

Assignment 3. Mobile Tracker of VGI

Due Mar. 11, at 11:59 EST

Worth 25 points

Using your assigned teams, you will create a web application that uses the GPS on your mobile device to track your team's movements when they have the html page open your team will create. The crowdsourced data will be displayed in an online map.

Goals

Learn how to combine hardware and web-based software tools to monitor and record users' VGI in space-time.

Write your first Javascript (JS) "app" (technically webpages).

Learn how to store data created in HTML/JS online - in this case as csv and json

Learn about distinguish different crowdsourced contributions in a database.

Learn how to display data on an online map.

Individually reflect on the assignment.

Resources

- https://www.w3schools.com/html/html5_geolocation.asp
- https://www.w3schools.com/js/js_timing.asp
- https://www.w3schools.com/js/js_dates.asp
- <https://docs.mapbox.com/mapbox-gl-js/example/simple-map/>

Tasks

1. You need to build an app that allows a user to collect data their phone's location over time. This is accomplished through JS and relies on several factors including the phone's location service hardware (usually a GPS). You will do this with JS embedded in an HTML page that will be opened on your phone in a browser. This one page needs to do numerous things:
 - a. The page needs to ask the user to enter (or select from a list/options) their user ID. It is perfectly acceptable for you to limit the users who can use the page (e.g., only users who are in your team and whose ids are in the list).
USER IDs MUST NOT CONTAIN SPACES, and only alphanumeric (A-Z, a-z and 0-9).
 - b. The page should display some typical things like: an app title, logo/picture, descriptive text. It must display somewhere on the page the user ID that was entered in the first step. (e.g., You are logged on as: USER1)
 - c. Take a screenshot of your user interface (i.e., what the user sees) 

- d. You need to write JS that collects the path travelled by the user while website is open and the tracker is enabled. It should get locations of the user's phone and the date/time for those locations every 30 seconds while the page is open.
2. You will need some place in the "cloud" to store your data. If it was stored on your phone alone then you would not be able to see anyone else's data. Plus as soon as you close your phone's browser, the data would disappear. We were hoping it would be easier (e.g., via direct upload to Google Sheets) but nope, Google changed things. Therefore we are going to have you "brute force" add to a .csv file.
 - a. In your webpage above, you will create a the function 'writeToCSV'. Because it is not obvious how to do this (and because Google!) we wrote the function for you. It can be found in the [writetocsv.js](#) under 'Week 7' on the course website.
 - i. Add the "writeToCSV" function into the JS section of your webpage.
 - ii. Save the [formatAsJson.php](#) file into the same folder as the webpage.
 - b. Every 30 seconds, you will need to send your data to this function.

For Your Understanding: *the programming language PHP, unlike Javascript, has the ability to easily read and write files. Here we use it to write our datasets to .csv and .json files.*

How it works:

The "writeToCSV" function will send your collected data to the [formatAsJson.php](#) file as often as you call the function. When it receives the data, [formatAsJson.php](#) saves your collected data to some files that it will create: [data.csv](#) and json files for each user (ie. [user1.json](#), [user2.json](#)).

[data.csv](#) is a csv record of all entries to the dataset

[<USERNAME>.json](#) are geojson files which can be read by the Mapbox API as line features.

These files will automatically appear the first time you run the function, and update accordingly.

BE CAREFUL: if your variables are wrong these files will not generate in the correct format, and you may have to delete them and run the code again.

3. Use the app to crowdsource location data. Instruct the user to load the app page and track themselves as they move around the city - this will should not use a lot of their data plan. The idea here is to show movement. We will not accept data that shows a stationary user (lots of dots all in the same area).
 - a. There is no limit to the amount of data you collect, but you should plan on at minimum tracking the user for at least one hour of movement.
 - b. A user can access the website to track themselves on multiple occasions to meet the tracking time goal but they should always be moving when they are tracking themselves.
 - c. It is a good idea to check the .csv on the server occasionally to see that it is receiving the data from the phone.
4. Map all the users' paths. Use Mapbox for your geoweb platform.
 - a. Set up an account on Mapbox and get an API key.
 - b. Create a second HTML page, with the needed JS code to embed (mashup) the map.
 - c. In that page, you will need to input the data collected by your users and have it display on the map. Use the [<USERNAME>.json](#) files and Mapbox javascript to achieve this. The paths need to be in different colours to represent each user.

You will find this function that Ian wrote useful for importing the geojsons into Mapbox:

```
//add a geojson(json) layer to the map
//replace everything within the < > appropriately
map.on('load', function() {
  //the url (name) of the .json file - must be in same folder as the javascript
  var url = '<USERNAME>.json';
  //add a source
  map.addSource('<SOURCE NAME>', { type: 'geojson', data: url});
  map.addLayer({
    id: "<THE USER'S NAME>",
    type: "line",
    source: "<SOURCE NAME>",
    'paint': {
      "line-color": "<HEXADECIMAL COLOUR eg. '#a63504'>"
    }
  });
});
```

// <SOURCE NAME> is like Style Url from KML. you can call it anything, but you have to repeat the name where it is needed (2 places here). This should change to a new name for each dataset.

- d. Take a screenshot of the map that contains the user coloured data paths 
 - e. **For extra credit**, augment the user interface so one can choose which user's data to display.
5. You should use css or html templates to make your pages look cohesive (i.e., part of a single app) and attractive.
 6. Upload all the files, including data.csv and the <USERNAME>.json files, your two webpages (tracker and map) and the writetocsv.php file in the same folder in your team's account on neogeoweb.ca.
 7. Your team needs to document your application in a report. That same report needs to include your individual reflections. In a word document your team should provide the following information:
 - a. Describe the workflow in your application (~1 page, which excludes screenshots but could include a diagram).
 - b. In approximately one page describe the process of collecting and sending the locations from your phone.
 - c. In no more than one page describe how you used Google Sheets
 - d. In approximately one page describe the process of setting up Mapbox and mapping the locations from your phone.
 - e. Include a user id so Sam can test the app.
- Individually (i.e., each person) reflect on the assignment (~0.5 pg/person)
- f. What was the most challenging part of the assignment?
 - g. Were you surprised at how easy it was to collect a user's space-time signature? Why or why not? Draw on material discussed in class to answer these questions.

- h. As part of your surprise (or not), does the tracker keep working if your phone's screen is off? Or if you close your browser? What about if you are not using the browser on your phone: is the website still running somewhere in the background?

Submission

You have two submissions

1. Email us your report as a single .doc/team, including the screenshots (renee.sieber@mcgill.ca and sam.lumley@mail.mcgill.ca) and individual reflections.
2. Upload your application to a new folder in your group's directory on the neogeoweb.ca server (you will need to do this anyway because it is important for your data collection and mapping)

****NOTE:** your FTP credentials will be deactivated at 00:00 on Tuesday, March 12th to ensure no files can be uploaded past the deadline.