


Name : Shawn Louis		Batch : B	Roll No : 31
EXPERIMENT 09			
Title	SIERPINSKY GASKET IN OPENGL		
Objective	To write a C program for sierpinsky gasket.		
Program	<pre> #include <GL/glut.h> #include<math.h> #include<time.h> /* a point data type typedef GLfloat point2[2]; /* initial triangle – global variables */ typedef GLfloat point2[2]; point2 v[]={{-1.0, -0.58}, {1.0, -0.58},{0.0, 1.15}}; int n; /* number of recursive steps */ void myinit() { glMatrixMode(GL_PROJECTION); glLoadIdentity(); gluOrtho2D(-2.0, 2.0, -2.0, 2.0); glMatrixMode(GL_MODELVIEW); glClearColor (1.0, 1.0, 1.0,1.0); glColor3f(0.0,0.0,0.0); } void triangle(point2 a, point2 b, point2 c) /* display one triangle */ { glBegin(GL_TRIANGLES); glVertex2fv(a); glVertex2fv(b); glVertex2fv(c); glEnd(); } void divide_triangle(point2 a, point2 b, point2 c,int m) { /* triangle subdivision using vertex coordinates */ point2 v0, v1, v2; int j; if(m>0) { for(j=0; j<2; j++) v0[j]=(a[j]+b[j])/2; for(j=0; j<2; j++) v1[j]=(a[j]+c[j])/2; for(j=0; j<2; 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); } } </pre>		

	<pre>else(triangle(a,b,c)); /* draw triangle at end of recursion */ } void display(void) { glClear(GL_COLOR_BUFFER_BIT); divide_triangle(v[0], v[1], v[2], n); glFlush(); } int main(int argc, char **argv) { n=4; glutInit(&argc, argv); glutInitDisplayMode(GLUT_SINGLE GLUT_RGB); glutInitWindowSize(500, 500); glutCreateWindow("2D Gasket"); glutDisplayFunc(display); myinit(); glutMainLoop(); }</pre>
Output	<pre>shawn@shawn-VirtualBox:~/Desktop\$ gedit SierpinskyGasket.c shawn@shawn-VirtualBox:~/Desktop\$ gcc SierpinskyGasket.c -lglut -lGLU -lGL shawn@shawn-VirtualBox:~/Desktop\$./a.out shawn@shawn-VirtualBox:~/Desktop\$</pre>

	<div data-bbox="349 191 1425 247"><div>2D Gasket</div><div><div></div><div></div><div></div></div></div> <div data-bbox="613 478 1156 953"></div>
Conclusion	Thus a C program to generate Sierpensky Gasket was written and executed.