Animated Page Navigation using QML

This article demonstrates how to create animated contents and page navigation using Qt Quick Components.

Introduction

Using Qt Quick's States and Transition we can create different forms of animation in our application. In this article we will see how to use easing effects for displaying animated content on the home screen of the application and also while navigating the pages. This application is build with a basic concept, where there are four icons on the home screen and when user clicks on the icons it brings a new page which emerges from the middle of the screen and hide the icons of the home screen. The icons are also animated, every time we in/out the application home screen the icons display in an animated manner.

Home Screen Animated Icons

Animated Second Page

.

Implementation

First we create an empty *Hello World* project selecting *Qt Quick Component for Symbian* template. This empty project will have the basic look and feel of the Qt Quick Component for Symbian application, including the Status Bar, Tool Bar.

First we create four icons on the home screen using Image element.

```
Image {
    id: image1
    source: "Images/doctor.png"
    x:-150
    height:128
    width:128
    anchors.top: parent.top
    anchors.topMargin: 100
}
```

Set States and Transition for each images on the home screen so that the *top-left* and *bottom-left* icons comes from left to right and *top-right* and *bottom-right* icons comes from right to left and seats on the home screen. All these icons have an easing effect.

```
states : State {
```


(C) Copyright Nokia 2014. All rights reserved.

```
    name: "animate" ; when: window.isAnimateIn == true
    PropertyChanges { target: image1; x: 40;}
}

transitions: Transition {
    NumberAnimation { properties: "x"; easing.type: Easing.OutBack; duration: 1000 }
}
```

When user clicks on the icon it select an image that should be displayed on the new page.

```
onClicked: {
    //hide the icons and display the popup window
    var imagePath = "Images/1.jpeg";
    deAnimateIcons(imagePath);
    window.isPopUpWindow = true;
}
```

The `deAnimateIcons()` function reverse the icons animation that happens while loading the home screen. That means when the user clicks on the icon not only the new page appears but also the icons goes out of the windows in an animated way.

```
//hide the icons and display the pupup window
function deAnimateIcons(imageSource) {
    window.isAnimateIn=false;
    popUpItem.visible=true;
    popUpImage.source = imageSource;
 }
```

The new page that comes out from the middle of the window has a `scale` property *0* initially. When user clicks the icon we change its `scale` property to *1* with an easing effect. Thus it seems that the page is coming out from the middle of the screen.

```
//popup window
Item {
    id: popUpItem
    anchors { left: parent.left; right: parent.right; top: statusBar.bottom; bottom:
toolBar.top }
    scale: 0.0
    Rectangle {
        id: popUpRect
        color: "black";
        opacity: 1
        anchors.fill: parent;
        Image {
            id: popUpImage
            source: ""
            anchors.horizontalCenter: parent.horizontalCenter
            anchors.verticalCenter: parent.verticalCenter
        }
        states: State {
            name: "Rectangle"; when: window.isPopUpWindow == true
            PropertyChanges { target: popUpRect; x: 0 }
            PropertyChanges { target: popUpItem;       scale: 1 }
        }
        transitions: Transition {
            NumberAnimation { properties: "x,scale"; duration: 500; easing.type:
```


(C) Copyright Nokia 2014. All rights reserved.

```
Easing.InOutQuad }
        }
    }
} // end of popup window
```

Note: There are many similar code in this project with minimal changes, for example creating four Image, `MouseArea`, States and Transition for each images etc, which are not added in the article page to reduce the size of the article. Also the pictures on the article page will not give you a proper understanding of the animation, so please run the sample code for better understanding.

Summary

Qt Quick gives us an easy and free environment for animation. This idea brought to my mind when I saw an OpenGL ES project on Ovi and though of making an animated home screen of similar kind using Qt Quick. For more on Qt Quick animation please refer here.

Source Code

The full source code of the example is available here: File:AnimatedPageNavigation.zip

(C) Copyright Nokia 2014. All rights reserved.