

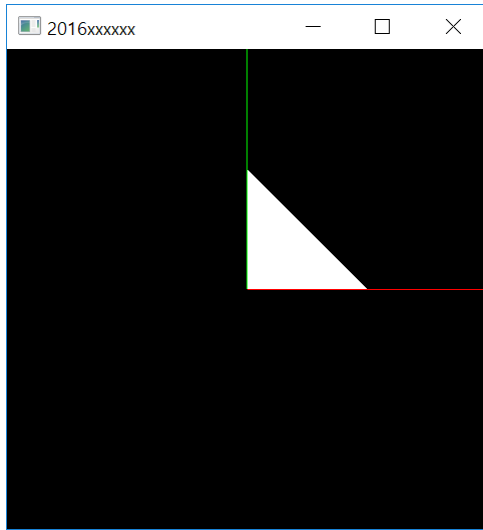
Daily Assignment 6

- Write down a Python program to..
- Draw a triangle using the render() function in 24 page of today's lecture slides (DO NOT modify it!)
 - Use **homogeneous coordinates**!
- If you press (not release) a key, the triangle should be transformed as shown in the Table:
- All transformations should be **accumulated** unless you press '1'.
 - You'll need a global variable to store current accumulated transformation
- **Set the window title to your student number.**
- Set the window size to (480,480).

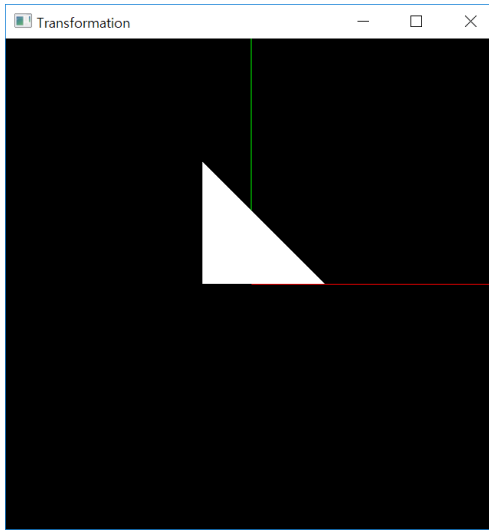
Key	Transformation
Q	Translate by -0.1 in x direction w.r.t global coordinate
E	Translate by 0.1 in x direction w.r.t global coordinate
A	Rotate by 10 degrees counterclockwise w.r.t local coordinate
D	Rotate by 10 degrees clockwise w.r.t local coordinate
1	Reset the triangle with identity matrix

An example sequence of continuous transformation

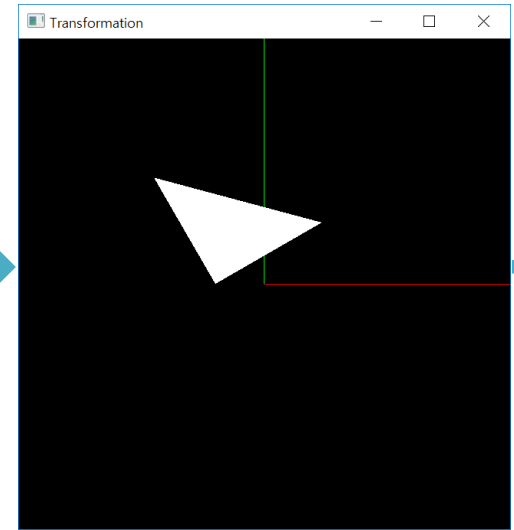
When starts



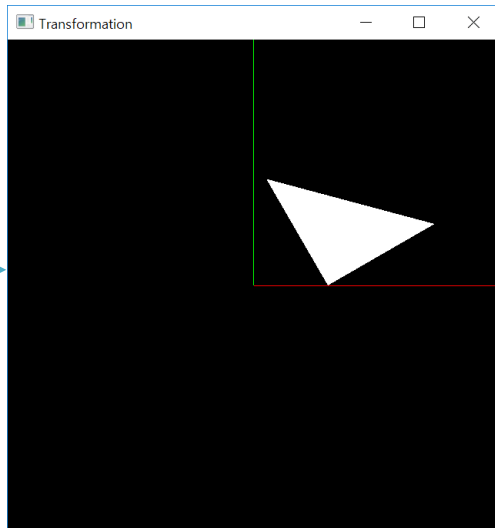
$Q * 2$



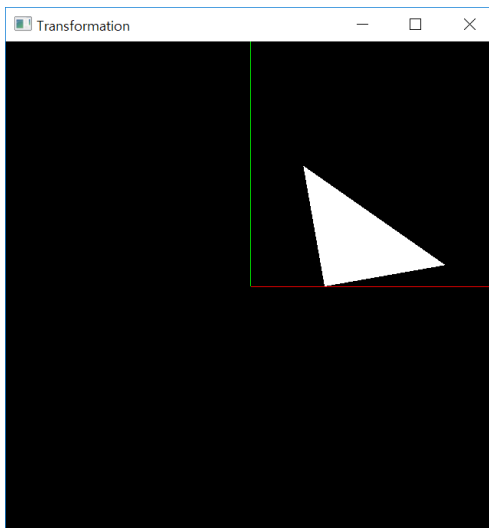
$A * 3$



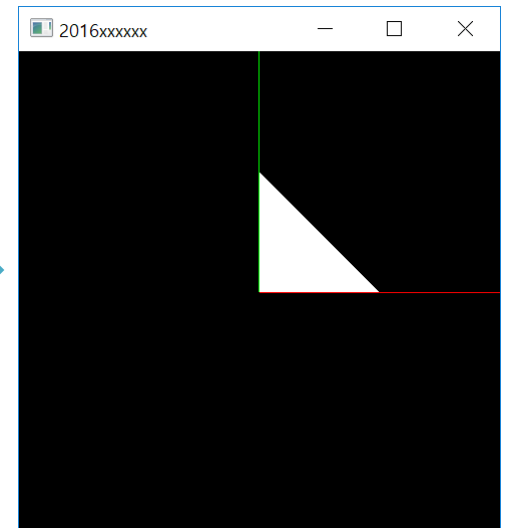
$E * 5$



$D * 2$



1



```

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

gComposedM = np.identity(3)
def render(T):
    #...
def key_callback(window, key, scancode, action, mods):
    global gComposedM
    if action==glfw.PRESS:
        if key==glfw.KEY_1:
            gComposedM = np.identity(2)
        elif key==glfw.KEY_Q:
            M = np.identity(3)
            M[:2,-1] = [-.1,0]
            # M = np.array([[1.,0.,-.1.],
            #               # [0.,1.,0.],
            #               # [0.,0.,1.]])
            gComposedM = M @ gComposedM
        elif key==glfw.KEY_E:
            M = np.identity(3)
            M[:2,-1] = [.1,0]
            gComposedM = M @ gComposedM
        elif key==glfw.KEY_A:
            M = np.identity(3)
            th = np.radians(10)
            M[:2,:2] = [[np.cos(th), -np.sin(th)],
                        [np.sin(th), np.cos(th)]]
            gComposedM = gComposedM @ M
        elif key==glfw.KEY_D:
            M = np.identity(3)
            th = np.radians(-10)
            M[:2,:2] = [[np.cos(th), -np.sin(th)],
                        [np.sin(th), np.cos(th)]]
            gComposedM = gComposedM @ M

```

```

def main():
    global gComposedM
    if not glfw.init():
        return
    window =
    glfw.create_window(480,480,"2016xxxxxx",
    None,None)
    if not window:
        glfw.terminate()
        return
    glfw.make_context_current(window)
    glfw.set_key_callback(window,
    key_callback)

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

        render(gComposedM)

        glfw.swap_buffers(window)

    glfw.terminate()

if __name__ == "__main__":
    main()

```