Image Selection Dialog In Qt

Basic Idea

In this section we will display images in Grid View. We will use `QGridLayout` to display the content in grid layout form. The `QListView` is used to store the image in it and `addWidget()` is used to add the images of `QListView` to the `QGridLayout` widget. This Image selection dialog uses `QFutureWatcher` so that it will make the UI non-responsive

Implementation

ImageSelectionDialog.cpp

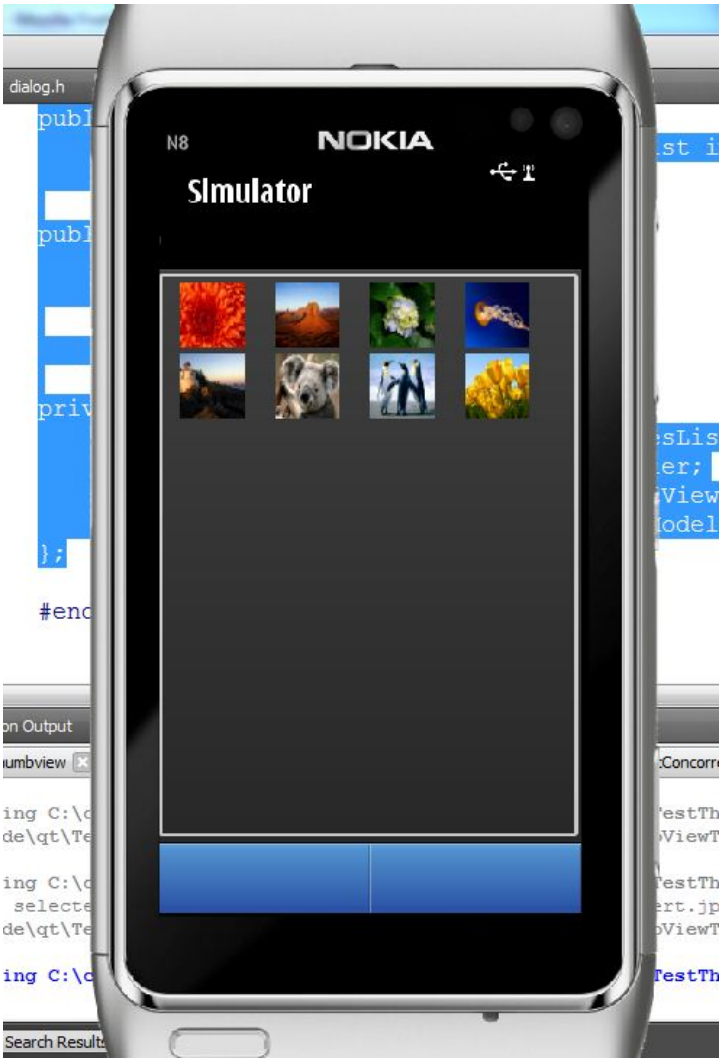
```
const int imageSize = 60;

QImage scale(const QString &imageFileName)
{
    QImage image(imageFileName);
    return image.scaled(QSize(imageSize, imageSize), Qt::IgnoreAspectRatio, Qt::SmoothTransformation);
}

ImageSelectionDialog::ImageSelectionDialog(QStringList imageNamesList, QWidget *parent) :
    QDialog(parent), m_imageNamesList(imageNamesList)
{

    QGridLayout* m_gridLayout = new QGridLayout(this);
    m_imageListView = new QListView(this);
    m_gridLayout->addWidget(m_imageListView);

    m_imageListView->setViewMode( QListView::IconMode );
    m_imageListView->setUniformItemSizes(true);
    m_imageListView->setSelectionRectVisible(true);
    m_imageListView->setMovement(QListView::Static);
    m_imageListView->setSelectionMode(QListView::SingleSelection);
    m_imageListView->setEditTriggers(QAbstractItemView::NoEditTriggers);

    m_imageListView->setResizeMode(QListView::Adjust);

    m_standardModel = new QStandardItemModel(this);
    m_imageListView->setModel(m_standardModel);

    m_imageScaler = new QFutureWatcher<QImage>(this);
    connect(m_imageScaler, SIGNAL(resultReadyAt(int)), SLOT(showImage(int)));
    connect(m_imageScaler, SIGNAL(finished()), SLOT(finished()));

    m_imageScaler->setFuture(QtConcurrent::mapped(m_imageNamesList, scale));

    connect(m_imageListView, SIGNAL(doubleClicked(QModelIndex)), SLOT(imageClicked(QModelIndex)));
}

ImageSelectionDialog::~ImageSelectionDialog()
{
    m_imageScaler->cancel();
    m_imageScaler->waitForFinished();
}

void ImageSelectionDialog::showImage(int num)
{
    QStandardItem* imageitem = new QStandardItem();
    imageitem->setIcon(QIcon(QPixmap::fromImage(m_imageScaler->resultAt(num))));
    m_standardModel->appendRow(imageitem);
}

void ImageSelectionDialog::finished()
{
}

void ImageSelectionDialog::imageClicked(QModelIndex index)
{
    if(index.row() < m_imageNamesList.count()) {
        qDebug() << "image selected " << m_imageNamesList.at(index.row());

        close();
    }
}
```

ImageSelectionDialog.h

```
class ImageSelectionDialog : public QDialog
{
    Q_OBJECT

public:
    explicit ImageSelectionDialog(QStringList imageNamesList, QWidget *parent = 0);
    ~ImageSelectionDialog();

public Q_SLOTS:
    void showImage(int num);
    void finished();

    void imageClicked(QModelIndex);

private:
```


(C) Copyright Nokia 2013. All rights reserved.

```
    QStringList m_imageNamesList;
    QFutureWatcher<QImage>*     m_imageScaler;
    QListView*                  m_imageListView;
    QStandardItemModel*         m_standardModel;
};
```



(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.