

Individual Lab Report #3

Sumit Saxena

Team F: Rescue Rangers

Teammates:

Juncheng Zhang

Karthik Ramachandran

Xiaoyang Liu

October 28, 2016

Contents

1. Individual Progress	1
1.1. Overview	1
1.2. Basic UI prototype.....	1
2. Challenges:	3
3. Teamwork:	3
4. Future plans:	4

1. Individual Progress

1.1. Overview

As dictated by the goals we had set for ourselves for the week, my major responsibility was to prepare a basic user interface to get likely search locations as inputs from the user on a map and convert them into respective GPS coordinates. This is important because we are using GPS to localize our aerial system in the environment and thus, we need to give the system, specific GPS coordinates as inputs in order to enable it to reach desired search locations.

1.2. Basic UI prototype

Main objectives for this task were:

1. Build a basic UI to get desired locations as inputs on a map (in form of placing a marker at the desired location, etc.)
2. Convert these location inputs into corresponding GPS coordinates

Some initial research revealed two probable approaches I could take:

1. Static map approach:
 - a. Download an area map as image with latitudes and longitudes known for certain intermediate locations
 - b. Write a standalone program to take inputs from the user on this map
 - c. Convert the location inputs into GPS locations by interpolation of latitude-longitude information available for intermediate locations
2. Dynamic map approach:
 - a. Write a Javascript to run in the browser to load Google maps
 - b. Utilize Google maps API to take user inputs and get corresponding GPS coordinates

Although I could not get into exact implementation details for both, I spent some time to get a fair idea of how both the approaches might work and compared the pros and cons.

Static map approach	Dynamic map approach
Pros: <ol style="list-style-type: none">1. Standalone offline program, no dependency on internet availability2. Highly customizable	Pros: <ol style="list-style-type: none">1. Flexible: You get to use the features of Google maps2. Looks more realistic (It's Google Maps)3. More accurate GPS locations4. Interesting to learn to work with Google maps API

<p>Cons:</p> <ol style="list-style-type: none"> 1. Approximate GPS locations as you take the map to be exactly flat 2. Time-consuming: need to code everything from scratch 	<p>Cons:</p> <ol style="list-style-type: none"> 1. Dependency on internet and browser 2. Limited flexibility
--	---

Since our aim was to build just a basic UI to report GPS coordinates and to do that in an accurate manner, it made sense to go with the Dynamic map approach. In addition, I was excited to learn to work with Google maps API. So, I went with the Dynamic map approach.

Programming platform used:

While browsing through the Google Maps JavaScript API tutorials, I came across ‘JSFiddle’, which is a custom environment to test JavaScript, HTML and CSS code, right inside the browser. It seemed fairly easy to use and thus I used it to make the prototype.

How I made it work?

1. Created the HTML page with placeholders to display the map and GPS coordinates for selected locations
2. With the help of Google Maps JavaScript API, I loaded Google maps and created ‘draggable’ markers on the map
3. Used addListener() event handlers to report the GPS location of the markers on the map, when they are clicked

Result:

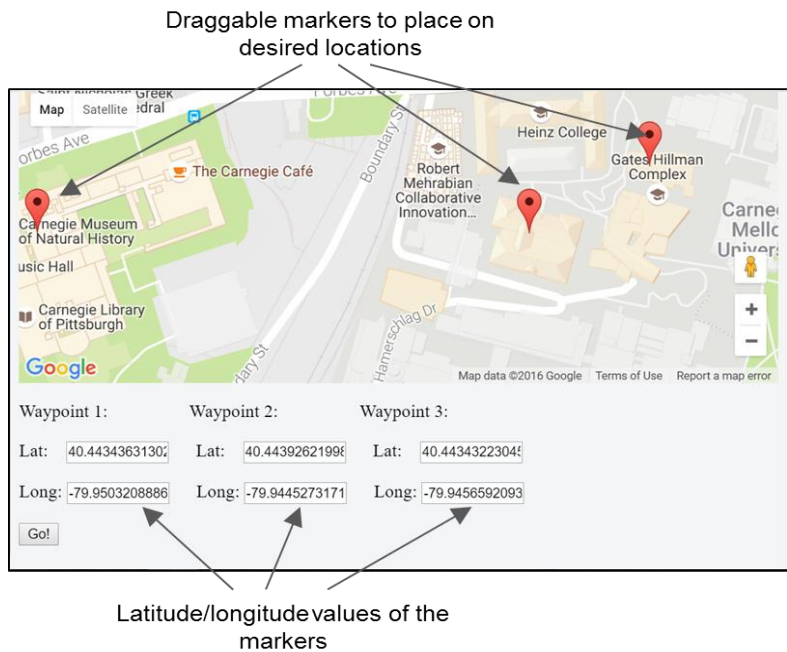


Figure 1: Basic User Interface prototype

As shown in the figure above, the UI gives us three 'draggable' markers (for now; we may increase the number of these markers in the future). The user can drag these markers to the locations he/she thinks might be best to search for the lost human being. After placing a marker at the desired location, the user needs to click on the marker to report its GPS location in the fields below.

Then, the user needs to click on the 'Go!' button. This will send an HTTP request to the server, which our base station system will operate on, with parameters as latitude-longitude information of the input locations. The server will further use this information to generate paths for navigation.

2. Challenges:

I faced following challenges during my work:

1. Starting with no prior knowledge, it was a bit difficult for me to think about how I could approach the problem. A good amount of research on how people have implemented such things before, helped me get a fair understanding of the approaches I could use.
2. After I was able to report the GPS coordinates of the desired locations in the UI, I was faced with the challenge of how to carry these GPS locations forward. I was trying to figure out a way of saving these GPS locations in a file and download it on the press of 'Go!' button but it seemed that it was not a straightforward thing. I had a discussion with Karthik and it became clear that there was a much simpler way of carrying these locations forward by passing them as parameters to an HTTP request to the server we will use.

3. Teamwork:

Work done by individual team members:

- Juncheng Zhang and Xiaoyang Liu:
 - PDS PCB Schematic design: They delved into details of the components required for the PDS PCB and prepared the schematic
 - They are also looking into algorithms we could use for human detection in RGB images
- Sumit Saxena:
 - Implemented prototype UI to take likely search locations as inputs from the user on a map and convert them into corresponding GPS coordinates for further use

- Karthik Ramachandran:
 - Implemented prototype software to accept GPS coordinates of the desired search locations as inputs and generate GPS coordinate waypoints around those locations to define the path (configurable for now) our hex-rotor should take to conduct search at those locations

Since my work directly feeds into Karthik's work as inputs, collaboration with him helped me figure out the way I could communicate the GPS coordinates to his software, which apart from simplifying my problem, helped us better structure the process. Close collaboration within the team and frequent discussions on the steps we take, till now, have helped us look at every situation from multiple perspectives and gain a clearer understanding.

4. Future plans:

Following are the tasks I plan to work on until the next ILR:

1. Improve the UI and make a standalone program (explore options to migrate the code from JSFiddle to some offline platform)
2. Work on integration of UI with the waypoint generation code, which Karthik has been working on
3. Work on DJI Matrice 100:
 - Assemble
 - Gain familiarization with simulator
 - Test teleoperated flight and basic autonomous waypoint navigation
4. Work on DJI Matrice 600:
 - Assemble
 - Gain familiarization with simulator
 - Test flight at an external site
 - Test portability of code between Matrice 100 and Matrice 600
5. Microphone testing:
 - Test feasibility of using a microphone in presence of drone's propeller noise

Tasks 1 and 2 will be completed in close collaboration with Karthik and will result in a software which accepts likely search locations as user input on a map and generates waypoints that will define the path our drone should take to carry out the search.

Tasks 3, 4 and 5 will be carried out by me/Karthik but there is no clear division of work as yet since we are still to get our hands on both the drones. The division of work will depend on when they become available to us. We are quite positive that we will get to work on both of them during the next two weeks and will be able to divide the work accordingly.