

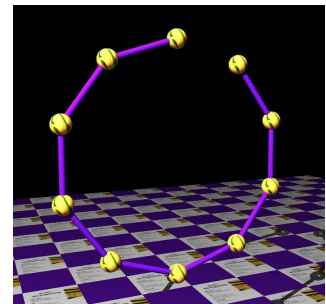
*Some of the effort for this homework assignment is in learning to use the various pieces of software, such as a text editor. Those less familiar with Linux software development procedures might seek out a more knowledgeable classmate to minimize frustration and wasted time.*

**Problem 0:** Follow the instruction 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 show a line of balls drop to the platform. *Promptly report any problems.*

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 `hw1.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:** Pressing `2` sets up scene 2, in which the balls form a pendulum. Modify scene 2 (look for code `ball_setup_2`) so that it starts with the balls arranged in a circle with the balls far enough apart so that the springs are stretched (see the screenshot on the right). Try to predict how the balls will move, then compare to what they actually do, using the space bar to advance frame by frame.



**Problem 2:** An obvious problem with the code is that the balls interpenetrate (overlap in space). Fix this by modify the program so that spring forces are added for interpenetrating balls. (This should not affect the existing springs.) The changes should go in the time step routine and will require adding a loop nest comparing every pair of balls. (Comparing every pair of objects for interpenetration is time consuming when the number of objects is larger than it is here. Later in the semester less costly methods for finding interpenetration will be described.)

- Only code in the `time_step_cpu` routine needs to be modified for this problem.
- The radius of a ball is in the `radius` member of the `Ball` class. For example, `balls[i].radius` is the radius of ball `i`.