

### Homework Assignment 1: Bouncing Ball

Due: Thursday, September 21

The goal of this assignment is to give you an introduction to the basic principles of developing a physically-based simulation, as well as to make sure everyone is familiar with basic programming to display simple data (e.g. OpenGL).

You are to create a simulation of a ball bouncing around in a box. Some of the characteristics of the simulation should be:

- Your box should have all six sides of the box (for simulation purposes). When drawing it, you should make some of the sides of the box transparent (i.e. don't draw them), so that you can see the ball bouncing. The key thing is that you should make sure your program makes it easy to see that the ball is bouncing correctly in the box (probably allowing for moving viewpoints and good lighting/shading, but as long as the motion is clear it is OK).
- You may use a spherical ball, to keep collision detection simpler.
- You should include gravity and air resistance
- You should allow for friction in the collision response
- Your simulation should clearly demonstrate bouncing in a variety of directions, including off of sides, and the top. You will probably need to allow for the user to set initial position and velocity in order to do this, so that all cases can be demonstrated.
- You should use basic Euler integration for this assignment, and *should allow the user to adjust the step size*. Note that adjusting the time step should **not** change the speed at which the simulation is displayed (i.e. use a timer to control that), assuming the step size ranges within reasonable limits, but rather just the way the equations are integrated. You should show that you are using a different step size, but that the simulation is still running at the same rate. If the step size becomes too small (such that it is impossible to compute all the steps within one frame), then your display rate should slow down.
- You should be able to have multiple different example cases run (e.g. by allowing the user to specify starting conditions, or having random starting conditions), in the same framework.

As a reminder, to receive more than a "B" level of credit for the assignment, you will need to enhance the simulation from the basic requirements. Some examples of this might include:

- Adding additional forces (e.g. wind, or user-defined ones – the more complex, the more credit). For example, a simple constant wind would be a small grade increase, while allowing mouse motion to generate temporary wind forces or a complex wind calculation, or reading in wind data from a file, or having a spatially varying wind, etc. might be more.
- Using a more complex container than just a regular box. Note that this will require more advanced collision detection.

- Including multiple balls that bounce off of each other. We have not discussed object-object interaction yet (and won't for a while), but a simple collision model can be used.
- Allowing the user to easily adjust parameters (such as mass, gravitational and air resistance constants, coefficients of friction/restitution, time step size, box dimensions, starting conditions, etc.), possibly in an interactive interface. The interface quality would affect how much bonus there would be.

You should be prepared to demonstrate your program in class on the day it is due, and will need to turn in the source code online to ecampus.

Remember that you need to clearly acknowledge any outside libraries or software that you use.

Regarding cooperation on this assignment, it is fine to use and discuss with others as much as you'd like aspects such as how to get OpenGL (or some other rendering system) working, how to handle basic rendering of the scene, etc. I will try to provide a sample rendering file for you, also. If you want to use a library that provides vector math, point/vector classes, or a user interface library to give the users a nice GUI, etc. you are welcome to do so. The parts that I want you to implement individually (i.e. you can talk with others in very broad terms, but you should avoid looking at others' code here) include the core parts of the assignment, such as:

- The overall simulation loop, including Euler integration
- Basic motion under forces
- The collision detection and response
- Changing step size

You should try to implement those parts on your own. Your "bonus" portion (above a "B" grade) will be based on things you implement. Incorporating a very nice interface does take additional work, but you need to make it clear what you did vs. what was done for you.