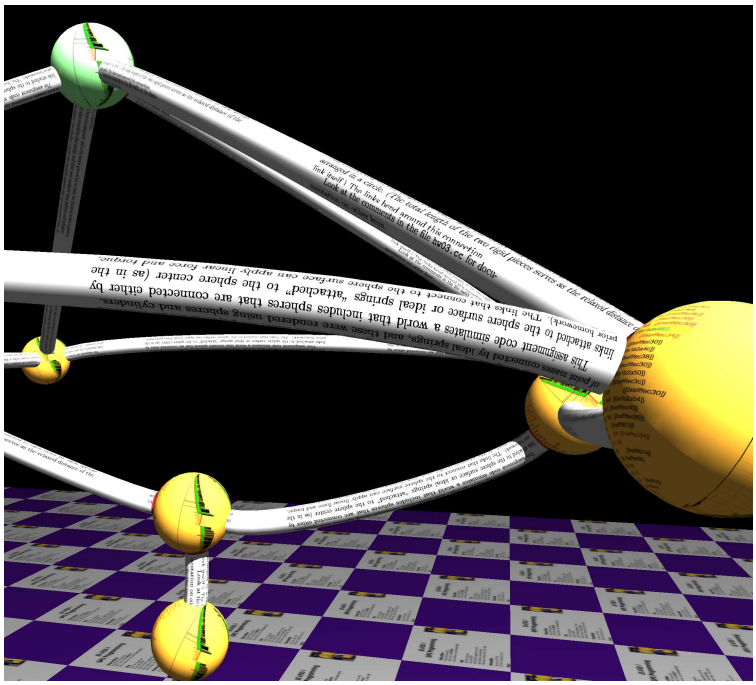
LSU EE 4702-1 **Homework 4** **Due: 10 October 2016**

Note: These problems were originally part of Homework 3. Solve this assignment in file `hw03.cc`*, the same one used for Homework 3.*

Problem 0: If necessary, follow the instructions on the `http://www.ece.lsu.edu/koppel/gpup/proc.html` page for account setup and programming homework work flow. Compile and run the homework code unmodified. It should initially show a vaguely umbrella-shaped object hanging in space.



The screenshot on the right shows what the object should look like after Problem 1 of Homework 3 is solved correctly.

Press digits `1` through `2` to initialize different scenes, the program starts with scene 2. Scene 1 shows a balls connected in a rectangular spiral. *Promptly report any problems.*

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.

Initially the arrow keys, PageUp, and PageDown can be used to move around the scene. 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 (which is what the arrow keys do when the program starts).

Look at the comments in the file `hw03.cc` for documentation on other keys.

In Homework 2 and in the demos given so far in class, the physical model consisted of point masses connected by ideal springs, and these were rendered using spheres and cylinders.

This assignment code simulates a world that includes spheres that are connected either by links attached to the sphere surface or ideal springs "attached" to the sphere center (as in the prior homework). The links that connect to the sphere surface can apply linear force and torque, and they can bend. In the physical model each link consists of two perfectly rigid pieces, one end connected to the respective balls and the other end connected to each other by ideal springs arranged in a circle. (The total length of the two rigid pieces serves as the relaxed distance of the link itself.) The links bend around this connection.

Since a link can bend it would not make sense to render it as a cylinder. Instead they are rendered as cubic Hermite curves. Consider a link that connects balls B_1 and B_2 with centers at C_1 and C_2. Let P_1 and P_2 denote the coordinate where the link attaches to B_1 and B_2. The link is rendered as a cubic Hermite curve parametrically described by $L(t) = (2t^3 - 3t^2 + 1)0\overrightarrow{P_1} + (-2t^3 + 3t^2)0\overrightarrow{P_2} + s(t^3 - 2t^2 + 1)\overrightarrow{C_1P_1} + s(t^3 - t^2)\overrightarrow{P_2C_2}$, where s is the stiffness of the links and is set to $s = \|2.5\overrightarrow{P_1P_2}\|$ in the code.

In `hw03.cc` routine `render_link_1` and `render_link_2` render these new curved links as triangle strips. The `i` loop moves along the curve, the value of $L(t)$ is assigned to variable `ctr`. The derivative $\frac{dL(t)}{dt}$ gives the direction of the curve. That is assigned to `tan`. The `j` loop draws a cylinder (sort of) around the curve.

Problem 1: Modify `render_link_2` so that it uses buffer objects for vertices, normals, and texture coordinates as described below. (To switch between `render_link_1` and `render_link_2` press `z`.)

(a) Modify the routine so that it uses a buffer object to hold texture coordinates. The buffer object should be written just once and used each time. (This will result in text appearing compressed on short links and stretched on long ones.)

(b) Modify the routine so that it uses a set of buffer objects to hold the vertices and normals of curved links. (See the next part.) These buffer objects will have to be updated each time `render_link_2` is called.

(c) Modify the routine so that it uses another set of buffer objects to hold the vertices and normals of straight links. These buffer objects should be written to once and used many times. Straight links, because they use re-cycled buffer objects, should be shown in pale green.

A link is straight if the link directions (`dir1` and `dir2` in the code) are the same direction as the line connecting the link endpoints (`pos1` and `pos2`) in the code. Consider a link straight if the link directions are close enough to the line connecting their endpoints. After the scene calms down, the vertical links in the umbrella should be considered straight, and perhaps the four links on top. If gravity is turned off (`g`) all links should be considered straight.

All straight links are in the shape of a cylinder. They differ however in their lengths and orientations and placement. For straight links you will need to use a transformation so that the straight link used to populate the buffer object is transformed for the link for which it will be used.

To solve this correctly storage will be needed for the buffer object names, and information about the link it describes. Declare members in World for this purpose. Do not declare variables in global scope.

For this problem, the best way to construct a rotation matrix is by combining unit vectors to form either the rows or columns of a matrix. These can be constructed using `pMatrix_Rows` and `pMatrix_Cols`. For example `pMatrix_Rows rot(v1,v2,v3);` constructs a matrix `rot` whose rows are `v1`, `v2`, and `v3`, respectively. Let `v1`, `v2`, and `v3` be some orthogonal unit vectors in some global space. Matrix `rot` can be used to convert a point to some local space, such that direction `v1` in object space is the x axis in the local space, etc. With `pMatrix_Cols rot(v1,v2,v3);` a point can be transformed to a global space. For this homework problem, one might set `v3` to the link direction of the straight link (making the link direction the z axis in a local space), see the code for suggestions for x and y directions.

To solve this problem two transformations are needed. One, to transform the straight link in the buffer object to some local space. A second one is needed to transform it for the link that's needed.

Do not construct a link in some local space and re-use that. It's not a bad thing to do in general, but don't do it here.