

Module specification

When printed this becomes an uncontrolled document. Please access the **Module Directory** for the most up to date version by clicking on the following link: [Module directory](#)

Module Code	GME607
Module Title	Advanced Games Programming
Level	6
Credit value	20
Faculty	FACE
HECoS Code	101020
Cost Code	GAGM

Programmes in which module to be offered

Programme title	Is the module core or option for this programme
BSc (Hons) Computer Game Development	Core
BSc (Hons) Computer Game Development (with Industrial Placement)	Core

Pre-requisites

None

Breakdown of module hours

Learning and teaching hours	24 hrs
Placement tutor support	0 hrs
Supervised learning e.g. practical classes, workshops	0 hrs
Project supervision (level 6 projects and dissertation modules only)	0 hrs
Total active learning and teaching hours	24 hrs
Placement / work based learning	0 hrs
Guided independent study	176 hrs
Module duration (total hours)	200 hrs

For office use only	
Initial approval date	21/06/2022
With effect from date	Sept 2022
Date and details of revision	10/05/2023 AB approval of revalidated Games suite March 24 Module code updated from COM654
Version number	3



Module aims

This module aims to further the use of contemporary programming languages within an industry standard game engine. Students will be tasked to replicate and experiment with a variety of game mechanics and scripts to make technical demonstrations of their programming competencies.

Students are expected to further utilise game engine specific tools and strategies to showcase an amount of visual development to effectively demonstrate technical abilities. Where relevant this module will explore industry programming challenges to support with future employment application processes.

Module Learning Outcomes - at the end of this module, students will be able to:

1	Critically analyse and assess existing game-based project(s) or brief(s) to identify development flaws.
2	Develop effective solutions to game development-based problems.
3	Implement gameplay mechanics using advanced game engine tools and strategies.
4	Evaluate effectiveness of development in terms of game engine optimisation.

Assessment

Indicative Assessment Tasks:

This module will indicatively be made of several coursework pieces that build on/focus on individual areas of expertise within game development. This may include smaller, sequential activities for students to build up skills and self-efficacy at the start of the module and finalise with a larger piece that demonstrates their cumulative skill learned throughout. These may not all represent completed games, but they will be tightly associated with game development.

Some examples of this include character movement, AI behaviour trees, gameplay mechanics, combat, cameras, packaging and interfaces.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1, 2, 3, 4	Coursework	100%

Derogations

N/A



Learning and Teaching Strategies

In line with the Active Learning Framework, this module will be blended digitally with both a VLE and online community. Content will be available for students to access synchronously and asynchronously and may indicatively include first and third-party tutorials and videos, supporting files, sections of code/diagrams or any additional content that supports their learning.

As this module progresses, a structured strategy will be used to support the students engaging with the key threshold concepts relating to the learning outcomes. The module will include a balanced mixture of engaging tutor-led lectures, demonstrations, and facilitation. As the module continues experiential and peer learning strategies will be encouraged as the students' progress with their coursework.

Indicative Syllabus Outline

Depending on the relevance to current industry trends a programming language will be chosen that aligns with a contemporary Game Engine. Indicatively C++ and Unreal Engine 4/5 will be the focus of teaching and assessment, and students will be required to demonstrate the module outcomes to the specified language and engine chosen. The following essential topics will be delivered through the syllabus as core programming concepts. These may be contextualised through a selection of the second list of indicative subject areas.

Essential topics:

- Non-visual Programming (e.g. C++)
- Industry Programming Challenges
- Visual and Non-visual programming relationships
- Visual and Non-visual conversion
- General Game Programming Mechanics
- Object Oriented Game Development
- Pointers, References and Memory Allocation

Indicative subject areas:

- Gameplay Mechanics & Collision Detection
- User Interface Development
- Artificial Intelligence and Finite State Machines
- Procedural Techniques
- Creation of Components
- Technical Portfolio Development



.

Indicative Bibliography:

Please note the essential reads and other indicative reading are subject to annual review and update. Please *ensure correct referencing format is being followed as per University Harvard Referencing Guidance*.

Essential Reads

Madhav, S. (2018), *Game Programming in C++: Creating 3D Games*, London: Addison-Wesley.

Other indicative reading

Nystrom, R. (2014), *Game Programming Patterns*, New York: Genever Benning.

Ulibarri, S. S. (2020), *Unreal Engine C++ the Ultimate Developer's Handbook*, London: Druid Mechanics.

