**PROGRAM: SIERPINSKY GASKET USING OPENGL**

#include&lt;stdio.h&gt;

#include&lt;math.h&gt;

#include&lt;GL/glut.h&gt;

void myInit()

{

glClearColor(1.0, 1.0, 1.0, 1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-7.0, 7.0, -7.0, 7.0, -7.0, 7.0); // set viewing volume to 7 X 7 X 7

glMatrixMode(GL\_MODELVIEW);

}

void triangle(GLfloat \*a, GLfloat \*b, GLfloat \*c)

{

glVertex3fv(a); // draw triangle using vertices a, b, c

glVertex3fv(b);

glVertex3fv(c);

}

void draw\_triangle(GLfloat \*a, GLfloat \*b, GLfloat \*c, GLfloat \*d)

{

glColor3f(1.0, 0.0, 0.0);

triangle(a, b, c);

glColor3f(0.0, 1.0, 0.0);

triangle(a, b, d);

glColor3f(0.0, 0.0, 1.0);

triangle(a, d, c);

glColor3f(0.0, 0.0, 0.0);

triangle(b, c, d);

}

void draw\_tetrahedra(GLfloat \*a, GLfloat \*b, GLfloat \*c, GLfloat \*d, int k)

{

int j;

GLfloat ab[3], bc[3], ac[3], ad[3], bd[3], cd[3];

if(k&gt;0)

{

for(j=0;j&lt;3;j++) ab[j] = (a[j] + b[j])/2.0;

for(j=0;j&lt;3;j++) bc[j] = (b[j] + c[j])/2.0;

for(j=0;j&lt;3;j++) ac[j] = (a[j] + c[j])/2.0;

for(j=0;j&lt;3;j++) ad[j] = (a[j] + d[j])/2.0;

for(j=0;j&lt;3;j++) bd[j] = (b[j] + d[j])/2.0; for(j=0;j&lt;3;j++) cd[j] = (c[j] + d[j])/2.0;

draw\_tetrahedra(a, ab, ac, ad, k-1); draw\_tetrahedra(ab, b, bc, bd, k-1);

draw\_tetrahedra(ac, bc, c, cd, k-1);

draw\_tetrahedra(ad, bd, cd, d, k-1);

}

else

{

draw\_triangle(a,b,c,d); }

}

void display()

{

GLfloat a[3] = {0.0, 4.0, 0.0},

b[3] = {0.0, 0.0, -7.0},

c[3] = {4.0, 7.0, 6.0},

d[3] = {-6.0, 6.0, -4.0};

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

glBegin(GL\_TRIANGLES);

draw\_tetrahedra(a, b, c, d, 1);

glEnd();

glFlush();

}

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

{

glutInit(&amp;argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB | GLUT\_DEPTHglutInitWindowSize(500, 500);

glutInitWindowPosition(0, 0);

glutCreateWindow("3D Sierpinski Gasket");

glutDisplayFunc(display);

myInit();

glEnable(GL\_DEPTH\_TEST); glutMainLoop();

return 0;

