LSU EE 4702-1 Homework 1 Solution Due: 12 September 2012

Most 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 1: 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.*

To check out the assignment use the command

Use the arrow keys, PageUp, and PageDown to move around the scene. Press lower-case 'b' and then use the same keys to move the fixed ball around. Press 'l' to move the light around and 'e' to move the eye. Look at the comments in the file `hw1.cc` for documentation on other keys. Try turning gravity on and off and changing the spring constant. *Note: There is nothing to turn in for this first problem.*

Problem 2: In the hw1.cc program the balls interpenetrate (overlap in space). Modify the program so that the spring forces prevent (or nearly prevent) interpenetration. Do this by making the relaxed distance of a spring connecting two balls equal to the sum of their radii. Also, the spring constant used for repulsion should be ten times that used for attraction.

- 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`.

The complete solution is in the repository in a file named hw1-sol.cc. An htmlized version is at `http://www.ece.lsu.edu/koppel/gpup/2012/hw1-sol.cc.html`. In the excerpt shown below the relaxed distance is set based on the neighboring balls radii, and the spring constant is set to one of two possible values, depending on whether the balls are overlapping.

```
// SOLUTION
const float distance_relaxed =
  ball->radius + neighbor_ball->radius;

// Compute how much the spring is stretched.
//
const float spring_stretch =
  ball_to_neighbor.magnitude - distance_relaxed;

// SOLUTION
const float spring_constant =
  spring_stretch < 0 ? 10 * opt_spring_constant : opt_spring_constant;
```