

CS 537/CPE 537: Homework Assignment 4
Due: November 25, 6:15pm

Philippos Mordohai
Department of Computer Science
Stevens Institute of Technology
Philippos.Mordohai@stevens.edu

Requirements. For each programming assignment you should submit a zip file containing the code and a brief report stating: what you did; how you did it; any particular features you want to draw attention to; or any problems with the program you know about. Submit the zip file by email to me with a subject line like “CS537: Homework 4”.

Make sure your programs compile and run on a generic PC with Windows XP with VC++ and glut installed. You run the risk of getting no credit if your program does not compile. Prefer simple portable code to alternative fancy solutions.

Collaboration Policy. Homeworks will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in understanding the material but not in solving the problems. Use of the Internet is allowed, but should not include searching for previous solutions or answers to the specific questions of the assignment. I will assume that you will be taking the responsibility of making sure that you personally understand the solution to any work arising from collaboration.

Late Policy. The penalty for late submission is 20% of the grade per day, enforced at 6:15 each day after the due date. If urgent or unusual circumstances prohibit you from submitting a homework assignment in time, please send me an e-mail explaining the situation.

Problem 1.

Write a program that constructs an indoor environment in which the viewer, who is identified with a flying camera, navigates. The requirements are as follows:

1. There should be at least one room completely enclosed by four walls, a roof and a floor. **(10%)**
2. There should be more than one window and all windows have to be identical. The windows do not have to show the room’s exterior. They can be painted by a fixed color or pattern. **(10%)**
3. There should be at least one poster on the walls. **(10%)**

4. There should be a crystal ball in the room onto which the entire scene, except the invisible camera-viewer, is reflected. If cube mapping is not supported by your graphics card driver, the ball can be metal or of any other highly specular material. **(10%)**
5. The viewer should be able to rotate and translate freely in the scene without colliding with other objects. An attenuated spot light should be attached to the camera. Think of it as a head-mounted flashlight which points in the direction in which the user is looking. **(20%)**
6. Try to use good style both in modeling and programming. **(20%)**
7. Write a brief report explaining what you did, emphasizing aspects that are not obvious from running or looking at the code. If something does not work, give possible explanations. Make sure to **include the scene description** (tree or scene graph). Also include instructions on how to navigate. Which keys to press to rotate, move forward etc. **(20%)**

Hints:

- You can reuse parts of the navigation code from Homework 2. You can also implement a viewer that rotates and moves only forward and backward.
- Do not worry about collision detection. The viewer can fly through the walls to nothingness. Just make sure that motions and velocities are small enough (or the environment large enough) to allow exploration.
- Using additional lights other than the head-mounted flashlight is optional. Do something that looks nice.
- A door and furniture are not necessary, but may be helpful for the final project.
- Implement the crystal ball last. It is only worth 10% of the total credit.

Resources:

- See code examples in Lecture XI.
- **Material properties:** see `materials.h` in <http://www.xmission.com/~nate/tutors/tutors-src.zip> on Nate Robins' web page at <http://www.xmission.com/~nate/tutors/>.
- Or Frederic Devernay's page at <http://devernay.free.fr/cours/opengl/materials.html>, or a copy of Chuck Hansen's list at http://www.cs.utk.edu/~kuck/materials_ogl.htm, or the examples on [opengl.org](http://www.opengl.org) <http://www.opengl.org/resources/code/samples/sig99/advanced99/notes/node153.html>.