

Qt 6 Graphics (from QRhi perspective)

Laszlo & friends

19/11/2019

Qt Contributors Summit 2019

Berlin, Germany



Contents

In-progress and planned (or at least known) TODO items are in red throughout the presentation.

- Removing direct OpenGL usage in Qt 6.0
- >RHI + shader pipeline
- > Qt Quick, Qt Quick 3D
- > What else
- > Research items



Removing direct OpenGL usage

- No direct OpenGL usage in Qt 6.0 modules.
 - > Minimum: in Qt Quick and Qt Quick 3D
 - > Ideally: in most places



Removing direct OpenGL usage

- QOpenGL* conveniences (not context+functions) move out from QtGui/Widgets
 - > purge then reuse existing QtOpenGL module
 - incl. OpenGL paint engine and QOpenGLWidget and backingstore bits and examples and ... (domino effect?)
- > ANGLE to be removed from Qt



Qt Rendering Hardware Interface

- >Qt RHI
 - > Part of the QtGui module
 - > Private API may become QPA-style semi-public in 6.x (not 6.0)
- Qt Shader Tools
 - > qt-labs/qtshadertools must become qt/qtshadertools by 6.0

QRhi

QRhiBuffer
QRhiRenderBuffer
QRhiTexture
QRhiSampler
QRhiTextureRenderTarget
QRhiRenderPassDescriptor
QRhiShaderResourceBindings
QRhiGraphicsPipeline
QRhiComputePipeline
QRhiCommandBuffer
QRhiSwapChain

QRhiResourceUpdateBatch
QRhiClearColorValue
QRhiViewport
QRhiScissor
QRhiVertexInputLayout
QRhiShaderStage
QRhiShaderResourceBinding
QRhiTextureRenderTargetDescription
QRhiTextureUploadDescription

...

Backends

Direct3D 11.1 (incl. DXGI bits for swapchain/present)

Metal 1.2 (targeting CAMetalLayer)

Vulkan 1.0 (incl. swapchain/present)

OpenGL (2.1 / ES 2.0)

- D3D12 sort of in progress, but may not come before 6.x.
 - Not a priority for 6.0.
- WebGPU in 6.x.
 - If/when Emscripten support materializes.



QML (Qt Quick) application

QtQuick + scenegraph

QtShaderTools

QtGui

(QGuiApplication, QWindow, QOpenGLContext, QPainter, ...)

QPA

(QPlatformWindow, QPlatformOpenGLContext, QPlatformVulkanInstance, ...)

Platform plugin (implementing QPlatform*)

Backends/plugins for the platform plugin

(if the platform has multiple windowing systems and related APIs)

The OS and platform:
Win32/xcb/Wayland/Cocoa/UIKit/Android, ...
EGL/GLX/WGL, DRM+GBM/DRM+EGLStream,
Screen, OpenWF, proprietary stuff, ...

QtRHI

QRhi backends

OpenGL

Vulkan

Metal

Direct 3D

Qt

Vulkan GLSL shader code Compile to SPIR-V Generate reflection data Translate to GLSL/HLSL/MSL Optional: invoke compiler on generated source to ship bytecode instead Option 1: offline For example: Option 2: at qsb --glsl "100 es,120,150" --hlsl runtime 50 --msl 12 color.vert -o QShaderBaker color.vert.qsb QShader (generated on the fly or loaded from .qsb)

glslang

SPIRV-Cross



RHI status

- > Qt 5.15 adds Metal on iOS and Vulkan on Wayland.
- > QRhi and QShader API seen as fairly stable.
 - > No big changes expected (or desired) short-term.
 - Some platform things perhaps
 - > D3D11 on UWP/WinRT?
 - D3D11 on Windows 7? (broken atm)
- > Focus will be on clients (Quick, Quick3D, paint engine, ...).

Qt Quick

- > RHI port ships and is opt-in in Qt 5.14.
- > Remove direct OpenGL code path in 6.0.
 - > Also involves breaking src.compat.
 - > QSGMaterialShader, QQuickWindow::createTextureFromId(), ... to be removed.
 - > Some classes/types will continue to be tied to OpenGL (== OpenGL through QRhi)
 - > QQuickFramebufferObject



Qt Quick

- The experimental D3D12 scenegraph context plugin added in 5.8 is to be removed.
- >QML Profiler scenegraph data collection needs adjusting.



Qt Quick

- > Implications for Design Studio architecture. (out of process rendering, qmlpuppet, etc.) TBD.
- To be be investigated what we can do with:
 - > QQuickRenderControl
 - > QSGEngine
 - > QQuickWidget
 - **>** ...



Qt Quick 3D

- > Must be ported to QRhi and the new shader pipeline by 6.0
 - > Direct OpenGL code path to be removed.
- > Further unification of the QQ and QQ3D scenegraphs



QPainter

- > Proof-of-concept paint engine using QRhi is in progress
- > What is this good for?
 - > QQuickPaintedItem ("FBO" mode)
 - → Draw widgets with it? (hello -graphicssystem rhi ⊕)
 - > QRhiWidget?
 - **>** ...
 - Open for ideas.



Some research items for 6.x

- > D3D12, WebGPU backends
- > Colorspaces, HDR, ...
- > Better approach for compressed textures: integrate Basis Universal?
- Threaded command list building (and applications of it in Quick/Quick3D)
- > Shader/material node system.
 - > C++ graph. Visual editing. Translate to source, or even directly to SPIR-V?

Chank	Time	Assembly Hall	1.3.14 (Zoo)	1.1.9 (Landsberger Allee)	1.1.8 (Greifwalder Str)
	9:00 - 9:40	OtCoro	Qt Marketplace	Rethinking serialization for Qt6	
	(Yc			Clang-based cpp parser for lupdate	Remote display of Qt applications in Qt 6
	10:30 - 10:50			Coffee Break	
	10:50 - 11:30	QtQml	Platform-specific APIs in Qt	Future of QStyle for widgets and controls	
	11:40 - 12:20		Refurbishing Qt Widget internals	Available, hidden and missing gems on the way of using Qt on embedded devices	
	12:20 - 13:20			Lunch Break	
	13:20 - 14:00	QtGUI, RHI, and 3D	Qt Wayland Client and extensions	Qt for Python and beyond	
	14:10 - 14:50		Improve the contributor experience of the Qt project	High DPI	