Assignment 2

COS 344

Department of Computer Science

Total: 34 Marks
Deadline: 21 April 2023 at 23:59

Objectives:

- Performing simple renders. Each of the five tasks builds upon the previous task.

General instructions:

- This assignment should be completed individually, **no group effort** is allowed.
- Be ready to upload your assignment well before the deadline, as no extension will be granted.
- You must use native WebGL (no external libraries).
- You may use the book’s helper files *webgl-utils.js, initShaders.js,* and *MV.js* if you wish.
- Your upload must have a separate folder for each task.
- **All submissions will be checked for plagiarism.**

Plagiarism:

The Department of Computer Science considers plagiarism as a serious offence. Disciplinary action will be taken against students who commit plagiarism. Plagiarism includes copying someone else’s work without consent, copying a friend’s work (even with consent) and copying material (such as text or program code) from the Internet. Copying will not be tolerated in this course. For a formal definition of plagiarism, the student is referred to `https://www.library.up.ac.za/plagiarism/index.htm` (from the main page of the University of Pretoria site, follow the *Library* quick link, and then choose the *Plagiarism* option under the

Services menu). If you have any form of question regarding this, please ask one of the lecturers, to avoid any misunderstanding. Also note that the OOP principle of code re-use does not mean that you should copy and adapt code to suit your solution.

After completing this assignment:

Upon successful completion of this assignment, you will have implemented the rendering of basic 3D geometric objects and animated them.

Task 1: A house in a flat land [5 Marks]

Implement the necessary WebGL to render a scene that contains a 2D depiction of a house.
The house must contain at least the following:

- A base.
- A roof.
- A door.

Each part must have a different colour.

Task 2: Moving up a dimension [8 Marks]

Extend Task 1 to the 3D space.

- The house must contain all the features listed in Task 1, now 3-dimensional, with the addition of a window on the left and right face of the house.
- The house must rotate around the y-axis.
- Each wall should be a different colour to make them easy to distinguish.

Task 3: Viewing your home from all angles [5 Marks]

Extend Task 2 by adding 3 radio buttons that makes the house rotate about the x, y, or z plane.
Only allow one to be enabled at a time.

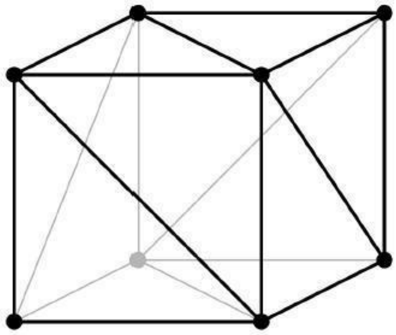
- You can reset the house between each button toggles.
- You do not need to use rotation matrices directly.

Hint: Shader files allow if statements.

Task 4: Adding wireframe rendering [8 Marks]

Given that WebGL is built on top of OpenGL ES and not full OpenGL, we lose some features. Specifically, rendering the wireframe equivalent of our scene cannot be done by simply changing a function call or a setting. If we want actual wireframe renders, we need to construct line primitives to trace the triangles used in our objects.

In this task you need to implement a function that converts a set of triangle primitives to a set of lines that represent the wireframe. You must add the functionality to Task 3 to enable or disable a wireframe render using your implemented function. You can implement this conversion function however you so choose, but it should be noted that the use of a wireframe render is very helpful from a debugging perspective as we go further in this module. As such it is worth making the conversion process general. As a simple illustration, if you had a mesh of a 3D cube, each face would consist of two triangles and your wireframe would look like:



Task 5: Render a sphere [8 Marks]

Replace your house mesh with a spherical mesh constructed using the approach described in Chapter 2.

- No two adjacent triangles can be the same colour, where we define adjacent to mean connected by a triangle side.
- The rotation radio buttons and the wireframe functionality must still work.
- Have two versions of the sphere; one with a low triangle count and one with a high triangle count. You must be able to switch between the two via a button.

Hint: Use the wireframe mode to help you spot render errors.

Submission instructions

For your submission, you need to place all your source files in a zip or tar/gzip archive (you need to compress your tar archive) named uXXXXXXXX.zip or uXXXXXXXX.tar.gz where XXXXXXXX is your student/staff number.

Submit your code for marking under the appropriate item on ClickUP (Assignment 2) before the deadline.