

Computer Graphics, Lab Assignment 5

Handed out: April 04, 2021

Due: 23:59, April 04, 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 down a Python program to draw three objects transformed with different modeling transformations and their local frames in a 3D space.
 - A. Set the window title to **your student ID** and the window size to (480,480).

```

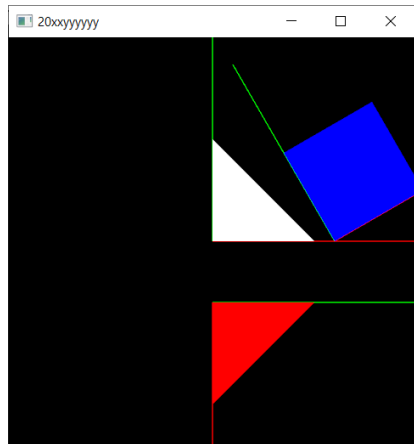
def drawFrame():
    glBegin(GL_LINES)
    glColor3ub(255, 0, 0)
    glVertex2fv(np.array([0.,0.]))
    glVertex2fv(np.array([1.,0.]))
    glColor3ub(0, 255, 0)
    glVertex2fv(np.array([0.,0.]))
    glVertex2fv(np.array([0.,1.]))
    glEnd()

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

def drawBox():
    glBegin(GL_QUADS)
    glVertex2fv(np.array([0.,.5]))
    glVertex2fv(np.array([0.,0.]))
    glVertex2fv(np.array([.5,0.]))
    glVertex2fv(np.array([.5,.5]))
    glEnd()

```

- B. Object 1 is a white triangle. Its modeling transform is the identity.
- C. Object 2 is a blue box. Its modeling transform is first rotation by 30 degrees about z and then translation by (0.6, 0, 0) w.r.t. the global frame.
- D. Object 3 is a red triangle. Its modeling transform is first translation by (0.3, 0, 0) and then rotation by -90 degrees about z w.r.t. the global frame.
- E. All objects should be rendered using the drawTriangle() and drawBox()
- F. Render the local frame of each object using the drawFrame().
- G. Do not use gluLookAt() or any other viewing & projection manipulation functions.
- H. Expected result:



i.

- I. 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 `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()

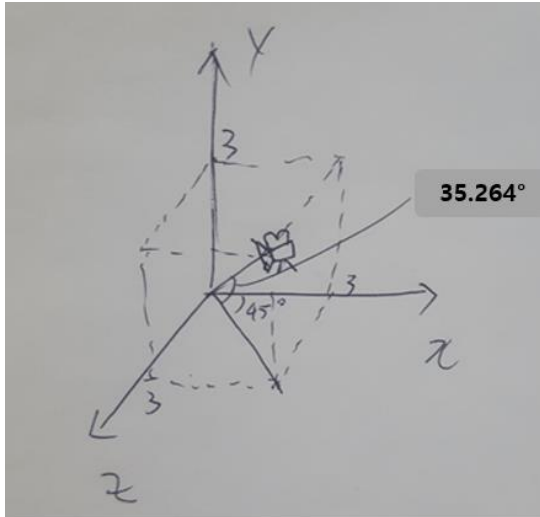
    gluPerspective(45, 1, 1, 10)

    # 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.

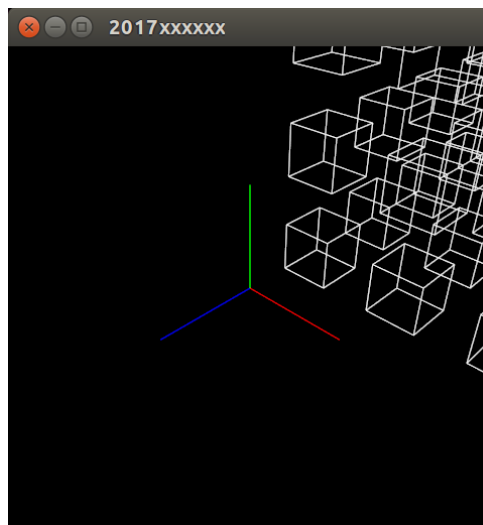


B.

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 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)