Hillary Ssemakula

COMP4270 Computer Graphics I

Professor Krishnan Seetharaman

**PROJECT REPORT.**

**Issues faced:**

* Trigonometry: There was a lot of trigonometry involved in scaling and rotating. For shapes like the ellipse, square and triangle there were up to 40 lines of code dedicated for each of those task s for each shape. Every shape required its own algorithm and therefore the script is about 1000 lines long.
* Object Oriented Format: To make this project work, I had to employ the object-oriented model of programming by making every shape an object. However, I was not very familiar with the java script syntax for classes and objects and hence a lot of research was poured into that which ate up development time.
* Large script File: As the script file for the program became larger, it became harder to to keep track of how the different components in the program interact, for example to view a variable or to find a function I would have to scroll through multiple lines of code.
* Algorithms: It was time consuming to come up with algorithms for selection, definition of polylines, rotation and the rest.
* IO API: Input and output for JavaScript is a pain, therefore implementing the load/save feature took more time than needed, almost 2 days.
* Styling: Being that canvas is an HTML element, the use of bootstrap on it would skew the coordinates therefore it made mouse clicks, or the reported coordinates of mouse clicks, inaccurate.
* There were limited resources online to help with this project, most aspects needed to be created from scratch.

**Lessons learned:**

* In the future, it would be better to divide up the code and have small chunks of code in different files interacting together, however, this is very hard in JavaScript. It merely requires a lot of variables in each file.
* Planning should be made before coding starts because specifically in this project I had a vague plan and when I got halfway through the project I had to redo huge chunks, which wasted almost a day out of my development time.

**Bugs**

* Currently the only 2 bugs that I know of are: every once in a while, when you scale a shape too fast, the program crashes by drawing a very big shape and the load feature sometimes takes two tries to load a file.

**Extra credit**

* I used bootstrap to style the page so that it doesn’t look like regular ‘boxy’ html. I also drew the design myself by hand.
* I implemented Line thickness.
* I implemented Line color.
* I implemented fill color.
* I implemented a delete button (RED) that deletes only one shape,
* I implemented a clear button, that clears everything.
* For color, I have rectangular markers that show you the color you’ve chosen.

**MANUAL**

I’ve included with this project a video that shows how to use most features of the program, but I’ll define a few here:

DRAWING SHAPES.

For a triangle, square, line, curve, rectangle, circle and ellipse, drag the button on to the canvas, the canvas will draw a standard shape with the line color, width and fill color selected. Then you can resize the shape as you see fit.

For a polyline and a polygon, click on the button and then move over to the canvas. Click and as you click different areas of the canvas, the polygon or polyline will be defined. When you feel you have got the shape you want, double click and the computer will register that as the new shape. For a polygon, the program automatically closes it, if it’s not closed.

SCALING

Every shape apart from the line has two circle icons when selected, the right most icon scales the shape with respect to the shapes horizontal(x) axis. The left icon scales the shape with respect to its vertical (y axis). Note: The shapes are scaled according to their axes, not the canvas axes, as this seems more natural and is what is used in most editing programs. The line can only be scaled on way, i.e. it can be made longer no matter what direction it faces but it cannot be made taller, that would make it a rectangle.

ROTATION

When you click the rotation icon, the shape rotates as long as you move the mouse, i.e. the rotation follows the mouse (demonstration in video).

SAVE/LOAD

All saves and loads are done in json format.

See demonstration video for further information.