CamFeatures: QML component to get camera capabilities

This article explains how to use the custom QML Element `CameraFeatures` to get camera capabilities in QML.

Note: This is an entry in the PureView Imaging Competition 2012Q2

Introduction

The standard QML Camera Element allows developers to set most camera capabilities, including flash mode, exposure mode, white balance, ISO modes etc. However there is no in-built mechanism to query whether these modes are supported on a particular device camera and no mechanism for getting an error back if the setting is not supported.

This component allows the camera capabilities to be queried from within QML code, thereby allowing the UI to be customised for the actual capabilities of the supported camera.

Usage

Start copying **camfeatures.cpp *and*** *camfeatures.h* into your project source directory. In the **Icode|main.cpp** file include the **.h**:

```
#include "camfeatures.h"
```

then register the component:

```
qmlRegisterType<cameraFeatures>("CameraFeatures", 1, 0, "CameraFeatures");
```

In your qml file you can now declare the CameraFeatures component:

```
CameraFeatures {
    id: camFeatures
    property int cameraDevice: 0

    function getFeatures()
    {
        var devices;
        devices = getAvailableDevices();
        startGetFeatures(devices[cameraDevice]);
    }

    Component.onCompleted: getFeatures()
    onCameraDeviceChanged: getFeatures()

    onFeaturesRetrieved: camera.start() // camera is your QML camera component
}
```

After the component is completed we can fetch its capabilities by calling `startGetFeatures()` with the device name as a parameter. The list of available devices can be retrieved by first calling `getAvailableDevices()`.

When all the features are stored, the `featuresRetrieved` signal is emitted you can query the `CameraFeatures` object using the JavaScript methods in the following sections to determine exactly what features are supported. In the code above, we also start the QML Camera.

The `QML Camera Element` does not support the front camera (at time of writing), but this component can get its information.

Exposure modes

This section shows how to retrieve the exposure modes - for more information see QCameraExposure (reference doc).

```
bool isExposureSupported(mode)
```

Example:

```
var b = camFeatures.isExposureSupported(CameraFeatures.ExposureAuto)
```

Modes:

```
CameraFeatures.ExposureManual
CameraFeatures.ExposureAuto
CameraFeatures.ExposureNight
CameraFeatures.ExposureBacklight
CameraFeatures.ExposureSpotlight
CameraFeatures.ExposureSports
CameraFeatures.ExposureSnow
CameraFeatures.ExposureBeach
CameraFeatures.ExposureLargeAperture
```


(C) Copyright Nokia 2013. All rights reserved.

```
CameraFeatures.ExposureSmallAperture
CameraFeatures.ExposurePortrait
CameraFeatures.ExposureModeVendor
```

Flash modes

You can refer to flash modes.

```
bool camFeatures.isFlashModeSupported(mode)
```

Example:

```
var b = camFeatures.isFlashModeSupported(CameraFeatures.FlashAuto)
```

Modes:

```
CameraFeatures.FlashOff
CameraFeatures.FlashOn
CameraFeatures.FlashAuto
CameraFeatures.FlashRedEyeReduction
CameraFeatures.FlashFill
CameraFeatures.FlashTorch
CameraFeatures.FlashSlowSyncFrontCurtain
CameraFeatures.FlashSlowSyncRearCurtain
CameraFeatures.FlashManual
```

White balance modes

You can refer to white balance modes.

```
bool camFeatures.isWhiteBalanceSupported(mode)
```

Example:

```
var b =  camFeatures.isWhiteBalanceSupported(CameraFeatures.WhiteBalanceAuto)
```

Modes:

```
CameraFeatures.WhiteBalanceAuto
CameraFeatures.WhiteBalanceManual
CameraFeatures.WhiteBalanceSunlight
CameraFeatures.WhiteBalanceCloudy
CameraFeatures.WhiteBalanceShade
CameraFeatures.WhiteBalanceTungsten
CameraFeatures.WhiteBalanceFluorescent
CameraFeatures.WhiteBalanceIncandescent
CameraFeatures.WhiteBalanceFlash
CameraFeatures.WhiteBalanceSunset
CameraFeatures.WhiteBalanceVendor
```

Iso modes

```
QList<int> camFeatures.getIsoModesSupportedList()
```

Example:

```
ListModel {
        id: isosModel
    }
Component.onCompleted: {
        var isoList = camFeatures.getIsoModesSupportedList()
        isosModel.clear();
        for (var i=0; i<isoList.length; i++)
        isosModel.append( {"iso": isoList[i]} );
}
```

Compensation modes

This function return a list of values that are commonly used by cameras. It come handy to have this for example to have a list of choice to give to the user.

```
QStringList camFeatures.getCompensationsModes()
```

Example:

```
ListModel {
        id: compensationsModel
}
Component.onCompleted: {
    var evList = camFeatures.getCompensationsModes();
    compensationsModel.clear();
    for (i=0; i<evList.length; i++)
        compensationsModel.append( {"compensation": evList[i]} );
}
```

Summary

This component is part of the OMCcam project which uses StarMenu: custom QML component plugin component to build the settings menu with available camera features. OMGcam uses also settings context property in QML to store user settings.

(C) Copyright Nokia 2013. All rights reserved.

Download

File:Camfeatures-v1.0.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.