CPSC 447 23Sep: Project Description

Over the rest of the term you will work in teams of four towards creating an interactive visualization project using JavaScript & D3. *Read this document carefully!*

You'll form a team, submit four main milestones, and give a live demo of your final visualization:

- **M0 Team formation** *(Due: Sep 27, Qualtrics)*
- **M1 Abstractions: Data and task abstractions** *(Due: Oct 4, Gradescope)*
- **M2 Design: A detailed project plan** *(Due: Oct 25, Gradescope)*
- **M3 WIP: A partial implementation of your visualization based upon your proposal** *(Due: Nov 22, Gradescope)*
- **M4 Final project and write-up** *(Due: Dec 6, Gradescope)*
- **Live demo** (approx. 5-10 minutes per group, face to face by Zoom; date TBA, after final submission)

General Project Requirements

Implement a visual analysis app or interactive data-driven story (narrative visualization) in D3 based on the following requirements:

Mandatory

- Data
    - Choose a dataset wisely. It must be complex and large enough to complete a project using it meeting the requirements below.
    - You must use publicly available dataset(s).
    - Your data must be either pushed to your repository or clearly connected via an API.
    - You may want to pre-process your datasets, either to derive data or to make a mashup of data gathered separately.
- Tasks
    - At least three distinct tasks pertaining to at least five distinct attributes.
    - Sufficient overall complexity that visualizing multiple attributes using multiple views is necessary.
    - All tasks must be plausible and must be addressable by the dataset(s) you are using.
- Design

    - At least three distinct views.
    - At least one view should be innovative, namely something beyond these standard chart types: bar chart, scatterplot, line chart, simple area chart, pie chart, donut chart, choropleth, dot density map, circle-pack (bubble) chart, tiled heatmap. You could design and implement a unique view that is appropriate to support one or more of your intended tasks. You could also augment a simple standard chart type as listed above with significant additional visual encoding or interaction functionality.
    - At least two views should be bidirectionally linked, and those should be simultaneously visible side by side onscreeen.
    - At least one view must support mouse-based interaction (eg. hover, click, drag, tooltip) in addition to the bidirectional linking interaction.
    - At least one UI widget (eg. dropdown, radio button, range slider, calendar).
    - Most projects will be a good fit for dashboard style (where all or nearly all views are visible at once), but you are also allowed to use a scrollytelling style.
    - Layout should use screen space wisely. Layouts that are too sparse force the user to scroll unnecessarily and fall short with supporting bidirectional linking. Layouts that are too dense are cluttered.
    - Interface should include enough headings, legends, axis labels, and any other explanatory text needed to make it self documenting.
    - You must provide detailed rationale about how all aspects of your design address your identified scenario.

- Implementation

    - Your web application should be stand-alone and have an `index.html` file as entry point, similar to the programming exercises. You can either store static (*optionally preprocessed*) datasets in your git repo or access live data through an API.
    - Your code must be readable, reasonably structured, and conform to a consistent style. You should follow the structure and code style of the assignments and tutorials.
    - Your git commits must be reasonably sized and include informative commit messages.

Extra Credit (+1% to +5%)

- High level of polish in final interface.
- Going above and beyond with functionality.
- Outstanding and thoughtful explanation in writeup.
- Demonstrates true mastery of foundations concepts.
- Overcoming a challenging problem.

Not Allowed

- We do not allow custom backends (Node.js, Python, etc) and database systems, such as Postgres or MySQL, although they could facilitate more powerful applications. In this course project, you should put your efforts into the front-end development (JavaScript, D3, HTML, CSS).
- You need to implement visualizations in D3. You can use CSS frameworks, such as Bootstrap or Materialize, and include external libraries (jQuery, leaflet.js, moment.js, ...). Layers such as NVD3, Vega-lite, Highcharts, etc. are not allowed. Using external libraries such as React.js for the structure/layout of your system is allowed, but you cannot use these libraries to generate any part of your visualizations. TAs can not help you with external libraries that are not used in the class.

More Details

Strong design work will make the implementation easier!

It is reasonable and encouraged to change your design as you go, if you learn that your initial ideas would benefit from refining. You should provide rationale for your changes, just as you should provide rationale for all of your other design choices.

Talk to a TA in office hours to discuss your idea if you have any uncertainty, particularly regarding your choice of components.

The goal of the project is to synthesize the material from both the programming exercises and the foundational exercises. We will evaluate projects using the following high-level criteria:

- Visual design: Effectiveness of visualizations and interactions, which should link back to usage scenarios, tasks, and intended audience.
- Level of technical difficulty and quality of implementation.
- Addresses the goals and requirements.
- Quality and clarity of your writing.
- Teamwork.

You will need to submit peer evaluations of the contributions of all team members (your teammates and yourself) as part of all four milestones. We may assign different grades to each group member based on these assessments.

Team Formation

Form a team by filling out the Project Team Formation Qualtrics Survey by Sep 27, 6pm.

- Use the pinned Piazza post to find teammates.
- Work closely in teams of **four**! We do not allow teams with more or less than 4 people. If you have a very unusual circumstance with a strong need for a team of 3, we will consider your request, since there will be

two such teams. Talk to a TA or instructor about it in office hours, and if your request is granted state who approved it in your proposal writeup.

- We will create empty GitHub team repositories for teams.
- Your team will be responsible for downloading the zip file with skeleton code/files.