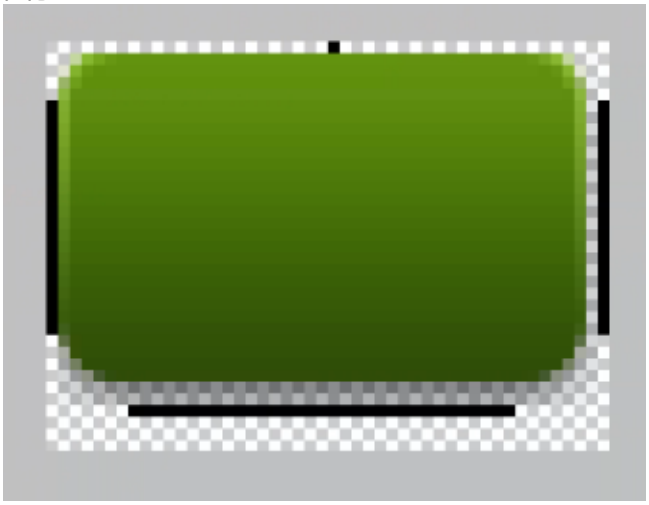
手机卫士第二天

.9.png(9-Patch图)

> 通过黑边来描述图片的可拉伸区域以及可填充文字的区域
>
> - 上边线表示水平方向可拉伸的区域, 左边线表示垂直方向可拉伸的区域
> - 右边线表示垂直可填充的区域, 下边线表示水平方向可填充的区域
>
> 图示:
>
> 

状态选择器的应用:制作按钮的动态背景

> 1. 先在res目录下创建drawable目录, 然后再目录中创建状态选择器的xml文件
>
> ```
> 1.  <selector xmlns:android="http://schemas.android.com/apk/res/android">
> 2.   <item android:state_focused="true" android:drawable="@drawable/button_
>     pressed"></item>
> 3.   <item android:state_pressed="true" android:drawable="@drawable/button_
>     pressed" ></item><!--被按下后要显示的按钮背景-->
> 4.   <item android:drawable="@drawable/button_normal"></item><!--自然组状态
>     下要显示的按钮背景-->
> 5.  </selector>
> ```
>
> 2. 在布局文件中应用创建的按钮背景选择器
>
> ```
> 1.  <Button
> 2.   android:layout_width="wrap_content"
> 3.   android:layout_height="wrap_content"
> 4.   android:layout_alignParentBottom="true"
> 5.   android:layout_alignParentRight="true"
> 6.   android:layout_margin="5dp"
> 7.   android:background="@drawable/button"<!--应用-->
> 8.   android:text="下一步"/>
> ```

自定义shape

> 可以通过在drawable文件夹下新建一个包含shape标签的xml文档来定义一个形状,这个形状可以

> 当作一个普通的图像来使用, 可以定义各种形状.
> 例:shape_textview.xml 这个shape被用来作为textview点击事件的背景

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
    <corners android:radius="5dp" />
    <solid android:color="#a000" />
</shape>
```

滑动手势识别器

> 检测用户手指在屏幕上横向滑动的事件来响应相应的事件

1. 创建手势检测器对象

```
gestureDetector = new GestureDetector(this,
 new SimpleOnGestureListener() {
 //重写该监听器的onfling方法, 监听手势滑动事件
 @Override
 public boolean onFling(MotionEvent e1, MotionEvent e2,float velocityX,
 float velocityY) {
       //as your wish
       //balabalabala....
       }
});
```

2. 重写activity的onTouchEvent方法, 将用户点击事件先交由自己创建的手势检测器对象(**这一步很重要, 如果没有这一步效果将无法达成!**)

```
@Override
 public boolean onTouchEvent(MotionEvent event) {
     //将activity的touch时间先交与手势识别器处理
     gestureDetector.onTouchEvent(event); //很重要
     return super.onTouchEvent(event);
 }
```

广播接收者接收短信

> 1. 创建BroadcastReceiver类的对象并在AndroidManifest.xml文件中注册

```
<receiver android:name="com.wangmeng.phonedefender.receiver.SMSReceiver
">
<intent-filter android:priority="4294967295">
   <action android:name="android.provider.Telephony.SMS_RECEIVED"/>  <!
-- 用于接收收到短信时的广播 -->
</intent-filter>
</receiver>
```

> 同时要添加一个接收短信的权限

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

2. 在OnReceive中进行如下的操作即可将短信的信息读取出来

```
1.  public void onReceive(Context context, Intent intent) {
2.  //获取短信的内容和发送短信的源地址
3.  Object[] objs = (Object[])intent.getExtras().get("pdus");
4.  for (Object object : objs) {
5.      SmsMessage message =SmsMessage.createFromPdu((byte[])object);
6.      String address = message.getOriginatingAddress(); //获取发送短信的手
    机号
7.      String body = message.getMessageBody(); //获取短信的内容
8.  }
```