**Experiment**

**Aim- Write a program for the 2d Transformation**

**2D Transformation**

Theory

2D transformations in computer graphics involve manipulating objects within a two-dimensional plane, allowing for operations such as translation, scaling, rotation, and reflection. These transformations are crucial in applications like animations, graphics design, and game development, enabling dynamic control over the positioning, size, orientation, and shape of objects.

2D transformations change an object’s position, size, or orientation in a plane:

1. **Translation**: Moves the object by shifting its coordinates.
2. **Scaling**: Resizes the object by multiplying coordinates by scaling factors.
3. **Rotation**: Rotates the object around a point by a specified angle.
4. **Reflection**: Flips the object across an axis.

These transformations are fundamental in graphics for creating dynamic visuals and animations.

**Code-**

#include <GL/glut.h> // Include GLUT library for OpenGL

// Initialize variables for transformation

float translateX = 0.0f, translateY = 0.0f;

float scaleX = 1.0f, scaleY = 1.0f;

float rotationAngle = 0.0f;

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

// Set up orthographic projection

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-10.0, 10.0, -10.0, 10.0, -1.0, 1.0); // Define a 2D projection (left, right, bottom, top, near, far)

// Set up model-view matrix for transformations

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

// Apply transformations

glTranslatef(translateX, translateY, 0.0f); // Translation

glRotatef(rotationAngle, 0.0f, 0.0f, 1.0f); // Rotation

glScalef(scaleX, scaleY, 1.0f); // Scaling

// Draw a simple rectangle to see transformations

glBegin(GL\_QUADS);

glColor3f(1.0f, 0.0f, 0.0f); // Red color

glVertex2f(-1.0f, -1.0f);

glVertex2f( 1.0f, -1.0f);

glVertex2f( 1.0f, 1.0f);

glVertex2f(-1.0f, 1.0f);

glEnd();

glFlush();

glutSwapBuffers();

}

// Keyboard callback to adjust transformations

void keyboard(unsigned char key, int x, int y) {

switch (key) {

case 'w': translateY += 0.1f; break; // Move up

case 's': translateY -= 0.1f; break; // Move down

case 'a': translateX -= 0.1f; break; // Move left

case 'd': translateX += 0.1f; break; // Move right

case '+': scaleX += 0.1f; scaleY += 0.1f; break; // Scale up

case '-': scaleX -= 0.1f; scaleY -= 0.1f; break; // Scale down

case 'r': rotationAngle += 5.0f; break; // Rotate clockwise

case 'R': rotationAngle -= 5.0f; break; // Rotate counterclockwise

}

glutPostRedisplay(); // Redraw the display with updated transformations

}

int main(int argc, char\*\* argv) {

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(600, 600);

glutCreateWindow("2D Transformations with glOrtho");

// Set clear color (background) to white

glClearColor(1.0f, 1.0f, 1.0f, 1.0f);

// Register display and keyboard callback functions

glutDisplayFunc(display);

glutKeyboardFunc(keyboard);

// Enter main loop

glutMainLoop();

return 0;

}



