Scott Gordon

2/25/21

COMP 5460

Assignment Three Report

The purpose of this paper is to show the issues faced, lessons learned, and any remaining bugs in my assignment three, as well as list any extra effort that was put into the assignment past the requirements.

First, how my program functions from the user’s perspective. The user will select a shape and a function. Translation is simple and intuitive, the user clicks and drags, and the selected object will move the same amount as the mouse did from its clicked position. For scaling, the user can click anywhere and drag, the object is scaled larger if the mouse is dragged to the right and smaller if to the left. Rotation works in a similar way in which the selected object is rotated clockwise if the mouse is dragged to the right and counterclockwise if it is moved to the left. The scaling and rotation functions operate more smoothly if the mouse is moved slowly rather than quickly, this is a bug that I was unable to fix due to the time restraints on the assignment. I also noticed that the rate at which the operations occur differs if the developer tools window is open in chrome, I am not sure why this is, but I also consider this to be a bug.

Getting started on this assignment was not as simple as the other projects since we did not have any starter code to help us out. Learning about the canvas element a little more and changing the html file to display the required items was how I started.

I used simple a simple drop-down menu as well as radio buttons to select the shape and function respectively, and in the JavaScript, I use the selection to perform the correct process on the correct shape. One of the main struggles I had with this assignment was caused by my original design to hold the information for each object. I originally created objects with its data members being the coordinates of each point. After working with the project for a while I realized I need the data in the form of matrices for the calculations to be done.

Getting the event handlers working was also relatively simple. The ones I used were window.onload, as well as the ones for the canvas: onmousedown, onmouseup, and onmousemove.

At first, to clear the previously drawn shapes, I would calculate a bounding box for that shape and place a clear rectangle over it. After working with some of the shapes I realized I could just create a clear rectangle to fill the screen then redraw all the shapes, this might be more costly, but always gets the job done no matter the shape.

The largest struggle I had during this assignment was to correctly get the ‘rubberbanding’ when doing transformations. In the case of scaling, I originally tried to calculate the change in size using the mouse movement, but this proved to be ineffective with my implementation. The solution I landed on was scaling by a specific amount if the mouse movement was detected, and not have the scale amount be based on the distance moved. This provides, in my opinion, a satisfactory solution to allowing the user to scale the objects. My solution for rotation works in a similar way, rotating by a specific amount when mouse movement is detected.

The matrix multiplication was not too difficult, just need to keep in mind how the multiplication works and implement it to work with my defined matrices. A still image from my program is added below to show program output.

