Module Title:	Games Engineering II
Language of Instruction:	English

Credits:	10

NFQ Level:	8

Module Delivered In	No Programmes

Teaching & Learning Strategies:	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 and learning concurrent and functional programming. Project work will be based on programming in C++11, C++14, Erlang and Haskell languages

Module Aim:	Analyse, evaluate and implement concurrent algorithms, 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

Learning Outcomes	
On successful completion of this module the learner should be able to:	
LO1	Select, design, develop and implement testing strategies appropriate to games development
LO2	Evaluate methods for synchronising concurrent processes and assess effects of concurrency in specific domains, applied to games development and title execution environments
LO3	Design algorithms that execute on multiple processes with core or processor affinity
LO4	Develop functional programs that express the logic of a computation (without defining flow) and integrate into a digital game
LO5	Evaluate, select and program appropriate design patterns within digital games

Pre-requisite learning	
Module Recommendations *This is prior learning (or a practical skill) that is recommended before enrolment in this module.*	
No recommendations listed	
Incompatible Modules *These are modules which have learning outcomes that are too similar to the learning outcomes of this module.*	
No incompatible modules listed	
Co-requisite Modules	
No Co-requisite modules listed	
Requirements *This is prior learning (or a practical skill) that is mandatory before enrolment in this module is allowed.*	
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
Concurrent programming Processes and Threads, Microchip Architectures (multicore, NUMA, hUMA) , GPU Architectures (SIMD, SIMT) Mutual Exclusion (Mutex, Semaphore), APIs (Pthreads, OpenMP, MPI, OpenCL, CUDA) and implementations
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
Functional Programming Definition, process creation, message passing, registering processes

Assessment Breakdown	%
Continuous Assessment	20.00%
Project	20.00%
Practical	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	1,2,3,4,5	20.00	Every Week

Project				
Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Project	Complete a project which includes games programming patterns and concurrency	2,5	20.00	Sem 1 End

Practical				
Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Practical/Skills Evaluation	Complete a series of module practicals	1,3,4,5	20.00	Every Second Week

End of Module Formal Examination				
Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Formal Exam	End of year exam	1,2,3,5	40.00	End-of-Semester

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

Module Workload

Workload: Full Time		
Workload Type	*Frequency*	*Average Weekly Learner Workload*
Lecture	30 Weeks per Stage	2.00
Laboratory	30 Weeks per Stage	2.00
Estimated Learner Hours	30 Weeks per Stage	3.00
	Total Hours	210.00