How to & Interactivity:

To start open the “lauch.html” file. Upon start the graph will begin to transform and layout. Once this operation is complete the GUI can be utilized. In the GUI are two folders with algorithms, a slider to select number of nodes, and a button to clear changes to the graph.

Changing Number of Nodes:

1. Allow graph to settle (very important or else there will be a lot of visual artifacts)
2. In the GUI slide the slider to desired value
3. Release slider
4. View the output of a new graph with specified node count

Running an Algorithm:

5. Allow graph to settle (very important or else there will be a lot of visual artifacts)
6. In the GUI select a folder to drop down
7. Click on DFS, BFS, or Dijkstra
8. View the output
9. Click the Clear Button to erase visual changes to the graph and output
    a. When using Dijkstra click the clear button twice

Project Documentation:

The goal of the project is to make a visualizer for some common graph tracing algorithms in a 3D space using ThreeJS. Such as, Depth First Search, Breadth First Search, and Dijkstra’s. The inclusion of a simple GUI to select the algorithm or reorganize the graph is also planned.

The demonstration would consist of running the aforementioned algorithms, with some visual indicator of their progression through the graph.

Resources:

- ThreeJS
- Public web articles on Force Directed Graphs
- Public web articles on Graph Tracing Algorithms
- DAT GUI by google
- WebGL compatible browser

Outline:

1. Random Generation of Nodes and Links
2. Process Nodes for Layout w/ Force Directed Graph Drawing
3. GUI for Algorithm Selection/Graph Regeneration
4. Graph Tracing Algorithm Selected and Processing
5. Color Links/ Highlight Nodes as Algorithm Executes
6. Clear Trace w/ GUI Clear Button

Classes and Data Structures:

- Queue: Used to implement Breadth First Search.
- Stack: Used to implement Depth First Search w/out recursion.
- Priority Queue: Used to implement Dijkstra.
- Graph: Facilitates creation of graph, such as nodes & links. As well as, initializing the graph in 3D space.

Algorithms:

- Eades: Used to do basic force layout of the graph. Consists of treating links like spring forces and unconnected nodes like electrically repulsive forces.
- DFS: Used to traverse graph.
- BFS: Used to traverse graph.
- Dijkstra: Used to find shortest path from origin to every other node.