CS 419 Winter, 2016

Group 18 Weekly Progress Report 3

Jennifer Erland
Charles Hardes
Ty Hatton
Jason Murray

o Progress during the past week:

The data layer is now in the form of vector tiles, which allows for only the zip code areas in view being drawn. This greatly decreases the load time, and has been successful on multiple mobile devices; whereas the previous methods would not work well on mobile. The fuel price data is now embedded (offline, beforehand) in the topoJSON file that contains the topology data for all the zip codes. By doing this, we are able to have all necessary data (locale name, zip code, and fuel prices) in the same data layer. Again this is accomplished using Mapbox, and the Mapbox GL JS API.

o Plans for the upcoming week:

The basic functionality of both the home and main pages are complete, but still to do:

Home page

- Look up latitude and longitude values corresponding to user inputs
  - This should be an external .js file (see below)
  - User inputs: either city, state or zip code
  - This will be passed as a GET parameter to the main page and used for the initial center of the map

Main page

- Styling : some degree of styling is needed…currently an unstyled div
- Search box: would be useful to have a search box so that new zip code or city,state can be entered
  - This would call external .js file (above) to return a lat/long, then move to this new location with the panTo() method
- Legend: need a legend displaying colors for corresponding prices

Miscellaneous

- Data source for fuel prices
- External .js to determine the intervals and colors for fuel prices
  - Currently the colors and intervals are pre-defined; at the very least the intervals should be determined based on the fuel prices

- .js could look at .csv file, then for each column/fuel type calculate the min and max, then return intervals (how many? 10?).
- Could also have this return the colors for each interval. Again, this is also currently hard-coded

o Any problems you encountered during the past week:

The only problem(s) would be the learning curve of becoming familiar with a new API (Mapbox GL js), and using Mapbox studio. Fortunately both are very well documented. The main concern now, is to ensure that any increase in data does not correspond to vastly increasing the loading time on mobile devices. For example, if another layer is added, how much data would make this unusable?