Assignment - 9

Name : Tanmay Karmarkar
Roll No : 21143

Performance Date:
Submission Date:

Problem Statement:

Write a C++ program to implement translation rotation & scaling on equilateral triangle & rhombus. Concept of operator overloading.

Theory:

1. Translation:

It moves an object to a different position on the screen. We can translate a point in 2D by adding translation co-ordinates to original co-ordinates.

Initial : (x_0, y_0)
New : (x_n, y_n)
Translation vector : (Tx, Tx)

$x_n = x_0 + Tx$
$y_n = y_0 + \cancel{y_n}\, Ty$

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} Tx \\ Ty \end{bmatrix}$$

2. Rotation:

The object is rotated at particular angle θ from original position.

$x_n = x_0 \times \cos\theta + y_0 \times \sin\theta$
$y_n = y_0 \times \sin\theta + y_0 \times \cos\theta$

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

3. Scaling :

It is used to change the size of an object. In scaling, we either expand or compress the dimensions. Can be achieved by multiplying original coordinates with scaling factor.

$$X_n = X_0 \times S_x$$

$$Y_n = Y_0 \times S_y$$

$$\begin{bmatrix} X_n \\ Y_n \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \times \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

Homogeneous Co-ordinates

Matrix multiplication is easier for computers rather than addition, so we replace addition by multiplication in translation. We convert matrices to size 3×3.

1] Translation :

$$\begin{bmatrix} X_n \\ Y_n \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_0 \\ Y_0 \\ 1 \end{bmatrix}$$

2] Rotation :

$$\begin{bmatrix} X_n \\ Y_n \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ 1 \end{bmatrix}$$

3] Scaling :

$$\begin{bmatrix} X_n \\ Y_n \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ 1 \end{bmatrix}$$

Transformation on equilateral △.

vertices : $(x_1, y_1), (x_2, y_2), (x_3, y_3)$

After operation : $(x_1', y_1'), (x_2', y_2'), (x_3', y_3')$

$$\begin{bmatrix} x_1' & x_2' & x_3' \\ y_1' & y_2' & y_3' \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}$$

Transformation on rhombus.

vert : $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$,

new : $(x_1', y_1'), (x_2', y_2'), (x_3', y_3'), (x_4', y_4')$

$$\begin{bmatrix} x_1' & x_2' & x_3' & x_4' \\ y_1' & y_2' & y_3' & y_4' \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Similar for rotation & scaling.

Test cases:

1. Equilateral △ ⇒ $T_x = 10, T_y = 20$ → Translated Pass
2. Equilateral △ - $T_x = 20, T_y = 0$ → Translated Pass
3. Equilateral △ $\theta = 45^\circ$, → Translated Pass
4. Equilateral △ $S_x = 1.1, S_y = 1.1$ → Scaled Pass
5. Equilateral △ ~~[illegible]~~, $T_x = 2, T_y = 0$ → Translated Pass
6. Rhombus $T_x = 5, T_y = 0$ → Translated Pass
7. Rhombus $\theta = 20^\circ$ Rotated Pass
8. Rhombus $S_x = 0.5, S_y = 0.5$ → Scaled Pass
9. Rhombus $\theta = 90^\circ$ → Rotated Pass
10. Rhombus $T_x = 45, T_y = 10$ → Translated Pass.

Conclusion:

We understood basic transformations like scaling, rotation & translation.