Assignment #4

Point Value: 120 points

Due: Thur., Dec 18th, 11:59pm

CS 184: Foundations of Computer Graphics
Fall 2014
Prof. James O’Brien

For this assignment you will write a basic Inverse Kinematics solver.

1. **This assignment is due as indicted above. There is no slack time between when the assignment is due and when the professor must submit grades to the university. *Assignments more that 24 hours late will receive a zero. No exceptions can be made for any reason!***

2. **This assignment should be done alone or in pairs. You may share ideas with other groups, but you may not share code.**

3. **You may develop on Unix, OS X, or Windows. The platform you use will be the one used to grade assignments. Keep in mind that there are slight variations due to OS versions, different libraries, and other factors, so you should verify that your code runs on the instructional machines appropriate for you platform choice.**

4. **We will be using the submit software for submission of this assignment. Instructions for using the submission software are here. You should include a README file that at the minimum contains the following information:**

   - Your (and your partner's) name
   - The platform your code runs on
   - The location of your source code (i.e. indicate who in your group has done the submission, and on what platform). Only one of the people in your group should submit the actual code. The other people should only submit the README file.

   All files needed to compile your code should appear in the submitted directory. It is your responsibility to make sure that they will compile and run properly.

   - Windows: The grader should be able to recompile your program by simply opening the project and rebuilding it from scratch.
   - Unix and OS X: The grader should be able to recompile your program simply by typing "make".

5. **Once you have your assignment working, you must also update your class web page to include an “Assignment 4” link to a page that documents your project. The requirements for this documentation are described below. This documentation of your work will be used to determine your grade so please follow instructions carefully.**

6. **If you work in a group, you should all link to the same web page and the web page should list your group members.**

7. **Grading will include a few points for aesthetics and creativity as demonstrated on your web page and video.**

8. ***Do not wait until the last minute to start this assignment.*** **As you may have learned with your raytracer, this assignment can go very smoothly if you start early and plan ahead. If you wait until the last minute to start, then this can be a very frustrating assignment.**

   Check the news group regularly for updates on the assignment or other clarification. We will assume that anything posted there is henceforth known to all.

9. **Questions should be posted to the discussion group or emailed to cs184.**

10. **Your program should implement an algorithm for solving inverse kinematics problems. At the very minimum you should demonstrate:**

    - Solving an IK problem on an arm that consists of at least four links that are connected in a serial chain by ball joints.
    - The lengths of the segments must not all be the same: the longest segment must be at least twice the length of the shortest.
    - The root link should be attached to an immobile base by a ball joint.
    - The goal is to position the end of the arm at a specified goal point **in 3D**.
    - The solver should be robust to out-of-reach goals.
    - You may use the algorithm described in class or any other algorithm that produces good results.

11. **To demonstrate your system, you should generate an animation that shows the arm tracking a moving goal.**

    - The goal should move over a path that takes it out of reach of the arm, takes it through the location of the base-joint, and that also moves over a non-trivial region of the arm's operating space.
    - The path should not be just a straight line. It must be at least a circle or figure-eight. Your demonstration should fully exercise the arm over its range of motion. Please see the example videos that were shown in class.
    - Your video should *clearly* demonstrate that your system works. If the grader cannot see what is happening then you will lose points.
    - Submitting a video that was generated using IK code that you did not write will be considered cheating. However, it is ok for you to use other software to render your images, encode the video, and do other things that are not part of the IK solver itself. If you are unsure about what it permitted, ask one of the instructional staff.

12. **Your web page should contain at least one demonstration video (as described above) and a full description of what you implemented.**

13. There are some things you can do for extra credit:

- Demonstrate your solver working with multiple different arms. For example, you can vary the length and number of segments. If you do multiple examples, then only the first example is subject to the requirements concerning the length and number of the links.
- Demonstrate the solver working on a system with one or more bifurcations. (*i.e.* a tree rather than just a chain)
- Demonstrate multiple constraints being solved for simultaneously.
- Demonstrate a system that tries to keep its center of mass over it’s base.
- Demonstrate joint limits.
- Demonstrate other joints besides ball joints.

Anything you do for extra credit must be described in your description and demonstrated in your video(s).