Assignment 8

Thursday, May 6, 2021

Part 1: Description

This website is an interactive portfolio that will present my body of work in more interesting manners apart from the usual portfolio website. The website presents about 8 years of noteworthy projects as an individual dot. The d3 presentation of this information shows each dot in relation to its topic (lines) and year (color). Users can get info on projects of interest by hovering on each dot, and the controls help to find specific types of projects or visualizations.

I found the result of this interactive display as a good overview of the kinds of work I am interested in. The time view is an interesting way to see how my body of work has changed over time too. It's maybe not the best for a recruiter if they have no way to view more information about a specific project closer, but maybe some additional integration into a bigger site would help.

Part 2: Interactions

- When loading the page, there is an animation of the dots representing Projects transitioning to their positions.
- When the user clicks on a button for Layout, it rearranges the dots according to Project Topic or Year.
- When the user clicks on a button for Filter, it shows all Projects, Major Projects or Minor Projects.
- When the user types into search (try "Picnic"), Projects matching the search term is highlighted.
- When the user hovers over a dot, the Project Type and Title appear. The user can then click for more information, like a video or description (try looking at Picnic Panic for a video).

Part 3: External Tool

- D3.js
- It is a JavaScript library that helps to bind HTML elements with data, such that it is useful to perform visualizations with that data by binding HTML attributes and animations with that data. It seemed like a popular way to perform this sorts of visualizations on the web with lots of examples, and the tutorial I was following used D3 as well.
- D3 was used to map the data from a JSON file describing the project, their details and their links to the dots on the canvas, and to also draw the layout for these dots that would show their relationships. The HTML elements also call on D3 functions to change the display according to the controls selected.
- D3 was a good way to program this sort of interactivity without hardcoding every single element, creating a system that would take in a document allows me to easily update this in the future.

Part 4: Iterations

- I decided to shift away from the eclipse-like visualization I had in my mockup as it wasn't as effective in highlighting individual projects as elements of the visualization.
- I found a tutorial that achieve something similar to what I had in mind. In this new design, the elements now takes a more atomic approach that builds towards a larger whole.
- I changed parts of the tutorial to fit the context and display of projects, and the details within them.

Part 5: Challenges

- First of all, D3 was a pretty new framework, and I had to understand it's system, especially the

enter and exit methods, to learn how the tutorial actually utilized it. CoffeeScript was also used, which was a mix of JavaScript in a python-like syntax which was new to me.

- I was unable to get the full interactive display responsive because it used a pre-determined height and width to determine where to draw the elements in the SVG representation of the projects. Perhaps with more time I would redraw the visualization with every resize of the window. The other controls are responsive, however.
- Lastly, identifying the right places in the code to change to fit a portfolio display too a little time but worked out in the end.

References:

https://flowingdata.com/2012/08/02/how-to-make-an-interactive-network-visualization/