Assignment 3 - Animated Transitions

Animation can make a visualization look polished, particularly when transitioning between views. Excessive animation, for example using animation to show timevarying data, can cause change blindness making users miss important findings in data.

In this assignment, you will be implementing animated transitions in between different *three* visualization layouts. I will be providing a dataset, but you are free to choose which three visual metaphors you use.

Note that, for the purposes of this assignment, you may use known "bad" visualization types like donut charts, pie charts, bubble charts, and so on. You may not use bad animated transitions, however.

You can either have the animations cycle as the visualization is loaded, or add buttons to allow people to switch between them.

Keep in mind that there are many, many different ways to transition from one visualization to another. Some are ineffective: for example having bars that disappear before showing pie slices. You should design your transitions to help the user understand how the data is being transformed. Designing for understanding also means your transitions should not be too fast that users can't track what's going on, or so slow and/or complicated that users lose interest.

For this assignment you should write everything from scratch. You may *reference* demo programs from books or the web, but since many animated transition demos already exist on the web, it will benefit you to create this one yourself.

Teams

This is the first assignment where you can have choose to have (1) teammate. If you choose to do so, give a one sentence description of what each teammate contributed to the most. I expect to see plenty of commits from both teammates.

Resources

The dataset is included in the repo. It's simple and will work for many basic visualization types.

See the original "D3 Show Reel" from October 2011. You should not use this code (it's outdated), but the general idea is the same.

D3 provides many "Layout" functions to help you transform visual elements, see the Layouts D3 Reference. If you use these, you may need to spend a few hours working through examples to understand how they work.

The main d3 api and especially the d3 transitions page will be useful.

Don't forget to run a local webserver when you're coding and debugging. See this ebook if you're stuck.

Requirements

- 0. Your code should be forked from the GitHub repo and linked using GitHub pages.
- 1. Your project should use d3 to build a visualization of the dataset.
- 2. Your writeup (readme.md in the repo) should contain the following:
- Working link to the visualization hosted on gh-pages.
- Concise description and screenshots of your three "stages".
- Description of the technical achievements you attempted with this visualization.
- Description of the design achievements you attempted with this visualization.

GitHub Details

- Fork the GitHub Repository for Assignment 3. You now have a copy associated with your username.
- Make changes to index.html to fulfill the project requirements.
- Make sure your "master" branch matches your "gh-pages" branch. See the GitHub Guides referenced above if you need help.
- Edit the README.md with a link to your gh-pages site: http://YourUsernameGoesHere.github.io/03-Animation/index.html
- To submit, make a Pull Request on the original repository.

Grading

Grades on a 120 point scale 100 points will be graded for functionality: the program does what the assignment requests. 20 points will be based on documentation in the README.

I will use Google Chrome to view submissions. Be sure to test your code there.

Total-120

(0 will be assigned if the code won't run.)

Functionality - 100

100 – Dataset loads and visualization works with no obvious bugs. Animations are appropriately designed (i.e. no disappearing elements, an individual element is easy to track, speed is sensible).

Writeup - 20

- Description of technical achievements 10 Description of design achievements