Module Specification

Games in C++

Version: 2023-24, v3.0, 27 Feb 2023

Contents

Part 1: Information

Module title: Games in C++

Module code: UFCFWA-30-1

Level: Level 4

For implementation from: 2023-24

UWE credit rating: 30

ECTS credit rating: 15

Faculty: Faculty of Environment & Technology

Department: FET Dept of Computer Sci & Creative Tech

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:
- Variables and operators
- Control structures and execution flow
- Functions
- Classes and object orientation
- IDEs and compilation / execution / debugging processes

Software development process:
- Problem solving with code / functional decomposition, from planning to implementation
- Object oriented decomposition and UML notation
- Coding style considerations and documentation practices
- Understanding of appropriate development techniques and processes to use when implementing game code

Mathematics:
- Set theory and logic: operators, truth tables, simple propositional / predicate logic.
- Computer arithmetic: binary and decimal representations.
- Algebra: basic manipulation, Cartesian coordinates, lines, curves and linear equations.
- Trigonometry, functions, tangents, and normals as applied to geometry.

Part 3: Teaching and learning methods

Teaching and learning methods: The syllabus will be explored through a combination of lectures and studio 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 studio sessions guided by tutorial tasks.

Students will be set tutorial tasks to form a lab logbook. In addition, coursework tasks will be set to explore further taught concepts. It is expected that the majority of this coursework 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 and run high-level computer programs demonstrating appropriate use of the C++ language syntax

MO2 Utilise the debugging facilities of an IDE (Integrated Development Environment) to identify, analyse and resolve run-time errors

MO3 Use methodical processes to analyse and decompose typical games programming problems in order to design, implement and evaluate their algorithmic solutions

MO4 Employ software engineering techniques and associated notation to illustrate and interpret small-scale software designs

MO5 Apply fundamental mathematical concepts from algebra, trigonometry, computational arithmetic, logic and set theory, to solve games programming problems

Hours to be allocated: 300

Contact hours:

Independent study/self-guided study = 228 hours

Face-to-face learning = 72 hours

Total = 300

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

Part 4: Assessment

Assessment strategy: A set number of the tutorial tasks are to be completed to form individual lab logbooks. The tutorial tasks set for the module will be peer and/or tutor reviewed in studio/practical sessions. Completed tasks will contribute to a logbook, which forms part of the students' portfolios. While this logbook contributes to the summative assessment, it also allows the students formative feedback from tutors in studio sessions.

In addition to the tutorial tasks, coursework tasks will be set. These coursework tasks 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 tasks with the topics being taught, and distribute workload for the module across the year.

Students' work will be overseen during the practical sessions and through the logbook tasks. Allowing module staff to see students' independent work and give formative feedback.

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

Assessment tasks:

Portfolio (First Sit)
Description: Part 1: Portfolio of practical exercises and lab logbook
Weighting: 35 %
Final assessment: No
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4, MO5

Portfolio (First Sit)
Description: Part 2: Portfolio of practical exercises and lab logbook
Weighting: 65 %
Final assessment: Yes
Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO4, MO5

Portfolio (Resit)
Description:
Weighting: 35 %
Final assessment: No
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4, MO5

Portfolio (Resit)
Description: Portfolio of practical exercises
Weighting: 65 %
Final assessment: Yes
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4, MO5

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Games Technology [Frenchay] BSc (Hons) 2023-24

Games Technology {Foundation} [Frenchay] BSc (Hons) 2022-23