Computer Graphics, Lab Assignment 8

Handed out: May 6, 2020

Due: 23:59, May 6, 2020 (NO SCORE for late submissions!)

- *Only accept answers submitted via git push to this course project for you at https://hconnect.hanyang.ac.kr (<Year>_<Course no.>_<Class code>/<Year>_<Course no.>_<Student ID>.git).*
- *Place your files under the directory structure <Assignment name>/<Problem no.>/<your file> just like the following example.*

```
+ 2020_ITE0000_2019000001
  + LabAssignment2/
    + 1/
      - 1.py
    + 2/
      - 2.py
    + 3/
      - 3.py
```

- *The submission time is determined not when the commit is made but when the git push is made.*

1. Write a program that draws a color-changing cube.

   A. Set the window title to **your student ID** and the window size to (480,480).

   B. Start from the code in 7-Lighting&Shading slides. Draw a flat-shaded cube. Make sure camera manipulation shortcuts '1', '3', '2', 'w' work.

   C. Use the following light setting:

```
lightPos = (3.,4.,5.,1.)
glLightfv(GL_LIGHT0, GL_POSITION, lightPos)

ambientLightColor = (.1,.1,.1,1.)
glLightfv(GL_LIGHT0, GL_AMBIENT, ambientLightColor)

specularObjectColor = (1.,1.,1.,1.)
glMaterialfv(GL_FRONT, GL_SPECULAR, specularObjectColor)

glMaterialfv(GL_FRONT, GL_SHININESS, 10)
```

D. If you press or repeat a key, the diffuse & specular color of the light and the ambient & diffuse color of the object should be changed as shown in the Table:

Key	Action
A	Change the light color to red
S	Change the light color to green
D	Change the light color to blue
F	Change the light color to white
Z	Change the object color to red
X	Change the object color to green
C	Change the object color to blue
V	Change the object color to white

A. Files to submit: A Python source file (Name the file whatever you want (in English). Extension should be .py)

2. Write a program that draws a smooth-shaded cube.

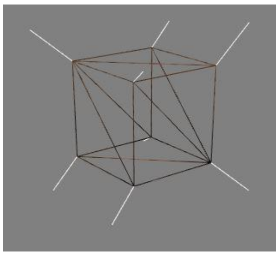
   A. Set the window title to **your student ID** and the window size to (480,480).

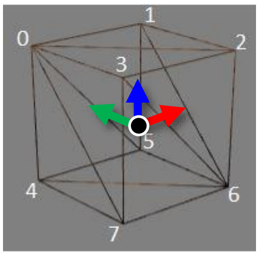
   B. Start from the code in 8-Lighting&Shading slides. Make sure camera manipulation shortcuts ‘1’, ‘3’, ‘2’, ‘w’ work.

   C. Use **glDrawElements(),** not glDrawArray(). Refer the code in 7-Hierarchy&Mesh slides.

      i. Hint: In Gouraud shading, one vertex has only one normal. This makes using glDrawElements() easier.

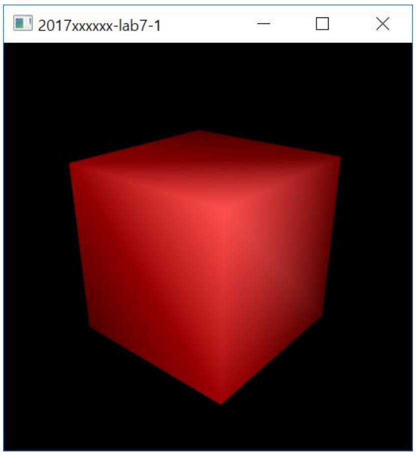
   D. Use the following normal vector data:





vertex index	position	normal
0	(-1 , 1 , 1)	(-0.5773502691896258 , 0.5773502691896258 , 0.5773502691896258)
1	(1 , 1 , 1)	(0.8164965809277261 , 0.4082482904638631 , 0.4082482904638631)
2	(1 , -1 , 1)	(0.4082482904638631 , -0.4082482904638631 , 0.8164965809277261)
3	(-1 , -1 , 1)	(-0.4082482904638631 , -0.8164965809277261 , 0.4082482904638631)
4	(-1 , 1 , -1)	(-0.4082482904638631 , 0.4082482904638631 , -0.8164965809277261)
5	(1 , 1 , -1)	(0.4082482904638631 , 0.8164965809277261 , -0.4082482904638631)
6	(1 , -1 , -1)	(0.5773502691896258 , -0.5773502691896258 , -0.5773502691896258)
7	(-1 , -1 , -1)	(-0.8164965809277261 , -0.4082482904638631 , -0.4082482904638631)

E. Expected result:



F. Files to submit: A Python source file (Name the file whatever you want (in English). Extension should be .py)