Programming Assignment #3 (20 marks)
Due on: Monday, April 8, at 23:59

Plagiarism is a serious academic offense: passing off someone else's work or ideas as your own in order to get a higher mark. Plagiarism is treated very seriously. The assignment you hand in must represent your own work. Submitting source code downloaded/copied from the WWW or your classmates' solutions as your own is deemed cheating, and an F grade will be awarded. However, reading and studying code from the web and discussing ideas with your classmates is allowed, but you must acknowledge their help, and still do your own work. In the README file you submit with your source code, you must list all the open source code that you have studied, and all the people's names with whom you have discussed your assignment.

Late Policy: Late submissions are possible, but they will be penalized. One day late: 10% penalty; Two days late: 20% penalty; Three days late: 40% penalty; Four days late: 60% penalty; Five days late: 80% penalty; Six or more days late: 100% penalty.

Problem 1 (8 marks): Controlling a robotic arm

You will write a program called myRobot to display and control a 3D robotic arm to reach a desired location in the XY plane. The robot is composed of three parts: the base, a lower arm, and an upper arm. Please refer to Sec. 9.3 in the textbook for the specification of joint angles. No collision detection needs to be implemented in this assignment. Do not use any other third party libraries other than the Common folder.

You have three tasks to complete:

- **Recursive Rendering (2 marks):** Render your robot model recursively as described in Sec. 9.5. The robot base should be located at the center of your window. The whole robot arm should be within your view for any possible robot movement.

- **Forward Kinematics (2 marks):** Use three sliders to interactively control the rotation of the three joint angles.

- **Inverse Kinematics (2 marks):** For this step, please fix the base rotation to zero and only update the remaining two arm joint angles. Allow the user to click somewhere on the screen to identify a desired target in the XY plane (drawn as a small red ball). The tip of your robot should then touch the ball. You may assume that the target will be given at a location that is reachable by the robot.
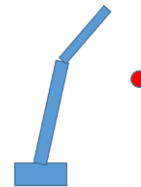
- **Smooth Animation (2 marks):** Animate the robot arm so that the reaching motion is smooth with a reasonable speed.

Problem 2 (12 marks): Create an interesting static scene or dynamic animation

You are to create and render an interesting scene called myScene. The scene can be either static or animated. We will judge your work by its *creativity*, *sophistication*, and/or *realism*. Note that you do not have to achieve all these attributes. Apply some of the techniques that you have learned in Chapters 6-12 and let your imagination fly. If you can WOW us then you will get full marks. Do not use any other third party libraries other than the Common folder and three.js. You don't have to use three.js. We actually expect more sophistication and realism from your scene if you do use three.js.

Submission: Please submit a zip file with student number and your name (i.e., `300000001_TerryFox.zip`). The zip file contains all the source files and a rendered image in jpeg format .jpg that you are proud of from your problem 2 solution. Please also write a README acknowledge any help you have received and any discussion you have participated; document any steps not completed for problem 1; describe techniques used and cool features in your problem 2 solution; describe how to interact with your program if your problem 2 solution contains interactions and animations; and write any extra instructions for the TA to mark your assignment.