1.1 Module 10: Procedural Programming and Generation

1.1.1 Headline information about the module

Module title	Procedural Programming and Generation
Module NFQ level (only if an NFQ level can be demonstrated)	7
Module number / reference	CT010
Parent programme(s) the plural arises if there are embedded programmes to be validated	BA (Hons) in Creative Technologies and Digital Art
Stage of parent programme	2
Semester (semester1/semester2 if applicable)	Semester 1
Module credit units (FET/HET/ECTS)	ETCS
Module credit number of units	10
List the teaching and learning modes	Full-Time, Direct Contact / Blended
Entry requirements (statement of knowledge, skill and competence)	Learners must have achieved the programme entry requirements
Pre-requisite module titles	Introduction to Programming Programming for Games
Co-requisite module titles	N/A
Is this a capstone module? (Yes/no)	No
Specification of the qualifications (academic, pedagogical and professional/occupational) and experience required of staff (staff includes workplace personnel who are responsible for learners such as apprentices, trainees and learners in clinical placements)	Lecturing staff must be qualified to a minimum of NFQ Level 9 in computer science, or related discipline, or hold an equivalent professional qualification. Experience in the computer science or programming industry would be desirable. Ideally, they would also hold a third level teaching qualification (e.g. the Griffith College Certificate in Education, Learning and Development).
Maximum number of learners per centre (or instance of the module)	25
Duration of the module	12 weeks
Average (over the duration of the module) of the contact hours per week	4
Module-specific physical resources and support required per centre (or instance of the module)	• Computer lab with capacity for 25 learners equipped with a projector • Access to Unity 3D software and user licences for Photoshop. • 3DsMax.

Analysis of required learning effort		
***Effort while in contact with staff**	**Minimum ratio teacher / learner**	**Hours**
Classroom and demonstrations	1:25	48
Monitoring and small-group teaching	-	-
Other (specify)	-	-
Independent Learning		
Directed e-learning (hours)		-
Independent learning (hours)		67
Assignment		135
Work-based learning hours of learning effort		-
Total Effort (hours)		250

Allocation of marks (within the module)					
	Continuous assessment	**Supervised project**	**Proctored practical examination**	**Proctored written examination**	**Total**
Percentage contribution	**100%**	**-**	**-**	**-**	**100%**

1.1.2 Module aims and objectives

The aim of this module is to introduce the learner to procedural creation in both game asset and world building. This aim is met by deconstructing existing procedurally driven game mechanics and repurposing them for use in the learner's game prototype(s). The learner builds an entirely procedurally driven game or tool capable of either creating levels or world component.

1.1.3 Minimum intended module learning outcomes

Upon successful completion of this module, the learners are able to:

(i) create code driven Unity prefabs;
(ii) explain the use of XML and Json;
(iii) create Unity Editor runtime code;
(iv) create efficient Editor Co-routines and Editor UI;
(v) explain 'mesh generation' and 'texture coordinates';
(vi) produce a procedurally driven game or tool;
(vii) design technical documentation which includes detailed code descriptions.

1.1.4 Rationale for inclusion of the module in the programme and its contribution to the overall MIPLOs

An understanding of procedurally generated UI and content provides the learner with a key game industry skillset. Learners are able to build reactionary / contextual content capable of transforming its appearance and nature, depending on certain variants. This may be seen in localisation, avatar creation / editing, or seasonal offers.

The minimum intended module learning outcomes address programme learning outcomes (i), (iii), (iv), (vi) and (xii).

1.1.5 Information provided to learners about the module

Learners are provided with a number of sources of information about this module, such as the induction session which presents learners with an overview of the modules. The induction session touches upon key areas of study such as the module aims, expectations and supports available.
At the commencement of each module, the learner is provided with a detailed overview of the module, the assessment strategy and schedule. The learner is then issued assignment briefs that fall in line with the deliverables outlined in the module objectives / outcomes.

The Learner Handbook, included with this submission, demonstrates how the learning in this module fits in to the overall structure of the programme. The handbook contains detailed module descriptors including teaching, learning and assessment strategies. Learners are provided with access to a learner Google account and to Google Classroom. Here, information regarding module descriptors, programme timetables and assessment information is uploaded. Google Classroom is for use by both learners and staff for the presentation of class notes and content as well as a point for assignments to be issued and submitted to.

1.1.6 Module content, organisation and structure

Learners are provided with notes, sample code and projects for each class. They are expected to review, extend and build upon this content independently, outside of lectures and lab sessions.

Intermediate programming (75%)

- XML Parsing
- JSon Objects and Parsing
- Save / Load data
- Co-routines
- Mesh generation
- Procedural objects

Middleware and game engines extended (25%)

- Scriptable objects
- Prefabs
- UI – Canvas advanced
- Editor scripts.

1.1.7 Module teaching and learning (including formative assessment) strategy

The module is delivered through a combination of lectures and lab sessions. The lecture sessions centre on detailed discussion of complex theory-based systems. The practical lab sessions facilitate learners to put material explored in lectures into practice, and to build working examples of procedurally generated objects, Unity editor scripts and tools and procedural environments and objects.

Activity	Teaching / Learning Strategy	Learning Environment
Lectures and demonstrations (48 hours)	Formal lectures and demonstrations on various aspects of procedural programming, including XML parsing, save / load data, mesh generation, scriptable object, procedural objects, etc.	College
Independent work (67 hours)	Self-directed work	College / Home

Assignment (135 hours)	Excludes time for continuous assessment assignments (worth 40% of overall grade which are done in class); assessment consists of: (i) production of a tool technical document (15%); (ii) building a functioning game tool (45%).	College / Home

1.1.8 Work-based learning and practice-placement

There is no work-based learning or practice-placement within this module.

1.1.9 E-Learning

Google Classroom acts as a reference point for the learner where all relevant information regarding the module is compiled. It also provides the learner with a messaging service between classmates and staff. Any changes or updates to module content is reflected on the platform along with a notification of change / messaging service. Google Classroom also accommodates for the submission of larger file types, a common feature of this programme. Learners also have access to additional academic material and supports through the Moodle virtual learning environment (VLE).

1.1.10 Module physical resource requirements

The module requires a computer lab with capacity for 25 learners, equipped with a projector, access to Unity 3D software and 3DsMax, as well as user licences for Photoshop.

1.1.11 Reading lists and other information resources

Primary Reading

Microsoft.NET C# Guide: https://docs.microsoft.com/en-us/dotnet/csharp/

Unity User Manual: https://docs.unity3d.com/Manual/index.html

Blackman, S, (2011) *Beginning 3D Game Development with Unity: All-in-One, Multi-Platform Game Development*. New York: Apress.

Schell, J. (2019) *The Art of Game Design: A Book of Lenses, 3*rd Edition. Boca Raton, Fla: A K Peters/CRC Press.

Secondary Reading

Creighton, R., (2010) *Unity 3D Game Development by Example Beginner's Guide*. Birmingham: PACKT Publishing.

Rabin, S. (2009) *Introduction to Game Development*, 2nd Revised Edition. Boston, MA: Course Technology/Cengage Learning

1.1.12 Specifications for module staffing requirements

For each instance of the module, one lecturer must be qualified to at least master's level (NFQ level 9) in computer science, or related discipline, or an equivalent professional qualification. Experience in the computer science or programming industry would be desirable.

Ideally, they would also hold a third level teaching qualification (e.g. the Griffith College Certificate in Education, Learning and Development).

1.1.13 Module summative assessment strategy

The module is assessed through both continuous assessment and project work. In regard to the former, learners are tasked with completing a short assignment each week at the end of class that is

based on their specific tool design. The second mode of assessment, project work, can be subdivided again into two components: the production of documentation in the form of a technical document tool and the building of a functioning game tool based on the plan outlined in that document. The marking criteria for assessments is set out in the Sample Assessment Handbook.

The assessed work breakdown can be seen in the table below.

No.	Description	MIMLOs	Weighting
1	In-Class Weekly Assignments	(i) to (v)	40%
2	Tool Technical Document	(iv), (v), (vii)	15%
3	Working Game Tool	(iii) to (vi)	45%

1.1.14 Sample assessment materials

Please see sample assessment supplementary document submitted with this proposal.