

**Problem 0:** Follow the instruction on the <http://www.ece.lsu.edu/koppel/gpup/proc.html> page for programming homework work flow, substituting hw2 where appropriate. Compile and run the homework code unmodified. It should show a chain of balls swinging next to a starkly illuminated large chocolate cone. (Cone refers to the shape, it's not a chocolate ice cream cone.)

The green thing on the left-hand side is the light, it should leave a brightly lit spot on the chocolate cone but it does not because the cone is rendered using large primitives. In this assignment that problem will be fixed, but that's the problem below. This problem is just to familiarize yourself with the code.

The code for rendering the chocolate cone and the links in the chain is in the class `Cone` near the top of the file `hw2.cc`. The routine `Cone::render` is called when it's time to render a cone. The chocolate cone is an instance named `cone_fixed`, the chain links are all a second instance named `link`.

In this assignment the `Cone` class will be rendered to fix light problem by using a larger number of primitives to tessellate the cone shape.

Use the arrow keys, PageUp, and PageDown to move around the scene. Use key `h` to toggle between the first (head) ball being locked in place and free. Use key `t` to do the same for the last (tail) ball. Press (lower-case) `b` and then use the arrow and page keys to move the first ball around. Press `l` to move the light around and `e` to move the eye. Press `1` to set up scene 1, press `2` to set up scene 2.

Look at the comments in the file `hw2.cc` for documentation on other keys. One fun thing to do is to lock both the first and last ball, move the head ball until the spring is stretched tight, then release one of the balls. Press `p` to pause, then the space bar to single step. *Note: There is nothing to turn in for this first problem.*

**Problem 1:** Modify the member function `render_proba` so that it constructed from  $10 * \text{opt\_lod}$  quads along the circumference and `opt_lod` quads along the length, for a total of  $10L^2$  quads, where  $L$  is the value of global variable `opt_lod`. (LOD is an abbreviation for level of detail.) Variable `opt_lod` can be adjusted using the keyboard by pressing the `Variables` section of the comments at the top of the code.

**Problem 2:** Add code to `render_prob` and elsewhere so that a buffer object will be used to render the cone.

- Declare storage for the buffer object name and other variables in the `Cone` class.
- The buffer object should be updated whenever the level of detail changes.
- Be careful not to leak memory. That's a common problem in assignments like this. (A memory leak is when memory is repeatedly allocated but not freed.)

Use the following resources to help solve the problem: The `demo-7-vtx-arrays.cc` example shows how to use buffer objects. The file is available several different ways, including a colorized version meant for a Web browser,

<http://www.ece.lsu.edu/koppel/gpup//code/gpup/demo-7-vtx-arrays.cc.html>. If you followed the account setup instructions you can find `demo-7-vtx.arrays.cc` in the `gpup` directory in your class account. (Executing `svn update` in that directory will update those files with the latest changes.)

The documentation for buffer objects can be found in Chapter 6 of the OpenGL 4.3 compatibility profile specification which can be found at <http://www.ece.lsu.edu/gp/refs/glspec43.compatibility.20120806.pdf>. Commands for specifying vertices and attributes can be found in Section 10.3, however the section might be difficult to read because of the many different ways vertices can be specified.