Dear Future Team,

Welcome to senior design! Let’s talk about Ventana. As a student proposed project, we came in with an idea for a HoloLens application. We met with George Matthews, our customer from the Microsoft Garage, to better define our problem statement. After that meeting, as well as some surveying of potential customers, we decided that technology in the home is hard to use. Now, Ventana is an open-source Universal Windows Platform application for controlling internet connected devices in the home, specifically designed for makers to expand upon and use to their heart’s content.

The devices Ventana currently supports are: Sonos speakers, GE Light Bulbs and Pivot Power Genius Power strip. The light bulbs and power strip are both supported through a Wink Hub. Therefore, adding support for more wink devices could be done by simply building upon those modules. A main goal of Ventana, especially as an open-source project, was to be as modular and easy to build upon, as possible.

To begin working on Ventana, you will need to clone both the Ventana and HoloHub repositories. The programs needed are: Blender, Unity, and Visual Studio. The Vuforia SDK is also needed, as well as a license key. Once the server is running, the IP address of the server needs to be set in the Ventana project in Unity. All the steps to set-up the project have been outlined extensively in the user manual. Some of the key tips and tricks that come to mind are:

1. Make sure to install a supported version of Unity. This project was built in 5.5.0f3.
2. When installing the project in Unity, check that the references to the scripts are not null. If it is null, find the appropriate script, and simply drag it to the reference in the inspector tab.
3. When the Raspberry Pi is running Win 10 IoT Core, it cannot be powered off without properly shutting it down. If it does happen and the disk gets corrupted, reflash it.
4. UWP is not the same as the Mono Runtime that Unity uses, at some cases you might (or must) use compiler directives to develop code that works in the Unity Editor and the HoloLens. I recommend this because developing from the editor is much faster than deploying to the HoloLens each time.
5. Coordinate systems are gross, but beautiful at the same time. They’re your friends, but also really gross.
6. If you don’t know about rotations, look them up. A quaternion is a change in rotation about some key angles. Important concepts to know are:
   1. Yaw: rotation about the axis aligned with gravity (z for some things, y for Unity)
   2. Pitch: rotation about the x axis in unity
   3. Roll: rotation about the z axis in unity.
7. Even if it works with Unity, it doesn’t mean it will work on the HoloLens.
8. Test your button placement with the HoloLens. Buttons might appear more clickable than they actually are.
9. Any button that is clickable, can be clicked by gazing on it (until it is highlighted in Ventana), and saying, “Select.” When you’re cloning a hologram from a VuMark on your phone, this is especially helpful.
10. Do the HoloLens 101 Tutorial. Right now. Bookmark it for later.

Some tips about demoing Ventana:

* With mixed reality capture, the live preview feature does not work when Vuforia is using the camera. To turn off Vuforia, the user should say, “stop recognition.” The light camera light will go off on the HoloLens, and live preview will work. To turn Vuforia back on, the user should say, “start recognition.”
* With mixed reality capture, the record feature will work with Vuforia, and can be utilized at any time when Ventana is running, without making any changes first.
* If a user has not used the HoloLens before, have them complete the learn gestures tutorial first.

A typical workflow for adding a new device:

* The UI design for a new controller, designed in Blender, or a similar program.
* The VuMark design for the new device.
* Importing the newly created blender asset in Unity to make a prefab for the new controller, and adding the necessary scripts to it.
* Updating the Unity scripts that pertain to the list of devices and corresponding VuMarks to include the new controller’s ID and VuMark.
* Building the server module for the new device, or expanding an existing module.

Some features that would be really cool to see in Ventana:

* Using the Keyword Manager to add voice control of certain things.
* A “home base” that displays the controls of all active devices.
* Multiple room support (load a room instead of load last session).
* More devices, more devices, more devices.

You’re in for a fun year! Have fun, work hard, and make Ventana even more awesome!

All the best,

Team Ventana