Ex. No. : 3 Date:

Register No.: 231701055 Name: Suga V

2D Transformations – Translation, Scaling, and Rotation

AIM:

To write a program that performs:

- Translation
- Scaling
- Rotation on basic 2D shapes (e.g., triangle, rectangle) using transformation matrices.

Procedure:

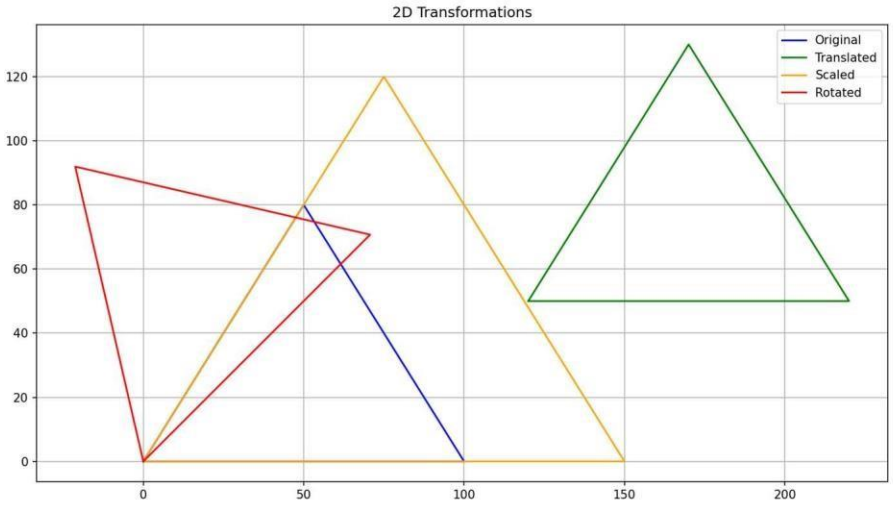
1. Define the 2D shape using its vertices.
2. Use matrix multiplication to perform:
    - Translation by adding offsets to coordinates.
    - Scaling by multiplying coordinates with scale factors.
    - Rotation by applying rotation matrix.
3. Display original and transformed shapes.

Program:

```
import numpy as np import
matplotlib.pyplot as plt

def draw_shape(points, label, color):
x, y = zip(*points)    x += (x[0],)    y
+=
```

```
(y[0],)          plt.plot(x,  y,  color=color,
label=label)

def translate(points, tx, ty):
   T = np.array([[1, 0, tx],
                 [0, 1, ty],
[0, 0, 1]])
   return apply_transform(points, T)

def scale(points, sx, sy):
   S = np.array([[sx, 0, 0],
                 [0,    sy,    0],
[0, 0, 1]])
   return apply_transform(points, S)

def         rotate(points,         angle_deg):
angle_rad = np.radians(angle_deg)
   R = np.array([[np.cos(angle_rad), -np.sin(angle_rad), 0],
                 [np.sin(angle_rad),  np.cos(angle_rad), 0],
                 [0, 0, 1]])
   return apply_transform(points, R)

def apply_transform(points, matrix):
transformed = []     for x, y in points:
vec = np.array([x, y, 1])          result
= matrix @ vec
      transformed.append((result[0], result[1]))     return transformed
```

```
# Original triangle triangle = [(0, 0),
(100, 0), (50, 80)]

# Transformations
translated = translate(triangle, 120, 50) scaled
= scale(triangle, 1.5, 1.5) rotated = rotate(triangle,
45)

# Plot
plt.figure(figsize=(8, 8))  draw_shape(triangle, "Original",
'blue') draw_shape(translated, "Translated", 'green')
draw_shape(scaled, "Scaled", 'orange') draw_shape(rotated,
"Rotated", 'red')

plt.title("2D Transformations")
plt.legend() plt.grid(True) plt.axis("equal")
plt.show()
```



Result:

The 2D transformations (translation, scaling, and rotation) were successfully applied using matrix operations on a triangle.