CSCI 3260 Principles of Computer Graphics

Assignment One: Creating a 3D World

Due Time: 5:00pm, Oct. 12 (Mon), 2015

Late penalty: 10% per day.

Fail the course if you copy

I. Introduction

This first programming assignment will introduce you to 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 get you familiar with OpenGL programming.

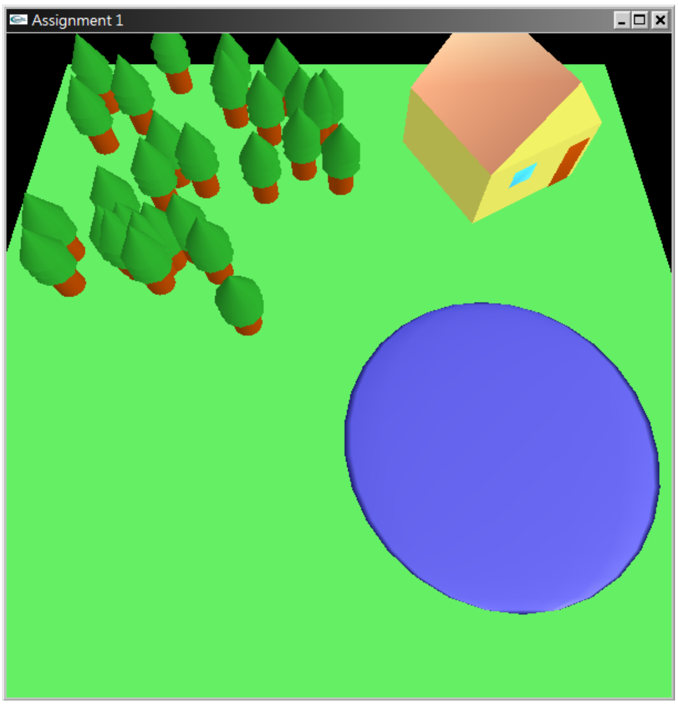

Fig. 1 The scene drawn by the demo program

Your goal is to design a 3D world, which consists of a ground (the green flat surface), and at least three complex objects constructed by different geometric primitives, such as the cone, sphere, cylinder, cube, etc. You should be able to apply arbitrary transformations (translation, rotation, scaling) to them and organize them properly. Your application should allow users to use the mouse and the keyboard to navigate around the virtual 3D world and the keyboard to translate, rotate, and scale the objects. In addition, some basic animations are required to be implemented in your application. Rather than only implementing the basic functionalities, we expect you to construct an interesting scene or animation with your program. You are recommended to use primitives to construct different complex objects. Your 3D world can be any form and shall not be limited by the demo program shown in Fig. 1.

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 a ground and at least three complex objects formed by geometric primitives;
2. Ensure the objects are in good lighting condition;
3. Create at least 3 keyboard events
4. Design diverse object transformations, such as rotation, translating, scaling;
5. Set up 4 mouse navigation zone;
6. World navigation using the keyboard (e.g. ‘→’， ’←’,’ ↑ ’,’ ↓ ’)
7. 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.

World Navigation with the mouse

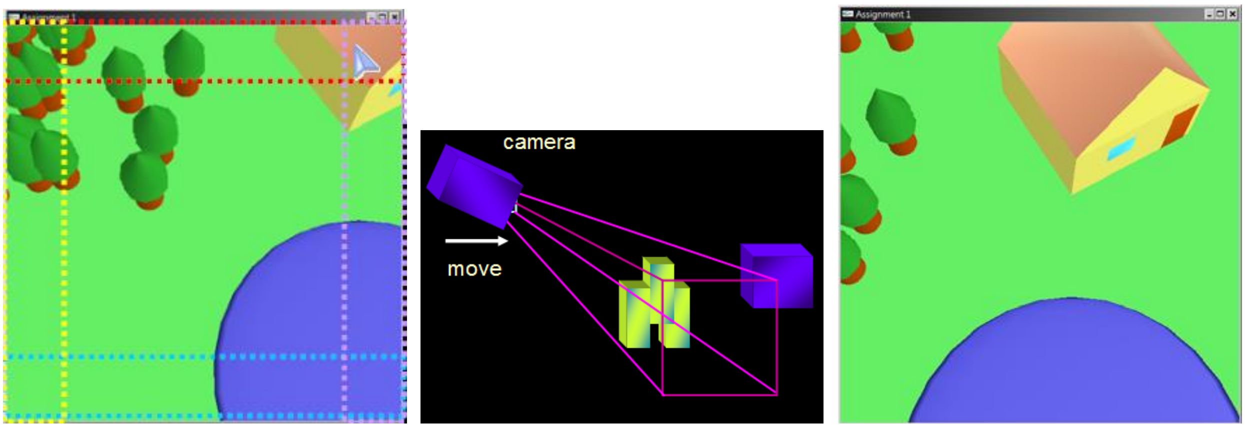

Fig. 2 Illustration of mouse navigation (dotted area), camera movement and the corresponding shifted view

To navigate around the virtual world, we can use the mouse position information as a camera controller. GLUT toolkit would receive a passive call whenever mouse moves. You need to analysis the mouse position and arrange suitable camera movement. Remember that camera movement should be bounded and gentle, duration of 2 or 3 seconds, traveling from one end to the other, is suitable.

World Navigation with the keyboard

You are also required to use the keyboard to navigator the virtual world, for example, when the key '→' is pressed, the camera moves right.

III. Grading Scheme

Your assignment will be graded by the following marking scheme:

Basic (70%)	
● Ground	5%
● Ensure the objects are in good lighting condition	5%
● Al least 3 keyboard events	10%
● Object transformation animation (rotation, translating, scaling)	10%
● Different material properties (color, diffuse, specular)	10%
● 4 mouse navigation zone (bounded and gentle)	15%
● World navigation with the keyboard	15%
Additional tasks (up to 30%)	
● Well organized 3D world	10%
● Complex meaningful objects constructed by different primitives	10%
● Additional light (with different properties, on/off or transformation)	10%
● Collaborative animations involving 2 or more objects	10%
● 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) Submit your assignment in one of the following two ways:

   1. For students who have unix account:

Unencode the zip file and mail it to csci3260 account with the subject "asg1 wkchan". asg1 stands for assignment 1. wkchan is your username. Keyword asg1 is critical for the mail filter to work correctly. For example, type (on UNIX)

uuencode wkchan.zip wkchan.zip | elm -s "asg1 wkchan" csci3260@cse.cuhk.edu.hk

Note that you have to type the "wkchan.zip" twice in the uuencode command.

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.

2. *For students who do not have unix account:*

   Please send your assignments to the tutor's email address:

   (*lzhu@cse.cuhk.edu.hk*)

5) In case of multiple submissions, only the latest one will be considered.

6) ***Fail the course if you copy.***