Objective: Experiment with lights, shadows, textures, user interaction and other OpenGL features.

This final project must be completed only if you decide not to submit an individual proposal. You have to start from your own implementation of the HW#4 (Robot Arm simulation) and add features as described. The HW#4 that you extend must be the one you submitted. The Baseline features will be valued up to 80% of the full credit. Implementing the extended features may increase the grade of your project up to 100%, depending on how many you add, how complex is your implementation, how general are your assumptions, etc. The extended features are given as a list of suggestions. Feel free to add whatever you like, as soon as you document and explain it, you will receive some credit for it. Implementing these features can be very complex. This will be reflected by your grade.

Baseline (Up to 80%):

1. Use the robot to solve automatically the Hanoi Towers puzzle. Solving the problem for three discs and three towers is sufficient. You may assume the posts are in well known positions and the discs have known sizes. Check the literature for the algorithm. It is very simple (5/10 lines of code).
2. Add lights and materials to the objects in the scene.
3. Allow the user to change the position and the color of the light source, define your own interface.

Extended (Up to 100%):

1. Add shadows to the objects in the scene.
2. Change the shape of the discs, increase their number, make them transparent, use textures instead of colors, etc...
3. Include in the scene objects that the robot must avoid while moving.
4. Change the scene into a maze, with walls that the robot must avoid.
5. Replace the flat surface on which the robot moves with something more interesting.
6. Add another robot to the scene, program them so that they work together to solve the problem, while avoiding each other.
7. Change the robot joints into a “desk lamp” style, change the gripper into a three-fingers gripper.
8. Add wheels or track-drive to the robot, move it like a vehicle (steer left or right, move forward and backward).
9. Use the robot to solve some other puzzle.
10. Add a joint so that the gripper can operate in other directions, add walls and manipulate objects on the walls (for example, solve the Hanoi tower with the three posts being on three different planes).

11. …

**Submission:**
- Submit the C program both on paper (in my mailbox) and by email (cs155@cs.brandeis.edu).
- Write a few pages with a detailed description of the features you implemented. Mention assumptions and restrictions. Do the same for all known bugs (it is normal to have some). Comment on possible improvements.
- Comment the program with your name and email. Indicate on which platform has been developed (Windows, Mac OS, Linux, etc.).
- Before submitting, test the program on one of the computers in the Berry Patch and make sure it works properly on Linux. State otherwise.