Lab 2 — OpenGL Triangles, Lines, Line Loops, and Triangle Fans

CS 320

Feb. 6, 2013

1. Download the Lab 2 ZIP archive and the commonFiles ZIP archive from the class home page and unpack them.

2. Start Eclipse and create a new C++ project. Import all the files from both unpacked ZIP archives into your project.

   Don't forget to add the OpenGL, GLEW, and GLUT libraries to your project.

3. Compile and run the program. All should be well.

4. Modify the program so that it draws triangle outlines rather than filled triangles:

   (a) For the first approach, we'll generate the triangle outlines using three line segments per triangle. Referring to the line numbered source code listing in the `example2` PDF file, this will involve changes to `NumVertices` in line 10, `triangle()` in lines 17–23, and `glDrawArrays()` in `display()` in lines 82–88.

   (b) (I'd suggest starting with a new project for this second approach — "It don't cost nothin'.") For the second approach, we'll generate triangle outlines using a single line loop per triangle. Again referring to the line numbered source code listing in the `example2` PDF file, this will involve changes to `display()` in lines 82–88.

5. Write a small OpenGL program to render the outlines of three circles. As a starting point, use a copy of your line loop Sierpinksi triangle program from above. You will no longer need `triangle()` and `divide_triangle()`. You can also dispose of some of the `const`s at the top of the program. What you will need is a function to generate the vertices along the circumference of the circle for you. Here's the function's stub:

   ```
   void
   circle( const vec2& center, const GLfloat& radius, const GLuint& slices)
   {
      /* The code generating the vertices to render a circle goes here. */
   }
   ```

   We'll develop the algorithm for this function in class. `init()` will make three calls to this function.

6. Write a small OpenGL program to render three filled circles. You'll need to use a triangle fan for this. We'll discuss the details of the algorithm in class.