Week 9: Introduction to VFX
Part 2: Shaders and Materials

COMP270: Mathematics for 3D Worlds and Simulations

Objectives

- **Define** the function of a shader
- **Understand** how the graphics pipeline is implemented in UE4

Programmable Units

- The programmable units of the pipeline include:
  - Vertex Processor 
  - Tessellation Control
  - Tessellation Evaluation
  - Geometry Processor
  - Fragment Processor 
- Programs for these units are called shaders

Vertex and Fragment Shaders

- Required for any rendering to occur in D3D or OpenGL (other units are optional)
- Vertex shader: responsible for geometric transformations, deformations, and projection
  - Takes in exactly one vertex as input
  - Outputs one vertex
  - Typical operations include transformations and animation
- Fragment shader: responsible for the visual appearance of the surface
  - Takes in a pixel fragment (see rasterization)
  - Outputs colour and depth values
  - Typically used for shading calculations and texturing

AKA Pixel Shader

Shaders and Game Engines

- Most game engines abstract shaders into Materials
- These materials encapsulate a series of shaders and any other rendering states required to draw the effect
- These systems allow greater control for performance
- In addition, materials fit onto an artist's workflow

UE4 Material System

- This system uses a visual programming language to control the look of an object in the scene
- It consists of nodes called Material Expressions
- These nodes are simply bits of shader code designed to perform a single task, e.g.
  - Multiplication
  - Texture Sample (also know as a look-up)
  - Blends
- This allows you to build up a complex effect by chaining nodes together

UE4 Material Node Graph