MODULE **DESCRIPTOR**

MODULE TITLE	Computer Graphics		
MODULE CODE	CO2409 (L5)	**CREDIT VALUE**	20 / 10 ECTS
SCHOOL	SCHOOL OF SCIENCE		

MODULE **AIMS**

1. To introduce the basic concepts and terminology of computer graphics.
2. To develop understanding of representations and techniques used in computer graphics.
3. To extend the students' basic mathematical skills to encompass the representation and manipulation of 3D geometry.
4. To apply the students' programming skills to the generation, manipulation and display of 2D and 3D images using graphics APIs such as DirectX or OpenGL.
5. To explain techniques for increasing realism of 3D graphics, e.g. texturing, lighting and shading.

6.To examine computer architectures supporting high-performance graphics.

MODULE **CONTENT**

Computer graphics are a vital part of modern multimedia applications, but are highly resource intensive. This module explores the mathematics, algorithms and hardware that attempt to meet the demands of real-time rendering of two- and three-dimensional computer graphics. It will introduce the basic principles of 2D computer graphics, and also describe the mathematical concepts needed to represent 3D geometry. The methods to realise 3D scenes in real-time are introduced through the use of graphic libraries. These fundamentals are built upon to produce more complex and realistic 3D graphics using textures, materials, shading, lighting and animation.

Graphics Hardware

A survey of graphics hardware, graphics libraries, hardware emulation
Computer architectures for graphics (GPUs, graphics pipelines, graphics/system memory, display architecture)

2D Graphics

Basic 2D geometry: points, lines, polygons and circles
Windows and clipping
Rendering: Pixels; colour and colour spaces; palettes; anti-aliasing; sprites

3D Graphics

Basic 3D geometry: points, lines, polygons, planes and spheres; meshes, normals; position and vector manipulation
Three-dimensional transformations: translation, rotation, scaling; combining transformations
Material properties: Diffuse, specular, and transparency (alpha) components; textures, texture mapping and UV coordinates
Lighting: parallel, point, and spot lights; colour; attenuation
Projection from 3D to 2D: parallel and perspective; cameras
Back-face culling, clipping and depth buffers
Polygon rendering: Vertex/face colours; lighting and shading; texturing and texture blending; alpha-blending

Graphics Programming using an API

Comparison of APIs: DirectX, OpenGL, Java3D
Preparation of 3D artwork: creating simple 3D objects; conversion of existing 3D objects
Initialisation: loading/preparing 3D geometry; positioning cameras and lights; preparing a window
Rendering: instantiating objects; back buffer rendering and swapping

Advanced topics: pixel and vertex shaders, animation

INTENDED **LEARNING OUTCOMES**

On successful completion of this module a student will be able to:

1. Explain the architectures used to support computer graphics.
2. Investigate and evaluate graphics algorithms.
3. Render 3D scenes using a modern graphic API.
4. Use relevant mathematical techniques (e.g. vector and matrix operations) to manipulate graphics objects.

TEACHING METHODS

Concepts will be introduced from a programming viewpoint and illustrated practically.

Lectures will present concepts illustrated with examples, and will be used to direct student reading and research in to relevant topics. Tutorials will be used to reinforce the topics covered in the lecture but will also allow the student to examine and evaluate potential approaches to topics introduced in the lectures.

Practical classes will be used primarily to develop the necessary skills for a graphics developer. Practical classes will be used to study the approaches taken by different graphic libraries. Students will develop simple applications that illustrate the material presented in the lectures.

A graphics engine will be used as a development platform for the more advanced graphics topics. The engine will provide a support framework allowing the students to concentrate on particular graphics topics. The engine itself will be studied to some degree as an example of practical graphics development.

The summative assessment is designed to test the students' comprehension and application of the concepts taught or discovered in a written examination and their practical skills in the use of graphics tools and techniques in a coursework assignment.

ASSESSMENT METHODS

This module is assessed through a Graphics Programming assignment (50%) and an examination (50%).