Custom Drop Down Menu

This article demonstrates how to create drop down menu using Qt Quick/QML Elements.

Introduction

We can use built-in ToolBar Element to create a toolbar with drop down menu, but in this article we will create custom toolbar on top of the screen with drop down menu.

Drop Down Menu

Idea to Implementation

We create a *ToolBar.qml* file with a fixed rectangle and gradient color and set an icon for the drop down menu.

ToolBar.qml

```
import Qt 4.7
Item {
    id: toolbar
    width:  parent.width
    height: 50
    signal menuClicked()
    z: 1
    Rectangle {
        id: menubar
        y: 0
        height: 50
        x: 0
        width: parent.width
        state: "shown"
        gradient: Gradient {
            GradientStop { position: 0.08; color: "lightblue" }
            GradientStop { position: 0.5; color: "grey" }
            GradientStop { position: 1.0; color: "black" }
        }
        Image {
            id: menuButton
            source: "menuicons/menu_icon.png" //
            x: parent.width - width - 10
            anchors.verticalCenter: parent.verticalCenter
            opacity: quitIcon.pressed ? 0.5 : 1.0
            MouseArea {
                id: quitIcon
                width: parent.width + 20
```


(C) Copyright Nokia 2014. All rights reserved.

```
                height: parent.height + 20
                anchors.centerIn: parent
                onClicked: {
                    toolbar.menuClicked();
                }
            }
        }
    }
    Image {
        source: "menuicons/bottom_shadow.png"
        width: parent.width
        x:0
        y:50-1
    }
}
```

In *main.qml* we call the `ToolBar` and set the `state` of menu (see *MainMenu.qml*) to **hidden** and **shown** on `onMenuClicked` action.

main.qml

```
  ToolBar {
        id: toolBar
        onMenuClicked: {
            if(menu.state!="shown") {
                menu.state = "shown";
            } else {
                menu.state = "hidden";
            }
        }
    }
```

We add a rectangle to display the changing image on each menu clicked. By default we set to a particular image to the image source.

```
Rectangle
    {
        id: thumbnailsRect
        width: 150
        height:  113
        x: 350
        y:200
        Image {
            id: thumbnailsImage
            source: "thumbnails/1.jpeg"
        }
    }
```

To create the menus, first we create *MainMenu.qml*. In main rectangle we creates two rows inside one column. Inside the rows we implement the menu icon (see *MenuIcon.qml*). On clicking the menu items it calls the `changeImage()` function which changes the display image on the screen.

```
function changeImage(aImageName)
    {
        thumbnailsImage.source = aImageName;
```


(C) Copyright Nokia 2014. All rights reserved.

```
    }
```

Source Code

The full source code of the example is available here: File:MenuN950.zip

Note: The files *MenuIcon.qml* and *MainMenu.qml* are not added in this article page, please refer to the source code for the same.

(C) Copyright Nokia 2014. All rights reserved.