Graphics

5CM507

Second Assessment Brief

Wayne Rippin

Module Leader

- The UoD module leader for 5CM507 is Youbing Zhao, but this assessment has been set by Wayne Rippin.
- The MC module leader is Panagiotis Perakis. Please address all questions about this assignment to Panagiotis.
- Email: p.perakis@mc-class.gr

Key dates and details

Assessment Detail	Assessment Information
Assessment Type:	Coursework 2. Individual assignment.
Assessment Weighting:	50%
Word count/Length:	No specific word count
Learning Outcomes:	2
Submission Method:	Blackboard assignment
Submission Date:	17:00 EET time, Friday 23rd May 2025
Provisional Feedback Release Date:	12th June 2025 for all assignments submitted by the due date. It might be later for assignments submitted later due to an AED or RAC.

Description of the assessment

This assessment gives you an opportunity to demonstrate that you have understood and can apply the practical techniques covered in the course of the second part of this module, using the theoretical material covered in the first part of the module as well as any additional knowledge that you gain through your own studies. You will be showcasing your grasp of the principles of 3D graphics engine design by building an interactive 3D graphics demo on top of your own object-oriented engine framework. This is a progressive assessment that builds on the work you will do each week starting from week 4 of the spring semester.

It is important to note that, although the submission date is likely to be around the same time as the submission date for assessments for other modules, this is a progressive assessment and, as such, each week you should have something that could be submitted with little extra effort. If you keep up with the module, you will not have much to do at the end in order to be able to make a good submission. It is also important that you take backups of your work as you reach milestones so that you have something that could be submitted to roll back to if necessary.

Relationship to Programme Assessment Strategy

This assessment is a very practical assessment that gives you more of an understanding of core concepts in game engines as well as allowing you to demonstrate your ability to use a modern graphics API. It also enables you to practice using more modern C++ programming techniques. All of these build on topics covered earlier in your course and provide a foundation for your final year modules.

The creation of the implementation log prepares you for the type of activity that you may need to do once you start work, particularly if you are going to be working on a contract basis and billing clients for work undertaken.

Assessment Content

Starting in the fourth week of teaching of this part of the module (26th February 2025) and continuing through to the end of this semester, the practical sessions each week will take you through the steps needed to create a basic 3D object-oriented game engine. This will include, but is not limited to, the following:

- Scene graphs
- Loading and rendering of 3D model files created with a variety of tools
- Terrain generation and following
- Sky boxes and domes
- First-person and third-person cameras
- Keyboard and game controller input
- Collision detection

At the end of the module, you will submit the engine that you have written as well as a terrain-based interactive demo that you will demonstrate via a video recording. The demo does not have to be a fully-fledged game, but it must allow a user to navigate a 3D game environment and interact with it in some way.

You must also include in your submission an implementation log which you should start at the point you start work in this assessment. This should include an entry for each date and time you worked on this assessment. Each entry should contain:

- The amount of time you worked on the assessment in that session
- A brief summary of the tasks done in that session
- A brief summary of any problems encountered and how you resolved them
- How you tested your work.

There is no fixed format or word count for your implementation log, but it must be submitted as a PDF file.

Use of code that is not your own

You are free to use any example program code provided to you in this module in your submission without needing to reference it in any way. Any other code that you use that is not your own MUST be referenced, both in the implementation log at the point that you used the code AND as a comment in the code that clearly indicates which code is not your own. The reference must indicate the source of the code and the date it was used. This includes code taken from book examples, Internet web sites, code generated by AI tools and code copied from another student. If any code is detected in your submission that has come from another source (other than example code provided to you in this module) and has not been referenced, then it will be treated as an academic offence. Code taken from elsewhere will not be considered when your assessment is marked.

Submission Requirements

You must submit to the submission point by the due date and time. The only students who may submit later are those who have automatic extensions to deadlines (AED) due to having a support plan or those who have an approved Request for Additional Consideration (RAC). Any other submission that shows on Blackboard as being submitted late will not be marked and will be given a grade of 0. So please ensure that you allow plenty of time for your submission to be uploaded – do not leave it until the last minute.

You must submit ONE zip file that is named using your student number (for example, if your student number was 100123456, then your zip file would be named as 100123456.zip).

Your zip file must contain:

- A folder called ‘Source’ that contains the complete solution folder for your submission. This includes your game engine and the demo code. All .vs and x64 folders must be removed from this folder.
- A folder called ‘Demo’ that contains an executable version of your demonstration that is built for Release. As well as any executable files needed, this folder must also contain any shader (HLSL) files and any model or texture files required.
- A video recording, made by you, of your demonstration. It should show all of the features you want to highlight. You do not need to provide an audio commentary, but if you wish to do so, that is fine. The video recording should be supplied in .mp4 format.
- Your implementation log provided as a PDF file.
- A copy of the Functionality Checklist (described below in the Assessment Rubric section). This should be provided in Microsoft Word (.docx) or PDF format.

Failure to follow these submission requirements in any way (for example, using a compression format other than zip, multiple files supplied, folders incorrectly named, files provided in the wrong format, etc) will result in the reduction of the grade given for the assessment by up to 20%. Required components that are completely missing may result in a failure grade for this assessment.

Assessment Rubric

There are certain criteria that your submission must meet in order to be able to be given a pass grade or higher. These are:

- The program code must be written in C, C++ or HLSL and developed using Visual Studio 2022 (Community, Professional or Enterprise Edition).
- You must base the engine you write on the core C++ Framework and DirectXFramework classes that you will have been given by 26th February 2025.

- All Matrix and Vector data structures must use the SimpleMath library types and SimpleMath methods must be used to manipulate them if available.
- All access to COM interfaces must use ComPtr objects.

The grade for this assessment will be calculated as follows:

- 80% will come from the level of functionality implemented. The levels of functionality that make up the different grade bands will be given to you in the Functionality Checklist that will be provided to you by 26th February 2025.
- 20% will come from the grading of the Implementation Log. Note that if the implementation log is not provided at all, the assessment will receive a failing grade.

Grading of the Implementation Log

Grade	Description
70% or higher	A very detailed log. All entries include the amount of time spent on the assessment, the entry is detailed with full details what was done in this session, problems encountered while working on the assessment and how they were resolved. It is clear how each part has been tested as development continued. For higher grades, screen shots are used to illustrate relevant points and relevant examples of program code to illustrate specific points are included (do NOT provide screen shots of program code – copy the program code into your document as text).
60 – 69%	The log entries include the amount of time spent on the assessment. Each entry gives details of what was done in the session, but there is not much information on problems encountered and how they were resolved. There is evidence of testing as development progressed but is not very detailed.
50 – 59%	The log entries include the amount of time spent on the assessment. Each entry gives information about what was done in the session, but there is little or no information on any

	problems encountered or how they were resolved. There is very basic information provided about how the code was tested.
40 – 49%	The log entries are quite sparse, and it appears that some are missing. Very basic information is given about what as done in the session, but not much else.
35 – 39%	Very few log entries provided, and it looks as though much of it has been written just before submission.
Less than 35%	Minimal information provided.

Anonymous Marking

You must submit your work using your **student number** to identify yourself, not your name. You must NOT use your name in the text of the implementation log, in your source code or in the name of any file. When you submit your work in Turnitin you must submit your student number within the assignment document and in the *Submission title* field in Turnitin. Guidance is available showing how to do this.