

Android开发入门



胡家威

计研135班

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

内容概要(下)



- ❖ 不可思议的9Patch图
- ❖ 那些绚丽的动画效果
- ❖ 我们来传递消息吧！
- ❖ 我想存数据肿么办？
- ❖ 不能没有的声音效果！
- ❖ 终于到了传感器应用！
- ❖ 亲，联个网试试？！
- ❖ 开发工具和技巧总结

不可思议的9Patch图



- ❖ 9Patch图片是一种特殊的png图片，以`.9.png`结尾，它是为了解决Android设备不同屏幕分辨率而设计的
- ❖ `%ANDROID_SDK%/tools/draw9patch`工具



不可思议的9Patch图



- 它在原始的图片四周各添加一个宽度为1像素的线条，这4条线条决定了该图片的缩放规则和内容显示规则
- 左边直线和上边直线围成的交集是可以水平和竖直缩放；右边直线和下边直线围成的交集是内容显示区域。

水平缩放

竖直缩放



那些绚丽的动画效果



- ❖ 动画的种类：逐帧动画(Frame)和补间动画(Tween)
- ❖ 逐帧动画：利用“人眼视觉暂留”原理，将多张静态图片按照顺序依次显示出来，看起来就像“动画”。
- ❖ **AnimationDrawable类**



以地鼠动画为例



文件 res/anim/rat_animation.xml [代码片段]

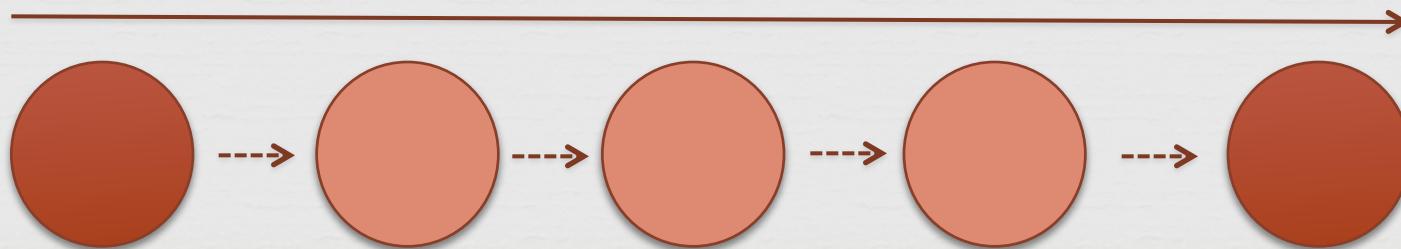
```
<animation-list xmlns:android="http://schemas.android.com/apk/res/  
    android" >  
    <item android:drawable="@drawable/show1" android:duration="100"/>  
    <item android:drawable="@drawable/show2" android:duration="100"/>  
    <item android:drawable="@drawable/show3" android:duration="100"/>  
    ...  
</animation-list>
```



那些绚丽的动画效果



- ❖ 补间动画：定义动画的开始帧和结束帧，动画中间的帧由系统计算出来自动补齐。
- ❖ AlphaAnimation：透明度改变的动画
- ❖ ScaleAnimation：大小缩放的动画
- ❖ TranslateAnimation：位移变化的动画
- ❖ RotateAnimation：旋转变化的动画

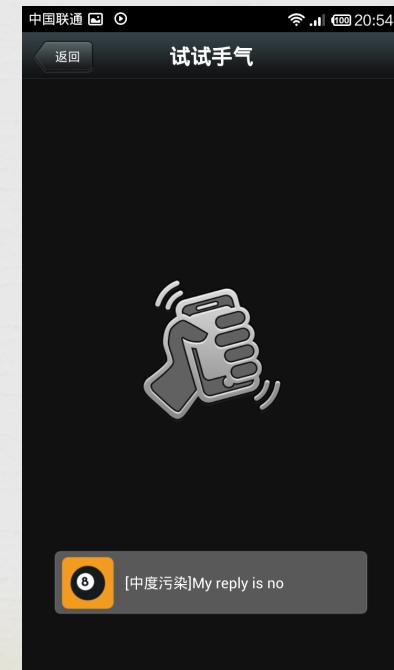


以位移动画为例



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

```
TranslateAnimation downAnimation = new TranslateAnimation(  
    Animation.RELATIVE_TO_SELF, 0f, //fromXType, fromXValue  
    Animation.RELATIVE_TO_SELF, 0f, //toXType, toXValue  
    Animation.RELATIVE_TO_SELF, -1.2f,  
    Animation.RELATIVE_TO_SELF, 0f);  
downAnimation.setDuration(1000);  
downAnimation.setFillAfter(true);  
rl_shake_result.setVisibility(View.VISIBLE);  
rl_shake_result.startAnimation(downAnimation);  
//1. Animation.RELATIVE_TO_SELF  
//2. Animation.RELATIVE_TO_PARENT  
//3. Animation.RELATIVE_TO_ABSOLUTE
```



我们来传递消息吧！



- ❖ 众多界面编程中广泛存在的多线程问题：“非UI线程不能直接访问UI线程中的组件！”
- ❖ Android如何解决多线程问题呢？**Handler**！
- ❖ 通过Handler的消息传递机制可以实现新启动的线程在运行时改变界面组件的属性值。
- ❖ 实现步骤：
 - (1)在新启动线程中发送消息
 - (2)在主线程中获取并处理消息
- ❖ 只要重写Handler类提供的消息处理方法，当新启动线程的消息发送时，上面的消息处理方法会自动回调

Java线程简要回顾



❖ 继承Thread类或者实现Runnable接口

Thread thread = new CustomThread()

Thread thread = new Thread(new CustomRunnable())

❖ 重写run()方法，调用start()方法启动线程

thread.start() -> thread.run()

❖ 线程同步 **synchronized** 关键字修饰方法或者同步块

public void synchronized method()

synchronized(obj){

//临界资源区 notify() notifyAll() wait()

}

Handler消息传递机制示例

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

//[主线程中]在创建Handler对象的时候指明如何处理消息

```
handler = new Handler() {
    public void handleMessage(Message message) {
        //handle message
    }
}
```

};

//[新线程中]创建一个消息对象，然后使用handler将消息发送出去

```
Message message = new Message();
if ("".equalsIgnoreCase(result.toString())) {
    message.obj = null;
} else {
    message.obj = result.toString();
}
handler.sendMessage(message);
```

我想存数据肿么办？



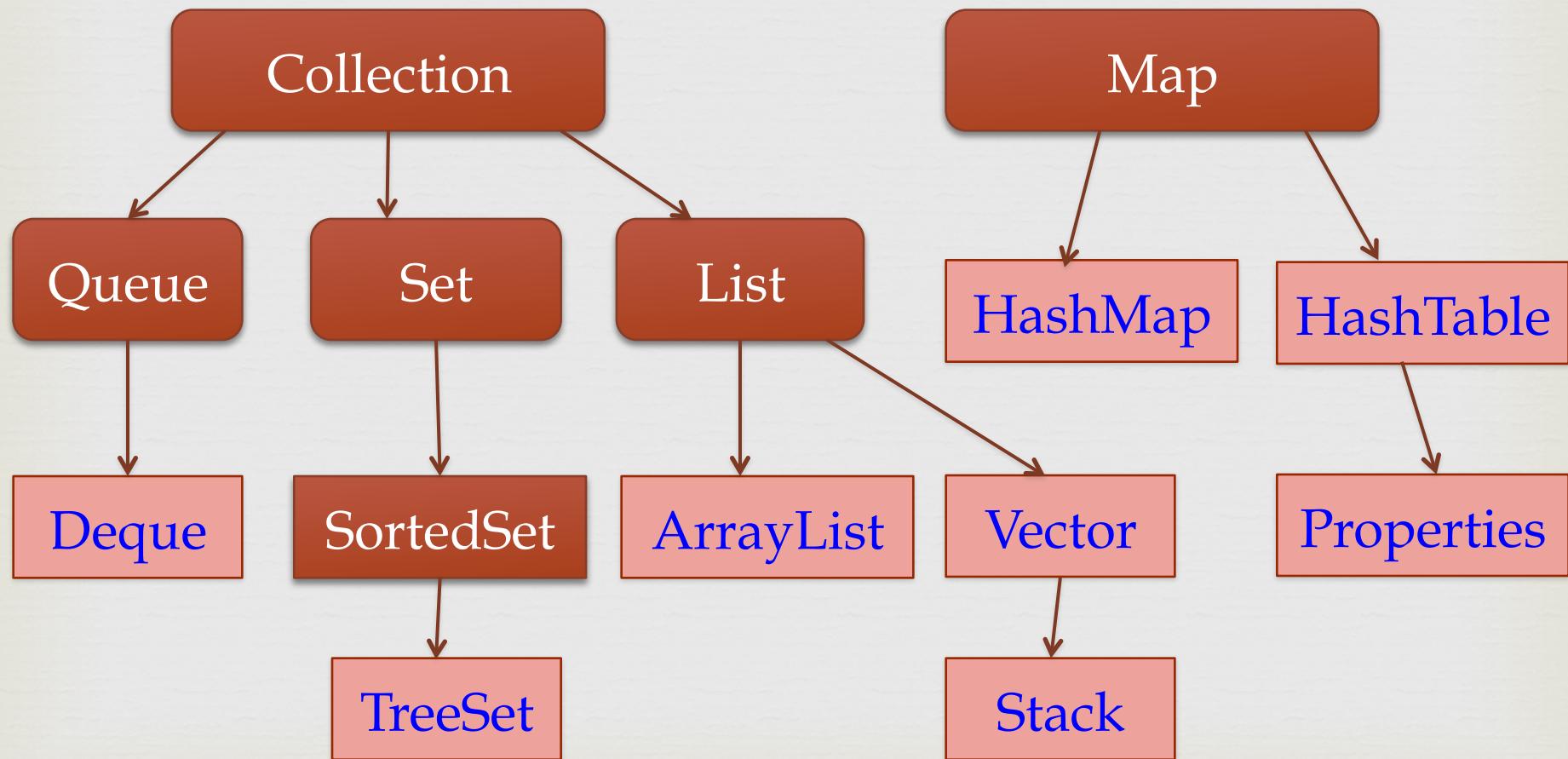
- ❖ 数据存储的几种方式：
- ❖ SharedPreferences：轻量级的键值对数据存储
- ❖ 读写设备存储中的文件
- ❖ 读写SD卡中的文件
- ❖ SQLite：轻量级关系型数据库 [很少使用，不讲解]

SharedPreference



- ❖ **SharedPreference**适用于保存少量的类似配置信息格式的键值对数据，其本质是Java集合框架中的Map。
- ❖ 主要方法**getXXX(String key, XXX defaultvalue)** 其中的XXX可以是int、float、boolean等基本类型
- ❖ SharedPreference接口本身没有提供写数据的功能，而其内部接口**Editor**提供了，通过edit()方法可以得到Editor对象。
- ❖ Editor对象的主要方法**putXXX(Stirng key, XXX value)**，存入指定key对应的数据，XXX同样可以是int、float、boolean等基本类型，要调用**commit()**方法才完成修改

Java集合框架回顾



SharedPreference示例



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

```
preferences = getSharedPreferences(CommonUtil.MAGIC_8, MODE_PRIVATE);
if (!preferences.contains(SOUND)) {
    Editor editor = preferences.edit();
    editor.putBoolean(SOUND, true);
    editor.commit();
}
```

// MODE_PRIVATE MODE_WORLD_READABLE MODE_WORLD_WRITEABLE
// 数据实际上保存在/data/data/%application_package%/shared_prefs目录下

读写设备存储中的文件



❖ Java IO流体系中包括了FileInputStream和 FileOutputStream，在Android中同样支持。

❖ Android提供的两个简便方法：

`FileInputStream openFileInput(String name);`

`FileOutputStream openFileOutput(String name, int mode);`

`MODE_PRIVATE`

只有当前程序能够读写

`MODE_APPEND`

以追加的形式打开该文件

`MODE_WORLD_READABLE`

该文件可以被其他应用程序读

`MODE_WORLD_WRITEABLE`

该文件可以被其他应用程序读写

❖ 文件实际保存在`/data/data/%application_package%/files`目录下

Java IO回顾: 装饰器模式



分类	字节输入流	字节输出流	字符输入流	字符输出流
抽象基类	InputStream	OutputStream	Reader	Writer
访问文件	FileInputStream	FileOutputStream	FileReader	FileWriter
访问数组	ByteArrayInputStream	ByteArrayOutputStream	CharArrayReader	CharArrayWriter
访问管道	PipInputStream	PipOutputStream	PipReader	PipWriter
缓冲流	BufferedInputStream	BufferedOutputStream	BufferedReader	BufferedWriter
对象流	ObjectInputStream	ObjectOutputStream		
转换流			InputStreamReader	OutputStreamWriter

读写SD卡中的文件



❖ 手机的内置存储毕竟有限，将大的数据文件保存在SD卡中则可以大大扩充手机的存储能力。

❖ 读写步骤：

(1) 判断手机是否插入了SD卡

`Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED)`

(2) 得到SD的根目录路径

`Environment.getExternalStorageDirectory()`

(3) 用`FileInputStream/FileOutputStream, FileReader/FileWriter`读写文件

❖ 必须声明的权限

`<uses-permission`

`android:name="android.permission.WRITE_EXTERNAL_STORAGE" />`

不能没有的声音效果



- ❖ 使用MediaPlayer来播放音频
- ❖ 当程序控制MediaPlayer对象装载音频完成之后， 程序可以调用start()、 stop()、 pause()方法来控制播放
- ❖ static createMediaPlayer(Context context, int resid)
装载resid资源ID指定的音频文件并返回MediaPlayer
- ❖ void setDataSource(……)方法来指定要播放的音频文件， 然后使用prepare()方法来装载音频文件， 这样就可以使用一个MediaPlayer对象来重复播放了， 大大地节省了系统资源。
- ❖ 需要调用release()自行释放MediaPlayer资源

音频播放示例



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

```
private int[] sounds = { R.raw.shake_sound, R.raw.shake_match,
R.raw.shake_nomatch };

soundPlayers = new MediaPlayer[sounds.length];
for (int i = 0; i < soundPlayers.length; i++) {
    soundPlayers[i] = MediaPlayer.create(this, sounds[i]);
    soundPlayers[i].setAudioStreamType(AudioManager.STREAM_MUSIC);
}

// 播放音频部分 首先查看应用配置是否开启音效
if (preferences.getBoolean("sound", true)) {
    soundPlayers[0].start();
}
```

那视频呢？



❖ 使用VideoView播放视频

❖ 使用步骤：

(1)在布局文件或者代码中定义VideoView组件

(2)调用VideoView的方法装载视频文件

setVideoPath(String path)

setVideoURI(Uri uri)

(3)调用VideoView的**start()**、**stop()**、**pause()**来控制视频播放

终于到了传感器应用



❖ Android中的传感器

方向: SensorManager.SENSOR_ORIENTATION

磁场: SensorManager.SENSOR_MAGNETIC_FIELD

光线: SensorManager.SENSOR_LIGHT

距离: SensorManager.SENSOR_PROXIMITY

温度: SensorManager.SENSOR_TEMPERATURE

加速度: SensorManager.SENSOR_ACCELEROMETER

陀螺仪: SENSOR_TYPE_GYROSCOPE

加速度传感器使用示例



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

```
sensorManager=(SensorManager)mContext.getSystemService  
(Context.SENSOR_SERVICE);  
if (sensorManager != null) {  
    sensor = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);  
}  
if (sensor != null) {  
    sensorManager.registerListener(this, sensor, SensorManager.SENSOR_DELAY_GAME);  
}  
  
// 加速度传感器数据发生变化时的回调函数  
public void onSensorChanged(SensorEvent event) { ... ... }
```

亲， 联个网试试？！



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

```
URL url = new URL(CommonUtil.AQIDATA_URL);
HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
urlConnection.setRequestMethod("GET");
BufferedReader reader = new BufferedReader(new
InputStreamReader(urlConnection.getInputStream()));
StringBuffer result = new StringBuffer();
String line = null;
while ((line = reader.readLine()) != null) {
    result.append(line);
}
System.out.println(result);
// 2013-12-8-14-Very Unhealthy-234-120-43-29-5-9-9--6-34-1019
urlConnection.getInputStream().close();
urlConnection.disconnect();
```

Java网络编程回顾



- ❖ **java.net** 包为Java提供了基本网络支持
- ❖ URL(Uniform Resource Locator) 统一资源定位器
协议名 主机名 端口 资源
<http://hujiaweibujidao.github.io/>
- ❖ **openStream()**: 打开与此URL的连接，并返回用于读取该URL资源的输入流，得到了输入流我们便可以对资源进行读取和分析。
- ❖ 基于TCP协议的网络编程: Socket
- ❖ 基于UDP协议的网络编程: DatagramSocket

空气质量数据来源



- ❖ 初始来源：
<http://aqicn.org/city/beijing/m/>
- ❖ BAE应用BeijingAQIData采集数据
- ❖ 简易接口：
<http://beijingaqidata.duapp.com/hello>
- ❖ 返回结果：
2013-12-22-21-Hazardous-398-197-54-62-7--5-2--12-59-1031

开发工具和技巧总结



❖ 真机调试还是用模拟器？

简单应用直接使用模拟器就行了，但是模拟器能力有限，开发复杂应用(例如一些和传感器有关的应用)时无法用它测试；真机调试需要开启手机的“USB调试模式”，使用久了对手机硬件会有影响。

❖ USB调试还是无线调试？

如果手机获取了root权限的话，可以使用一些无线调试应用(例如adbwireless)进行无线调试，不需要用数据线将手机连接电脑，好处是方便，但是前提是手机一定要root

开发工具和技巧总结

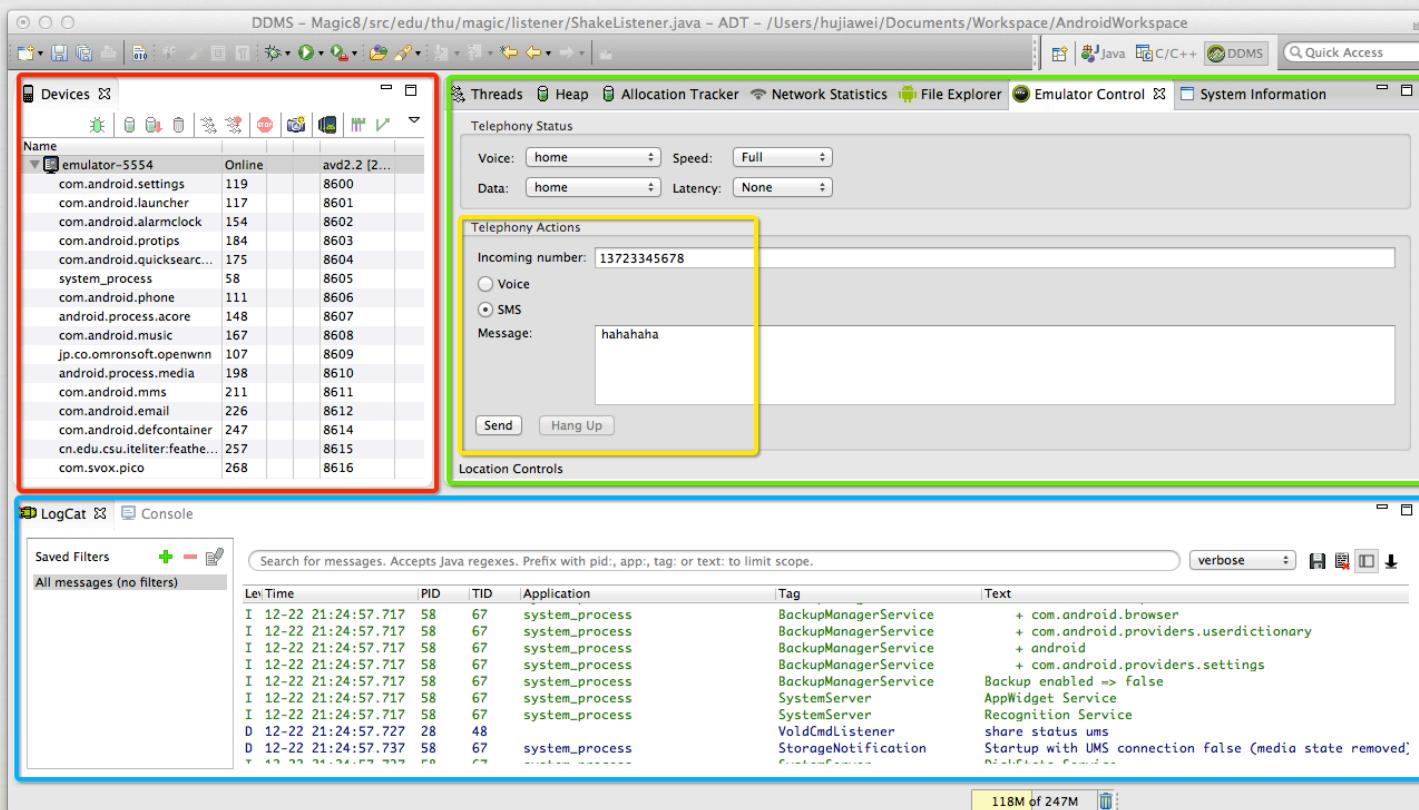


- ❖ ADB(Android Debug Brige)工具：帮助开发者管理设备或者模拟器的状态，或者执行一些操作
- ❖ 常用的命令
- ❖ **adb start-service/kill-service** 启动和关闭adb服务
- ❖ **adb devices** 显示与adb相连的设备和模拟器
- ❖ **adb push <local> <remote>** 将电脑中的文件发送到设备或者模拟器中
- ❖ **abd pull <remote> <local>** 将设备或者模拟器中的文件拷贝到电脑中
- ❖ **adb shell** 进入设备的系统中，可以执行shell命令

开发工具和技巧总结



❖ DDMS(Dalvik Debug Monitor Service)



写在最后



- ❖ 关于本课程的PPT资料和魔力8号球的源代码以及其他移动开发资料推荐请看我博客中的一篇文章
《移动开发资料汇集》
- ❖ 分Android、iOS和Windows Phone三个部分
- ❖ 分别推荐了一些比较实用的网站和书籍

Merry Christmas



谢谢！



胡家威
计研135班

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