
Programming with Qt 3D

Based on Qt 5.7

The logo for AKDAB, featuring a stylized blue 'A' with a lightning bolt and wavy lines, followed by the letters 'K', 'D', 'A', and 'B' in a blue, sans-serif font.

Overview of Qt 3D

- Feature Set
- Entity Component System? What's that?
- Hello Donut
- Qt 3D ECS Explained

Drawing with Qt 3D

- Introduction
- Geometries and Materials
- Lights

Input Handling with Qt 3D

- Picking
- Basic Input Handling
- Logical Devices
- Accumulators

Integrations and Helpers

- Dynamic QML Scenes
 - The NodeInstantiator Element
 - The EntityLoader Element
 - The SceneLoader Element
- Starting the Qt 3D Engine
- Painted Textures
- Integrating Qt Quick with Qt 3D

The Qt 3D Scene Graph

- The Simplified Graphics Pipeline
- Shader Programs
 - Introduction to Shader Programs
 - Configurable shader programs
- Geometry and Coordinate Systems
 - How to Specify Geometries?
 - Coordinate Systems
- Texturing
 - Texturing Basics
 - Texturing Geometry
 - Texture Sampling
 - Using Multiple Textures

The Qt 3D Frame Graph

- Viewports and Layers
- Transparency
- Selecting Shaders at Runtime
- Image-Based Techniques
 - Rendering to a Texture
 - Post-Processing Effects
 - Edge Detection
 - Television Effect
 - Gaussian Blur
 - Effect Chains

Animation with Qt 3D

- Key Frame Animations
- Combining Animations

Optional Topics

- Instanced Rendering
- Procedural Texturing
- Capturing the Rendering
- Level of Detail
- Displaying Text