# CIS 4930/6930-002: Data Visualization (Spring 2019)

## Project 8: Force Directed Graph Layout (Grad Only)

### 1 Objectives

In this assignment you look to address scalability concerns with your previous sketches by using aggregation techniques. Again, take care to use good software engineering practices.

### 2 Ground Rules

This assignment is intended to be done alone. You may ask others for help with figuring out how details of Processing. However, code must be your own (MOSS will be used!). Furthermore, NO additional libraries (such as giCentre utilities) may be used. Doing so will result in a 0 for those sketches.

### 3 Assignment Instructions

- Download the skeleton code and data provided in Canvas→Files. This dataset contains character coappearence in Victor Hugo's Les Misérables, compiled by Donald Knuth. Node in the graph are characters. Edge in the graph signify characters appearing in the same chapter of the novel.

- Fill in the portion of the code for loading the data, noting that it is in JSON format. JSON is a more flexible storage format than CSV, but it is also more complicated to parse and load. Please see Processing reference on parsing JSON Objects[1] and JSON Arrays[2].

- Using the data, fill in the code sections for creating a force-directed graph layout[3]. You will only need to calculate forces, draw the graph, and code interactions.

- (50% extra credit) Create a distance matrix for the points using Shortest Path Distance[4]. Now draw the graph by projecting the points using Multidimensional Scaling (MDS)[5].

- Add any additional linking or interactions that you think will make your dashboard more useful. Your selection and their implementation will have an impact on your grade.

- Modify your sketches such that they use additional visual channels to encoding additional variables. Consider using color, size, shape, depth, etc. Your selection and their implementation will have an impact on your grade.

- Add embellishments of your choice. These can include but are not limited to: axis lines, labels, and tick marks. Your selection and their implementation will have an impact on your grade.

- Make sure your visualizations are robust by designing them to support other data (number of elements or value range) and by designing them to support any size of canvas.

[1] https://processing.org/reference/parseJSONObject_.html

[2] https://processing.org/reference/parseJSONArray_.html

[3] See section 12.2 of https://cs.brown.edu/~rt/gdhandbook/chapters/force-directed.pdf and the extra powerpoint slides provided on canvas

[4] https://en.wikipedia.org/wiki/Dijkstra's_algorithm

[5] You can use the MDSJ library for this, http://algo.uni-konstanz.de/software/mdsj/

## 4 Submission

All of your work should be done in your git repository in the directory named **project8**. If put it anywhere else, our script will fail (and so will you). Make sure things are labeled well, so that your peers can find them.

As you work on the files make sure you frequently add the files to the repository (i.e. *git add*), commit the changes (i.e. *git commit*), and push changes to the remote server (i.e. *git push*). If you fail to do this, we won't get your files.

## 5 Grading and Feedback

- Your grade will be combination of objective measures (based on the assignment instructions) and subjective grading by the instructor.
- Breakdown
  - Force directed layout - 7 points
  - MDS layout - 5 points
  - Additional interaction, embellishment, and additional Visual Channels - 1.5 points
    - 0.5 points for none used
    - 1.0 point for a few
    - 1.5 points for many
  - Code Professionalism - 1.5 points
    - 0.5 no comments, no classes, "hard coded" values
    - 1.0 minimally commented, few "hard coded" values
    - 1.5 commented, properly used classes, few "hard coded" values