

Qt Developer Conference 2011





QtQuick 培训课程



第一部分



第一部分

目标

1 了解 Qt

什么是Qt？

为什么用Qt？

谁用Qt？

2 QtQuick 入门

Declarative UI

语法

示例

Qt Creator IDE

不同语言的比较

示例: Hello world



第一部分

目标

3 布局 和 交互

Item

Rectangle

Image

Text

Mouse Events

Key Events



第一部分

主题

1 了解 Qt

2 QtQuick 入门

3 布局 和 交互

4 问题解答

5 练习



了解Qt

什么是Qt?

跨平台的框架

可以用于 GUI 应用程序和 non-GUI 应用程序

应用广泛(桌面系统, 网络系统和嵌入式系统的开发)

2009 年起引入 LGPL

任何人都可以免费使用: <http://qt.nokia.com/>



了解 Qt

为什么选择 Qt?

Qt 是直观的

代码可以跨平台使用

可以根据需要选择不同的许可证书(Commercial, LGPL or GPL)

有巨大的开发者社区支持

为初学者提供免费的工具

有250.000 开发者使用 (包含商业授权和开源部分)

2011
Qt Developer Conference





了解 Qt

一些使用Qt 的应用程序

Autodesk software (Maya and 3d Studio)

Google Earth

Skype for Linux

Opera

全面支持Nokia的设备



第一部分

主题

1 了解 Qt

2 QtQuick 入门

3 布局和交互

4 问题解答

5 练习



QtQuick入门

Qt Quick 包含

QML (language)

QtDeclarative (native module, C++ API)

Qt Creator

```
Rectangle {  
    width: 200  
    height: 200  
    Text {  
        x: 66  
        y: 93  
        text: "Hello World"  
    }  
}
```

查看示例: <addon/module-001/examples/hello-world.qml>



QtQuick入门

工具包

Qt 4.7

<http://qt.nokia.com/downloads>

Qt Creator IDE

- 可视化的编辑和创建QML 组件 (Qt Quick Designer)
- 项目和构建管理工具包
- 完整的上下文相关的帮助系统
- 可视化的调试工具
- 快速的代码导航和浏览工具

2011
Qt Developer Conference





QtQuick入门

Qt Creator

The screenshot shows the Qt Creator IDE interface with several annotations:

- Mode Selector**: Points to the left sidebar containing icons for Welcome, Edit, Design, Debug, Projects, and Help.
- Run**: Points to the Run button (a green play icon) in the bottom toolbar.
- Build**: Points to the Build button (a green play icon with a hammer) in the bottom toolbar.
- Quick Search**: Points to the search bar at the bottom left with the placeholder text "Type to locate".
- Symbol Overview**: Points to the "Symbol Overview" button in the top right corner.
- Output Panes**: Points to the bottom right area containing tabs for "Build Issues", "Search Results", "Application Output", and "Compile Output".

The main editor displays the `main.qml` file with the following QML code:

```
1 import Qt 4.7
2
3 Item {
4     id: root
5     width: 640
6     height: 360
7
8     property string folder: "images/s60/";
9     property real scaleFactor: root.height/480.0;
10    property int dayWidth: 75;
11    property int dayHeight: 71;
12    property real centerFactor: (root.width - ((dayWidth * scaleFactor)*7))/2;
13
14    Script {
15        source: "basiccalendar.js"
16    }
17
18    Pager {
19        id: pager
20        anchors.fill: parent
21        onDayClicked: {
22            menuSelection.xPos = rx - 6;
23            menuSelection.yPos = ry + 40;
24            menuSelection.model = rmodel;
25            menuSelection.index = index;
26            menuSelection.state = "openMenu";
27        }
28    }
29
30    Image {
31        id: multitask
32        anchors.top: parent.top
33        anchors.left: parent.left
34        source: folder + "icon_multitask.png"
35    }
36
37    Image {
38        id: arrowLeft
39        anchors.left: parent.left
```

Qt Developer Conference





不同语言的比较

Actionscript: **MenuButton.as**

```
public class MenuButton extends MovieClip {  
    public function MenuButton() {  
        this.x = 60;  
        this.addEventListener(MouseEvent.CLICK,  
ClickBt);  
    }  
    function ClickBt(e:MouseEvent) {  
        trace("clicked");  
    }  
}
```

QtQuick: **MenuButton.qml**

```
Item {  
    x:60;  
    MouseArea: {  
        anchors.fill: parent;  
        onClicked: print("clicked");  
    }  
}
```



第一部分

主题

1 了解 Qt

2 QtQuick 入门

3 布局和交互

4 问题解答

5 练习



布局和交互

布局

Item

Rectangle

Image

Text and TextInput



布局和交互

Item

是QtQuick中所有可视化元素的基础

Item 没有可视化的外观

它定义的所有可视化元素的各种属性

常见的属性如: anchors, clip, width, height, opacity, x, y, rotation or scale



Item 相关代码片段

QtQuick中所有可视化元素的基础

```
Item {  
    id: label1  
    x: 80  
    width: 100  
    height: 100  
}
```

Anchors

```
Item {  
    id: label2  
    anchors.left: label1.left  
    anchors.top: label1.top  
    anchors.topMargin: 30  
}
```

使用Anchor 属性代替确定的 x 和 y坐标. 在第二部分我们会详细介绍anchors元素



布局和交互

布局

Item

Rectangle

Image

Text and TextInput



Rectangle

是用确定的颜色绘出的有(无)边框的矩形

```
Item {  
    id: label1  
    width: 100  
    height: 100  
  
    Rectangle {  
        anchors.fill: parent  
        color: "red"  
        border.color: "black"  
        border.width: 5  
        radius: 10  
    }  
}
```

可以使用radius 属性创建圆角矩形.

查看示例: <addon/module-001/examples/item-example.qml>



布局与交互

布局

Item

Rectangle

Image

Text and TextInput



Image

该元素可以在场景中添加位图.

```
Item {  
    id: label1  
    width: 100  
    height: 100  
  
    Image {  
        id: button  
        source: "pngs/img.png"  
    }  
}
```

在这里不使用确定的image元素的width和height数据. QtQuick 会自动调整.

了解更多支持的格式参见:

<http://doc.qt.nokia.com/4.7-snapshot/qml-image.html>

查看示例: `addon/module-001/examples/image-example.qml`



布局 and 交互

布局

Item

Rectangle

Image

Text and TextInput



Text and TextInput

该元素可以在场景中添加格式化的文本.

```
Text {  
    id: text1  
    text: "Hello World!"  
    font.family: "Helvetica"  
    font.pixelSize: 18  
    color: "red"  
}
```

```
TextInput {  
    id: input  
    color: "red"  
    text: "Default Text"  
    width: 200; height: 24  
    focus: true  
}
```

了解更多的文本属性参见:

<http://doc.qt.nokia.com/4.7-snapshot/qml-text.html>



布局和交互

交互

MouseArea

FocusScope

Flickable



MouseArea

该元素处理场景中items的鼠标事件.

```
Rectangle {  
    width: 100; height: 100  
    color: "green"  
  
    MouseArea {  
        anchors.fill: parent  
        onClicked: { parent.color = 'red' }  
    }  
}
```



MouseArea

在规定的区域拖动一个Item可以创建一个滚动条.

```
Rectangle {  
    id: slider;  
    width: 320; height: 40;  
    color: "green"  
    Rectangle {  
        id: handle; width: 40; height: 40  
        color: "red"  
        MouseArea {  
            anchors.fill: parent  
            drag.target: parent; drag.axis: "XAxis"  
            drag.minimumX: 0  
            drag.maximumX: slider.width - handle.width  
        }  
    }  
}
```



布局和交互

MouseArea

信号:

onCanceled

onClicked

onDoubleClicked

onEntered

onExited

onPositionChanged

onPressAndHold

onPressed

onReleased

了解更多MouseArea 的信号参见:

<http://doc.qt.nokia.com/4.7-snapshot/qml-mousearea.html>



布局和交互

交互

MouseArea

FocusScope

Flickable



Key focus

当一个键被按下，会产生一个按键事件。

```
Rectangle {  
    color: "lightsteelblue"; width: 240; height: 25  
    Text { id: myText }  
    Item {  
        id: keyHandler  
        focus: true  
        Keys.onPressed: {  
            if (event.key == Qt.Key_A)  
                myText.text = 'Key A was pressed'  
            else if (event.key == Qt.Key_B)  
                myText.text = 'Key B was pressed'  
            else if (event.key == Qt.Key_C)  
                myText.text = 'Key C was pressed'  
        }  
    }  
}
```



Key navigation

在以键盘操作为主的界面上，使用方向键导航到不同item是很常见的。

```
Grid {
    columns: 2
    width: 100; height: 100
    Rectangle {
        id: item1
        focus: true
        width: 50; height: 50
        color: focus ? "red" : "lightgray"
        KeyNavigation.right: item2
    }
    Rectangle {
        id: item2
        width: 50; height: 50
        color: focus ? "red" : "lightgray"
        KeyNavigation.left: item1
    }
}
```

了解更多的关于keyboard focus:

<http://doc.qt.nokia.com/4.7-snapshot/qdeclarativefocus.html>

查看示例: [addon/module-001/examples/key-navigation.qml](#)



布局和交互

交互

MouseArea

FocusScope

Flickable



Flickable

所有放在Flickable 元素中的item, 可以拖动和闪烁.

```
Flickable {  
    width: 200; height: 200  
    contentWidth: image.width  
    contentHeight: image.height  
  
    Image { id: image; source: "bigImage.png" }  
}
```

关于如何调整到满足需要参见:

<http://doc.qt.nokia.com/4.7-snapshot/qml-flickable.html>

查看示例: `addon/module-001/examples/flickable-area.qml`



第一部分

主题

1 了解Qt

2 QtQuick入门

3 布局和交互

4 问题解答

5 练习



问题

什么是Qt 和 QtQuick?

QtQuick 组件文件是怎么组织的?

QtQuick中可视化元素的基础是什么?

如何固定一个item相对于另一个item的位置?

QtQuick 中如何捕获鼠标事件?

列出一些继承自Item的元素.

在一个QtQuick文件中, 能声明多少个组件?



第一部分

主题

1 了解 Qt

2 QtQuick入门

3 布局和交互

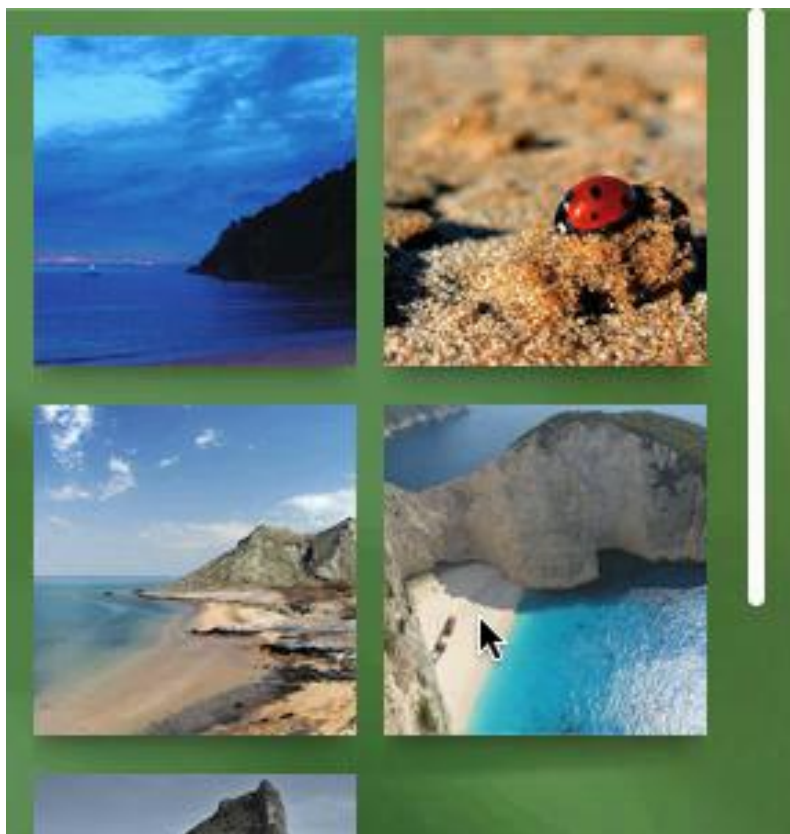
4 问题解答

5 练习



练习

按照下面:使用 anchors 排列items 到一个Flickable区域



查看视频: <addon/module-001/videos/lab-video-example.mov>

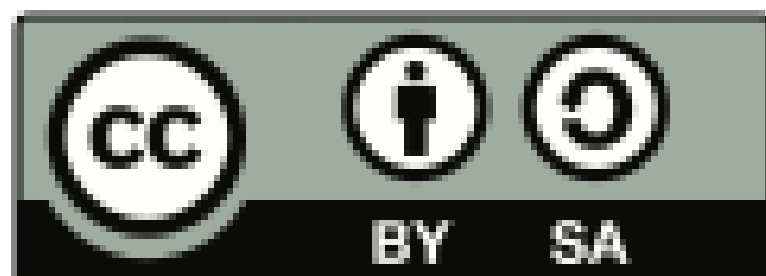
选作部分: 为该区域创建一个滚动条

查看练习: <addon/module-001/labs/lab-flickable/LabOne.qmlproject>



版权 © (2011)属于Nokia公司及其附属公司，Nokia公司及其附属公司保留全部权利。

包含附带的Qt 培训材料都是根据Creative Commons Attribution ShareAlike 2.5 License Agreement发布.



完整的证书参见:<http://creativecommons.org/licenses/by-sa/2.5/legalcode>

Nokia, Qt 和 所有基于 Nokia 和 Qt 的标志是Nokia公司在芬兰及全球其他国家的商标或注册商标.

