**SSN COLLEGE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**UCS1712 – GRAPHICS AND MULTIMEDIA LAB**

**EX NO: 5a – 2D Transformations – Translation, Rotation and Scaling**

**Name: Nachammai Devi Pooja S**

**RollNo:185001096**

**Date:29/08/2020**

**Aim:**

To write a C++ menu-driven program using OPENGL to perform 2D transformations – translation, rotation, scaling for line and polygon

**Algorithm:**

Step 1: Obtain no. of edges of polygon from user

Step 2: Obtain coordinates of vertices

Step 3: Obtain coordinates of line

Step 4: Plot the original polygon and line

Step 5: Obtain transformation option from user

Step 6: Translation:

• Get translation factor (x,y) as input from user

• Add (x,y) coordinate of polygon and line and plot

Step 7: Scaling:

• Get scaling factor (x,y) as input from user

• Multiply each coordinate of polygon and line by (x,y) and plot

Step 8: Rotation:

• Get angle of rotation in degrees as input from user

• Convert angle from degrees to radians

• Apply rotation formula on coordinates of polygon and line and plot

**Code:**

#include <stdio.h>

#include <math.h>

#include <iostream>

#include <vector>

#include <GL/glut.h>

using namespace std;

int pntX1, pntY1, op = 0, edges;

vector<int> pntX;

vector<int> pntY;

int transX, transY, lineX1, lineX2, lineY1, lineY2;

double scaleX, scaleY;

double angle, angleRad;

double round(double d)

{return floor(d + 0.5);}

void drawPolygon()

{

glBegin(GL\_POLYGON);

glColor3f(0.4, 0, 0.2);

for (int i = 0; i < edges; i++)

{

glVertex2i(pntX[i], pntY[i]);

}

glEnd();

glBegin(GL\_LINES);

glVertex2d(lineX1, lineY1);

glVertex2d(lineX2, lineY2);

glEnd();

}

void translation(int x, int y)

{

glBegin(GL\_POLYGON);

glColor3f(0.08, 4.67, 0.23);

for (int i = 0; i < edges; i++)

{

glVertex2i(pntX[i] + x, pntY[i] + y);

}

glEnd();

glBegin(GL\_LINES);

glVertex2d(lineX1 + x, lineY1 + y);

glVertex2d(lineX2 + x, lineY2 + y);

glEnd();

}

void scaling(double x, double y)

{

glBegin(GL\_POLYGON);

glColor3f(0.3, 0.4, 0.7);

for (int i = 0; i < edges; i++)

{

glVertex2i(round(pntX[i] \* x) + 300, round(pntY[i] \* y));

}

glEnd();

glBegin(GL\_LINES);

glVertex2d(round(lineX1 \* x), round(lineY1 \* y));

glVertex2d(round(lineX2 \* x), round(lineY2 \* y));

glEnd();

}

void rotation(double theta)

{

glBegin(GL\_POLYGON);

glColor3f(9.08, 0.67, 2);

for (int i = 0; i < edges; i++)

{

glVertex2i(round((pntX[i] \* cos(theta)) - (pntY[i] \* sin(theta))), round((pntX[i] \* sin(theta)) + (pntY[i] \* cos(theta))));

}

glEnd();

glBegin(GL\_LINES);

glVertex2d(round((lineX1 \* cos(theta)) - (lineY1 \* sin(theta))), round((lineX1 \* sin(theta)) + (lineY1 \* cos(theta))));

glVertex2d(round((lineX2 \* cos(theta)) - (lineY2 \* sin(theta))), round((lineX2 \* sin(theta)) + (lineY2 \* cos(theta))));

glEnd();

}

void myInit(void)

{

glClearColor(1.0, 1.0, 1.0, 0.0);

glColor3f(0.0f, 0.0f, 0.0f);

glPointSize(4.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-640.0, 640.0, -480.0, 480.0);

}

void myDisplay(void)

{

while (true) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0, 0.0, 0.0);

drawPolygon();

cout << "\nSelect the required transformation:\n";

cout << "1. Translation\n";

cout << "2. Scaling\n";

cout << "3. Rotation\n";

cout << "4. Exit\n";

cout << "Enter your choice : ";

cin >> op;

if (op == 4) {

break;

}

if (op == 1)

{

cout << "Enter the translation factor for X and Y: "; cin >> transX >> transY;

translation (transX, transY);

}

else if (op == 2)

{

cout << "Enter the scaling factor for X and Y: "; cin >> scaleX >> scaleY;

scaling(scaleX, scaleY);

}

else if (op == 3)

{

cout << "Enter the angle for rotation: "; cin >> angle;

angleRad = angle \* 3.1416 / 180;

rotation(angleRad);

}

glFlush();

}

}

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

{

cout << "\n2D-Transformations\n" << endl;

cout << "\ni)Polygon:\n" << endl;

cout << "Enter no of edges: "; cin >> edges;

cout << "\nEnter Polygon Coordinates : \n";

for (int i = 0; i < edges; i++) {

cout << "Vertex " << i + 1 << " : "; cin >> pntX1 >> pntY1;

pntX.push\_back(pntX1);

pntY.push\_back(pntY1);

}

cout << "\nii)Line:\n" << endl;

cout << "\nEnter Line Coordinates : \n";

cout << "Point 1 : "; cin >> lineX1 >> lineY1;

cout << "Point 2 : "; cin >> lineX2 >> lineY2;

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(640, 480);

glutInitWindowPosition(100, 150);

glutCreateWindow("2D-Transformations");

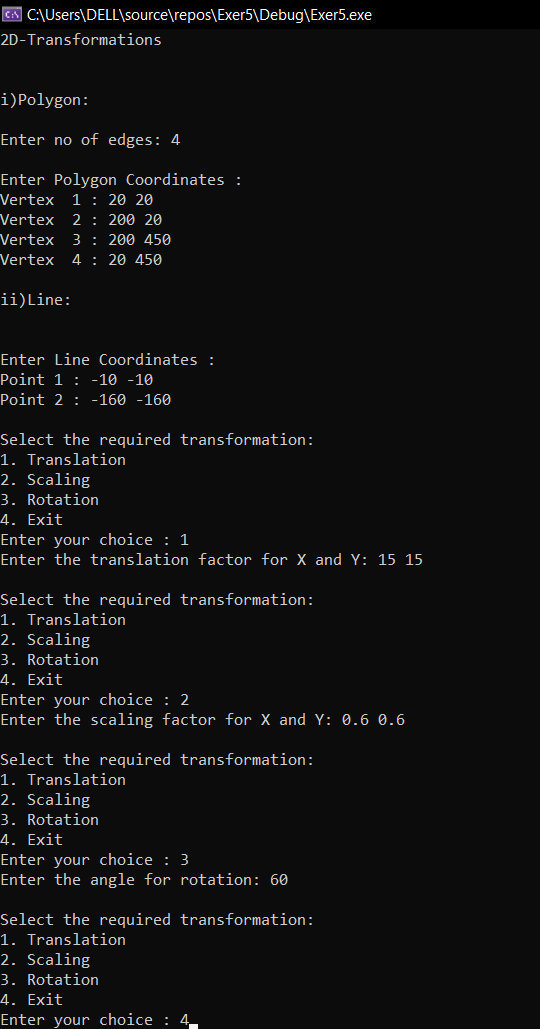
glutDisplayFunc(myDisplay);

myInit();

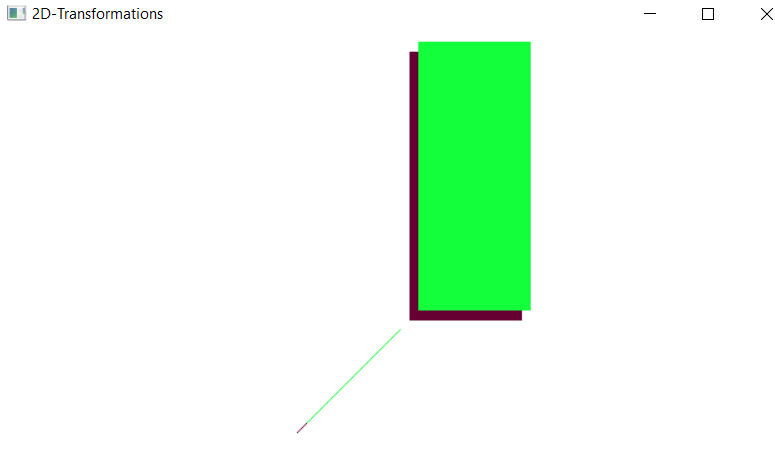
glutMainLoop();

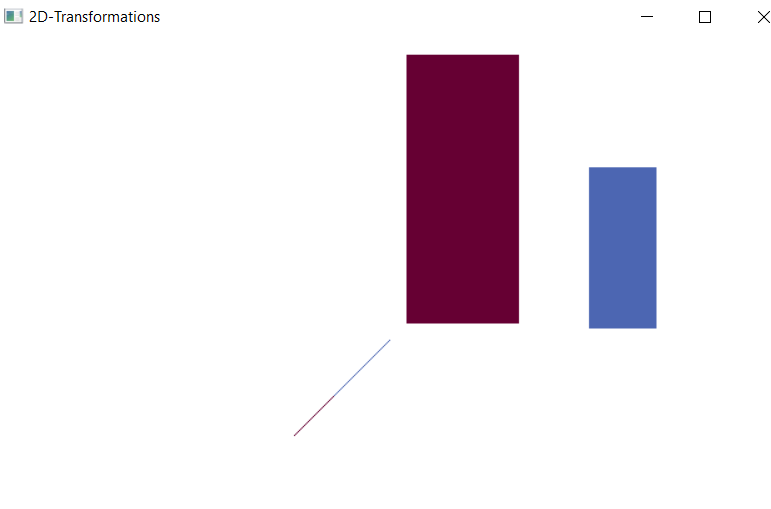
}

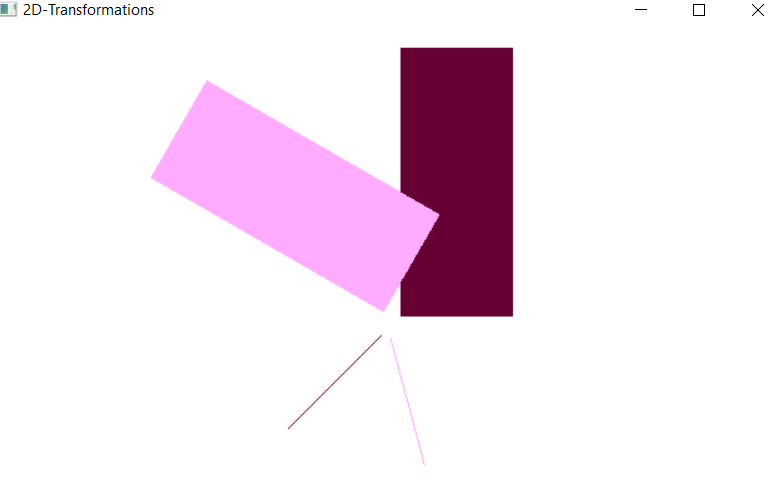
**OUTPUT:**

****

**1)Translation:**

**2)Scaling:**

**3)Rotation:**

****

**Result:**

A C++ menu-driven program using OPENGL to perform 2D transformations – translation, rotation, scaling for line and polygon was written and implemented successfully.