Creating a QML Marquee Component

Introduction

This article shows **how to create a Marquee component by using QML**. A Marquee is a UI component that shows scrolling text.

Creating the Component

Creating the static structure

Start by creating an empty Marquee.qml file in a Qt Quick project.

Then, let's **define a Rectangle** that will contain the scrollable text. Since only the text contained in the Rectangle's boundaries must be visible, let's **set its *clip* property to *true***. The Rectangle's height will be later defined by binding it to the Marquee's text height.

```
Rectangle {
    clip: true
    width: parent.width
}
```

In order to allow for customization of the Marquee text, the Component defines a series of **properties that can be used to modify the Marquee's appearance and scrolling speed**:

```
// text to be displayed by the Marquee
property string text

// top/bottom text padding
property int padding : 10

// the font size used by the Marquee
property int fontSize : 20

// the scrolling animation interval
property int interval : 100
```


(C) Copyright Nokia 2014. All rights reserved.

```
// the text color
property color textColor: "black"
```

Now, a **Text object is added to the root Rectangle**, in order to display the Marquee text. The properties defined above are used to bind the Text properties, as its font size and text color. Finally, the **root Rectangle height is bound to the Text's height** (plus the padding defined by the Component *padding* property)

```
Rectangle {
    clip: true
    width: parent.width
    height: marqueeText.height + padding

    [...]

    Text {
 anchors.verticalCenter: parent.verticalCenter
 id: marqueeText
 font.pointSize: fontSize
 color: textColor
 text: parent.text
    }
}
```

Adding the animation

The Component defined above defines the base structure for the Marquee component. Now, **it's time to animate the Text**: in this article, text will be scrolled **from right to left**, restarting once its right boundary has exited the parent Rectangle. With small modifications the same code can be used to create different type of scrollings (for instance, left to right, or alternate).

First, let's **place the Text in its starting position**, by setting its *x* property.

```
    Text {

 [...]

 x: parent.width
    }
```

The **animation is implemented by using a Timer object**, that will run with the *interval* defined by the above property. The following Timer is added to the Component:

```
Timer {
    interval: parent.interval
    onTriggered: moveMarquee()
    running: true
    repeat: true
}
```

The actual **scrolling animation is performed by the JavaScript moveMarquee() function**, defined below:

```
function moveMarquee()
{
    if(marqueeText.x + marqueeText.width < 0)
    {
```


(C) Copyright Nokia 2014. All rights reserved.

```
 marqueeText.x = marqueeText.parent.width;
    }
    marqueeText.x -= 10;
}
```

Using the Marquee

The Marquee component can be easily used as shown in the code snippet below. To use it, be sure to **import Marquee.qml in your qml file**.

```
Rectangle {
    width: 360
    height: 640

    Marquee {
 width: 300
 padding: 2
 anchors.centerIn: parent
 text: "This is a very very long text that should never end!"
 color: "black"
 textColor: "white"
    }
}
```

A video of the Marquee in action on a Nokia N8 device can be seen here:

The media player is loading...

Related content

The **complete Qt Creator project** for the Marquee component can be downloaded here: File:MarqueeQuickApp.zip

A SIS file for Symbian^3 devices is also available here: File:Marqueequickapp qt-4 7 0 exp m1 1 0 s3 v0 9.sis.zip

(C) Copyright Nokia 2014. All rights reserved.