Archived:Translucent reflection with Qt Graphics View

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 on Qt C++ for the UI) is deprecated.

This article shows how to achieve a translucent reflection effect with Qt's Graphics Framework.

Introduction

Graphics effects are usually done subclassing QGraphicsEffect, but in this example another way is shown.

The mirror object has been designed so that it reflects the item to which it is associated to. Once it has been associated, the mirror places itself under the other item, it updates its size and it makes use of the item public paint method to draw it in a buffer. The buffer image is rotated of 180 degrees and a QLinearGradient (transparent to the background color) is applied. Eventually the buffer is drawn in the scene.

Code

Here the main class which makes the job:

```
#ifndef MIRROR_H
#define MIRROR_H
#include <QGraphicsRectItem>
#include <QGraphicsItem>
#include <QPainter>
#include <QLinearGradient>
#include <QStyleOptionGraphicsItem>
#include <QDebug>

class mirror : public QGraphicsRectItem {

public:
    mirror(): mItem(0){
    }

    void setItem(QGraphicsItem* item){
        mItem = item;
        this->setRect(item->boundingRect().translated(0, item->boundingRect().height()));
        this->setParentItem(mItem);
    }

    void paint(QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget = 0){
        if (!mItem && mItem->isVisible())
            return;

        QSize buffSize = option->rect.size() + QSize(1,1);

        // Option
        QStyleOptionGraphicsItem opt = *option;
        opt.rect = QRect(QPoint(0,0), buffSize);

        // Buffer
        QImage buff(buffSize, QImage::Format_RGB32);
        QPainter p(&buff);
```


(C) Copyright Nokia 2013. All rights reserved.

```
        //Draw into the buffer
        p.fillRect(opt.rect, QBrush(Qt::white));

        // Paint the item in the image starting from 0,0
        p.save();
        qreal dx = mItem->boundingRect().x();
        qreal dy = mItem->boundingRect().y();
        p.translate(-dx, -dy);
        mItem->paint(&p, &opt, widget);
        p.restore();

        QLinearGradient g(0,0,0,buffSize.height());
        g.setColorAt(1, Qt::transparent);
        g.setColorAt(0,Qt::white);
        p.fillRect(opt.rect, g);

        // Draw buffer on the screen
        painter->drawImage(option->rect.adjusted(0,0,1,1), buff.mirrored(), opt.rect);
    }

private:
    QGraphicsItem *mItem;
};
#endif // MIRROR_H
```

Here how to use the mirror class:

```
mirror *m = new mirror;
scene->addItem(m);
QGraphicsPixmapItem* item = new QGraphicsPixmapItem(QPixmap(":/fn.jpg"));
scene->addItem(item);
mirror->setItem(item);
```

Source code

The full code of this example is available here (Media:TranslucentReflection.zip). It runs on any platform supported by Qt.

Related Topic

Graphics View Framework

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.