Eli Wennberg Smith

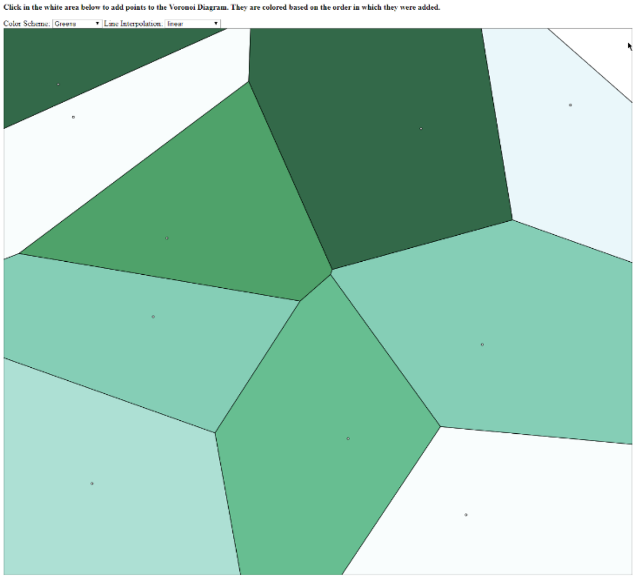
Homework 5

Interactive Visualization

2/13/2020

For This homework assignment, I revisited my project from homework 4. This project was an interactive Voronoi diagram in which the users could insert points into the diagram by clicking. The color scheme for the original project was simply random. Each region in the Voronoi diagram was colored randomly as it was inserted and did not change as more points were added. There was no data associated with the color in each region. For this assignment, I went back and tweaked the program to color the regions based on the order of insertion. I originally wanted to color them based on size of the region, but this proved to be a complex task as I am still new to D3.js and have no idea how it stores point values for Voronoi diagrams. If I did implement it this way, I would have used a diverging color scheme to display how each region's size compares to the average. But instead, I decided to color them based on order of insertion so I'm using a sequential color scheme. As more points are added, each region is updated. In addition to the changes made to the color scheme, I also started playing around with the different line interpolations that D3 provides. I saw someone else do this in a similar example online and thought it made some pretty cool visualizations. I'm not entirely sure what the different interpolations signify, but they look cool and are fun to interact with.

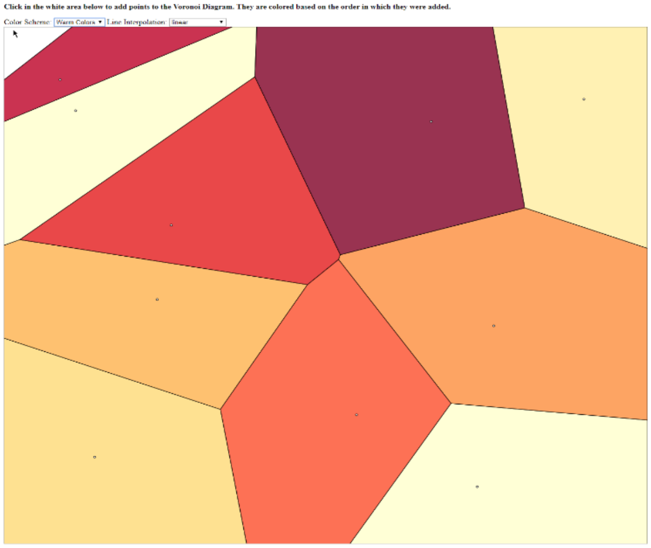
Fig 1



Click in the white area below to add points to the Voronoi Diagram. They are colored based on the order in which they were added.

Color Scheme: Greens ▾ Line Interpolation: linear ▾

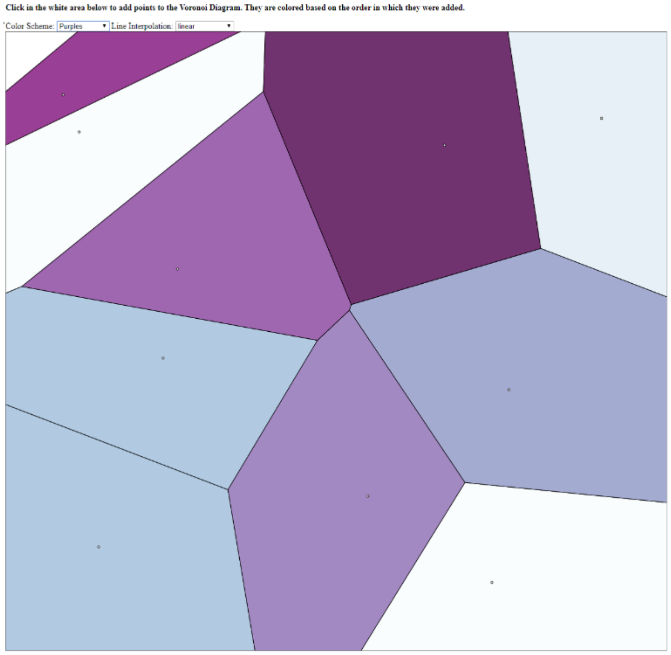
Fig 2



Click in the white area below to add points to the Voronoi Diagram. They are colored based on the order in which they were added.

Color Scheme: Warm Colors ▾ Line Interpolation: linear ▾

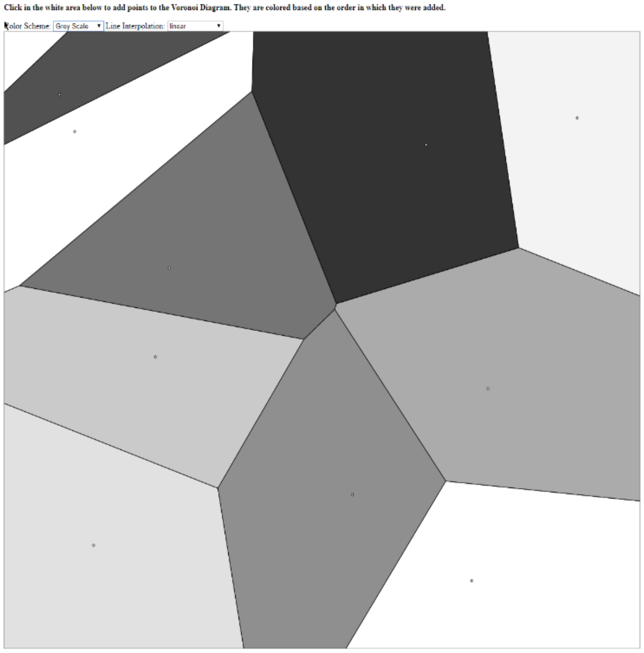
Fig 3



Click in the white area below to add points to the Voronoi Diagram. They are colored based on the order in which they were added.

Color Scheme: Purples ▾ Line Interpolation: linear ▾

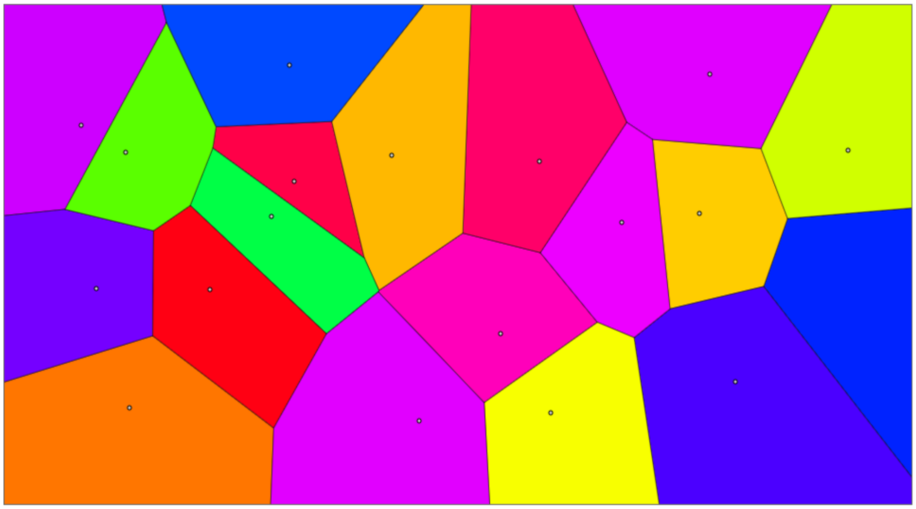
Fig 4



Click in the white area below to add points to the Voronoi Diagram. They are colored based on the order in which they were added.

Color Scheme: Grey Scale ▾ Line Interpolation: linear ▾

Fig 5

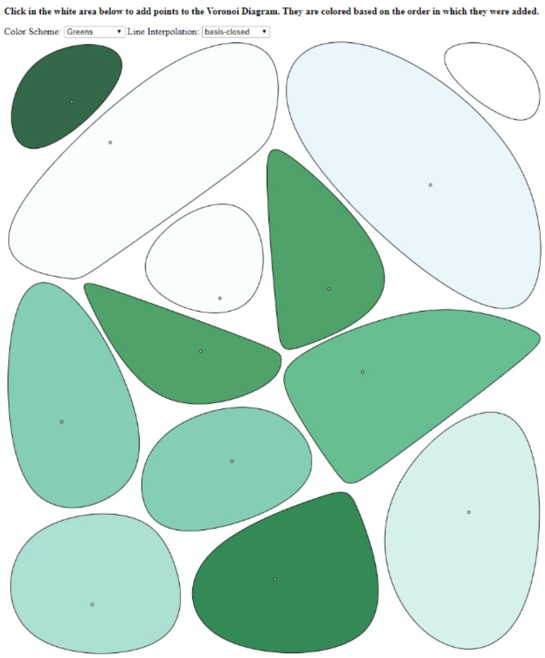


Figures 1 – 4 display the four different sequential color schemes. There are greens(F1), warm colors(F2), purples(F3), and Greyscale (F4). Because the data associated with each color is random (user input), there is not much reason to choose one color scheme over another. The warm colors may be more eye catching than the greyscale ones, but that is a matter of personal preference and is up to the viewer. This is why I made it interactive, allowing the user to choose any of these four as they please. If there was metadata associated with each of these points, or if the vertex points of each polygon were easier to obtain, the color choice would have to be more closely considered. For example, If this were a Voronoi diagram of all the fire stations in NYC, and each region was colored based on how many fires have happened there in the last decade, then I would probably choose the warm colors for that visualization as they are closely associated with fire and make it easier for the viewer to understand the data.

The original color scheme (F5), however difficult it was to look at, was actually a qualitative color scheme. Each region was assigned a unique color. However, as there was no data associated with the color, it was meaningless.

One issue with the new visualization is that there are only 9 defined colors for the program to use. This is an issue because the user can add as many points as they like. When more than 9 points are added, multiple regions can have the same color. This creates ambiguity in the visualization when there is no ambiguity in the data. To address this, I would have to implement a similar algorithm to the one used on colorbrewer to dynamically create color scales as needed. However, this only addresses the problem to an extent as after a certain number of points are added, the user would not be able to differentiate between two colors that are in fact different.

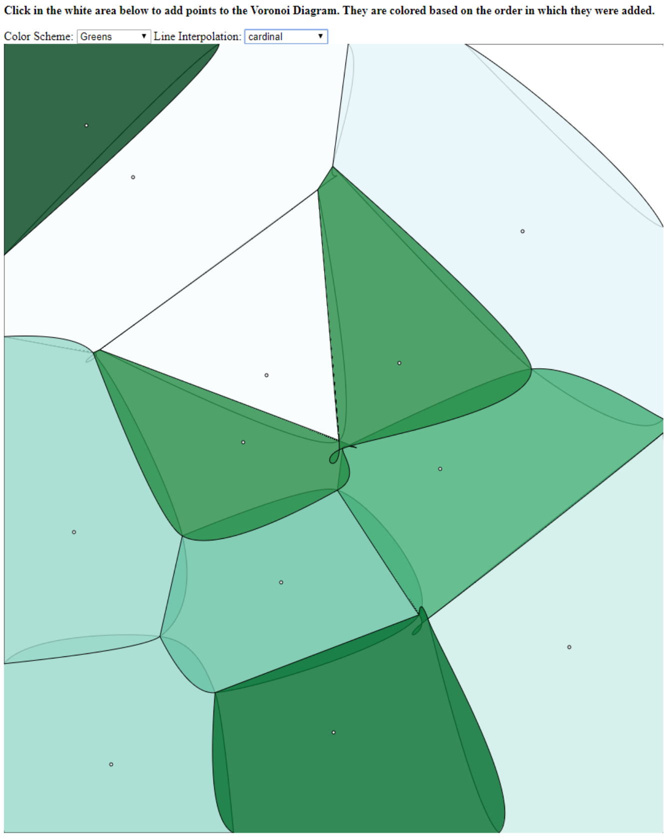
Fig 6



**Click in the white area below to add points to the Voronoi Diagram. They are colored based on the order in which they were added.**

Color Scheme: Greens Line Interpolation: basis-closed

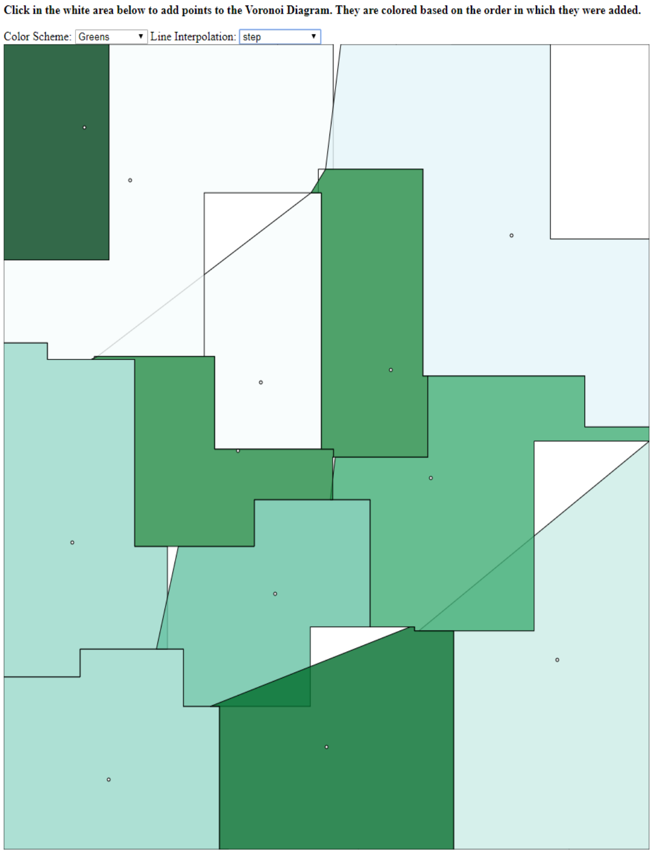
Fig 7



**Click in the white area below to add points to the Voronoi Diagram. They are colored based on the order in which they were added.**

Color Scheme: Greens Line Interpolation: cardinal

Fig 8



**Click in the white area below to add points to the Voronoi Diagram. They are colored based on the order in which they were added.**

Color Scheme: Greens Line Interpolation: step

Figures 6-8 display the various lline interpolations that D3 offer. I added this in because I thought they were very visually pleasing, especially the basis-closed interpolation seen in Figure 6. Other changes to the HTML page include a small written instruction, and the diagram now spans the entire page.

Sources:

Code was adapted from projects seen here:

http://bl.ocks.org/shawnbot/8059739

https://bl.ocks.org/mbostock/675512

Color schemes were taken from:

http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3