QML Sliding Pages

This article shows how to create a book-like page transition using a simple QML ListView element.

Overview

Many of you have maybe used the OffScreen's books apps available on OVI store. The way that offscreen has chosen to display pages is simple, clever and really usable, and better than many other solutions I've seen implemented on Nokia's or other phones.

You can watch this component in action here: The media player is loading...

Sliding Pages Component

Here are the QML files of this component.

main.qml The Main.qml shows you how to create an instance of the component. Declaring the component size is important because it could happen that the delegate's width is 0.

```
import QtQuick 1.0
import "book"

Rectangle {
    width: 480
    height: 800
    color: "gray"

    Book {
        width: parent.width
        height: parent.height
    }
}
```

book/Book.qml This file defines the component and the delegate. Here you don't have to define any size since it make use of the one defined in the main file. Animations could be added to make it fancy but I preferred to keep this example as simple as possible for a better understanding. Further improvements are left to the reader as exercise.

```
import QtQuick 1.0
Item {
    id: book
    anchors.fill: parent
    Component {
        id: myPageDelegate
        Rectangle {
            id: page
            color: "white"
            height: parent.heigh

            //BORDER
            border.color: "black"
            border.width: 1
            Component.onCompleted: page.width = book.width

            Rectangle {
                anchors.fill: parent
                anchors.margins: 15
```


(C) Copyright Nokia 2014. All rights reserved.

```
                Text {
                    anchors.top: parent.top
                    text: modelData
                }
                Text {
                    anchors.bottom: parent.bottom
                    anchors.horizontalCenter: parent.horizontalCenter
                    color: "grey"
                    text: index
                }
            }
        }
    }

    ListView {
        id: list
        anchors.fill: parent
        model: dataModel
        delegate: myPageDelegate
        orientation: ListView.Horizontal
        snapMode: ListView.SnapToItem
        spacing: 5
    }
}
```

Full code is downloadable: File:SlidingPages.zip

Conclusion

This example shows how versatile is the `ListView` element. With few lines a QML developer can customize `ListView` adding new features which could make your application experience better from the user point of view.

These pages can display images, text and so on - even `QWebPages` with videos! Moreover usage of gradients or background images can be used to add even more eye candy.

(C) Copyright Nokia 2014. All rights reserved.