



Module Specification

Game Engine Programming

Version: 2024-25, v4.0, 27 Mar 2024

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	6

Part 1: Information

Module title: Game Engine Programming

Module code: UFCF9M-30-2

Level: Level 5

For implementation from: 2024-25

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: Not applicable

Features: Not applicable

Educational aims: Regardless of your position in the modern games industry, you will be expected to work with complex game engine software and asset pipelines. Game engines, and the tools contained within them, are intricately crafted software solutions. Designed to allow creativity, flexibility and productivity in the game

development process, they must also ensure quality and performance of the game produced.

Game developers require an understanding of the design principles and decisions that dictate a game engine's architecture, and the impact of these on the development pipeline and eventual game performance

Through this module, students examine software design and implementation aspects of modern game engines and their pipelines, completing tasks designed to highlight details of engine architecture and pipelines.

Outline syllabus: The following provides an indicative list of module content, which may vary with delivery to respond to current trends:

Game engines / frameworks, rationale and examples.

Game engine software requirements and how they relate to 'traditional' software engineering.

Scripting tools and languages, and provisioning for these within game engine design.

Software design roots, tools and considerations.

The creation of a game sub-system using multiple game engines.

Additional content covering asset management, engine design and component / subsystem interaction, source control / build management and appropriate design documentation will also be covered to support professional practices in game engine management.

Part 3: Teaching and learning methods

Teaching and learning methods: Teaching and learning will be split as follows:

Lectures to introduce concepts and theoretical underpinnings of engines / frameworks.

Studio sessions to encourage practical exploration and provide a vehicle for formative feedback on work for the module.

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

MO1 Analyse the features of typical game engines, with a focus on critical game engine architecture principles, evaluating the suitability in a given game development scenario.

MO2 Implement specific software and hardware requirements of game software, with understanding of how game engines / frameworks enables productivity within this context.

MO3 Implement best practices when using asset pipeline functionality guided by potential requirements of the art and design side of a game development team.

Hours to be allocated: 300

Contact hours:

Independent study/self-guided study = 228 hours

Face-to-face learning = 72 hours

Total = 0

Reading list: The reading list for this module can be accessed at [readinglists.uwe.ac.uk](https://uwe.rl.talis.com/modules/ufcf9m-30-2.html) via the following link <https://uwe.rl.talis.com/modules/ufcf9m-30-2.html>

Part 4: Assessment

Assessment strategy: This portfolio module requires students to work on individual tasks implementing game simulations using multiple engines, supported by research

and iterative design practices.

Subsequently, students will be given a brief, typically with a technical twist, to develop small-scale game simulations using an engine appropriate to the context. Industry standard version management tools will be used, providing an introduction to industry development practice, which will be developed further in the third year.

Formative feedback will be offered throughout the module, through the practical studio sessions and periodic progress meetings will provide a vehicle for continuous formative feedback.

The resit strategy is the same as for first sit.

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

Description: This portfolio module requires students to work on an individual task implementing game feature/s using specified game engines, supported by research and iterative design practices.

Weighting: 100 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO3

Portfolio (Resit)

Description: This portfolio module requires students to work on individual tasks implementing game simulations using specified game engines, supported by research and iterative design practices.

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 [Frenchay] BSc (Hons) 2023-24

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