Modern OpenGL with Qt

Based on Qt 5.2 and Open GL 4.4
Core Concepts
- Introduction to Modern OpenGL with Qt
- OpenGL Windows and Contexts
- The (basic) OpenGL pipeline
- Introduction to shader programs
- Vertex Buffer Objects
- Basic drawing calls and primitives
- Making life easier with Vertex Array Objects
- Configurable shader programs
- Module Summary

Transformations
- Introduction to Transformations
- Basic Transformations
- Combining Transformations
- Coordinate Systems
- View and Projection Transformations
- Module Summary

Lighting
- Physics of Light
- Diffuse Lighting
- Phong Lighting
- Flat Shading
- Per-fragment Phong Lighting
- Directional Lighting
- Multiple Lights
- Simple Toon/Cell Shading
- Spotlights
- Fog
- Module Summary

Texturing
- Texturing Basics
- Texturing Geometry
- Texture Sampling
- OpenGL Extensions
- Using Multiple Textures
- Specular Maps
- Baked Ambient Occlusion
- Bump Mapping
- Alpha Maps
- Environment Mapping
- Module Summary
- Procedural Texturing
  - Introduction to Procedural Texturing
  - Simple Functions
  - Anti-aliasing
  - Generating and Using Noise
- Image-Based Techniques
  - Rendering to a Texture
  - Post-Processing Effects
- Transparency
  - Depth peeling

- Geometry Shaders
  - OpenGL Pipeline Revisited
  - Geometry Shader Basics
  - Shader Interface Blocks
  - Removing Geometry
  - Modifying Geometry
- Tessellation
  - Introduction to Tessellation
  - Tessellation in Detail
  - Bezier Curves and Patches
  - Mesh Refinement
  - Case Study: Dynamic Tessellation

- Debugging OpenGL
- Synchronisation Mechanisms
- Measuring
  - Performance
    - Faster drawing
  - Culling
  - Optimizing state changes
  - Advanced context usage
  - Optimising Buffer Transfers
  - Threading
  - Buffering Schemes

- Shadowing
  - Shadow Mapping
  - Shadow Volumes
Introduction to Qt Quick

- Introduction to Qt Quick
- Composing User Interfaces
  - Graphical Elements
  - Text Elements
  - Anchor Layout
- User Interaction
  - Mouse Input
  - Touch Input
  - Keyboard Input
- Components

Animations and Data Presentation

- Animations
- Presenting Data
  - Arranging Items
  - Simple Data Models
  - Views
    - Customizing the Views
    - The Path View

Under the hood - Introduction to Qt

- Objects in Qt
  - Common Features of Qt's Object Model
  - Object Communication using Signals & Slots
  - Event Handling
- Core Classes
  - String Handling
  - Container Classes
  - File Handling
  - Variants and Properties
    - Properties

Qt Quick Integration

- Integrating QML with C++
  - Exporting C++ objects to QML
  - Creating new QML elements
    - Creating Non-GUI elements
    - Creating GUI elements
  - Using Custom Types
  - Attached Properties
  - Tips and Tricks
- Integrating OpenGL and Qt Quick 2
  - Qt Quick 2 Rendering
  - OpenGL under or over Qt Quick 2
  - Separating Renderers
  - Qt Quick 2 as the Slave