What’s new in Qt 3D and Kuesa

Paul Lemire – paul.lemire@kdab.com
What’s new in Qt 3D 5.14?
Qt 3D 5.14

• Goals
  - Improving performance and maintainability
    • Threading overhaul
    • Aspects/Nodes synchronization
  - Ground work for Qt 6
    • Research on how to best leverage modern Graphics API
    • API reviews

• New additions
Threads in Qt 3D 5.12/5.13

Main Thread

Aspect Thread

Aspects

ThreadPool

Render Thread or Qt Quick SceneGraph Thread

ChangeArbiter

Simulation Loop

waitForNextFrame

syncChanges

Aspects → Launch Jobs

Wait for jobs to be completed

Job

Submit Render Commands

ProceedToNextFrame

The Qt, OpenGL and C++ experts
Threads in Qt 3D 5.14
Threading architecture overhaul

• Simplify Architecture
  - Easier to maintain
  - Easier to control the Qt 3D run loop
    • Makes integration with 3rd party engines / Qt Quick a lot nicer
    • Ground work for Qt 6 and RHI integration
  - Allows for some optimizations
    • Direct sync of frontend/backend trees
  - Reduces the numbers of frames between a frontend action and response from the backend (down to 1 frame)
Improved synchronizations between Aspect

- Deprecation of message based syncing
- Direct frontend node access instead
  - Possible because of Aspect thread removal
  - Syncing 5000 entities (12ms vs 4ms)
What’s next for Qt 3D and Qt 6

- Currently investigating RHI rendering backend
  - Current limitations:
    - Assumes single threaded rendering pipeline
    - Pre compiled shaders
- Linearized FrameGraph description
  - Will allow tooling around that
  - Will make caching commands between frames easier
- Rendering optimization information
  - Display number of drawn entities/triangles ...
New additions in 5.14

- **Scene3D**
  - underlay mode
  - Scene3DView

- **FrameGraph**
  - QNoPicking
  - QSubtreeEnabler (Michael Brasser - Ford)
  - QSortPolicy
    - Sorting render commands by Texture

- **QTransform:**
  - worldMatrix property added

- **QTexture**
  - ktx support added

- **QShaderImage**
  - OpenGL 4 Image support

- **QObjectPicker**
  - NearestPriorityPick picking mode added
What’s new in Kuesa?
What is Kuesa?

• Asset creation and integration workflow
• Let designers export from designer tools (Blender, 3DSMax) in glTF2 format
• Let developers import glTF2 assets into applications
• Provides tooling to preview and condition assets
• Built on top of Qt 3D
• Code available on https://github.com/KDAB/kuesa
• Free and commercial licenses available
What is Kuesa?

3D design tool
- 3ds Max®
- Blender

runtime

designers

developers

tools

digital content creation plugins

Qt3D

Qt

The Qt, OpenGL and C++ experts
Kuesa 1.1 Runtime Highlights

- Runs on top of Qt 3D 5.12.5 or later
- C++/QML APIs and Python bindings
- Asset Collections
  - Retrieve resources by name after import
- Up to date examples and documentation

- glTF Importer
  - glTF 2 compliant
  - Physically Based Materials
  - Skinned and Morph Target animation support
  - Mesh compression with draco

- Post Processing Effects
  - Blur
  - Depth Of Field
  - Tone Mapping
Kuesa 1.1- Tools

- gltfEditor
- gltfViewer
- assetprocessor
- cubemaptooctahedralmap
  - Used to support PBR on ES 2
Demo
Questions?