Animation in Qt Quick using a strip of frames in a single image

This article explains a method to make an animation in Qt Quick using a number of images in a single **.png**.

Tip: There are a lot of methods in Qt Quick for animating images. Other approaches include using an animated GIF and displaying individual frames, using property animations rather than a timer to move the visible area, or constructing the image out of a number of parts, which are animated together.

Introduction

This article explains one (of many) methods to animate an image in Qt Quick. In this approach the animation frames are drawn across the horizontal width of a png file. The QML code clips the view to show only one image at a time, and uses a timer to progress the frames into the visible area.

Implementation

To do this, first obtain a PNG file with horizontally aligned images (for this example we use the one below):

Create a new project in the Qt SDK. Add a QML file for cropping the image, and moving frames if needed. In this case we name it **Kameha.qml**:

```
 import QtQuick 1.1
   Rectangle{
    id: animacion
    //We make the size of one frame, in this case is 180 x 110
    width: 180
    height: 110
    // this is trasparent color, very important if you want to put the animation if some
scene
    color: "#00000000"
    // Add a property that make the count of the frames
    property int contador:0
    // clip true, if you put clip false, you see the image complete, not only the frame
you want
    clip: true
    // Put the image you want to animate
    Image{
        id:animacion1
        x: 0
        y: 0
        source:"kameha.png"
        //If you want to scale the animation this is very important.
        smooth: true
    }
     //This function if called by the timer in the main page.
     function advance(){
        ++animacion.contador
        if(animacion.contador<=9){animacion1.x = animacion.contador*180*(-1)}
        else if (animacion.contador>9){animacion.contador=0}
     }
 }
```

Now in your *main.qml'* put this code to load your file and advance the frame:

```
 import QtQuick 1.1
 Rectangle {
```


(C) Copyright Nokia 2014. All rights reserved.

```
    width: 640
    height: 360
    color: "#335b80"
    //import our qml
    Kameha{
        id:kameha
        x: 212
        y: 134
    }
    //put a timer to make the animation
    Timer{
        id:timer
        // This interval will be ok, depends the quantity of frames you want
        interval:60
        repeat:true
        // You can put running false if you want to make actions and call the timer
        running:true
        onTriggered: kameha.advance()
    }
}
```

Now "Run" the app to see the image animate.

Summary

This article shows one approach for animating a graphic. Please add comments if you have any questions.

Version Hint

Windows Phone: [[Category:Windows Phone]]
[[Category:Windows Phone 7.5]]
[[Category:Windows Phone 8]]

Nokia Asha: [[Category:Nokia Asha]]
[[Category:Nokia Asha Platform 1.0]]

Series 40: [[Category:Series 40]]
[[Category:Series 40 1st Edition]] [[Category:Series 40 2nd Edition]]
[[Category:Series 40 3rd Edition (initial release)]] [[Category:Series 40 3rd Edition FP1]] [[Category:Series 40 3rd Edition FP2]]
[[Category:Series 40 5th Edition (initial release)]] [[Category:Series 40 5th Edition FP1]]
[[Category:Series 40 6th Edition (initial release)]] [[Category:Series 40 6th Edition FP1]] [[Category:Series 40 Developer Platform 1.0]] [[Category:Series 40 Developer Platform 1.1]] [[Category:Series 40 Developer Platform 2.0]]

Symbian: [[Category:Symbian]]
[[Category:S60 1st Edition]] [[Category:S60 2nd Edition (initial release)]] [[Category:S60 2nd Edition FP1]] [[Category:S60 2nd Edition FP2]] [[Category:S60 2nd Edition FP3]]
[[Category:S60 3rd Edition (initial release)]] [[Category:S60 3rd Edition FP1]] [[Category:S60 3rd Edition FP2]]
[[Category:S60 5th Edition]]
[[Category:Symbian^3]] [[Category:Symbian Anna]] [[Category:Nokia Belle]]

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.