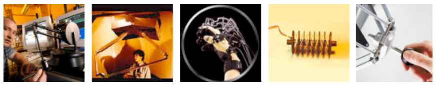
Spring 2014 CS277 - Experimental Haptics



Programming Assignment #4

Due Date: Thursday, May 8, 11:59 PM

You have probably noticed by now that all the objects in the virtual worlds you have rendered in the assignments so far have been static. Unfortunately, these stationary objects tend to make for somewhat uninteresting virtual worlds. In this assignment, you will implement mass-spring models and simulate Newton's laws to create dynamic objects that respond to your interaction in physically reaslistic ways.

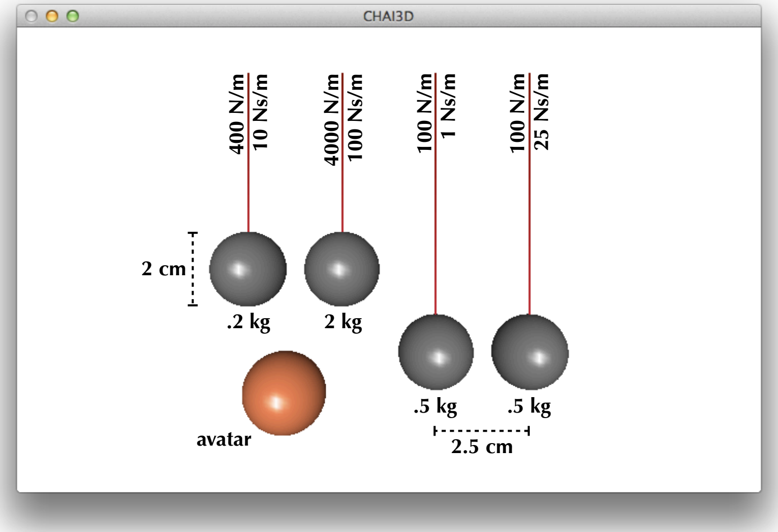
This assignment is worth a total of 15 points, with an additional possibility of earning a "bonus" designation. Since the bonus is a binary state, it will only be awarded to submissions that do an exemplary job of meeting the requirements.

Getting Started

Hopefully by completing the first three assignments you have now gained a certain level of comfort and familiarity with using the CHAI3D library. We believe you are ready to go out on your own now. Thus, we recommend that you start this assignment just as you did the first one: by making a fresh copy of the minimalist application located in the "templates" folder of the CHAI3D distribution.

Part I - Point Masses & Springs (6 points)

The goal of the first part of the assignment is to simulate the dynamic behavior of spheres (represented as point masses) suspended by spring-dampers with varying stiffnesses and damping coeffients. Create a scene containing four spheres, each of radius 1 cm, arranged in a row as shown below. Each sphere should be suspended by a spring with a natural length of 5 cm, meaning that the top end of the spring should be fixed in space, and the bottom end attached to the center of its corresponding sphere. Visually render the springs as lines, cylinders, or any other way you like.



Assign masses, stiffnesses, and damping coefficients to the system as shown in the diagram. Simulate the dynamic behavior of the spheres by computing forces, accelerations, velocities, and positions as described in class. Once you have assembled the simulation, apply a gravity force equal to *mg* in the downward direction, where $g = 9.81\ \mathrm{m/s^2}$, to the spheres. You should see the springs elongate under the weight of the spheres they carry. Add a little bit of general viscous or “air damping” equal to 1 Ns/m to the motion of the point masses to keep them from swaying too much.

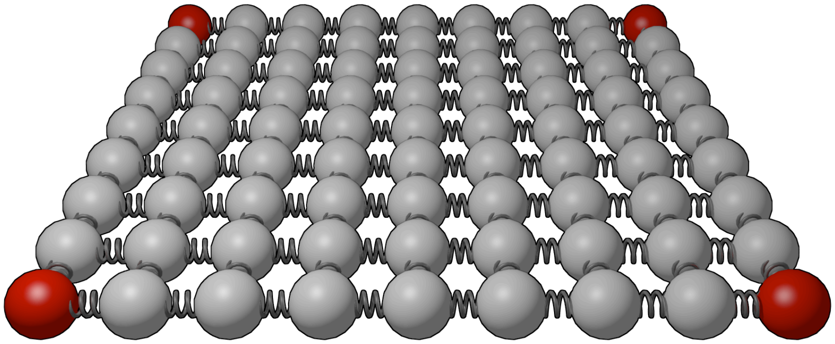
Finally, enable interaction with your scene by detecting collisions between your avatar and the sphere objects. Represent that avatar as another sphere of the same or slightly larger radius. When your avatar collides with a sphere in the scene, apply a force proportional to the depth of interpenetration to the device, and one of opposite direction equal in magnitude to the sphere object. Use a contact stiffness of 1000 N/m of interpenetration. The sphere objects themselves need not collide with each other, but you may make them collide if you think it makes for a more interesting scene.

Notes & Hints:

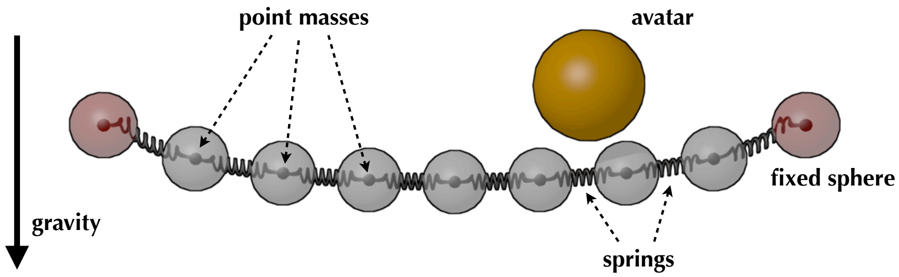
- When applying damping for the springs, ensure that the damping is computed only on the velocity component in the direction along the length of the spring. This damping is independent from the viscous “air damping” in the scene.

Part II - Deformable Models (6 points)

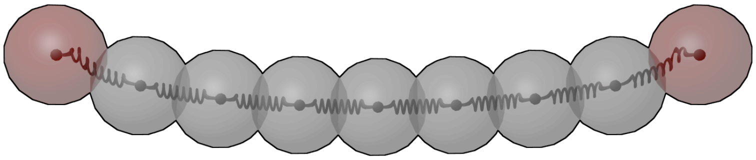
In this second part, you are to simulate the dynamic behavior of a deformable membrane modeled as a mass-spring network. Create a 9 × 9 grid of spheres and link them with springs as shown in the diagram below. Fix the positions of the four corner spheres (shown in red) in space, but allow the other 77 spheres to freely move about. As in Part I, model each sphere as a point mass with forces applied to it from the springs connecting it to its neighbors, and possibly from collision with the avatar.



Add gravity with $g = 9.81\ \mathrm{m/s^2}$ to this scene as well. Select appropriate masses and stiffness constants for the springs so that membrane exhibits a slight deformation in the middle with its own weight under gravity. Haptic interaction with the deformable membrane should be compelling and realistic. Add damping terms to the springs as needed to keep the simulation stable. A simulation that goes unstable very easily, is too difficult to move, or sags more than expected will not receive full credit.



Note that it is not necessary to compute collisions and interaction forces between the spheres which compose the membrane. In fact, you may want to increase the radii of the spheres so that they slightly overlap, as shown in the diagram below, so that it is more difficult for the avatar to pop through the membrane.



Notes & Hints:

- If you like, you may examine the implementation of GEL module in the CHAI3D library or similar dynamics simulations from other sources. However, the implementation of the deformable membrane simulation used here must be your own code. *Submitting code you did not write for this assignment is plagiarism, which is a violation of the Honor Code!* Please cite any external sources you referred to in a large capacity (but did not copy).

Bonus Part - Rigid Bodies & Stiff Deformable Models

There are two potential opportunities to earn a bonus in this assignment. The first is to extend your simulation in Part I to treat the spheres as rigid bodies, so that you can simulate the "tetherball" dynamics as shown in class. To do this, you will need to attach the springs to the tops of the spheres, rather than at the centers, so that they may impart moments on the balls. Make the simulation more realistic by using "strings", which only exert a force when elongated, and not when compressed, instead of springs. Apply a visual texture or pattern to the spheres so that we may see them spin.

In Part II, you may have noticed that no matter what values you choose for the physical properties of the deformable membrane, it still tends to sag under its own weight. If you try to compensate by making your springs very stiff, your simulation might even explode. The goal of this second bonus is to try to simulate the dynamics of a light, but stiff deformable material, such as a piece of cloth or wire mesh, rather than something that feels like Jell-O. You can do this by adding torsional or other structural springs that resist bending, or you may explore more sophisticated means of numeric integration that can handle stiff mass-spring models better than the basic implementation.

Assignment Questions (3 points)

These questions help you think about various aspects of implementing the assignment, and it may be helpful to read through and think about them before starting on your code. You may include a separate file in your submission with your answers, or simply append your answers to your "readme" file.

1. When you integrate accelerations to obtain velocities and positions for the point masses, you need to make use of a time interval, Δt. How do you obtain this time interval? Is it possible that this time interval may occasionally take on a larger-than-expected value, and if so, how would it affect your simulation?
2. In Part I, even though you created all four springs with the same natural length, the spheres on the left sit lower than those on the right. Why is that? The two spheres on the left have a very similar dynamic behavior, but you may notice that they have a very different feel when you interact with them. Briefly describe the difference you perceive between the two spheres on the left, and also the two on the right.
3. In Part II, what mass, stiffness, and damping coefficients did you use to build the deformable membrane? Briefly explain how you arrived at these values.

Submitting the Assignment

Submit the source code for any file(s) you modified or created. Please do not send us back the CHAI3D libraries or any other large compiled binaries! We will compile your program with a pristine copy of CHAI3D when we grade your assignment.

Include a "readme" file with your submission. Please indicate which platform and development environment you used to do the assignment, in case we run into any problems compiling or running your program. Also give brief instructions on how to use your software, and any gotchas we may encounter while grading it. You may also write in this document anything else you'd like us to know about your submission.

Completed assignments should be emailed as attachments to cs277.2014@gmail.com before 11:59 PM of the due date. Please indicate any late days used. If your files are abnormally large, you may send a web download link, drop off a copy on a flash drive, or make a suitable alternate arrangement.