Archived:Transparent QDialog and QListWidget in Qt

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 (using C++ for the Qt app UI) is deprecated.

Overview

This code snippet demonstrates how to create a transparent `QDialog` that has a `QPixmap` picture as the background and a transparent `QListWidget` in the centre of the screen.

Header

```
#include <QDialog>
#include <QListWidget>

class TransparentDlg : public QDialog
{
public:
    TransparentDlg(QWidget *parent = 0);
    ~TransparentDlg();

    void resizeEvent(QResizeEvent *);
    void paintEvent(QPaintEvent *);

private:
    QPixmap m_pixmapBackground;
    QListWidget* m_listWidget;
};
```

Source

Transparent `QDialog` that shows a picture as the background.

```
#include <QPainter>

TransparentDlg::TransparentDlg(QWidget *parent)
 : QDialog(parent)
{
    // Set QDialog background as transparent
    setAttribute(Qt::WA_NoSystemBackground);

    // Load picture for dialog background
    m_pixmapBackground.load(":/background.png");

    // Create QListWidget
    m_listWidget = new QListWidget(this);
    new QListWidgetItem("Row One",m_listWidget);
    new QListWidgetItem("Row Two",m_listWidget);
    new QListWidgetItem("Row Three",m_listWidget);
    new QListWidgetItem("Row Four",m_listWidget);
}

TransparentDlg::~TransparentDlg()
{
}

void TransparentDlg::resizeEvent(QResizeEvent *event)
{
    // Set size for the background picture
    QSize backSize = size();
    backSize -= QSize(size().width()/4,size().height()/4);
    m_pixmapBackground = m_pixmapBackground.scaled(backSize);

    // Set size for the QListWidget
    QSize listSize = backSize - QSize(40,40);
    QPoint p = QPoint((size().width()-listSize.width())/2,
        (size().height()-listSize.height())/2);
    m_listWidget->setGeometry(QRect(p,listSize));
}

void TransparentDlg::paintEvent(QPaintEvent *event)
{
    QDialog::paintEvent(event);

    QPainter p(this);
    p.drawPixmap((size().width()-m_pixmapBackground.size().width())/2,
        (size().height()-m_pixmapBackground.size().height())/2,m_pixmapBackground);
}
```

To make `QListWidget` look transparent, define a new style and set a style sheet for it using `QApplication::setStyleSheet()`.

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
```


(C) Copyright Nokia 2013. All rights reserved.

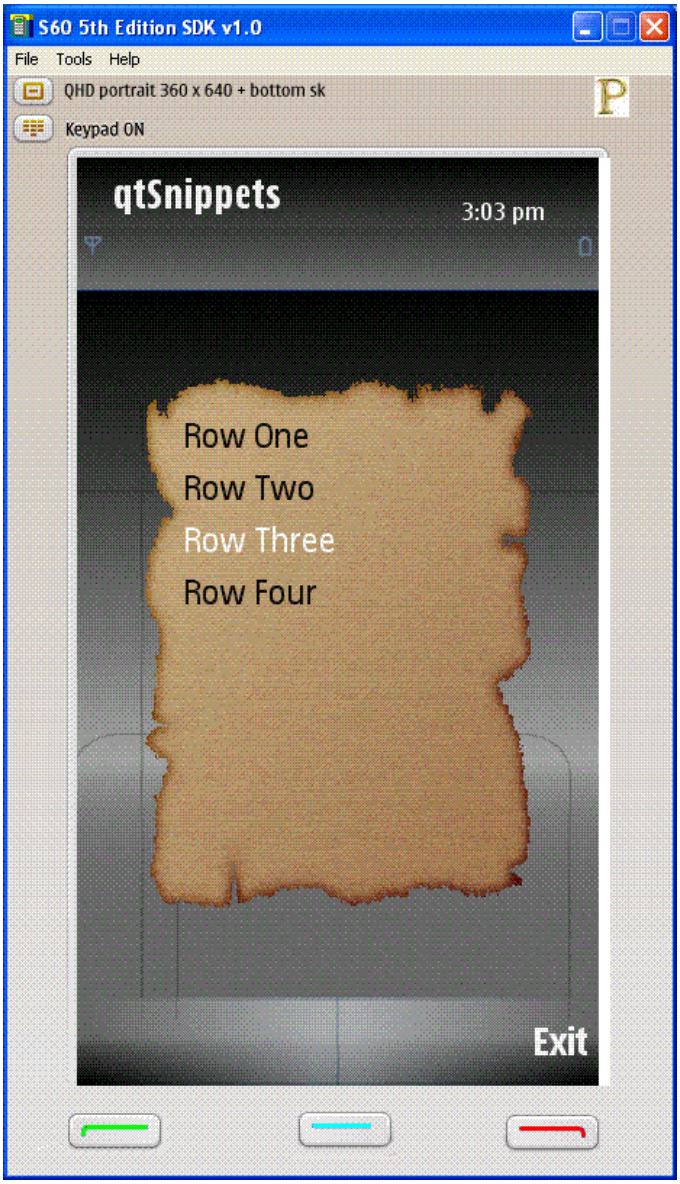
```
    // Set style for the QListWidget and QListWidgetItem
    QString style;
    style += "QListWidget {background-color: transparent;}";
    style += "QListWidget::item {background-color: transparent;}";
    style += "QListWidget::item {selection-color: white;}";
    style += "QListWidget::item {color: black;}";
    a.setStyleSheet(style);

    qtSnippets w;
    w.showMaximized();
    return a.exec();
}
```

Postconditions

`QDialog` has a transparent background.

See also

Qt Style Sheets Reference



(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.