Effective QML

This 3-day course focuses on design patterns and best practices to use when creating applications with QML user interfaces. Entering as intermediate QML programmers, students will leave with the knowledge of when to use various QML constructs in the best manner. This course focuses heavily on applications that are driven by C++ backends. Students will learn to create QML that is more declarative, maintainable and testable. The students will also learn about optimizing for startup and runtime performance as well as memory consumption.

Course Outline

Day 1
Course Introduction
• Welcome and Introductions
• Agenda
• Logistics
Review of Important Qt Concepts
• Qt Properties
  • Combination of Get/Set/Notify
  • Declaration of a Qt Property
  • Qt Property with Enum
  • Getting & Setting Qt Properties
  • Dynamic Properties
Building Blocks of QML
• Introspection
• Meta Object Compiler (MOC)
• QQuickItem
• QVariant
• QVariant and QML
• QVariant Containers
  • QVariantList
  • QList
  • QQmlListProperty
• Qt C++ Layer
Layered Design
• Introduction to Layered Design
• Building an Application as a Library
Questions and Answers Session

Lab 1

Day 2
Review and Solution of Lab 1
Extending/Creating New Items
• Dividing Code (QML) into Components
  • Creating New Items (Create New .qml File)
• Alias Properties
• Property Scope
  • Public Scope
  • Private Scope
  • Example: Public Members, Private Member
• Private Properties
  • Avoid Inheriting Public Members
Dynamic Creation of Items
• Creating Items Dynamically
  • Procedural Method
  • Declarative Method
• Procedural Creation
  • Procedural/Declarative Creation
• Declarative Creation
• Creating Multiple Items
  • Repeaters
Questions and Answers Session

Lab 2

Day 3
Review and Solution of Lab 2
User Input
• Keyboard Navigation and Focus
  • Input Focus
  • Focus Order
  • Focus property
  • Active Focus
  • FocusScope
• Handling Keyboard Input
  • Key-Specific Handlers
  • All Keys Handler
  • Key Event Propagation
  • Event Propagation is Convenient
• Multi-Touch
  • Common Multi-Touch Gestures
  • Handling Overlapping Touch Areas
• Mouse/Single-Touch
  • Tap
  • Double-Tap
  • Tap and hold
(continued on back)
Day 3 (continued)

QQmlFileSelector
- Basic Selector Concept
- Automatic Selectors
- Custom Selectors
- Custom Selectors via Environment
- Conflict Resolution Order
- Asset Management

Testing
- QTest Framework in QML
  - Overview
  - Adding a Test Suite
  - Implementing a Test
  - Add a Testing Main()
  - Running the Test
  - Simulating GUI Events
  - Mocking C++ Classes with QML
  - SignalSpy
  - Design for Testability

Course Wrap-up
- Summary of Key Points
- Where to Get More Information and Help
  - Examples and Reference Materials Included with Course Materials
  - Pointers to Other Materials & Other ICS Classes
- Optional Take Home Lab
- Questions & Answers