移动平台应用软件开发

界面布局

主讲：张齐勋

zhangqx@ss.pku.edu.cn

《移动平台应用软件开发》课程建设小组

北京大学

二零一八年

北京大学

提纲

- 布局
- **LinearLayout**
- **RelativeLayout**
- **FrameLayout**
- **TableLayout**
- **AbsoluteLayout**

布局

- 布局是一种可用于放置很多控件的容器。
- 多层布局的嵌套：布局的内部除了可以放置控件外，还可以放置布局。

LinearLayout

- LinearLayout(线性布局)是一种常用的布局。
- 它所包含的控件在线性方向上依次排列。
- 通过android:orientation属性指定排列方向
  - vertical（垂直方向）
  - horizontal（水平方向,缺省）

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent"
    android:orientation="horizontal">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="点我"
        android:id="@+id/button" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button2" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button3" />
</LinearLayout>
```

android:layout_gravity

- 指定控件在布局中的对齐方式。
- gravity VS. layout_gravity
- 值包含：top、bottom、center_vertical等。
- 该属性的效果取决于布局方式，如
  - 线性布局
  - 排列方向为horizontal（水平）
  - 因此，只有垂直方向的值是有效的

android:layout_weight

- 通过比例的方式来指定控件的大小。
- 在界面适配时，作用很大。

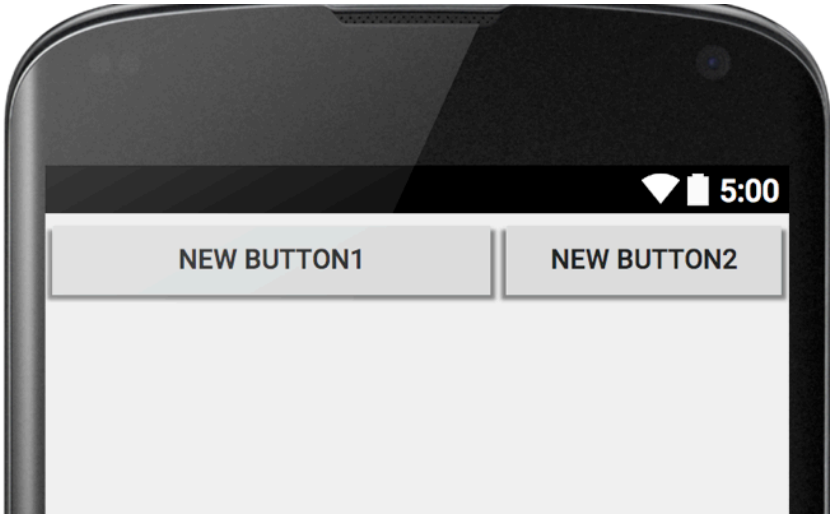
```
<Button
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="New Button1"
    android:id="@+id/button2"
    android:layout_weight="1" />

<Button
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="New Button2"
    android:layout_weight="2"
    android:id="@+id/button" />
```

屏幕适配的例子

```
<Button
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="New Button1"
    android:id="@+id/button2"
    android:layout_weight="1" />

<Button
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:text="New Button2"
    android:id="@+id/button" />
```



- 大家考虑一下，这样有什么好处？

RelativeLayout

- 相对布局通过相对位置的方式使控件出现在布局的任意位置。
- 可通过相对于父布局定位，也可以通过相对于控件进行定位。
- 属性比较多，但是非常简单，很容易掌握

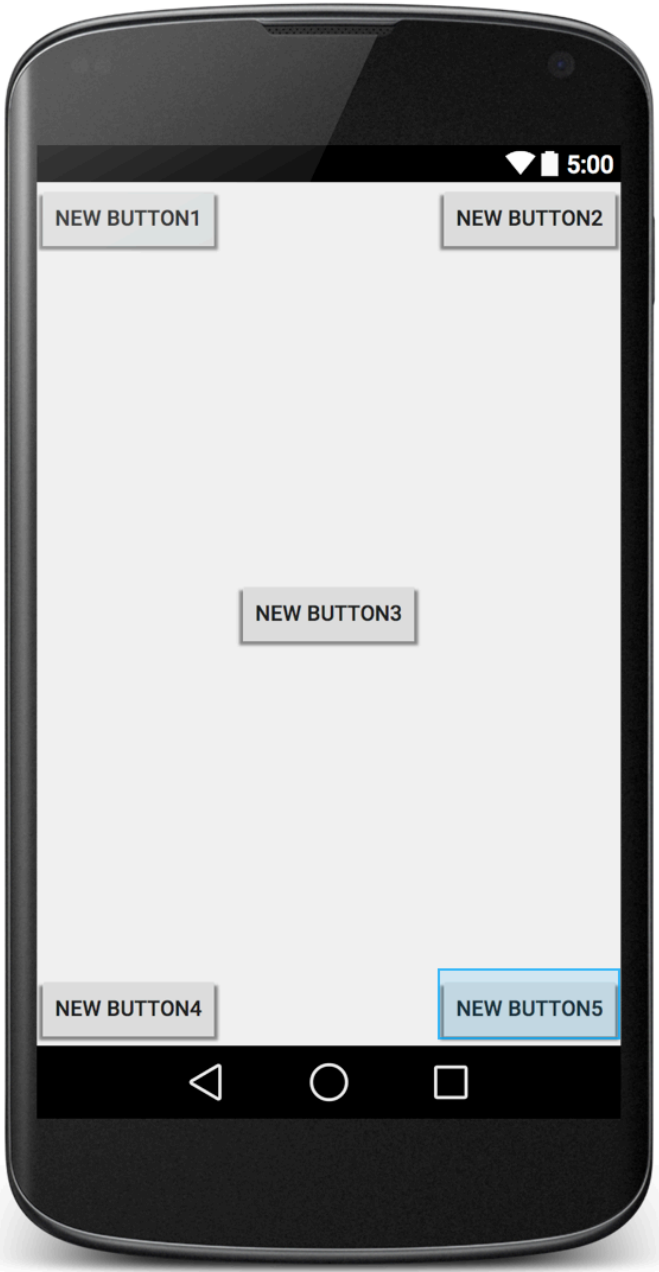
```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button1"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:id="@+id/button1" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button2"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:id="@+id/button2" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button3"
    android:id="@+id/button3"
    android:layout_centerInParent="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button4"
    android:id="@+id/button4"
    android:layout_alignParentLeft="true"
    android:layout_alignParentBottom="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button5"
    android:id="@+id/button5"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true" />
```



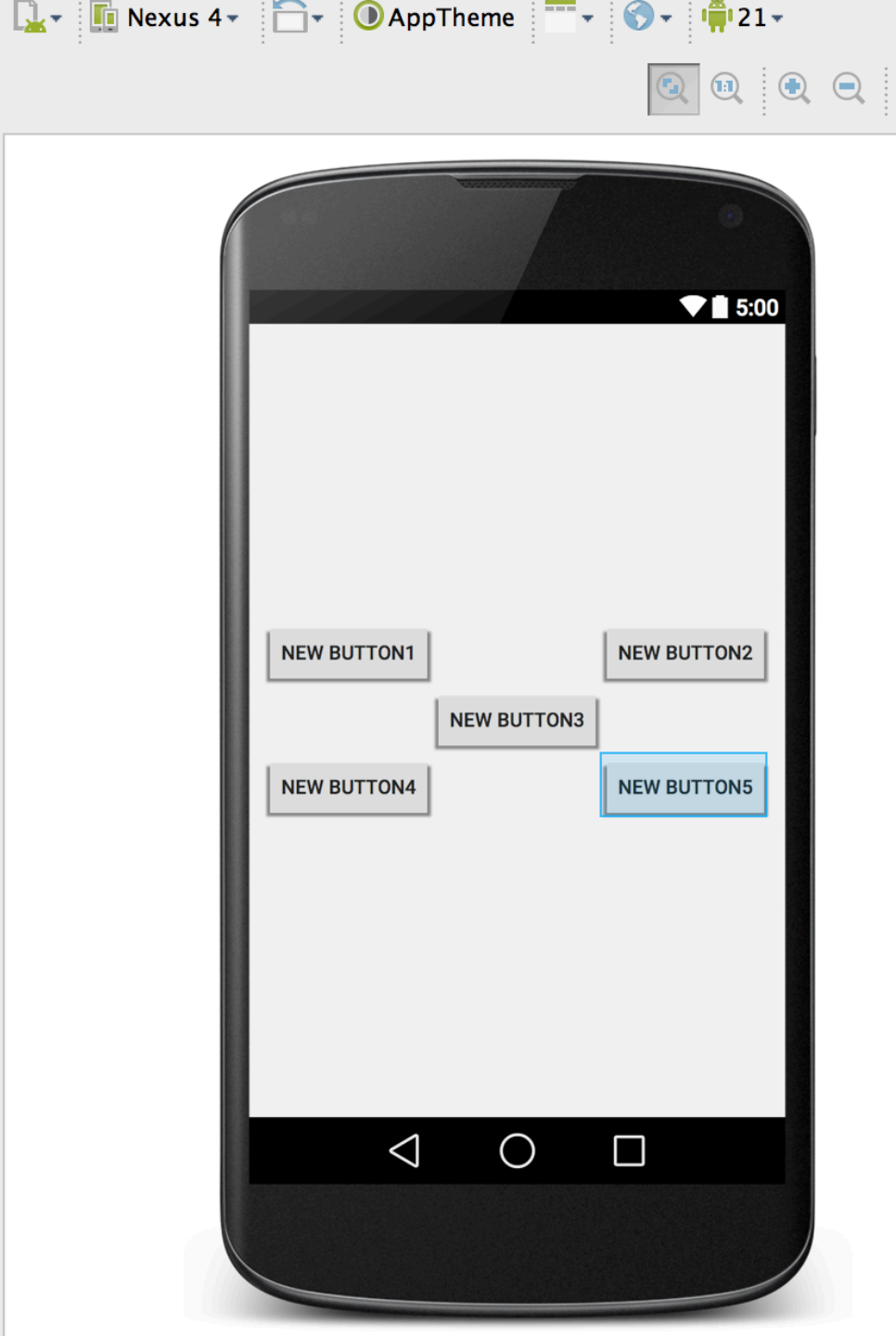
```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button1"
    android:layout_above="@id/button3"
    android:layout_toLeftOf="@id/button3"
    android:id="@+id/button1" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button2"
    android:layout_above="@id/button3"
    android:layout_toRightOf="@id/button3"
    android:id="@+id/button2" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button3"
    android:id="@+id/button3"
    android:layout_centerInParent="true"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button4"
    android:id="@+id/button4"
    android:layout_below="@id/button3"
    android:layout_toLeftOf="@id/button3"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button5"
    android:id="@+id/button5"
    android:layout_below="@id/button3"
    android:layout_toRightOf="@id/button3"/>
```



FrameLayout

- 所有的元素都被放置在FrameLayout区域的最左上的区域，而且无法为这些元素指定一个确切的位置。
- 如果一个FrameLayout里面有多个元素，那么后面的子元素的显示会重叠在前一个元素上。

TableLayout

- 把每个控件以行和列的形式进行排列
- 每行为一个TableRow对象
- 在TableRow中还可以继续添加其他的控件，每添加一个子控件就成为一列。

AbsoluteLayout

- 绝对布局又叫坐标布局，可以直接指定子元素的绝对位置layout_x、layout_y
- 这种布局简单直接，但由于手机屏幕尺寸差别比较大，使用绝对定位的适应性较差。
- 在绝对定位中，如果子元素不设置layout_x和layout_y，那么它们的默认值是0。

Q&A

本讲结束 ！

北京大学