

Drawing Points and Lines in OpenGL

Loyola Marymount University

CMSI-371: Introduction to Computer Graphics

Professor Alex Wong

Previously, we created a simple OpenGL program to run on Mac OSX and Ubuntu systems. Note that there existed different headers needed for each operating system.

To make your header files compatible with both operating systems, I have added checks to import the correct headers. I also added `glPointSize()` to better visualize the points.

Below is a short program that creates four points and draws two lines connecting them.

A quick explanation on some of the functions:

`glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)` clears the RGB color and the depth (for perspective) bits.

The `display` function is passed to `glutDisplayFunc` as a function pointer

`glBegin` and `glEnd` closures tells OpenGL to perform a given task such as draw points at a location or lines between a pair of points.

```
glBegin(...);
```

```
glEnd();
```

`glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)` clears the RGB color and the depth (for perspective) bits.

*** These should technically be the only tools you will need to extend to drawing curves as our algorithms for sketching curves are essentially creating new points that lie on the curve and connecting them.

opengl_points_lines.cpp

```
#ifdef __APPLE__
#include <OpenGL/gl.h>
#include <OpenGL/glu.h>
#include <GLUT/glut.h>
#else
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glut.h>
#endif

void setup() {
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
}

void display() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    // Set our color to black (R, G, B)
    glColor3f(0.0f, 0.0f, 0.0f);
    // Draw 4 points with coordinates (x, y) counter-clockwise
    // (0.5, 0.5), (-0.5, 0.5), (-0.5f, -0.5f) (0.5f, -0.5f)
    // Note if we want clockwise then add: glFrontFace(GL_CW)
    glPointSize(10.0f);
    glBegin(GL_POINTS);
        glVertex2f(0.5f, 0.5f);
        glVertex2f(-0.5f, 0.5f);
        glVertex2f(-0.5f, -0.5f);
        glVertex2f(0.5f, -0.5f);
    glEnd();
    // Draw 2 lines from
    // (0.5, 0.5) to (-0.5, 0.5) and (-0.5f, -0.5f) to (0.5f, -0.5f)
    glBegin(GL_LINES);
        // First line
        glVertex2f(0.5f, 0.5f);
        glVertex2f(-0.5f, 0.5f);
        // Second line
        glVertex2f(-0.5f, -0.5f);
        glVertex2f(0.5f, -0.5f);
    glEnd();
    glutSwapBuffers();
}
```

```
int main(int argc, char *argv[]) {  
    glutInit(&argc, argv);  
    glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH | GLUT_DOUBLE);  
    glutInitWindowSize(800, 600);  
    glutCreateWindow("Points and Lines");  
    setup();  
    glutDisplayFunc(display);  
    glutMainLoop();  
  
    return 0;  
}
```