Liverpool John Moores University

Title:	Software Engineering for Games
Status:	Definitive
Code:	**5210COMP** (127989)
Version Start Date:	01-08-2021
Owning School/Faculty:	Computer Science and Mathematics
Teaching School/Faculty:	Computer Science and Mathematics

Team	Leader
Chris Carter	Y
Yann Savoye	

Academic Level:	FHEQ5	**Credit Value:**	20	**Total Delivered Hours:**	45.5
Total Learning Hours:	200	**Private Study:**	154.5		

Delivery Options
Course typically offered: Semester 1

Component	Contact Hours
Workshop	44

Grading Basis: 40 %

Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Artefacts	AS1	Object-Oriented Game Application as an Agile Development Team	60	
Exam	AS2	Examination of Software Engineering Techniques	40	1.5

Aims

To practically apply the concepts Object-Oriented Design and Programming to games software such as Software Design Patterns
To explain the models, tools and techniques of the software development process for game software.
For students to critically evaluate the phases of the Software Development Lifecycle

and different methodologies that are used in the games industry.
To explain formal principles of game software modelling.
To provide skills in using software APIs relevant for the computer games industry.
To manage the software development lifecycle using industry-standard approaches to Software Configuration Management (SCM) using Version Control Systems (VCS)
To practically apply the principles of Agile Development to the design and development of a non-trivial game application.

Learning Outcomes

After completing the module the student should be able to:

1	Design and manage the development of a specified software game using appropriate software engineering tools and methods
2	Use an object-oriented programming language and one or more appropriate Application Programming Interface (API) to implement a non-trivial software design
3	Define and construct Object-Oriented Designs and code specifications relating to common gameplay challenges and game design patterns
4	Evaluate various tools and techniques used in the Software Development Lifecycle of a software game

Learning Outcomes of Assessments

The assessment item list is assessed via the learning outcomes listed:

Object-Oriented Game App	1	2
Examination	3	4

Outline Syllabus

Games Software Engineering characteristics and Software Engineering paradigms.
The Software Development Lifecycle as applied to Games Software
Software Planning and Project Management – Software Configuration Management as a standard.
The Agile Manifesto and its application to games development.
Object-Oriented Design Techniques and UML Modelling.
The syntax and semantics of class specification and Object creation.
Encapsulation
Association, Aggregation and Composition and the language concepts which facilitate these.
Inheritance, Sub-Typing, Polymorphism and the role of Dynamic Dispatch.
Language constructs and Standard Libraries for Memory Management
Software Architectures: SOLID Principles, Cohesion and Coupling
Data-Driven games development
Quality Assurance and Testing.
Introduction to Parametric Polymorphism.
Game Object representation and Object Models
Software Design Patterns for Games Architectures.

Program Structures and Execution Models
CPU vs GPU Software Engineering
Introduction to Concurrent Programming.
Common Optimisation Techniques
Test Driven Development.

Learning Activities

A Workshop will consist of a blended approach to deliver the theoretical concepts on software engineering applied to games mixed with practical software development exercises and Agile planning techniques in order to solve games development problems.
Further exercises – additional exercises for students to work on in their own time.
Directed learning – provides additional reading to enable practical work to be completed.
Learning materials can be accessed digitally via University Virtual Learning Environment (VLE).

Notes

In this module, students will build upon their foundational programming skills by introducing – in both a practical and theoretical manner – some of the fundamental ideas of software engineering, enabling students to develop and communicate designs for small scale games software systems.
As a group, students will apply the Agile Development methodology and its associated design techniques using Object-Oriented principles to produce solutions for games development scenarios using Object-Oriented Programming. Students will gain experience of Software Engineering techniques for the design, development and testing of game software and understand the impact of these techniques on the architecture of modern games application, using industry-led tools and techniques.