Implementing parent change animation with QML

This code snippet shows how to implement an animation when the QML element's parent changes.

Overview

In the following code snippet there are two QML files. The file **ui.qml** contains the main view which places six differently coloured rectangles in the 3 x 2 QML `Grid` element. **MyRect.qml** contains the implementation of our self-made rectangle which will scale to full screen when clicked. At the same time its parent is changed.

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 {
    id: view

    width: 800; height: 480
    color: "black"

    Grid {
        id: grid

        property int itemWidth: grid.width / grid.columns
        property int itemHeight: grid.height / grid.rows

        anchors.centerIn: parent
        width: parent.width * 0.75; height: parent.height * 0.75
        spacing: 10; columns: 3; rows: 2

        MyRect {
            width: grid.itemWidth
            height: grid.itemHeight
            color: "red"
        }

        MyRect {
            width: grid.itemWidth
            height: grid.itemHeight
            color: "green"
        }

        MyRect {
            width: grid.itemWidth
            height: grid.itemHeight
            color: "blue"
        }

        MyRect {
            width: grid.itemWidth
            height: grid.itemHeight
            color: "yellow"
        }

        MyRect {
            width: grid.itemWidth
            height: grid.itemHeight
            color: "brown"
        }

        MyRect {
            width: grid.itemWidth
            height: grid.itemHeight
            color: "gray"
        }
    }
}
```

MyRect.qml

When `MouseArea` is clicked, the state `maximized` is entered. This is triggered by the `rect.maximized` property. `Transition` will animate the change of parent by using `ParentAnimation`.

The z-order of `container` is changed between the states in order to prevent the rectangle from going behind other rectangles when coming back from the `maximized` state to the `default` state.

Because we are using the QML `Grid` to position the `MyRects` in the scene, we want to have an invisible `Item` to be left in the grid when the `rect` is re-parented to the `view`. Without the `Item`, the `grid` would rearrange its children when `MyRect` is re-parented to the `view`. This behavior is not desired here.

```
import Qt 4.7
```


(C) Copyright Nokia 2013. All rights reserved.

```
Item {
    id: container

    property alias color: rect.color

    Rectangle {
        id: rect

        property bool maximized: false

        width: parent.width; height: parent.height; radius: 10

        MouseArea { anchors.fill: parent; onClicked: rect.maximized = !rect.maximized }

        states: State {
            name: "maximized"
            when: rect.maximized
            ParentChange {
                target: rect
                parent: view
                x: 0; y: 0
                width: parent.width; height: parent.height
            }
            PropertyChanges { target: rect; radius: 0; z: 1 }
            PropertyChanges { target: container; z: 1 }
        }

        transitions: Transition {
            from: "*"
            to: "maximized"
            reversible: true    // reverses the animation when going back to default state
            ParentAnimation {
                target: rect
                SequentialAnimation {
                    PropertyAction { target: container; property: "z" }
                    PropertyAnimation {
                        target: rect
                        properties: "x,y,width,height,radius"
                        easing.type: "InOutQuart"; duration: 1000
                    }
                }
            }
        }
    }
}
```

Postconditions

The snippet demonstrated using `ParentChange` and `ParentAnimation` QML elements to implement a smooth animation of a self-made rectangle to full screen when the rectangle was clicked. Also, the animation was reversed when the maximised rectangle was clicked again.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.