

CIS 515: COMPUTER GRAPHICS  
LAB – 4  
UNIVERSITY OF MICHIGAN – DEARBORN  
FALL 2024

By,  
VISHVENDRA REDDY BHOOMIDI  
[bhoomidi@umich.edu](mailto:bhoomidi@umich.edu)

## Question 1 Pyrr Transformation:

```
import glfw

from OpenGL.GL import *

import pyrr

import numpy


if not glfw.init():

    raise Exception("GLFW can't be initialized")


window = glfw.create_window(800, 600, "PyOpenGL with Pyrr Example", None, None)

if not window:

    glfw.terminate()

    raise Exception("GLFW window can't be created")


glfw.make_context_current(window)


glViewport(0, 0, 800, 600)


glEnable(GL_DEPTH_TEST)


vertices = [

    -0.5, -0.5, 0.5, 1.0, 1.0, 0.0, 0.0, 1.0,

    0.5, -0.5, 0.5, 1.0, 1.0, 1.0, 0.0, 1.0,

    0.5, 0.5, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0,

    -0.5, 0.5, 0.5, 1.0, 1.0, 0.0, 1.0, 1.0,

    -0.5, -0.5, -0.5, 1.0, 0.0, 0.0, 0.0, 1.0,

    0.5, -0.5, -0.5, 1.0, 0.0, 1.0, 0.0, 1.0,

    0.5, 0.5, -0.5, 1.0, 0.0, 1.0, 1.0, 1.0,

    -0.5, 0.5, -0.5, 1.0, 0.0, 0.0, 1.0, 1.0

]


vertices = numpy.array(vertices, dtype=numpy.float32)


indices = [
```

```
0, 1, 2, 0, 2, 3,  
5, 4, 7, 5, 7, 6,  
3, 2, 7, 7, 2, 6,  
2, 1, 5, 2, 5, 6,  
1, 0, 5, 5, 0, 4,  
3, 7, 4, 3, 4, 0  
]
```

```
indices = numpy.array(indices, dtype=numpy.uint32)
```

```
vao = glGenVertexArrays(1)
```

```
vbo = glGenBuffers(1)
```

```
ebo = glGenBuffers(1)
```

```
glBindVertexArray(vao)
```

```
glBindBuffer(GL_ARRAY_BUFFER, vbo)
```

```
glBufferData(GL_ARRAY_BUFFER, vertices.nbytes, vertices, GL_STATIC_DRAW)
```

```
glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, ebo)
```

```
glBufferData(GL_ELEMENT_ARRAY_BUFFER, indices.nbytes, indices, GL_STATIC_DRAW)
```

```
glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 8 * 4, ctypes.c_void_p(0))
```

```
glEnableVertexAttribArray(0)
```

```
glVertexAttribPointer(1, 4, GL_FLOAT, GL_FALSE, 8 * 4, ctypes.c_void_p(3 * 4))
```

```
glEnableVertexAttribArray(1)
```

```
glBindVertexArray(0)
```

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

```
    glfw.poll_events()
```

```
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
```

```

time = glfw.get_time()

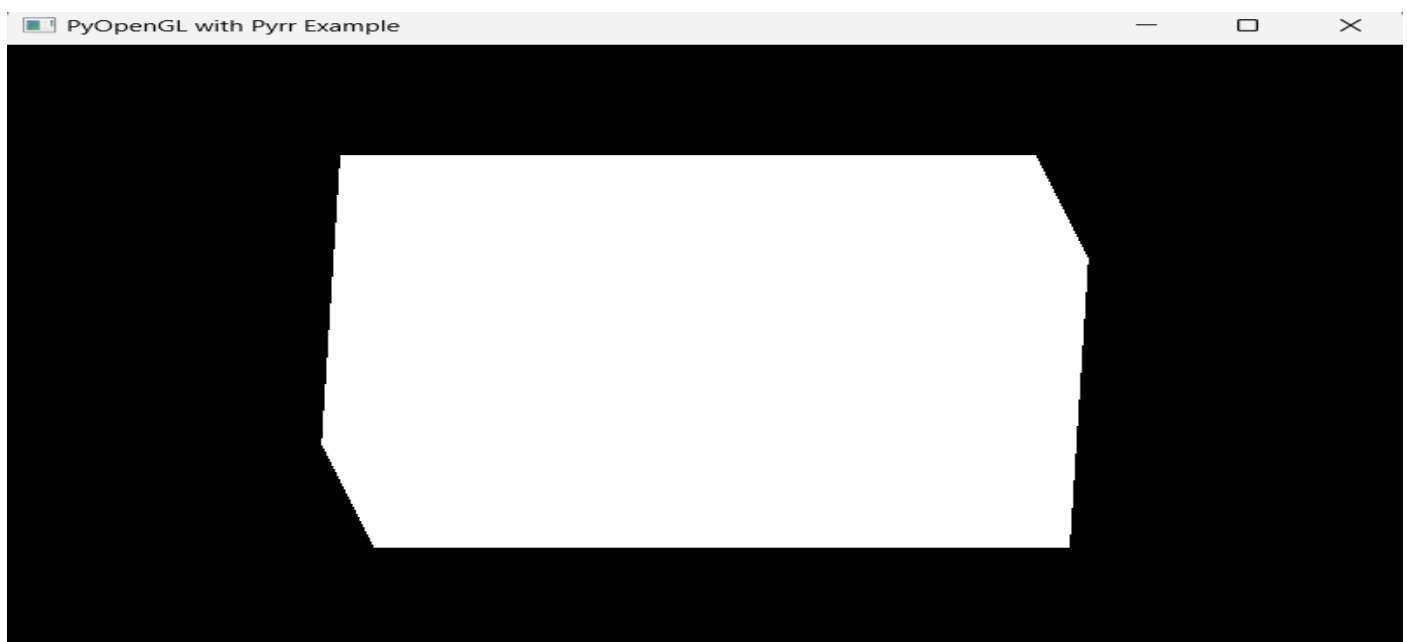
translation_matrix = pyrr.matrix44.create_from_translation([0.5, 0, 0])
scaling_matrix = pyrr.matrix44.create_from_scale([0.5, 0.5, 0.5])
rot_x = pyrr.matrix44.create_from_x_rotation(0.5 * time)
rot_y = pyrr.matrix44.create_from_y_rotation(0.8 * time)
overall_transformation_1 = rot_x @ rot_y
overall_transformation_2 = scaling_matrix @ rot_x @ rot_y
overall_transformation_3 = translation_matrix @ scaling_matrix @ rot_x @ rot_y
transform = overall_transformation_1
#transform = overall_transformation_2
#transform = overall_transformation_3
glLoadMatrixf(transform)
glBindVertexArray(vao)
glDrawElements(GL_TRIANGLES, len(indices), GL_UNSIGNED_INT, None)
glBindVertexArray(0)

glfw.swap_buffers(window)

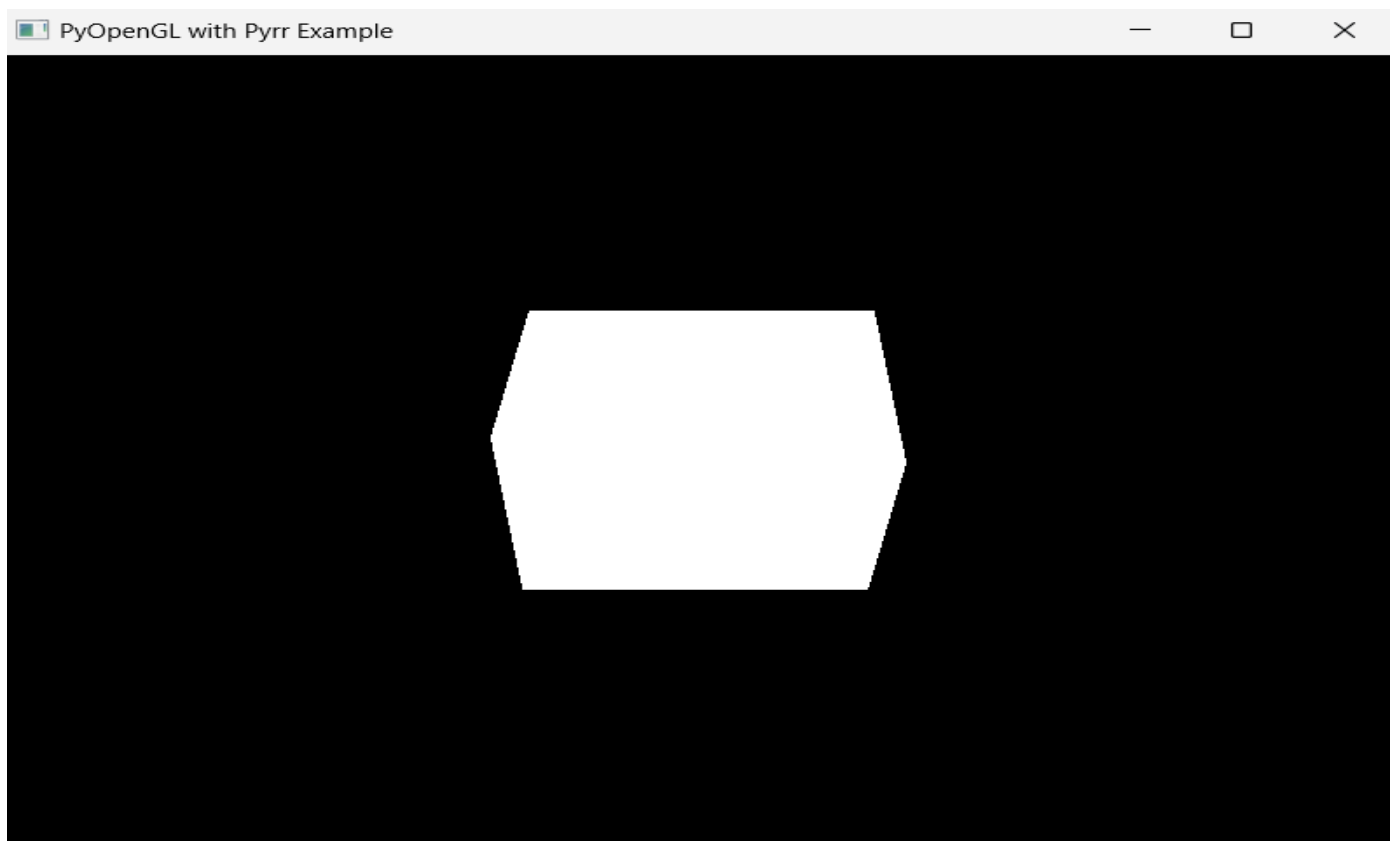
glfw.terminate()

```

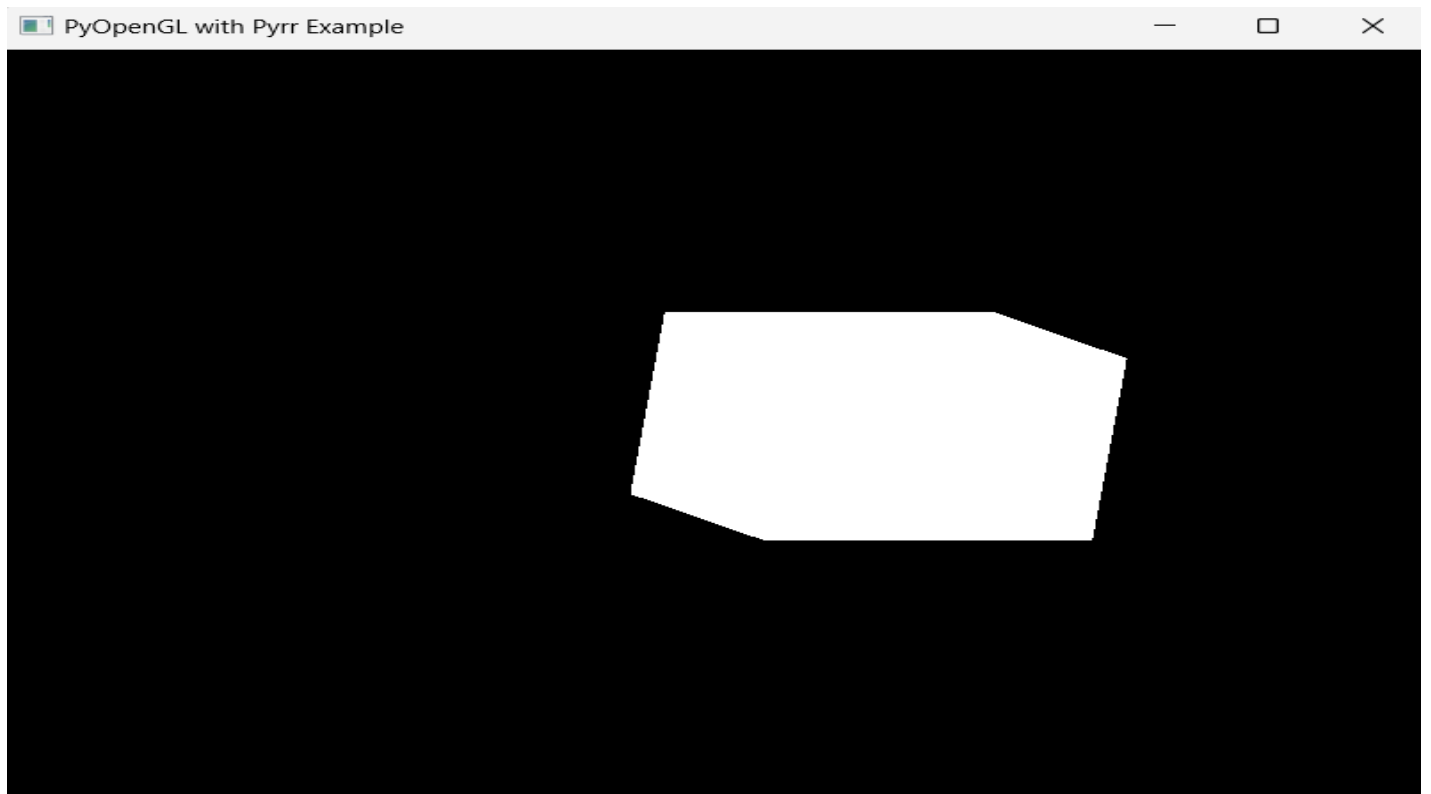
a)



b)



c)





```
faces.append((face, normal_indices))
```

```
return vertices, normals, faces
```

```
def render_obj_with_normals(vertices, normals, faces):
```

```
    for face, normal_indices in faces:
```

```
        glBegin(GL_POLYGON)
```

```
        for i in range(len(face)):
```

```
            glNormal3fv(normals[normal_indices[i]])
```

```
            glVertex3fv(vertices[face[i]])
```

```
        glEnd()
```

```
def main():
```

```
    pygame.init()
```

```
    display = (800, 600)
```

```
    pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
```

```
    gluPerspective(45, (display[0] / display[1]), 0.1, 50.0)
```

```
    glTranslatef(0.0, 0.0, -25)
```

```
    glShadeModel(GL_SMOOTH)
```

```
    glEnable(GL_LIGHTING)
```

```
    glEnable(GL_LIGHT0)
```

```
    glLightfv(GL_LIGHT0, GL_POSITION, [1, 1, 10, 1])
```

```
    glEnable(GL_DEPTH_TEST)
```

```
    glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, [0.5, 0.5, 1.0, 1.0])
```

```
    vertices, normals, faces = load_obj_with_normals(".\cube2.obj")
```

```
    while True:
```

```
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        pygame.quit()
        quit()

glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

glLineWidth(5.0)
glColor3f(1.0, 1.0, 0.0)
glBegin(GL_LINES)
glVertex3f(0.0, 0.0, 0.0)
glVertex3f(100.0, 0.0, 0.0)
glEnd()

glRotatef(1, 3, 1, 1)

render_obj_with_normals(vertices, normals, faces)

pygame.display.flip()
pygame.time.wait(10)

if __name__ == "__main__":
    main()
```

