Creating custom QML Components - NumberSpinnerComponent

Introduction

This article shows how to create a simple custom component in QML - `NumberSpinner`. `NumberSpinner` is a variant of the spinning wheel or the BusyIndicator component. The difference is that it shows the progress of a operation, like loading an image from the internet or that progress of a HTTPRequest. Although it can be argued that a progressbar is sufficient for the above purpose, `NumberSpinner` can be used in cases where there is insufficient horizontal space to legibly show progress.

During download, the progress is indicated

After download, the spinner is hidden.

Implementation

The component created below has been created using Symbian's QML BusyIndicator element. To port it to the Harmattan platform simply change the import statement in the NumberSpinnerComponent.qml file.

```
import QtQuick 1.0
import com.nokia.symbian 1.0

Item {
     property int value: 0
     property bool animate: false

    BusyIndicator {
        id:spinWheel
        running: animate
        width: 100
        height : 100
        x : -width/2
        y : -height/2
    }
    Text {
        id:valueTXT
        color: "red"
        text : value
        anchors.horizontalCenter: spinWheel.horizontalCenter
        anchors.verticalCenter: spinWheel.verticalCenter
    }
}
```

The component has a spinner component that can be animated. Additionally, there is textbox that shows the progress at the center of it. Note that there are two properties of the NumberSpinner that are accessible outside to control the component. These are animate(bool type) and value(int type), whose nomenclature is self explanatory.

Usage

The usage of this component is illustrated using a Image loading example. The progress of image load is shown in the NumberSpinner and it automatically

(C) Copyright Nokia 2013. All rights reserved.

disappears after completion of download. Logical sequence of events is:

1. Initiate download of a relatively large image.
2. Upon start of download, display the spinner with number.
3. Start a timer to that fetches the progress of download and show it in the NumberSpinner component
4. Hide the NumberSpinner when download is complete or when progress = 1.

```
import QtQuick 1.0
import com.nokia.symbian 1.0

Rectangle {
    width: 360
    height: 640
    property int progress: 0

    NumberSpinnerComponent {
        id:spinner
        anchors.horizontalCenter: parent.horizontalCenter
        y : 350
        visible : false

    }
    Timer { id:ticker
             interval: 100; repeat: true
             onTriggered: { spinner.value = img.progress*100;  }
    }

    Image {
        id:img
        anchors.horizontalCenter: spinner.horizontalCenter
        anchors.verticalCenter: spinner.verticalCenter
        width: 350
        height : 360
        onStatusChanged: {
            if(status == Image.Loading) {
                spinner.visible = true;
                ticker.start()
                spinner.animate = true;
            }
            else if(status == Image.Ready) {
               // spinner.visible = false;
                spinner.animate = false;
                ticker.stop()
            }
        }

    }

    ButtonRow {
        anchors.horizontalCenter: parent.horizontalCenter
        width: 200
        Button {
            text: "Download"
            onClicked: {
                img.source = "http://designdelight.net/wp-content/uploads/2011/07/nokia_n9_ux_guidelines.jpg"
            }
        }
    }
}
```

Download

The project files are available for download File:NumericSpinner.zip

Summary

Creation of a custom component `NumberSpinner` is illustrated.

(C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.