#include <GL/glut.h>

#include<iostream>

typedef GLfloat point[3];

point v[] = { {30.0, 50.0, 100.0}, {0.0, 450.0, -150.0},

{-350.0, -400.0, -150.0}, {350., -400., -150.0} };

int n;

void triangle(point a, point b, point c)

{

glBegin(GL\_TRIANGLES);

glVertex3fv(a);

glVertex3fv(b);

glVertex3fv(c);

glEnd();

}

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

{

point v0, v1, v2;

int j;

if (m > 0)

{

for (j = 0; j < 3; j++) v0[j] = (a[j] + b[j]) / 2;

for (j = 0; j < 3; j++) v1[j] = (a[j] + c[j]) / 2;

for (j = 0; j < 3; j++) v2[j] = (b[j] + c[j]) / 2;

divide\_triangle(a, v0, v1, m - 1);

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

divide\_triangle(b, v2, v0, m - 1);

}

else(triangle(a, b, c));

}

void tetra(int m)

{

glColor3f(1.0, 0.0, 0.0);

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

glColor3f(0.0, 1.0, 0.0);

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

glColor3f(0.0, 0.0, 1.0);

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

glColor3f(0.0, 0.0, 0.0);

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

}

void display()

{

glClearColor(1.0, 1.0, 1.0, 1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

tetra(n);

glFlush();

}

void myinit()

{

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-499.0, 499.0, -499.0, 499.0, -499.0, 499.0);

glMatrixMode(GL\_MODELVIEW);

}

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

{

printf("enter divisions");

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutCreateWindow("3D Gasket");

glutDisplayFunc(display);

myinit();

glutMainLoop();

}