

Interactive Vizualization

Assigment 2

Ian Ooi
Max Espinoza

September 8, 2014

1 Discussion of Results

The easiest graph to implement was a general undirected graph. VTK uses layout strategies to control how those nodes where displayed. Some are familiar in that they include strategies such as forced-directed, clustering, and randomly assigned 2d layout. While these were great to start to visualizes inputs quickly, it actually provided difficult when we wanted to implement our own strategy or modify existing ones. Visualizing a clique, disconnected graph, non planar graphs were easy. However trying to implement trees, planar graphs, and bipartite graphs proved a challenge because we had to dig deeply into implementation of VTK to control where vertices's were being placed. I couldn't have done better by hand, so I can't complain, I do wish that the tool offered more control or documentation on the underlying data structures and how to use them to generate new 2D layouts strategies.

2 Review Of VTK

VTK has a very poor community presence and only moderate presence online, which makes getting started and continuing with the tool very difficult. It has a fairly high cost of entry, requiring either a package to be present in your system's package manager, or for you to be comfortable building from source. Building from source was not that difficult for the C++ libraries, but it is extremely difficult to get other bindings to compile and install properly as there seems to be an issue in the cmake files, which require knowledge of cmake. When such issues arise, the lack of online presence severely limits the available solutions. This overall tends to suggest another tool should be used if possible.

As a whole, the toolkit seems to be an industrial level solution but without ease to its power. A lot of the features that are available to make more involved, complex visualizations get closer and closer to OpenGL, making the appeal of the tool limited. There is more functionality available for 3D datasets, making it a bit easier to get something on the screen than it would be in OpenGL, but for 2D the functionality provided using the automatic precludes manual input. Customizations were also difficult to achieve, such as changing the size of a displayed vertex in the graph. Overall this is not a tool for easily creating 2D visualizations, and certainly not for those who expect tutorials or community help for figuring out issues.

3 Results

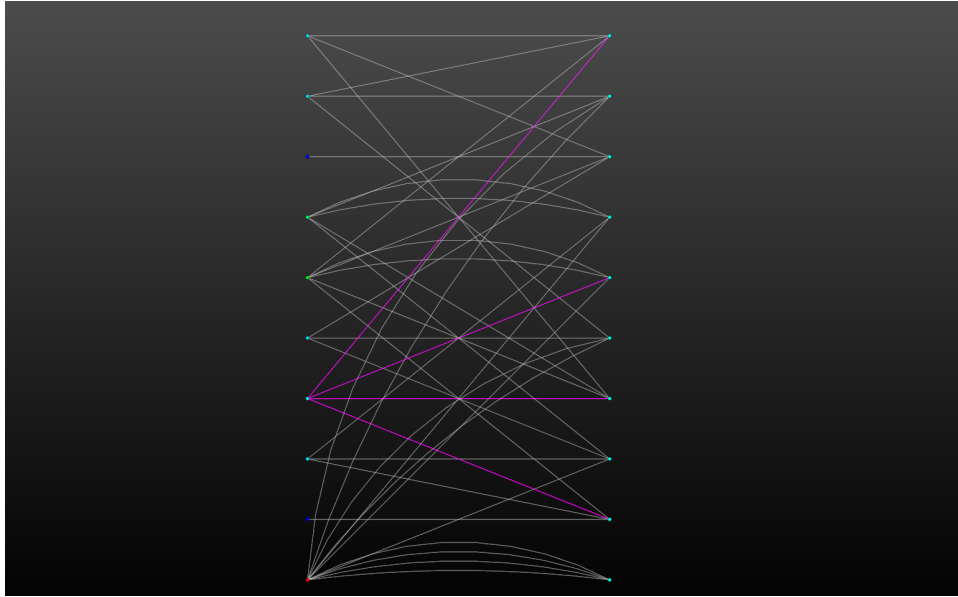


Figure 1: A bipartite graph.

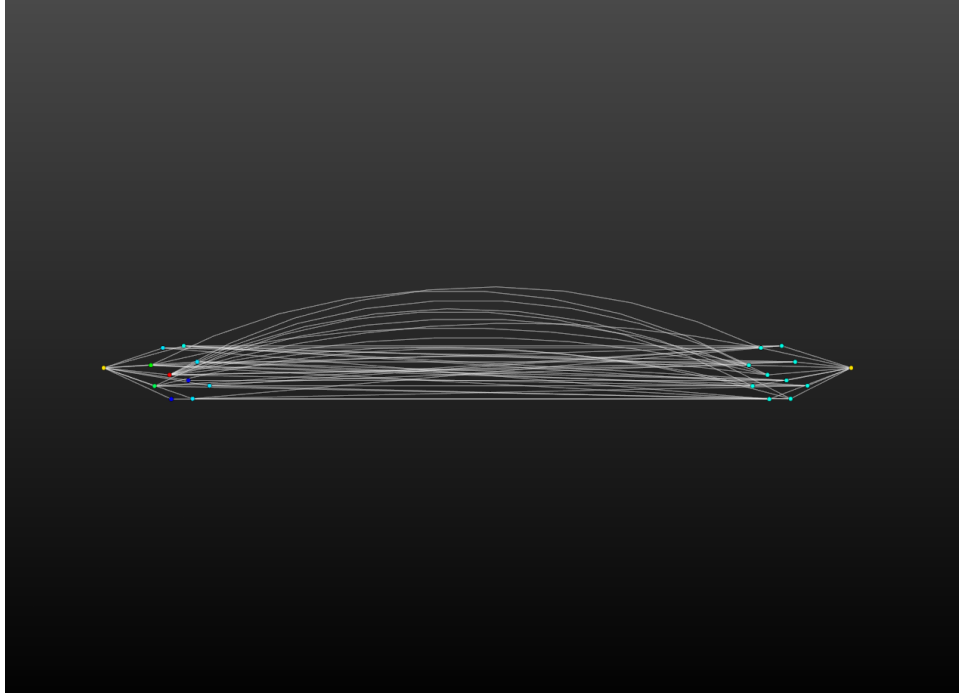


Figure 2: A bipartite graph drawn with a clustering layout.

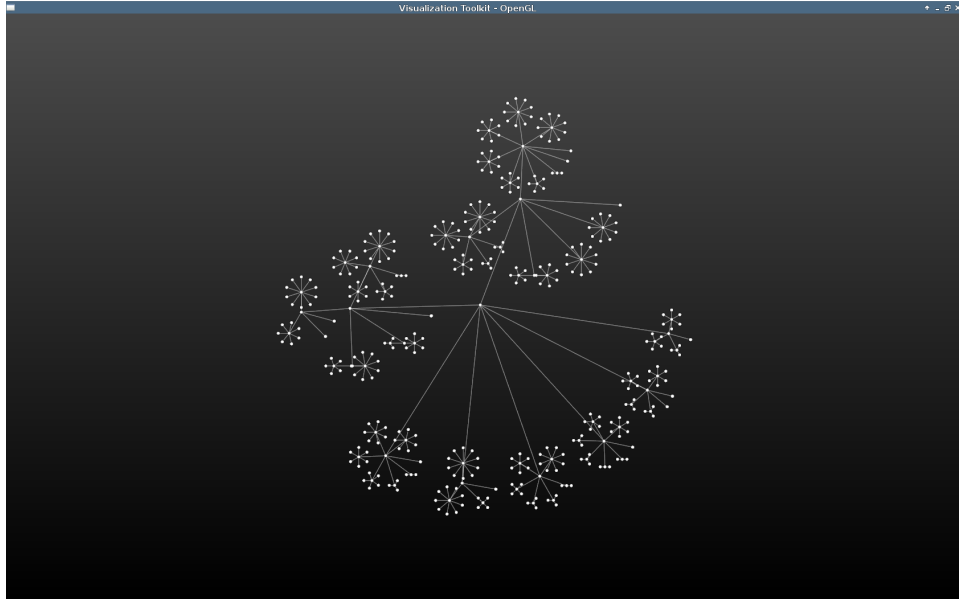


Figure 3: A tree drawn with the cosmic tree layout.

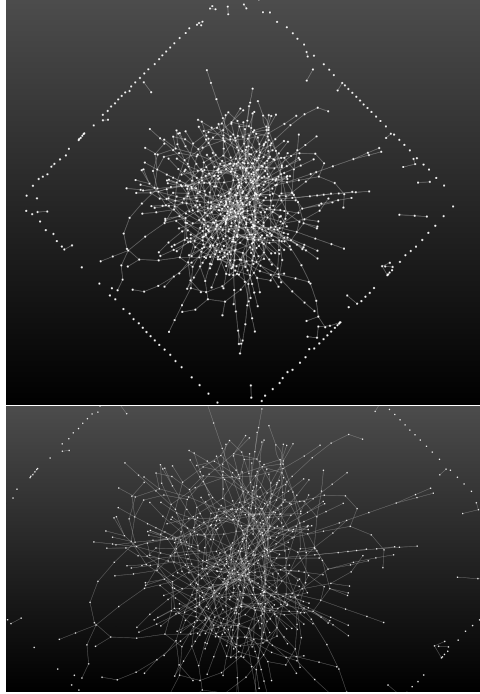


Figure 4: A disconnected graph.

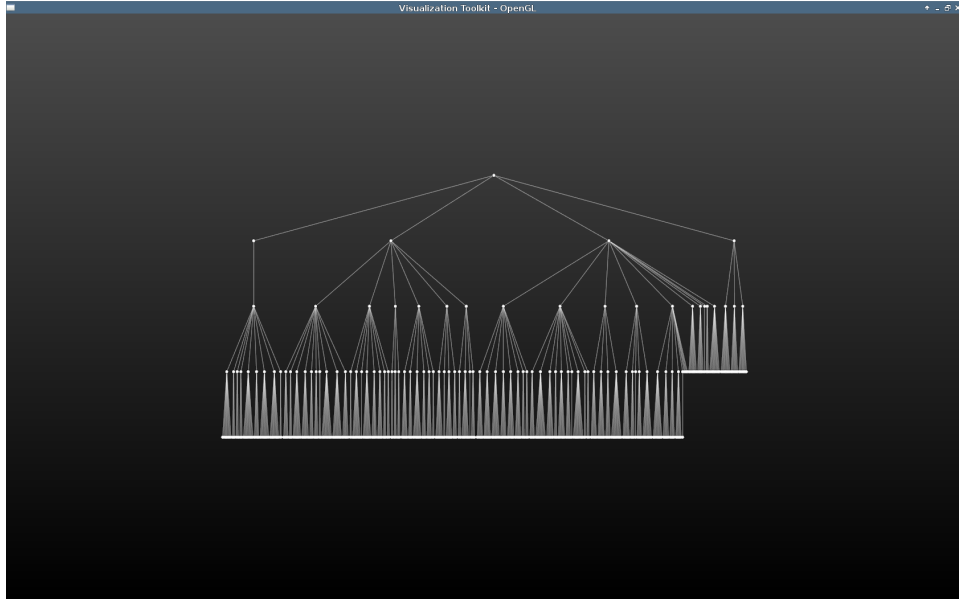


Figure 5: A tree.

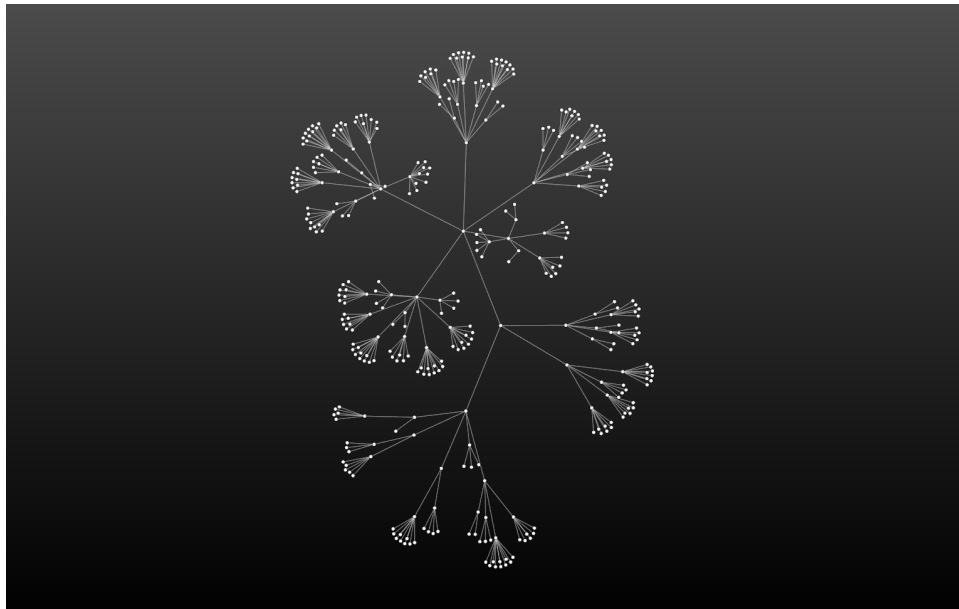


Figure 6: An alternate tree layout.

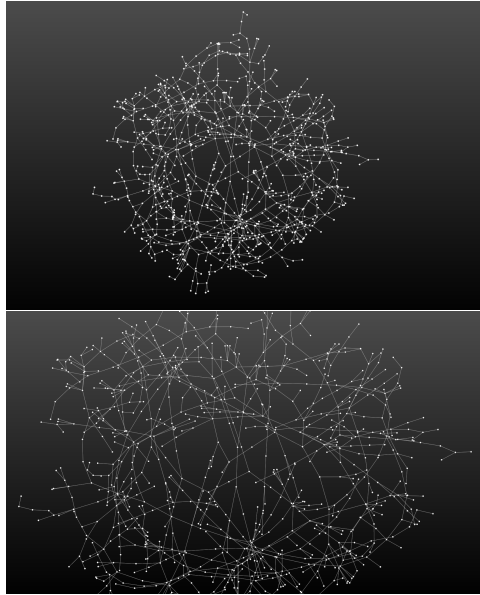


Figure 7: A non planar graph.

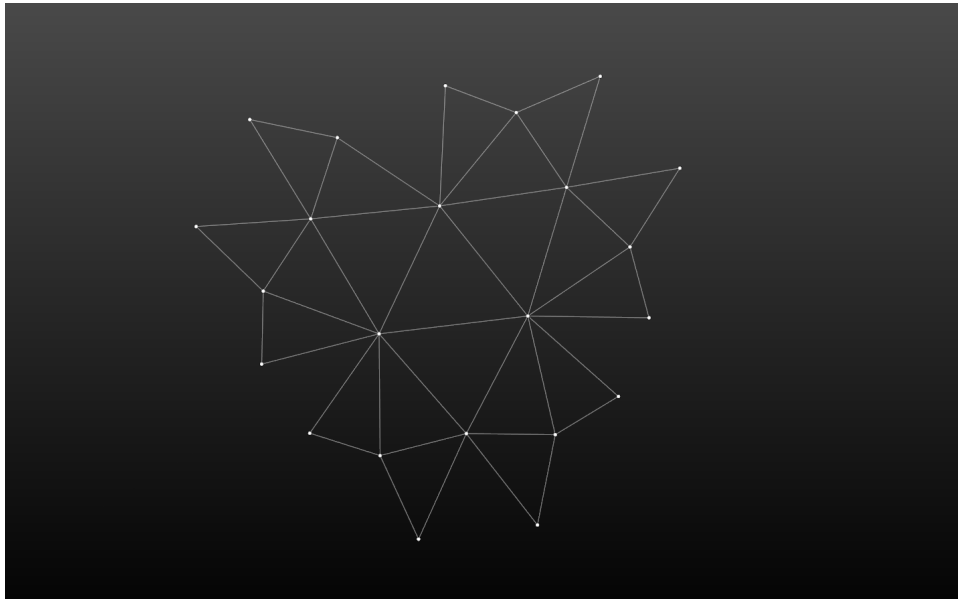


Figure 8: A planar graph.