

# SHADING LANGUAGE FOR 3D GRAPHICS

Khaleel Roshan Ali

## Abstract

3D API libraries such as Open Graphics Language (OpenGL) and Microsoft DirectX® (DX) hide the implementation algorithms of rendering, such as lighting, from the developer. This prohibits the developer from making any changes to these built-in rendering algorithms. Methods now exist that let developers override the built-in algorithms of 3D APIs by using Shading Languages such as OpenGL Shading Language (GLSL) or DirectX High Level Shading Language (HLSL). Graphics applications created using 3D APIs that do not override the built-in functionality are said to be using the “fixed functionality pipeline”, and the applications that do override the built-in functionality are said to be using the “programmable functionality pipeline”.

This thesis is a comparative analysis between the fixed and programmable functionality pipeline. To measure the difference between the two, tables are created and quality is compared by visually examining the results of the fixed functionality pipeline with that of the programmable functionality pipeline. The fixed functionality pipeline is programmed using OpenGL and the programmable functionality pipeline is programmed using the OpenGL Shading Language (GLSL). This thesis will focus on the OpenGL features: color material; per-vertex and per-pixel lighting; and textures. Basic knowledge of C/C++ and OpenGL is assumed and required to understand the code segments presented in this thesis.



Teapot with per-vertex and per-pixel lighting using a spot-light