

## 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	COM559
Module title	Games Programming
Level	5
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	36 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
<b>Total active learning and teaching hours</b>	<b>0 hrs</b>
Placement / work based learning	0 hrs
Guided independent study	164 hrs
<b>Module duration (total hours)</b>	<b>200 hrs</b>

For office use only	
Initial approval date	10/06/2021
With effect from date	September 2021

<b>For office use only</b>	
Date and details of revision	
Version number	1

## Module aims

This module aims to revisit the essential pillars of Object-Oriented Programming (OOP) through the scope of game development. Early work through the module will ensure that students have the understanding the concepts of objects and data structures whilst continuing to expand on more complex techniques and strategies.

This module will further students understanding of the OOP paradigm through experimentation and development within a game engine environment. Within this there will be a specific focus on visual and non-visual tools and strategies, how they relate to one another and when it is best to use them within game development. Students will finish this module with portfolio work that demonstrates their learning and programming acumen to date.

## 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	Produce technical portfolio pieces that demonstrate the Object-Oriented Programming paradigm.

## 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.

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 OOP. 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. 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:

- Object-Oriented Programming paradigm.
- Visual Programming (e.g. UE4 Blueprint)
- Non-visual Programming (e.g. C++)
- Visual and Non-visual conversion
- Object Communication (Interfaces, references)
- Data Structures
- Pointers
- Algorithms

Indicative subject areas:

- Artificial Intelligence
- Audio and Sounds
- Animation in Engine
- Visual Effects in Engine
- Procedural Techniques
- Creation of Components

## Indicative Bibliography:

---

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

### Essential Reads

### 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.

## 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  
Ethical

### Key Attitudes

Commitment  
Curiosity  
Resilience  
Confidence  
Adaptability

### Practical Skillsets

Digital Fluency  
Organisation  
Leadership and Team working  
Critical Thinking  
Emotional Intelligence  
Communication