Memo

To: Professor Pisano, Professor Osama AlShaykh

From: Allison Durkan, EJ Fitzpatrick, Johan Ospina Buitrago, Santiago Beltran, Tess Gauthier

Team: 13

Date: 2/16/17

Subject: Ventana Second Deliverable Test Report

1.0 Initiate HoloHub with Windows 10 IoT Core

1.1 Description & Goal

The HoloHub server is the way the internet of things, IoT, devices communicate with the HoloLens running Ventana. Our customer gave us a new constraint which required us to change the operating system on the Raspberry Pi, from Raspian, a Linux-based OS, to Windows 10 IoT Core. Due to this new constraint, we migrated from Flask to a Node.Js server running on top of the Windows OS. Therefore, we needed to test that the raspberry pi started up properly and reached stable network connectivity, which is critical for server requests.

1.2 Procedure

The Raspberry Pi with the Windows 10 IoT Core SD card was plugged into a 2 Amp USB 5V power source. Failure to power the Raspberry Pi with the correct power supply can lead to corruption of the file system. For verification purposes, the HDMI cable was plugged into the Raspberry Pi and which displayed Windows logo on the monitor as the Pi was booting up.

1.3 Results

Once powered on and connected to the network of the router, the ethernet jack status light on the Raspberry Pi was lit. We are also able to make a remote connection into the pi from the Windows IoT Remote Client application on a computer and see the network connection to “tenda_48e238” from the Win 10 IoT Core settings menu. On the display source of the Raspberry Pi, the IP address is visible on the desktop.

1.4 Conclusion

The Raspberry Pi successfully runs Win 10 IoT Core and allows Ventana to take advantage of the wide variety of Universal Windows Platform Applications available today.

2.0 Ensure servers operational on HoloHub Node.js Server

2.1 Description & Goal

With the new HoloHub, it was necessary to find a new solution for controlling Sonos speakers. This test was used to confirm that all of the same functionality exists through http requests with this solution, before the Ventana application was used to communicate with the Sonos server. The HoloHub now runs a Node.JS server. This test was used to verify if the Node.JS servers can run on the HoloHub.

2.2 Procedure

Utilizing Windows IoT Core Dashboard, powershell connection to the HoloHub was opened. The commands 'schtasks' was run and the results of this output were recorded. Next was to verify that the Node executable and server code was stored on the device. The follower commands were entered on the HoloHub:

```
1) cd c:\Node\
   ls
2) cd C:\Projects\HoloHub-js
   ls
3) cd C:\Projects\sonos-http-server
   ls
```

2.3 Result

The initial task, run `schtasks` displayed a list of all the running on the Raspberry Pi. The two important processes are the tasks named 'HoloHub' and 'SonosHTTP' with the status of 'Running.' Command 1 will return display a Node.exe executable and a corresponding a DLL. Command 2 and 3 displayed a server.js file.

2.4 Conclusion

Using the HoloHub server and modules in their own Node.JS server files improves the modularity of the platform, by having standalone components to run the Sonos, the HoloHub, and eventually the Wink. On this architecture, the HoloHub maintains more reliability between modules and servers. In the rare event that a module crashes, the HoloHub server and the other standalone modules will be unaffected, and continue to run. Currently, the requests are hard-coded per module-matching function, but in the following sprints, a more systematic and class-based method of linking HoloLens 3D model functions to the module functions will be generated.

3.0 Communicate with HoloHub and Sonos via Node.js Server

3.1 Description & Goal

This test was used to verify the connection to the servers (network flow into the HoloHub) and from the servers to the dependencies (network flow out of the HoloHub). On the Raspberry Pi, we performed a test to see that the HoloHub and Sonos server modules are operational. Next, we performed a test to demonstrate the ability to communicate with these server modules.

3.2 Procedure

To test the connection to the Sonos HTTP server, a request was sent to the specific Sonos port on the raspberry pi via "192.168.0.105:5005/living room/volume/10". This command, which was sent from PostMan, called the volume function in the Sonos library. To test the connection to the HoloHub Server, a request was sent to the specific HoloHub Server port on the raspberry pi via "192.168.0.105:8081/status". This command, also sent from PostMan, requested the status of the HoloHub server from that Node.js module. These requests should be GET requests to the server modules.

3.3 Result

On success, the Sonos HTTP server will return a JSON success message.


```
{
    "status": "success"
 }
```


On success, the HoloHub server will return a JSON status message. (this content will vary based on the current status of the HoloHub)


```
{
 "album": "The Temptations Sing Smokey",
 "artist": "The Temptations",
 "title": "My Girl",
 "current_transport_state": "PAUSED_PLAYBACK",
 "uri":
"x-sonos-spotify:spotify%3atrack%3a6jWkZvd1URGktyTTwcpPpB?sid=12&flags=8224&
sn=1",
 "playlist_position": 1,
 "duration": 165,
 "position": "00:02:35",
 "metadata": 155,
 "album_art":
"http://192.168.0.101:1400/getaa?s=1&u=x-sonos-spotify%3aspotify%253atrack%253a
6jWkZvd1URGktyTTwcpPpB%3fsid%3d12%26flags%3d8224%26sn%3d1"
}
```

3.4 Conclusion

The HoloHub successfully responded to HTTP requests that the user sent to it. Furthermore, it demonstrated the ability to call an internal module within the HoloHub based on that request. This will allow the HoloHub to modularly interact with other IoT devices that the team can define later.

4.0 Recognize Multiple VuMarks

4.1 Description & Goal

First deliverable testing examined the capabilities of Vuforia, and the maximum allowable distance between the HoloLens' user and Internet of Things device with an image target. Since then, Vuforia's image targets have been switched with VuMarks. VuMarks represent customized images that can have up to 9 bytes of data encoded into them. They can be designed for enhanced recognition capability. In addition, Vuforia is implemented at runtime to associate a VuMark with a specific device's controller. This allows for scalability of Ventana as a platform, and customization for each user.

4.2 Procedure

With the HoloLens on and the Ventana application running, the user looked at VuMark 0. After the hologram appeared, the user looked at VuMark 1, and then looked at VuMark 2.

4.3 Result

Once the Ventana application was started, we placed the VuMarks in sight of the HoloLens' camera and the correct models showed up for each VuMark. VuMark 0 corresponds to the new music controller, Vumark 1 corresponds to the original music controller, and VuMark 2 corresponds to the light controller. The HoloLens could see the VuMark from a few feet away.

4.4 Conclusion

Confirming that we can tell the difference between multiple VuMarks means that we now have the ability to differentiate between multiple devices by placing unique VuMarks on each device.

5.0 Examine Positional Tracking using World Coordinates

5.1 Description & Goal

Microsoft provides an open-source framework, HoloToolkit, that utilizes built-in features of the HoloLens, such as spatial mapping. Spatial mapping allows the HoloLens to analyze its surroundings and continuously update its environmental understanding, as its sensors gather more data. The HoloLens maintains a coordinate system and spacial mesh for each room, which allows a user to place a hologram and then have it persist across

sessions of opening and closing an application, using world coordinates. This provided an opportunity to display the holographic controls to the user during the initial deploy of Ventana using Vuforia, and then the user could move copies of holographic controls to spaces that make sense in the physical world.

5.2 Procedure

Deployed the version of Ventana that supports world coordinates, located on the feature-WorldAnchorAndVuMarkMerge branch on the VentanaIoT Ventana Github repository. After the user gazed at VuMark 0, and saw the holographic controls, the user double tapped anywhere on the hologram to duplicate the holographic control that then appeared in the user's world. The user moved the new hologram by single tapping on a non-interactable area of its model and then pinned it to a new location. Once the hologram reached its desired location, the user tapped again to anchor it. The user repeated the previous steps with VuMarks 1 and 2, in order to place the other holographic control panels in the world.

5.3 Result

The user held each of the three VuMarks up, and then, once the controller attached to the VuMark appeared, he double tapped to create a spawned holographic control. Then, the user put the VuMark down, and tapped to move the spawned holographic control to a new position in the Senior Design lab. The figures below illustrate the successful completion of this test with the light controller. The figures have been taken from a video recorded during testing from the HoloLen's camera.

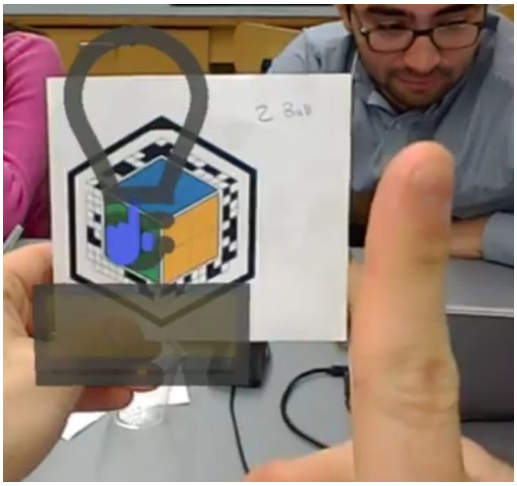

Figure 1: Spawning light controller

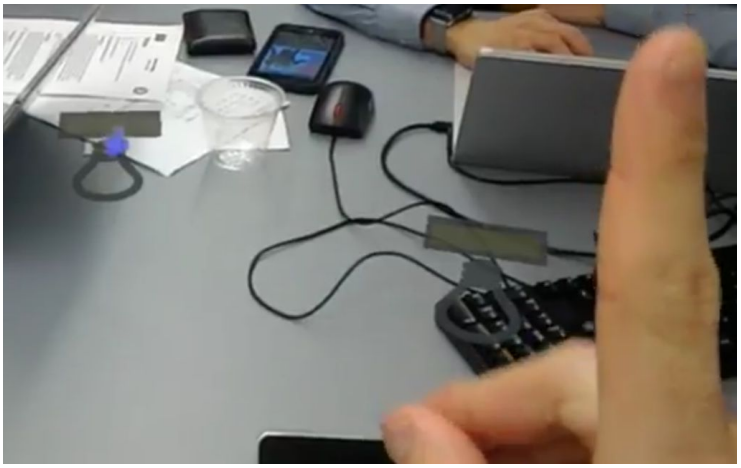

Figure 2: Placing a spawned light controller

The test verified the ability to transition from Vuforia to Spatial Mapping, to fulfill Ventana's positional tracking requirements. Specifically, testing the tap to place and world coordinates functionality on the HoloLens and in an actual room.

Conclusion

The test allows the team to move forward with implementing world anchors, in addition to Vuforia. The test highlights some areas for improvement, to make the experience as intuitive as possible for the user. The tap to place functionality, especially on a controller with other buttons, should be explicitly clear to the user. In addition, a user could accidentally double tap to spawn another controller, so rethinking this gesture and using something like click and hold will be considered. Moving forward, both of these represent two areas to improve and expand upon the testing of the Vuforia to spatial mapping transition.

6.0 Control Internet of Things Device Using Holographic Display

6.1 Description & Goal

We aimed to test the integration of all of the separate components of Ventana, including the VuMarks, spatial coordinates, and the new Windows IoT based HoloHub. Ultimately, we hoped to see that the user experience with the holograms has been expanded upon since the first deliverable testing. Features such as the highlighting of the buttons when gazed upon, additional volume controls, and audio confirmation after tapping a button were examined. Seeing how this improved the user's overall interaction with the holograms and how it can be further improved represented that main goals of this test.

6.2 Procedure

The user gazed upon the music controller and airtapped each of the respective controls.

6.3 Result

Using the hologram we can see it has the same attributes as the model generated from the VuMark. All of the separate controls worked including play, pause, next, and previous, as well as the dynamically updating album artwork.

6.4 Conclusion

The user can interact with the holograms in the same way he interacted with the VuMark Models because the models are exactly the same, with the same attributes, including any request scripts that are associated with them. This allows the user to "copy and paste" models anywhere the user would like in the room, and they will be there persistently.

7.0 Final thoughts

Through testing, the team discerned that a Win 10 IoT Core Raspberry Pi represented a viable replacement for the Raspbian OS that came before it. This allows the server application to utilize more Universal Windows Platform components moving forward. With the advances made to the HoloLens application, Ventana also has the ability to generate and place controls at runtime using the Vuforia VuMark that the team designed. Moving forward, the team aims to focus their efforts on user experience and scaling the application, so that many IoT devices, of varying types, can be connected and controlled through Ventana.