#include <GL/freeglut.h>

#include <GL/gl.h>

#include <math.h>

#include <iostream>

void multiplyMatrices(int a[4][2], int b[3][3], int result[4][2]) {

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 2; j++) {

result[i][j] = 0;

for (int k = 0; k < 3; k++) {

result[i][j] += a[i][k] \* b[k][j];

}

}

}

}

void rotateacw(int theta, int mat[4][2]) {

double thetaradians = theta \* M\_PI / 180.0;

int trans[3][3];

trans[0][0] = trans[1][1] = cos(thetaradians);

trans[0][1] = sin(thetaradians);

trans[1][0] = -sin(thetaradians);

trans[2][2] = 1;

int ans[4][2];

multiplyMatrices(mat, trans, ans);

glBegin(GL\_POLYGON);

for (int i = 0; i < 4; i++) {

glVertex2i(ans[i][0], ans[i][1]);

}

glEnd();

}

// Render function

void render() {

glClearColor(1, 1, 1, 1); // Set clear color to white

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0, 0, 0);

glBegin(GL\_POLYGON);

glVertex2i(150, 150);

glVertex2i(250, 150);

glVertex2i(250, 250);

glVertex2i(150, 250);

glEnd();

int mymat[4][2] = {{250, 250}, {350, 250}, {350, 350}, {250, 350}};

rotateacw(30, mymat);

glFlush();

}

// Main function

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(400, 400);

glutCreateWindow("Rotated Polygon");

gluOrtho2D(0, 400, 0, 400);

glutDisplayFunc(render);

glutMainLoop();

return 0;

}