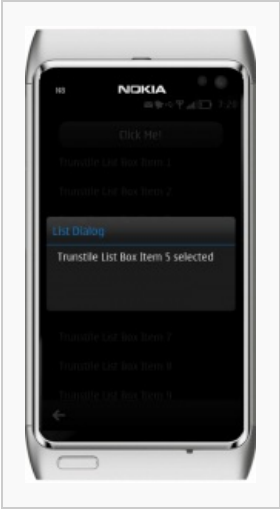
Turnstile ListView using Qt Quick

This article demonstrates how to create a Turnstile ListView item using Qt Quick.

Introduction

In this article we will see how to create a Turnstile ListView using QML Rotation Element. We will create a Button and when user clicks on the Button it will display a list of items in Turnstile manner. User can click on any of the list item to get the click event of that item using MouseArea.

Turnstile List View



List Item Clicked

Note: We tried using QML elements to bring an animation effect like Windows Phone Turnstile effect. The effect might not be exactly the same as that of Windows Phone Turnstile effect. Please try the example and share your feedback. To know more about Windows Phone Turnstile, please visit MSDN.

Implementation

Let's create an empty QML project. We will first create a ListModel with 9 items.

```
ListModel {
        id: modelListBox
        ListElement { rectState: ""; name: "Trunstile List Box Item 1";   }
        ListElement { rectState: ""; name: "Trunstile List Box Item 2"; }
        ListElement { rectState: ""; name: "Trunstile List Box Item 3";  }
        ListElement { rectState: ""; name: "Trunstile List Box Item 4";  }
        ListElement { rectState: ""; name: "Trunstile List Box Item 5";  }
        ListElement { rectState: ""; name: "Trunstile List Box Item 6"; }
        ListElement { rectState: ""; name: "Trunstile List Box Item 7"; }
        ListElement { rectState: ""; name: "Trunstile List Box Item 8";}
        ListElement { rectState: ""; name: "Trunstile List Box Item 9"; }
        }
```

We create a delegate which displays each ListModel items with a Rotation effect with an angle of *90* degree. To change the State smoothly we add the Transition element with a certain time duration.

```
...
                transform: Rotation
                {
                    id: rotationimageGreen
                    origin.x: 0;
```


(C) Copyright Nokia 2014. All rights reserved.

```
                    origin.y: 0;
                    origin.z: 0
                    axis { x: 1; y:-1 ; z: 0}
                    angle:-90
                }
                smooth: true
                states : State {
                    name: "right"
                    PropertyChanges { target: rotationimageGreen; angle : 0;
origin.x:0;   origin.y: 0; origin.z:0 }
                }
                transitions: Transition {
                      NumberAnimation { target: rotationimageGreen; property: "angle";
duration: 500 }
                }
...
```

To avoid displaying all the list items at the same time we set a Timer which expires after a certain time interval and displays the next item from the ListModel.

```
Timer {
                    id: changeStateTimer
                    interval: index*200; running: false; repeat: false
                    triggeredOnStart: false
                    onTriggered: { console.log(interval); rect.state = rect.stateKind}
                }
```

We use QML ListView to display the list box items

```
ListView
        {
            id: listView
            anchors.top: buttonClickMe.bottom
            anchors.bottom: parent.bottom
            anchors.left: parent.left
            anchors.right: parent.right
            model: modelListBox
            delegate:delegate
            clip:true
            height: 500
        }
```

When user clicks on any of the list item it opens up the CommonDialog with selected list item information.

```
CommonDialog{
            id:dialog
            titleText:  "List Dialog"
            onClickedOutside: close();
            content: Item {
                id: itemDialog
                Text {
                    id: textDialog
                    text:""
                    font.pixelSize: 21
```


(C) Copyright Nokia 2014. All rights reserved.

```
                    color: "white"
                    anchors.top: parent.top
                    anchors.topMargin: 10
                    anchors.left: parent.left
                    anchors.leftMargin: 20
                }
            }
        }
```

Summary

This article shows how to use the QML Rotation Element to create a Turnstile effect for ListView.

Source Code

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

(C) Copyright Nokia 2014. All rights reserved.