GAME2012 - 3D Graphics Programming
Assignment 2

Instructors: Alex Richard (arichard6@georgebrown.ca)
Hooman Salamat (Hooman.Salamat@georgebrown.ca)
Connor Smiley (Connor.Smiley@georgebrown.ca)

Due: See D2L for due date

1 Introduction

In this assignment you will have to get out of your 2D comfort zone and start thinking in a whole extra dimension.

2 Logistics

This assignment is to be completed individually. All submissions for this assignment are electronic via Blackboard. This assignment is worth 10% of the course grade.

3 Instructions

There are two main parts to this assignment. In the first part, you will need to setup the project. In the second part, you need to render a couple of cubes. And animate the cubes along with the camera!

4 Deliverables

Submit your **entire project** zipped through the Assignment link on D2L, make sure that your project includes the following:

a. The sources file: .cpp/.h files (including those that load shaders)
b. Your shader files
c. Name it: GAME2012_A1_YourlastnameYourfirstname.zip

You will also need to submit **5 screenshots** of your running application. You can submit them separately over D2L or include them in a folder in your zip.

5 What you need to do

*Part 1 (worth **1**)*

1. Use any project as a starting point.

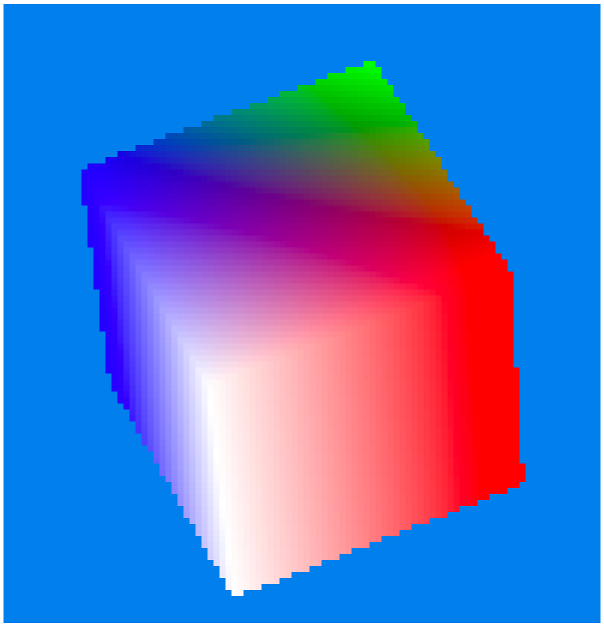
2. Change the title of the window to: GAME2012_A2_LastnameFirstname

3. Change the main .cpp file to the following format: GAME2012_A2_LastnameFirstname.cpp

4. Add the following comment header in your file and make sure to change my name to your name and add your ID. Also, make sure to add some useful description to your file.

```
//****************************************************************************
// GAME2012_A2_LastFirst.cpp by Your Name (C) 2020 All Rights Reserved.
//
// Assignment 2 submission.
//
// Description:
//   Click run to see the results.
//
******************************************************************************
```

5. Customize the project according to your taste:
    a. Change the background color from black to a different color.
    b. Change the window size to a different size.



*Part 2 (worth **9**)*

1. You need to setup your camera into Perspective mode. **(1)**
    a. Set the starting location of the camera to (0,0,5)
    b. The camera should be looking at the origin point (0,0,0).

2. You need to create vertices and use **indexed draws,** so you render a cube like the image above. Make sure your cube dimensions are set so that it fits within the window. An interpolated cube consists of 8 vertices, therefore, you are only allowed to have 8 vertices to render your cube. Each vertex should have a distinctive color. **(3)**
    a. Create a VAO to describe your cube.
    b. Create an IBO for your cube indices.
    c. Create a VBO for your cube vertices.
    d. Create another VBO for the vertex colors.
    e. Bind all buffers to a single VAO.
    f. Translate your cube to place it at (0,0.45,0).
    g. Render your cube and take the **first screenshot**!

3. You need to render a second cube using the same VAO you created for the first cube. **(1%)**
    a. Translate a new cube to place it at (0,-0.45,0).
    b. Render the second cube.

By now your scene should have two cubes positioned on top of each other. Take the **second screenshot**!

4. The next step is to animate your cubes! **(2)**
    a. Rotate each cube constantly in opposite directions

5. The final step is to move the camera! **(2)**
    a. Read keyboard inputs.
    b. Use “W” and “S” to move the camera in and out.
    c. Use “A” and “D” to move the camera left and right.
    d. Use “R” and “F” to move the camera up and down.
    e. The camera should never change the look at position. It should always look at the center of the world. This will look weird when you move it.
    f. Take **3 more screenshots** from different positions of the camera in the scene.

6 Penalties

- You don’t name your .cpp and .zip properly: 1 mark off each
- Late penalty: -10% per day up to -50% and then not accepted
- Missing screenshots: ½ mark off for each missing screenshot
- Handing in code from an example without significant changes: 0% and a plagiarism report. Adding a few changes to comments or title text is not significant.
- Old OpenGL used: If you used the fixed function pipeline (legacy OpenGL) with functions like glBegin() and glEnd() to draw, you get 0. We teach modern OpenGL in this course and if you’re using FFP, then I know you’re plagiarizing from another source.