CIS 515: COMPUTER GRAPHICS
LAB – 5

UNIVERSITY OF MICHIGAN – DEARBORN

FALL 2024

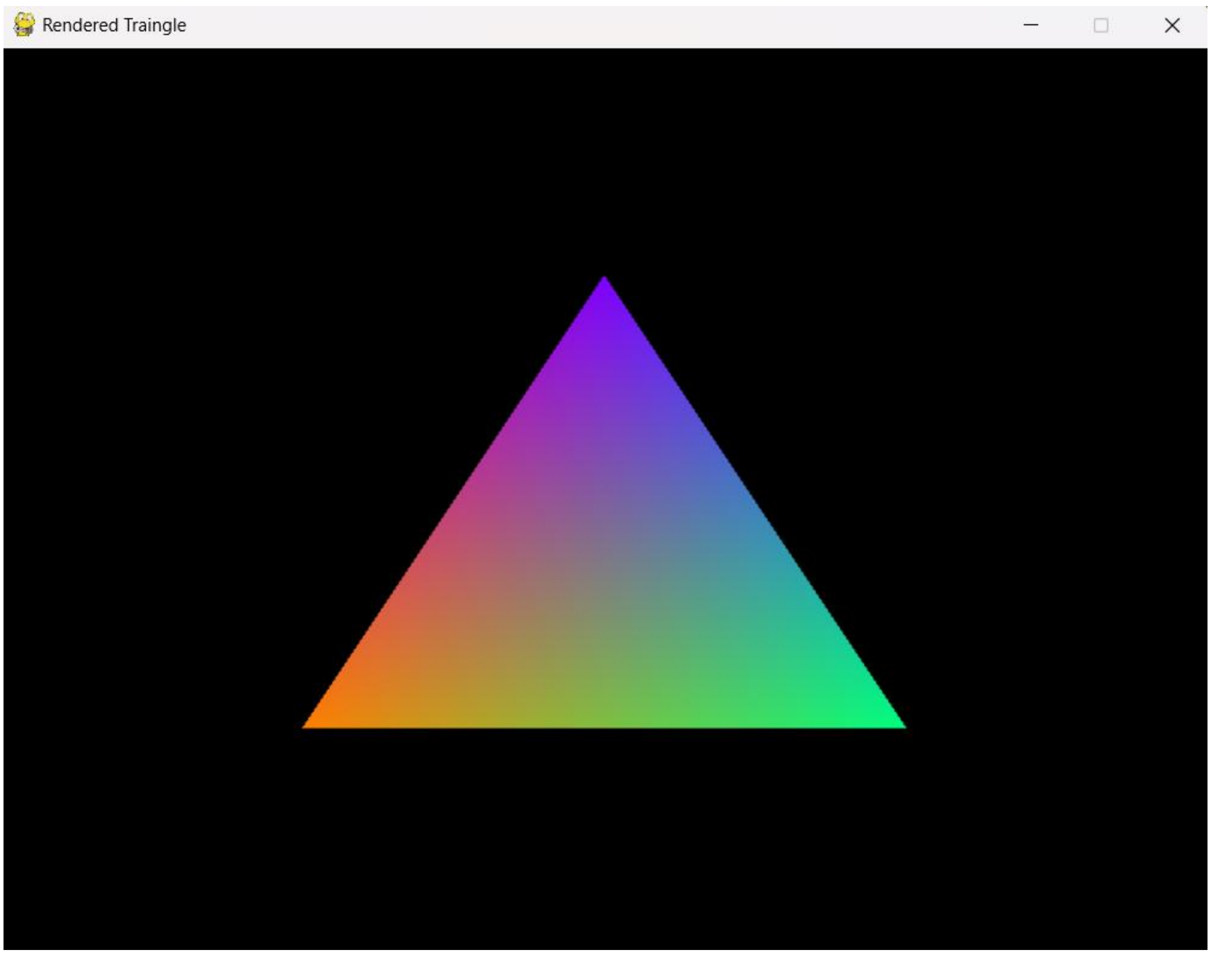
By,
VISHVENDRA REDDY BHOOMIDI
bhoomidi@umich.edu

Question 1 Shaders in a Python Triple-quote String

```
import pygame as pg
from OpenGL.GL import *
import OpenGL.GL.shaders
import numpy as np
import ctypes

pg.init()
pg.display.set_mode((800, 600), pg.OPENGL | pg.DOUBLEBUF)
pg.display.set_caption("Rendered Traingle")

vertex_src = """
#version 330 core

layout (location=0) in vec3 vertexPos;
layout (location=1) in vec3 vertexColor;

out vec3 fragmentColor;

void main()
{
    gl_Position = vec4(vertexPos, 1.0);
    fragmentColor = vertexColor;
}
"""

fragment_src = """
#version 330 core
```

```
in vec3 fragmentColor;

out vec4 color;

void main()
{
    color = vec4(fragmentColor, 1.0);
}
"""

vertex_shader = glCreateShader(GL_VERTEX_SHADER)
glShaderSource(vertex_shader, vertex_src)
glCompileShader(vertex_shader)

if glGetShaderiv(vertex_shader, GL_COMPILE_STATUS) != GL_TRUE:
    raise RuntimeError(glGetShaderInfoLog(vertex_shader))

fragment_shader = glCreateShader(GL_FRAGMENT_SHADER)
glShaderSource(fragment_shader, fragment_src)
glCompileShader(fragment_shader)

if glGetShaderiv(fragment_shader, GL_COMPILE_STATUS) != GL_TRUE:
    raise RuntimeError(glGetShaderInfoLog(fragment_shader))

shader_program = glCreateProgram()
glAttachShader(shader_program, vertex_shader)
glAttachShader(shader_program, fragment_shader)
glLinkProgram(shader_program)
```

```
if glGetProgramiv(shader_program, GL_LINK_STATUS) != GL_TRUE:
    raise RuntimeError(glGetProgramInfoLog(shader_program))


class Triangle:
    def __init__(self):
        self.vertices = (
            -0.5, -0.5, 0.0, 1.0, 0.5, 0.0,
             0.5, -0.5, 0.0, 0.0, 1.0, 0.5,
             0.0,  0.5, 0.0, 0.5, 0.0, 1.0
        )
        self.vertices = np.array(self.vertices, dtype = np.float32)

        self.vertex_count = 3

        self.vao = glGenVertexArrays(1)
        glBindVertexArray(self.vao)
        self.vbo = glGenBuffers(1)
        glBindBuffer(GL_ARRAY_BUFFER, self.vbo)
        glBufferData(GL_ARRAY_BUFFER, self.vertices.nbytes, self.vertices,
GL_STATIC_DRAW)

        glEnableVertexAttribArray(0)
        glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 24, ctypes.c_void_p(0))

        glEnableVertexAttribArray(1)
        glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, 24, ctypes.c_void_p(12))

    def destroy(self):
        glDeleteVertexArrays(1, (self.vao,))
        glDeleteBuffers(1, (self.vbo,))


triangle = Triangle()


running = True
clock = pg.time.Clock()
while running:
    for event in pg.event.get():
        if event.type == pg.QUIT:
            running = False

    glClear(GL_COLOR_BUFFER_BIT)

    glBindVertexArray(triangle.vao)
    glUseProgram(shader_program)
    glDrawArrays(GL_TRIANGLES, 0, triangle.vertex_count)

    pg.display.flip()

    clock.tick(60)

triangle.destroy()
pg.quit()
```

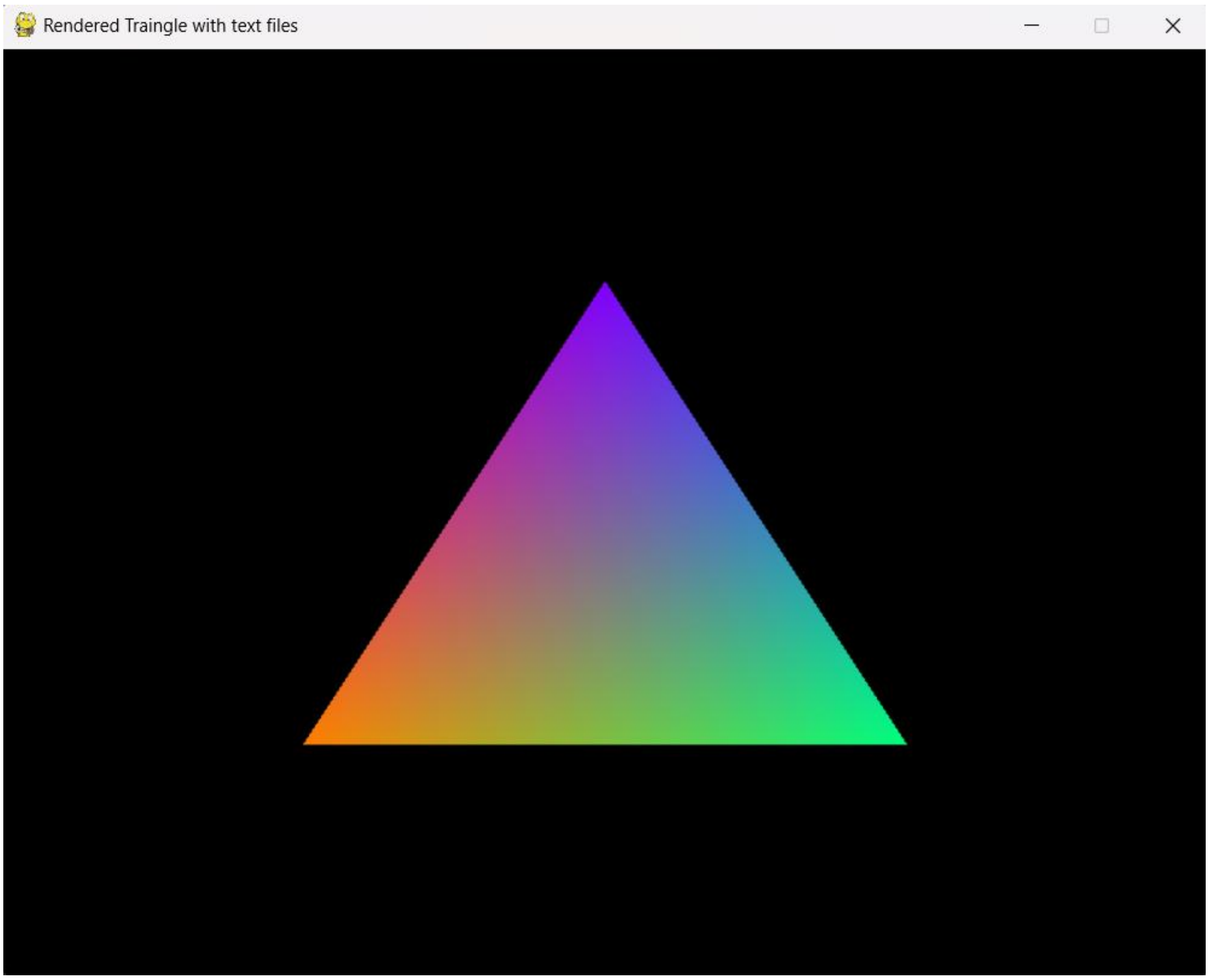


Question 2 Shaders in a text file

```
import pygame as pg

from OpenGL.GL import *

import OpenGL.GL.shaders

import numpy as np

import ctypes


pg.init()

pg.display.set_mode((800, 600), pg.OPENGL | pg.DOUBLEBUF)

pg.display.set_caption("Rendered Traingle with text files")


with open("./vertex.txt", 'r') as f:

    vertex_src = f.read()
```

```
with open("./fragment.txt", 'r') as f:
    fragment_src = f.read()

vertex_shader = glCreateShader(GL_VERTEX_SHADER)
glShaderSource(vertex_shader, vertex_src)
glCompileShader(vertex_shader)

if glGetShaderiv(vertex_shader, GL_COMPILE_STATUS) != GL_TRUE:
    raise RuntimeError(glGetShaderInfoLog(vertex_shader))

fragment_shader = glCreateShader(GL_FRAGMENT_SHADER)
glShaderSource(fragment_shader, fragment_src)
glCompileShader(fragment_shader)

if glGetShaderiv(fragment_shader, GL_COMPILE_STATUS) != GL_TRUE:
    raise RuntimeError(glGetShaderInfoLog(fragment_shader))

shader_program = glCreateProgram()
glAttachShader(shader_program, vertex_shader)
glAttachShader(shader_program, fragment_shader)
glLinkProgram(shader_program)

if glGetProgramiv(shader_program, GL_LINK_STATUS) != GL_TRUE:
    raise RuntimeError(glGetProgramInfoLog(shader_program))

class Triangle:
    def __init__(self):
        self.vertices = (
            -0.5, -0.5, 0.0, 1.0, 0.5, 0.0,
             0.5, -0.5, 0.0, 0.0, 1.0, 0.5,
             0.0,  0.5, 0.0, 0.5, 0.0, 1.0
        )
        self.vertices = np.array(self.vertices, dtype = np.float32)

        self.vertex_count = 3

        self.vao = glGenVertexArrays(1)
        glBindVertexArray(self.vao)
        self.vbo = glGenBuffers(1)
        glBindBuffer(GL_ARRAY_BUFFER, self.vbo)
        glBufferData(GL_ARRAY_BUFFER, self.vertices.nbytes, self.vertices,
GL_STATIC_DRAW)

        glEnableVertexAttribArray(0)
        glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 24, ctypes.c_void_p(0))

        glEnableVertexAttribArray(1)
        glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, 24, ctypes.c_void_p(12))

    def destroy(self):
        glDeleteVertexArrays(1, (self.vao,))
        glDeleteBuffers(1, (self.vbo,))

triangle = Triangle()

running = True
clock = pg.time.Clock()
```

```
while running:
    for event in pg.event.get():
        if event.type == pg.QUIT:
            running = False
    glClear(GL_COLOR_BUFFER_BIT)
    glBindVertexArray(triangle.vao)
    glUseProgram(shader_program)
    glDrawArrays(GL_TRIANGLES, 0, triangle.vertex_count)
    pg.display.flip()
    clock.tick(60)

triangle.destroy()
pg.quit()
```


Rendered Traingle with text files