

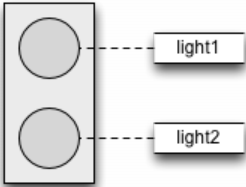
- 第1209篇 • 图腾
- 评论 |
- 肖鹰：柴静的意义——她为何值得
- 第1207篇 • 日货
- 北美崔哥：美国正称霸世界，春晚
- 某些国家为何拿高铁来戏弄中国？
- 谁才是莫斯科刺杀事件的最大受益
- 第1202篇 • 冤死
- 北美崔哥：中国式上访，已正式输
- 蝗虫之日的启示



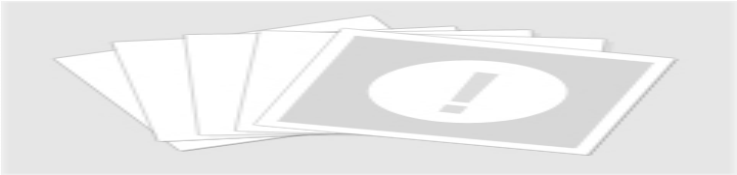
查看更多>>

谁看过这篇博文

- | | |
|------------|--------|
| moonlqer | 3月2日 |
| 柴小希要... | 2月28日 |
| 右右youyou | 2月24日 |
| 鑫语馨缘 | 1月30日 |
| yony | 1月21日 |
| 乖乖醉长安 | 1月7日 |
| 360度转身 | 1月4日 |
| 期望 | 12月16日 |
| 用户21328... | 12月1日 |
| 贰喜 | 12月1日 |
| 宋海凡 | 11月26日 |
| guoke312 | 11月18日 |



例如一个交通信号灯可能有两个信号灯。上面的一个信号灯使用红色，下面的信号灯使用绿色。这个例子中，两个信号灯不会同时闪烁。让我们看看状态图。



当系统启动时，它会自动切换到停止模式作为默认状态。停止状态改变了`light1`为红色并且`light2`为黑色（关闭）。一个外部的事件能够触发一个状态切换为`go`状态。在`go`状态下，我们改变颜色属性，`light1`变为黑色（关闭），`light2`变为绿色。

为了实现这个方案，我们绘制一个用户界面的两个灯，为了简单起见，我们使用两个包含圆边的矩形框，设置圆半径为宽度的一半（宽度与高度相同）。

```
Rectangle {
  id: light1
  x: 25; y: 15
  width: 100; height: width
  radius: width/2
  color: "black"
}

Rectangle {
  id: light2
  x: 25; y: 135
  width: 100; height: width
  radius: width/2
  color: "black"
}
```

就像在状态图中定义的一样，我们有一个`go`状态和一个`stop`状态，它们将会分别将交通灯改变为红色和绿色。我们设置`state`属性到`stop`来确保初始化状态为`stop`状态。

注

我们可以只使用`go`状态来达到同样的效果，没有显式的使用`stop`状态，设置颜色`light1`为红色，颜色`light2`为黑色。初始化状态`""`（空字符串）定义初始化属性，并且扮演类似`stop`状态的角色。

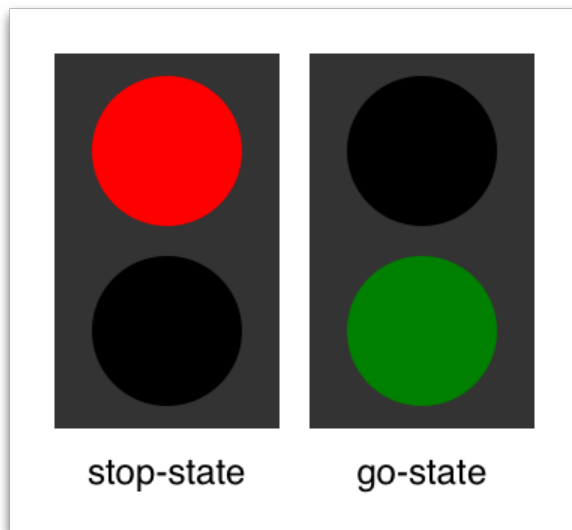
```
state: "stop"

states: [
  State {
    name: "stop"
    PropertyChanges { target: light1; color: "red" }
    PropertyChanges { target: light2; color: "black" }
  },
  State {
    name: "go"
    PropertyChanges { target: light1; color: "black" }
    PropertyChanges { target: light2; color: "green" }
  }
]
```

在这个例子中使用`PropertyChanges { target: light2; color: "black" }`不是必要的，因为`light2`初始化颜色已经是黑色了。在一个状态中，只需要描述属性如何从它们的默认状态改变（而不是前一个状态的改变）。

使用鼠标区域覆盖整个交通灯，并且绑定在点击时切换`go`和`stop`状态。

```
MouseArea {
  anchors.fill: parent
  onClicked: parent.state == "stop"? "go" : "stop"
}
```



我们现在已经成功的实现了交通灯的状态切换。为了让用户界面看起来更自然，需要使用动画效果来增加一些过渡。一个过渡能够被状态的改变触发。

注

可以使用一个简单逻辑的脚本来替换QML状态。开发人员很容易陷入这种陷阱，写的代码更像一个JavaScript而不是一个QML程序。

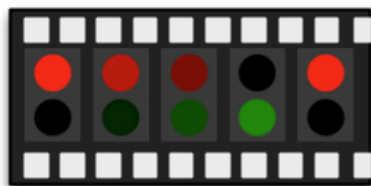
5.2.2 过渡

一系列的过渡能够被加入任何元素，一个过渡由状态的改变触发执行。你可以使用属性的`from:`和`to:`属性来定义状态改变的指定过渡。这两个属性就像一个过滤器，当过滤器为`true`时，过渡生效。也可以使用`""`表示任何状态。例如`from:"*"; to:"*"`表示从任一状态到任何其它状态的`from`和`to`的默认值，这意味着过渡用于每个状态的切换。

在这个例子中，我们期望从状态“go”到“stop”转换时实现一个颜色改变的动画。对于从“stop”到“go”状态的改变，我们期望保持颜色的直接改变，不使用过渡。我们使用`from`和`to`来限制过渡只从“go”到“stop”时生效。在过渡中我给每个灯添加两个颜色的动画，这个动画将按照状态的描述来改变属性。

```
transitions: [
  Transition {
    from: "stop"; to: "go"
    ColorAnimation { target: light1; properties: "color"; duration: 2000 }
    ColorAnimation { target: light2; properties: "color"; duration: 2000 }
  }
]
```

可以点击用户界面来改变状态。尝试点击用户界面，当状态从“stop”到“go”时，你将会看到状态会立刻发生改变。



接下来，你可以修改这个示例，例如缩小未点亮的灯来突出点亮的灯。为此，需要在状态中添加一个属性用来缩放，并且操作一个动画来播放缩放属性的过渡。另一个选择是添加一个“attention”状态，灯会出现黄色闪烁，为此你需要添加为这个过渡添加一个一秒连续的动画来显示黄色（使用“to”属性来实现，一秒后变为黑色）。甚至可以改变缓冲曲线来使这个例子更加生动。

5.2. 高级技巧

//todo

注：

技术在于交流、沟通，转载请注明出处并保持作品的完整性。

作者：`☆奋斗ing♥孩子` 原文：http://blog.sina.com.cn/s/blog_a6fb6cc90102v1x8.html。

1

👍喜欢

分享：         

阅读(128) | 评论 (0) | 收藏(0) | 已有1人转载▼ | 喜欢▼ | 打印 已投稿到： 排行榜

前一篇：流元素
后一篇：QML之鼠标样式

评论









重要提示：警惕虚假中奖信息

[发评论]

做第一个评论者吧！ 抢沙发>>

发评论

一去、二三里：

☐  分享到微博  ☐ 匿名评论

验证码： [请点击后输入验证码](#) [收听验证码](#)

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

< 前一篇
流元素

后一篇 >
QML之鼠标样式