

Computer Graphics, Lab Assignment 8

Handed out: May 09, 2021

Due: 23:59, May 09, 2021 (NO SCORE for late submissions!)

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

```
+ 2021_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**.
 - Your files must be committed to the **master branch**. Otherwise, it will not be scored.
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 8-Lighting&Shading 2, Hierarchical Modeling 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. Expected result: Uploaded LabAssignment8-1.mp4
- B. Files to submit: A Python source file (Name the file whatever you want (in English). Extension should be .py)
2. Write down a Python program to draw a hierarchical model of boxes.
- A. Set the window title to **your student ID** and the window size to (480,480).
- B. Start from the following code skeleton.

```

import glfw
from OpenGL.GL import *
import numpy as np
from OpenGL.GLU import *

def render():
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glEnable(GL_DEPTH_TEST)

    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    glOrtho(-2,2, -2,2, -1,1)

    glMatrixMode(GL_MODELVIEW)

```

```

glLoadIdentity()

drawFrame()
t = glfw.get_time()

# blue base transformation
glPushMatrix()
glTranslatef(np.sin(t), 0, 0)

# blue base drawing
glPushMatrix()
glScalef(.2, .2, .2)
glColor3ub(0, 0, 255)
drawBox()
glPopMatrix()

# red arm transformation
glPushMatrix()
glRotatef(t*(180/np.pi), 0, 0, 1)
glTranslatef(.5, 0, .01)

# red arm drawing
glPushMatrix()
glScalef(.5, .1, .1)
glColor3ub(255, 0, 0)
drawBox()
glPopMatrix()

glPopMatrix()
glPopMatrix()

def drawBox():
    glBegin(GL_QUADS)
    glVertex3fv(np.array([1,1,0.]))
    glVertex3fv(np.array([-1,1,0.]))
    glVertex3fv(np.array([-1,-1,0.]))
    glVertex3fv(np.array([1,-1,0.]))
    glEnd()

def drawFrame():
    # draw coordinate: x in red, y in green, z in blue
    glBegin(GL_LINES)
    glColor3ub(255, 0, 0)
    glVertex3fv(np.array([0.,0.,0.]))
    glVertex3fv(np.array([1.,0.,0.]))
    glColor3ub(0, 255, 0)
    glVertex3fv(np.array([0.,0.,0.]))
    glVertex3fv(np.array([0.,1.,0.]))
    glColor3ub(0, 0, 255)
    glVertex3fv(np.array([0.,0.,0.]))
    glVertex3fv(np.array([0.,0.,1.]))
    glEnd()

def main():
    if not glfw.init():
        return
    window = glfw.create_window(480,480,'2017123456-lab6-1', None, None)
    if not window:
        glfw.terminate()
        return
    glfw.make_context_current(window)
    glfw.swap_interval(1)

    while not glfw.window_should_close(window):
        glfw.poll_events()
        render()
        glfw.swap_buffers(window)

```

```
glfw.terminate()  
  
if __name__ == "__main__":  
    main()
```

C. Add a green arm at the end of the red arm, and rotate the green arm about its local z axis.

- i. Render the green arm using `drawBox()`. The size of the green arm is the same as the size of the blue base.
- ii. To make the green arm draw in front of the red arm, add translation slightly in the z-axis direction (refer to the drawing of the red arm in the skeleton code).

D. Also render local frames of the blue base, red arm, green arm using `drawFrame()`.

E. Expected result: Uploaded LabAssignment8-2.mp4

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