LSU EE 4702-1 Homework 2 Due: 10 October 2012

Follow the Programming Homework Work Flow *instructions on the procedures page, substituting* `hw2` *for* `hw1`.

The `hw2` code, based on `demo-8-texture`, draws a helix. Due to missing normals, the surface of the helix will not be shaded properly (see Problem 1).

The comments at the top of the file describe controls for the simulation. For this assignment controls have been added to change the amount of detail in the helix (search for "Big Circle" and "Small Circle"), to change how vertices are ordered (look at the `i` and `m` options), and to force computation and communication (look at `d` and `r`).

Problem 1: Run the unmodified `hw2` code and pay attention to the way the helix is lit. Notice that the shading of the helix is wrong. That's because normals have not been specified. Modify the code so that normals are correctly specified. To do this new arrays and a new buffer object will have to be added.

Problem 2: The use of triangle strips reduces vertex redundancy compared to individual triangles. However for the `hw2` program there is still redundancy. In particular, point `s1` (search for "Construct a Helix" and scroll down to the `j` loop) is the same as `s0` in a prior `i` iteration.

By using `glDrawElements` instead of `glDrawArrays` one can avoid this redundancy. That is, the command sends each point to the GPU once and a separate index array specifies which of those points are used, a point can be used multiple times.

(*a*) Modify the code to use `glDrawArrays` when variable `opt_use_elements` is true. (Use key `m` to change the value of `opt_use_elements`.) Some code is already in place, including code declaring and initializing the index array and the call to `glDrawElements`. However, the existing code still adds the same point to the `helix_coords` list multiple times. The index array (`indices`) is set to 0, 1, 2, ..., which makes `glDrawElements` equivalent to `glDrawArrays`.

Modify the code so that each point is added only once, making appropriate changes to `prep_indices` (which is used to set `indices`) so that the helix is drawn correctly. See Section 10.5 of OpenGL version 4.3 for use `glDrawArrays`.

(*b*) The code has a variable `opt_use_strips`. Modify the code so that when it's true the helix will be rendered using triangle strips and when it's false it will be rendered using individual triangles. *Note: This part did not appear in the original assignment.*

Problem 3: In this problem try to measure the impact of using `glDrawElements` to reduce the redundancy.

(*a*) Compute the amount of data sent from the CPU to the GPU in terms of the number of triangles used in the helix with and without `glDrawElements`. Use T to denote the number of triangles.

(*b*) Measure the impact on performance of this difference in data size. Use the various program options to make this performance difference large. Explain which options were used and show the results.

(*c*) Try to determine if there is any difference in performance due to computation. There would be less computation using `glDrawElements` if the vertex shader were run on each vertex only once. But does that actually happen? Explain which options were used and show the results.