

Bronson Duran (Bduran)
Noam Habot (nhabot)

Assignment 3 Writeup

Brief Description:

Our final visualization enhances user's discoverability of trees in San Francisco. By default, all trees in the dataset are shown on the map. The user can zoom in and out, and pan around, to dive deeper into the map. Users can enable a Point of Interest (POI), which illuminates a circle on the screen. This circle is draggable and the radius is editable via a slider such that a user can specify the area of trees in which they are interested. Users also have the option of adding another POI, in which case the trees displayed are filtered by those only within the intersection of the two circles. Having draggable and editable circles make visualizing the trees very intuitive. As an added option, users can select the Diameter Breast Height (DBH) range that they are interested in and only trees with DBH's in that range will be displayed. There are also error messages if the min DBH is greater than the max DBH. Overall, the visualization is very interactive and supports very prompt response times to changes in the query by the user.

Work splitting:

We split the work evenly throughout. We worked together on plotting the trees, adding the draggable POI filters, adding the DBH range, and constructing the UI.

Resources:

We wrote almost all of the code from scratch. We used the cats and dogs example shown in class to get started, and we used a function from StackOverflow.com to compute the distance between two geographical points (pairs of latitude/longitudes). The link to it is here:

<https://stackoverflow.com/questions/27928/calculate-distance-between-two-latitude-longitude-points-haversine-formula>

Another resource we used was how to resize an SVG circle via a slider:

<https://bl.ocks.org/d3noob/147791d51cf6516715914c49cb869f57>

Development Process:

Overall, the development process was very fun and intuitive. We both have had experience using Javascript, so picking up D3 was not too difficult. It was very entertaining to see the visualization come to life throughout the process of writing the code. We spent a combined total of around 10 hours producing this code. The task that took the most amount of time was constructing the callback to filter the data on move of circles, on change of radius slider, and on update of the DBH ranges.

To view a live demo of this, navigate to:

<http://stanford.edu/~nhabot/cs448b>