



## Programming Assignment #1

**Due Date: Tuesday, April 15, 11:59 PM**

The purpose of this homework assignment is to familiarize yourself with programming a haptic device using the CHAI3D software libraries. You will be using force fields to render basic touchable shapes and other effects that you can feel. This assignment has three parts, and your job is to create a different scene for each part.

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

You will need to use a Novint Falcon or an equally capable haptic device to complete this assignment. If you are a registered student in the course, and have been assigned a haptic device, be sure to pick one up during one of the scheduled distribution times.

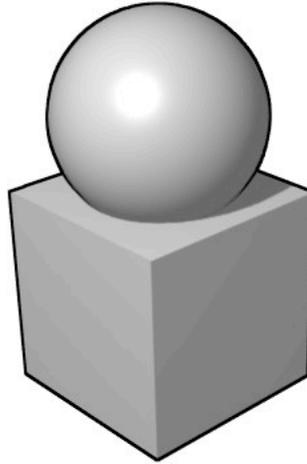
If you are using Microsoft Windows, you will first need to install the drivers for the Novint Falcon provided on the course web site. *Do not use the drivers provided by Novint!* No drivers are needed for the other platforms.

Download the CHAI3D library from the course web site (Assignments page) for the platform of your choice. Unzip the files to your favorite projects directory, open the project file from your development environment, and then hit build to compile the CHAI3D library and example programs. After a successful build, plug in your haptic device and run some of the example programs to verify that everything is working.

You may start this assignment by copying and renaming the template file located in the “templates/application” folder of the CHAI3D package. We also suggest that you explore the source files of the example programs in the package. They demonstrate the basic capabilities of the library, and provide good starting points for the different assignments you will complete in this course.

### Part I - Shapes Using Force Fields (4 points)

Use the force field algorithm discussed in lecture to render two touchable virtual objects: a sphere and a box. Each shape should be described as a force rendered to the haptic device as a function of position when it intersects the object. Position the two objects so that they slightly overlap, but remain roughly centered and entirely within the workspace of the haptic device, as shown in the diagram below. You should visually render the two objects as well – do this any way you’d like, so long as it correlates reasonably well with what is haptically displayed.



#### Notes & Hints:

- ▶ The class `cMesh` may provide an easy means of visually rendering a sphere. Look for the functions `cCreateSphere()` and `cCreateBox()`, and see how they are used in conjunction with `cMesh` in other examples.
- ▶ Make sure you are rendering your shapes using force fields. Many of the CHAI3D examples render similar objects using the god-object algorithm, which will be discussed later in the course.
- ▶ Sharp changes in force over a small amount of space can drive your device unstable – be careful! Choose a stiffness for your force fields that reasonably balances pushing the device out of the objects and stability of the device.

### **Part II - Inverting the Force Fields** (4 points)

If you change the direction of the force fields so that they pull instead of push, you can create an interesting effect that acts like a magnet. Create a second scene, this time with two magnetic objects: a magnetic point and a magnetic line. Use a force field that attracts the haptic device toward the object when it passes within a certain distance. Make the objects exhibit a stronger pull as you get closer, like a real magnet. Again, place both objects within the workspace of the haptic device, and visually render the two objects any way you'd like.

#### Notes & Hints:

- ▶ Experiment with different force field functions – a nonlinear field may produce a better magnetic effect than a linear function.
- ▶ Be very careful with controlling the forces on this one! The center of your object may be a singularity in your field, and cause your forces to spike or rapidly flip directions. Your simulation must be reasonably stable to receive full credit.

- ▶ The class `cShapeLine` can be used to visually render the magnetic line. However, you may get a better 3D visual effect by rendering it as a cylinder. The magnetic point can be rendered as a small sphere.

### Part III - Animating the Objects (4 points)

Create a third scene by taking the two objects from Part II (or another object you like) and making them move around the workspace. You can make the objects move any way you like – randomly, chasing the avatar, or otherwise. The objects should continue to exert appropriate forces on the device as they move. Experiment with changing the size, shape, and other properties of the objects over time as well. Full credit will be given to a program that animates the motion and one other feature of the objects in an interesting manner.

Your final program for this assignment should have a means for switching between the three scenes described in parts I, II, and III. You can use the keyboard, mouse, or other buttons on the Falcon to change scenes. If you are unable to achieve this, you may submit three separate programs – one for each part of the assignment – instead.

### Bonus Part - Cat & Mouse Game

In the spirit of warming up for the course project, turn the scene from Part III into a simple haptic “cat-and-mouse” game, where the goal is to avoid getting “stuck” by one of the objects moving around while keeping your avatar within a spherical arena. The game ends when the avatar gets sucked into one of the traps, or is moved outside the arena. Keep score by timing how long the player manages to avoid the traps while staying within the sphere, and use one of the buttons on the Falcon to reset/restart the game. The bonus will be awarded to the submission(s) that demonstrate the best balance of challenge, fun, and other elements in the gameplay.

Notes & Hints:

- ▶ A transparent or wireframe sphere may be a reasonable way of visually rendering the extent of the arena. You may still find it difficult to perceive the 3D position of the device relative to the workspace. Try changing the color of the avatar as it gets closer or farther from the edges of the arena.

### 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. What force field functions did you use to create the shapes in Parts I and II? Briefly describe how or why you chose these functions.
2. In Part I, observe what happens when you try to move the device between the sphere and the box, and comment on what you feel. Is this effect desirable?

3. In Part III, the haptics loop may run at a different rate on different machines, but you probably want your objects to move at the same real-world speed irrespective of the speed of the computer. Is there currently something in the template application that regulates the speed of the haptic feedback loop? What means can you take or have you taken to address this computer speed issue? (Hint: Try commenting out the code that reads position and/or sends forces to the haptic device, and see if it has any effect on the haptic rate of the program.)

### **Submitting the Assignment**

You should normally just 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](mailto: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.