

## Computer Graphics, Lab Assignment 5

Handed out: March 30, 2021

**Due: 23:59, March 30, 2021 (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.

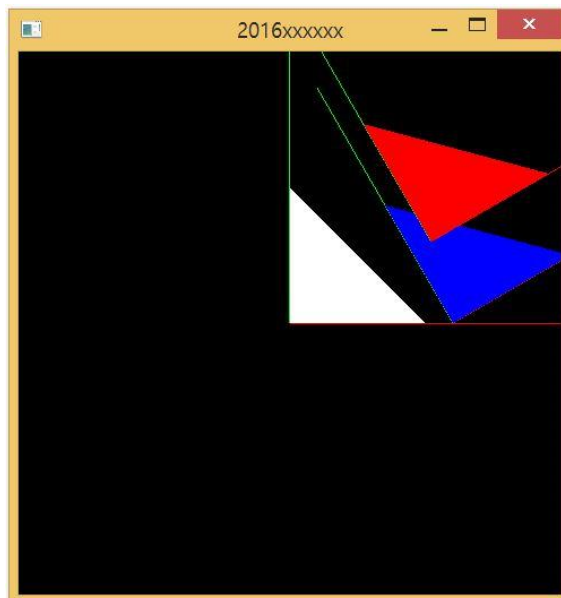
```
+ 2021_ITE0000_2019000001
+ LabAssignment5/
+ 1/
+   - 1.py
+ 2/
+   - 2.py
+ 3/
+   - 3.py
```

- The submission time is determined not when the commit is made but when the git push is made.
1. Write down a Python program to draw transformed triangles and its local frame in a 3D space.
    - A. Set the window title to **your student ID** and the window size to (480,480).
    - B. Use the following drawFrame() and drawTriangle() to draw the frame and triangle:

```
def drawFrame():
    glBegin(GL_LINES)
    glColor3ub(255, 0, 0)
    glVertex2fv(np.array([0.,0.]))
    glVertex2fv(np.array([1.,0.]))
    glColor3ub(0, 255, 0)
    glVertex2fv(np.array([0.,0.]))
    glVertex2fv(np.array([0.,1.]))
    glEnd()

def drawTriangle():
    glBegin(GL_TRIANGLES)
    glVertex2fv(np.array([0.,.5]))
    glVertex2fv(np.array([0.,0.]))
    glVertex2fv(np.array([.5,0.]))
    glEnd()
```

- C. First draw an untransformed white triangle and a global frame.
- D. Then draw a transformed blue triangle and its local frame. The triangle should be first rotated by 30 degrees and then translated by (0.6, 0, 0) w.r.t. the global frame.
- E. Then draw a transformed red triangle and its local frame. The triangle should be first translated by (0.6, 0, 0) and then rotated by 30 degrees w.r.t the global frame.
- F. Expected result:



i.

- G. Files to submit: A Python source file (Name the file whatever you want (in English). Extension should be .py)
2. As mentioned in the lecture, “moving camera” and “moving world” are two equivalent operations. Based on the following figure, replace the `gluLookAt` call() in the following code with **two** replace `glRotatef()` calls and one `glTranslatef()` call and complete the program.

```
def render():
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glEnable(GL_DEPTH_TEST)
    glPolygonMode( GL_FRONT_AND_BACK, GL_LINE )
    glLoadIdentity()

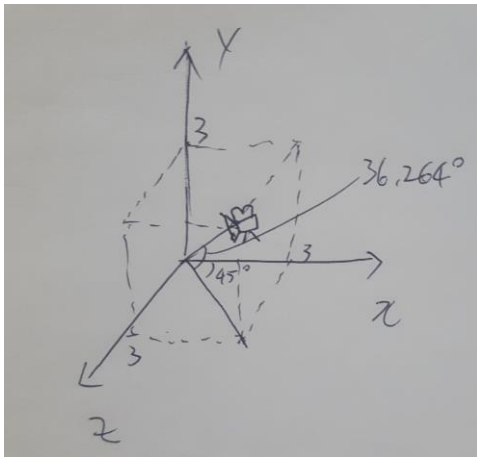
    gluPerspective(45, 1, 1,10)

    # Replace this call with two glRotatef() calls and one
    # glTranslatef() call
    gluLookAt(3,3,3, 0,0,0, 0,1,0)

    drawFrame()

    glColor3ub(255, 255, 255)
    drawCubeArray()
```

A.

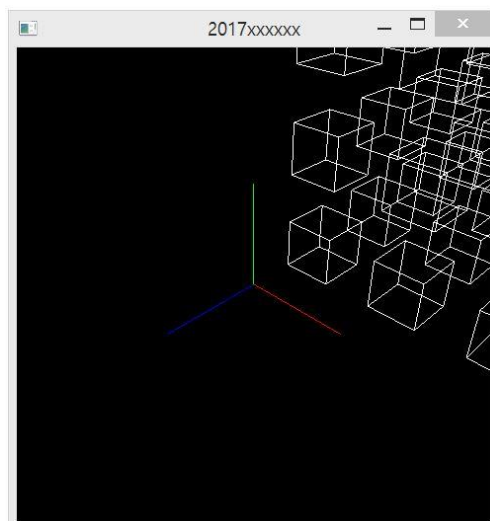


B.

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

D. Find code for drawFrame(), drawCubeArray() from 5-RenderingPipeline, viewing&projection1 slides.

E. Your program should render the following scene:



i.

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