**Lab 4**

Purpose

This lab had a few purposes. One of them was to learn how to use the Git command line tool to manage the development of a project. The other purpose was to learn to use SVG and JQuery to draw shapes to the screen.

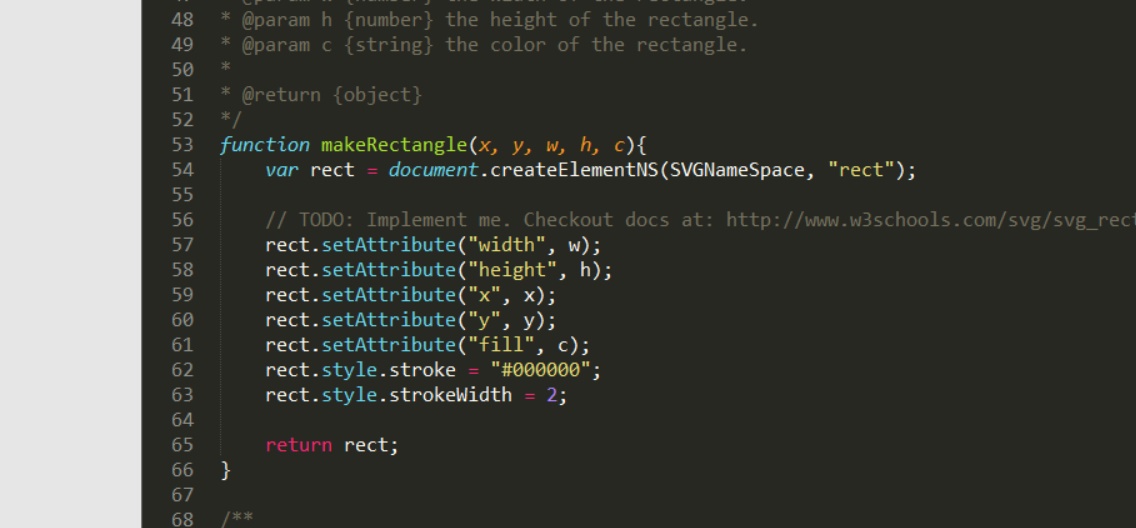
Overview

The first thing we did in the lab was go over some important Git commands, such as status, add, commit and log. We created a new repo and added and changed files inside the repo, adding and committing them to the git repository. The next thing we did in the lab was go over SVG and JQuery, and how to manipulate objects and print them to the screen. We drew elements by putting them into the html, and also drawing them in the JavaScript.

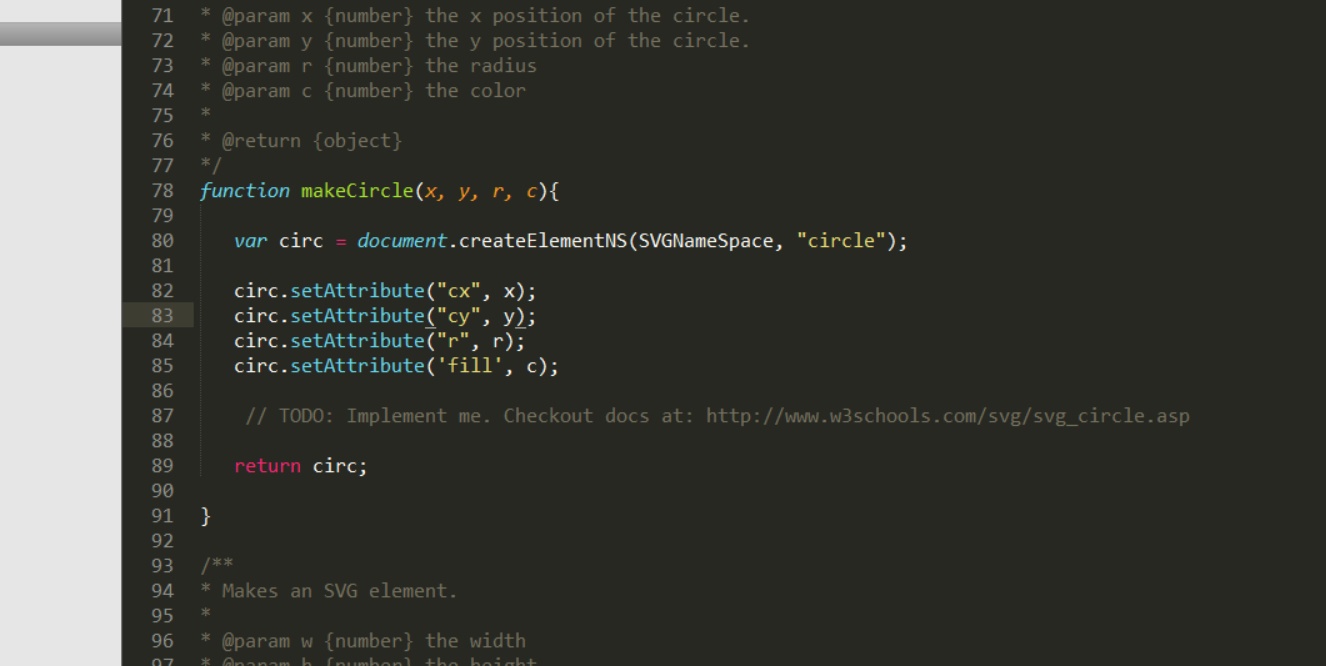
Lab Exercise

The first thing I did in the lab exercise was download the code from github. In downloading the code, I think it also downloaded the .git file, and saved all the previous commits.

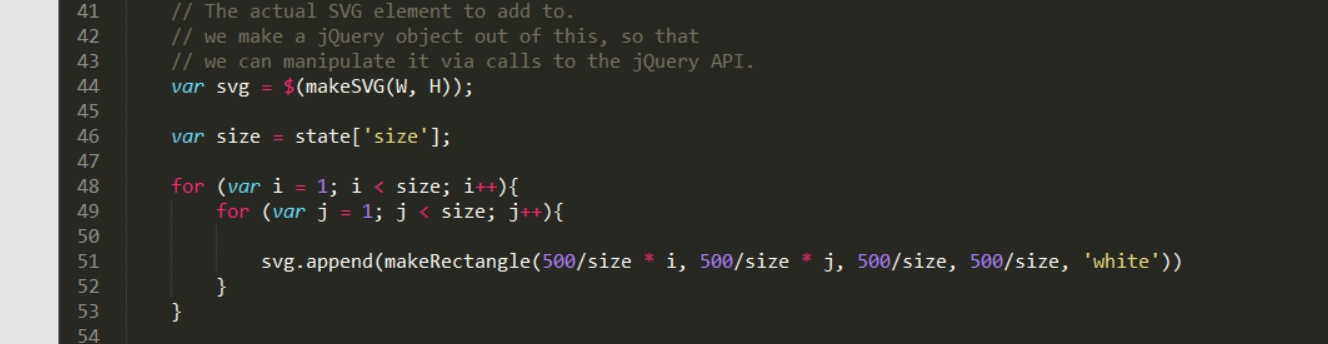
The next thing I did was implement the makeRectangle function in the svgFactory JavaScript. The function had already created the SVG rectangle object and returned it, all I had to do was take the parameters of the function and set their corresponding SVG square attributes to their values. The x and y coordinates, height and width, opacity, color, and border thickness.



Next I implemented the makeCircle function, which essentially followed the same steps as the makeRectangle implementation, setting the circle attributes to their corresponding function parameters.



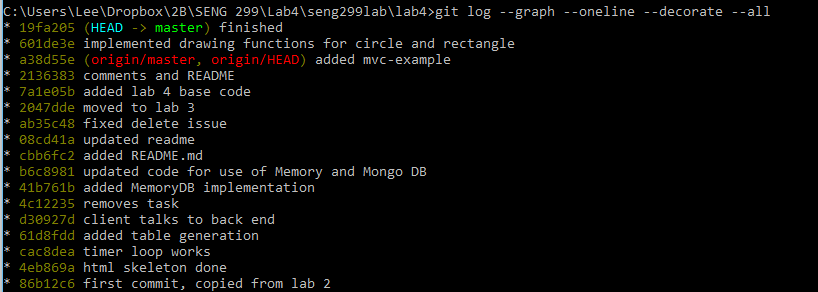
Lastly, I finished implementing the drawBoard function in the script JavaScript. I needed to add the drawing of squares and circles based on the data received from the board. I started by drawing the squares. I took the size of the board, and drew that minus one rows of that minus 1 squares by attaching them to the SVG object. I originally sized the squares by dividing the size of the canvas by the size of the board, but I found that forced any of my circles that were on the edge of the board to only be half drawn, so I used a number slightly smaller than the size of the canvas.



To draw the circles, I parsed the board object and for each element in each row-array of the board, I drew the corresponding colored circle, or lack thereof. Zero was no token, 1 was black and 2 was blue. The logic can be seen in the code snippet below, but essentially what I did was iterate through the board object and draw any of the specified tokens onto the screen by attaching them to the SVG object, keeping track of where each token should be placed using their indices.

Git

Here is a screenshot of the git logs for this project:



Problems

The only real problem I encountered during this lab was when the circles I was drawing on the edge of the board were being cut off by the edge of the canvas, but as soon as I thought about the canvas and how pinning SVG objects to it works, I realized I just needed to make the board smaller and keep it away from the edge of the canvas.