Computer Graphics Lab 2 – 2D Transformations and Event Capturing

We will continue our study of 2D graphics through transformations. The next lab will be 3D graphics and we will harness the power of openGL. We will also look at capturing events in our graphics programmes which is very important.

- Write an openGL program that plots a square in the centre of the window.
- You should translate the square with the arrow keys (left, right, up, down)
- If you press 'R' the square should start/stop rotating about its centre point. So press 'R' once and the rotation is turned on, press it again and its turned off.
- If you press 'S' the square should scale bigger and smaller with the centre point remaining fixed throughout the scaling. You need to check the area of the square (or length of one of the sides) and when it gets close to 0, you should change the scaling to make it larger. When the area approaches the size of the window you should change the scaling to make it smaller. Press 'S' again and the scaling should be turned off.

So here is what your program will need:

- Eight variables for the x,y coordinates of the vertices
- A function called translate(dx,dy) that moves the square vertices by dx and dy
- A function called rotate(theta) that rotates the square vertices by theta about its centre point
- A function called scale(sf) that scales the square bigger and smaller by the sf

In the update function call the above functions to update the square vertices.

Hint: You can use Boolean variables to toggle rotation/scaling on/off. So for example if the user presses 'R' you make the boolean variable for rotation true and test this value in the update to decide whether to rotate the square or not.

In the display function draw the square using a GL_LINE_LOOP and glVertex instead of GL_POINTS

Your class should implement the java KeyListenenr interface and provide implementations of keyPressed/keyReleased etc.

The square should start off in the middle of the window. We can stick with the window size from the last lab and make the square 50 x 50 for example.

When the user presses one of the arrow keys the square will start moving (experiment with values of dx and dy so the square does not move too fast/slow). If the user keeps the button pressed the square should keep moving. Do not let the square leave the window.

When the user presses 'R' or 'S' the square will start rotating/scaling (possibly both at the same time). Again experiment with different values of theta/sf so that the square will update at a moderate rate.