

<b>Module Title:</b>	Games Engineering II
<b>Language of Instruction:</b>	English
<b>Credits:</b>	5
<b>NFQ Level:</b>	8
<b>Module Delivered In</b>	<a href="#">1 programme(s)</a>
<b>Teaching &amp; Learning Strategies:</b>	As well as traditional lectures students will undertake various laboratory exercises implementing various algorithms. They will be expected to participate in class on the materials covered. A term papers will involve a more in-depth study of the issues raised. Combination of lecture and laboratory sessions. Lectures will provide traditional theory. Laboratory sessions will employ formative practical assessment. Project work will be based on programming in C++11, C++14.
<b>Module Aim:</b>	Analyse, evaluate and implement design patterns and testing techniques which allow computational processes to be executed efficiently within digital games. Design and develop programs to perform tasks in parallel on single, multi-core and distributed CPU's and GPU's
<b>Learning Outcomes</b>	
<i>On successful completion of this module the learner should be able to:</i>	
LO1	Select, design, develop and implement testing strategies appropriate to games development
LO2	Evaluate, select and program appropriate design patterns within digital games
<b>Pre-requisite learning</b>	
<b>Module Recommendations</b>	
<i>This is prior learning (or a practical skill) that is recommended before enrolment in this module.</i>	
No recommendations listed	
<b>Incompatible Modules</b>	
<i>These are modules which have learning outcomes that are too similar to the learning outcomes of this module.</i>	
No incompatible modules listed	
<b>Co-requisite Modules</b>	
No Co-requisite modules listed	
<b>Requirements</b>	
<i>This is prior learning (or a practical skill) that is mandatory before enrolment in this module is allowed.</i>	
Game Engineering 1 or equivalent	

**Module Content & Assessment**

**Indicative Content**

**Testing**

Continuous Testing, Unit Testing, Automated Testing, Testing Tools, Coverage Techniques (Graph, Logic, Partitioning), White versus Black box testing, Managing Testing

**Game Programming Patterns**

Patterns in games development, pattern languages, examples and case studies.

**Game Engine Architecture**

Concepts and techniques utilised in implementing and using a game engine

Assessment Breakdown	%
Continuous Assessment	40.00%
Project	20.00%
End of Module Formal Examination	40.00%

**Continuous Assessment**

Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Written Report	Reading and criticism of industry/academic papers. Personal research/educational essay writing Complete a series of module practicals	1,2	40.00	Every Week

**Project**

Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Project	Complete a project which includes games programming patterns and concurrency	2	20.00	Sem 1 End

No Practical

**End of Module Formal Examination**

Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Formal Exam	End of year exam	1,2	40.00	End-of-Semester

SETU Carlow Campus reserves the right to alter the nature and timings of assessment

**Module Workload**

<b>Workload: Full Time</b>		
<i>Workload Type</i>	<i>Frequency</i>	<i>Average Weekly Learner Workload</i>
Lecture	12 Weeks per Stage	1.00
Laboratory	12 Weeks per Stage	4.00
Estimated Learner Hours	15 Weeks per Stage	4.33
Total Hours		125.00

**Module Delivered In**

Programme Code	Programme	Semester	Delivery
CW_KCCGD_B	<a href="#">Bachelor of Science (Honours) in Computer Games Development</a>	7	Mandatory