

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	GME517
Module Title	Games Programming
Level	5
Credit value	20
Faculty	FACE
HECoS Code	101020
Cost Code	GAGM
Pre-requisite module	N/A

Programmes in which module to be offered

Programme title	Core/Optional/Standalone
BSc (Hons) Game Development	Core

Breakdown of module hours

Learning and teaching hours	30 hrs
Placement tutor support hours	0 hrs
Supervised learning hours e.g. practical classes, workshops	0 hrs
Project supervision hours	0 hrs
Active learning and teaching hours total	30 hrs
Placement hours	0 hrs
Guided independent study hours	170 hrs
Module duration (Total hours)	200 hrs

Module aims

This module aims to do the following:

- Revisit traditional and contemporary programming solutions as part of an industry standard game engine.
- Explore Object Oriented Programming techniques and implementations as part of small demonstrations.
- Combine OOP techniques as part of a contextualised portfolio piece.
- Explore the uses of technical demos to demonstrate effective practice in a portfolio setting.

Module Learning Outcomes

At the end of this module, students will be able to:

1	Apply concepts of Object-Oriented Programming within game development.
2	Utilise Data Structures and Object Communication to solve project-based problems.
3	Differentiate Visual and Non-Visual programming strategies within game engines and relate them to appropriate uses.
4	Assemble a technical, industry-facing showcase piece that demonstrates the Object-Oriented Programming paradigm.

Assessment

Indicative Assessment Tasks:

This section outlines the type of assessment task the student will be expected to complete as part of the module. More details will be made available in the relevant academic year module handbook.

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

Assessment number	Learning Outcomes to be met	Type of assessment	Duration/Word Count	Weighting (%)	Alternative assessment, if applicable
1	1, 2, 3, 4	Portfolio	N/A	100%	N/A

Derogations

None

Learning and Teaching Strategies

The primary skill base of this module will be delivered through a series of lectures, demonstrations and practical work that will equip the students with the technical skills and confidence to develop small computer programs. This content will also be fully represented online through VLE content and will be available to students as they progress with the module. Indicatively these could be sections of code/diagrams/PDF tutorials/pre-recorded videos depending on what is most appropriate.

Assessment will occur throughout the module to solidify key threshold concepts of programming fundamentals. Individual briefs will be given to clearly identify areas of focus through each stage, so students are fully aware of their progress throughout.

Welsh Elements

Students will have the right to submit assessed work in Welsh, to receive academic supervision and access learning resources and correspondence in Welsh. Students are encouraged to engage with Welsh cultural themes, creative elements and enterprising opportunities as part of student-led activities and assignments to facilitate wider contextual awareness and distinctive creative outputs.

Indicative Syllabus Outline

Indicative Syllabus includes the following topic areas:

- Object-Oriented Programming Paradigm
- Visual Programming (e.g. Unreal Engine 5 Blueprint)
- Non-Visual Programming (e.g. C++)
- Structs, Classes & Objects
- Encapsulation
- Abstraction
- Inheritance & Derived Classes
- Polymorphism in Game Development
- Pointers, References & Memory Allocation
- Source Control & GitHub
- Technical Portfolio Development
- Artificial Intelligence Strategies & Finite State Machines

Indicative Bibliography

Please note the essential reads and other indicative reading are subject to annual review and update.

Essential Reads:

Butler, S., Oliver, T. (2024), *Game Development Patterns with Unreal Engine 5: Build maintainable and scalable systems with C++ and Blueprint*, Birmingham: Packt Publishing.

Other indicative reading:

Ferrone, H. (2025), *Learning C# by Developing Games with Unity 6: Get to grips with coding in C# and build simple 3D games in Unity from the ground up*, Eighth Edition, Birmingham: Packt Publishing.

Horton, J. (2024), *Beginning C++ Game Programming: Learn C++ from scratch by building fun games*, Third Edition, Birmingham: Packt Publishing.

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

Administrative Information

For office use only	
Initial approval date	July 2026
With effect from date	September 2027
Date and details of revision	
Version number	1