

State-of-the-Art OpenGL with Qt

This 5-day, hands-on course provides an introduction to modern OpenGL using programmable shader pipelines. It introduces the use of the Qt framework for infrastructure and support to quickly achieve highly efficient and visually appealing, state-of-the-art, interactive graphics applications. The class covers basic topics such as handling of geometry, graphics pipeline, transformations as well as advanced topics such as vertex, fragment and geometry shaders and their applications. Qt and C++ based Labs are designed to let participants experiment with OpenGL easily in order to deepen the understanding of the material.

Course Outline

This course will be roughly divided into four 90 minute sections and each section will target an even mix of lecture and labs. The goal is to give students plenty of time to experiment with the lab material.

Based on request the class is based on QGLWidget which is deprecated in Qt-5 and will be replaced in QOpenGLWidget in Qt-5.4. However, the class uses QGLWidget for Qt5 with the help of two classes OpenGLScene and OpenGLWindow which should make it very easy to adjust to different requirements such as varying OpenGL versions and the use of the newer QOpenGLWindow and QOpenGLWidget respectively.

Larger OpenGL Topics are organized into parts that are not necessarily equal in length as follows:

Part 1 – The Basics

- What is OpenGL
- Steps to Render a Scene with OpenGL
- Qt's OpenGL Support
- OpenGL Rendering Pipeline
- Shaders
- Labs: Contexts and Windows, Draw Triangles

Part 2 – Qt Refresher

- Lab: Wavefront .obj File Reader (Relevant to the Rest of the Course)

Part 3 – GLSL

- OpenGL Shading Language in Detail
- Lab: Rectangles, Grids

Part 4 – A Day at the Movies

- Vertices and Vectors
- Models and Cameras, the Model-View Transformation
- Demo Programs
- Matrices
- Depth and Culling
- Simplest Possible Light Model
- Lab: Three Dimensional Objects

Part 5 – Directing the Scene

- Primitives
- Indexed Drawing
- Vertex Array Objects (VAO)
- Handling Multiple Components of Objects
- Lab: Three Dimensional Scenes

Part 6 – Let there be Light

- Let There be Light (at the End of the Tunnel?)
- Lab: Diffuse Light Model, ADS (Phong) Light Model, Multiple Lights

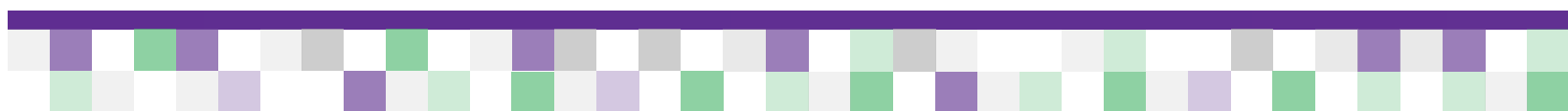
Course Availability:

On-site, open enrollment

Duration: 5 days

Course Prerequisites:

- Working knowledge of C++ programming
- Basic knowledge of geometry
- Previous experience with OpenGL and/or Qt is not required



Part 10 – More Advanced OpenGL

- More Advanced Topics (Spotlight, Fog, Smoke, Shadows, Cartoon Shading)
- Reflection and Refraction with Cubemaps
- Point Sprites
- Transform Feedback and Particle Systems
- Instanced Rendering

Part 11 – Timing is Everything

- QtOpenGLExtensions Module
- OpenGL Debugging and Timing with Qt

Part 12 – Qt and OpenGL

- Integration of OpenGL and QtQuick, Custom OpenGL QQuickItem
- QPainter on QOpenGLPaintDevice
- Qt3D

Part 13 – Optimizations

- Best Practices
- GLSL Optimizations
- Vertex Specification Best Practices
- Common Mistakes
- Convert GL 3.3 to GLES 2.0