

Liverpool John Moores University

Title: SOFTWARE ENGINEERING FOR GAMES
Status: Definitive
Code: **5110COMP** (121232)
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
Abdenmour El-Rhalibi	

Academic Level: FHEQ5 **Credit Value:** 20 **Total Delivered Hours:** 56.5
Total Learning Hours: 200 **Private Study:** 143.5

Delivery Options

Course typically offered: Semester 2

Component	Contact Hours
Lecture	22
Practical	33

Grading Basis: 40 %

Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Exam	AS1	Examination	40	1.5
Artefacts	AS2	Object Oriented Game Application	60	

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.

Learning Outcomes

After completing the module the student should be able to:

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

Learning Outcomes of Assessments

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

Examination	1	2	
Object Oriented Game App	3	4	5

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 behavioral 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.

Learning Activities

Lectures – to deliver the theoretical concepts on software engineering applied to games.

Practical – Tutor-led practical session in the computer laboratory.

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 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.