Finding the Way

Background Story

As a pioneer in the sharing economy, Uber is one of the most influential companies in the world. With 50 million riders and 7 million drivers, how does Uber efficiently match its drivers with riders?

Introducing the Concept

Dijkstra's Algorithm -- an algorithm for finding the shortest paths between nodes in a graph (from Wikipedia).

Interactive Visualization Implementation:

(Start with a simple undirected graph)

Display a small/simple graph with nodes marked with numbers and edges marked with tentative distance values.

Steps:

1. Identify the initial node and the destination node
   Users are able to select the initial node and the destination node by themselves

2. Set the initial node as current and the distance value to be zero. Then consider all of the current node's unvisited neighbor nodes and calculate the tentative distance value for each of the unvisited neighbor nodes. Then, compare the calculated distance values and assign the smaller one.
   Users are able to click on different paths (edges). If the edge the user clicked has the smaller calculated distance value, the edge and the current node will change color from black to green. Now the current node and the edge are marked as visited, and the visited node and edge will never be checked again.

3. Step 2 will be repeated/iterated until the destination node has been marked visited or if the smallest tentative distance in the unvisited set is infinity.

4. The algorithm is finished and the users will receive two results: (1) a green path which represents the shortest path between the initial node and the destination node; (2) no path.

(Do a more complicated directed graph)

- Similar to the simple undirected graph

- Maybe apply the Uber background story to the visualization design (add cars, roads, and riders)
- Maybe add a new interactive visualization technique -- once the user picks an initial node, the visualization design will run through the algorithm by itself and list all the shortest path results of different destination nodes to the user.