#include<gl/glut.h>

#include<stdio.h>

int m;

typedef float point[3];

point tetra[4] = { {0,100,-100},{0,0,100},{100,-100,-100},{-100,-100,-100} };

void tetrahedron(void);

void myinit(void);

void divide\_triangle(point a, point b, point c, int m);

void draw\_triangle(point p1, point p2, point p3);

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

{

printf("Enter the number of iterations: ");

scanf\_s("%d", &m);

glutInit(&argv, argc);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB | GLUT\_DEPTH);

glutInitWindowPosition(100, 200);

glutInitWindowSize(500, 500);

glutCreateWindow("Seirpinski Gasket");

glutDisplayFunc(tetrahedron);

glEnable(GL\_DEPTH\_TEST);

myinit();

glutMainLoop();

}

void divide\_triangle(point a, point b, point c, int m)

{

point v1, v2, v3;

int j;

if (m > 0) {

for (j = 0; j < 3; j++)

v1[j] = (a[j] + b[j]) / 2;

for (j = 0; j < 3; j++)

v2[j] = (a[j] + c[j]) / 2;

for (j = 0; j < 3; j++)

v3[j] = (b[j] + c[j]) / 2;

divide\_triangle(a, v1, v2, m - 1);

divide\_triangle(c, v2, v3, m - 1);

divide\_triangle(b, v3, v1, m - 1);

}

else

draw\_triangle(a, b, c);

}

void myinit()

{

glClearColor(1, 1, 1, 1);

glOrtho(-500.0, 500.0, -500.0, 500.0, -500.0, 500.0);

;

}

void tetrahedron(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glColor3f(1.0, 0.0, 0.0);

divide\_triangle(tetra[0], tetra[1], tetra[2], m);

glColor3f(0.0, 1.0, 0.0);

divide\_triangle(tetra[3], tetra[2], tetra[1], m);

glColor3f(0.0, 0.0, 1.0);

divide\_triangle(tetra[0], tetra[3], tetra[1], m);

glColor3f(0.0, 0.0, 0.0);

divide\_triangle(tetra[0], tetra[2], tetra[3], m);

glFlush();

}

void draw\_triangle(point p1, point p2, point p3)

{

glBegin(GL\_TRIANGLES);

glVertex3fv(p1);

glVertex3fv(p2);

glVertex3fv(p3);

glEnd();

}







