**Koch curve**

#include <iostream>

#include <GL/glut.h>

#include <math.h>

class Point

{

public:

float x, y;

void setxy(float \_x, float \_y)

{

x = \_x;

y = \_y;

}

};

static int ITERATIONS = 4;

static Point startPoint = {100, 200};

static Point endPoint = {500, 200};

void init(void)

{

glClearColor(0.0, 0.0, 0.0, 0);

glColor3f(1.0, 1.0, 1.0);

glPointSize(2.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0, 600.0, 0.0, 400.0);

}

void setLine(Point p1, Point p2)

{

glBegin(GL\_LINES);

glVertex2f(p1.x, p1.y);

glVertex2f(p2.x, p2.y);

glEnd();

}

Point calculateKochPoint(Point p1, Point p2)

{

Point p;

p.x = (2 \* p1.x + p2.x) / 3;

p.y = (2 \* p1.y + p2.y) / 3;

return p;

}

void drawKochCurve(Point p1, Point p2, int iterations)

{

if (iterations == 0)

{

setLine(p1, p2);

}

else

{

Point p3, p4, p5;

float angle = 60 \* M\_PI / 180;

p3.x = p1.x + (p2.x - p1.x) / 3;

p3.y = p1.y + (p2.y - p1.y) / 3;

p5.x = p1.x + 2 \* (p2.x - p1.x) / 3;

p5.y = p1.y + 2 \* (p2.y - p1.y) / 3;

p4.x = p3.x + (p5.x - p3.x) \* cos(angle) + (p5.y - p3.y) \* sin(angle);

p4.y = p3.y - (p5.x - p3.x) \* sin(angle) + (p5.y - p3.y) \* cos(angle);

iterations--;

drawKochCurve(p1, p3, iterations);

drawKochCurve(p3, p4, iterations);

drawKochCurve(p4, p5, iterations);

drawKochCurve(p5, p2, iterations);

}

}

void display()

{

drawKochCurve(startPoint, endPoint, ITERATIONS);

glFlush();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutInitWindowSize(600, 400);

glutInitWindowPosition(100, 100);

glutCreateWindow("Koch Curve");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}