

CSC 3260 Principles of Computer Graphics

Assignment One: Creating a 3D Room (10%)

Due Time: 5:00pm, Feb 5 (Fri), 2010

Late penalty: 10% per day.

Fail the course if you copy

I. Introduction

This first programming assignment will introduce you to the OpenGL graphics programming interface. In this programming assignment, you will be creating different 3D objects to model interesting shapes. The objective of this assignment is to apply your understanding of the computer graphics theories and give you an introduction to the OpenGL programming library.

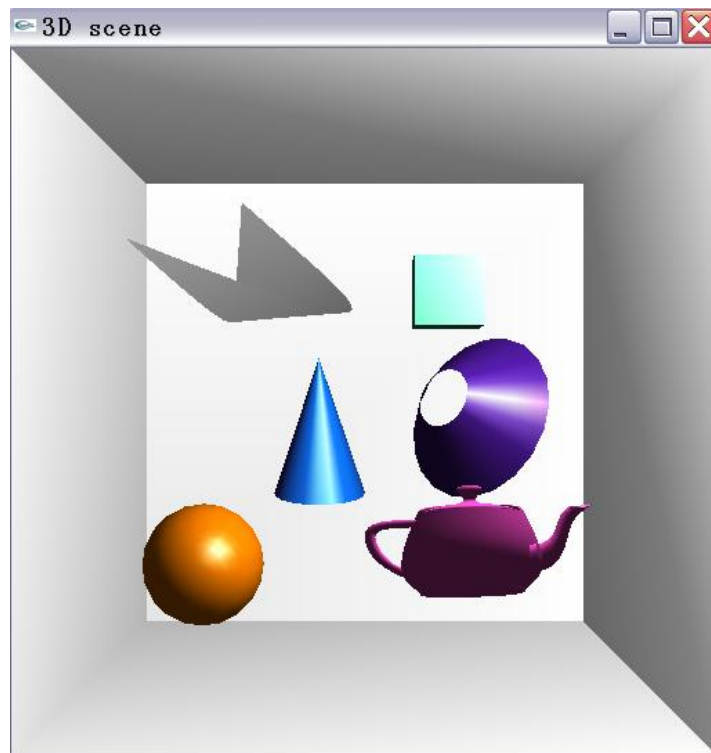


Fig. 1 The scene drawn by the demo program

Your goal is to design a 3D room, which consists of five planes, i.e., the left, right, and back walls, the ceiling and the floor, at least five different geometric primitives, such as the cone, sphere, cylinder, cube, etc. And you should be able to apply arbitrary transformations — translation, rotation, scaling — to them. The user should be able to use the keyboard (and/or the mouse) to translate, rotate, and scale the object. After you have implemented all the basic functionality, we expect you to construct an interesting scene with your program. In order to make your scene more realistic, you may use the perspective projection instead of orthographic projection. You can also use different primitives to construct a complex object in the room (furniture or electrical equipment).

II. Implementation Details

In this assignment package, we have provided you with a template program (i.e., *submit.cpp*) that includes the necessary functions you are going to use and callback functions in the GLUT interface toolkit. Use this template as the basis for your implementation. There is also a file (i.e., *readme.txt*) indicating the keyboard usage of the demo program (i.e., *demo.exe*) for the users. You may design your own function to process the keyboard events, but you should also submit a file like this to specify the keyboard (and/or mouse) events you designed in your program. Otherwise, the mark for related items will be deducted.

All programs should meet reasonable programming standards: header comment, in-line comments, good modularity, clear printout, efficiency.

Constraints:

1. Draw at least five geometric primitives in the 3D space bounded by the five planes;
2. Ensure the objects are in good lighting condition;
3. Create at least 5 keyboard or mouse events;
4. Design diverse objects transformations, such as rotation, translating, scaling;
5. Use perspective projection to draw the scene;
6. Set interesting material properties to different objects.

Non-constraints

You are free to add objects, move them, organize them, deal with their material attributes, and whatever you wish to make your scene interesting.

III. Grading Scheme

Your assignment will be graded by the following marking scheme:

Basic (80%)

I	Planes (the left, right, and back walls, the ceiling and the floor)	10%
I	At least five different geometric primitives	20%
I	At least 5 keyboard events (mouse event is optional)	15%
I	Object transformation animation (rotation, translating, scaling)	15%
I	General lighting control (different material properties setting)	20%

Bonus (up to 20%)

I	Well organized room	5%
I	Complex meaningful object constructed by different primitives	5%
I	Additional light (with different properties, on/off or transformation)	10%
I	Other creativities	10%

Total	100%
-------	-------------

Note: no grade will be given if the program is incomplete.

IV. Guidelines to submit programming assignments

- 1) You are suggested to write your programs on Windows, since there will be enough technical support. If you developed the program in other platforms, make sure your program can be compiled and executed on Windows as the program will only be tested on this platform.
- 2) Modify the provided *submit.cpp*, and provide all your code in this file. No more additional .cpp or .h files are allowed. Type your full name and student ID in *submit.cpp*. ***Missing such essential information will lead to mark deduction.***
- 3) Zip the source code file (i.e. *submit.cpp*), the executable file (i.e., *submit.exe*), and the readme file (i.e., *readme.txt*) in a .zip or .rar file. Name it with your own username (e.g. wkchan.zip). That is, there should be exactly **three** files in your submitted package.
- 4) Unencode the zip file and mail it to graphics account with the subject "*ps1 wkchan*". *ps1* stands for problem set 1. *wkchan* is your username. Keyword *ps1* is critical for the mail filter to work correctly. For example, type (on UNIX)

```
uuencode wkchan.zip wkchan.zip | elm -s "ps1 wkchan" graphics@cse.cuhk.edu.hk
```
- 5) Note that you have to type the "wkchan.zip" twice in the uuencode command.
- 6) An acknowledgement email will be sent to you once your assignment is received. The acknowledgement email will contain what you have submitted. Make sure it included your uuencoded zip file. Otherwise, resubmit your assignment since you have submitted a null email.
- 7) In case of multiple submissions, only the latest one will be considered.
- 8) ***Fail the course if you copy.***