MST Google Project 2 – 11/03/16

Daniel Eldar, Carla Hyenne

Overview of previous week

This week did not have significant progress. Individually, all members are completing their work according to the timeline so everything is on track. What has been started is the 3D terrain representation from the 2D areal view, and retrieving images from the Open Street Map data.

Meetings

08/03/16

Attendees:

Garrett May
Carla Hyenne
Aron Monszpart

Overview:
A brief meeting where we discussed retrieving GPS coordinate data and Open Street Map images, as well as the server to set up for our final production.

Plan for the upcoming weeks

The priorities and tasks we have to complete are as follows:

- Finish sensor data display, and save this data in an NHS file format through a database
- Finish implementing the gimbal functionality to allow the movement of camera with head movements
- Set up the video transmitter and receiver
- Collect GPS coordinates from the drone
- Collect GPS coordinate data for landmarks via OpenStreetMap, as well as collect an image of a map by providing coordinates
- Terrain generation, and add pointers to landmarks (points of interests) on the terrain representation

Work Packages Completed

Carla Hyenne

I have been working on implementing the Mapnik module that will translate Open Street Map XML data to an image that we can render on the Google Cardboard View. However I have been unsuccessful with using this API, so I switched to querying the OSM server for data. This allows us to retrieve nodes (points of interest) and ways (polygons or areas on a map) by providing key and value pairs. We can also retrieve bounding boxes (taking four coordinates as arguments to return a rectangular area). This can be saved as XML data, and I am working on getting an image for display.

Daniel Eldar

I have been working on porting my demo of the pointer code over to java and displayed on the cardboard as opposed to the original C++ code which would have needed special requirements to run on Android. As the OpenGL API is very similar on both C++ and Java it has been straightforward so far. However, the maths library used in C++ needs to be changed (GLM - OpenGL Mathematics) as it is not available on Android.