Version 1.7

First thing done on Thursday, 2/5/15 at the TSA meeting was pruning intersection connections which would clog up the algorithm, which include duplicates and connections with .distance\_to = 0.

After that, I made the inputs for start and end locations variable.

I tested out the last improvement with using “1000 N Peace Haven Road, Winston-Salem” as the end location. I had built in an exception to deal with pts that had no road\_data, but I had no idea what was causing it. I rebuilt my algorithm to distribute road\_data more efficiently, but it still did not solve the problem. Then, after noticing large swaths of pts having no road\_data even though there was a valid road name, I checked the coordinates of one of the points, and it was not inside the bounds of any of the overpass data. This was done the night of Saturday, 2/7/15. I realized that the bounds did not account for all the route queries, and I wrote a short function to find the bounds of all the points. This was done on Sunday 2/8/15. I also realized that in order to write a Djikstra algorithm I would need to know the index of the start and end intersection. The start is the first and last elements of pts, but I do not know the index of the end, and therefore cannot search for it.

Wednesday 2/11/15 and Thursday 2/12/15 I created a short function to make all the possible paths through the intersections, compiling a list of lists of indexes. The algorithms starts at a point, and gets a list of the indexes that connect. Then, it iterates through that list, and branches off to the indexes that it has not gone to before, and repeats this process until it either has already gone to all possible points or has reached the end. If it reaches the end, it returns a list of indexes and a cost. After the process is finished, it compares the energy cost and chooses the route with the lowest one. I needed to build another function to filter out mistakes from some problem I will identify later.

I added instructions on Monday and Tuesday, which at a change of road name, decides the angle of the turn and makes text for instructions. It also draws a Polyline on the map, which two points for start and end.

Finally, I made the input box turn transparent when the mouse is not over it, to improve visibility of the map.

Icons from: <http://miftyisbored.com/a-complete-list-of-standard-google-maps-marker-icons/>