Implementing of fading animation with QML

This code snippet shows how to implement a fading animation with QML.

Overview

In the following QML code there are four ways to implement the fading of a `Rectangle` element using `PropertyAnimation` when the rectangles are clicked. Different implementations suit different purposes, so use the most suitable one.

The code snippet is meant to be run with `qmlviewer`.

Preconditions

- Qt 4.7 or higher is installed on your platform.

Source

ui.qml

```
import Qt 4.7

Rectangle {
    width: 640; height: 480
    color: "black"

    Row {
        anchors.centerIn: parent; spacing: 10

        // Implements fading using Behavior
        Rectangle {
            id: fadingByBehavior

            width: 100; height: 100; color: "red"
            Behavior on opacity { PropertyAnimation { duration: 1000 } }

            MouseArea { anchors.fill: parent; onClicked: fadingByBehavior.opacity = 0.2 }
        }


        // Implements fading using States and Transitions
        Rectangle {
            id: fadingByState

            width: 100; height: 100; color: "green"

            MouseArea { anchors.fill: parent; onClicked: fadingByState.state = "hidden" }

            states: State { name: "hidden"; PropertyChanges { target: fadingByState; opacity: 0.2 } }

            transitions: Transition {
                from: "*"
                to: "hidden"
                PropertyAnimation { target: fadingByState; property: "opacity"; duration: 1000 }
            }
        }


        // Implements fading using animation as an element
        Rectangle {
            id: fadingByAnimationElement

            width: 100; height: 100; color: "blue"

            MouseArea { anchors.fill: parent; onClicked: hideAnimation.start() }

            PropertyAnimation {
                id: hideAnimation
                target: fadingByAnimationElement; property: "opacity"; to: 0.2; duration: 1000
            }
        }


        // Implements fading directly from the event handler
        Rectangle {
            id: fadingFromSignalHandler

            width: 100; height: 100; color: "yellow"

            MouseArea {
                anchors.fill: parent
                onClicked: PropertyAnimation {
                    target: fadingFromSignalHandler; property: "opacity"; to: 0.2; duration: 1000
                }
            }
        }
    }
}
```

Postconditions

The code snippet demonstrated four different ways to implement a fading animation with `PropertyAnimation`.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.