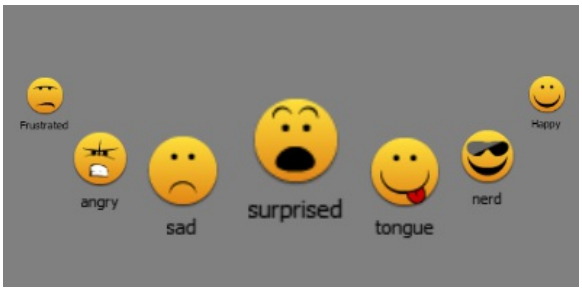
PathView QML Element

Introduction

Here is one example of how to use PathView QML Element in your Qt Quick Application. The PathView element lays out model-provided items on a path. A PathView displays data from models created from built-in QML elements like ListModel and XmlListModel, or custom model classes defined in C++ that inherit from QAbstractListModel.



ListView QML File

On this QML File's ListView you can write the images and text you want to show on pathview.

```
import Qt 4.7

// list of items
ListModel {
    ListElement { name: "nerd"; icon: "pics/chat_smiley_nerd.png" }
    ListElement { name: "Happy"; icon: "pics/chat_smiley_happy.png" }
    ListElement { name: "Frustrated"; icon: "pics/chat_smiley_frustrated.png" }
    ListElement { name: "angry"; icon: "pics/chat_smiley_angry.png" }
    ListElement { name: "sad"; icon: "pics/chat_smiley_sad.png" }
    ListElement { name: "surprised"; icon: "pics/chat_smiley_surprised.png" }
    ListElement { name: "tongue"; icon: "pics/chat_smiley_tongue.png" }
}
```

PathView QML File

Here is the code to show pathview.

```
import Qt 4.7

Rectangle {
    id: rect
    width: 400; height: 200
    radius: 0

    // bg
    color: "gray"

    Component {
        id: appDelegate
        Item {
            width: 200; height: 200
            scale: PathView.iconScale

            // images
            Image {
                id: myIcon
                y: 40; anchors.horizontalCenter: parent.horizontalCenter
                source: icon
                smooth: true
            }
            // bottom lable
            Text {
                anchors { top: myIcon.bottom; horizontalCenter: parent.horizontalCenter }
                text: name
                smooth: true
            }

            MouseArea {
                anchors.fill: parent
                // here you write the code you want to see on onclick event
                onClicked: view.currentIndex = index + 1
            }
        }
    }

    // bg of listItem/Pathview Item
    Component {
        id: appHighlight
        Rectangle { width: 80; height: 80; color: "Transparent" }
```


(C) Copyright Nokia 2013. All rights reserved.

```
    }

    PathView {
        id: view
        anchors.fill: parent
        highlight: appHighlight
        preferredHighlightBegin: 0.5
        preferredHighlightEnd: 0.5
        focus: true
        model: ListViewQML {}
        delegate: appDelegate
        path: Path {
            startX: 20
            startY: 50
            PathAttribute { name: "iconScale"; value: 0.5 }
            PathQuad { x: 200; y: 150; controlX: 50; controlY: 200 }
            PathAttribute { name: "iconScale"; value: 1.5 }
            PathQuad { x: 390; y: 50; controlX: 350; controlY: 200 }
            PathAttribute { name: "iconScale"; value: 0.5 }
        }
    }
}
```

Source Code

Here is the Sample source code. You can download it from File:PathView.zip.

~~ chintandave_er 14:35, 5 February 2011 (UTC)

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.