COSC 1430 FINAL PROJECT – OPTION 4

2D Graph Project (Send emails to gxf.xisha@gmail.com, if you have any questions)

This project exposes you to the world of Java 2D and enables you to create and edit a 2D graph. Besides swing usage, this project enables you to be familiar with the concepts: 2D canvas/paint, mouse event, point, line, and graph.

The following provides a simple tutorial to help you finish this project step by step.

First, create a Jframe object that contains a canvas, RadioButton/CheckBox and a panel (Figure 1). The panel shown in the lower right of Figure 1 contains two lists and one button.

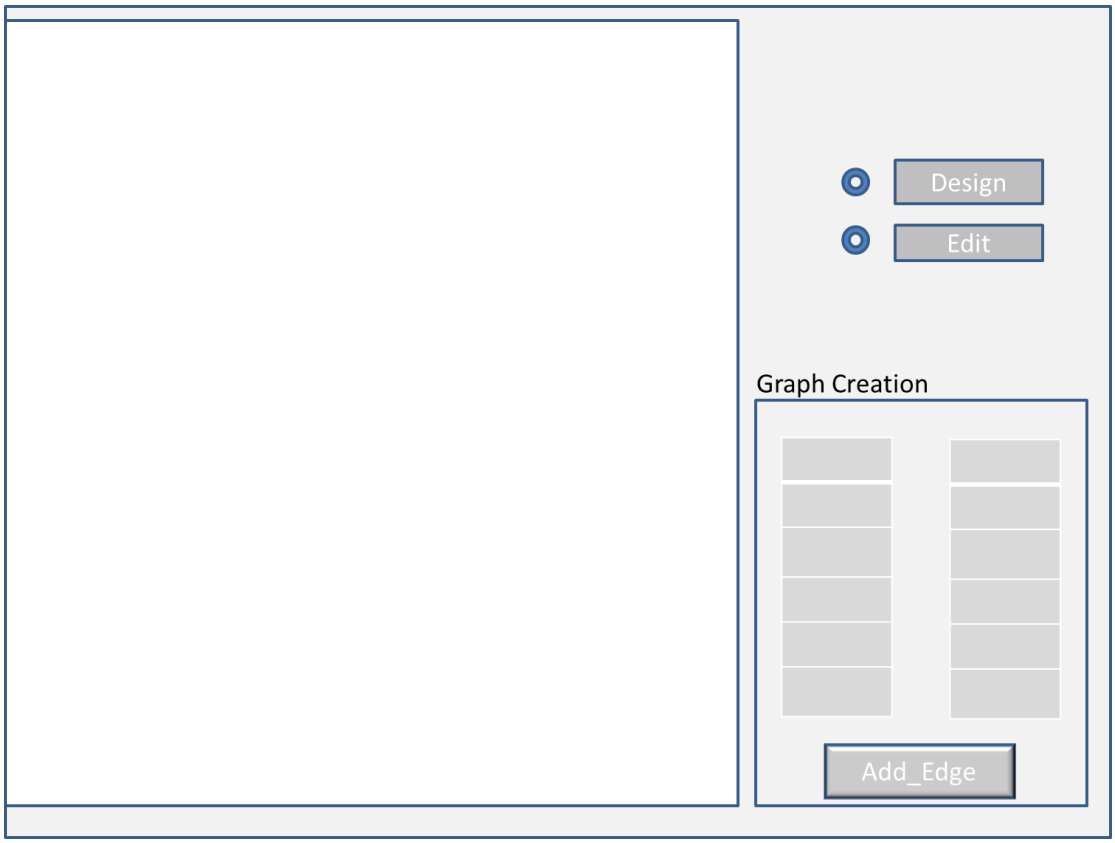

Figure 1. The blank area is the canvas. You can create canvas class extended from Jframe or Jpanel. The two lists at the right-bottom should not be seen unless there are content added, here, just for illustration.

Second, create a number of points on the canvas by moving the cursor to the canvas area and left-clicking the mouse. The two lists will simultaneously show the references to the points. At this moment, we assume the two lists show the same list of nodes (Figure 2).

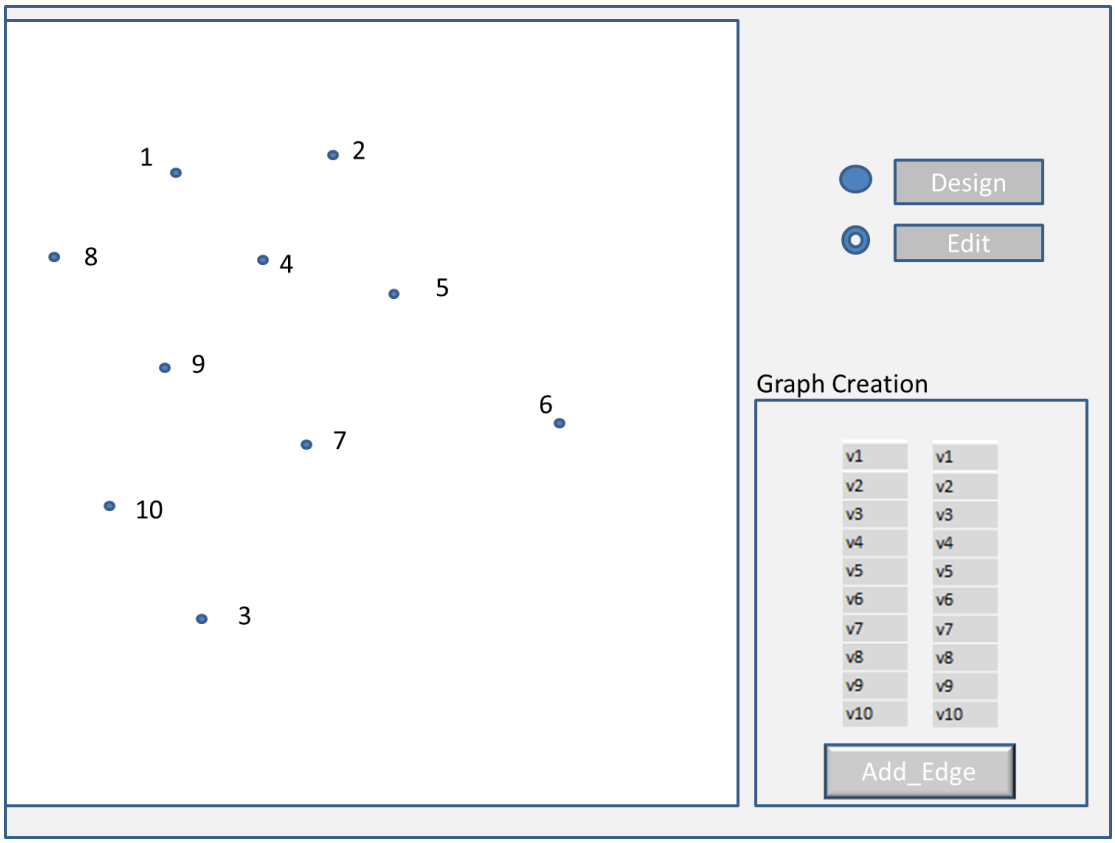

Figure 2. You are not required to show the point notations (1, ,2 ,3, etc.) in the canvas, although it's easy to implement.

Third, create edges. An edge is represented by two end points. An edge can be created by (1) selecting two points from the two lists respectively, and (2) press the *Add_Edge* button. The created edges should be shown on the canvas simultaneously (Figure 3).

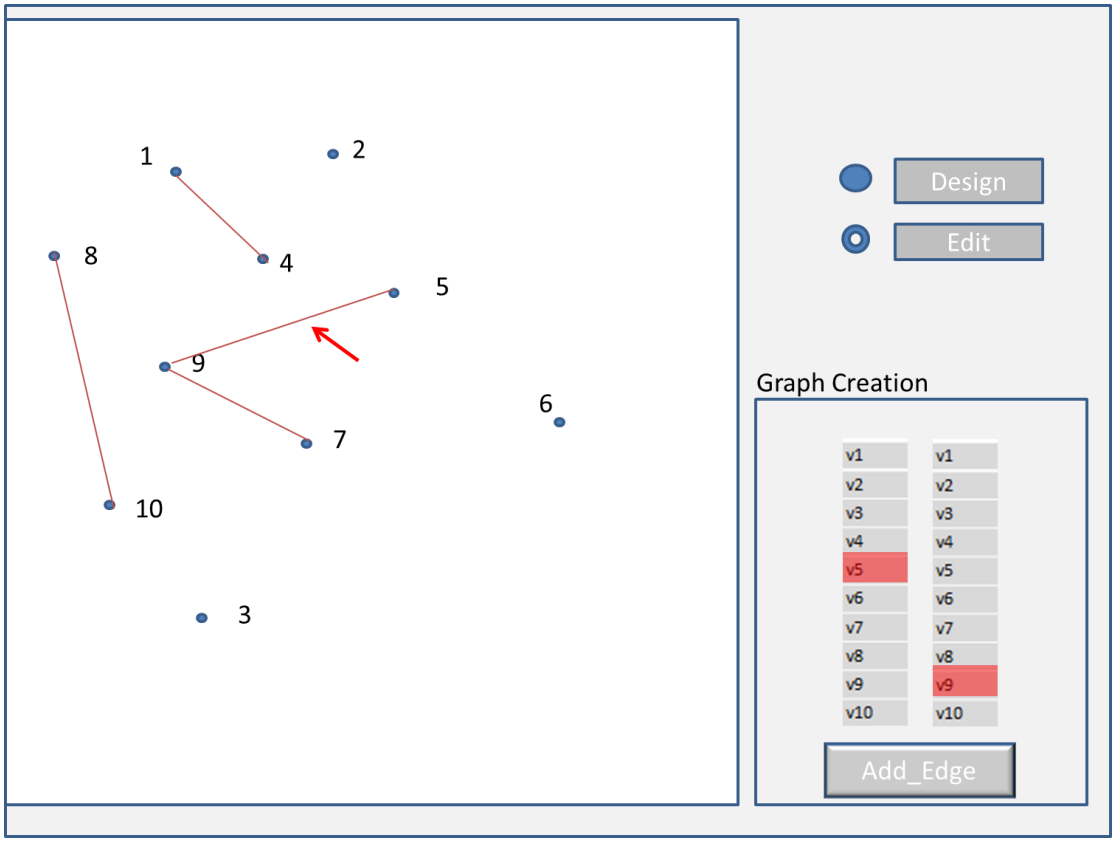

Figure 3. By (1) selecting v5 and v9 in the lists respectively and (2) pressing button Add_*Edge*, an edge connecting point 5 and point 9 is created, and the edge is simultaneously shown in the canvas.

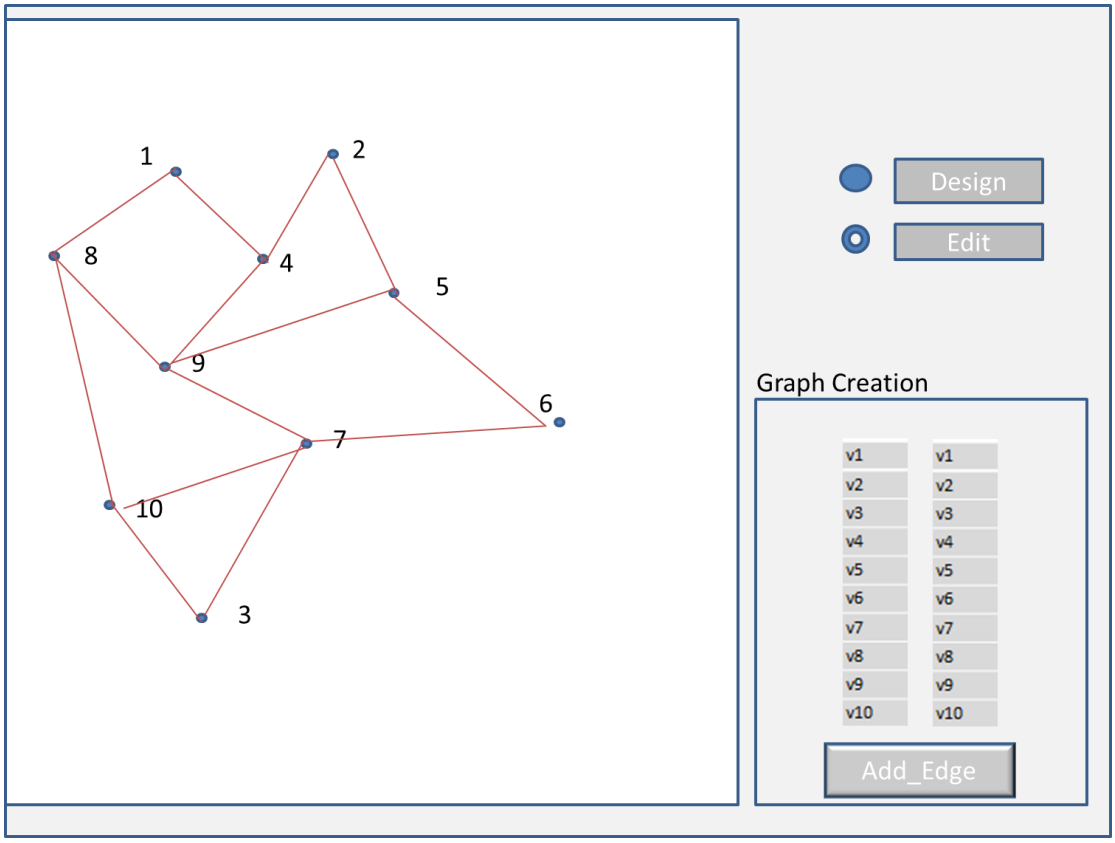

Figure 4. A finished graph.

Fourth, after finishing your graph creation, you can edit your graph by dragging any point using your mouse to any place in the canvas (Figures 4 and 5). You need to first select the radio button called "Edit" to enable this functionality. Note that, although the length, location, direction of the edges can be changed, the point connections will not be changed in this process. You may consider a functionality of deleting an existing node or edge.

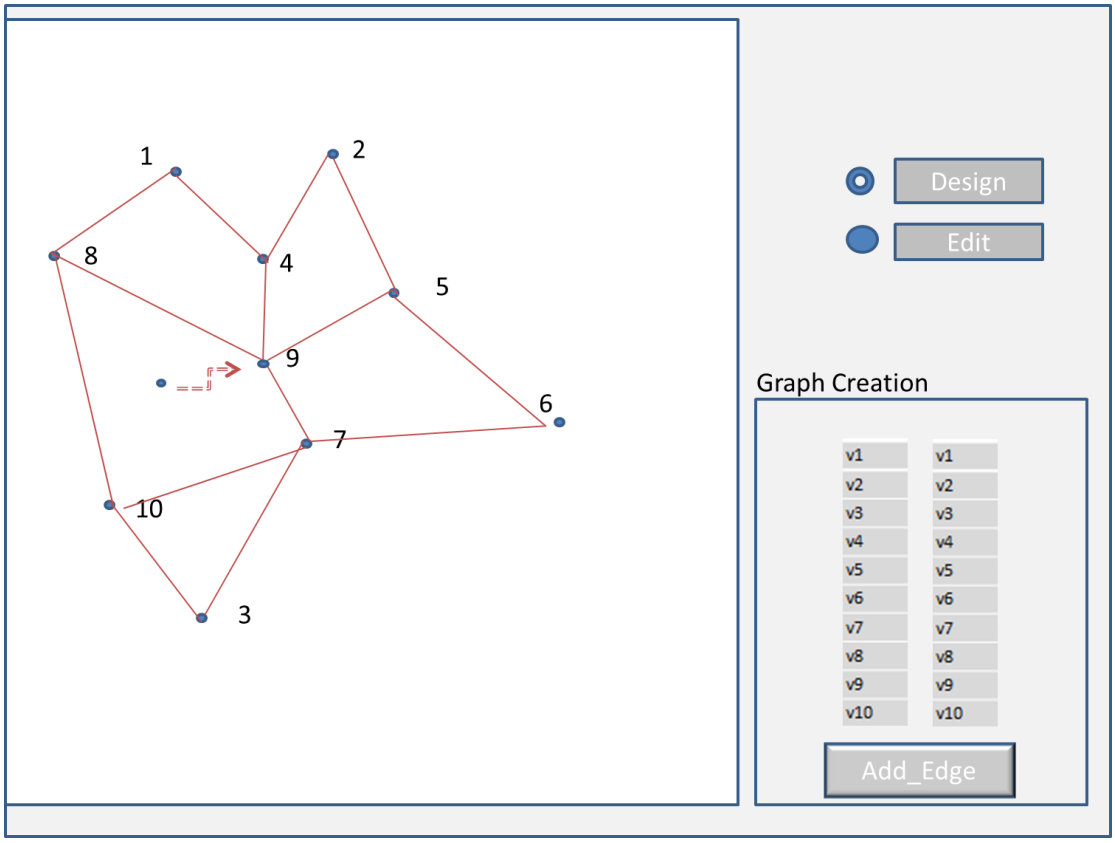

Figure 5. Point 9 has been dragged to another place, while its connections are not changed.

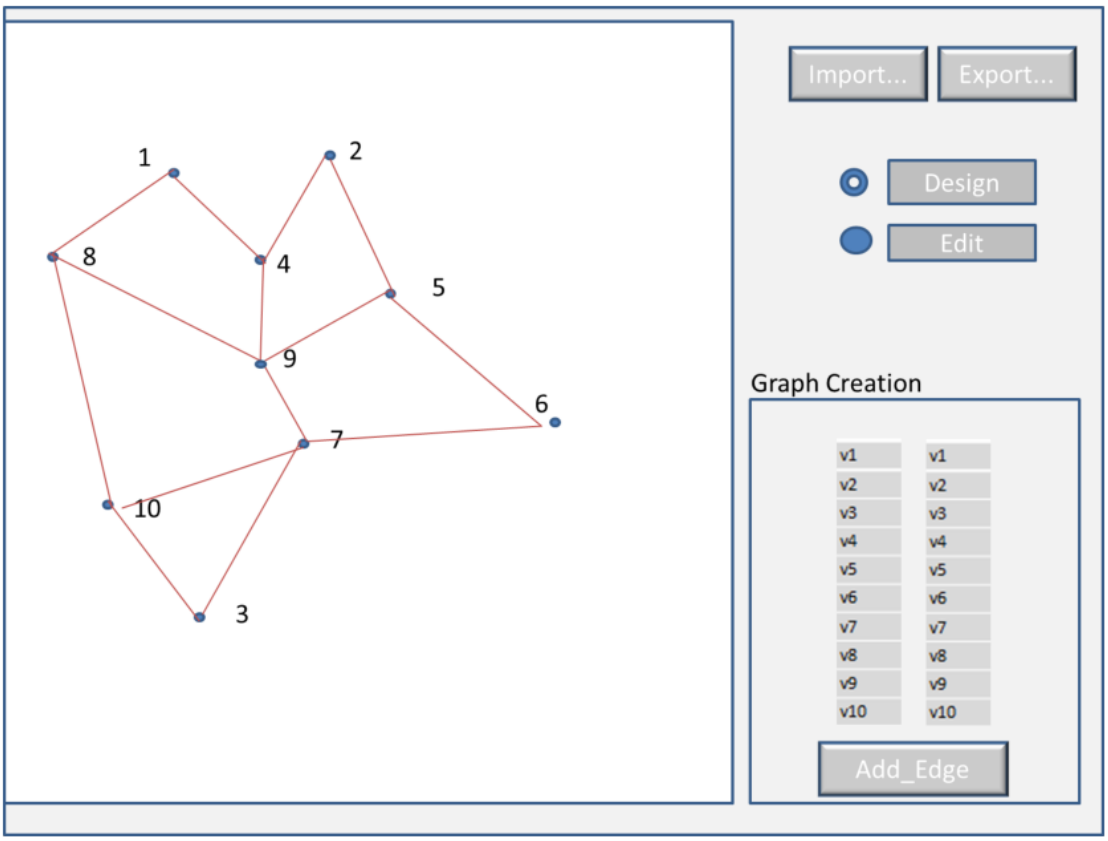

Figure 6. The graph shown in canvas is loaded from the input file by pressing button Import….

Fifth, load/save graph data from/to file. By pressing *Input* button, a file choosing-dialog should pop-up for file loading operation. By click *Output* button, a file save-dialog should pop-up for file saving operation (Figure 6). The file contains all the data of a graph. An example format is shown as follows:

“

V E ----------V is the number of points, E is the number of edges.
1 x y
2 x y
3 x y
i x y ---------- i is i_th point, x is x-coordinate, y is y-coordinate.
.
.
----------//there are totally V points.
1 a b
2 c d
3 m n
i p q ----------i is i_th edge, p is the index of start point, q is the index of end point.

----------//there are totally E edges.
”

Based on this format, the graph shown in Figures 4 and 5 can be stored as:

10 14
1 0.2 0.82
2 0.4 0.88
…
1 1 4
2 1 8
3 2 4
4 2 5
…

Important:

Please note that all term projects are open-ended questions. In other words, you are free to add any functionality and improve the user interfaces.