COAP: 2D Game Programming

Module Title:	2D Game Programming
Language of Instruction:	English

Credits:	5

NFQ Level:	6

Module Delivered In	1 programme(s)

Teaching & Learning Strategies:	The course material will be delivered by a mixture of traditional lectures and laboratory based lectures where learners will complete a series of incremental practical exercises towards building a game prototype. Towards the end of the semester, the students will work fulltime on a two week minor project that is undertaken in conjunction with the other second year course modules.

Module Aim:	Upon completion of this module, learners should be able to explain the main components of a game engine and use external libraries to implement a 2D game.

Learning Outcomes	
On successful completion of this module the learner should be able to:	
LO1	Use a game library API to develop properly architected short game prototypes.
LO2	Implement design patterns that are applicable to interactive applications.
LO3	Work as a member of a development team and use a version control system to manage source code in a team project.

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.*
Successful completion of year 1 or equivalent

COAP: 2D Game Programming

Module Content & Assessment

Indicative Content
2D Fundamentals The game loop, event loop and timers. Textures, sprites, basic sprite tranformations.
Data serialization formats Representing game data using data serialization formats. Using a parsing library to implement a level loader.
Input management Handling player input.
Collisions management Collision detection and response.
Steering behaviours NPC navigation with collision avoidance.
Game states Implementing game states and a HUD system.
Game design patterns Implementing a common game design pattern, for example object pool pattern.
Version control systems Committing, checking out, branching and merging.

Assessment Breakdown	%
Project	50.00%
Practical	50.00%

No Continuous Assessment

Project

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Project	Use a game library API to develop a short game prototype.	1	25.00	Sem 1 End
Project	Students will work as part of a team to create a playable game given a prescribed theme and gameplay specification. The assessment is staged horizontally across all 5 modules of this semester. This will be a studio style project, where teams work full time (without classes) over a period of two weeks.	1,2,3	25.00	Week 11

Practical

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Practical/Skills Evaluation	Participation in and completion of practical work. Students will work in-class on a series of practical exercises distributed over the second semester.	1,2	50.00	n/a

No End of Module Formal Examination

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

COAP: 2D Game Programming

Module Workload

Workload: Full Time		
Workload Type	*Frequency*	*Average Weekly Learner Workload*
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	Bachelor of Science (Honours) in Computer Games Development	4	Mandatory