3D表現を簡単に作成
『Qt Quick 3D』

自己紹介

- 柳　云善：リュウ　ウンソン (Ryu, Unseon)
- Qt Company Oy、 Japan Office
- Senior Pre-sales Engineer
  - 主に車業界の顧客向けに活動
    - Qt Design Studio / QMLを活用したワークフローの提案
    - IVIとMeter ClusterのUI Architectureの提案

Qt Quick 3Dとは？

- QML用3D機能のモジュール
  - 3Dコンテンツの作成を可能とするライブラリ
  - Qt 5.15から正式リリース
- High Level（粒度高い）QML APIを提供
  - モデル
    - マテリアル
  - ライト
  - カメラ
- Importツールを提供
- Qt Design Studio 1.5の3D Editorを提供

Qt Quick 3Dの特徴

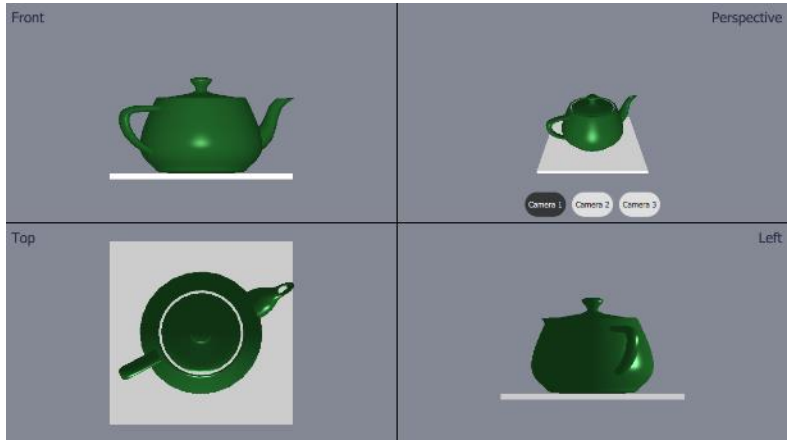
- 簡単な書き方で強力な3Dを表現
  - QML
  - Entity-Component Systemの**不採用**
  - 組み込みでも快適なパフォーマンス
  - Image-Based Lighting
  - Physics-Based Material
    - Metalness / Roughness
  - Effect、サンプリング、影の表現
    - Graphics知識がなくても利用可能

従来のQtの3Dモジュール

- Qt 3D
  - QML・C++ライブラリ
  - Low-level API
  - Qt Quick (2D）と別の描画エンジン
  - Entity－Component System
- Qt 3D Studio
  - Qt 3D Studioツール向けのRuntime
  - QMLライブラリではない
  - 決まっているアニメーションの再生に特化
- Qt Quick 3D
  - QMLライブラリ
  - Qt Quickと統合された描画Architecture
    - 高いパフォーマンス
    - ２D/3D 同期
  - Qt Design Studioでサポート

QMLでの3D画面の作成

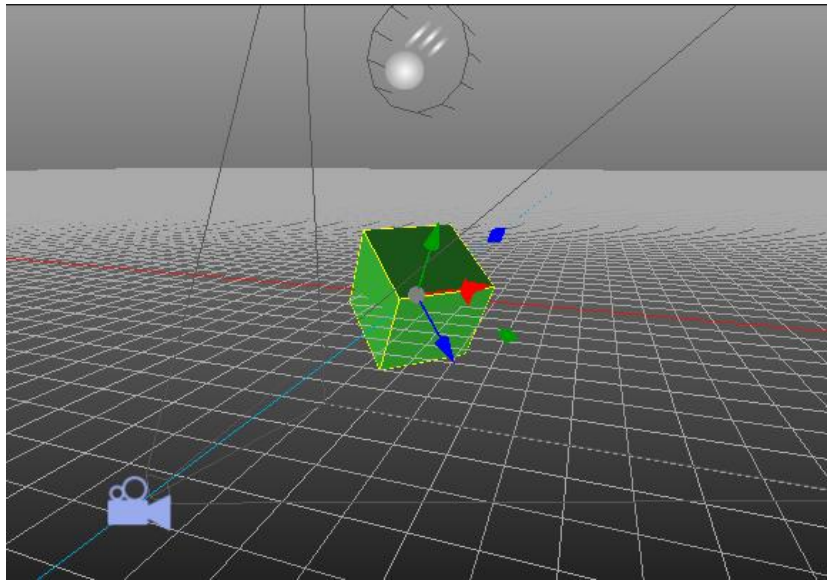
- 2 D in 3D
  - 既存のQt Quick画面でView3D部品を追加
  - Single-Scene / Multi-View




```
import QtQuick 2.15
import QtQuick.Window 2.15
import QtQuick3D 1.15

Window {
    visible: true
    width: 1280
    height: 720
    title: qsTr("Basic 3D Example")

    View3D {
        id: viewport
        anchors.fill: parent
    }
}
```

3Dコンテンツの作成

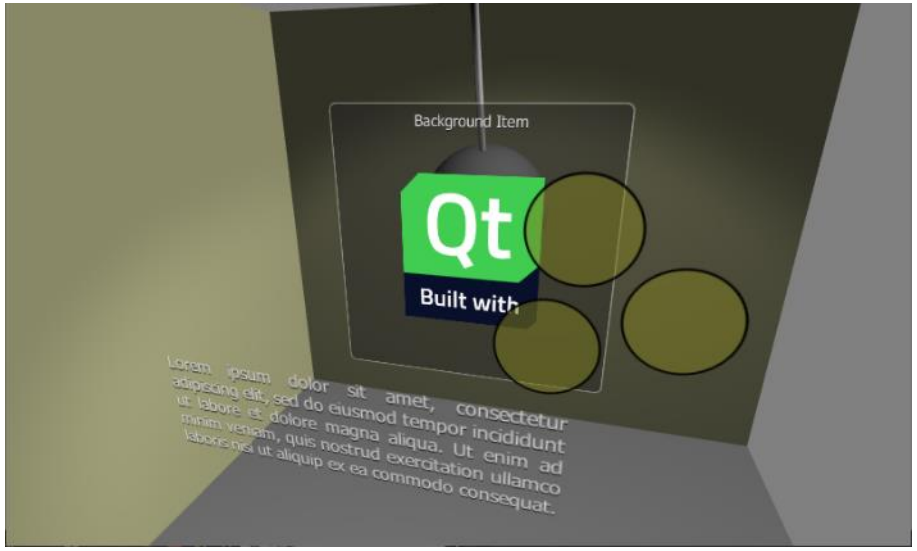
- View3Dの配下で３D部品を配置
  - カメラ
  - ライト
  - モデル
    - マテリアル



```
View3D {
    id: viewport
    anchors.fill: parent

    PerspectiveCamera {
        id: camera
        z: 100
    }

    DirectionalLight {
        id: light
    }

    Model {
        id: cube
        source: "#Cube"
        materials: DefaultMaterial { }
    }
}
```

3Dコンテンツの中での2Dの表現

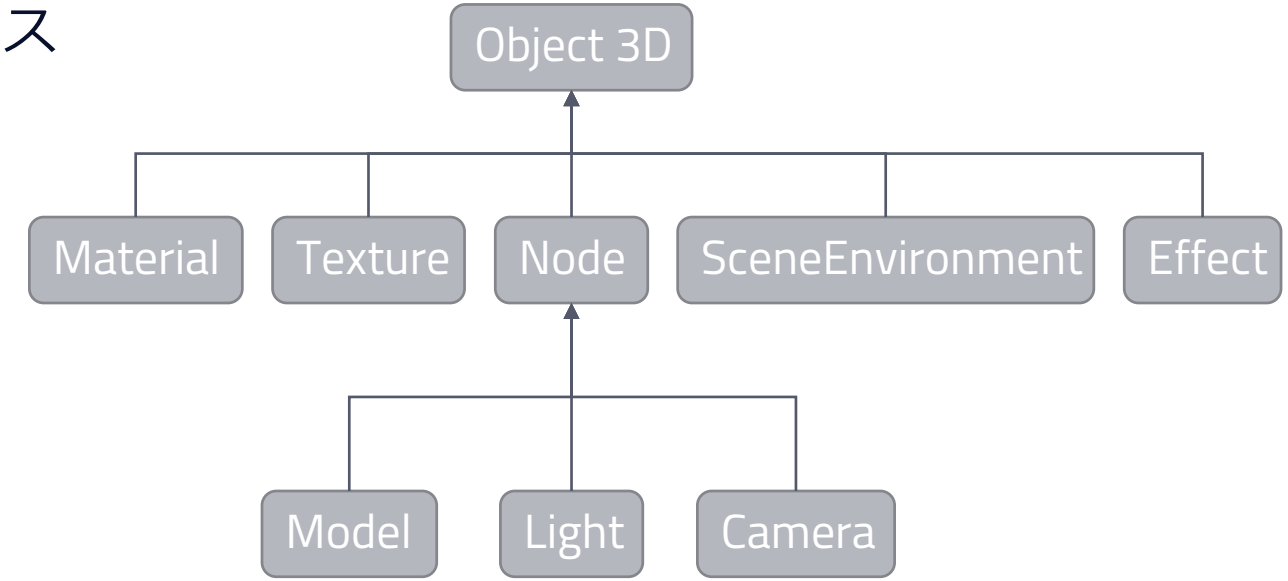
- 3 D部品の配下でQt Quick部品を配置

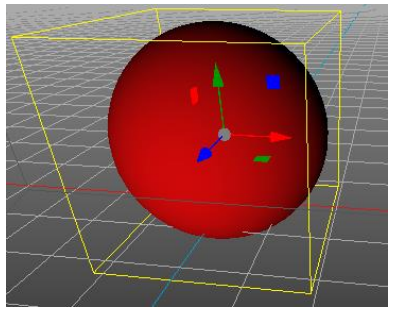



```
Node {
    position: Qt.vector3d(0, 100, -120)
    Item {
        width: 400
        height: 400
        Rectangle {
            anchors.fill: parent
            opacity: 0.4
            color: "#202020"
            radius: 10
            border.width: 2
            border.color: "#f0f0f0"
        }
        Text {
            anchors.top: parent.top
            anchors.topMargin: 10
            anchors.horizontalCenter: parent.horizontalCenter
            font.pixelSize: 20
            color: "#e0e0e0"
            style: Text.Raised
            text: qsTr("Background Item")
        }
        Image {
            anchors.centerIn: parent
            source: "Built_with_Qt_RGB_logo_vertical"
        }
    }
}
```

The Node Item

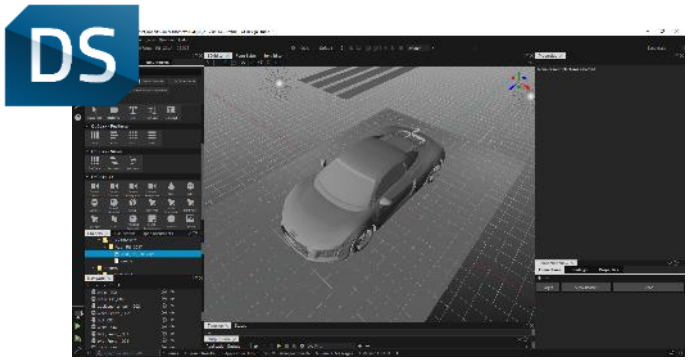
- 3Dオブジェクトを表現するためのベースクラス
- Local 3D Transform
  - Position (translation) / Rotation / Scale
- 階層構造（親子関係）
- Entity Component Systemを採用せず
  - モデル、ライト、カメラもNodeの一種
  - Entityにぶら下げる構造にしない
- Right-handed Orientation
  - 右方向: X+
  - 上方向: Y+
  - 奥行: -Z





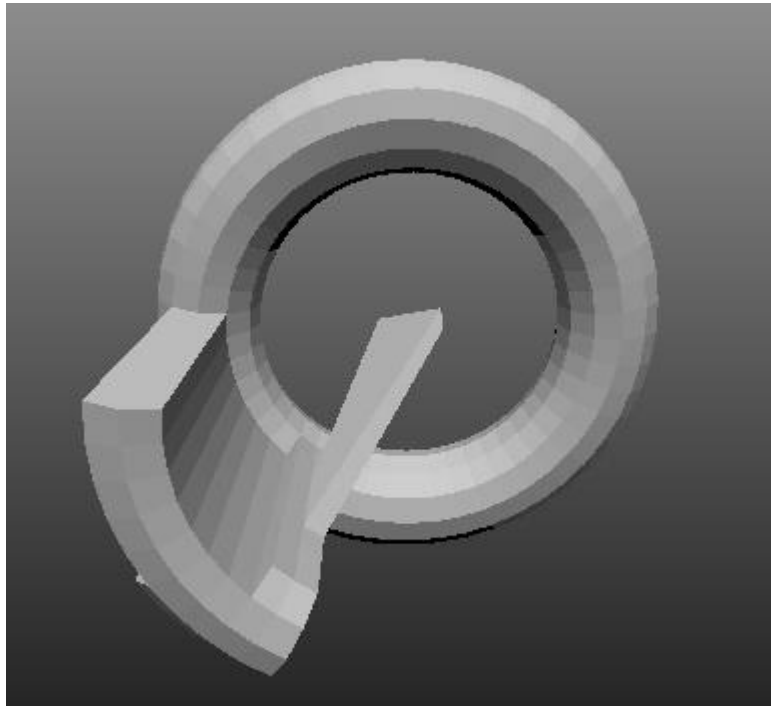
3Dアセットのインポート

- Command Line Tool ：balsam
- Qt Design Studio
- 3Dアセットファイルは QML形式に変換
  - MyCarModel.fbx -> MyCarModel.qml
- QMLファイルはオリジナルのアセットとほぼ同様の階層構造を再現
  - .mesh files contain geometry
  - Image files for Textures



Model Node

- GeometryとMaterialの組み合わせ
  - Geometry
    - モデルの形の情報
    - Static Mesh：.meshファイルをロード
    - Dynamic Mesh：Geometryを使いメモリから取得



Materials

- 基本提供 Materials
  - DefaultMaterial
    - Diffuse / Specular
  - PrincipledMaterial
    - Metalness：乱反射→鏡面反射か
    - Roughness：鮮明に反射するかぼかすか
- カスタムMaterial API
  - QML API
  - GLSL shaders

Model Node Shadow

- Light Node
  - castShadow property
- モデル事に適用
  - castShadow: 影を映せるように設定
    - 例：車のモデル
  - receivesShadow: 影が映るように設定
    - 例：地面

Camera Nodes

- PerspectiveCamera
  - Near + Far
  - Field of View
- OrthgraphicCamera
  - Near + Far
- FrustumCamera (extends PerspectiveCamera)
  - Top, Bottom, Left, Right, Near, Far
  - glFrustum(⋯)
  - Used for creating a Asymmetric Frustums (needed for VR)
- CustomCamera
  - Define your own projection matrix

Lighting Nodes

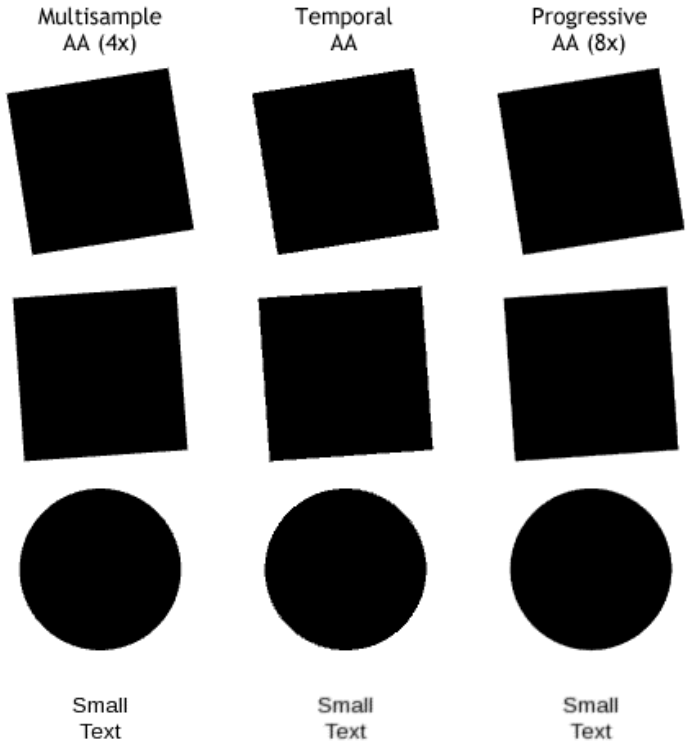
- DirectionalLight
- PointLight
- SpotLight
- AreaLight

SceneEnvironment

- View3D全体に影響がある設定を行う
  - Anti-aliasing
  - Image-Based Lighting
  - Post Processing Effects

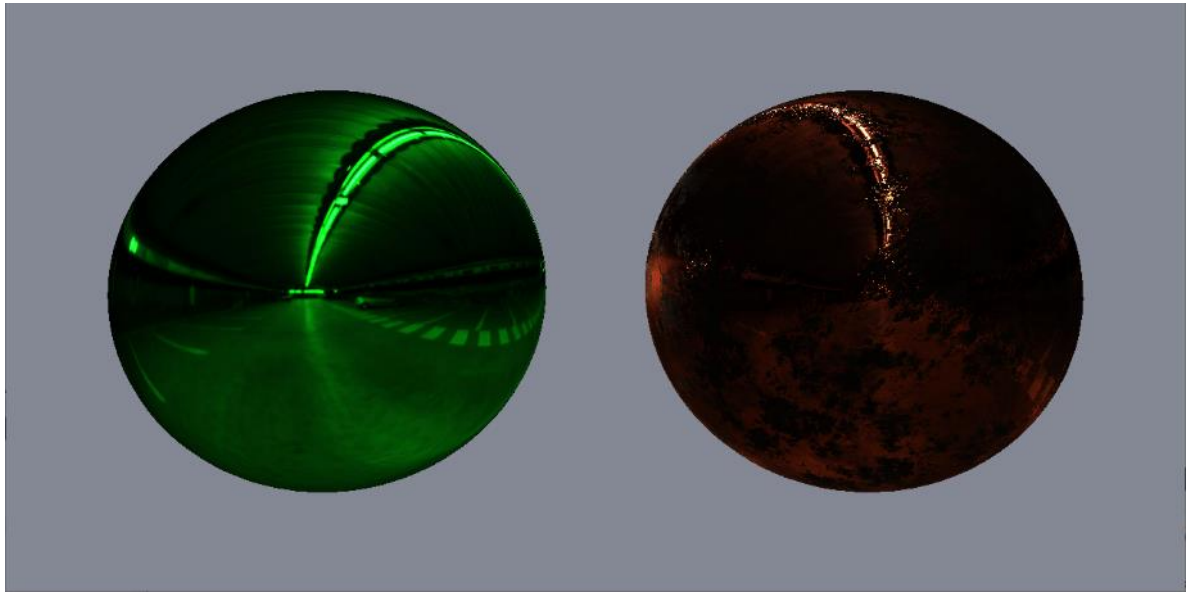
SceneEnvironment Anti-aliasing Techniques

- Supersampling Anti-aliasing (SSAA)
  - Render the scene to a higher resolution and scale down
- Multisampling Anti-aliasing (MSAA)
  - Sample geometry edges multiple times
- Progressive Anti-aliasing
  - Smooth a non-animating scene out by “jittering” the camera and blending
- Temporal Anti-aliasing
  - Smooth an animating scene by “jittering” the camera and blending previous frames



SceneEnvironment Image Based Lighting

- HDRI
  - High Dynamic Range Images
- Material事の設定も可能
- PrincipledMaterialと組み合わせて使うことで高品質の表現が可能



SceneEnvironment
Post Processing Effects

- QtQuick3D.Effects 1.15
  - https://doc.qt.io/qt-5/qtquick3d-effects-qmlmodule.html
- Shader Effects用 API提供
- Screen Space Ambient Occlusion (SSAO)

マウス/タッチ操作

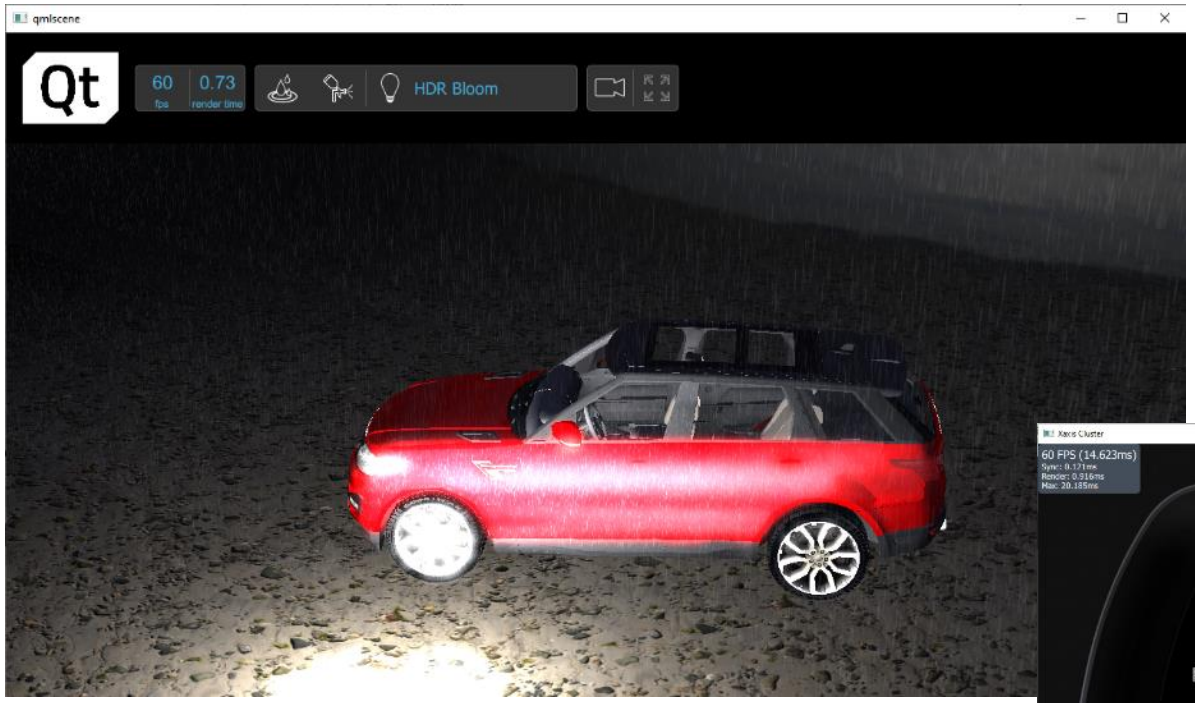
- Ray-cast based picking support
- Model
  - pickable: true
- View3D
  - pick(x, y)
- PickResult
  - objectHit
  - distance
  - scenePosition
  - uvPosition

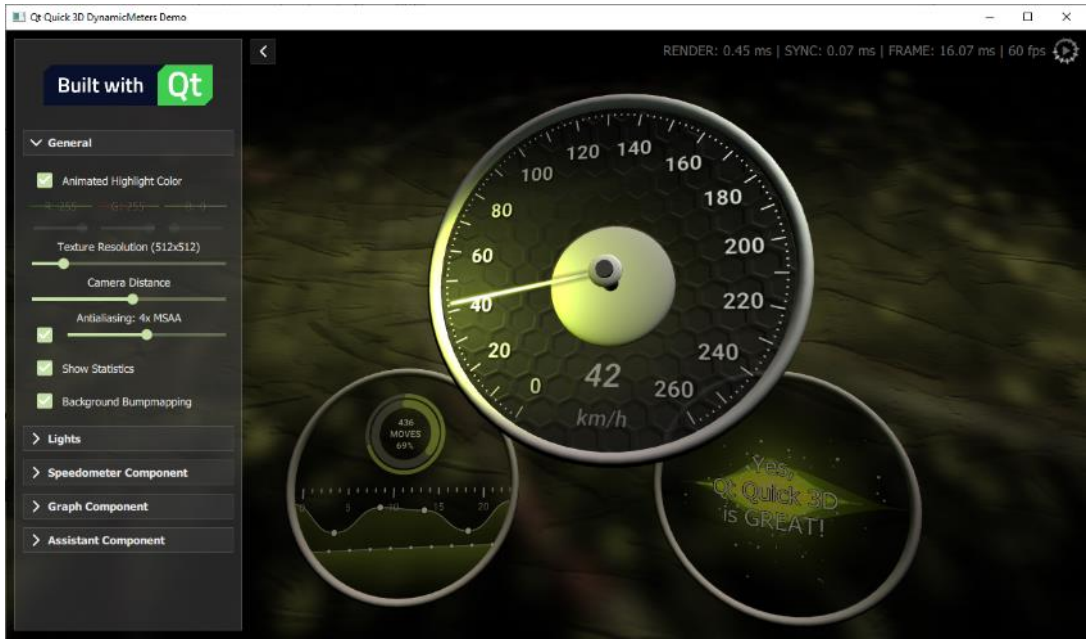
キー操作、Animation

- 既存のQt Quickの機能を利用
  - Keys Property
  - Qt Quick Animation
  - Timeline Animation

Demos for Autmotive Industry - Meter, ADAS, Car

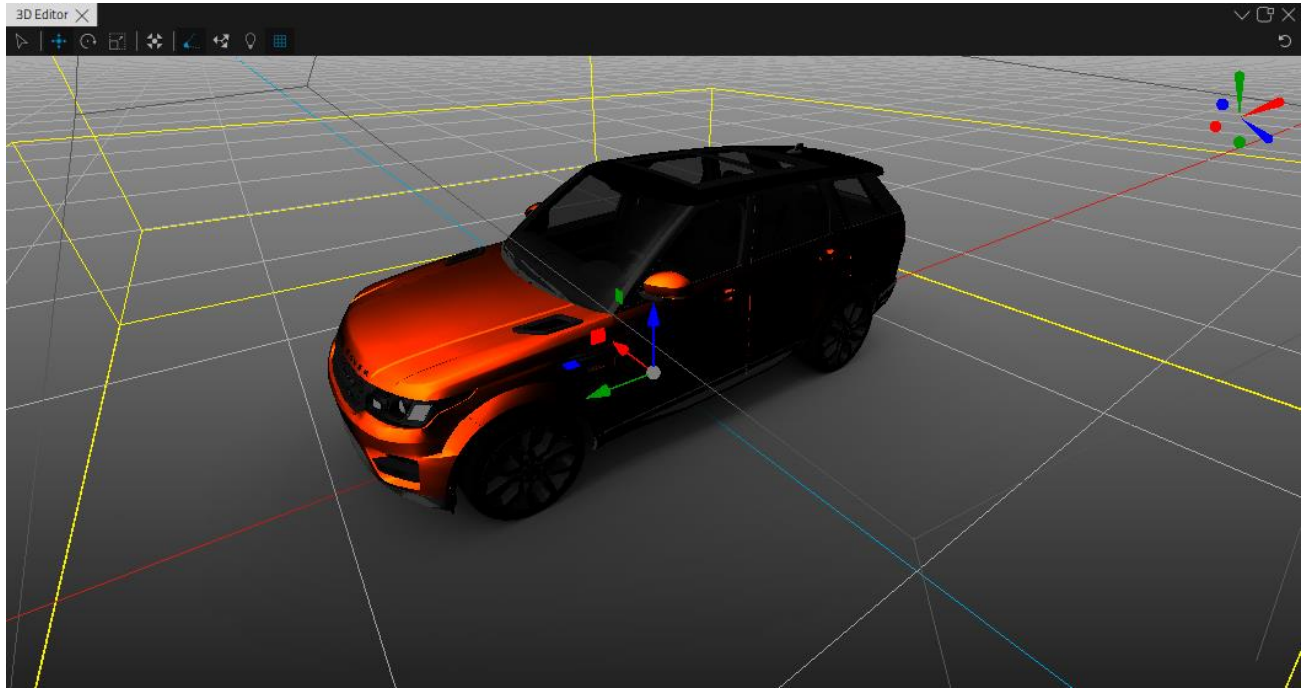
› https://git.qt.io/public-demos/qtquick3d/





サポート 条件

- Qt 5.15
  - OpenGLES 3.1 以上をサポート
  - OpenGLES 2.0 一部機能不可
    - Shadow mapping
    - Image based lighting
    - Multi-sampling
- Qt 6.0 (2020年12月)
  - Qt Rendering Hardware Interface (RHI)
    - OpenGL(ES), Vulkan, Metal, D3D
- Qt Design Studio 1.5
  - 3D Editor
- Dual Licensed
  - GPLv3
  - Commercial




Thank you