Module Code	CSU44052
Module Name	Computer Graphics
ECTS Weighting[1]	5 ECTS
Semester taught	Semester 1
Module Coordinator/s	Rachel McDonnell
Module Learning Outcomes	On successful completion of this module, students will be able to: LO1. Write graphical programs, using OpenGL or a similar graphics API, of moderate complexity; LO2. Select an object or scene representation, create a model using modelling software, and export this model for use in an interactive application; LO3. Discriminate between the different rendering choices for displaying objects, such as global or local illumination algorithms, and select the correct solution for the application area; LO4. Derive and solve the mathematical formulations that underpin the practical aspects of creating, animating and rendering objects and scenes; LO5. Critically appraise current computer graphics topics.
Module Content	The objective of this module is to equip the students with the fundamental understanding of the major elements of Computer Graphics and explore related areas including geometric modelling, rendering and animation. The main focus of the module is on the mathematics and algorithms used in the synthesis of computer graphics imagery and animation, and their practical application. Students are introduced to the standard architectures of modern graphical applications including details on the underlying hardware and low-level software components common to all such systems. The module is intended to enable students to bridge the gap between these low-level fundamental, components common to all computer applications, and the high-level abstract output in most interactive graphical applications. Students are also introduced to OpenGL, a modern high-level graphics API which is widely used for 3D Design and Visualisation, along with industry standard modelling software, and this software is used throughout the module to demonstrate concepts and to allow the students to develop their own 3D models, scenes and applications. Specific topics addressed in this module include: • An introduction to computer graphics; problem domain and applications; • Linear algebra - two and three dimensional transforms; geometric operations; hierarchical 3D transformations; • The computer graphics pipeline and the OpenGL API for 3D computer graphics; • Projection and viewing; window to viewport transformation; • Illumination models and rendering algorithms; colour, shading algorithms (Gouraud and Phong), local and global illumination;

[1] TEP Glossary

Teaching and Learning Methods

The module will be delivered through lectures and labs

Assessment Details[2]

Assessment Component	Brief Description	Learning Outcomes Addressed	% of total	Week set	Week due
Examination	2 hour written examination	LO3, LO4, LO5	80%	n/a	n/a
Labs	2 lab Assignments	LO1, LO2	20%	1	5

Reassessment Details

Examination (2 hours, 80%), Labs (2 Assignments, 20%)

Contact Hours and Indicative Student Workload

Contact Hours (scheduled hours per student over full module), broken down by:	**33 hours**
lecture	22 hours
laboratory	11 hours
tutorial or seminar	0 hours
other	0 hours
Independent study (outside scheduled contact hours), broken down by:	**92 hours**
preparation for classes and review of material (including preparation for examination, if applicable)	36 hours
completion of assessments (including examination, if applicable)	56 hours
Total Hours	**125 hours**

Recommended Reading List

Course Text:

Shirley, Peter et. al. *Fundamentals of Computer Graphics*. 2nd ed. Wellesley: A K Peters, 2005

Students are also encouraged to use appropriate texts and reference documentation, where necessary. For example:

1. *Introduction to Computer Graphics*, Foley, Van Dam, Feiner, Hughes and Phillips
2. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, (5th edition), Dave Shreiner, Mason Woo, Jackie Neider and Tom Davis
3. *Interactive Computer Graphics, A top-down approach with OpenGL* (6th edition), Edward Angel and Dave Shreiner

[2] TEP Guidelines on Workload and Assessment

Module Pre-requisites	**Prerequisite modules:** **Other/alternative non-module prerequisites:** High level of C or C++ programming is essential, at least freshman-level mathematics.
Module Co-requisites	
Module Website	Blackboard
Last Update	31/07/2019 by Rachel McDonnell