University of Southampton	***School of Electronics and Computer Science***	***Coursework (1 of 4) Instructions***
Module: COMP3214	Title: Principles and Practice of Computer Graphics.	Lecturer: Dr. J N Carter
Deadline: see handin	Feedback: + 28 days.	Weighting: 10%

Instructions

Introduction

The coursework for this module is in five parts plus some optional pieces. The first and last are about programming in OpenGL. The first coursework is very restrictive and your aim is to produce images that conform to deliberate specifications. You are required to produce a program which displays things on a number of different screens, under user control

To be successful and obtain 100% (i.e. 10 out of 10) you must meet the specification below. You must submit a single zip file containing the code to build your program.

In this coursework you must produce a number of different screens, selected by typing '`A, B, ..` or `E`', optionally F and G with the following functionality.

A. Draw a wire-frame sphere by calculating all the vertex positions and drawing lines between them.
B. For the sphere, work out the surface normal direction[1]. Use the normal information calculated in (b) above work out the amount of illumination falling on each vertex. Assume that the light source is at infinity and is co-axial with the viewpoint. Use this to draw a shaded sphere. You can calculate the illumination in GLSL shaders and pass normal to the GPU or pass colours per vertex calculating the illumination on the CPU.
C. Develop a simple animation showing a number of cylinders, cones and spheres moving along regular paths. These can be wireframe or solid. The objects must be grouped to look like a crude rocket.
D. Draw a textured object, such as a cone, cylinder or sphere.
E. Optional, (no marks), load and display a model in '.obj' format. Or, (no marks) use radio buttons from imGUI rather than letters. This will count as the control element and if you do this there is no need for letters A to F.

The program must respond to both upper and lower case. Typing the letter 'Q', 'q' or 'Esc' must cause the program to exit cleanly. The program must start displaying screen A.

Example pictures can be found in the introductory lecture.

The following **specific restrictions** apply:

- Your solution must involve OpenGL Core > 4. If for any reason (an intel gpu) you can not get OpenGL > 4 spesk with John Carter.
- Functions in available libraries that draw cylinders and spheres etc. are explicitly forbidden.
- Only functions relating to video and keyboard events may be used.

[1] The surface normal is the unit vector at right angles to the surface triangle, assuming it is flat.

See the general submission and rules and regulations on the coursework page of the notes.

Lab Zero

In the first week of term ‘Lab Zero’ will be held in the Electronics Lab. This is optional but recommended and there are no marks awarded for it. The aim of the lab is to take you through the process of installing OpenGL and its support environment on a Windows PC. You will use a Windows 10 VM to do this or your own laptop.

If you want you can use your own laptop instead then that is ok. Alternate instructions will be provided if you have a Mac. Consult the notes on the web for further details.

Submission

Your submission should be in a zip file and should contain all your code, shaders and images/textures used. You may organise this any way you like but your code must be able to find shaders etc on a relative path.

All the source files must be in the root of your archive. If it contains a folder called ’include’ the test software will add this to the search path for header files. Be careful #include <test.h> and #include “test.h” look in very different places.

You must not include binary files, executables, libraries or object files

Names should not be case sensitive, i.e. names like X.c and x.c must not be used together. Names must not clash with system files. Otherwise there are no restrictions on the names or the number of ‘c’, ‘h’ or ‘cpp’ files submitted.

You must include a file ‘config.lua’ to indicate to the test and checking systems how to build your program. See the Coursework web page for further details.

Failure to conform to this structure is the most common cause of submissions being rejected.

Help and Advice

In general I will answer questions after lectures and will respond to e-mail queries as soon as I can, please mark them “COMP3004 Coursework Query". Any student who wishes to talk to me for a longer period should e-mail for an appointment. If you are not sure about any element of the coursework specification or the rules, talk to me before writing lots of code.

If there are any inconsistencies in these note please let me know as soon as possible.

Learning Outcomes

1. Program complex 3-D scenes using a modern computer graphics system, OpenGL
2. The properties of surfaces and their simulation
3. The fundamental display algorithms for raster graphics systems
4. Demonstrate the ability to generate basic shape from first principles.
5. Use a simple interface to control a graphics program.
6. Demonstrate basic concepts of shading, lighting and animation
7. Demonstrate an understanding of the texture mapping process.

Marking Scheme

In the event of conflict the hand in page is correct.

Criterion	Description	Outcomes	Marks
Function	Demonstrate a functioning graphics program.	1,5	1
Shapes	Make shapes	4,2	2
Surface Normals, Shade and Light.	Demonstrate understanding of geometry, shape and lights.	6	3
Animation	Combine basic shapes in a simple animation, system. Should look like a crude rocket for full marks,	5	2
Control	Switching display and orderly termination.	5	1
Texture	Demonstrate a texture mapped object.	7	1

Submission

Please submit your program files in a zip file, to the C-Bass system. There is no penalty for multiple

Late submissions will be penalised at 10% per working day. No work can be accepted after feedback has been given.
You should expect to spend up to10 hours on this assignment. Please note the University regulations regarding academic integrity.