2024 Fall CIS515/451 – Lab4
Pyrr and OBJ

Instructor: Dr. Jie Shen
Release date: Sept 28, 2023
Due date: Oct 6, 2023

Student Name:

Undergraduate students may work on each lab in teams of two members, while graduate students are required to work on the lab individually.

For each of the following tasks, you need to present your running results in the format of screenshots. Source code should be provided through copy and paste.

Pyrr is a useful Python package for defining various transformation matrices.

Shader is a small program written in GLSL (OpenGL Shading Language) that should run on the GPU for the best performance. The syntax of GLSL is similar to that of C language. Essentially, a shader code is an example of parallel computing. It can still run on laptops without GPUs although the performance is much worse.

There two common shaders:

(1) Vertex Shader: Processes each vertex of a 3D object, performing operations like transformations (e.g., translating, scaling, rotating) and passing data to the fragment shader.
(2) Fragment Shader: Determines the color of each pixel on the screen, handling operations like texturing, lighting, and shading effects.

OBJ is a common file format that can be used to define the geometry, texture, and surface normal of a complex object.

Question 1 Pyrr Transformation (10 points)

Use pyrr to define four transformations as follows:

```
    #translation_matrix = Matrix44.create_from_translation([tx, ty, tz])
    translation_matrix = pyrr.matrix44.create_from_translation([0.5, 0, 0])

    #scaling_matrix = Matrix44.create_from_scale([sx, sy, sz])
    scaling_matrix = pyrr.matrix44.create_from_scale([0.5, 0.5, 0.5])

    rot_x = pyrr.matrix44.create_from_x_rotation(0.5 * pygame.time.get_ticks() /
1000, dtype=numpy.float32)
```

```
    rot_y = pyrr.matrix44.create_from_y_rotation(0.8 * pygame.time.get_ticks() /
1000, dtype=numpy.float32)
```

Create an overall_transformation with the following three combinations of translation_matrix, scaling matrix, rot_x and rot_y:

(a) overall_transformation = rot_x @ rot_y
(b) overall_transformation = scaling_matrix @ rot_x @ rot_y
(c) overall_transformation = translation_matrix @ scaling_matrix @ rot_x @ rot_y

Display a cube defined by the following vertices and triangle nodal indices:

```
vertices = [
    -0.5, -0.5, 0.5, 1.0, 1.0, 0.0, 0.0, 1.0,  # point 0: homogeneous coordinates
and RGBA
    0.5, -0.5, 0.5, 1.0, 1.0, 1.0, 0.0, 1.0,   # point 1:
    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,    # front face: two triangles
           5, 4, 7, 5, 7, 6,    # back         7b    6c
           3, 2, 7, 7, 2, 6,    # top       3m    2w
           2, 1, 5, 2, 5, 6,    # right
           1, 0, 5, 5, 0, 4,    # bottom       4k    5g
           3, 7, 4, 3, 4, 0     # left      0r    1y
           ]
indices = numpy.array(indices, dtype=numpy.uint32)
```

Create three screenshots of three matrices of overall_transformation.

Submission of Lab 3
You should create screenshots for all the tasks in this lab.
You should submit your work through Canvas site as an MS word document or a pdf file for cis 515/451 lab. The filename of the word document should follow the convention: FirstName_LastName_Cis515-451-Lab4.doc.