Qt World Summit 2017
October 10-12 | Berlin, Germany

Qt 3D as a Runtime Enabler

A Case Study for Implementing High-Level 3D UI Runtimes on top of Qt 3D

László Agócs (lagocs / @alpqr)
Graphics & Multimedia
The Qt Company, Oslo, Norway

The Qt Company

Qt World Summit 2017
October 10-12 | Berlin, Germany

~~Qt 3D as a Runtime Enabler~~

~~A Case Study for Implementing High-Level 3D UI Runtimes on top of Qt 3D~~

László Agócs (lagocs / @alpqr)
Graphics & Multimedia
The Qt Company, Oslo, Norway

The Qt Company

Qt World Summit 2017
October 10-12 | Berlin, Germany

Building the next generational Qt 3D Studio runtime on top of Qt 3D

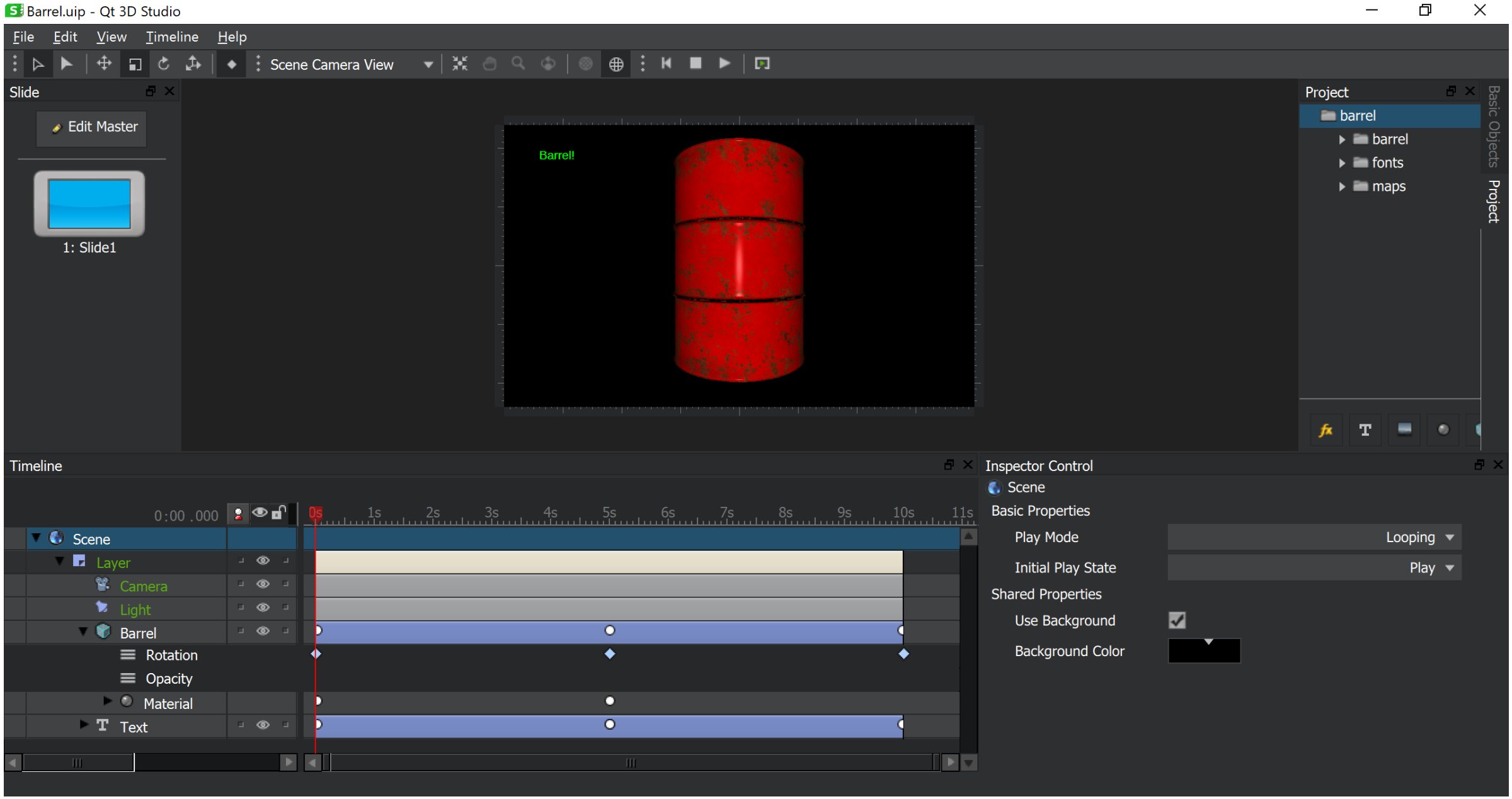
László Agócs (lagocs / @alpqr)
Graphics & Multimedia
The Qt Company, Oslo, Norway

The Qt Company

Anark Gameface -> NVIDIA UI Composer -> NVIDIA DRIVE Design -> **Qt 3D Studio**

Qt 3D Studio

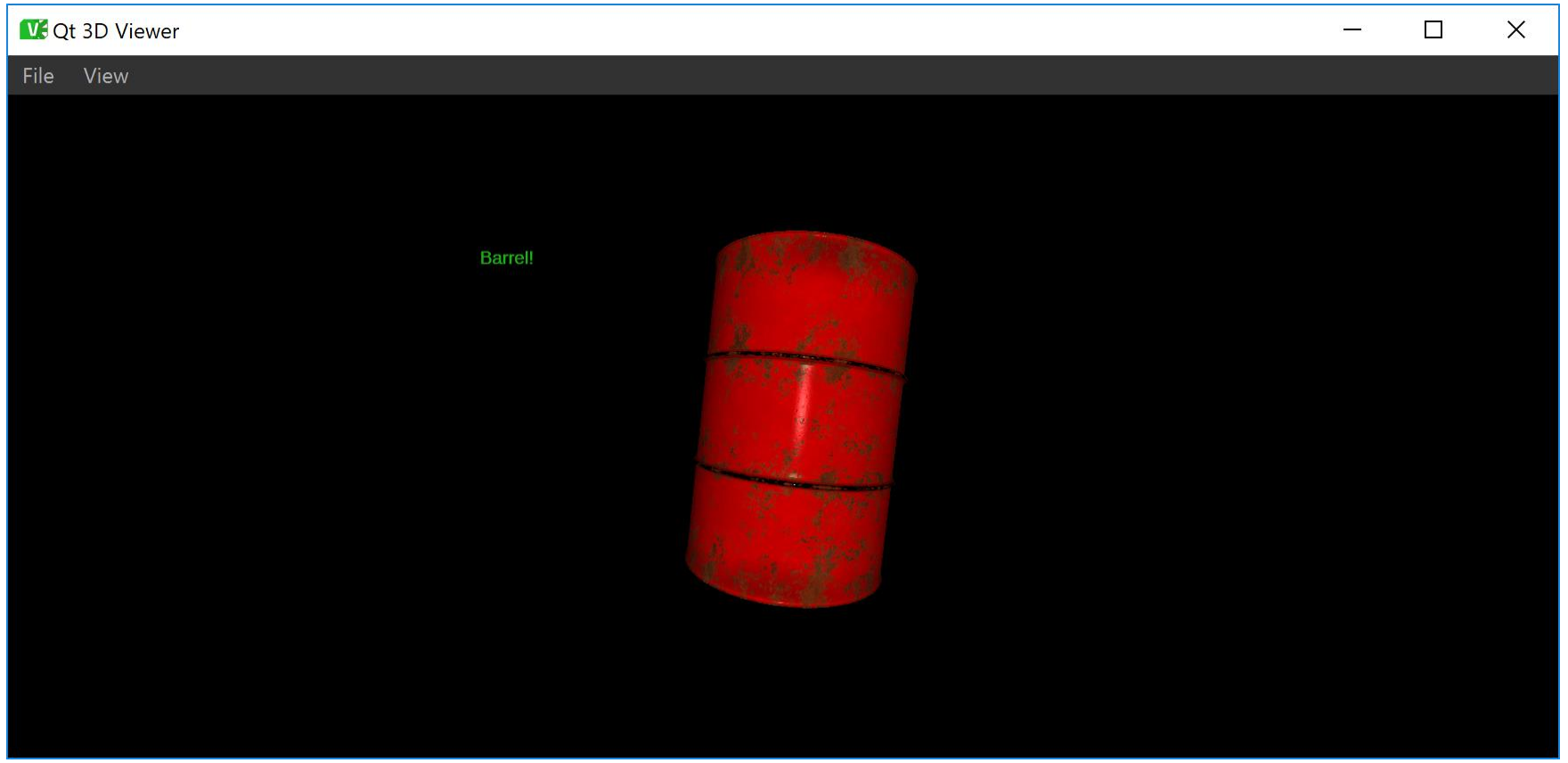
- Designer driven WYSIWYG 3D design tool
  - Scene
    - Layers (2D composition, blending)
      - Camera, Light, Model, Material, …
  - States (slides)
    - Animations
    - Property changes
- Desktop application (Windows, Linux, Mac) + Runtime


Barrel.uip - Qt 3D Studio
File
Edit
View
Timeline
Help
Scene Camera View
Slide
Edit Master
1: Slide1
Barrel!
Project
barrel
barrel
fonts
maps
Basic Objects
Project
Timeline
0:00 .000
0s
1s
2s
3s
4s
5s
6s
7s
8s
9s
10s
11s
Scene
Layer
Camera
Light
Barrel
Rotation
Opacity
Material
Text
Inspector Control
Scene
Basic Properties
Play Mode
Looping
Initial Play State
Play
Shared Properties
Use Background
Background Color

Qt 3D Studio Runtime

- Lightweight, C++, OpenGL
- Generally requiring OpenGL (ES) 3 [subject to change, some level of GLES 2.0 support coming]
- Original contribution came with a strong NVIDIA focus, understandably
  - Now fixed up to work on AMD, Intel, …
- Remains “foreign” (non-Qt) mostly, but:
- Certain pieces ported to Qt to improve portability (e.g. OpenGL usage goes through QOpenGLFunctions now)

Runtime = Engine + APIs (QWindow, QWidget integration) + QML elements (+ viewer application)



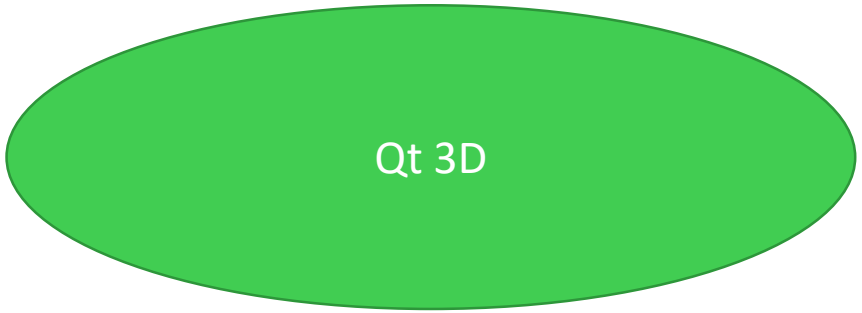
Can we do better?

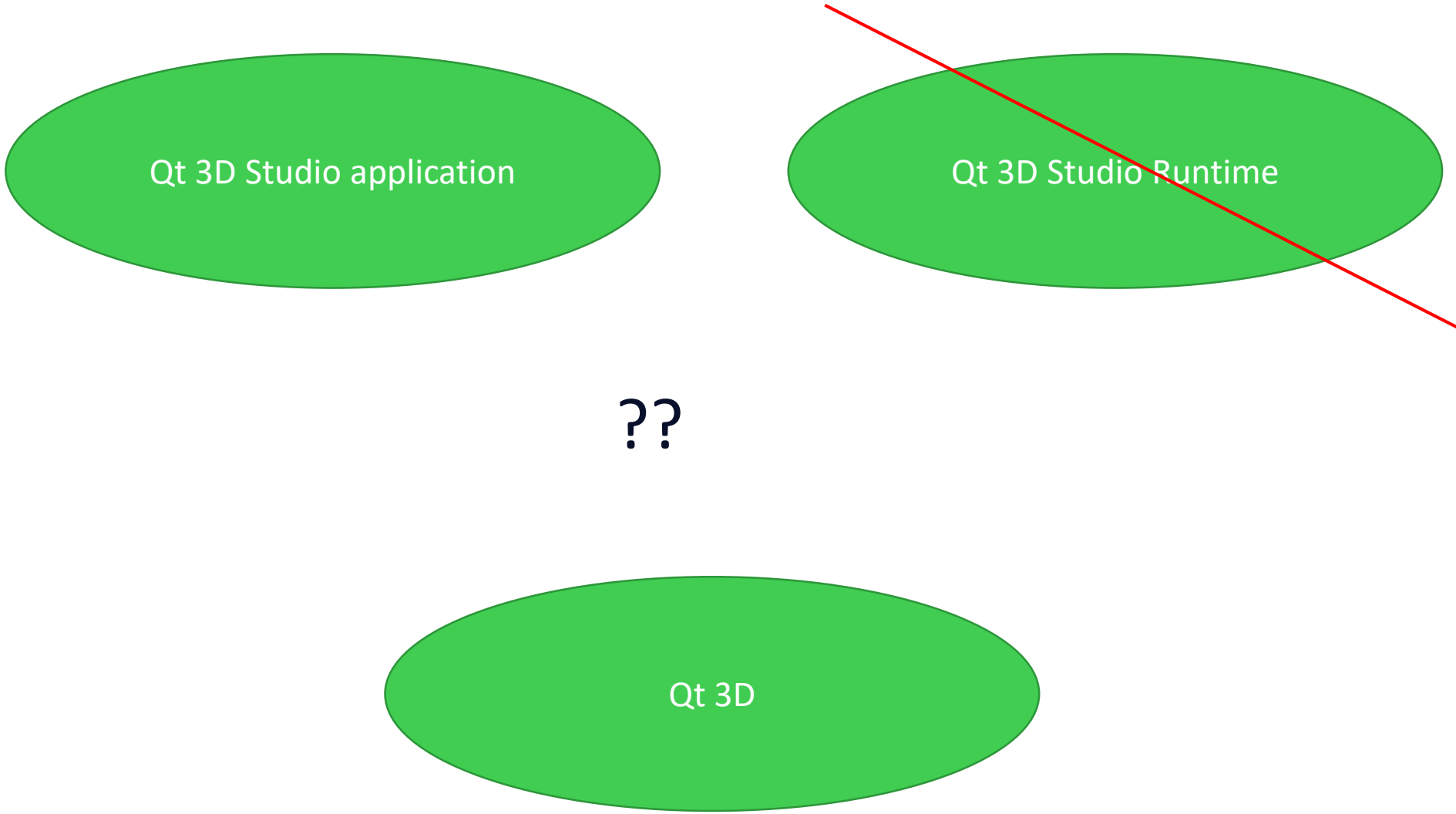
- Why is this good?
  - Lightweight, stable.
  - It exists, right here, right now.
  - CPU and memory usage better than corresponding scenes built via Qt 3D as of Qt 5.9.1.
- Why is this not so good?
  - Yet another 3D engine.
  - Which is not even Qt-based.
  - Portability issues and concerns.
  - Not necessarily future proof. (think graphics APIs other than OpenGL)

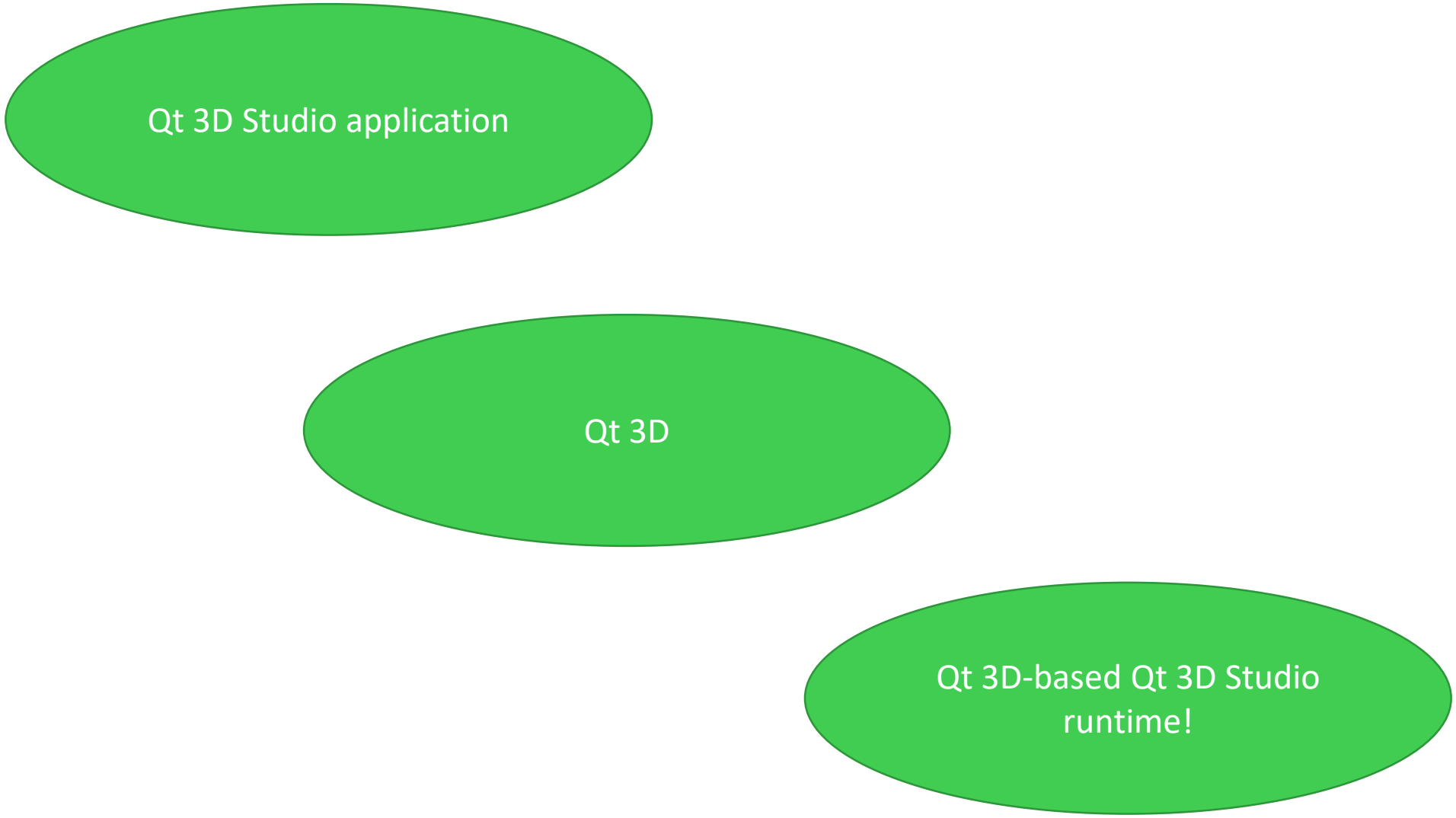
Can we do better?

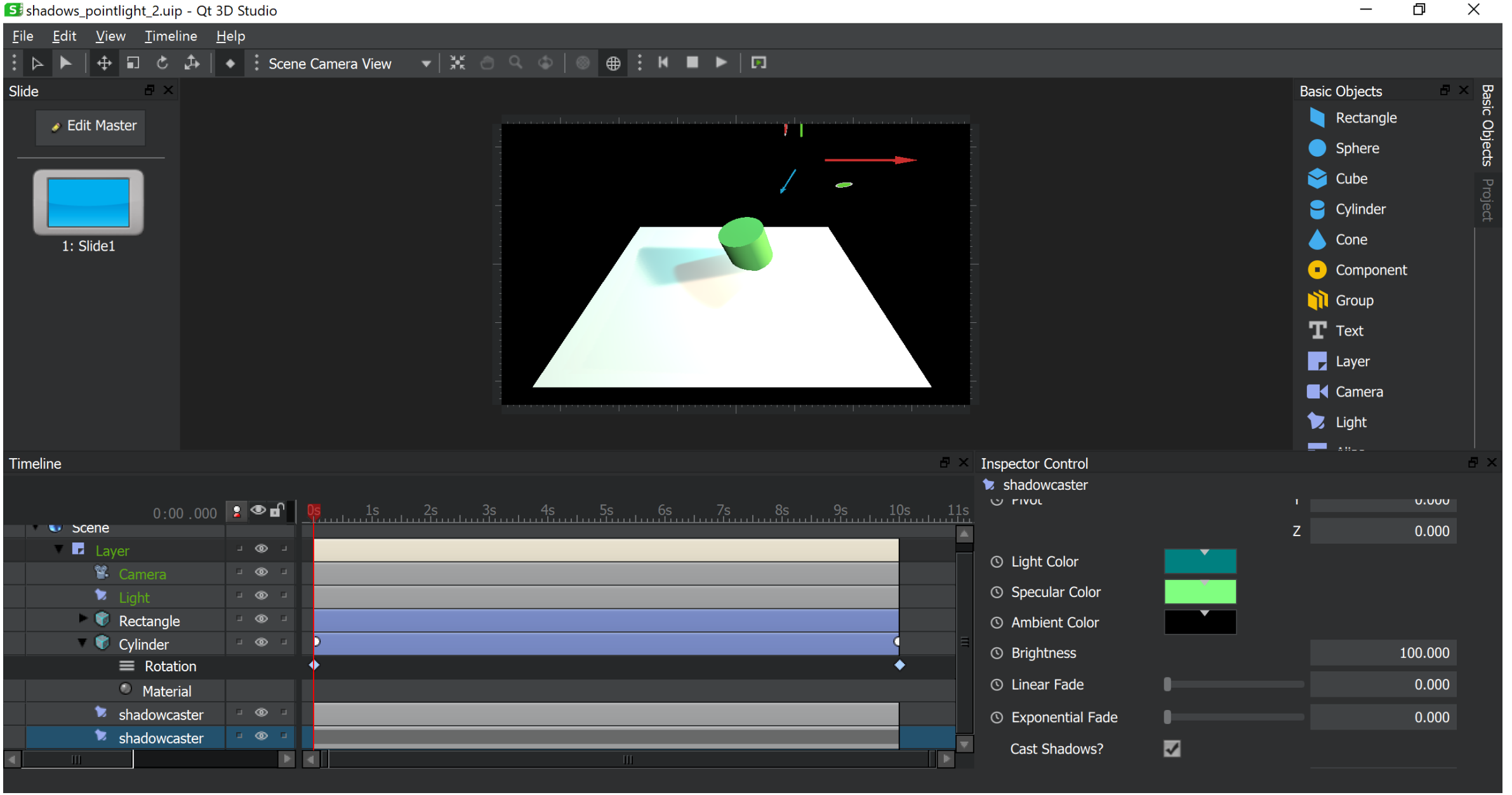
- Qt 3D can be optimized further.
  - “Task force” looking into reducing CPU and memory use.
    - Share, track dirty, cache, avoid copying, etc.
    - Mostly inside qt3d + few small optimizations in qtbase.
  - Great progress.
  - And there’s more to come.
    - May affect public API in Qt3DCore, let’s see.
  - Bonus: these optimizations help any Qt 3D application, not just 3D Studio based ones.

So how do we bridge these worlds?




Qt 3D Studio application
Qt 3D Studio Runtime
??
Qt 3D


Qt 3D Studio application
Qt 3D
Qt 3D-based Qt 3D Studio runtime!

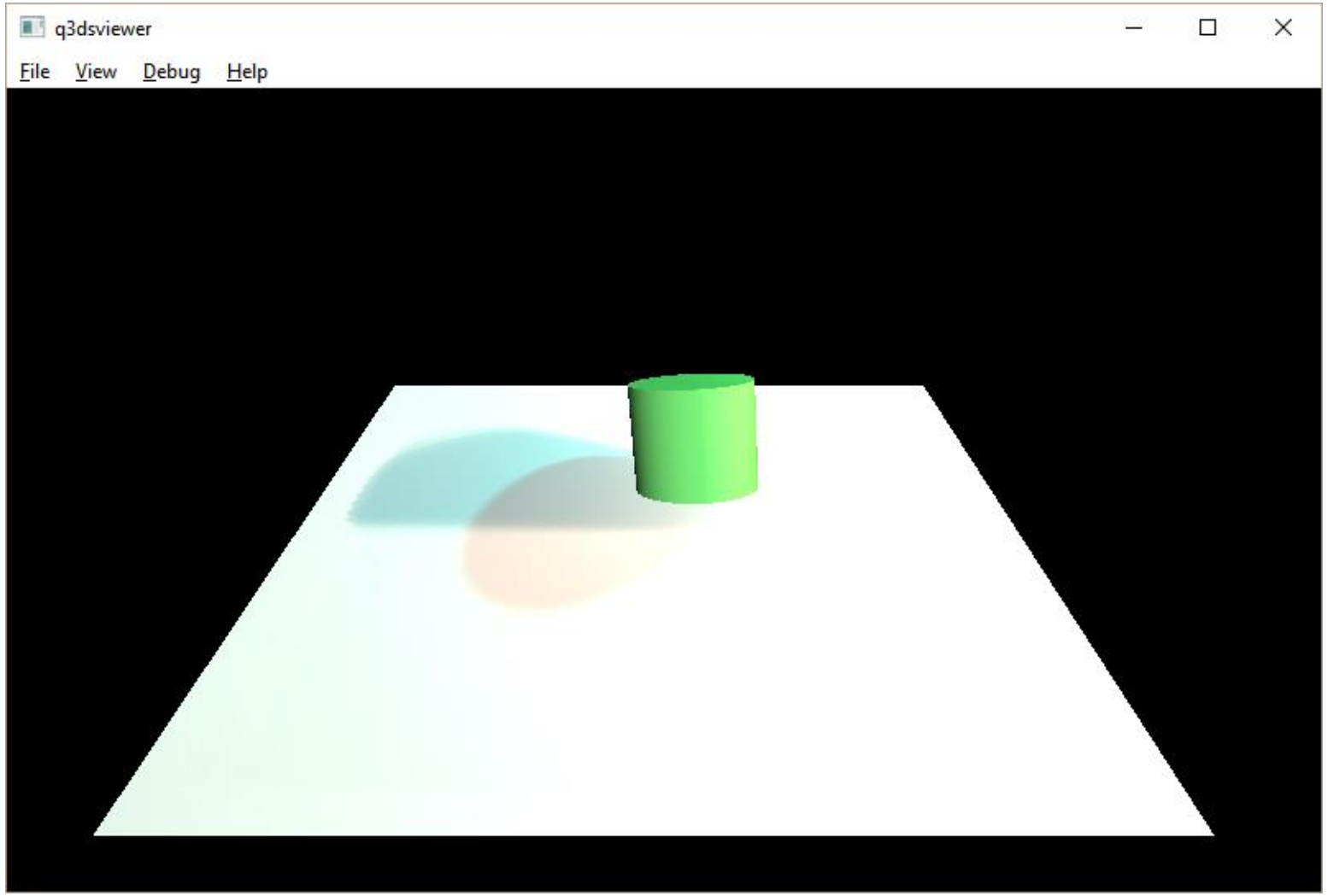

shadows_pointlight_2.uip - Qt 3D Studio
File Edit View Timeline Help
Scene Camera View
Slide
Edit Master
1: Slide1
Basic Objects
Rectangle
Sphere
Cube
Cylinder
Cone
Component
Group
Text
Layer
Camera
Light
Timeline
0:00 .000
Scene
Layer
Camera
Light
Rectangle
Cylinder
Rotation
Material
shadowcaster
shadowcaster
Inspector Control
shadowcaster
Z
0.000
Light Color
Specular Color
Ambient Color
Brightness
100.000
Linear Fade
0.000
Exponential Fade
0.000
Cast Shadows?

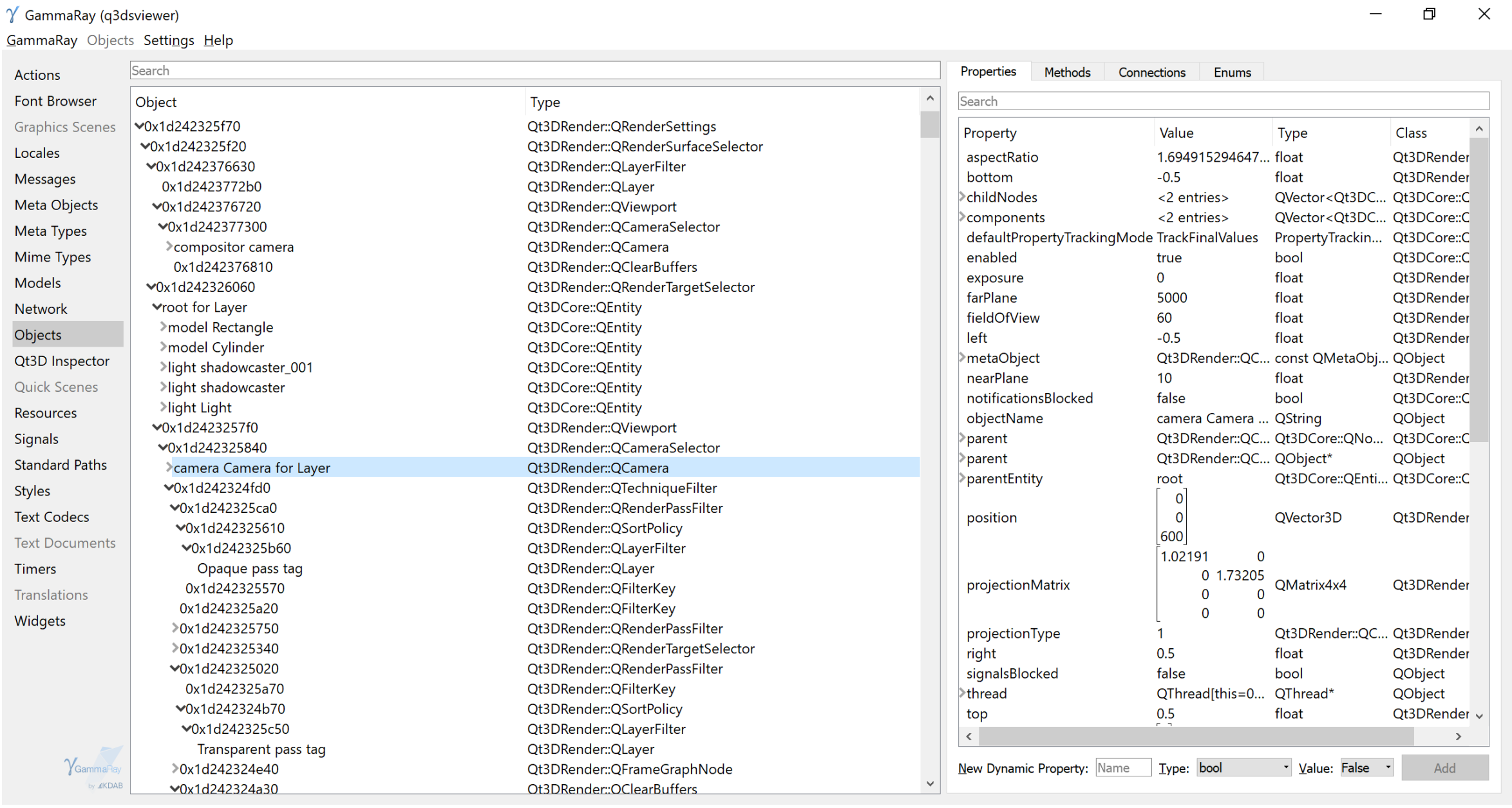
```
<?xml version="1.0" encoding="UTF-8" ?>
<UIP version="3" >
        <Project >
                <ProjectSettings author="" company="" presentationWidth="800" presentationHeight="480" maintainAspect="False" />
                <Graph >
                        <Scene id="Scene" >
                                <Layer id="Layer" >
                                        <Camera id="Camera" />
                                        <Light id="Light" />
                                        <Model id="Rectangle" >
                                                <Material id="Material" />
                                        </Model>
                                        <Model id="Cylinder" >
                                                <Material id="Material_001" />
                                        </Model>
                                        <Light id="shadowcaster_001" />
                                        <Light id="shadowcaster" />
                                </Layer>
                        </Scene>
                </Graph>
                <Logic >
                        <State name="Master Slide" component="#Scene" >
                                <Add ref="#Layer" />
                                <Add ref="#Camera" />
                                <Add ref="#Light" castshadow="False" lighttype="Directional" />
                                <State id="Scene-Slide1" name="Slide1" >
                                        <Add ref="#Rectangle" name="Rectangle" position="-5.77344 -34.641 0" rotation="53.5 0 0" scale="6.30691 5.36799 1" sourcepath=
                                        <Add ref="#Material" />
                                        <Add ref="#Cylinder" name="Cylinder" position="26.809 49.9481 25.2529" rotation="-52.8354 30.8347 -4.94139" scale="1.16992 0.9
                                                <AnimationTrack property="rotation.x" type="EaseInOut" >0 -52.8354 100 100 10 50 100 100</AnimationTrack>
                                                <AnimationTrack property="rotation.y" type="EaseInOut" >0 30.8347 100 100 10 30.8347 100 100</AnimationTrack>
                                                <AnimationTrack property="rotation.z" type="EaseInOut" >0 -4.94139 100 100 10 -4.94139 100 100</AnimationTrack>
                                        </Add>
                                        <Add ref="#Material_001" diffuse="0.501961 1 0.501961" />
                                        <Add ref="#shadowcaster_001" name="shadowcaster" castshadow="True" lightdiffuse="1 0.662745 0.501961" lighttype="Point" positi
                                        <Add ref="#shadowcaster" name="shadowcaster" castshadow="True" lightdiffuse="0 0.501961 0.501961" lightspecular="0.501961 1 0.
                                </State>
                        </State>
                </Logic>
        </Project>
</UIP>
```

What can we do with this?

- Why not parse those .uip files and build a Qt 3D scene?
- Animations are keyframe based -> good that AnimationClip API got introduced in Qt3D in 5.9.

Like this! (yes, that's Qt 3D)



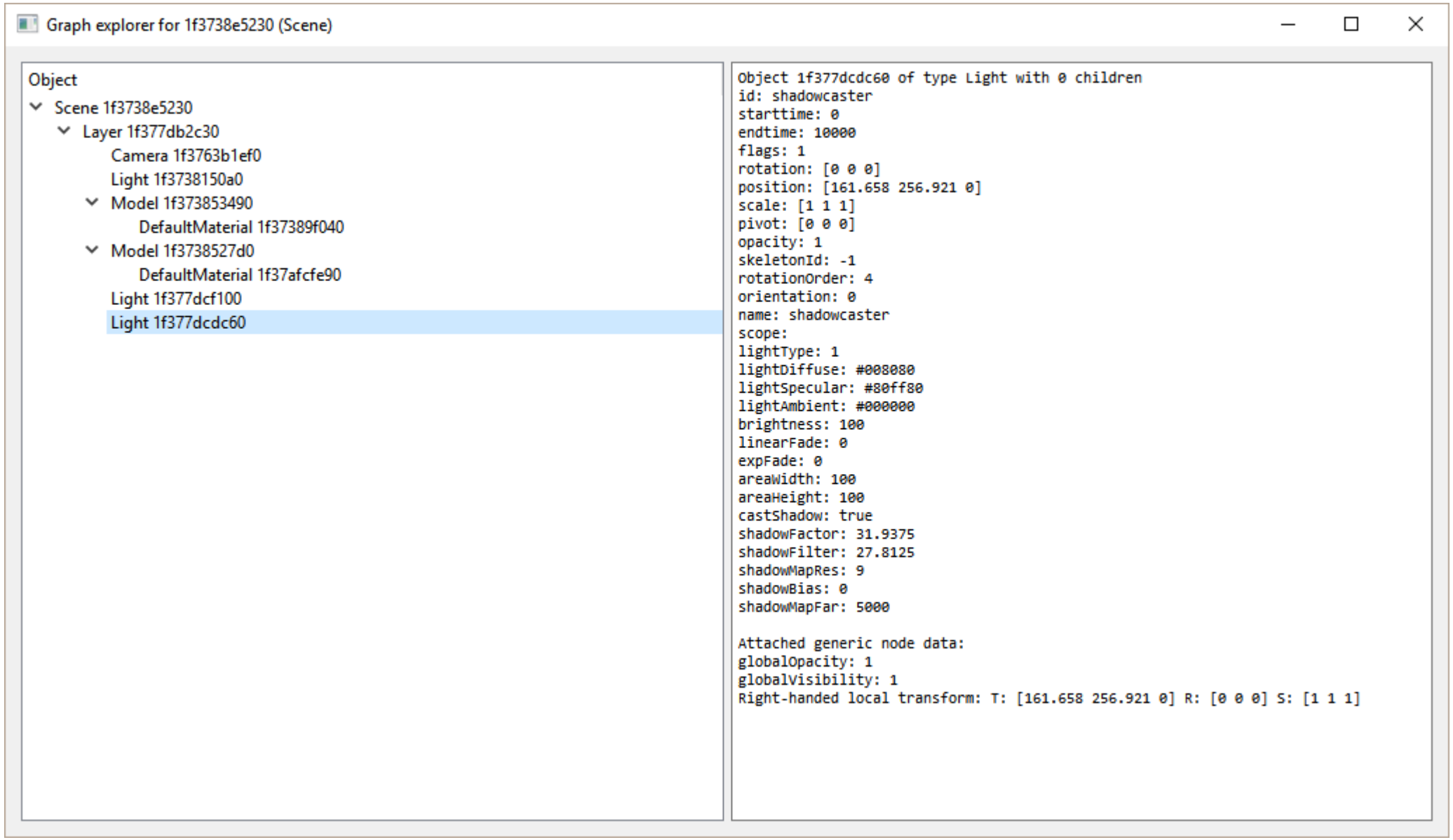

GammaRay (q3dsviewer)
GammaRay Objects Settings Help
Actions
Font Browser
Graphics Scenes
Locales
Messages
Meta Objects
Meta Types
Mime Types
Models
Network
Objects
Qt3D Inspector
Quick Scenes
Resources
Signals
Standard Paths
Styles
Text Codecs
Text Documents
Timers
Translations
Widgets
Search
Object
Type
0x1d242325f70 Qt3DRender::QRenderSettings
0x1d242325f20 Qt3DRender::QRenderSurfaceSelector
0x1d242376630 Qt3DRender::QLayerFilter
0x1d2423772b0 Qt3DRender::QLayer
0x1d242376720 Qt3DRender::QViewport
0x1d242377300 Qt3DRender::QCameraSelector
compositor camera Qt3DRender::QCamera
0x1d242376810 Qt3DRender::QClearBuffers
0x1d242326060 Qt3DRender::QRenderTargetSelector
root for Layer Qt3DCore::QEntity
model Rectangle Qt3DCore::QEntity
model Cylinder Qt3DCore::QEntity
light shadowcaster_001 Qt3DCore::QEntity
light shadowcaster Qt3DCore::QEntity
light Light Qt3DCore::QEntity
0x1d2423257f0 Qt3DRender::QViewport
0x1d242325840 Qt3DRender::QCameraSelector
camera Camera for Layer Qt3DRender::QCamera
0x1d242324fd0 Qt3DRender::QTechniqueFilter
0x1d242325ca0 Qt3DRender::QRenderPassFilter
0x1d242325610 Qt3DRender::QSortPolicy
0x1d242325b60 Qt3DRender::QLayerFilter
Opaque pass tag Qt3DRender::QLayer
0x1d242325570 Qt3DRender::QFilterKey
0x1d242325a20 Qt3DRender::QFilterKey
0x1d242325750 Qt3DRender::QRenderPassFilter
0x1d242325340 Qt3DRender::QRenderTargetSelector
0x1d242325020 Qt3DRender::QRenderPassFilter
0x1d242325a70 Qt3DRender::QFilterKey
0x1d242324b70 Qt3DRender::QSortPolicy
0x1d242325c50 Qt3DRender::QLayerFilter
Transparent pass tag Qt3DRender::QLayer
0x1d242324e40 Qt3DRender::QFrameGraphNode
0x1d242324a30 Qt3DRender::QClearBuffers
Properties
Methods
Connections
Enums
Search
Property
Value
Type
Class
aspectRatio 1.694915294647... float Qt3DRender
bottom -0.5 float Qt3DRender
childNodes <2 entries> QVector<Qt3DC... Qt3DCore::C
components <2 entries> QVector<Qt3DC... Qt3DCore::C
defaultPropertyTrackingMode TrackFinalValues PropertyTrackin... Qt3DCore::C
enabled true bool Qt3DCore::C
exposure 0 float Qt3DRender
farPlane 5000 float Qt3DRender
fieldOfView 60 float Qt3DRender
left -0.5 float Qt3DRender
metaObject Qt3DRender::QC... const QMetaObj... QObject
nearPlane 10 float Qt3DRender
notificationsBlocked false bool Qt3DCore::C
objectName camera Camera ... QString QObject
parent Qt3DRender::QC... Qt3DCore::QNo... Qt3DCore::C
parent Qt3DRender::QC... QObject* QObject
parentEntity root Qt3DCore::QEnti... Qt3DCore::C
position 0 0 600 QVector3D Qt3DRender
projectionMatrix 1.02191 0 0 0 1.73205 0 0 0 QMatrix4x4 Qt3DRender
projectionType 1 Qt3DRender::QC... Qt3DRender
right 0.5 float Qt3DRender
signalsBlocked false bool QObject
thread QThread[this=0... QThread* QObject
top 0.5 float Qt3DRender
New Dynamic Property: Name Type: bool Value: False Add

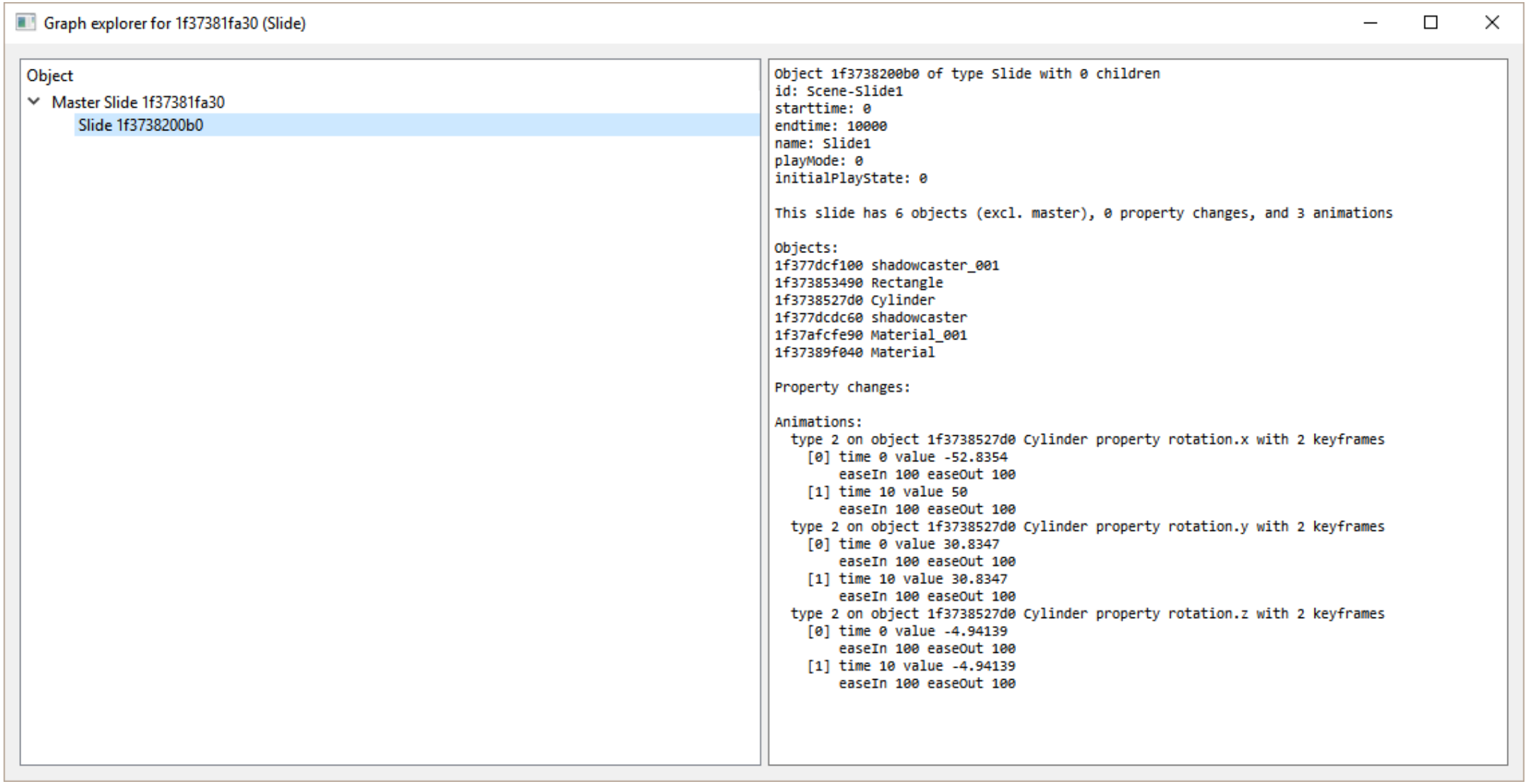
Introducing *qt3dstudio/qt3d-runtime* (on Qt Project Gerrit) a.k.a. Dragon3 a.k.a. Qt 3D Studio Runtime 2.0

- Project to produce a GPLv3/Commercial library and viewer application to load and “run” the .uip files Qt 3D Studio generates.
- Fully Qt and Qt 3D based. However, uses the same assets, including shaders (and so lighting system), as Qt 3D Studio.
- Qt 3D usage separated from the scene & logic graphs.
  - Could use the library to load and inspect .uip documents, without doing anything rendering related.
- https://codereview.qt-project.org/gitweb?p=qt3dstudio%2Fqt3d-runtime.git;a=summary

Phase 1: The simple one: Let's parse XML

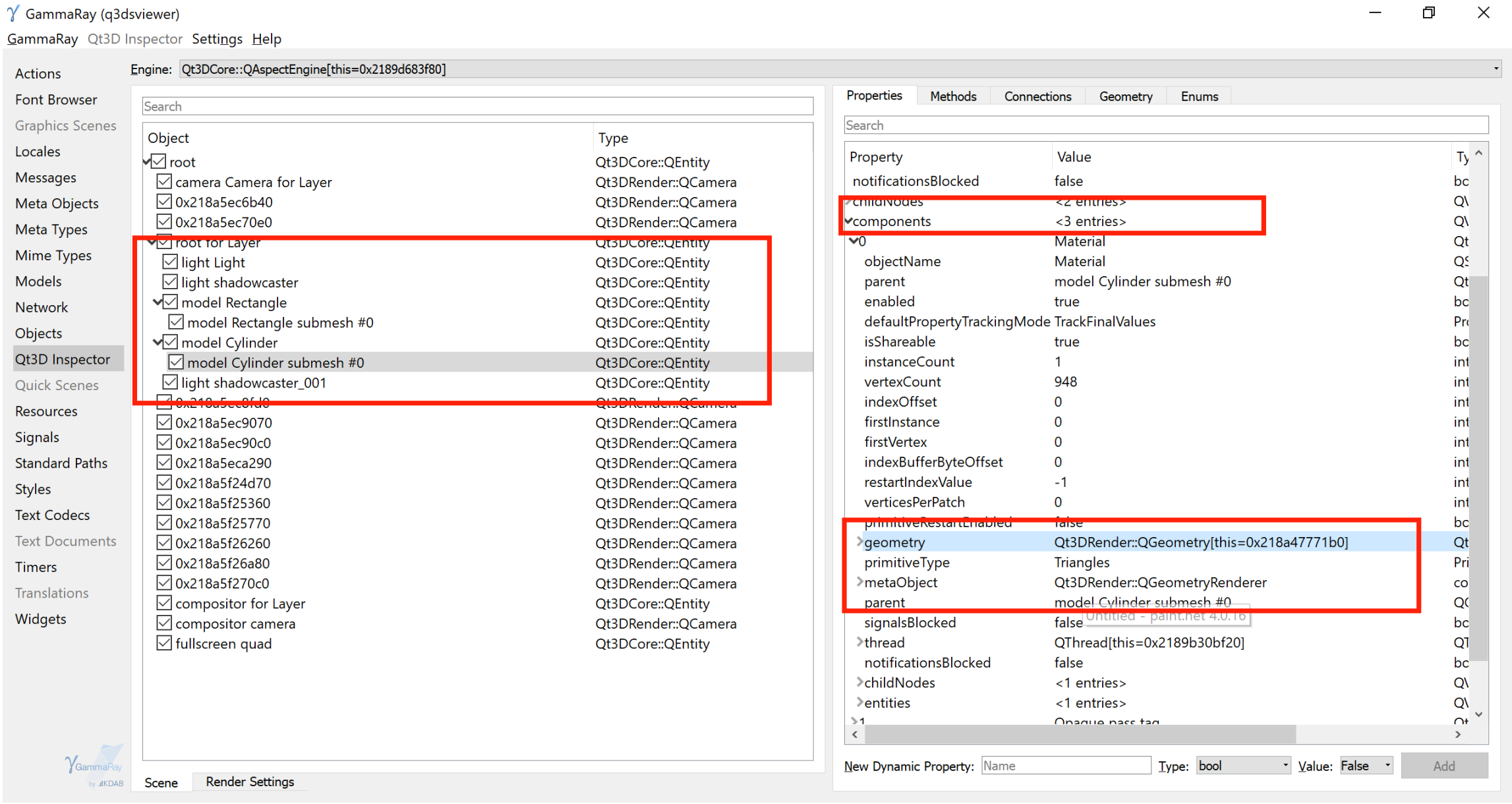
- Uses QXmlStreamReader.
- Graph and Logic sections map to a scene and a slide graph.
  - There are other sections as well (e.g. BufferData), not used in this example.
- Simple and efficient scene & slide graph representations. (based on the Qt Quick scene graph, unsurprisingly)

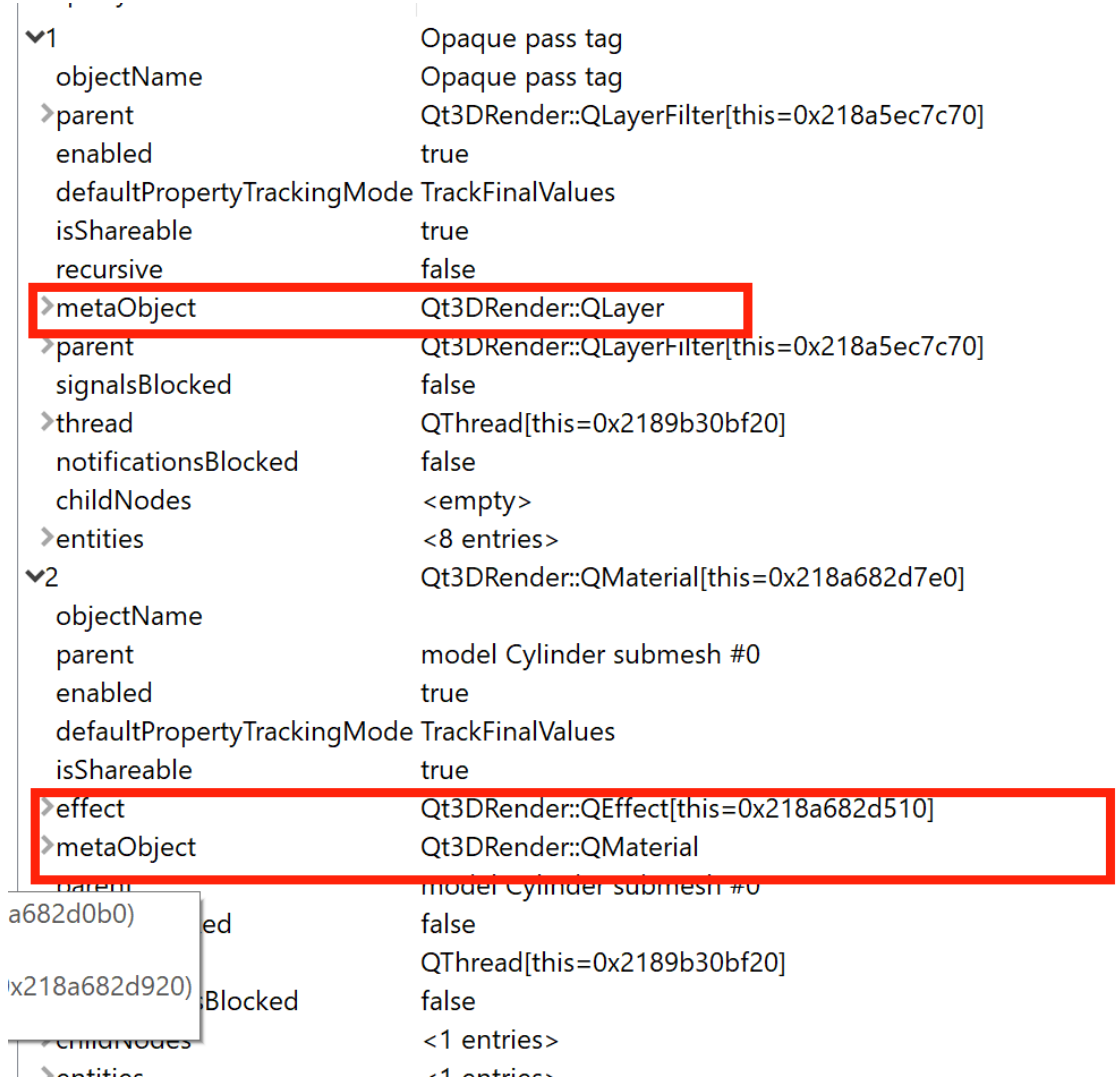

Graph explorer for 1f3738e5230 (Scene)
Object
Scene 1f3738e5230
Layer 1f377db2c30
Camera 1f3763b1ef0
Light 1f3738150a0
Model 1f373853490
DefaultMaterial 1f37389f040
Model 1f3738527d0
DefaultMaterial 1f37afcfe90
Light 1f377dcf100
Light 1f377dcdc60
Object 1f377dcdc60 of type Light with 0 children
id: shadowcaster
starttime: 0
endtime: 10000
flags: 1
rotation: [0 0 0]
position: [161.658 256.921 0]
scale: [1 1 1]
pivot: [0 0 0]
opacity: 1
skeletonId: -1
rotationOrder: 4
orientation: 0
name: shadowcaster
scope:
lightType: 1
lightDiffuse: #008080
lightSpecular: #80ff80
lightAmbient: #000000
brightness: 100
linearFade: 0
expFade: 0
areaWidth: 100
areaHeight: 100
castShadow: true
shadowFactor: 31.9375
shadowFilter: 27.8125
shadowMapRes: 9
shadowBias: 0
shadowMapFar: 5000
Attached generic node data:
globalOpacity: 1
globalVisibility: 1
Right-handed local transform: T: [161.658 256.921 0] R: [0 0 0] S: [1 1 1]


Graph explorer for 1f37381fa30 (Slide)
Object
Master Slide 1f37381fa30
Slide 1f3738200b0
Object 1f3738200b0 of type Slide with 0 children
id: Scene-Slide1
starttime: 0
endtime: 10000
name: Slide1
playMode: 0
initialPlayState: 0
This slide has 6 objects (excl. master), 0 property changes, and 3 animations
Objects:
1f377dcf100 shadowcaster_001
1f373853490 Rectangle
1f3738527d0 Cylinder
1f377dcdc60 shadowcaster
1f37afcfe90 Material_001
1f37389f040 Material
Property changes:
Animations:
type 2 on object 1f3738527d0 Cylinder property rotation.x with 2 keyframes
[0] time 0 value -52.8354
easeIn 100 easeOut 100
[1] time 10 value 50
easeIn 100 easeOut 100
type 2 on object 1f3738527d0 Cylinder property rotation.y with 2 keyframes
[0] time 0 value 30.8347
easeIn 100 easeOut 100
[1] time 10 value 30.8347
easeIn 100 easeOut 100
type 2 on object 1f3738527d0 Cylinder property rotation.z with 2 keyframes
[0] time 0 value -4.94139
easeIn 100 easeOut 100
[1] time 10 value -4.94139
easeIn 100 easeOut 100

Phase 2: The not quite so simple one: Build me a scene!

- Qt 3D uses an Entity-Component system.
  - Entity { components: [mesh, transform, material, …] … }
- The framegraph defines how to render, with nodes like RenderTargetSelector, CameraSelector, LayerFilter, TechniqueFilter, RenderPassFilter, Viewport, ClearBuffer.


GammaRay (q3dsviewer)
GammaRay Qt3D Inspector Settings Help
Actions
Font Browser
Graphics Scenes
Locales
Messages
Meta Objects
Meta Types
Mime Types
Models
Network
Objects
Qt3D Inspector
Quick Scenes
Resources
Signals
Standard Paths
Styles
Text Codecs
Text Documents
Timers
Translations
Widgets
Engine: Qt3DCore::QAspectEngine[this=0x2189d683f80]
Search
Object
Type
root Qt3DCore::QEntity
camera Camera for Layer Qt3DRender::QCamera
0x218a5ec6b40 Qt3DRender::QCamera
0x218a5ec70e0 Qt3DRender::QCamera
root for Layer Qt3DCore::QEntity
light Light Qt3DCore::QEntity
light shadowcaster Qt3DCore::QEntity
model Rectangle Qt3DCore::QEntity
model Rectangle submesh #0 Qt3DCore::QEntity
model Cylinder Qt3DCore::QEntity
model Cylinder submesh #0 Qt3DCore::QEntity
light shadowcaster_001 Qt3DCore::QEntity
0x218a5ec9070 Qt3DRender::QCamera
0x218a5ec90c0 Qt3DRender::QCamera
0x218a5eca290 Qt3DRender::QCamera
0x218a5f24d70 Qt3DRender::QCamera
0x218a5f25360 Qt3DRender::QCamera
0x218a5f25770 Qt3DRender::QCamera
0x218a5f26260 Qt3DRender::QCamera
0x218a5f26a80 Qt3DRender::QCamera
0x218a5f270c0 Qt3DRender::QCamera
compositor for Layer Qt3DCore::QEntity
compositor camera Qt3DRender::QCamera
fullscreen quad Qt3DCore::QEntity
Scene
Render Settings
Properties
Methods
Connections
Geometry
Enums
Search
Property
Value
notificationsBlocked false
childNodes <2 entries>
components <3 entries>
0 Material
objectName Material
parent model Cylinder submesh #0
enabled true
defaultPropertyTrackingMode TrackFinalValues
isShareable true
instanceCount 1
vertexCount 948
indexOffset 0
firstInstance 0
firstVertex 0
indexBufferByteOffset 0
restartIndexValue -1
verticesPerPatch 0
primitiveRestartEnabled false
geometry Qt3DRender::QGeometry[this=0x218a47771b0]
primitiveType Triangles
metaObject Qt3DRender::QGeometryRenderer
parent model Cylinder submesh #0
signalsBlocked false
thread QThread[this=0x2189b30bf20]
notificationsBlocked false
childNodes <1 entries>
entities <1 entries>
New Dynamic Property:
Name
Type:
bool
Value:
False
Add


1 Opaque pass tag
objectName Opaque pass tag
parent Qt3DRender::QLayerFilter[this=0x218a5ec7c70]
enabled true
defaultPropertyTrackingMode TrackFinalValues
isShareable true
recursive false
metaObject Qt3DRender::QLayer
parent Qt3DRender::QLayerFilter[this=0x218a5ec7c70]
signalsBlocked false
thread QThread[this=0x2189b30bf20]
notificationsBlocked false
childNodes <empty>
entities <8 entries>
2 Qt3DRender::QMaterial[this=0x218a682d7e0]
objectName
parent model Cylinder submesh #0
enabled true
defaultPropertyTrackingMode TrackFinalValues
isShareable true
effect Qt3DRender::QEffect[this=0x218a682d510]
metaObject Qt3DRender::QMaterial
a682d0b0)
x218a682d920)
ed false
QThread[this=0x2189b30bf20]
Blocked false
<1 entries>

Render passes in materials

- The material component of the entity can have multiple render passes.
- Render pass = shader program + render states (depth test, blending, etc.)
- Having N render passes on the material does not mean there will be N actual passes when rendering.
  - That is up to the framegraph.
- Having the Entity in the scene does not mean it is rendered at all.
  - That is up to the framegraph.

Render passes in the new runtime

- With Qt 3D Studio the default material will provide QRenderPasses (i.e. shaders + render state sets) for
  - Depth texture generation [optional]
  - SSAO texture generation [optional]
  - Shadow map generation [optional]
  - Depth pre-pass (opaque objects only, sort front to back, color write off)
  - Opaque pass (opaque objects only, sort front to back, depth write off)
  - Transparent pass (transparent objects only, sort back to front, blend on, depth write off)
    - NB sorting is to be ensured by the framegraph, not the material and QRenderPass; only listed here for completeness
- The Qt 3D light system and any built-in material in Qt3DExtras are not used at all.

Geometry in the new runtime

- With Qt 3D Studio we have a custom .mesh file format.
  - The baked, fast-to-load format output by the desktop application.
  - Hello real world asset conditioning pipelines…
  - …goodbye Mesh and SceneLoader.
- Handled by one or more entities with custom QGeometryRenderer components.

Intermission: Qt 3D tutorial: Framegraphs!

- Let’s see a framegraph snippet in QML.
- The new Qt 3D Studio runtime revolves around building an object tree with sub-trees like this in C++.

a.js - Qt Creator

File Edit Build Debug Analyze Tools Window Help

a.js

Line: 2, Col: 15

Welcome
Edit
Design
Debug
Projects
Help

```
RenderTargetSelector {
    target: rt
    Viewport { // Opaque+transparent passes
        normalizedRect: Qt.rect(0, 0, 1, 1)
        CameraSelector {
            camera: sceneCamera
            ClearBuffers {
                clearColor: Qt.rgba(0, 0, 0, 1)
                buffers: ClearBuffers.ColorDepthBuffer
                NoDraw { } // LEAF!
            }
            RenderPassFilter {
                matchAny: [ FilterKey { name: "pass"; value: "opaque" } ]
                LayerFilter { // LEAF!
                    layers: [ opaqueTag ]
                }
            }
            RenderPassFilter { // no clear between opaque and transparent passes
                matchAny: [ FilterKey { name: "pass"; value: "transparent" } ]
                SortPolicy {
                    sortTypes: [ SortPolicy.BackToFront ]
                    LayerFilter { // LEAF!
                        layers: [ transparentTag ]
                    }
                }
            }
        }
    }
}
```

qclearb

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 Debugger Console 6 General Messages

Framegraphs: Some Lessons Learned

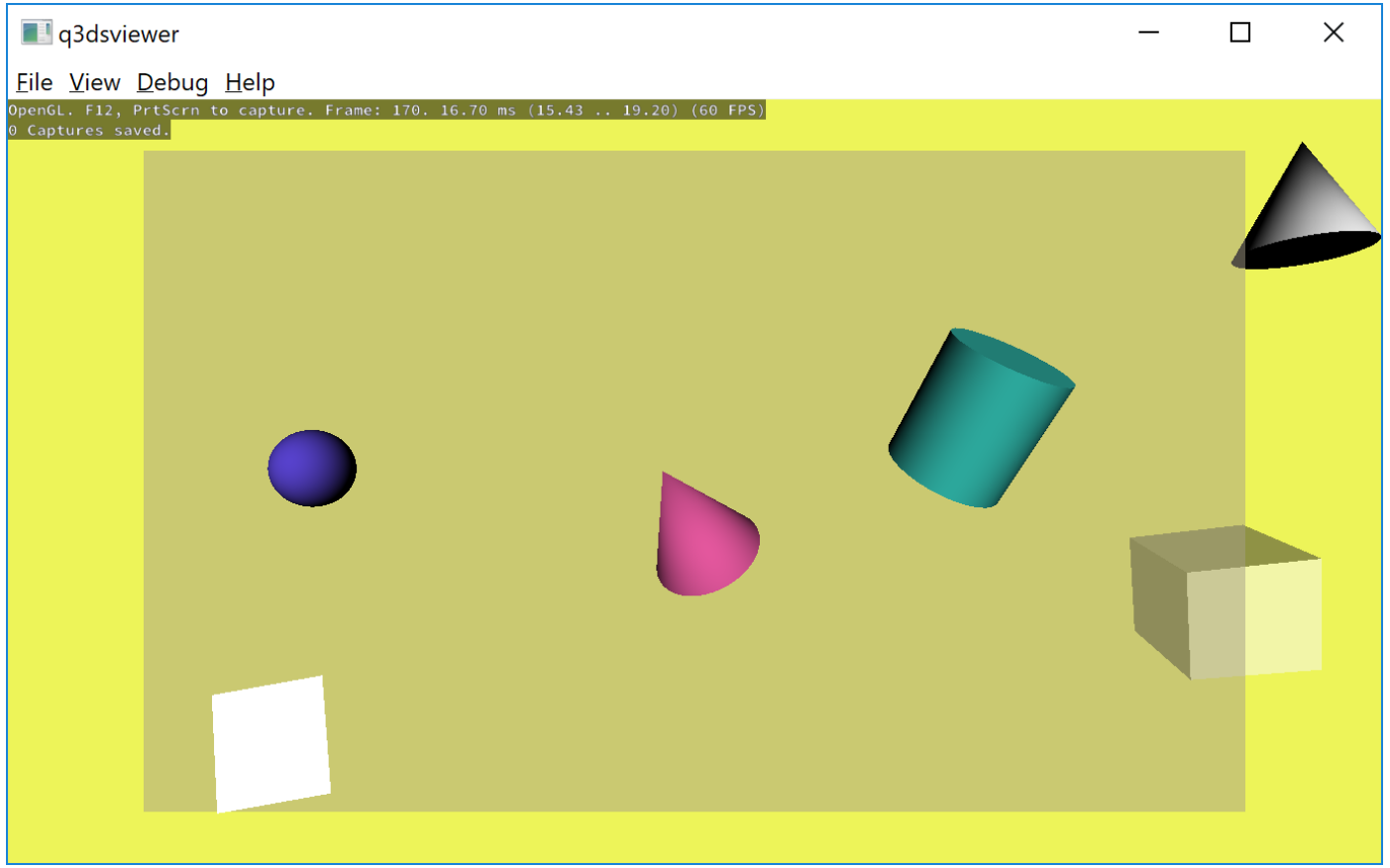
- Qt 3D's framegraphs are pretty much the right thing for more complex, multi-pass scenes.
- Error prone and unintuitive at first, however.
- **The value and importance of using a tool like RenderDoc cannot be emphasized enough.**
- The application's output may look correct but there may be an entire world of unexpected draw calls, clears, and other changes happening under the hood.
  - due to e.g. misparenting a node, or unintentionally introducing a new leaf node
- For debugging the Qt 3D scene, GammaRay can be used.
  - Good for checking the framegraph structure or layer tags.

Let’s verify

A Qt 3D Studio scene with 2 layers.

Layer 1: 4 objects, all with opaque materials.

Layer 2: 1 opaque (top-right cone), 2 transparent (cube on the right, semi-transparent big rectangle)



q3dsviewer_2017.09.20_12.55.25_frame232.rdc - RenderDoc v0.90

File Window Tools Help

Event Browser

Controls

EID	Name
0	Frame Start
12-74	Colour Pass #1 (1 Targets + Depth)
12	glClear(Color = <0.000000, 0.000000, 0.00000...
22	glDrawElementsInstancedBaseVertex(948, 1)
25	glDrawElementsInstancedBaseVertex(234, 1)
28	glDrawElementsInstancedBaseVertex(24, 1)
31	glDrawElementsInstancedBaseVertex(14700...
53	glDrawElementsInstancedBaseVertex(948, 1)
60	glDrawElementsInstancedBaseVertex(234, 1)
67	glDrawElementsInstancedBaseVertex(24, 1)
74	glDrawElementsInstancedBaseVertex(14700, 1)
89-153	Colour Pass #2 (1 Targets + Depth)
89	glClear(Color = <0.000000, 0.000000, 0.00000...
99	glDrawElementsInstancedBaseVertex(234, 1)
121	glDrawElementsInstancedBaseVertex(234, 1)
146	glDrawElementsInstancedBaseVertex(36, 1)
153	glDrawElementsInstancedBaseVertex(24, 1)
164-182	Colour Pass #3 (1 Targets + Depth)
164	glClear(Color = <0.929412, 0.956863, 0.34902...
177	glDrawElementsInstancedBaseVertex(6, 1)
182	glDrawElementsInstancedBaseVertex(6, 1)
190	SwapBuffers()

Depth pre-pass

Opaque pass

Depth pre-pass (layer 2)

Opaque pass (layer 2)

Transparent pass (layer 2)

> Compositor (textured quads)

API Calls

EID	API Call
29	glBindVertexArray
30	glProgramUniformMatrix*
31	**glDrawElementsInstancedBaseVertex**

Callstack

Pipeline State | Mesh Output | Texture Viewer | Capture Executable | q3dsviewer [PID 21120]

Display Controls | Show Disabled Items | Show Empty Items | Export

VTX → VS → TCS → TES → GS → Rasterizer → FS → FB → CS

Draw Buffers

Slot	Resource	Type	Width	Height	Depth	Array Size	Format
0	Texture2D RTV 111	2D	1366	760	1	1	RGBA8
Depth	Texture2D DSV 112	2D	1366	760	1	1	DEPTH24_STENCIL8
Stencil	Texture2D DSV 112	2D	1366	760	1	1	DEPTH24_STENCIL8

Target Blends

Slot	Enabled	Colour Src	Colour Dst	Colour Op	Alpha Src	Alpha Dst	Alpha Op	Write Mask
0	False	One	InvSrcAlpha	Add	One	InvSrcAlpha	Add	____

Blend State

Blend Factor:	0.00, 0.00, 0.00, 0.00

Depth State

Enabled: ✓ Write: ✓

Bounds: ✗ 0.000 - 0.000

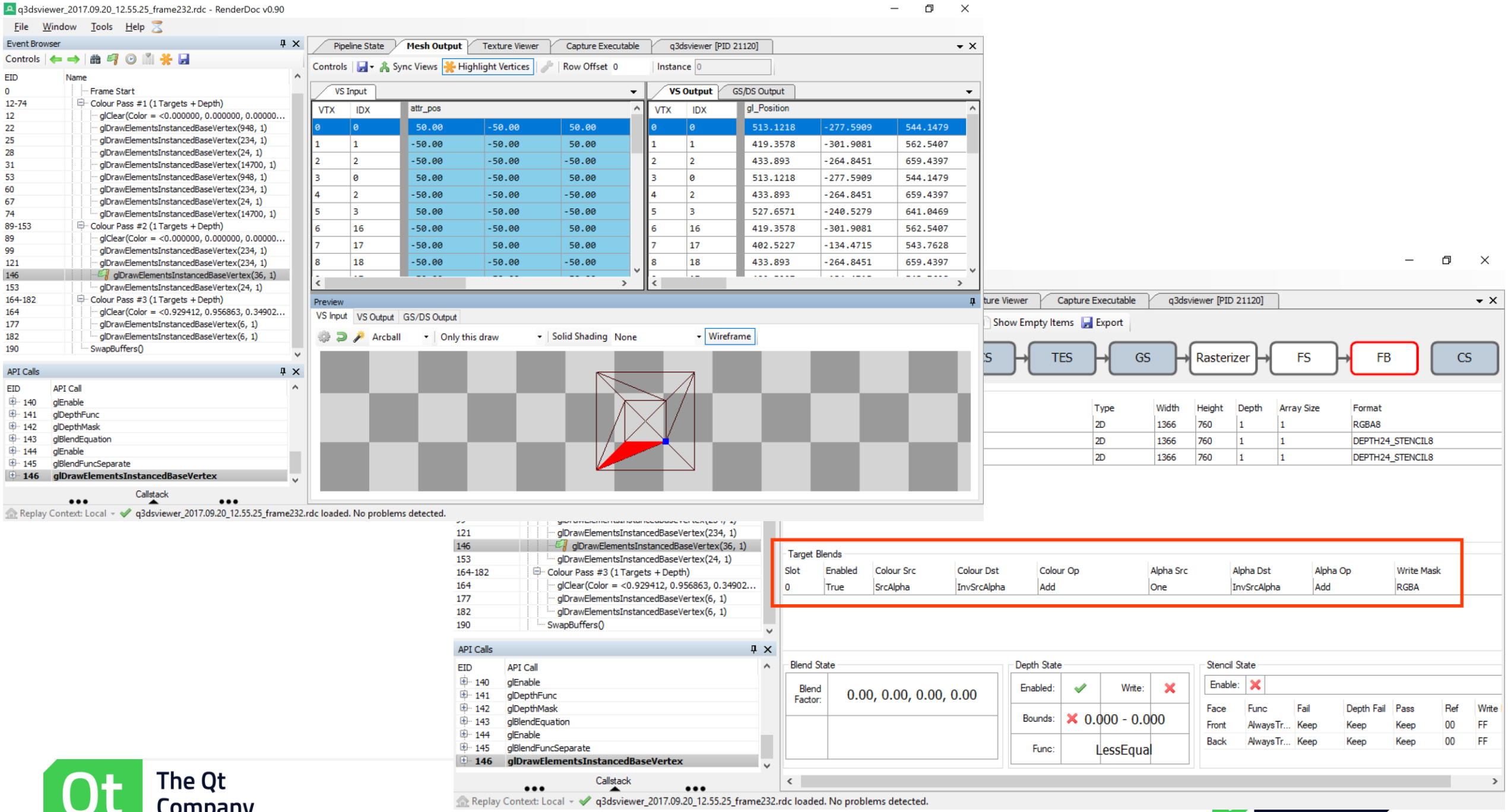
Func: LessEqual

Stencil State

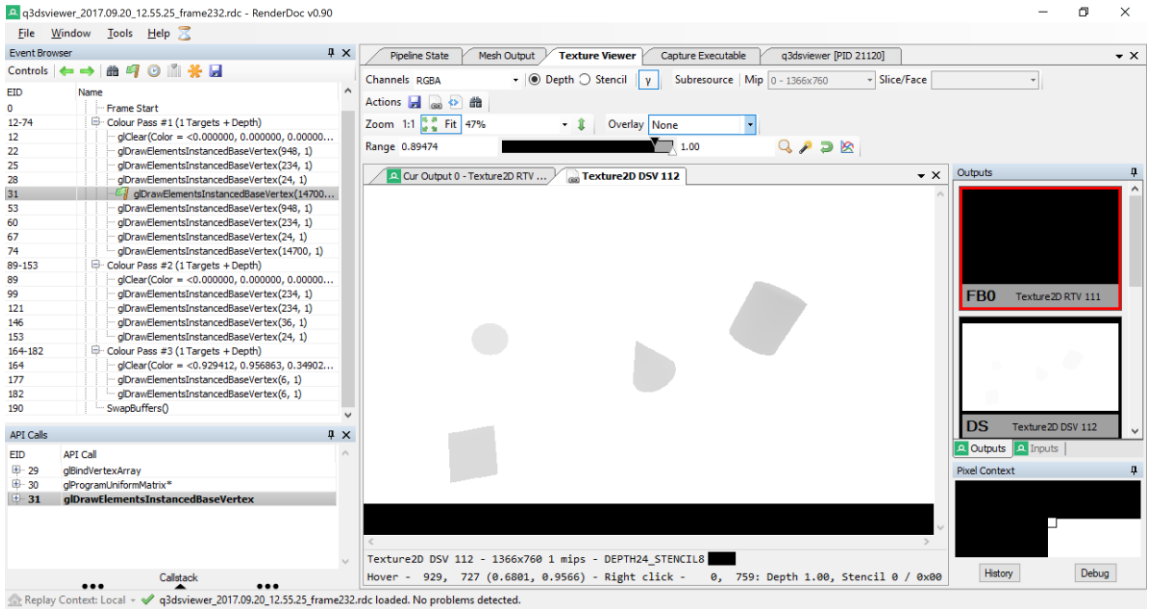
Enable: ✗

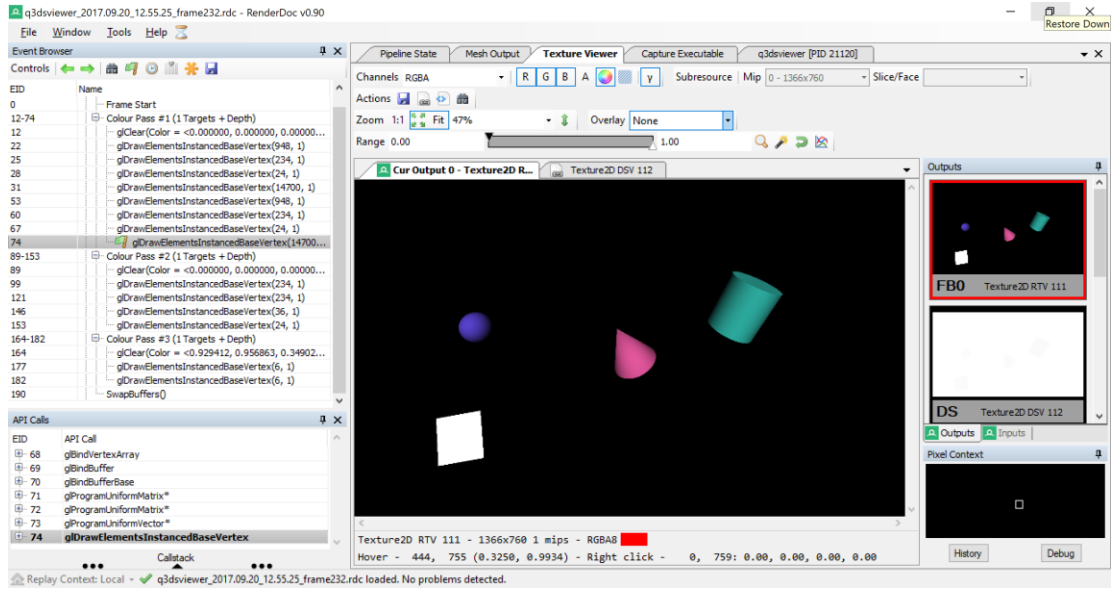
Face	Func	Fail	Depth Fail	Pass	Ref	Write
Front	AlwaysTr...	Keep	Keep	Keep	00	FF
Back	AlwaysTr...	Keep	Keep	Keep	00	FF

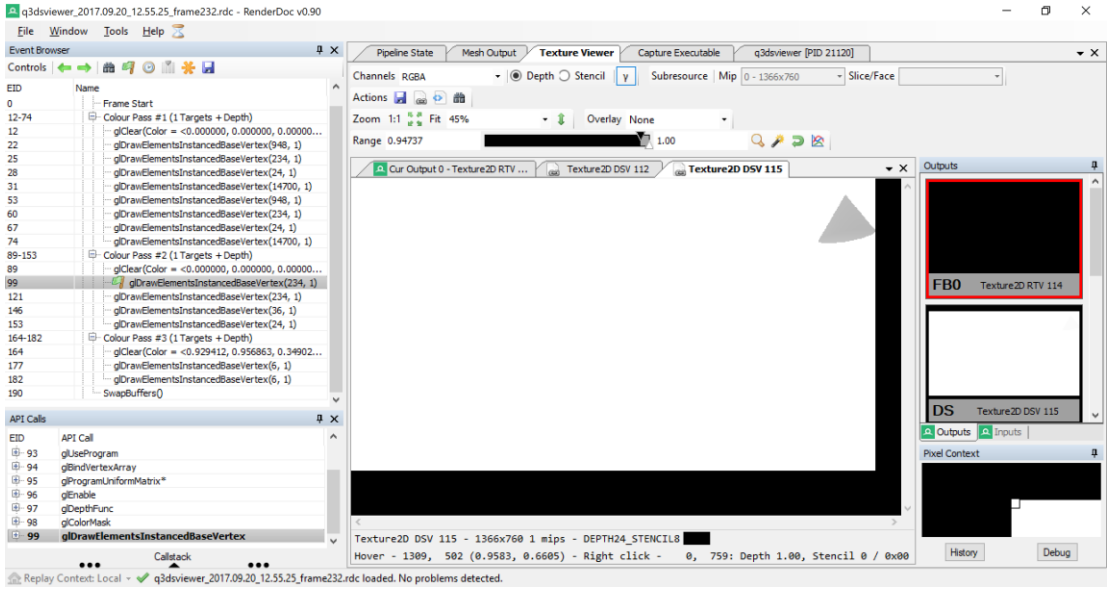
Replay Context: Local ✓ q3dsviewer_2017.09.20_12.55.25_frame232.rdc loaded. No problems detected.

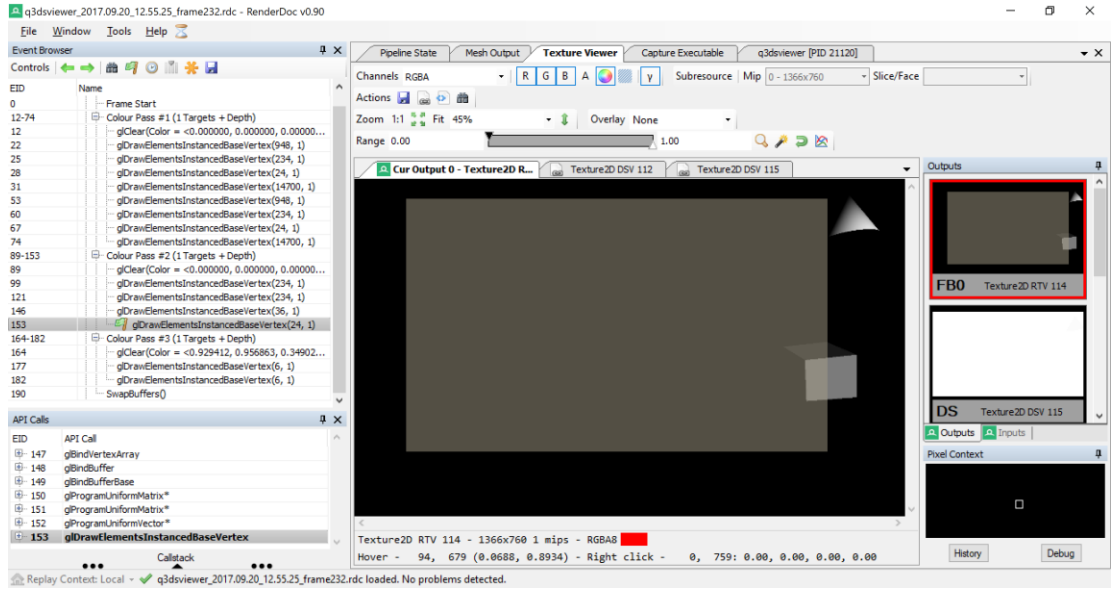


The Qt Company


q3dsviewer_2017.09.20_12.55.25_frame232.rdc - RenderDoc v0.90
File Window Tools Help
Event Browser
Controls
Pipeline State Mesh Output Texture Viewer Capture Executable q3dsviewer [PID 21120]
Channels RGBA Depth Stencil Subresource Mip Slice/Face
Actions
Zoom 1:1 Fit 47% Overlay None
Range 0.89474 1.00
Cur Output 0 - Texture2D RTV ... Texture2D DSV 112
Outputs
FB0 Texture2D RTV 111
DS Texture2D DSV 112
Pixel Context
API Calls
Callstack
Texture2D DSV 112 - 1366x760 1 mips - DEPTH24_STENCIL8
Hover - 929, 727 (0.6801, 0.9566) - Right click - 0, 759: Depth 1.00, Stencil 0 / 0x00
History Debug
Replay Context: Local
q3dsviewer_2017.09.20_12.55.25_frame232.rdc loaded. No problems detected.


q3dsviewer_2017.09.20_12.55.25_frame232.rdc - RenderDoc v0.90
File Window Tools Help
Restore Down
Event Browser
Controls
Pipeline State Mesh Output Texture Viewer Capture Executable q3dsviewer [PID 21120]
Channels RGBA R G B A Subresource Mip Slice/Face
Actions
Zoom 1:1 Fit 47% Overlay None
Range 0.00 1.00
Cur Output 0 - Texture2D R... Texture2D DSV 112
Outputs
FB0 Texture2D RTV 111
DS Texture2D DSV 112
Pixel Context
API Calls
Callstack
Texture2D RTV 111 - 1366x760 1 mips - RGBA8
Hover - 444, 755 (0.3250, 0.9934) - Right click - 0, 759: 0.00, 0.00, 0.00, 0.00
History Debug
Replay Context: Local
q3dsviewer_2017.09.20_12.55.25_frame232.rdc loaded. No problems detected.


q3dsviewer_2017.09.20_12.55.25_frame232.rdc - RenderDoc v0.90
File Window Tools Help
Event Browser
Controls
Pipeline State Mesh Output Texture Viewer Capture Executable q3dsviewer [PID 21120]
Channels RGBA Depth Stencil Subresource Mip Slice/Face
Actions
Zoom 1:1 Fit 45% Overlay None
Range 0.94737 1.00
Cur Output 0 - Texture2D RTV ... Texture2D DSV 112 Texture2D DSV 115
Outputs
FB0 Texture2D RTV 114
DS Texture2D DSV 115
Pixel Context
API Calls
Callstack
Texture2D DSV 115 - 1366x760 1 mips - DEPTH24_STENCIL8
Hover - 1309, 502 (0.9583, 0.6605) - Right click - 0, 759: Depth 1.00, Stencil 0 / 0x00
History Debug
Replay Context: Local
q3dsviewer_2017.09.20_12.55.25_frame232.rdc loaded. No problems detected.


q3dsviewer_2017.09.20_12.55.25_frame232.rdc - RenderDoc v0.90
File Window Tools Help
Event Browser
Controls
Pipeline State Mesh Output Texture Viewer Capture Executable q3dsviewer [PID 21120]
Channels RGBA R G B A Subresource Mip Slice/Face
Actions
Zoom 1:1 Fit 45% Overlay None
Range 0.00 1.00
Cur Output 0 - Texture2D R... Texture2D DSV 112 Texture2D DSV 115
Outputs
FB0 Texture2D RTV 114
DS Texture2D DSV 115
Pixel Context
API Calls
Callstack
Texture2D RTV 114 - 1366x760 1 mips - RGBA8
Hover - 94, 679 (0.0688, 0.8934) - Right click - 0, 759: 0.00, 0.00, 0.00, 0.00
History Debug
Replay Context: Local
q3dsviewer_2017.09.20_12.55.25_frame232.rdc loaded. No problems detected.

q3dsviewer_2017.09.20_12.55.25_frame232.rdc - RenderDoc v0.90

File Window Tools Help

Event Browser

Controls

EID	Name
0	Frame Start
12-74	Colour Pass #1 (1 Targets + Depth)
12	glClear(Color = <0.000000, 0.000000, 0.00000...
22	glDrawElementsInstancedBaseVertex(948, 1)
25	glDrawElementsInstancedBaseVertex(234, 1)
28	glDrawElementsInstancedBaseVertex(24, 1)
31	glDrawElementsInstancedBaseVertex(14700, 1)
53	glDrawElementsInstancedBaseVertex(948, 1)
60	glDrawElementsInstancedBaseVertex(234, 1)
67	glDrawElementsInstancedBaseVertex(24, 1)
74	glDrawElementsInstancedBaseVertex(14700, 1)
89-153	Colour Pass #2 (1 Targets + Depth)
89	glClear(Color = <0.000000, 0.000000, 0.00000...
99	glDrawElementsInstancedBaseVertex(234, 1)
121	glDrawElementsInstancedBaseVertex(234, 1)
146	glDrawElementsInstancedBaseVertex(36, 1)
153	glDrawElementsInstancedBaseVertex(24, 1)
164-182	Colour Pass #3 (1 Targets + Depth)
164	glClear(Color = <0.929412, 0.956863, 0.34902...
177	glDrawElementsInstancedBaseVertex(6, 1)
182	glDrawElementsInstancedBaseVertex(6, 1)
190	SwapBuffers()

API Calls

EID	API Call
178	glBindVertexArray
179	glActiveTexture
180	glBindTexture
181	glProgramUniformVector*
182	**glDrawElementsInstancedBaseVertex**

Callstack

Pipeline State | Mesh Output | **Texture Viewer** | Capture Executable | q3dsviewer [PID 21120]

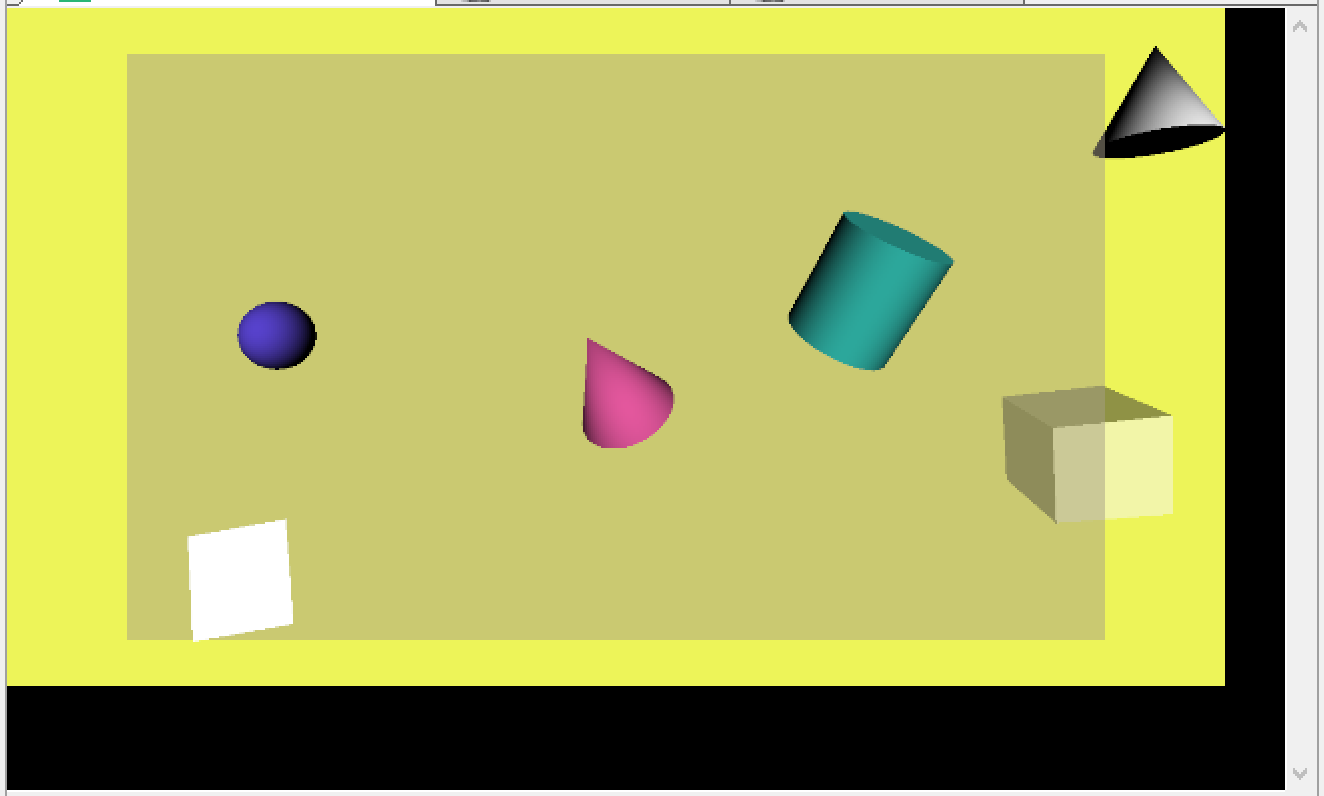
Channels RGBA | R G B A | γ | Subresource | Mip 0 - 1366x760 | Slice/Face

Actions

Zoom 1:1 | Fit 45% | Overlay None

Range 0.00 — 1.00

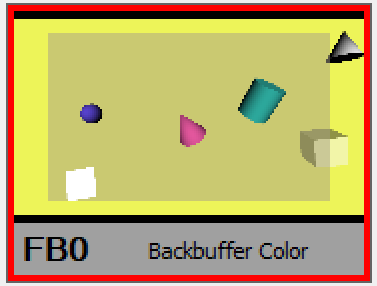
Cur Output 0 - Backbuffer C... | Texture2D DSV 112 | Texture2D DSV 115

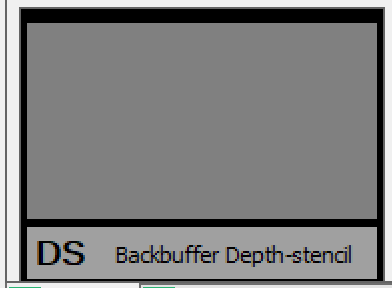


Backbuffer Color - 1366x760 1 mips - SRGB8_ALPHA8

Hover - 226, 143 (0.1654, 0.1882) - Right click - 0, 759: 0.84766, 0.90625, 0.1001, 1.0

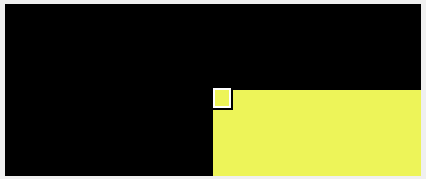
Outputs





Outputs | Inputs

Pixel Context



History Debug

Replay Context: Local — q3dsviewer_2017.09.20_12.55.25_frame232.rdc loaded. No problems detected.

Phase 3: The real annoying one: Run my scene!

- We have states and animations in our .uip documents.
- C++ and QML/JS APIs to control the Qt 3D Studio scene.
- The runtime has to take care of all this, covering both scenes, the internal Qt 3D Studio one and the “live” Qt 3D scene.
- Relies heavily on Logic and Animation aspects.
  - To perform scene updates once per frame, e.g. to update inherited (global) transform and opacity.
  - And to run the keyframe-based animations.
    - AnimationClip is great since it is tied to the Qt 3D render loop, not to some random timer.

Conclusion

- Complex framegraphs are the real power of Qt 3D.
  - Not the examples showing a rotating torus with the built-in PhongMaterial. ☺
- But, is this something you want to hand-code for each and every scene and project?
- No. You want tools and a higher level runtime that does this for you.
- Hence the need for Qt 3D Studio…
- …which works fine without Qt 3D in version 1.0 already…
- …but is getting a Qt 3D-based runtime for a more compact and future-proof story.

Thank You!

laszlo.agocs@qt.io

Twitter: @alpqr

IRC: lagocs on Freenode

https://codereview.qt-project.org/gitweb?p=qt3dstudio%2Fqt3dstudio.git;a=summary

https://codereview.qt-project.org/gitweb?p=qt3dstudio%2Fqt3d-runtime.git;a=summary