Department of Computing and Mathematics

ASSESSMENT COVER SHEET 2024/25

Module Code and Title:	6G5Z0025 Computer Graphics
Assessment Set By:	Dr Jon Shiach
Assessment ID:	1CWK100
Assessment Weighting:	100%
Assessment Title:	Coursework
Type:	Individual
Hand-In Deadline:	See Moodle
Hand-In Format and Mechanism:	Online submission via Moodle

Learning outcomes being assessed:

LO1 Select and use appropriate mathematical tools for constructing and manipulating geometry in 3D space.

LO2 Develop an interactive 3D graphics application using an industry-standard API.

LO3 Write shader code for the programmable pipeline on modern graphics hardware using an industry standard shader language.

Note: it is your responsibility to make sure that your work is complete and available for marking by the deadline. Make sure that you have followed the submission instructions carefully, and your work is submitted in the correct format, using the correct hand-in mechanism (e.g., Moodle upload). If submitting via Moodle, you are advised to check your work after upload, to make sure it has uploaded properly. If submitting via OneDrive, ensure that your tutors have access to the work. Do not alter your work after the deadline. You should make at least one full backup copy of your work.

Penalties for late submission

The timeliness of submissions is strictly monitored and enforced.

All coursework has a late submission window of 7 calendar days, but any work submitted within the late window will be capped at 40%, unless you have an agreed extension. Work submitted after the 7-day late window will be capped at zero unless you have an agreed extension. See ‘Assessment Mitigation’ below for further information on extensions.

Please note that individual tutors are unable to grant any extensions to assessments.

Assessment Mitigation

If there is a valid reason why you are unable to submit your assessment by the deadline you may apply for Assessment Mitigation. There are two types of mitigation you can apply for via the module area on Moodle (in the 'Assessments' block on the right-hand side of the page):

- **Non-evidenced extension**: does **not** require you to submit evidence. It allows you to add a **short** extension to a deadline. This is not available for event-based assessments such as in-class tests, presentations, interviews, etc. You can apply for this extension during the assessment weeks, and the request must be made **before** the submission deadline. For this assessment, the non-evidenced extension is 2 days.
- **Evidenced extension**: requires you to provide independent evidence of a situation which has impacted you. Allows you to apply for a longer extension and is available for event-based assessment such as in-class test, presentations, interviews, etc. For event-based assessments, the normal outcome is that the assessment will be deferred to the summer reassessment period.

Further information about Assessment Mitigation is available on the dedicated Assessments page.

Personal Learning Plans

If you have a Personal Learning Plan (PLP) which states you can negotiate an extended deadline, make an appointment to see the Department's Disability Coordinator to discuss to discuss your needs, and where appropriate agree on a revised submission deadline.

Plagiarism

Plagiarism is the unacknowledged representation of another person's work, or use of their ideas, as one's own. Manchester Metropolitan University takes care to detect plagiarism, employs plagiarism detection software, and imposes severe penalties, as outlined in the Student Code of Conduct and Academic Misconduct Policy. Poor referencing or submitting the wrong assignment may still be treated as plagiarism. If in doubt, seek advice from your tutor.

As part of a plagiarism check, you may be asked to attend a meeting with the Module Leader, or another member of the module delivery team, where you will be asked to explain your work (e.g. explain the code in a programming assignment). If you are called to one of these meetings, it is very important that you attend.

Use of generative AI

The use of generative AI is permitted in this assessment, so long as it is used in accordance with the instructions provided in the 'Are you allowed to use AI in assessments?' section of the AI Literacy Rise Study Pack. All submitted work must be your own original content.

If you are unable to upload your work to Moodle

If you have problems submitting your work through Moodle, you can send your work to the Assessment Management Team using the Contingency Submission Form. Assessment Management will then forward your work to the appropriate person for marking. If you use this submission method, your work must be sent **before the published deadline**, or it will be logged as a late submission. Alternatively, you can save your work into a single zip folder then upload the zip folder to your university OneDrive and submit a Word document to Moodle which includes a link to the folder. **It is your responsibility to make sure you share the OneDrive folder with the Module Leader, or it will not be possible to mark your work.**

Assessment Regulations

For further information see the Undergraduate Assessment Regulations on the Assessments and Results information pages

Formative Feedback:	You will be given formative feedback in the labs where you will be given an opportunity to apply the various methods required for the successful completion of this coursework assignment.
Summative Feedback:	You will receive written feedback on your work in the form of a commented assessment grid, with a short comment on each column, and a general comment covering your piece of work. There will also be general feedback document offered to all students.

Introduction

Computer graphics play a vital role in the creation of computer games, movies, and television programmes. Modern computer graphics can produce images and environments that look very lifelike and sophisticated, but at the fundamental level all the computer is doing is sending information that specifies the colour that individual pixel should be illuminated. In the Computer Graphics module, you will learn the fundamental operations that we use to tell the computer what to do to produce a virtual environment that we can navigate around using the keyboard and mouse.

This assessment aims to develop the following skills which are essential for the successful completion of this task:

- **Problem solving** – you will encounter various problems during this assessment, whether you have a bug in your code that needs fixing or working out how to implement a particular method or feature. When encountering a problem try not to be overwhelmed, have confidence in your ability to learn and look for a solution and using whatever resources are available to you (e.g., module materials, textbooks, internet etc.).
- **Programming** – this assignment requires some knowledge of C++ and OpenGL. We don't expect you to write programs from scratch, rather you will be provided with code template files and lots of example code in the lab sessions during the semester that you can use and adapt for your coursework submission.
- **Mathematics** – specifically vectors and matrices which you will have studied at level 4. You will learn how to calculate matrices that apply the transformations that are essential for graphics programming.

Assessment Overview

For this assignment you are required to create an application using C++ and OpenGL that demonstrates your knowledge and understanding of the fundamental concepts in computer graphics. Your application should consist of a virtual 3D world through which the user can navigate using keyboard and mouse inputs to move a virtual camera. The design of the 3D world is left up to you and can be as sophisticated or simplistic as you like, just make sure that your application satisfies the requirements listed in the mark scheme.

You will be supported in this through the weekly lab sessions where we will be covering content required for the successful completion of the assignment. If you can complete the examples and exercises for each lab, you should then be able to apply the techniques to your application. However, you must not use the same examples that are used in the module materials. Template files are available that contain code used to create a window using OpenGL and include the basic libraries required to complete the assignment.

In addition to writing the code, you will also need to complete a self-assessment document in which you will provide a brief explanation of how you have satisfied the requirements referring to specific sections of your source code. As well as demonstrating your understanding of the unit material, this will help you determine an estimate of the mark that you should expect to receive.

Getting the template files

Using GitHub desktop

1. Click on **Clone a Repository from the Internet...**
2. Select the **URL** tab and paste the URL https://github.com/jonshiach/Computer-Graphics-Coursework.git
3. Change **Local path** to a folder on your machine where you want to store the files.
4. Click **Clone**

Using the terminal/command line

1. Open the command line in Windows (press the Windows key and search for 'cmd') or the terminal in macOS.
2. Navigate to the parent folder on your machine where you want to store the files.
3. Enter the following command and press the enter key.

```
git clone https://github.com/jonshiach/Computer-Graphics-Coursework.git
```

Build instructions

1. Download, install and run CMake.
2. Click on **Browse Source...** next to 'Where is the source code?' and select the **Computer-Graphics-Coursework/** folder.
3. Do the same for 'Where to build the binaries?' and select the **Computer-Graphics-Coursework/build/** folder.
4. Click **Configure** and select Visual Studio (Windows) or Xcode (macOS) and click **Done**. Once the configuration is complete click **Configure** again.
5. Click **Generate**.

This will create a Visual Studio or Xcode project file in the **Computer-Graphics-Coursework/build/** folder. Double-click on it to open the project and edit the source code.

Submission

To submit your assignment, you must upload a simple text file containing the URL to a GitHub repository that contains your source code and self-assessment form. DO NOT update this repository after the deadline of the assessment.

Mark Scheme

Mark	LO1 Select and use appropriate mathematical tools for constructing and manipulating geometry in 3D space. (40%)	LO2: Develop an interactive 3D graphics application using an industry-standard API. (30%)	LO3: Write shader code for the programmable pipeline on modern graphics hardware using an industry standard shader language. (30%)
0 – 38	Implementation of approaches without grounding in the module material or accepted theory/practice for the subject	Incomplete, missing or erroneous implementation of functionality in most of the application.	No or minimal implementation of shaders or shader effects
42, 45, 48	Basic use of vector and matrix objects. *Lab 4. Vectors and matrices*	Application compiles and runs without alterations to the source code or CMake file. *Labs 1. Introduction to C++ and 2. Basic shapes in OpenGL*	Implementation of shaders to apply appropriate textures to objects. *Lab 3. Textures*
52, 55, 58	Basic use of translation, rotation and scaling transformations. Implementation of glm library functions for calculating view and projection matrices *Lab 5. Transformations*	3D virtual world has been created using instances of a single object type. *Lab 6. 3D Worlds*	Use of shaders to apply dynamic lighting from point light sources. *Lab 8. Lighting*
62, 65, 68	Implementation of students own functions for calculating view and projection matrices. *Lab 6. 3D Worlds*	3D world created using multiple object types Users can navigate the virtual world using keyboard and mouse inputs *Labs 7. Moving the camera and 8. Lighting.*	Use of shaders to apply dynamic lighting from different types of light sources. *Lab 8. Lighting*
72, 75, 78	Implementation of students own functions to replace glm functions (e.g., glm::length(), glm::dot(), glm::cross() etc.). Implementation of quaternions to calculate a rotation matrix. *Labs 4. Vectors and matrices and 10. Quaternions*	Interactive dynamic aspects of the virtual world are controllable by the user (e.g., position of objects, location and function of light sources etc.). *Labs 6. 3D Worlds and 7. Moving the camera*	Appropriate implementation of normal and specular maps. *Lab 9. Normal mapping*
85, 90, 100	Use of quaternions to calculate the view matrix. Use of SLERP to ensure smooth changes in the camera direction.	Implementation of a third person camera with the ability to switch between first- and third-person view. The position of the camera or character obeys the constraints of a physical space (e.g., can't pass through objects, can't hover in midair etc.).	Use of shaders to apply parameter driven effects within the scene, e.g., light properties controlled using camera or character position.

To achieve a mark in the upper mark bands you will need to have first satisfied all the criteria for the lower mark bands. Exception to this is where a criterion in the higher mark band supersedes a lower mark band, e.g., if you've written your own functions to calculate the view and projection matrices (LO1 mark band 60 - 70) then you don't need to also have used the glm functions.

Support

Help! I don’t know where to begin or what to do!

Do not panic! Any assessment can seem daunting at first, especially if it has been a while since you did one or if the concepts of the unit are relatively new to you. Your unit tutor will be happy to answer any questions that you may have and to talk you through the assessment in more detail.

During the semester we will be building up your knowledge of computer graphics starting with the basics of creating a window for your application and drawing simple shapes up to constructing a virtual environment with multiple models with textures and lighting. You need to learn to walk before you can run so don’t expect to be able to complete the assignment in the first few weeks. As we progress through the material you will be able to add more features to your application.

I am worried about this assessment because of a Personal Learning Plan (PLP)

That is okay! We appreciate that this might be the sort of assessment that could take someone with a PLP outside of their boundaries and reading this specification alone might cause a certain amount of anxiety. If you have a PLP which may affect this assessment, please contact the unit teaching team as soon as possible and we will establish an alternative way that you can be supported to meet the learning outcomes within the remit of your PLP.

If you do not currently have a PLP but think that you might need one (e.g., to facilitate a recognised learning disorder or disability, such as anxiety) please contact the University's Disability Support Team as soon as possible (contact details available at: https://www.mmu.ac.uk/student-life/wellbeing/disability).

Opportunities for Formative Feedback

In the weekly lab sessions, you will be given opportunities to put into practice what we have covered in the lectures and use this for your assignment.

Your Final Feedback

You will receive written feedback on your work in the form of a commented self-assessment grid and a general feedback document that is informed by how well the cohort as a whole performed on the assessment.

When, where and how can I get support from the unit tutor?

You are welcome to discuss the assignment in various sessions throughout the unit, and you should also make effective use of the tutor’s availability for discussion and support. The contact details for your tutors are:

Dr Jon Shiach
j.shiach@mmu.ac.uk
DB 3.27