



Module Specification

Games in C++

Version: 2026-27, v3.0, Approved

Contents

Module Specification	1
Part 1: Information	2
Part 2: Description	2
Part 3: Teaching and learning methods	3
Part 4: Assessment.....	4
Part 5: Contributes towards	5

Part 1: Information

Module title: Games in C++

Module code: UFCFWA-30-1

Level: Level 4

For implementation from: 2026-27

UWE credit rating: 30

ECTS credit rating: 15

College: College of Arts, Technology and Environment

School: CATE School of Computing and Creative Technologies

Partner institutions: None

Field: Computer Science and Creative Technologies

Module type: Module

Pre-requisites: None

Excluded combinations: None

Co-requisites: None

Continuing professional development: No

Professional, statutory or regulatory body requirements: None

Part 2: Description

Overview: Introducing students to C++ software development and making games using the C++ language.

Features: Not applicable

Educational aims: The aim of this module is to introduce students to fundamental concepts underpinning computer games programming, using the C++ programming language and fundamental software development practices, problem solving

techniques and mathematics which, together, will allow students to confidently write code that solves typical games programming problems.

Outline syllabus: Introduction to the C++ programming language, including topics such as:

Control structures and execution flow

Classes and object orientation

IDEs and compilation / execution / debugging processes

Software development process, including topics such as:

Problem solving with code / functional decomposition and processes to use when implementing game code

Coding style considerations

Documentation practices

Mathematics, including topics such as:

Logical operators

Cartesian coordinates, vectors, lines, curves and linear equations.

Mathematical functions and their uses in games programming

Part 3: Teaching and learning methods

Teaching and learning methods: The syllabus will be explored through a combination of lectures and practical sessions. Students will also be expected to learn independently and carry out directed study and/or reading outside of taught classes.

Lectures will introduce programming concepts whilst being practically explored within supervised practical sessions guided by tutorial exercises.

Students will be set a number of exercises to further explore taught concepts, both as worksheets and assignment exercises. It is expected that the majority of the assignment work will be carried out independently, outside of taught sessions,

however there will be assessment specific sessions available to provide targeted help prior to hand-in.

Module Learning outcomes: On successful completion of this module students will achieve the following learning outcomes.

MO1 Write, compile, debug and run computer programs demonstrating appropriate use of the C++ language.

MO2 Analyse and decompose typical games programming problems using methodical processes, such as designing solutions using software engineering techniques and diagrams.

MO3 Apply fundamental mathematical concepts and logic to solve games programming problems and produce appropriate algorithms.

Hours to be allocated: 300

Contact hours:

Independent study/self-guided study = 228 hours

Face-to-face learning = 72 hours

Reading list: The reading list for this module can be accessed at [readinglists.uwe.ac.uk](https://uwe.rl.talis.com/modules/ufcfwa-30-1.html) via the following link <https://uwe.rl.talis.com/modules/ufcfwa-30-1.html>

Part 4: Assessment

Assessment strategy: There will be a set number of the tutorial exercises during the teaching year. The tutorial exercises for the module will be peer and/or tutor reviewed in practical sessions and will contribute to the students' portfolio of work. While these contribute to the summative assessment, it also allows the students formative feedback from tutors in practical sessions.

In addition to the tutorial exercises, assignment exercises will be set. These assignment exercises will also form part of the portfolio for the module and will be set in order of increasing complexity. The reason behind this strategy is to align

assessed work with the topics being taught and distribute workload for the module across the year.

Students' work will be overseen during the practical sessions, allowing module staff to see students' independent work and give formative feedback.

The resit is a rework of the key elements of the portfolio.

Assessment tasks:**Portfolio (First Sit)**

Description: Portfolio of practical exercises

Weighting: 100 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO3

Portfolio (Resit)

Description: Portfolio of practical exercises

Weighting: 100 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO3

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Games Technology {Foundation} [Frenchay] BSc (Hons) 2025-26

Games Technology [Frenchay] BSc (Hons) 2026-27

Games Technology [Frenchay] BSc (Hons) 2026-27

Games Technology {Foundation} [Frenchay] BSc (Hons) 2025-26

