

# Subject & Assessment Guide

## Computer Graphics

CIP Code 11.0804

Advanced Diploma of Professional Game Development

Game Programming

# Table of Contents

Subject Overview .....	2
Overall Learning Outcomes.....	2
Subject Description .....	2
Industry Relevance.....	2
Assumed Knowledge.....	2
Subject Textbooks.....	3
Assessment Criteria .....	3
Assessment Description .....	3
Assessment Instructions for Candidate.....	5
Software .....	6
Core .....	6
References and Additional Material .....	7

# Subject Overview

## Overall Learning Outcomes

Graphics is an expansive domain of knowledge, tools and technology to achieve everything that eventually reaches our screens. This instruction module will cover techniques, libraries, and algorithms commonly used in modern real-time rendering. We will explore the relationship between and management strategies for vertex data, program shaders, and framebuffers.

The overall learning objectives for this subject are:

- Demonstrate an understanding of modern render pipelines on Graphics Processing Units (GPUs)
- Apply Industry standard rendering techniques for games, film and simulation
- Apply knowledge of industry standard game engine render pipelines

## Subject Description

This subject is designed to teach you the **techniques** and **algorithms** used in **modern real-time rendering**.

Computer Graphics (CG) has a history dating back decades. From humble beginnings in the 1940's to the first mainstream use in movies in the 1970's, CG kept improving over the years from simple wireframe renderings to complete 3-dimensional (3-D) representation of everyday objects.

As techniques improved so did their application in film and video games, with games evolving from text-based experiences to 2-dimensional and then the leap to real-time 3-D games was made.

Modern video games make use of advanced dedicated computer hardware to display amazingly life-like visuals, hardware which is now included with almost every mobile device today. With this subject you will learn how to take control of this hardware, making use of the **OpenGL** API to push the hardware to its limits.

## Industry Relevance

Many games studios implement their own game engines while others make use of third-party solutions such as Unity3D and Unreal Engine. The skills taught within this subject would allow a student to fully understand the workings of either in-house or third-party game engines at any studio, and to be able to create their own solutions. Additionally, the techniques used are the same as those used in related industries, such as film and simulation.

## Assumed Knowledge

- Knowledge of C++ programming sufficient to create complex real-time applications
- Knowledge of basic vector and matrix mathematics for 3-D coordinate systems

- Knowledge of game engines

## Subject Textbooks

Although not required, the following textbooks are recommended to aid in the completion of this subject:

- Wolff, D, **OpenGL 4 Shading Language Cookbook**, 2<sup>nd</sup> Ed, Packt Publishing (December 24, 2013)
- Sellers, G, Wright, R & Haemel, N, **OpenGL Superbible: Comprehensive Tutorial and Reference**, 7<sup>th</sup> Ed, Addison-Wesley Professional (July 31, 2005)
- Akenine-Miller, T, Haines, E & Hoffman, N, **Real-Time Rendering**, 3<sup>rd</sup> Ed, A K Peters / CRC Press (July 25, 2008)

# Assessment Criteria

## Assessment Description

### Assessment Milestones

*Please refer to your Class Schedule for actual dates on your campus*

### General Description

Throughout this subject you will be learning graphics programming skills that apply to games, film and simulations. You will be demonstrating your understanding through the development of two 3D applications; an **OpenGL application** showcasing the use of the GPU for low-level rendering, and an application produced using a **game engine** that showcases the use of advanced rendering techniques within industry standard tools.

The **3D OpenGL application** must be developed using C++ and demonstrate the following techniques within a single scene:

- Imported 3D models, rendered using custom GLSL shaders
- Texture mapping applied to at least one imported 3D model
- 3D lighting applied, using at least 2 light sources at a time

The **3D game engine-developed application** can be created using an engine agreed upon by your teacher, such as Unreal Engine 4 or Unity3D, and must demonstrate the following techniques within a single scene:

- An interactive 2D GUI that the user can use to interact with the 3D world in some manner, such as change materials on an object, spawn effects, or toggle effects on/off.
- Custom materials created using the engine's tools that are then applied to 3D objects
- 3D models animated using skeletal animation, that react to user input to change active animations using blending. For example, an animated character that can change from an Idle animation sequence to a Walking sequence when the user interacts with a GUI.
- At least 2 particle systems created with the engine's tools that interact with the 3D world, such as colliding with the floor or other objects

All source code and project files must also be submitted for assessment, including any and all assets used.

You will also be assessed on your ability to follow good coding practices throughout the development of both projects.

### *Evidence Specifications*

This is the specific evidence you must prepare for and present by your assessment milestone to demonstrate you have competency in the above knowledge and skills. The evidence must conform to all the specific requirements listed in the table below. You may present additional, or other evidence of competency, but this should be as a result of individual negotiation with your teacher.

### *Your Roles and Responsibilities as a Candidate*

- Understand and feel comfortable with the assessment process
- Know what evidence you must provide during your assessment
- Take an active part in the assessment process
- Be ready for the assessment at the nominated time

This table defines the individual requirements for each part of the assessment criteria. Please refer to provided subject assessment documentation for any additional requirements.

Assessment Tasks & Evidence Descriptions
<p><b>1. Completed Real-Time 3D OpenGL Application</b></p> <p>Evidence that includes:</p> <ul style="list-style-type: none"> <li>• Submitted stand-alone executable for a 3D real-time application that implements OpenGL, which must run error-free and demonstrate the following in a single scene: <ul style="list-style-type: none"> <li>○ 3D models rendered with custom GLSL shaders</li> <li>○ Texture mapping</li> <li>○ 3D lighting</li> </ul> </li> <li>• Submitted source code and assets for OpenGL 3D application</li> </ul>
<p><b>2. Completed Real-Time In-Engine 3D Application</b></p> <p>Evidence that includes:</p> <ul style="list-style-type: none"> <li>• Submitted stand-alone executable for a 3D real-time application that was created within a game engine that runs error-free</li> <li>• Application must demonstrate the following features within a single scene: <ul style="list-style-type: none"> <li>○ 2D GUI that can be interacted with by the user and interacts with the 3D world in some manner</li> <li>○ Custom materials applied to 3D objects</li> <li>○ Skeletal animation that reacts to user input</li> <li>○ Particle systems that interact with the 3D world</li> </ul> </li> <li>• Submitted project source and assets for in-engine 3D application</li> </ul>

### 3. Follow Good Coding Practices

Evidence that includes:

- Applications debugged and tested to ensure they run error-free
- Code follows consistent naming conventions
- Files are commented to an acceptable industry standard as specified by your instructor

### 4. Application Handover

Evidence that includes:

- A game engine project, containing all source code and resource files, that compiles without errors
  - All temporary and built executable files and folders, and unused assets have been removed
- A “readme” or client document explaining how to compile, run and operate the program
- All submitted material archived in a single compressed file (zip, rar, or 7z)

## Assessment Instructions for Candidate

### METHOD OF ASSESSMENT

Assessment is a cumulative process which takes place throughout a subject. A ‘competent’ or ‘not yet competent’ decision is generally made at the end of a subject. Your assessment will be conducted by an official AIE qualified assessor. This may be someone other than your teacher. The evidence you must prepare and present is described

above in this assessment criteria document. This evidence has been mapped to the units of competency listed at the beginning of this document. Assessments will be conducted on a specific milestone recorded above in this assessment guide document.

### ASSESSMENT CONDITIONS

Formative assessment takes place as your teacher observes the development of your work throughout the subject and, although the assessor is likely to be aware of the evidence you are submitting, it is your responsibility to be prepared for the interview where a competency judgement is made (summative assessment). Forgetting something, or making a small mistake at the time of the milestone assessment, can be corrected. However, the assessor may choose to assess other candidates who are better prepared and return to you if time permits.

Upon completion of the assessment you will be issued with feedback and a record of the summative assessment and acknowledge that you have received the result. If you are absent for the nominated assessment milestone (without prior agreement or a sufficiently documented reason) you will be assessed as not yet competent.

## GRADING

The assessment you are undertaking will be graded as either *competent* or *not yet competent*.

## REASSESSMENT PROCESS

If you are assessed as being not yet competent you will receive clear, written and oral feedback on what you will need to do to achieve competence. You will be given a reassessment milestone no more than one (1) week later to prepare your evidence. If you are unsuccessful after your reassessment, you may be asked to attend a meeting with your Head of School to discuss your progress or any support you may need and further opportunities to gain competency.

## REASONABLE ADJUSTMENTS

We recognise the need to make reasonable adjustments within our assessment and learning environments to meet your individual needs. If you need to speak confidentially to someone about your individual needs, please contact your teacher.

## FURTHER INFORMATION

For further information about assessment and support at AIE, please refer to the assessment and course progress sections of your student handbook.

# Software

## Core

### Microsoft Visual Studio

Microsoft's Visual Studio is the recommended IDE for this subject. Other IDEs may be employed if desired as the content of this subject is designed to be cross-platform and IDE agnostic, however we cannot guarantee that all subject material will operate as intended on other IDEs and platforms.

- <https://www.visualstudio.com/>

### Unity3D / Unreal Engine 4

Projects need to be done in a game engine of choice. Learners can use the engine of their choice, no restrictions.

Unity3D is a widely used 3D game engine. It has powered many financially and critically successful games. It has a wide array of features that aid with development, especially for a small team. Games made with Unity can be built to a large array of devices.

- <http://www.unity3d.com>

Unreal Engine 4 is a complete suite of game development tools used to make games from 2D mobile games to console blockbusters and VR. Unreal 4 is a 3rd party development tool used in many game studios and offers professional development experience.

- <https://www.unrealengine.com/>

# References and Additional Material

## *Useful Web Links*

- [Rendering \(computer graphics\) Wikipedia Article](#)
- [Real-Time Rendering](#)
- [Real-Time Rendering Wikipedia Article](#)
- [OpenGL Home Page](#)
- [GL Programming](#)
- [OpenGL Tutorials](#)