**2d Transformation**

**Sample I/O:**

This time I didn’t include an inputs.js file because it would have been more cluttered that way without any context. In index.html, you can see where the input values are defined from lines ~35 to line 69. **MAKE SURE** that you uncomment/comment the other lines from 67 to 69 to test out the other kinds of transformations. I’ve implemented a switch statement to conditionally render one of the three types of transformations based on what the value of the variable “MODE” is at runtime. After changing this, resave the file and refresh the page!

**Lessons learned:**

This assignment really engraved how important matrix multiplication is to transformations in computer graphics. I thought I’d have difficulty with the scaling/rotations because they’re more complex in nature due to the offset points, etc, but going back to what we learned about applying multiple transformations at once made it very easy. Instead of having to fight with a shape’s local origin not being (0,0), I could simply multiple the shape by the matrix to move it back to (0,0), rotate the shape based on that origin, and then moving it back was super easy!!

**Issues faced:**

My biggest issue was trying to wrap my head around how exactly I’d express matrixes in code, and even harder was figuring out how I’d perform operations on the matrixes. Using 2-D arrays worked great, and using nested loops to iterate over the indices worked out just fine. I spent a while mapping out a plan on paper to figure out what indexes in each array were to be multiplied by each other, etc.

**Remaining bugs:**

Scaling up the 1px lines made it so the resolution was very bad – I’m not sure if this a bug because it’s how the algorithm is supposed to function. I use a 4k TV as a monitor and the lines didn’t really look any good scaled up past 1.5x their original 1px thickness!

**Additional Functionality:**

Not really applicable, except I modularized my code again to make it a lot easier to read. No values throughout the program are hardcoded so inputs need only be changed on the first page!