Using QMicrophone

Introduction

This article explains how to use QMicrophone, a class to easily manage microphone on your device. It will also explain how to record a file and manage volume. The idea behind the volume management is the ability to trigger events with sounds. Take, for example, to change a page, snapping his fingers, or trigger an event when environment noise exceeds a threshold.

Before starting

1. Include **QMicrophone** directory into your project

.pro

```
        include(./QMicrophone/microphone.pri)
```

main.cpp

```
#include <QtGui/QApplication>
#include "qmlapplicationviewer.h"

#include <QtDeclarative>
#include "QMicrophone/qmicrophone.h"

Q_DECL_EXPORT int main(int argc, char *argv[])
{
    QScopedPointer<QApplication> app(createApplication(argc, argv));

    qmlRegisterType<QMicrophone>("Microphone", 1, 0, "Microphone");

    QmlApplicationViewer viewer;
    viewer.setOrientation(QmlApplicationViewer::ScreenOrientationLockPortrait);
    viewer.setMainQmlFile(QLatin1String("qml/QMicrophoneSample/main.qml"));
    viewer.showExpanded();

    return app->exec();
}
```

QML

How to Record Audio Files

```
import QtQuick 1.1
import Microphone 1.0

Rectangle {
Column {
        height: parent.height
        width: parent.width
        anchors.horizontalCenter: parent.horizontalCenter
        Rectangle {
            width: parent.width
            height: 100
            radius: 1

            Text {
                text: qsTr("Start recording for 5 sec")
                horizontalAlignment: Text.AlignLeft
                anchors.centerIn: parent
            }
            MouseArea {
                anchors.fill: parent
                onClicked: {
                    microphone.startRecording("test.wav", 5000);
                }
            }
        }

        Rectangle {
            width: parent.width
            height: 100
            radius: 1

            Text {
                text: qsTr("Start recording")
                anchors.centerIn: parent
            }
            MouseArea {
                anchors.fill: parent
                onClicked: {
                    microphone.startRecording("test.wav");
                }
            }
        }

        Rectangle {
```


(C) Copyright Nokia 2013. All rights reserved.

```
            width: parent.width
            height: 100
            radius: 1

            Text {
                text: qsTr("Stop recording")
                anchors.centerIn: parent
            }
            MouseArea {
                anchors.fill: parent
                onClicked: {
                    microphone.stopRecording();
                }
            }
        }

    }

    Microphone {
        id:microphone
        onRecordingStarted: console.log("Start recording")
        onRecordingStopped: console.log("Stop recording")
    }
}
```

startRecording has two parameters:

- filename where audio will be saved
- recording time in msec. If not set recording will continue until stopRecording will be called

Note: Note that audio is saved in raw PCM format.

How to manage Volume

```
import QtQuick 1.1
import Microphone 1.0

Rectangle {
Microphone {
        id:microphone
        threshold: 150
        onThresholdExceeded: {
            console.log(volume)
        }
    }

    Component.onCompleted: {
        microphone.startVolumeAnalysis();
    }
}
```

This functionality is very useful if you wish to trigger an event when Volume exceed the threshold.

Restrictions

At moment, if you are already using Microphone to record an audio file, you cannot use in the same time volume management, due the fact the resource it's already busy.

Work in progress

- recording and volume management in the same time
- clap or fingers snapping detection
- simple command voice detection, like "up", "down", "left", "right", "stop"...

Source Code

```
You may download the source code from here
```

Related Links

- How to access and manage the Microphone raw data in WP7

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.