

Android开发入门

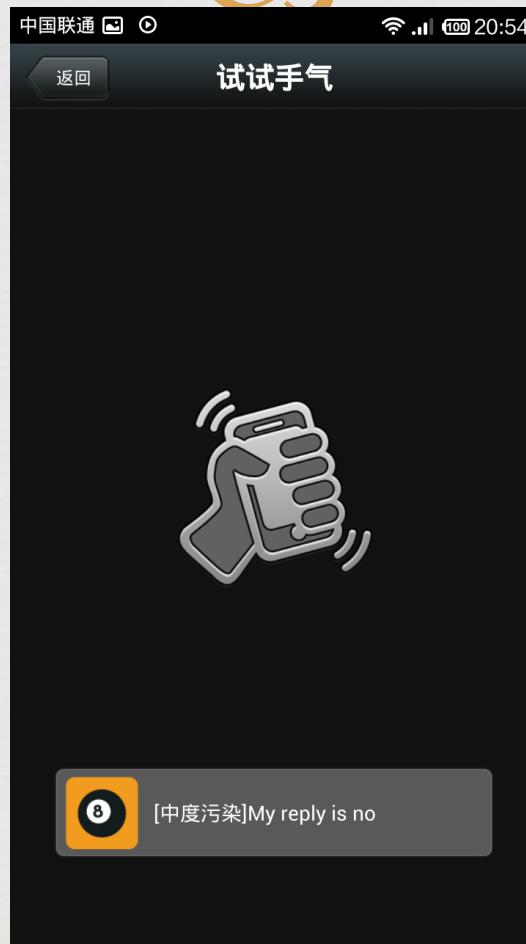


胡家威

计研135班

<http://hujiaweibujidao.github.io/>

小应用： Magic8



内容概要(上)



- ❖ Android系统简介
- ❖ Android应用结构
- ❖ Android四大组件
- ❖ Activity生命周期
- ❖ Android资源管理
- ❖ UI组件和容器组件

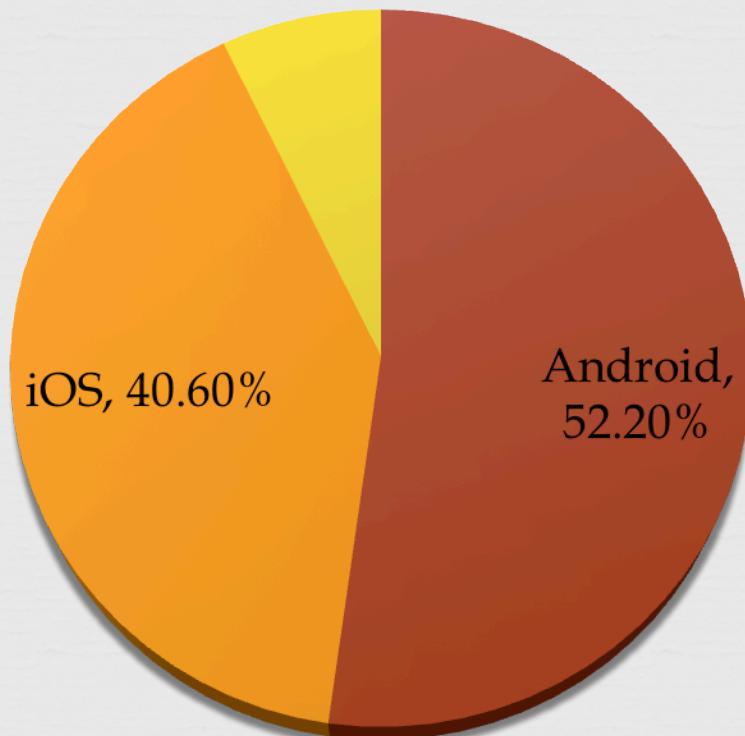
Android系统简介



Android市场份额



其他, 7.20%



智能手机操作系统在美国的市场份额(截至2013年10月31日)

Android是如何诞生的?



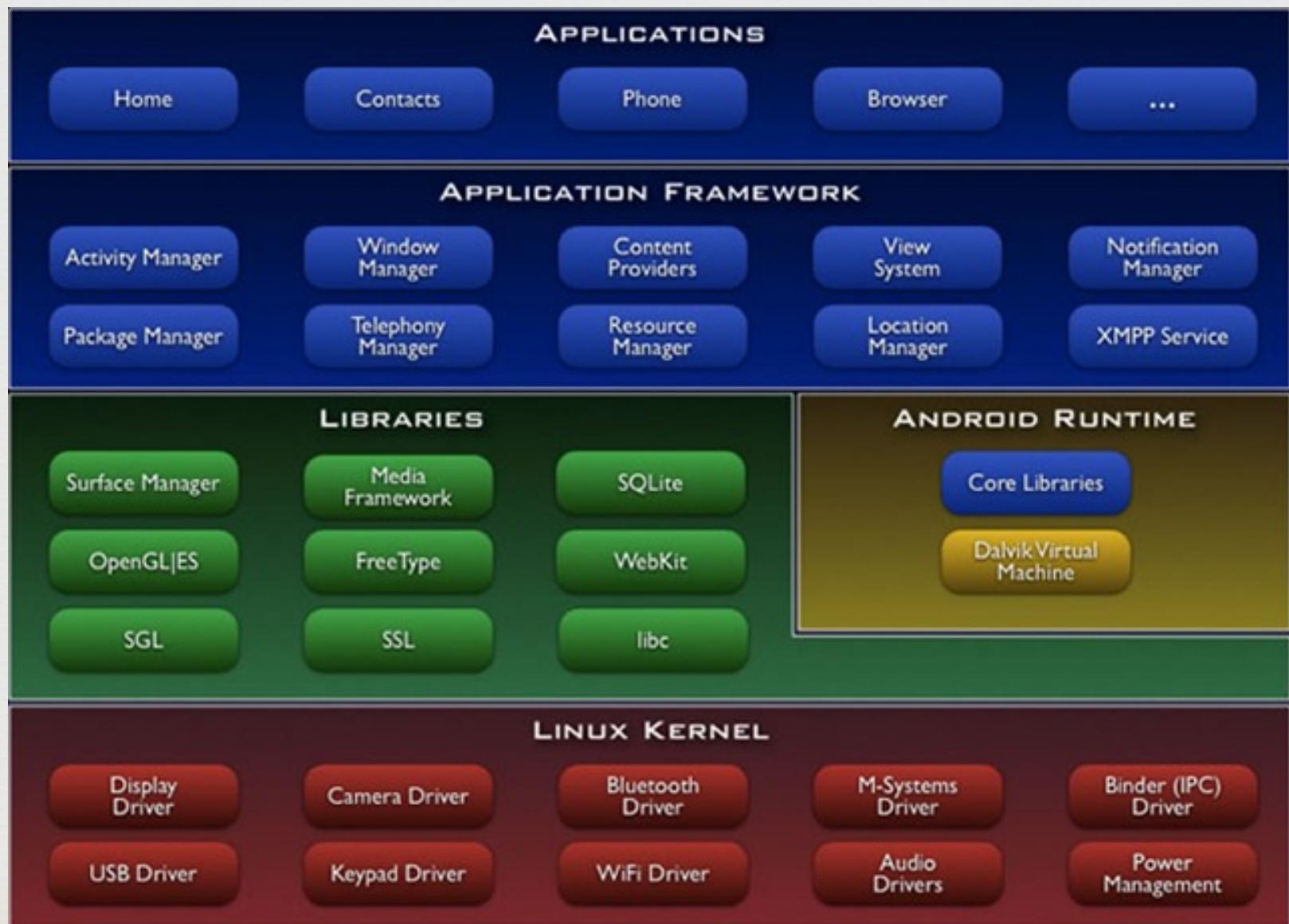
- ❖ Android之父： Andy Rubin
- ❖ 现就职于Google， 曾一直主管Android项目， 现在正在进行一项秘密的机器人项目。



那些年的Android系统



Android 系统架构



Android运行时



- ❖ JRE(Java Runtime Environment)包括JVM(Java Virtual Machine)和其他功能函数库。
- ❖ Android Runtime Environment包括Dalvik 虚拟机和核心库集。
- ❖ Dalvik虚拟机和JVM不同， 它并没有遵守JVM规范， JVM运行的是Java字节码(.class)文件， 而Dalvik虚拟机运行的是其专有的Dalvik Executable (.dex)文件。
- ❖ 每个Android应用程序都运行在单独的Dalvik虚拟机中， 以便实现对应用程序的隔离。 Dalvik虚拟机针对移动平台进行了很多优化， 所以性能很好。

Android应用结构



Android应用标准目录结构



- ❖ `AndroidManifest.xml`文件是应用程序的清单文件
- ❖ `src`: 存放Java源代码
- ❖ `res`: 存放应用程序的资源文件，包括图片、音效、界面布局文件等，它们都能通过R资源清单类进行访问
- ❖ `assets`: 存放原生资源文件，其中的资源不能通过R资源清单类进行访问
- ❖ `libs`: 存放第三方或者应用内NDK开发得到的库文件
- ❖ `gen`: 存放aapt工具自动生成的Java代码
- ❖ `bin`: 存放编译结果和打包得到的文件，例如class文件、dex文件和apk文件等

AndroidManifest.xml文件



- ❖ 整个Android应用的全局描述文件
- ❖ (1)应用兼容的最低版本

```
<uses-sdk  
    android:minSdkVersion="8"  
    android:targetSdkVersion="18" />
```

- ❖ (2)应用所需的权限声明

```
<uses-permission  
    android:name="android.permission.INTERNET" />  
<uses-permission  
    android:name="android.permission.VIBRATE" />
```

AndroidManifest.xml文件



❖ (3)应用的基本信息(名称、图标、主题等)

```
<application  
        android:icon="@drawable/magic8"  
        android:label="@string/app_name"  
        android:screenOrientation="portrait"  
        android:theme="@style/android:Theme.NoTitleBar" >
```

❖ (4)应用的组件声明(activity、service、contentprovider等)

```
<activity android:name="edu.thu.magic.MagicActivity" />  
<activity android:name="edu.thu.magic.AboutActivity" />
```

R.java文件



应用资源字典类，为各种资源生成索引
文件: gen/edu/thu/magic/R.java [代码片段]

```
public static final class layout {
    public static final int activity_about=0x7f030000;
    public static final int activity_logo=0x7f030001;
    public static final int activity_magic=0x7f030002;
}
public static final class raw {
    public static final int shake_match=0x7f040000;
    public static final int shake_nomatch=0x7f040001;
    public static final int shake_sound=0x7f040002;
}
public static final class string {
    public static final int app_name=0x7f070000;
}
```

Android四大组件



Activity组件



Activity是应用程序的交互界面，类似Swing编程中的JFrame控件。所有Activity组件都是继承自Activity基类，它们拥有自己的生命周期。



Service组件

❖ Service和Activity地位并列，只是它通常位于后台运行，一般不需要和用户交互。所有Service组件都要继承自Service基类，它拥有自己的生命周期。



BroadcastReceiver组件

❖ BroadcastReceiver组件是一个广播接收器，类似于事件编程中的事件监听器，它监听的事件源是Android应用中的其他组件。



ContentProvider组件

ContentProvider为系统中为跨应用的数据交换提供了标准。应用可以通过ContentProvider提供数据，其他应用可以通过ContentResolver来访问数据。



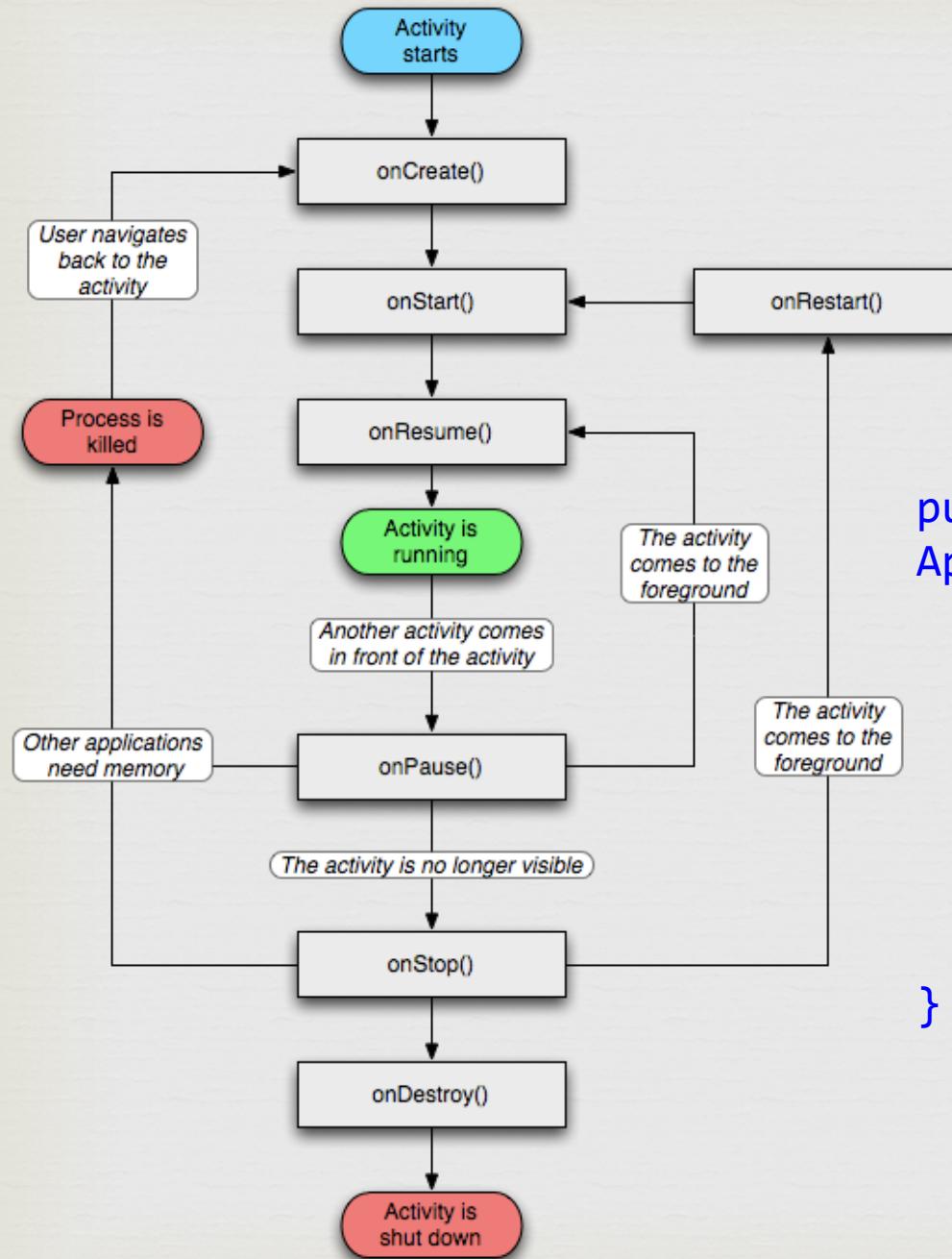
Activity生命周期



Activity生命周期



- ❖ 结合Servlet程序，浅谈对“生命周期”的理解；
- ❖ 生命周期是指某个对象具有几个基本状态(生命周期中的不同阶段)，对象实例在不同的状态下运行对应状态下的方法(生命周期方法)；
- ❖ 开发者不需要创建Servlet或者Activity的实例，它们的实例都由“外部容器”创建，对于Servlet来说是指Web容器，对于Activity来说是指Dalvik虚拟机；
- ❖ 开发者不需要调用它们的方法，它们的生命周期方法都是由“外部容器”通过回调的方式调用，开发者只需要实现生命周期方法即可；



```

public class Activity extends
ApplicationContext {
protected void onCreate(...);
protected void onStart();
protected void onRestart();
protected void onResume();
protected void onPause();
protected void onStop();
protected void onDestroy();
}
  
```

Activity简单实例



❖ “关于作者”界面

文件:src/edu/thu/magic/AboutActivity.java

```
public class AboutActivity extends Activity {  
    protected void onCreate(Bundle savedInstanceState){  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_about);  
    }  
  
    // 返回主界面  
    public void btn_about_back(View view) {  
        AboutActivity.this.finish();  
    }  
}
```

setContentView



- ❖ View组件是所有UI组件和容器组件的基类，它们需要放在容器组件中，然后通过Activity调用setContentView方法显示出来。
- ❖ setContentView()方法有两种使用方式：

1.传递一个布局管理器(Layout);

```
LinearLayout ll = new LinearLayout(this);
setContentView(ll);
```

2.传递一个布局管理资源的ID。

```
setContentView(R.layout.activity_about);
```

Android资源管理



Android应用资源



- ❖ Android应用资源可以分为两大类：
- ❖ (1) 可以通过R清单类访问的原生资源，保存在res中；
- ❖ (2) 不可以通过R清单类访问的资源，保存在assets中。
- ❖ 大部分资源都是放在res目录下，在应用进行编译的时候会在R类中为资源生成对应的索引项。

文件：gen/edu/thu/magic/R.java [代码片段]

```
public static final class layout {  
    public static final int activity_about=0x7f030000;  
    public static final int activity_logo=0x7f030001;  
    public static final int activity_magic=0x7f030002;  
}
```

资源的类型和保存位置



res/layout	布局XML文件
res/menu	菜单XML文件
res/anim	动画XML文件
res/drawable	图片文件或者是能够编译成各种Drawable对象的XML文件 多个命名为“drawable-?dpi”的文件夹存放的都是图片资源，只不过是为了适应不同的屏幕大小存放不同分辨率的图片而已
res/values	存放各种简单值的XML文件，这些简单值包括字符串、颜色、尺寸、样式、数组等，它们分别有对应的XML文件名，例如strings.xml, colors.xml, dimens.xml, styles.xml, arrays.xml
res/raw	原生资源，例如音频或者视频文件。在Java代码中可以通过openRawResource方法得到该资源的二进制流。但是，如果应用需要使用原生资源的话，推荐将原生资源存放在assets目录下，在Java代码中可以通过AssetManager来访问这些资源。

资源的使用方式



❖ 在Java代码中使用资源: R.[资源类别].[资源名称]

文件 `src/edu/thu/magic/AboutActivity.java` [代码片段]

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_about);  
}
```

❖ 在XML文件中使用资源: @[资源类别]/[资源名称]

`android:icon="@drawable/magic8"`

`android:label="@string/app_name"`

`android:textColor="@color/text_default"`

`android:textSize="@dimen/text_titlebar_center"`

`android:theme="@style/android:Theme.NoTitleBar"` [系统内资源]

字符串资源



❖ 定义应用中常用的字符串

文件 res/values/strings.xml

```
<resources>
    <string name="app_name">Magic8</string>
</resources>
```

颜色资源



❖ 定义应用中常用的颜色

文件 res/colors.xml

```
<resources>
    <color name="text_default">#ffffffff</color>
    <color name="app_default">#711098</color>
    <color name="text_orange">#ff6600</color>
    <color name="text_green">#009e0e</color>
    <color name="text_light">#666</color>
    <color name="text_blue">#36c</color>
    <color name="text_preference_key">#000</color>
    <color name="text_preference_value">#f60</color>
</resources>
```

尺寸资源



❖ 定义应用中常用的尺寸大小

❖ dp=dip=device independent pixels, sp=scaled pixels

文件 res/dimens.xml

```
<resources>
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="btn_big_textsize">18sp</dimen>
    <dimen name="text_default_size">15sp</dimen>
    <dimen name="titlebar_height">45dp</dimen>
    <dimen name="text_titlebar_size">12sp</dimen>
    <dimen name="text_titlebar_center">20sp</dimen>
    <dimen name="btn_titlebar_size">70dp</dimen>
</resources>
```

Drawable资源



- ❖ Drawable的各种子类都代表了一种Drawable资源，例如 `StateListDrawable`, `ShapeDrawable`, `AnimationDrawable`等等。
- ❖ StateListDrawable最常用，例如：

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable	btn_style_disabled"
          android:state_enabled="false"/>
    <item android:drawable="@drawable	btn_style_green_pressed"
          android:state_focused="true" android:state_pressed="true"/>
    <item android:drawable="@drawable	btn_style_green_pressed"
          android:state_focused="false" android:state_pressed="true"/>
    <item android:drawable="@drawable	btn_style_green_focused"
          android:state_focused="true"/>
    <item android:drawable="@drawable	btn_style_green_normal"
          android:state_focused="false"/>
</selector>
```

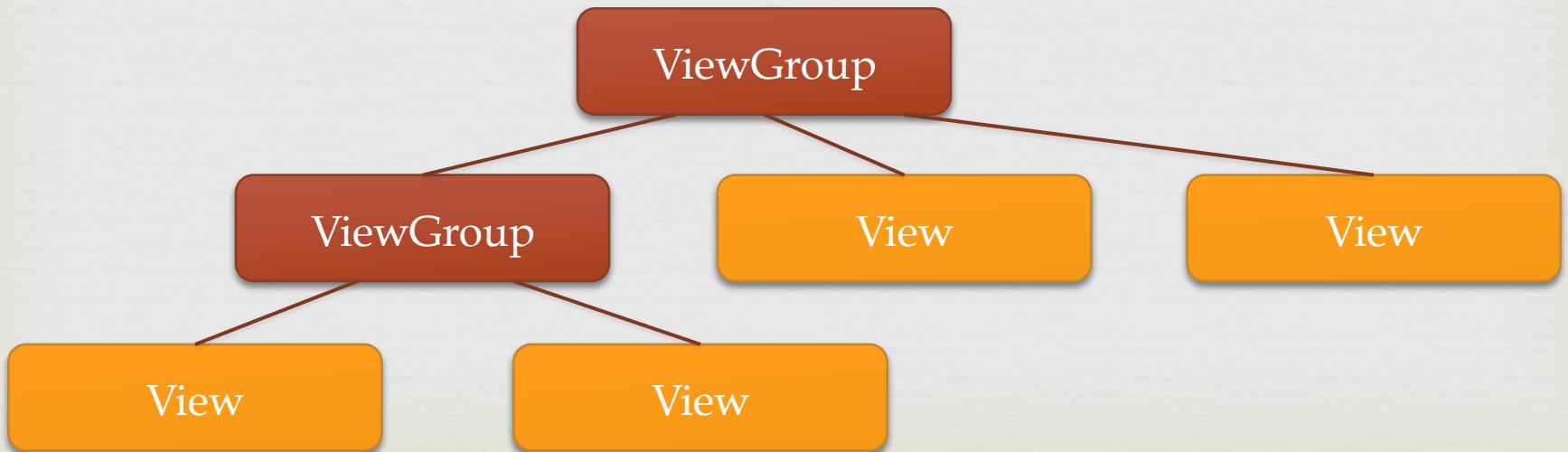
UI组件和容器组件



View和ViewGroup



- ❖ Android系统中的UI组件很多，它们都是建立在View和ViewGroup之上的，两者都是View的子类。
- ❖ ViewGroup通常是作为其他View组件的容器来使用，但是它也是View的子类，所以ViewGroup中还可以包含其他的ViewGroup。



组件的两种控制方式



❖ 在XML布局文件中通过XML属性进行控制

<Button

```
    android:id="@+id	btn_logo_magic"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="试试手气"/>
```

❖ 在Java代码中通过方法进行控制

```
Button button = new Button(this);  
button.setText("试试手气");  
button.setLayoutParams(new LayoutParams(LayoutParams.FILL_PARENT,  
LayoutParams.WRAP_CONTENT));
```

组件常用的XML属性



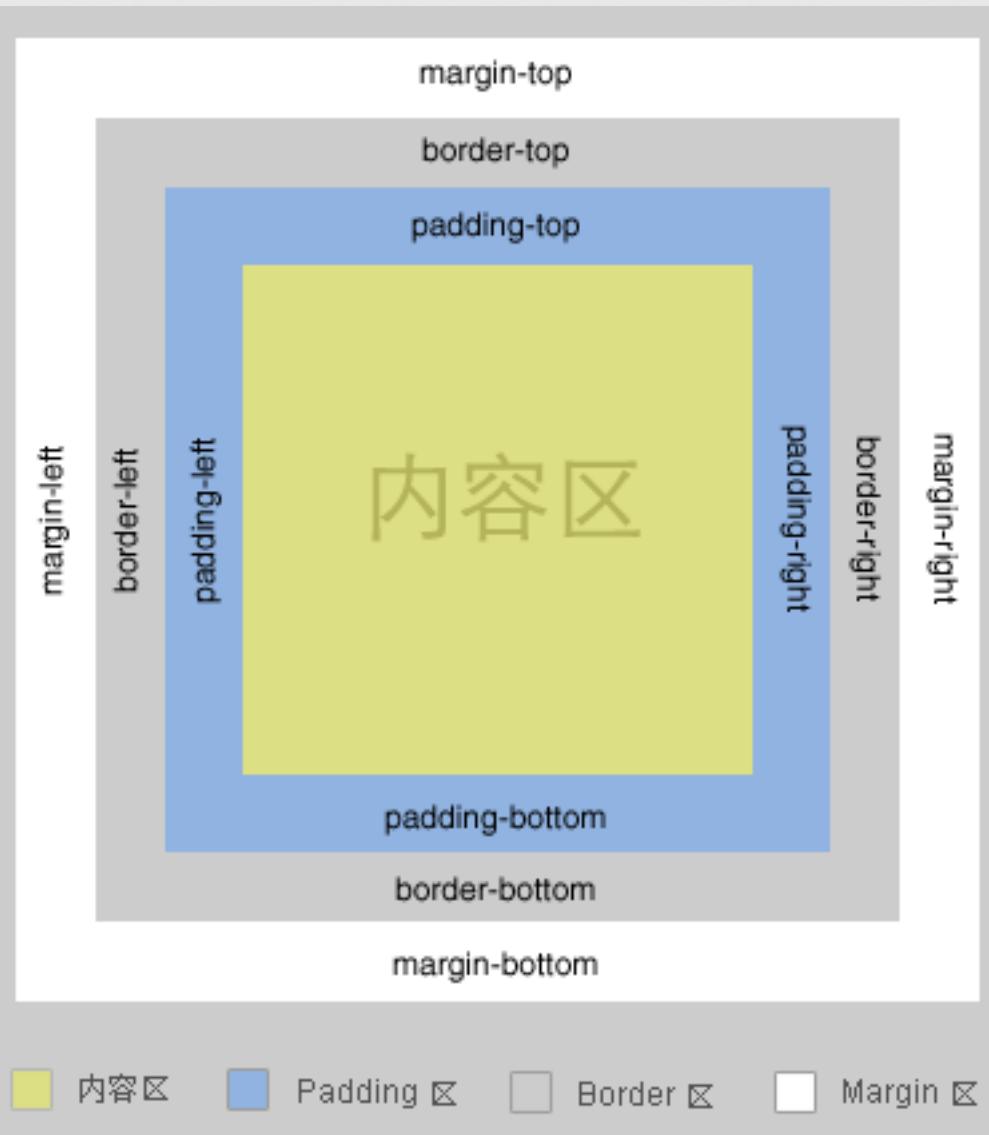
android:id	设置组件的唯一标识。Java代码可以通过 <code>findViewById</code> 方法来获取该组件对象 <code>android:id="@+id btn_logo_magic"</code>
android:layout_width	设置组件宽度，值为 <code>fill_parent</code> 或者 <code>match_parent</code> 或者 <code>wrap_content</code> <code>android:layout_width="fill_parent"</code>
android:layout_height	设置组件高度，值同上 <code>android:layout_height="wrap_content"</code>
android:text	设置组件上要显示的文本 <code>android:text="试试手气"</code>
android:background	设置组件的背景，可以是某张图片或者是某种颜色 <code>android:background="#111"</code> <code>android:background="@drawable btn_green"</code>

组件常用的XML属性



android:layout_margin [Top Bottom Left Right]	设置组件的外边距值[上下左右] <code>android:layout_marginLeft="10dp"</code>
android:layout_padding [Top Bottom Left Right]	设置组件的内边距值[上下左右] <code>android:layout_paddingLeft="10dp"</code>
android:visibility	设置组件的可见性，值为VISIBLE或者INVISIBLE或者GONE <code>android:visibility="GONE"</code>
android:onClick	为组件的单击事件绑定监听器 <code>android:onClick="btn_about_back"</code>

盒子模型



常用组件—文本框 TextView



文件: res/activity_about.xml

关于作者

```
<TextView  
    android:id="@+id/tv_aboutus_title"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerHorizontal="true"  
    android:layout_centerVertical="true"  
    android:text="关于作者"  
    android:textColor="@color/text_default"  
    android:textSize="@dimen/text_titlebar_center" />
```

常用组件—按钮 Button



文件: res/activity_logo.xml
<Button

```
    android:id="@+id	btn_logo_magic"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="10dp"  
    android:layout_marginRight="10dp"  
    android:background="@drawable	btn_style_green"  
    android:onClick="btn_logo_magic"  
    android:text="试试手气"  
    android:textColor="@color/text_default"  
    android:textSize="@dimen	btn_big_textsize" />
```



常用组件—图片 ImageView



文件: res/activity_about.xml

<ImageView

```
    android:id="@+id/iv_about_version"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_centerVertical="true"
    android:src="@drawable/settings_version" />
```



其他常用组件



- | | |
|---|------------------|
| ❖ 编辑框 EditText | 提示消息 Toast |
| ❖ 单选按钮 RadioButton | 复选框 CheckBox |
| ❖ 列表视图 ListView | 网格视图 GridView |
| ❖ 进度条 ProgressBar | 拖动条 Seekbar |
| ❖ 日期选择器 DatePicker | 时间选择器 TimePicker |
| ❖ 选项卡 TabHost | 滚动视图 ScrollView |
| ❖还有一大波组件没有列出来..... | |
| ❖ 更多内容参考[Android开发者网站：User Interface] | |

容器组件



- ❖ Android中的容器组件(布局管理器)大致有:
- ❖ **LinearLayout**: 线性布局
- ❖ **RelativeLayout**: 相对布局
- ❖ **AbsoluteLayout**: 绝对布局
- ❖ **FrameLayout**: 帧布局
- ❖ **TableLayout**: 表格布局
- ❖ **GridLayout**: 网格布局(Android4.0新增)

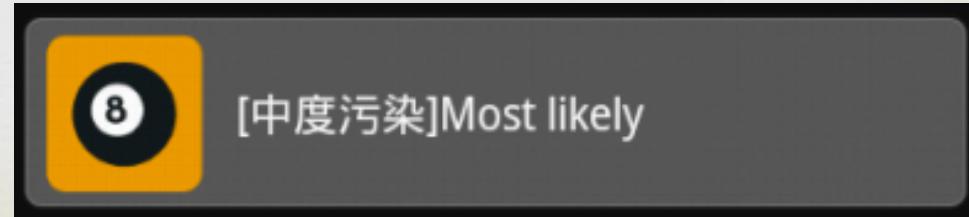
```
<LinearLayout
    android:id="@+id/rl_about_copyright"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:gravity="center_horizontal"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/tv_about_copyright"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Copyright © 2013 Tsinghua University."
        android:textColor="@color/text_default"
        android:textSize="12sp" />
    <TextView
        android:id="@+id/tv_about_copyrights"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:text="All Rights Reserved. "
        android:textColor="@color/text_default"
        android:textSize="12sp" />
</LinearLayout>
```

线性布局

Copyright © 2013 Tsinghua University.
All Rights Reserved.

```
<RelativeLayout
    android:id="@+id/rl_shake_result"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="60dp"
    android:background="@drawable/shake_match_ok" >
<ImageView
    android:id="@+id/iv_shake_magic8"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_alignParentLeft="true"
    android:layout_marginLeft="2dp"
    android:layout_centerVertical="true"
    android:src="@drawable/magic8" />
<TextView
    android:id="@+id/tv_shake_content"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:layout_toRightOf="@+id/iv_shake_magic8"
    android:padding="10dp"
    android:text="You are lucky!"
    android:textColor="#fff" />
</RelativeLayout>
```

相对布局



菜单 – Menu



菜单和其他组件一样，也有两种创建方式，既可以通过XML文件声明，也可以通过Java代码来完成。不论使用哪种方式，都要重写`onCreateOptionsMenu`方法。

文件 `src/edu/thu/magic/LogoActivity.java` [代码片段]

```
public boolean onCreateOptionsMenu(Menu menu) {  
    menu.add(0, 0, 0, "关于作者");//groupId,itemId,order  
    menu.add(0, 1, 0, "关闭音效");  
}
```

菜单事件处理



❖ 重写**onOptionsItemSelected**方法对菜单事件进行处理。

文件 `src/edu/thu/magic/LogoActivity.java` [代码片段]

```
public boolean onOptionsItemSelected(MenuItem item) {
    if (item.getItemId() == 0) {
        //点击了“关于作者”
    } else {
        //点击了“关闭音效”
    }
    return true;
}
```

未完待续...



胡家威

计研135班

<http://hujiaweibujidao.github.io/>