#include<GL/freeglut.h>

#include<GL/gl.h>

#include<math.h>

void koch(int xa, int ya, int xb, int yb, int n){

int xc, xd, yc, yd, midx, midy;

xc= (2 \* xa + xb) / 3;

xd= (2 \* xb + xa) / 3;

yc= (2 \* ya + yb) / 3;

yd= (2 \* yb + ya) / 3;

midx=xc+((xd-xc)\*cos(60\*3.14/180)) + ((yd-yc)\*sin(60\*3.14/180));

midy= yc-((xd-xc)\*sin(60\*3.14/180)) + ((yd-yc)\*cos(60\*3.14/180));

if(n>0){

koch(xa,ya,xc,yc,n-1);

koch(xc,yc,midx,midy,n-1);

koch(midx,midy,xd,yd,n-1);

koch(xd,yd,xb,yb,n-1);

}else{

glColor3f(0,0,0);

glBegin(GL\_LINES);

glVertex2i(xa,ya);

glVertex2i(xc,yc);

glVertex2i(xc,yc);

glVertex2i(midx,midy);

glVertex2i(midx,midy);

glVertex2i(xd,yd);

glVertex2i(xd,yd);

glVertex2i(xb,yb);

glEnd();

glFlush();

}

}

void draw(int n){

glColor3f(0,0,0);

glBegin(GL\_LINES);

koch(600, 100, 800, 400, n);

koch(400, 400, 600, 100, n);

koch(800, 400, 400, 400, n);

glEnd();

glFlush();

}

void renderfunc(){

glClearColor(1,1,1,0);

glClear(GL\_COLOR\_BUFFER\_BIT);

gluOrtho2D(0,1400,0,700);

draw(4);

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(1400,700);

glutInitWindowPosition(100,100);

glutCreateWindow("koch curve");

glutDisplayFunc(renderfunc);

glutMainLoop();

return 0;

}