Liverpool John Moores University

Warning: An incomplete or missing proforma may have resulted from system verification processing

Title:	3D COMPUTER GRAPHICS
Status:	Definitive
Code:	**5108COMP** (121230)
Version Start Date:	01-08-2021
Owning School/Faculty:	Computer Science and Mathematics
Teaching School/Faculty:	Computer Science and Mathematics

Team	Leader
Sud Sudirman	Y
Chris Carter	

Academic Level:	FHEQ5	**Credit Value:**	20	**Total Delivered Hours:**	55
Total Learning Hours:	200	**Private Study:**	145		

Delivery Options
Course typically offered: Semester 1

Component	**Contact Hours**
Lecture	22
Practical	33

Grading Basis: 40 %

Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Artefacts	AS1	Shader-Driven Interactive Graphics Application using Procedural Animation.	100	

Aims

To provide mathematical knowledge essential in complex 3D graphics and animation.
To explain the key principles of 3D computer graphics.
To develop skills in 3D computer graphics operations using modern 3D graphical

API.
To explain GPU graphics programming using shaders.

Learning Outcomes

After completing the module the student should be able to:

1	Explain the real-time programmable rendering pipeline and the mathematical concepts underpinning each stage.
2	Assemble a 3D scene using polygonal mesh techniques.
3	Implement complex 3D affine transformations and procedural algorithms for transform control in a real-time GPU-accelerated interactive 3D graphical application.
4	Compare and contrast the key conceptual differences and algorithmic processes between offline rendering and real-time rendering.
5	Render complex 3D geometry using both local and physically based global illumination schemes in real-time using the programmable rendering pipeline.

Learning Outcomes of Assessments

The assessment item list is assessed via the learning outcomes listed:

Shader & Procedural Animation	1	2	3	4	5

Outline Syllabus

Mathematics principles:
- Solving linear inequalities.
- Discrete sampling and interpolation.
- Revision on Vectors and Matrices: Mathematical and geometric definitions of vector, Vectors vs. Points, Vector additions, subtraction, and multiplications, Vector dot product and cross product, unit vector, Transforms and Matrices. Applying these concepts in 3D space.
- 3D Coordinate space: Euclidean Geometry, 3D Cartesian Coordinates. Affine and coordinate system transformations.
- Theory of rotation in 3D and its implementation: Euler Angle, Tait-Bryan, Axis-Angle and Quaternion (including Complex Numbers).

Introduction to Programmable Graphics Pipeline using GPU Shaders.
- Pipeline Stages.
- Local, World, View and Screen Spaces.
- Simple triangle rasterization.

Polygonal representation
- Polygon Meshes: Vertices, Edge and Faces, Graphics primitives, Indexed triangle mesh, surface normal. Buffer formations and Topologies.
- DCC (Digital Content Creation) Content Importing and Data Parsing for Polygonal Meshes and Texture/Buffer Resources.

Texture mapping, including
- Diffuse, Specular and Normal mapping.
- Multi-Texturing.
- Magnification (point sampling, linear sampling)
- Minification and MIP Mapping
- Texture as a Resource, RenderTargets and RTT.
- Texture mapping implementation using shaders.

Illumination and shading model including,
- Basic radiometry
- Rendering in nature: Introduction to Physically-Based Lighting.
- Local Illumination vs. Global Illumination– Indirect vs. Direct Lighting.
- Simple BRDF-based lighting techniques.
- Phong Illumination Model: Ambient, Diffuse and specular lighting.
- Light sources: Direct, Point and Spot light sources.
- Normal vector calculations.
- Local illumination implementation using shaders.

Theory of viewing and projection in 3D and their implementation.
- Specifying output window, window aspect ratio, view frustum, field of view, and zoom.
- View Matrix
- Orthographic and Perspective Projection.
- Projection matrix.

Pipeline Control:
- Data Semantics, State Objects, Blend Equations, Z-Culling, Winding Order, Constant Buffers, Shader Parameters and Function Specifications.
- Sampling and anti-aliasing

3D Animation Techniques:
- Euler, Axis-Angle, Quaternion
- Key-Frame
- LERP and SLERP

Learning Activities

Lectures – to deliver the theoretical concepts on maths applied to interactive 3D computer graphics.

Practical – Tutor-led practical session in the computer laboratory.

Further exercises – additional exercises for students to work on in their own time.

Directed learning – provides additional reading to enable practical work to be completed.

Learning materials can be accessed digitally via University Virtual Learning Environment (VLE).

Notes

This module builds on the students' knowledge of 2D graphics and furthers their knowledge in 3D computer graphics, from the underlying mathematical principles to their application in the development of 3D computer games. The module uses a modern GPU-driven graphics API to demonstrate how complex 3D scenes can be constructed using a wide range of 3D graphical techniques. Students will be taught about the programmable pipeline, including shader implementations of lighting and texture calculations.