Data View

Sam Sweere, Julian Besems

s4403142 s4783751

New Devices Lab 2018, Radboud University

Concept

The main idea of our project was to use the Hololens to be able to instantly observe different data in the user's direct surroundings. This concept could be implemented for various different reasons, either as an extension of domotics, the option that we chose to develop, but possibly also for professionals or researchers that find themselves in situations where they have to observe multiple different measurements associated with various sources in real time. This system would give the user a means to do this in one glance if the measurement equipment would be plugged into the server side of this project.

Figure 1: Artist impression of the project

Development

At the start of the project the following goals were set that we were aiming to realise:

- First display real time data in Hololens
- Make the data hover above the relevant object
- Implement real time sensor data
- Extend to more sensors of various types in more locations

The first steps in the projects were indeed completing the first two goals. We then however found out that the third and fourth point would mainly involve development on the server side of the project, which was less interesting that working with extending the functionality specific to the Hololens. Therefore we opted to not spend more time on realising these points, but rather spend our time in making the Hololens not only receive data but also send data, which would then be interpreted by the server in order to perform certain actions.

Product

The hardware in our project consisted of a raspberry pi serving as the server, and a Hololens.
The raspberry pi sends the collected data of various sources to the Hololens, and interprets the commands sent by the Hololens to then perform the associated action on the relevant device. On the Hololens an app was created that gives the user the possibility to open various data panels, that displays the data received from the server, and possibly features buttons that can be clicked to send commands back to the server.
The programmed software running on the was written in Python, which includes the program for the local network TCP server, and a controller for VLC called by the server. The app on the Hololens was created through Unity, with scripts in C#.
Aside from the menu that features buttons to open the different modules, the final product included the implementation of four different modules.

Clock panel

This was the initial panel that we created in the app. It's a simple panel object in Unity, with text on it, which shows the time that was sent to the Hololens by the server. The C# script associated with the object, GetTextFromServer, is responsible for fetching the time from the data received from the server, and calls the ServerClientScript to retrieve this.

The ServerClientScript is the main program of the Hololens app, establishing the connection with the raspberry pi and parsing the received data of the server, and in turn sending the data back to the server. In order to be able to test this in Unity as well for debugging purposes, the code is partially written for the Unity compiler, and partially written for the UWP compiler.

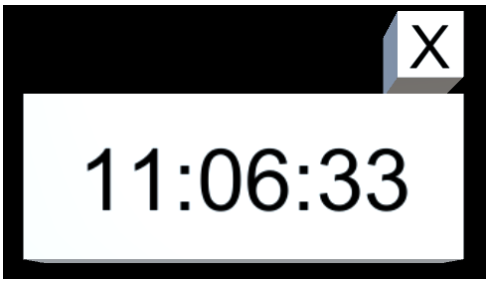

Figure 2: Clock panel

Weather panel

Similar to the Clock panel this panel also implements the GetTextFromServer script to retrieve the information to be shown. The main difference is that the data consists of various different fields, and that server-side the data was retrieved through the openweather API rather than simply the python time library.

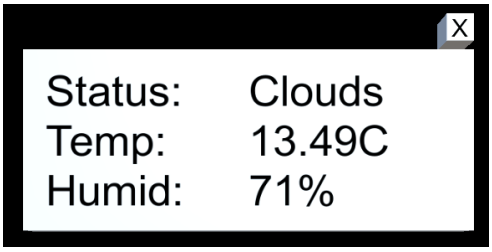

Figure 3: Weather panel

Music controller

Aside from receiving data this object also features buttons that enable the user to send commands back to the server. The object consists of three bars, one serving as a handle to grab the object in order to move it, another showing the title of the song currently playing, and one featuring a bar showing the time of the song and the duration of the song. This information is again obtained through the GetTextFromServer script. The movement of the time bar is handled by the MusicMovingBar script, making the line move proportionally to the progression of the song.
Below these three bars there are 5 buttons: pause/play, previous, next, volume +, volume -. These buttons have on click commands that trigger sending certain commands back to the server. The sending of the data is handled by the SendToServerOnClick script. The server parses the received data, in order to identify the given commands. The commands are performed by calling the vlcPlayer program, which uses the vlc-python 3.0 API to play and control music on the raspberry pi through VLC media player.

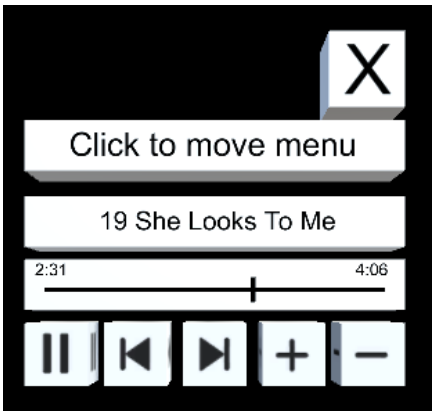

Figure 4: Music controller

General functionality

In order for the panels to be placed in the relevant location the app uses spatial mapping to map the room. Each module can then be grabbed by pointing the cursor to the object and making the Hololens tap gesture, the user will then be able to move the object along the environment of the room, always with a slight offset of the surfaces so that the object doesn't get stuck in a wall. The orientation of the object is such that when moving the object, it keeps facing the user. Once the object has been placed the location and orientation is preserved. If a user doesn't need the module anymore it can be closed by tapping the X button on each object. These actions are handled by the following scripts: WorldCursor, TapToPlace, TapToDestroyParent, GazeGestureManager.

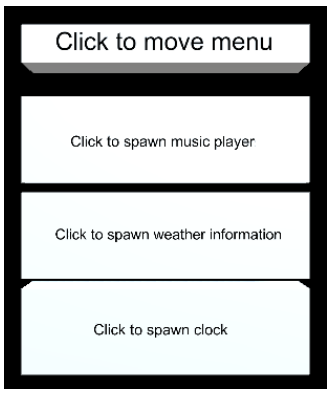

Figure 5: General menu

Bugs

As far as we found out during testing the current version contains one bug, which is that after the first button press, each next press performs the action that should have occurred by the previous press. So say we start by pressing the play button, and then proceed to press the volume + button twice, the actions that will be performed are: play, pause, volume up. Instead of the desired play, volume up, volume up.

Installation

Hardware used:
Raspberry pi
Hololens

Software used:
Unity
Python 2.7
VLC, 32 bit version
Visual Studio 2017

How to install the software: https://docs.microsoft.com/en-us/windows/mixed-reality/install-the-tools

On the Raspberry pi VLC needs to be installed. Other than that the python-vlc, and the pyowm package for the openweather API's are used. Then to run the server, the server.py file should be run using python 2.7.

When running the server on a new network make sure that the hololens is connected to the same network, also update the ip-address in the serverClient script in the unity files.

To build the Hololens app, the project should be built in Unity, after which the Hololens should be connected with the computer. Then open the DataView_Hololens.sln file inside the App folder. This should open Visual Studio. Next make sure that the app has the rights to use the local network in the appxmanifest file. Finally deploy the app as a release for a x86 platform to device. The first time for a new build this can give an error. Just click don't continue and try again. The second time should be the charm. Now the app should install on the Hololens and open it automatically. After this the app is installed on the Hololens and can be opened from the Hololens menu.

For more in-depth instructions check out: https://docs.microsoft.com/en-us/windows/mixed-reality/academy

To login on the hololens we used a dummy account:

Dummy account windows:
Login: ndlHololens@outlook.com
Pass: holoNDL2018
Birth date april 7 1993 Netherlands

Hololens remote viewer login:
Username: ndlHololens
Password: holoNDL2018
Ip adress (asus router 26-02-18): 192.168.1.144