January 17 at 08:40

# How to make friends with strangers QML OpenGL context. Part II: Load QML tutorial

Qt * , C ++ * , Programming *

In this article I will try to talk about how to load the QML if you, for some reason, you can not use QQuickView , and the need to work directly with QQuickWindow .
In my case, that of the reasons is that QQuickRenderControl able to work Only QQuickWindow. In your case, this is the reason may be such that you needed to load the QML not because of any file, such as memory, allowing the generation of QML «on the fly", or query the contents of QML, or part thereof, the user - interesting, is not it?

*In case you did not read the beginning: Part I is available at this link .*

In fact, in the task at hand, there is almost nothing complicated enough carefully read the documentation or look into the source QQuickView.

## So everything in order

The first thing we need is QQmlEngine :

```
QQmlEngine qmlEngine * = new QQmlEngine;
```

Next we need QQmlComponent - with its help being downloaded QML. An important feature is that, depending on the source of QML, QQmlComponent can ship it both synchronously and asynchronously. Process is as follows (this is the code used in QQuickView):

```
const QUrl source = QStringLiteral ( "http://example.com/main.qml" );
qmlComponent = new QQmlComponent (& qmlEngine, source);
if (qmlComponent-> isLoading ())
     Connect (qmlComponent, & QQmlComponent :: StatusChanged, componentStatusChanged);
 else
    componentStatusChanged (qmlComponent-> status ());
```

but me personally more impressed this implementation:

```
const QUrl source = QStringLiteral ( "http://example.com/main.qml" ) ;
qmlComponent = new QQmlComponent (qmlEngine) ;
Connect (qmlComponent, & QQmlComponent :: StatusChanged, componentStatusChanged) ;
qmlComponent-> loadURL (source) ;
```

Since in this case the synchronous and asynchronous loading processed identically (and smaller branches in the code - less to go wrong). If necessary, download QML of QString, the code would look like this:

```
const QUrl qmlUrl = QStringLiteral ( "http://example.com/main.qml" ) ;
const QString QML = QStringLiteral ( "Import QtQuick 2.0; Rectangle {color: 'green';}" ) ;
qmlComponent = new QQmlComponent (qmlEngine ) ;
Connect (qmlComponent, & QQmlComponent :: StatusChanged, componentStatusChanged) ;
qmlComponent-> setData (QML .toUtf 8 (), qmlUrl) ;
```

In this case, despite the fact that the line is loaded from QML can specify the URL from which the active QML is associated. This is necessary if the text strings are used QML are any external elements (links to other QML components or files) to be searched, and will be carried out with respect to the transferred URL. Well, we just have to handle the result of loading:

```
void componentStatusChanged (QQmlComponent :: Status status)
{
    Q_ASSERT (! M_rootItem);
    if (QQmlComponent :: Ready! = Status) {
         Return ;
    }

    QObject * rootObject = qmlComponent-> create ();
    QQuickItem * rootItem qobject_cast = <QQuickItem *> (rootObject);
     if (! rootItem) {
         Return ;
    }

    rootItem-> setParentItem (quickWindow-> contentItem ());
    rootItem-> setSize (QSizeF (quickWindow-> width (), quickWindow-> height ()));
}
```

*Important Note: In the above code, deliberately ignored completely as the ownership objects and error handling issues.*

Actually it was all that was needed to address the second part of the original problem . The class implements the above concept, as usual, is available on GitHub: FboQuickView.h , FboQuickView.cpp Well, as always, comments, questions, healthy criticism - welcome. Continued: Part III: Treatment of user input

Qt, QML

⇧ 11 ⇩ 3640 ☆ 47 RSATom 1.3

## Similar publications

About QML and the new REST API Yandex.Disk July 20, 2014 at 14:46

Qt 5.2, from the desire to Google Play April 25, 2014 at 17:10

Model-View in QML. Part Three: Models in QML and JavaScript September 29, 2013 at 21:38

Model-View in QML. Part One: Representations of Component Based June 24, 2013 at 10:29

Model-View in QML. Part is zero, opening June 1, 2013 at 00:51

The insides of the engine QML. Part 2: Binding August 25, 2012 at 20:36

Webinars for QML and QtQuick: Qt Mobility and work with sensors 13 July 2012 at 14:21

Webinars for QML and QtQuick: integration QML and C ++, June 22, 2012 at 16:13

How to quickly start developing on Qt / QML for the BlackBerry PlayBook and get all you jealous of February 27, 2012 at 13:51

PyQt4 and QML February 2, 2011 at 21:26

## Comments ( 1 )

**mapron** January 17, 2015 at 17:18 # 0 ⇧ ⇩

Ohrenet. I was just thinking this morning about such a possibility at all, not even had time to bury the dock.
Somehow missed the first part. Write a sequel!

Only registered users can add comments. Sign in , please.

Lead Frontend-developer

**Support engineer in the Data Center**

Software developer in e-commerce projects

Ruby on Rails front-end developers

Software Engineer microcontrollers

IOS Developer

all vacancies

Design for Landing a share

You want to find forums

UI / UX, prototyping, coding: looking for a superhero

Change in the pattern shop on php

Develop brenbuk

Sticker design

all orders