she can release the fingers and the object would no longer follow the user’s hand.

Once the object is settled, features in the preview mode are still enabled. The user can select the object to perform enlargement and rotation. The object can also be drag & drop again. However, there would be an additional red cube generated on top of the object when selected. Clicking the cube would delete the object.
3.4 Additional features

3.4.1 Visual Mesh

When an object is first dragged from the preview mode, visual meshes of the spatial mapping will appear so the user can have a clearer idea of the colliders.

The visual mesh option can also be turned on/off in the main menu or by the voice commands “visual mesh on” and “visual mesh off”.

[Image: Visual Mesh with options on and off]
3.4.2 Spatial Mapping

By default, spatial mapping is on so objects can’t be dragged through real world objects. However, this might cause problems if the user is in a small crowded place so spatial mapping can be disabled by the voice command “spatial mapping off” and enabled by “spatial mapping on”.

3.4.3 Generate 3D model

For users to have an overview of their created environment, a 3D model of the room including the furniture can be generated by the Generate Button in the main menu.
4. Methodology

4.1 Google Tango

Originally, we planned to work on both Tango device and HoloLens and develop a creative block placing application like Minecraft. A simple demo application with fundamental features was created.

Users can select virtual objects and add them into mixed reality by tapping on the screen. The objects will only be placed if the destination of the tap is a horizontal plane with an incline or decline of no more than 30 degrees. If the destination does not fulfil the requirements, the application is smart enough to detect the objects cannot be placed on the targeted area. Other than that, objects can also be stacked on top of each other by tapping directly on any object.
After successfully instantiating objects on physical surfaces, I worked on modifying light effects on objects according to the surrounding environment so virtual objects would look more realistic.

As you can see above, the cars on the left have a lower brightness and a more realistic texture. The cars on the right stand out immediately from the physical environment. The is the result of modifications in the texture, brightness and shader of the object.
Other than placing objects, there are also special features such as driving a virtual remote car on top of the real terrain of the environment. A top-down view of the scanned room is also rendered at the bottom right corner.

However, after deciding to change our application to a more realistic furniture design application. The Tango tablet was given up since the only input source available is from the touch screen. Therefore, users can only control objects via buttons on the screen and this doesn’t give a very good user experience. We aim to let users navigate through our application with very simple and natural hand gestures.

Originally, I tried to integrate the motion sensor Leap Motion with the Tango tablet to capture hand movements. However, Leap motion requires to be connect to a computer via a cable and mounting it on the Tango tablet would defeat the purpose of wireless.
4.2 Ovrvision Pro + Oculus DK2 + Leap Motion

After giving up on the Tango tablet, I tried integrating a Ovrvision Pro, an Oculus, a Leap Motion device and a structural sensor to achieve mixed reality similar to having a HoloLens. However, during the process of integrating the Ovrvision Pro, Oculus and Leap Motion, I discovered that Ovrvision Pro itself is still in a very new development stage. The documentation for the SDK is very limited and there are numerous bugs once integrated with the other two devices. For example, one of the technical issues is that the graphics rendered to the Oculus through Ovrvision Pro would sometimes be distorted and only half of the screen in Oculus would be successfully rendered.

Other than some technical issues, there is also a big difficulty integrating leap motion and Ovrvision Pro. Ovrvision Pro is a stereo camera so the graphics captured from the left and right camera is transferred to the respected lens on Oculus. Therefore, it means that the image rendered to the Oculus lens would be slightly different just like when you use different eyes to look at the same object, the position of the object would look different to you with one eye open. This phenomenon is called parallax.

However, the detected hands from Leap Motion that are rendered into Oculus also come from a stereo camera in the Leap Motion. Therefore, we have two sets of stereo cameras rendering graphics into the Oculus. The problem is that the two sets of cameras are mounted in different position on the Oculus and therefore captures a different view of your hands just like you’re your left and right eyes. As a result, the detected hands from Leap Motion would not match the position of the actual hands rendered by Ovrvision Pro. There would also be blind spots where only one of the cameras will detect your hands and the other does not.

At last, due to time constraints, I have decided to abandon this plan and work on HoloLens.
4.3 Microsoft HoloLens

4.3.1 Menu

All the objects are categorized as an Item Object and each Item has an id, name, image and a 3D model. The objects are stored in an array so they can be accessed easily. The left and right button in the preview mode is used to traverse through the list of the type of item. If it reaches the end of the list it would continue at the start of the list.

Whenever an item is being previewed its 3D model would be instantiated at the position of the previous item’s 3D model. Once a model is dragged out, an empty object would take the place of the model so it can be replaced next time user enters preview mode. Other than that, two scripts would be added to the model of the item. A RotNScale script for the rotation and scaling controls and a DragtoMove script so the object can be dragged out of preview mode by user. A mesh collider is also added so the objects can be detected by HoloLens cursor.
4.3.2 Rotation and Scaling

Once an object is selected in the preview mode, 8 cubes and 4 spheres would be created for rotation and scaling. The positions of the handlers are calculated according to the MeshRenderer of the object.

Once a sphere is pressed, the hand position of the user would be recorded and the object would then be rotated according to difference of position in the horizontal axis.

Similarly, if a cube is pressed, the hand position would be recorded and the distance changed from the original press point would be used to scale the object.
4.3.3 Dragging

The dragging function is implemented by detection of a press that holds longer than 0.3s on the object. The mesh collider on the object allows it to detect the press.

Once an object is dragged, it would exit the preview mode and the scale would be increased three times. A Rigidbody and a script DetectCollision is added to the object for detecting object collision. The object would also become a child of a parent object called “Objects” so all furniture objects are grouped together for later use.

When dragged, the position of the user’s hand is recorded and the movement of the hand is projected to the object. Originally, we simply modified the position of the object by adding the vector of the current hand position minus the original pressed hand position. However, this causes problems in terms of colliders since directly interact with the position coordinated of the object would ignore the interactions between colliders. Therefore, objects would go through physical surfaces like walls or floor and this is rather unrealistic.

After that, we modified the function using the MoveTowards function which is designed for rigidbody movement from unity to navigate the movement of the object. The function requires a starting point and end point. The starting point is the current position of the object and the end point is the calculated destination based on the movement of the hand from user. At this point, if the starting point and end point at the opposite side of a mesh collider, the movement of the object would obey the physics and would not pass through the mesh collider like before.

![Diagram of MoveTowards function](image)

Originally, we tested this in the unity editor and it works perfectly. However, when the application is loaded into HoloLens, we discovered that for unknown reasons the collider detection is a bit slow and the object dragged towards a wall would still be moved 1 frame even though it is already in collision with a wall. Therefore, once the starting point is has passed the collider and is at the same side as the end point, it would go through the collider. We tried fixing the problem with multiple methods including using the FixedUpdate() function which is designed for Rigidbody movements but it still doesn’t work on the HoloLens.
To tackle this problem, we added a script for detecting collision as a trigger on the object. In every frame, the position of the object is stored in the variable called lastcheckpoint. Once a collision is detected, it means that the current position of the object is invalid so the variable lastcheckpoint would be used as the starting point of the MoveTowards function.

If lastcheckpoint is initialized as the position of the user so if the object is already in collision once dragged out of preview mode that position of the user would act as the starting point of the moveTowards function.
4.3.4 Generate model function

To create the generate function, a lot of studying of code was done to figure out where the data of visual meshes are stored. After some research and trial and error, I have discovered that the visual mesh objects are instantiated under the parent object “spatial mapping”. Therefore, all child objects of “spatial mapping” are cloned along with the furniture in “objects” and scaled in a factor of 0.2. All the objects are grouped together and projected as a 3D model in front of the user.
4.3.5 Delete function

When an object is no longer in preview mode, selecting it would generate the 12 handlers and an extra red cube on top of the object. The red cube is for deleting the object. If the user wants to delete all objects in the application, there is also a voice command “delete all” for the purpose. As mentioned before, all the objects are grouped as child objects under a parent object “Objects” so the command basically deletes all child objects of that object.
5. Problems

5.1 HoloLens

At the beginning of the project, unity does not support the simulation of HoloLens so every time we want to run the application, we must build the application and run it on an emulator or the HoloLens itself which takes a lot of time.

There was also a huge change in the SDK during our project and everything needs to be redo due to this. All the official tutorials from Microsoft are also no longer suitable and we had to analyse and learn the code from the new SDK again.

Other than that, there was also some technical issues encountered. As mentioned before, there are times when a certain piece of code would work in Unity and is logically correct but it simply doesn’t work on HoloLens. The reasons for this is yet to be known but we suspect it is due to bugs in the HoloLens SDK since it is still rather new. For example, a basic cooldown function to prevent a button being click twice is implemented as below.

```csharp
float t;
void Update () {
    t += Time.deltaTime;
}

void clickButton()
{
    if (t > 500f)
    {
        //do stuff
        t = 0;
    }
}
```

The code works perfectly in Unity and is logically correct but it doesn’t work when the application is loaded onto HoloLens. These unknown bugs have created some huge setbacks in our project since we couldn’t identify the problem.

Moreover, Mixed Reality and HoloLens is still a rather new technology so there isn’t must documentation or examples that we can look at. Deciding the scope of the project is also difficult since we do not have enough knowledge of the technologies. As mentioned earlier, I encountered many problems while using other devices to achieve mixed reality and eventually had to give up.
6. Future Developments

If the application wants to be used as an actual commercial application for selling furniture, a lot of additional features must be added. Price of the item and information about it should appear in a panel when it is selected.

Other than adding extra text data, there can also be a multiplayer mode where people can design together in the same environment. Moreover, users should be able to connect in different physical locations and share the same spatial data. For example, designers can be connected to clients and design in their environment without needing to travel to the actual apartment. Imagine a designer sitting in his own office helping design your apartment while you are watching beside and giving opinions.
7. Conclusion

After multiple tests done on the Tango tablet and other devices, we decided that HoloLens is the best device for building an interior design application. All the basic functionalities like moving, scaling, or rotating an object can be done in simple hand gestures to create a great user experience. The movement of the objects are also restrained by the physical terrain to generate a more realistic Mixed Reality environment.

Although the main functionalities of the application are completed, there are still many extra features that can be added to it for improvement and actual commercial usage.