Visualizing network relationships

**Project Overview**

**Problem**
The vast majority of network visualizations use connecting lines that communicate only link value information. Nodes are either connected to each other or not. Information about the nature of the connection—its strength, frequency, or direction—is either not available or not represented in these kinds of visualizations.

Extensive research has contributed incremental improvements to methods for drawing single graphs, rendering more efficient renderings, and intelligently clustering nodes and edges that reduce visual clutter. Little work, however, has addressed visualizing directed graphs with multiple edges, the focus of this project.

**Motivation**
Working with more robust edge data provides the opportunity for each edge to convey not just a connection but the nature of that connection. Since directed graphs incorporate the directionality of each connection, there are opportunities to visualize the relationships between nodes, the two-way relationships, in other words. Also, when working with multiple edges, each edge can be assigned a quantitative value, which enables a measure of time, a party ranking, or any other relative value. Potential data sets for directed graphs with multiple edges include phone records, social networks, economic trade data, website links, and network traffic.

**Approach**
I began the project with the phone records of two individuals (the primary nodes, including the length of each phone call) (the edge values). I often experimented with ways to encode the edges with multiple values for each node-to-node relationship, starting with simply the number of phone calls and the direction of the call (e.g., from A to B or from B to A). Then, I implemented methods to more efficiently convey the directionality of calls as well as a scale for each call so that the length of each phone call (the edge’s quantitative value) was represented.

Since my goal was to build a tool that could be used with a wide range of data sets, I used a simplified data model. The application takes an input of a plain text file with each line representing one edge in the format:

```
fromNodeID toNodeId quantitativeValue
```

Recognizing that no single visualization could be ideal for all possible data sets, I incorporated some basic interaction tools to enable users to filter the data, modify visual properties, and focus on what is most relevant to them. PDG expert functionality allowed users to easily save a static visualization for later review.

**Results**
"Relationship Visualizer" is an interactive application built with Processing, that encourages users to layer their own graph data and manipulate the visualization, modifying parameters until arriving at a rendering appropriate for the user’s needs.

**Future Work**
- Incorporate additional data dimensions, such as date/time values for edges, and design methods of accommodating each additional layer of data into the multiscale display.
- Implement alternate layout methods.
- Incorporate mouse roll-over behavior to reveal information on selected nodes and/or connected nodes.
- Further automating scaling so that min/max/quantitative edge values will be scaled appropriately for the display resolution.
- Build a more intuitive user interface, less reliant on keyboard commands.
- Make more parameters: angle, color, the weight, quantity, and modification.
- Solicit user feedback and incorporate additional visualization methods.