
Advanced Modern OpenGL: Pipeline and Performance

Based on OpenGL 4.5, created on March 16, 2023

The logo for KDAB, featuring a stylized white lightning bolt icon to the left of the text "KDAB" in white, all on a blue background.

KDAB

The Qt, OpenGL and C++ Experts

- What is OpenGL?
- Terminology
- Qt helper classes

No part of this publication may be made available to others than the named licensee which is shown on every page by any means, electronic, mechanical, photocopying, recording or otherwise, or translated into any language, without the prior written permission of the publisher.

Performance, Profiling and Debugging

- Debugging OpenGL
- Synchronization Mechanisms
- Measuring
- Performance
 - Culling
 - Optimizing state changes
- Advanced Context Usage
- Optimising Memory Transfers
- Threading
- Buffering Schemes

The OpenGL Memory Model

- Introduction
- Memory coherence
- Memory barriers
- Atomics

Advanced Buffer Usage

- Instanced Rendering
- Uniform Buffer Objects
- Multi-draw Indirect
- Shader Storage Buffer Objects

Advanced GLSL

- Shaders and Shader Programs introspection
- Shader Subroutines
- Image Load/Store

Advanced Pipeline - Geometry

- Geometry Shaders
 - OpenGL Pipeline Revisited
 - Geometry Shader Basics
 - Shader Interface Blocks
 - Removing Geometry
 - Modifying Geometry
 - Augmenting Geometry Data
 - Changing the Primitive Type
- Tessellation
 - Introduction to Tessellation
 - Tessellation in Detail
 - Bezier Curves and Patches
 - Mesh Refinement

Advanced Pipeline - Computing

- Transform Feedback
 - Capturing Vertex Data
 - Setting Up Buffers
 - Using Transform Feedback
 - Feedback Rendering
 - Transform Feedback Object
- Compute Shaders

Reading List

- Recommended Books