

Memo

To: Professor Pisano
From: Santiago Beltran, Allison Durkan, EJ Fitzpatrick, Tess Gauthier, Johan Opsina
Team: Ventana, Team 13
Date: 4/30/17
Subject: Customer Installation Report

1.0 Details of Customer Installation

The Customer Installation Meeting took place on Thursday, April 13th at 3:30pm in PHO 113. All team members were present, as well as George Matthews, the customer. George tried Ventana on the HoloLens for the first time; he had never used any previous version of the application. He was pleased by the application, and also provided some feedback. He noted that the cursor lagged, so gaze distance and number of samples were reduced in Unity. He also noted anti-aliasing on the text on the holograms. The next steps he suggested were to update the HoloLens to the latest build (from RS1 to RS2), in order to confirm that Ventana will run on the new build. It is being rolled out in phases, however, and is not accessible on the HoloLens we have, yet. The team discussed spectator view, which is how many of the demos for the HoloLens are done by the HoloLens team at Microsoft. Since we do not have access to another HoloLens, and building the parts was a time-consuming task, it was decided the team would forego spectator view, and continue to rely on the live preview and record features. George was curious about running Ventana in another Windows environment, such as the Oculus, because it is a Universal Windows Platform application. A user should be able to upload Ventana to an Oculus and have the same user experience, although some of the features of the application, such as turning off a light, might not be as applicable in virtual reality. The meeting closed with George vowing to go home and fork Ventana's Github repository. A maker being able to fork Ventana and easily start using the application is important, so the team made it a goal to go through the setup process as a new user would, with the user manual as a guide.

2.0 List of All Requirements (and Degree of Competition)

a) Create a platform that allows the HoloLens to interface with internet connected devices in the home: Completed

With the HoloHub running on top of Windows 10 IoT core on a Raspberry Pi, we completed the platform that allows the HoloLens to interact with internet connected devices in the home. The HoloHub acts as the middleman between the VuMark, which represents a specific controller of the users choice, and the HoloLens, which renders the holographic controllers on top of the VuMark. The HoloHub uses a database that stores each device currently paired with Ventana and associates each one with a unique VuMark. The HoloHub also has a universal web app that allows users to spawn new controllers in their world by

presenting the VuMark for a specified device. This allows the user to look down upon it to make a new controller spawn. In addition, pairing and unpairing devices from Ventana is also handled through this web app.

Beyond simply being an application, the HoloHub has been built with modularity in mind to allow developers to add in new devices for Ventana to pair with, thus acting as a platform for developers to build upon. For example, if a developer decides they want to add a Nest thermostat to Ventana, they can add their own Node Js module to expand Ventana's functionality.

b) A Universal Windows Platform application to run on the Microsoft HoloLens: Completed

The Ventana application for the HoloLens acts as the viewer. It is built to be a stripped down application that does two primary things. First, it handles all of the rendering and accounting for the holographic controllers. This includes rendering the controllers when looking at a VuMark and keeping track of permanently placed controllers through world coordinates. Second, the HoloLens application communicates with the HoloHub to both receive pertinent information (like album artwork and track information) and make web requests to the HoloHub (like pressing pause, or changing the volume).

c) A component that will allow us to track internet connected devices in the home: Completed

The solution for tracking objects is two-fold. First, we use a custom designed VuMark and VuForia to spawn new controllers through our HoloHub hosted universal web app. We call this the "controller wallet." The controller wallet is where one can go in the web app to find all the devices currently paired with Ventana, and open them up to present a unique VuMark associated with that device, which the HoloLens can detect and spawn a new controller in your hand. From here we utilize our second tracking technology, world coordinates. World coordinates are provided by Microsoft on the HoloLens allows us to define specific coordinates in the room mesh. When the user is looking at the controller spawned through the VuMark, a "clone" button appears below it. Tapping this button spawns a clone of the controller in the background, which the user can then drag and place into a permanent place in their home.

d) Function Means Chart Accounting for Completion

Characteristics	O	C	F	M	Completed
Create a useable software platform for Microsoft HoloLens	X				✓
Improve user's interactions with in-home IoT devices	X				✓
Accessible and easy to use, even for user's with limited technical experience	X				✓

Recognize, and distinguish between, IoT devices in a room			X		
Establish connections with the IoT devices			X		
Communicate with IoT devices			X		
Recognize the spatial location of an IoT device			X		
Display information through AR about the IoT device			X		
Synchronize AR display with IoT device position			X		
Facilitate interaction with IoT device through AR display			X		
Constellation ID Tracker, or similar hardware component				X	
API-based controller				X	
Network protocols to send to, and receive data from, a specified device				X	
Use Unity and Universal Windows Platform for Software Development		X			
Portability of hardware component		X			
Must consider limitations of HoloLens spatial mapping		X			
Must consider ability to control existing IoT devices		X			

3.0 Overall Assessment of Project Installation Based on Requirement

a) Create a platform that allows the Hololens to interface with internet connected devices in the home.

The HoloHub, is a Universal Windows Platform application developed to function independently on a Raspberry Pi 3 running Windows 10 IoT Core. This application

allowed the functionality of engaging the communication between the HoloLens and the 3rd party vendor APIs to have a modular, scalable, and seamless experience for both developers and the users of our platform. Since the objective of Ventana is to have a platform open to developers to add future functionality, the server (The HoloHub) needed to be built on a development platform that has extensibility and is easy to use across various platforms. This need was met by having the server built on Node.JS and the Express API framework. For the developers to add new modules, the server was setup in such a way, that allows a new 'route' or Vendor API connector module to be added and use all the existing endpoints that interface with the HoloLens. For this objective, the server has support for Music, Lights, and Power Outlet Internet-of-Things devices. As the HoloLens and the Ventana application running on it need to connect to the HoloLens, the server provides two methods of network communication: HTTP requests and a web socket communication channel for push notifications. This allows the HoloLens application to have a consistent and standardized format for requesting actions from the holographic interfaces to actions on the physical devices.

b) A Universal Windows Platform application to run on the Microsoft HoloLens

The Universal Windows Platform application that was made to tap into the spatial mapping functionality of the HoloLens. This functionality allows users to map the environment around them, and most importantly -remember. The team found ways to import 3D assets to assemble a sophisticated, well structured, application. By using a mixture of HTTP requests and Websockets, the team created a non-resource intensive application for a HoloLens to deploy. Websockets allowed Ventana to respond to server events without having to constantly poll the server for a change. This freed up a lot of computational power that went towards spatial mapping and feature creation. The HTTP API structure also took advantage of the HoloLens' on board hardware to bypass any requirement to be connected to a physical device. The viewer side of Ventana was successfully finished, and its structure will allow anyone coming into the project to add new controllers by simply placing them in a directory after designing and programming them, which was a critical part of the success of its design.

c) A component that will allow us to track internet connected devices in the home

The second component critical to the success of the HoloLens application was accomplishing positional tracking of controls that mean something to a user. The team had to make a compromise to not track moving objects, as the inside out tracking functionality of the HoloLens did not support this type of behavior. What was discovered, is that there still is an abundant use of pseudo-stationary devices that the Ventana Viewer is perfect for. These devices include powerstrips, lightbulbs, and music speakers. To add to this, people are also able to have a use for these appliances immediately after placement. Utilizing the unique features of the

Microsoft HoloLens, specific rooms may be mapped and loaded at distinct application launches.

4.0 Follow-Up Plans

Due to the nature of the project, the follow-up plans with the client are less traditional than other projects. Ventana, in its entirety, is an open-source, public repository on GitHub. George Mathews, who works for the Microsoft Garage, is going to help the team push this project into the spaces where developers find projects to tinker with and build upon. Through the duration of the year, the team has composed an extensive list of directions and information about Ventana. This information contains set-up instructions for new users, detailed guidelines about how to structure new devices and vendors, and tutorial-like description of using Blender and Unity to construct controllers. Because this is a project aimed to get developers working with IoT devices and HoloLens development, the team structured every piece of the Ventana application and the HoloHub server to be easily built upon and added to. This design choice, along with the extensive documentation the team has provided, will allow the people excited about technology to fork the Ventana organization's repositories, and continue to add to its already incredibly successful progress.

Note: George Matthews has approved the installation, but due to travel plans, will be unable to provide email confirmation till ECE Day where he will be present during the presentations.