

Week 4 Homework – CMSC405

1. Show the results and intermediate steps for a translation of (-45,68), a rotation of 65 degrees and a scale of (0.5,2.2) applied to a starting point of (80,-100). You should use 3x3 matrix math for this example. Note: Your results should not reset to the starting point of 80,-100). The starting point of your subsequent transformation should be the output of the previous transformation. Also, you do not need to show matrix math calculations, but you should show the matrix and the starting points and the results for each geometric transformation.

Translation of (-45, 68):

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -45 \\ 0 & 1 & 68 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 80 \\ -100 \\ 1 \end{pmatrix} = \begin{pmatrix} 35 \\ -32 \\ 1 \end{pmatrix}$$

Rotation of 65 degrees:

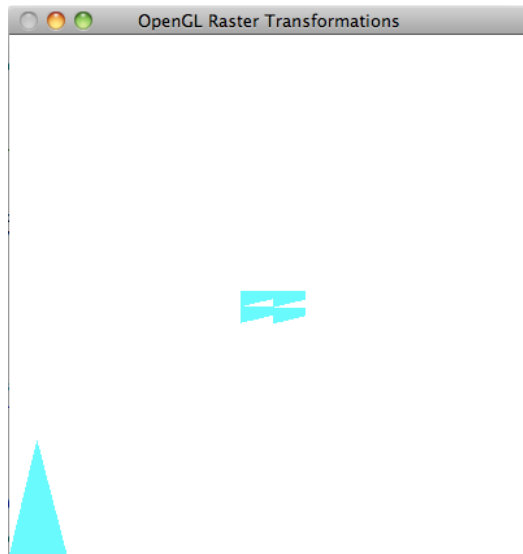
$$\begin{aligned} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} &= \begin{pmatrix} \cos(65) & -\sin(65) & 0 \\ \sin(65) & \cos(65) & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 80 \\ -100 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 0.5226 & -0.9063 & 0 \\ 0.9063 & 0.5226 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 80 \\ -100 \\ 1 \end{pmatrix} = \begin{pmatrix} 132.438 \\ 20.244 \\ 1 \end{pmatrix} \end{aligned}$$

Scale of (0.5, 2.2):

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 2.2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 80 \\ -100 \\ 1 \end{pmatrix} = \begin{pmatrix} 40 \\ -220 \\ 1 \end{pmatrix}$$

2. Using Visual C++ and your OpenGL configured environment, write an application that uses OpenGL Raster transformations to draw a triangle of your choice of size, location and color, and then rotate that triangle by 90 degrees and scale it by a factor of 1.5 in both the x and y directions.

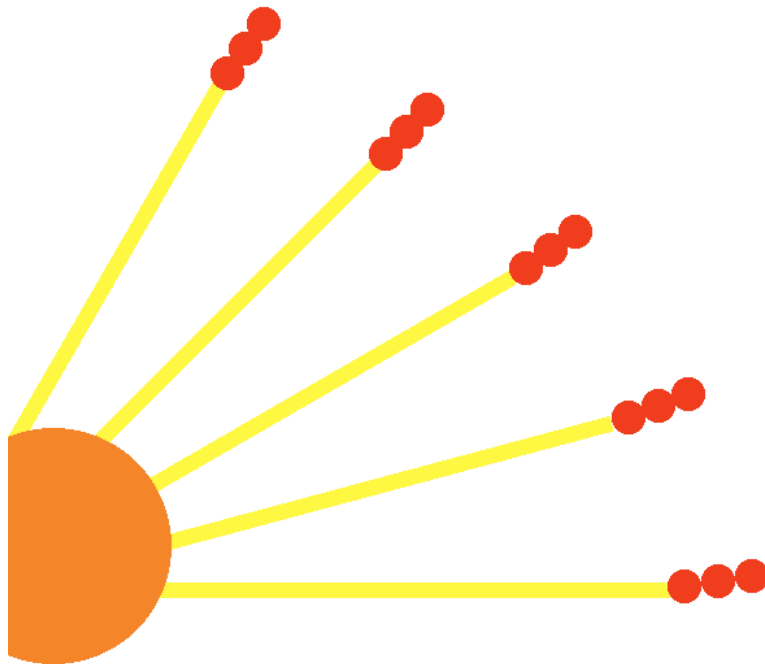
My execution result:



I am sorry I couldn't solve this problem...

3. Using Visual C++ and your OpenGL configured environment, write an application that uses both OpenGL geometric transformation functions and matrix multiplication to display a polygon shape, location, color and size of your choice. You should display the polygon in at least 6 different positions that include different rotation, translation and scale presentations. You should use matrix multiplication for at least half of the transformations. There are many different output results that are possible here. Be creative and try to output an interesting and unique geometric design.

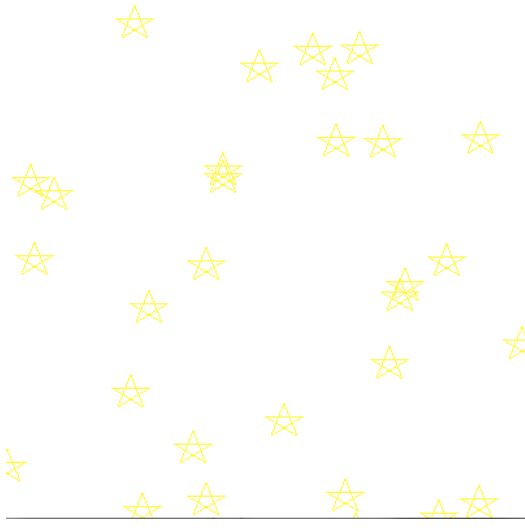
My execution result:



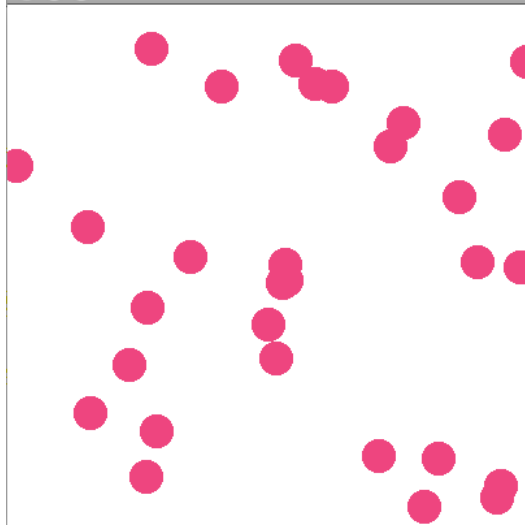
4. Using Visual C++ and your OpenGL configured environment, write an application that provides 3 windows for listing help commands, object inventory and a map for a simple game of your design. Once again, there is much flexibility in what you do here. There is no functionality in your game at this point. Think of this as 3 windows showing some text and graphics that would make sense to have in your game. Your help window would have a list of 5-10 commands you would use in the game. The object inventory would show 5-10 graphical objects your users might have in their inventory. These objects could be as simple as squares, rectangles and polygons or as complex as guns, swords, hats and pants. The map window provides a top down view of the game map. Keep this simple for now with a few shapes connected together with some text labeling each area. You are welcome to make it more complex but you may run out of time if you get too fancy. Have fun and be creative.

My execution result:

CMSC405 Window1



CMSC405 Window2



[HELP](#)

[CONTACTS](#)

[CREDITS](#)

[ITEMS](#)

[LEVEL](#)