Archived:Blur effect with the Qt Graphics View Framework

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Qt Quick should be used for all UI development on mobile devices. The approach described in this article (based UIs created using C++) is deprecated.

Introduction

This code example shows how to use the blur effect in image using the Qt Graphics View Framework. The framework offers a convenient way to apply graphics effect to Graphics Items.

Code

This is a minimal application which blurs a QGraphicsPixmapis using the QGraphicsBlurEffect. A blur effect blurs the source. This effect is useful for reducing details, such as when the source loses focus and you want to draw attention to other elements.

```
#include <QtGui/QApplication>
#include <QGraphicsView>
#include <QGraphicsScene>
#include <QGraphicsPixmapItem>
#include <QGraphicsEffect>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QGraphicsView view;
    view.setScene(new QGraphicsScene);

    QGraphicsPixmapItem *p = view.scene()->addPixmap(QPixmap(":/meditate.png"));

    QGraphicsBlurEffect blur;
    blur.setBlurHints(QGraphicsBlurEffect::QualityHint);
    blur.setBlurRadius(10); // default value is 5px
    p->setGraphicsEffect(&blur);

    view.show();
    return a.exec();
}
```

setBlurRadius() holds the blur radius of the effect. Using a smaller radius results in a sharper appearance, whereas a bigger radius results in a more blurred appearance. `setBlurHints()` holds the blur hint of the effect. The `QualityHint` is used to prefer a higher quality blur.

Source code

Code is downloadable from here: Media:GraphicsEffect-Blur.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.