Peyton Somerville

Final Project Report

This program is available to view online with the following link:

https://peyton-somerville.github.io/CompGraphicsProject/index.html

Issues Faced:

The first issue I faced was needing to refactor my code from HW3. My code for HW3 was not very efficient, because I used a different function to draw, rotate, scale, and translate for every shape. This led to the code having a ton of functions and it being very long. I knew that this code could be refactored, but I did not have time to do it before I submitted HW3. So the first thing I did for the project was refactor by making it so one function can handle the drawing for any shape, and one function can handle the rotation for any shape, etc. This took a while to get all of the functionality working again, but it was definitely worth it because my code is a lot more organized now. The next issue I faced was the way I was erasing shapes. In my HW3 the way to erase shapes involved drawing the same shape but with white as the color which is the same color as the background so it looks like the shape went away. The problem with this was if the shape that is getting erased is over another shape then the erasing would take out a chunk of the other since it is just changing pixels to white. This was fine in my HW3 because only one shape would be drawn at a time, but with the shape editor this needed to be fixed. The solution I found was to erase the whole canvas and then redraw every single shape. This meant that I needed a way to store all of the shapes that were currently on the screen. I accomplished this with an array that stores arrays of the x and y values. The size of each array was how I was able to tell which shape it was. If there were 6 values, then I knew there were 3 points, which means the shape is a triangle. If there were 10 values, then I knew there were 5 points, which means the shape is a pentagon, and so on. So when I redraw all of the shapes, I just go through each array in the shapes array, and use those points to draw the shapes. Similarly when we want to select a shape, I go through the shapes array, and check if the point clicked on is in any of the shapes, and if it is then select that shape.

Bugs/ Limitations:

Rotation and scaling work for the shape editor, but only with buttons. I really wanted to be able to rotate and scale using move movements, but I did not have time to make that work, so the user just has a bit less control and has to use the buttons. Some other limitations include exporting to png and jpeg. I have a shape menu that is a part of the canvas, so when the

images get exported they include the shape menu. I was able to fix this issue for the pdf, because I was able to specify the size to export, but I was not able to with png and jpeg. Also for some reason the background in the jpeg images is black instead of white. Some features that I did not get a chance to implement are drawing lines, curves, and polylines as well as multiple thickness support. There are a couple of bugs that I have found with my project. You can save and load with json, but when you load you will not be able to interact with those shapes. Another bug is if you change color multiple times on a shape without moving it first, it will create a duplicate shape.

Extra Credit:

There are only a couple of small extra credit features. The user can save a pdf. The user can save to png.