Assignment 3

COS 344

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Department of Computer Science

Total: 25 Marks
Deadline: 09 June 2021 at 17:00

Objectives:

- Implementing various transformations.
- Use interface for controlling transformations.

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.
- **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 3D transformations and geometry of a 3D object.

Task 1: 3D coloured cube [25 Marks]

Create a WebGL web page that displays a 3D coloured cube. Allow the user to transform the cube via key input. Keystrokes should modify the specified transformation (scaling, rotation and translation). The cube should only change/move (incrementally) in response to a keystroke. The cube is **NOT** to be animated.

- The cube geometry and colour are defined by the example code `cubev.js` from Lecture 15.
- Enable the depth test to ensure proper display.
- Allows the user to specify the transformation to be modified on the interface (Scale, Rotate, Translate). Apply all of the transformations in the order of scaling, rotation, then translation, at each update.
- Define six keys for increasing and decreasing the x,y,z components of the current transformation. The cube should only be transformed with each key stroke.
- Define two keys for increasing and decreasing the delta added to the transformations. Each transformation should have its own delta value.
- Define a key for resetting all of the transformations and delta values.
- You need to calculate the necessary transformation matrices in the vertex shader. You are **NOT** allowed to use the transformation functions in `MV.js`.
- The transformation matrix should be applied to your vertices in the vertex shader.
- Do not apply camera or projection transformations.
- Your web page should display instructions below the WebGL canvas describing the functionality of your keyboard inputs.

Note that since you will be using the default orthographic projection, translation in the z-direction should have no visible effect on the size of your cube. Of course if you transform the cube outside of the default view volume, it will be clipped.

Your cube should look something like the following:

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 3*) before the deadline.