Bi-Weekly Report 4

Date: *27/11/2015*

Project: *MSF Google Project 1*

Project Title: *Video and sensor display on the Google Cardboard*

(Previously: Alzheimer's Experience using the Google Cardboard)

Team Members:

Garrett May (Team Leader)

Hekla Helgadóttir

Overview

We have agreed with all parties to focus on sensor readings and displaying sensor information on the Google Cardboard as our proof of concept for this term, but have adjusted our schedule for next term to include features desirable by MSF.

We are currently working on plotting graphs from data and displaying on the Google Cardboard, and implementing the features for the User Interface, including indicating when looking at a certain graph, and allowing the user to click on the graph for a full screen version to replace the main menu.

Progress has continued on the Google Cardboard application, with several successes having occurred. We were able to greatly improve our prototype model, adding boxes that will eventually hold our graphs, and adding the ability to render images. Image rendering is at the very least acceptable in terms of rate of switching images, but will possibly be improved in the future.

Meetings

Date: *19/11/2015*

Attendees: *Garrett May*

Hekla Helgadóttir

Carla Hyenne

Daniel Eldar

Usama Inam

Aron Monszpart

We talked about our previous meeting with Dr Patty Kostkova and the representatives from MSF, and explained the MSF requests focusing on video to Usama Inam. As we had previously talked about focusing on the sensors and displaying information from different sensors on the drone, we had ordered parts for the sensor part of the project, and started

research on drawing graphs to display on the cardboard. We decided to continue the sensor part of the project for this term.

Usama agreed that the usage of a drone for MSF would be ideal, but also agreed we should make that a task for the second part of the project, in the second term. We will change the proof of concept slightly to align with the requirements requested by MSF, but will in general keep to our schedule. Using Google Cardboard to sense head movement, translate to instructions for a gimbal with a camera on the drone – effectively control a camera on the drone with your head – would be a desired functionality for our application, as well as being able to apply data analysis on the video / image footage from the drone.

We also displayed a very early prototype of the menu boxes on the cardboard demonstrating how we intend to display graphs in the Google Cardboard application. As the requirements of MSF and the requirements of Google were quite different, we agreed on a compromise; to have a 3D menu be displayed in the app, with boxes, each showing either a graph or a video stream. Clicking on a box will allow the content to zoom in, and for a video stream, it will also allow the Google Cardboard's head rotations to change the direction the camera faces on the drone. Since sending over video data quickly is difficult, the video stream will be of low quality. Instead of video analysis, image analysis will most likely be easier and more useful; say, analysing several images, and noticing a certain population of people are moving. Through this we would also be able to display population as a number or a graph, as needed.

As we wish to show a prototype of the application in the coming weeks, we chose to focus on finishing the basics of the Google Cardboard app. This was to be able to show the menu in 3D, and to at least be able to display one graph moving in real-time, using dummy data. Our client agreed that this would be a good idea, as it would show a basic example of the app, which could easily be scaled up if necessary.

We discussed what would be efficient methods to get a real-time graph up and running optimally. One suggestion was to use a JavaScript graphing library, such as D3.js, to create graphs on an HTML page, and to then render these HTML pages in the application on the boxes.

Completed Tasks and Project Projection

We have been able to create boxes in our Google Cardboard application, allowing us to show a visualisation to our client of what our prototype intends to look like. We were also able to render an image onto a box, by making use of textures, and to change this image successfully.

The boxes in front of the viewer have become the layout for the main menu, where we can see small versions of the data we will be showing, and can click to view the relevant graph or video full screen. We have a basic working Google Cardboard application for this.

We are going forward with graph plotting and displaying sensor information, but have scheduled creating a desirable video feature for our drone project in next term. We are still

waiting on the sensors and transmitters, which have been ordered, and are using dummy data for plotting the graphs until we get the parts.

We are working on uploading the graph and the video to the main menu, to offer a live overview over the information offered. We have experimented plotting the graph based on sensor readings to a HTML page using D3.js, take a picture and upload as texture to the "boxes" on the main menu. This has proven to be rather slow, but with further experiments it seems that if we change the dimensions of the images, we can improve the speed.

We have also had a response that some parts have been received. We can begin looking up documents to learn how to program them, and once we have received all components, we will be able to begin creating the drone and networking systems.

Problems to be resolved

We have encountered several problems over the last couple of weeks, some which we have found possible solutions to. Uploading the graph or video as a texture to the "boxes" is proving challenging, but we are looking into changing the dimensions of the images to ensure speed of uploading them as textures. We could also possibly draw the graph in 3D to avoid the updates to be too slow, but that is a challenging way of addressing the problem.

In our most recent team discussion, whilst going through the possible methods in which to display the graph, it became apparent that loading an image onto a texture each time would not be as fast as we would like, and may look less aesthetically pleasing. We had also looked into HTML rendering in OpenGL; however, this proved fruitless. Instead, we intend to draw our graphs, using OpenGL, to a Frame Buffer Object. This can then be used as a texture on an object for displaying. By doing the graph rendering this way, it may be possible, if needed, to create our own graphs in OpenGL, allowing a greater freedom and control over the look and feel. Otherwise, at any rate, a graph created in OpenGL should be much faster than using a JavaScript library, as the method calls would be done in the same Android application as the rest of the Google Cardboard app. As well as that, by utilising Frame Buffers in our drawing, the result should be a much faster rendering speed, reducing latency and minimising any loss of immersion within the project.

We also discussed the networking side of the project, as it would be useful to consider how this will be built before all our components arrive. Our recent views are to take an Arduino and to connect it to Telemetry modules. We hope that this will enable us to send sensor data and video data over long distances, though we may have to look into this, as we may need more processing power from the hardware (a Raspberry Pi might be a good alternative).

Encrypting the data sent would be desirable, and to be able to livestream both the video and the sensor readings, however these are not top priority at the moment.

Workload - Past Two Weeks

Garrett:

I continued working on the Google Cardboard application for our Proof of Concept.

I first decided to find out how to create a box in OpenGL, as this was going to be the most fundamental concept needed to progress in our application. After some time, I was able to render a box by uploading its model matrix, its position matrix, and each eye's view matrix and projection matrix. This allowed me to make a menu, with around fifteen boxes displayed in a convex manner, which faced towards the viewer. The idea was to create several graphs (and possibly a video) that acted as monitors, which the user could choose at whim. Each box was represented through 3D polar coordinates, with a radius, elevation angle, and polar angle. This allowed a more natural and easier way to program the menu.

Having created this, I chose to improve the shaders needed for our program, in order to correctly colour and texture the boxes. The shaders were coded in GLSL, OpenGL's shading language specifically used for a project such as this. Once I had generated a texture, I was able to transform a PNG image as a Bitmap, and then load it into the texture, so that it could be used on a box.

Because we were rendering fifteen boxes, this initially had a large latency. We found that this slowness was mainly due to the fact the images should be of a power of 2 (e.g. 16, 64, 128), in order to optimally render the image. Changing the images to a smaller resolution that satisfied this was the solution to the problem.

The final task I was able to complete this week was to detect whether a box was being looked at and to zoom it in on input. This was done by changing a box's radius in a function which was called when the Cardboard's trigger was pulled. Thus, when rendering for each eye, the selected box would be placed closer than other boxes.

Hekla:

I have been looking into the zoom function, to make the desired video or graph appear full screen when clicked on, and then to go back to main menu when the button is clicked again. I have found a way to achieve this using the JavaScript library Three.js, and have experimented with a simple demo app. I will have to integrate it with our current main menu.

I have also researched how to display images taken from the graphs or the video, on the main menu boxes. We might be able to use the same library for this.

Workload - Next Two Weeks

Garrett:

Due to our most recent meeting, I will most likely hide the image texturing, as its only use is as a possible backup, and work on creating FBOs, for making draw calls to. Afterwards, I will need to apply an FBO as a texture on each box, so that our graphs

can be displayed on the boxes. We will need to check whether using FBOs are indeed faster than the simpler image texturing, as latency is one of the most important points in our project.

Hekla:

I will continue working on the zoom function, and to integrate the function with the main menu and the rendered images. I will also find out how to display in a visible way which “box” from the main menu is being gazed at, making it easier for the user to tell which item he will have full screen when clicked on.

I will also work with Garrett on creating textures from the video and from the plotted graphs to be put on the main menu boxes.