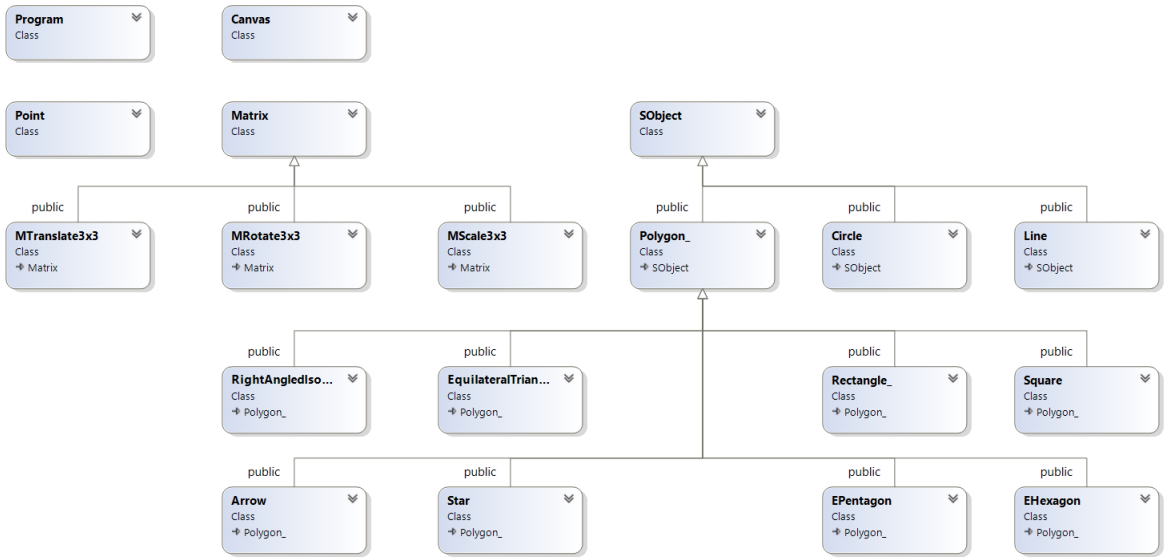
CS411 - Lab 04 - Report

Bui Xuan Bao - 18125063

A Class structure



B Details

1 Canvas

- Canvas stores 2D array representing for shape edge.
- If the value is positive number so the pixel is set to White. If the value is negative so the pixel is set to Red (the shape has that edge is being selected). Otherwise, the pixel is set to Black (no shape's edge has that pixel).

2 Line

- Drawing line bases on Digital Differential Analyzer algorithm.
- The starting point will be the left mouse down (LDOWN) location and the ending point will be the left mouse up (LUP) location.
- Drag the mouse to make the line drawing motion effect before LUP is found. The ending point will be temporally the mouse location (the same effect for other drawing).

3 Circle drawing

- Drawing bases on Midpoint algorithm.
- The center point will be LDOWN location.
- The distance from LUP location to LDOWN will be radius.

4 Polygon drawing

- For every LDOWN, the polygon will add that position to be its vertex then automatically connect the first vertex and the most recent vertex.
- The shape will stop drawing when a right mouse event is detected.
- Every 2 vertices are connected by using Line drawing (2).

5 Other shapes drawing

- All shapes inherits from Polygon class, but override its drawing method.
- First, calculate all the vertices of the shapes from LDOWN and LUP location.
- Then connect those vertices in order by using line drawing (2).

6 Selection

- For Line, we consider the direction of line vector with the direction of the vector created by the starting point and the selected point.
- For Circle, we consider the distance from the center point to the selected point with the radius.
- For Polygon and other shapes, we use the odd-even algorithm.

7 Transformation

- Polygon class and Line class have a transformation matrix to store all the old transformation.
- The reason is that the canvas store a finite number of pixels but the transformation makes a real number coordinate which is infinite for canvas to store. So after few transformations, we cannot make a inverse transformation to retrieve the initial shape.
- The transformation matrix will be initialized to be 3x3 identity matrix.
- For Circle class, the translation is applied to the center point and the scaling will scale the radius. Rotating circle is useless because it will remain the shape.

vii Translation

The pair of value (x,y) for UP, DOWN, LEFT, RIGHT will be (0,-1), (0,+1), (-1,0), and (+1,0), respectively.
The transformation matrix will multiply with a translate matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x & y & 1 \end{bmatrix}$$

vii Rotation

For clockwise rotation $< L >$, $\theta = \pi/180$ and for counter-clockwise rotation $< R >$, $\theta = -\pi/180$.
The rotation pivot point is the shape's center point (xp,yp). The transformation matrix will multiply the origin translation matrix, rotation matrix, and the pivot translation matrix.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -xp & -yp & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ xp & yp & 1 \end{bmatrix}$$

vii Scaling

For 10% size increase, S = 1.1 and for 10% size decrease, S = 0.9.
The scaling pivot is the shape's center point (xp, yp). The transformation matrix will multiply with a scale matrix and then with a pivot translation matrix:

$$\begin{bmatrix} S & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ xp*(1-S) & yp*(1-S) & 1 \end{bmatrix}$$

8 Program

- Main class for handling execution.
- Having:
  - A array of created SObject for finding the selected objects and do the transformation on it.
  - 2 canvasses: one for background storing all the drawn SObject, another for storing temporally drawing motion path.