

Program 1

Due Oct 4 at beginning of class

/* This program heavily based on program found in the out-of-print book "Interactive Computer Graphics a top-down approach with OpenGL" by Edward Angel

Type in, compile, and run this program. The main purpose of this program is to make sure your environment is correctly set up so that you can compile and run the program. The reason you are typing in this program is so that you pay attention to the basic syntax of a OpenGL program.

I am looking for a visually pleasing result. Change the background color and vary the colors of the plotted points.

*/

```
#include < stdlib.h >
#include < GL/glut.h >
#include < GL/gl.h >
#include < GL/glu.h >

int num_iterations=5000;
#define ITERATIONS 2500
#define WIN_XSIZE 500 // x dimension of window is 500
#define WIN_YSIZE 500 // y dimension of window is 500
//These must be smaller values than the window size
#define VIEW_XMIN 0.0 // minimum x dimension of viewport is 0
#define VIEW_YMIN 0.0 // minimum y dimension of viewport is 0
#define VIEW_XMAX 500.0 // maximum x dimension of viewport is 500
#define VIEW_YMAX 500.0 // maximum y dimension of viewport is 500

void myinit(void)
{
    //Attributes
    glClearColor(1.0, 1.0, 1.0, 1.0); // white background
    glColor3f(1.0, 0.0, 0.0) ; //draw in red

    /******
    //set up viewing
    //500x500 window with origin lower left
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    //define viewing area All of these values are defined at top
    //what happens if these vales are changed?
    gluOrtho2D(VIEW_XMIN, VIEW_XMAX, VIEW_YMIN, VIEW_YMAX);
    /******

    //return to GL_MODELVIEW matrix
    glMatrixMode(GL_MODELVIEW);
}

void display(void)
{
    int i, j, k;

    //define a point data type
    typedef GLfloat point2[2];

    //define a triangle
    point2 vertices[3] = {{0.0,0.0}, {250.0, 500.0}, {500.0, 0.0}};

    point2 p = {75.0, 50.0}; //arbitrary initial point inside triangle

    glClear(GL_COLOR_BUFFER_BIT); //clear the window
```

```

//compute and plot ITERATIONS new points
for (k=0; k < ITERATIONS; k++)
{
    j=rand()%3; //pick a vertex at random

    //compute point halfway between selected vertex and old point
    p[0] = (p[0] + vertices[j][0])/2.0;
    p[1] = (p[1] + vertices[j][1])/2.0;

    //plot new point
    glBegin(GL_POINTS);
        glVertex2fv(p);
    glEnd();
}
glFlush(); //clear buffers
}

int main (int argc, char **argv)
{
    //Standard GLUT initialization
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB); //default not needed

    //Initialize the pixel window to be WIN_XSIZE x WIN_YSIZE
    glutInitWindowSize(WIN_XSIZE, WIN_YSIZE);

    glutInitWindowPosition(0,0); //place window at top left of display
    glutCreateWindow("Sierpinski gasket"); //window title
    glutDisplayFunc (display); //display callback when window invoked

    myinit(); //set attributes

    glutMainLoop(); //enter event loop
}

```