The storyboard from part 1 included a pretty simple layout with the map on the left and a series of checkbox filters on the right. The checkboxes were grouped into different boxes that were shaded with different colors.

Initially, as was described in the storyboard, I simply had two circles to represent points A and B, and they were represented in the filter legend simply by a corresponding color. I showed the visualization to a few people, and this was the major pain point, as many of them did not know what to do when they first looked at the visualization. I updated the visualization to include a tooltip that would pop up if you hovered over the circle (it would tell the user to use the two circles to find points at the intersection), but this obscured the information tooltips on each data point. If anyone tried to hover over the individual data points to see the data, they would just get the helpful tooltip for how to use the home and work circles. So, I opted to create a help icon alongside the scale and filter legend in order for users to find instructions.

I also initially used purely checkboxes for my filters, including for days of week and for crime types. However, this became hard to judge because there were so many filter checkboxes. Even with color differentiation, it was quite jumbled and difficult to see how each interacted. Thus, I opted to allow the filters to toggle certain sections on and off and hide the divs that contained the individual buttons. This made it much easier to tell what filters were being used at any given time.

I also chose to bucket the types of crime into violent, nonviolent, financial (“embezzlement”, “fraud”), property (“vandalism”, “theft”, etc.), and other. Initially, I was going to offer the user the capability of filtering each type of crime, but I opted not to do so because it cluttered the interface and provided far too much information (it was also just too slow and confusing to change that many buttons each time).

I also added in a scale and title for the interface. Google Material icons were used instead of dots at the center of the circles.

The final interactive visualization features

\*filtering by intersection

\*filter by crime type

\*filter by day of week

\*filter by date range

\*filter by time range

\*shows individual data points when you hover over them

\*

The source code for the application is linked. In addition, it is uploaded as a webpage here.

<http://web.stanford.edu/~bliu3/Assignment3/assign3vis.html>

I spent about 25-30 hours on this assignment total. The majority of time in the development process was refining and improving the filtering and design aspects. I spent about 4-5 hours fiddling around with D3 to figure out syntax errors. I also spent a substantial amount of time (10-15 hours) trying to figure out how to properly and efficiently filter the data and coding in each of the filters properly. It was difficult at first to figure out how to combine filters when multiple boxes were checked. Initially, I was using a bunch of if-statements to turn the visibility of data on and off. But I got a major boost in performance when I began using Javascript’s filter() function to simply filter each condition and gradually reducing, then re-rendering a temporary array of data points. I optimized further by event handlers to not update and recheck every condition any time a single change occurred, but to just focus and update the conditions that were affected by any single user click.

The remaining 10+ hours were spent refining and optimizing the filters and design. I found a pretty good resource for tooltips, using tipsy.js and was able to provide information on individual data points. I also found a useful CSS template that provided animations for some of my filtering buttons (like days of week and types of crime).

* The description with storyboards from part 1.
* A brief description of your final interactive visualization application.
* An explanation of changes between the storyboard and the final implementation.
* The bundled source code for your application, uploaded as a webpage or file (either a .zip or .tar.gz archive) linked to your assignment wiki page. Please ensure that the software submitted is in working order. If any special instructions are needed for building or running your software, please provide them.
* For submissions by groups of two, please also include a breakdown of how the work was split among the group members.
* Finally, please include a commentary on the development process, including answers to the following questions: Roughly how much time did you spend developing your application? What aspects took the most time?