Logic Gates Example App using Meego Harmattan Qt Quick Components

This code example creates an app for testing the operation of logic gates. It demonstrates how you can use MeeGo Qt Quick Components to create attractive user interfaces.

09 Oct 2011

Introduction

The example uses following MeeGo Harmattan Qt Quick Components:

- QML PageStackWindow Element
- QML Page Element
- QML ListView Element
- QML QueryDialog Element
- QML Switch Element
- QML Label Element
- QML Menu Elements
- QML ToolBar Element
- QMl Image Element
- QML Column Element
- QML Flickable Element and others.

main.qml (QML PageStackWindow Element)

```
import QtQuick 1.1
import com.meego 1.0

PageStackWindow {
  id: appWindow

  showStatusBar: false //Hides the statusbar of device giving more area to application
body

  initialPage: mainPage
  MainPage{id: mainPage}

  //Change the Theme of application to dark. Light theme is selected by default

  Component.onCompleted: {
  theme.inverted = !theme.inverted
  }

  //ToolBar of the application showing only the View Menu.
  ToolBarLayout {
  id: commonTools
  visible: true
  ToolIcon { platformIconId: "toolbar-view-menu";
  anchors.right: parent===undefined ? undefined : parent.right
  onClicked: (myMenu.status == DialogStatus.Closed) ? myMenu.open() : myMenu.close()
  }
  }

  //QML Menu Elements, shown when View-Menu icon is clicked
  Menu {
  id: myMenu
  visualParent: pageStack
  MenuLayout {
```

 (C) Copyright Nokia 2014. All rights reserved.

```
  MenuItem { text: "About Logic Gates App"; onClicked: query.open() }
  }
  }

  //Application information screen usinf Query Dialog with icon and OK Button

  QueryDialog {
  id: query
  icon: "qrc:/lgates.png"
  titleText: "Logic Gates"
  message: "Version: 1.0.0"
  +"<br><br>Logic Gates Example App using Meego Harmattan Qt Quick Components.<br>"
  +"<br>&copy; 2011 Chintan K. Umraliya"
  +"<br>http://www.developer.nokia.com/"
  acceptButtonText: "OK"

  }
}
```

MainPage.qml(QML Page Elements)

Complete code can be found in the package file. Following are the main parts of the code:

mainPage: Locking Orientation to Portrait Mode

```
orientationLock: PageOrientation.LockPortrait
```

mainPage: Application Background

Using Tile mode for a small image for background

```
  Rectangle {
  id: background
  anchors.fill: parent
  color: "#343434"
  Image {
  source: "qrc:/stripes.png"
  fillMode: Image.Tile
  anchors.fill: parent
  opacity: 0.3
  }
  }
```

mainPage: ListView

The main application body is a QML ListView Element. Delegate and ListModel can be found in source file.

```
  ListView{
  id: gatesListView
  anchors.fill: parent
  clip: true
  model: pagesModel
  delegate: mydel
  header: topbar
  }
```

mainPage: Marquee Text for Listview header

 (C) Copyright Nokia 2014. All rights reserved.

```
  Component{
  id: topbar
  Rectangle {
  id: banner
  anchors.left: parent.left
  anchors.bottomMargin: 5
  clip: true
  width: parent.width
  height: 50
  color: "black"
  Text {
  id: title
  x: 96
  y: 11
  color: "white"
  text: "Logic Gates"
  horizontalAlignment: Text.AlignHCenter
  font.pixelSize: 23
  font.bold: true
  }
  Timer {
  interval: 100
  onTriggered: {
  if(title.x + title.width < 0)
  {
  title.x = title.parent.width;
  }
  title.x -= 8;
  }
  running: true
  repeat: true
  }
  }
  }
```

Animating Gate Thumbnails

Using NumberAnimation when the Gate Thumbnails appear on screen.

```
Component.onCompleted:NumberAnimation {
  target: gateThumbnail
  property: "scale"
  easing.type: Easing.InOutCirc
  from: 0.0; to: 1.0
  duration: 2250
 }
```

gateView: Displaying details of the selected Gate

We create second QML Page for showing Gate details and providing QML Switch Elements for simulating Gate behaviour.

```
Component {
  id: gateView
  Page {
  //Page toolbar showing only back option
  tools: ToolBarLayout {
  id: staticToolbar1
  ToolIcon { iconId: "toolbar-back"; onClicked: pageStack.pop(); }
```

 (C) Copyright Nokia 2014. All rights reserved.

```
}
//Lock Orientation to Portrait mode
orientationLock: PageOrientation.LockPortrait

//Page background
Rectangle {
id: background
anchors.fill: parent
color: "#343434"
Image {
source: "qrc:/stripes.png"
fillMode: Image.Tile
anchors.fill: parent
opacity: 0.3
}
}

//function to check status of the switches and set proper text for Output Label
function updatestatus(myVar) {
switch(myVar) {
case "AND GATE":
if((switchA.checked) && (switchB.checked) === true){
t1.text = "True"
}
else{t1.text = "False"}
break;
case "OR GATE":
if((switchA.checked) || (switchB.checked) === true){
t1.text = "True"
}
else{t1.text = "False"}
break;
case "NOT GATE":
if(!(switchA.checked)=== true){
t1.text = "True"
}
else{t1.text = "False"}
break;
case "NAND GATE":
if(!((switchA.checked) && (switchB.checked)) === true){
t1.text = "True"
}
else{t1.text = "False"}
break;
case "NOR GATE":
if(!((switchA.checked) || (switchB.checked)) === true){
t1.text = "True"
}
else{t1.text = "False"}
break;
case "EX-OR GATE":
if(switchA.checked != switchB.checked){
t1.text = "True"
}
else{t1.text = "False"}
break;
}
}

Flickable {
```

 (C) Copyright Nokia 2014. All rights reserved.

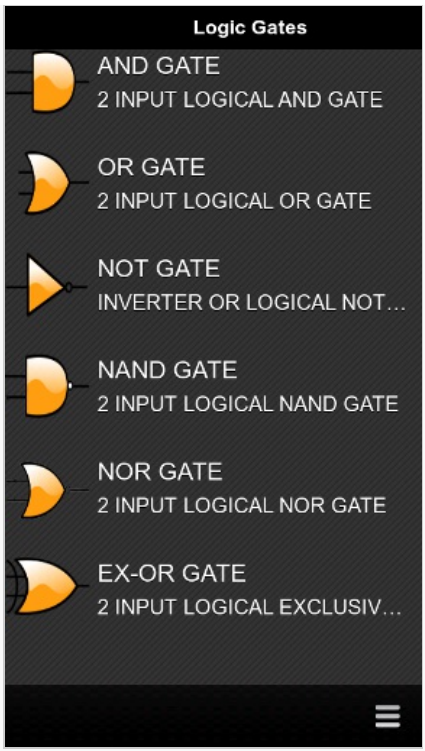
```
id: container
anchors.fill: parent
contentWidth: parent.width
contentHeight: parent.height
flickableDirection: Flickable.VerticalFlick
pressDelay: 100
//Using Animation for Gate Symbol
Component.onCompleted: NumberAnimation {
target: picture
property: "scale"
easing.type: Easing.InOutCirc
from: 0.0; to: 1.0
duration: 1750
}

Column {
id: inColumn
anchors.topMargin: 25
anchors.bottomMargin: 10
anchors.horizontalCenter: parent.horizontalCenter
width: parent.width
spacing: 15

Text {
id:gateNameText
text: myVar
anchors.bottomMargin: 15
anchors.horizontalCenter: parent.horizontalCenter
style: Text.Raised
font.pixelSize: 36
color: "white"
}

Image {
id: picture
anchors.horizontalCenter: parent.horizontalCenter
width: parent.width
fillMode: Image.PreserveAspectFit
source: "qrc:/"+myVar+"_b.png"
smooth: true
}
//Gate truth-table. You can also use inline-html table.
Image {
id: truthtable
anchors.horizontalCenter: parent.horizontalCenter
width: parent.width
fillMode: Image.PreserveAspectFit
source: "qrc:/"+myVar+"_tt.png"
}

Label {
id: description
width: parent.width
anchors.leftMargin: 10
text: destxt
}

Row {
id:rowA
anchors.horizontalCenter: parent.horizontalCenter
```

 (C) Copyright Nokia 2014. All rights reserved.

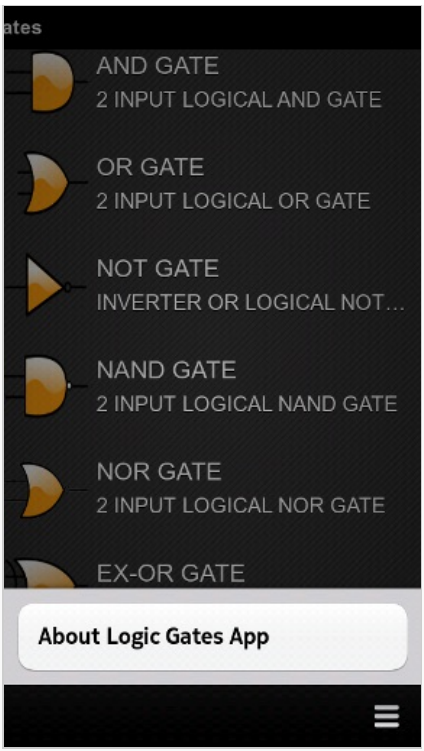
```
spacing: 50
Label {
text: "Input A"
}
Switch {
id: switchA
platformStyle: SwitchStyle {
inverted: true
}
onCheckedChanged: updatestatus(myVar)
}
}
Row {
id:rowB
anchors.horizontalCenter: parent.horizontalCenter
visible: myVar!="NOT GATE"
spacing: 50
Label {
text: "Input B"
}
Switch {
id: switchB
platformStyle: SwitchStyle {
inverted: true
}
onCheckedChanged: updatestatus(myVar)
}
}
Row {
id:rowOp
anchors.horizontalCenter: parent.horizontalCenter
spacing: 50
Label {
text: "Output:"
}
Label {
id: t1
text: "xxxx"
}
}
}
}
}
}
```

ScreenShots of the application running on Nokia N950 device

 (C) Copyright Nokia 2014. All rights reserved.

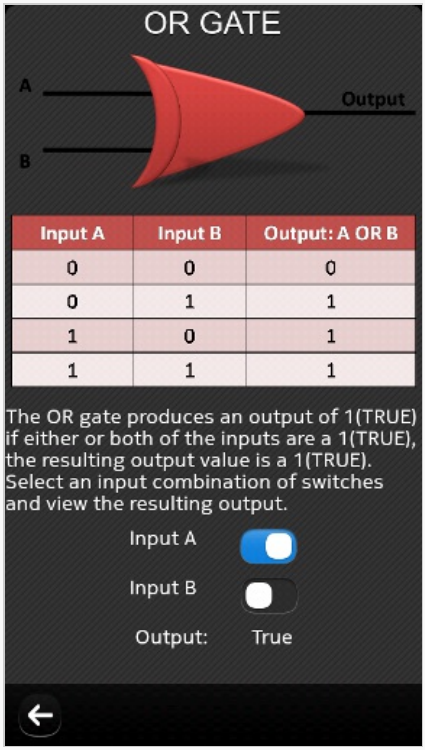


Application Homescreen

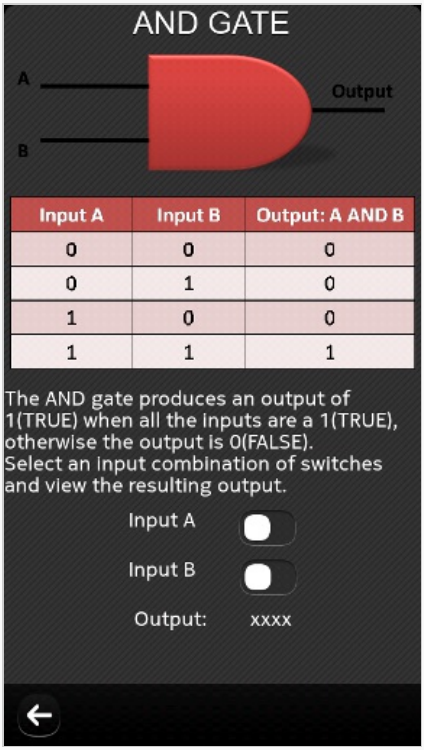


Menu Item

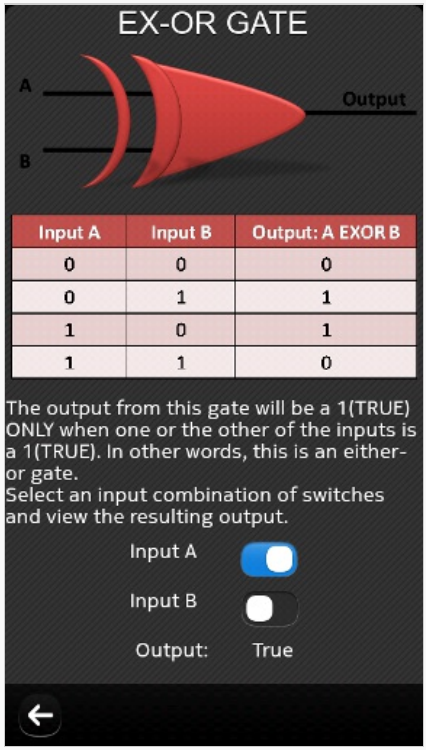
Application Information Screen



OR GATE View



AND GATE View



EX-OR GATE view

ck.umraliya 22:32, 26 September 2011 (EEST)

 (C) Copyright Nokia 2014. All rights reserved.

 (C) Copyright Nokia 2014. All rights reserved.