

TP3 (due Wed Dec 9 at 4:30pm)

Design Proposal (+Changes) [25 pts]

Project Proposal [15 pts]

Write up a proposal file (in the file proposal.txt, or .docx, or .pdf) which should include the following components

- **TP3 Update*****

There are minor changes to the app's design, including many well thought out small enhancements to make it more user friendly (handle lots of base cases). The scatterplot now has zoom + - buttons. The twitter page leads to live data- current hashtags in the form of the twitter logo word cloud (gradient colored).

- **TP2 Update*****

There are some minor changes to the app's user interface, with the user selecting all visualization and graph modes on the options page and not on the graph itself. This includes the variables and type of graph, with the exception of the visualization rate on the time animation bar group graph only.

- **Project Description** [2.5 pts]:

Graphos is a general purpose data visualizer that allows the user to input a csv file (or choose from the predefined datasets) and display the data through different graphical/visual options (scatter, bar, histogram, time animation).

- **Competitive Analysis** [2.5 pts]:

The most similar websites and apps to the project I am developing include Excel, Demos, Granafa, and Tableau. My project differs from others because it includes the option for time-series animated data and will support live-streaming data from Twitter as an added bonus. Similar modules to my project include plotly and Matplotlib, which are both programmatic libraries with various visualization features.

- **Structural Plan** [2.5 pts]:

My final project will be split into 5 main components. The first is the main app entrypoint that uses the cmu_graphics framework and imports the rest of the files to do the heavy lifting. The second component is the data manager, where the predefined or user-specified csv file will be processed into in-memory data structures and transformed for later use. The third component is the app's home screen view generator, which will have the option of letting the user input a file or play around with our built-in data sets; in there, we will also add the help/info screen. The fourth component will be our VisualizationOptions class, which defines the various attributes supported by the application and will drive how the visualization is generated. The fifth and final component handles generating the

graph view with various modes (scatter, bar, histogram, time-series, etc) based on the VisualizationOptions defined previously.

- **Algorithmic Plan** [2.5 pts]:

Visually, the trickiest part is maintaining the visual state of our application given the numerous options the app supports. The view generation should handle all visual combinations, but we will combat this problem with a modular approach by encapsulating the option combinations in a class so we can enumerate the various visualization modes and handle them appropriately in distinct methods.

Internally, the trickiest part is handling the large data volumes. We need efficient in-memory representation of the data to support the various use cases of our options for fast retrieval. This in-memory representation needs to support being updated efficiently in real-time.

- **Timeline Plan** [2.5 pts]:

TP1 - Have the project foundation set up with the 5 components defined in the structural plan. Have the home screen, options selection screen, and one graphing visualization mode implemented.

TP2 - Add the other graphing modes and finish features discussed in the storyboard for MVP.

TP3 - Add support for Twitter live-streaming of data as another visualization mode.

- **Version Control Plan** [1.5 pts]:



I will use GitHub as version control for the project.

- **Module List** [1 pts]:

I'm only using built-in modules for now, but plan to use Tweepy as a Twitter API client after MVP.

- **Storyboard** [10 pts]

[Attached]

- **Preliminary Code** [75 pts]

[Attached]