QML布局管理

目录

[概述 1](#_Toc1246197587)

[水平布局 1](#_Toc1147696068)

[垂直布局 2](#_Toc458511598)

[网格布局 3](#_Toc30726057)

[混合使用 4](#_Toc1627470713)

[其他 5](#_Toc381524526)

## 概述

使用Qt做过UI的一定对QHBoxLayout, QVBoxLayout, 和QGridLayout这三个最重要也最常使用的layout managers非常熟悉。那么在QML中又是如何控制和管理UI布局的呢？那么我们这篇文章就为大家介绍这些基础知识。

首先，QML同样允许大家使用硬编码的方式将位置数值直接写到代码中，但是这样做首先难以适应UI的调整，其次代码维护起来也很困难。因此我们不推荐这样做。推荐大家使用的是以下三种布局管理器：Row,、Column、Grid，以及使用Anchor进行布局。

## 水平布局

qml中使用Row{}进行水平布局，会将其子控件都排列在同一行，相互不重叠。我们还可以使用它的spacing 属性来定义子控件之间的距离。



图中将两个Text进行了水平布局，布局间隔为50。

## 垂直布局

qml中使用Column{}进行垂直布局，会将子控件都排列在同一列，相互不重叠。我们还可以使用它的spacing 属性来定义子控件之间的距离。



图释，将三个Text进行了垂直布局。

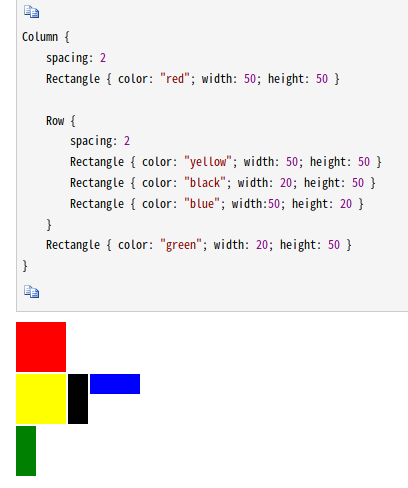
## 网格布局

qml中使用Grid{}进行网格布局，网格布局时需要给出行数和列数。子控件都均匀地排列在一个网格内，相互不重叠，每一个子控件都被放置在一个网格单元的（0，0）位置，也就是左上角。Grid的rows 和columns属性定义网格的行数和列数，列数默认是4。我们还可以使用Grid的spacing 属性来定义网格单元之间的距离，这里注意水平和垂直方向的spacing都是一样的。



## 混合使用

我们还可以将Grid、Row 和 Column 进行混合应用。



## 其他

除了上述几项布局方式外，还有类似于QtGUI框架的RowLayout、ColumnLayout、GridLayout使用方式类似，不做描述。布局中不可不说的一种模式为anchors(锚点布局)模式，请查看“qml-anchors(锚点布局)说明”文档。