CSCI 3260 Principles of Computer Graphics

Assignment One: Creating an interactive 3D Scene

Due Time: 11:59pm, Oct 05 2017 (Friday)

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 and programmable pipeline. In this programming assignment, you need to create a 3D scene with user interaction (see the good examples in Fig. 1). The objective of this assignment is to apply your understanding of the computer graphics basic concepts; get familiar with the OpenGL programming library and give you an introduction to the programmable pipeline.
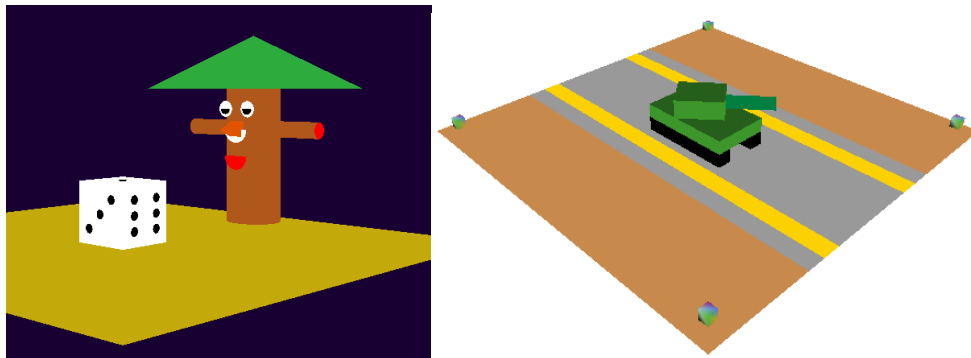

Fig. 1 Good examples of 3D scene.

Your goal is to design a 3D scene with user interaction. Specifically, in your scene, there must have **points**, **lines**, **2D objects** and **3D objects** (see Fig. 2 as an example), and you should be able to apply arbitrary transformations including **translation**, **rotation**, and **scaling** to them. The user should be able to use the keyboard (and/or the mouse) to translate, rotate, and scale the object. The object color, window size, window title and window background color are all up to you. In order to make your scene more realistic, you should use the **perspective projection** instead of orthographic projection. You are recommended to draw objects with **indexing**. Your 3D scene shall not be limited by the demo program.
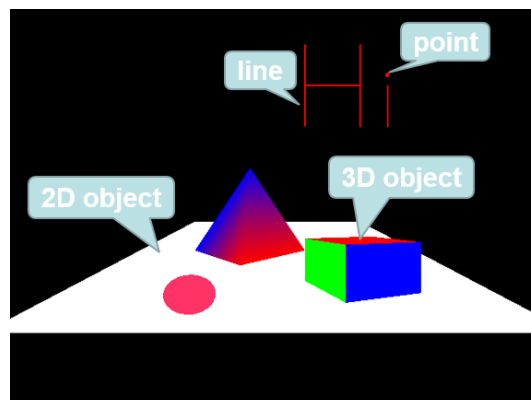

Fig. 2 Basic requirements of assignment 1.

II. Implementation Details

In this assignment package, we have provided you with two shader programs (i.e., *VertexShaderCode.glsl & FragmentShaderCode.glsl*) and a template program (i.e., *main.cpp*) which include 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. You need to design your own function to process the keyboard events, and you should also submit a file like *readme.txt* to specify the keyboard (and/or mouse) events you design 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, and efficiency.

Basic Requirements:

1. OpenGL code should be written using programmable pipeline instead of fixed pipeline with OpenGL 2.0+;

2. Draw at least a point, a line, a 2D object and a 3D object;

3. Ensure at least one object is drawn with indexing;

4. Create at least three kinds of keyboard and/or mouse events, such as rotation, translation and scaling;

5. Use perspective projection to draw the scene;

6. Enable depth test to realize occlusion;

Additional self-design requirements:

You are free to add objects, move them, organize them, and whatever you wish to make your scene interesting.

III. Grading Scheme

Your assignment will be graded by the following marking scheme:

Basic (80%) (e.g., Fig. 2)

Draw points, lines, 2D objects and 3D objects	25%
At least one object is drawn with indexing	10%
At least three kinds of keyboard (and/or mouse) events	15%
Include three kinds of object transformations (rotation, translation, scaling)	15%
Perspective projection	10%
Depth test	5%
Advanced (20%) (e.g., Fig.1(right))	
Complex and meaningful objects constructed by different primitives	10%
Interesting and creative interactions	10%
Total:	**100%**

Note: no grade will be given if the program is incomplete or fails compilation or using fixed pipeline.

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, you may have to show your demo to tutor in person if your codes cannot be compiled successfully under Windows system.

2) Modify the provided *main.cpp & VertexShaderCode.glsl & FragmentShaderCode.glsl*, and provide all your code in this file. It is not recommended to create or use other additional .cpp or .h files. Type your full name and student ID in *main.cpp*. ***Missing such essential information will lead to mark deduction (up to 10 points).***

3) We only accept OpenGL code written in programmable pipeline. No points will be given if your solution is written in fixed pipeline.

4) Zip the source code file (i.e. *main.cpp & VertexShaderCode.glsl & FragmentShaderCode.glsl*), the executable file (i.e., *openGL.exe*), and the readme file (i.e., *readme.txt*) in a .zip. Name it with your own student id (e.g. *1155012345.zip*).

5) Submit your assignment via eLearn Blackboard. (https://blackboard.cuhk.edu.hk)

6) Please submit your assignment before 11:59 p.m. of the due date. Late submission will be penalized by 10% deduction per day.

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

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