

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	COM654
Module title	Advanced Games Programming
Level	6
Credit value	20
Faculty	FAST
Module Leader	Jack Harker
HECoS Code	100461
Cost Code	GACP

Programmes in which module to be offered

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

Pre-requisites

N/A

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	21/06/2022
Date and details of revision	
Version number	1

Module aims

This module aims to further the use of traditional 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 towards 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 and at least some will represent a portfolio-worthy technical project. 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

Due to the nature of the subject, many of the sessions will represent didactic segments that include demonstration of key concepts that student will be able to watch and/or follow. This content will also be fully represented online through VLE content that will be available to students as they work. Indicatively these could be sections of code/PDF tutorials/pre-recorded videos depending on what it most appropriate.

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

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 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. This may include elements of game programming without the use of a traditional game engine. The following essential topics will be delivered through the syllabus as core programming concepts. These may be contextualised through the second list of indicative subject areas.

Essential topics:

- Non-visual Programming (e.g. C++)
- Visual and Non-visual conversion
- General Game Programming Mechanics
- Pointers
- Algorithms & Design

Indicative subject areas:

- Industry Programming Challenges
- Artificial Intelligence
- Animation in Engine
- Visual Effects in Engine
- Procedural Techniques
- Creation of Components
- Collision Detection

Indicative Bibliography:

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

Essential Reads

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

Other indicative reading

Ferrone, H. (2020), *Learning C# by Developing Games with Unity 2020*, Fifth Edition, Birmingham: Packt Publishing.

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

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

Employability skills – the Glyndŵr Graduate

Each module and programme is designed to cover core Glyndŵr Graduate Attributes with the aim that each Graduate will leave Glyndŵr having achieved key employability skills as part of their study. The following attributes will be covered within this module either through the content or as part of the assessment. The programme is designed to cover all attributes and each module may cover different areas. [Click here to read more about the Glyndwr Graduate attributes](#)

Core Attributes

Engaged
Enterprising
Creative

Key Attitudes

Commitment
Curiosity
Resilience
Confidence
Adaptability

Practical Skillsets

Digital Fluency
Organisation
Critical Thinking
Communication