# 实验二：UI设计

## 实验目的

本次实验的目的是让大家熟悉Android开发中的UI设计，包括了解和熟悉常用控件的使用、界面布局和事件处理等内容。

## 实验要求

1. 熟悉和掌握界面控件设计
2. 了解Android界面布局
3. 掌握控件的事件处理

## 实验内容

### 一、常用控件

#### 1.常用控件介绍

Android中有许多常用控件（简单分类）：

文本框：TextView、EditText

按钮：Button、RadioButton、RadioGroup、CheckBox、ImageButton

列表：List、ExpandableListView、Spinner、AutoCompleteTextView、GridView、ImageView

进度条：ProgressBar、ProgressDialog、SeekBar、RatingBar

选择器：DatePicker、TimePicker

菜单：Menu、ContentMenu

对话框：Dialog、ProgressDialog

常用的控件有文本框、按钮和列表等。

#### 2.控件的实现

TextView测试：点击TEXTVIEW测试按钮，通过按钮的监听事件监听，然后上面的TextView会更改文本内容为“测试TextView”。



Button测试：点击BUTTON测试按钮，通过按钮的监听事件监听，然后上面的TextView会更改文本内容为“测试Button”。



ToggleButton测试：点击“打开”，其状态会改为“关闭”，再次点击，其状态会再次改为“打开”。



RadioButton测试：点击“女”，通过按钮的监听事件，上面的TextView会更改文本内容为“你的性别是：女”；点击“男”，通过按钮的监听事件，上面的TextView会更改文本内容为“你的性别是：男”。



Spinner测试：下拉选择框选择城市，通过事件监听改变上面的TextView文本内容为对应的城市。



#### 3.控件的属性

可以直接在类中为控件设置属性，也可以在xml文件中设置属性。每个控件都有一系列的属性，例如id、size、text、color等等，大家可以找找API查看控件的属性。

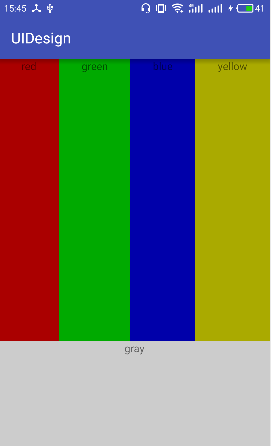
### 二、界面布局

Android中有很多布局：

* FrameLayout：最简单的一个布局对象。它里面只显示一个显示对象。Android屏幕元素中所有的显示对象都将会固定在屏幕的左上角，不能指定位置。但允许有多个显示对象，但后一个将会直接在前一个之上进行覆盖显示，把前一个部份或全部挡住（除非后一个是透明的）。
* LinearLayout：以单一方向对其中的显示对象进行排列显示，如以垂直排列显示，则布局管理器中将只有一列；如以水平排列显示，则布局管理器中将只有一行。同时，它还可以对个别的显示对象设置显示比例。
* TableLayout：以拥有任意行列的表格对显示对象进行布局，每个显示对象被分配到各自的单元格之中，但单元格的边框线不可见。
* AbsoluteLayout：允许以坐标的方式，指定显示对象的具体位置，左上角的坐标为(0, 0)，向下及向右，坐标值变大。这种布局管理器由于显示对象的位置定死了，所以在不同的设备上，有可能会出现最终的显示效果不一致。
* RelativeLayout：允许通过指定显示对象相对于其它显示对象或父级对象的相对位置来布局。如一个按钮可以放于另一个按钮的右边，或者可以放在布局管理器的中央。

布局中可以放置控件，也可以嵌套其他布局，这和java、c#都非常相似。

如图所示：一个LinearLayout中包括了两个LinearLayout，每个LinearLayout下都有TextView控件。



### 三、事情处理

事件监听和时间处理的概念大家以前都学习过，android中主要有如下事件方法：

* onClick(View v) 一个普通的点击按钮事件
* boolean onKeyMultiple(int keyCode,int repeatCount,KeyEvent event)用于在多个事件连续时发生，用于按键重复
* boolean onKeyDown(int keyCode,KeyEvent event) 用于在按键进行按下时发生
* boolean onKeyUp(int keyCode,KeyEvent event） 用于在按键进行释放时发生
* onTouchEvent(MotionEvent event)触摸屏事件，当在触摸屏上有动作时发生
* boolean onKeyLongPress(int keyCode, KeyEvent event)当长时间按时发生

下面以点击一个Button为例子：

**public** **class** HelloWorld **extends** Activity {

/\*\* Called when the activity is first created. \*/

@Override

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*main*);

Button button = (Button)findViewById(R.id.*btn*);

button.setOnClickListener(buttonListener);

}

**private** OnClickListener buttonListener = **new** OnClickListener(){

**public** **void** onClick(View v) {

Toast.*makeText*(v.getContext(), "You have clicked a button!",Toast.*LENGTH\_LONG*).show();

}

};

}

为button添加了一个监听buttonListener，对应的onClick方法则抛出了一则文本框提示用户。

这是其中一种Button点击事件监听方式，另一种比较简单的方式是利用xml自己配置button的监听方法。例子如下所示：

activity\_main.xml中的部分代码：

<Button  
 android:text="测试布局"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/btnSP"  
 android:layout\_marginTop="72dp"  
 android:id="@+id/btnLayout"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentEnd="true"  
 android:onClick="btnLayout\_Click" />

btnLayout\_Click方法代码：

public void btnLayout\_Click(View view)  
{  
 Intent intent = new Intent(MainActivity.this,LayoutActivity.class);  
 startActivity(intent);  
}