

### Summary Information

Module Code	5110COMP
Formal Module Title	Software Engineering for Games
Owning School	Computer Science and Mathematics
Career	Undergraduate
Credits	20
Academic level	FHEQ Level 5
Grading Schema	40

### Teaching Responsibility

LJMU Schools involved in Delivery
Computer Science and Mathematics

### Learning Methods

Learning Method Type	Hours
Lecture	22
Practical	33

### Module Offering(s)

Display Name	Location	Start Month	Duration Number Duration Unit
JAN-CTY	CTY	January	12 Weeks

### Aims and Outcomes

Aims	To develop the concepts of object oriented philosophy as applied to development for computer games.To explain models, tools and techniques of development process for game software. To explain formal principles of game software modelling.To provide skills in using software APIs relevant for the computer games industry.To provide students with knowledge, skills and experience in interactive application and games development.
------	--

**After completing the module the student should be able to:**

### Learning Outcomes

Code	Number	Description
MLO1	1	Apply the concepts of object oriented design to software development for computer games.
MLO2	2	Use UML modelling techniques and other object oriented design approaches to formally document game software design.
MLO3	3	Explain and implement game mechanics in game software development.
MLO4	4	Implement various techniques applicable to the games software development lifecycle using the object oriented programming paradigm.
MLO5	5	Utilise software engineering methodology for computer games software development.

### Module Content

Outline Syllabus	Introduction to Software Engineering: Game Software characteristics, software engineering paradigms, Software Development Models. Software Project Management: Scheduling, Milestones, Option and Risk Analysis. Source Control Management Systems. Technical Requirements Specification for Games. Principles of Object Oriented Analysis & Design: Use Case, Class, State, Activity, Communication and Sequence UML Diagrams. Role of Functional Programming. Object Oriented programming approach: - Object-oriented design- Decomposition into objects carrying state and having behaviour- Class-hierarchy design for modelling- Definition of classes: fields, methods, and constructors- Subclasses, inheritance, and method overriding- Dynamic dispatch: definition of method-call- Subtyping. - Subtype polymorphism; implicit upcasts in typed languages- Notion of behavioural replacement: subtypes acting like supertypes- Relationship between subtyping and inheritance- Object-oriented idioms for encapsulation- Privacy and visibility of class members- Interfaces revealing only method signatures- Abstract base classes- Data-Type Composition – Structs and Records. - Encapsulation, Coupling, Inheritance, and Polymorphism.- Templated/Generic Classes. Native Programming techniques: Pointers, Memory Allocation and De-allocation, Modelling the Game Economy via Objects: Arithmetic and Logical approaches to game economy. Behavioural design patterns for game design realisation. Overview of Game Engine Architecture approaches to modelling game objects. Overview of Data-driven game development; Hard and Soft Architectures and incorporating scripting into an OO game application. Game Software Testing and QA: Approaches to Analysis, Approaches to Testing, Test Plans, Testing Activities.
Module Overview	
Additional Information	In this module, students will build upon their foundational programming skills and develop object-oriented analysis and design skills in order to produce solutions to game development scenarios using object oriented programming. Students will gain experience of software engineering techniques for the design, development and testing of applications and understand the impact of these techniques on the architecture of modern computer games applications and how these are employed in industry in order for a development team to produce a robust solution.

### Assessments

Assignment Category	Assessment Name	Weight	Exam/Test Length (hours)	Module Learning Outcome Mapping
Centralised Exam	Examination	40	1.5	MLO1, MLO2
Report	Object Oriented Game App	60	0	MLO3, MLO4, MLO5

## Module Contacts

### Module Leader

Contact Name	Applies to all offerings	Offerings
Christopher Carter	Yes	N/A

### Partner Module Team

Contact Name	Applies to all offerings	Offerings
--------------	--------------------------	-----------