

《Android UI设计(面向创建移动应用的产品经理、设计师和开发者，通过众多案例深入解读Android UI设计的方法和技巧，从实用角度出发，诠释以用户为中心的设计方法)》

书籍信息

版次：1

页数：

字数：

印刷时间：2015年03月01日

开本：16开

纸张：胶版纸

包装：平装

是否套装：否

国际标准书号ISBN：9787111488552

丛书名：UI/UE系列丛书

内容简介

《Android UI设计》通过众多案例深入解读Android UI设计的方法和技巧，从实用角度出发，诠释以用户为中心的设计方法。以Google设计语言Material Design指导UI设计模式，轻松自信地设计和交付精美的移动App

目录

前言	
第1章 Android开发基础	
1.1 Eclipse中的Android开发	
1.1.1 创建项目	
1.1.2 创建AVD	
1.1.3 运行项目	
1.2 Android项目架构	
1.2.1 Java代码解析	
1.2.2 项目资源解析	
1.2.3 AndroidManifest.xml解析	
1.3 Eclipse中的常用窗口	
1.3.1 Console窗口	
1.3.2 LogCat窗口	
1.3.3 DDMS窗口	
前言第1章 Android开发基础 1.1 Eclipse中的Android开发 1.1.1 创建项目 1.1.2 创建AVD 1.1.3 运行项目 1.2 Android项目架构 1.2.1 Java代码解析 1.2.2 项目资源解析 1.2.3 AndroidManifest.xml解析 1.3 Eclipse中的常用窗口 1.3.1 Console窗口 1.3.2 LogCat窗口 1.3.3 DDMS窗口 1.4 移动App的设计原则 1.5 移动App的设计风格 1.5.1 扁平化设计 1.5.2 卡片式设计第2章 ADT中的UI设计 2.1 图形布局编辑器 2.2 几种常见的布局方式 2.2.1 创建布局 2.2.2 相对布局 2.2.3 线性布局 2.2.4 帧布局 2.3 优化布局 2.3.1 复用布局 2.3.2 多设备支持 2.3.3 使用Hierarchy Viewer调试用户界面 2.4 界面布局技巧 2.4.1 布局设计原则 2.4.2 布局设计技巧 2.5 习题第3章 Activity与UI管理 3.1 Activity基础 3.1.1 创建Activity 3.1.2 Activity的生命周期 3.1.3 退出Activity 3.2 Activity之间的调用 3.2.1 调用其他Activity 3.2.2 Activity的回调 3.3 Activity之间的数据传递 3.3.1 使用Intent传递数据 3.3.2 使用Bundle传递数据 3.3.3 使用Application共享数据 3.4 Activity栈与任务 3.4.1 Activity栈 3.4.2 任务管理 3.4.3 Activity的加载模式 3.4.4 保存Activity的状态 3.5 应用主页设计技巧 3.6 用户体验设计 3.7 习题第4章 Android组件编程 4.1 Service与后台服务 4.1.1 创建Service 4.1.2 Service的生命周期 4.1.3 Started Service 4.1.4 Bound Service 4.2 ContentProvider与数据共享 4.2.1 系统中的ContentProvider 4.2.2 通用资源标志符 4.2.3 使用ContentProvider 4.3 BroadcastReceiver与广播意图 4.3.1 BroadcastReceiver的工作机制 4.3.2 广播的类型 4.3.3 接收广播 4.3.4 注册广播 4.4 Intent与组件通信 4.4.1 Intent处理机制 4.4.2 Intent对象 4.4.3 Intent的解析 4.5 基于组件的应用模型 4.6 习题第5章 Widgets设计与事件处理 5.1	

表单控件设计 5.1.1 文本控件 5.1.2 按钮控件 5.1.3 单选/复选按钮控件 5.1.4 进度条控件 5.2 适配器控件设计 5.2.1 适配器概述 5.2.2 Gallery 5.2.3 Spinner 5.2.4 ListView 5.2.5 GridView 5.2.6 适配器控件的大数据加载 5.3 Widgets事件处理 5.3.1 按键事件处理 5.3.2 触屏事件处理 5.3.3 手势事件处理 5.3.4 感应器事件处理 5.4 Widgets设计技巧 5.4.1 官方设计指引 5.4.2 表单控件设计技巧 5.4.3 数据加载模式设计 5.4.4 搜索设计技巧 5.5 习题第6章 对话框、菜单与导航 6.1 对话框设计 6.1.1 Toast通知 6.1.2 Notification提示 6.1.3 AlertDialog对话框 6.1.4 对话框的托管 6.2 菜单设计 6.2.1 Options Menu 6.2.2 Context Menu 6.3 动作栏与导航设计 6.3.1 动作栏设计 6.3.2 ActionMode设计 6.3.3 导航设计 6.3.4 导航设计技巧 6.4 用户通知设计技巧 6.4.1 Android中的消息提示 6.4.2 通知设计策略 6.4.3 通知设计原则 6.4.4 通知的导航机制 6.4.5 声音提醒 6.5 习题第7章 容器UI设计 7.1 导航类容器设计 7.1.1 使用ViewPager设计导航页 7.1.2 使用ViewFlipper设计滑屏窗口 7.1.3 使用TabHost设计标签页 7.2 特定容器设计 7.2.1 使用WebView显示网页 7.2.2 使用MapView显示地图 7.2.3 使用VideoView播放视频 7.3 第三方容器控件设计 7.3.1 使用SlidingMenu设计菜单容器 7.3.2 使用TimesSquare设计日期 7.4 引导页设计技巧 7.5 习题第8章 自定义控件设计 8.1 概述 8.2 定制控件 8.3 重载控件 8.3.1 重构AdapterView 8.3.2 应用控件 8.4 绘制UI 8.5 开源UI工具 8.6 习题第9章 样式、主题与动画设计 9.1 样式与主题 9.1.1 Style 9.1.2 Theme 9.2 动画设计 9.2.1 帧动画 9.2.2 补间动画 9.2.3 属性动画 9.3 应用风格设计 9.4 动态效果设计 9.4.1 动态设计原则 9.4.2 动态设计技巧 9.5 习题第10章 桌面UI设计 10.1 设计简单的桌面组件 10.1.1 RemoteViews 10.1.2 AppWidgetProviderInfo 10.1.3 AppWidgetProvider 10.1.4 声明App Widgets 10.2 配置和管理桌面组件 10.2.1 Configuration Activity 10.2.2 AppWidgetManager 10.3 设计集合桌面组件 10.3.1 Collection Views 10.3.2 RemoteViewsService 10.3.3 RemoteViewsFactory 10.3.4 子视图事件 10.4 桌面组件设计规范 10.4.1 桌面组件的种类 10.4.2 桌面组件的尺寸 10.4.3 桌面组件设计技巧 10.5 习题第11章 平板UI设计 11.1 Fragment概述 11.1.1 Fragment布局特性 11.1.2 Fragment与Activity 11.2 创建Fragment 11.2.1 创建ListFragment 11.2.2 创建Fragment 11.2.3 添加Fragment到Activity 11.2.4 使用Support Library 11.3 管理Fragment 11.3.1 Fragment的生命周期 11.3.2 使用FragmentManager处理事务 11.3.3 Fragment之间的通信 11.4 PreferenceFragment 11.5 平板UI设计技巧 11.5.1 Google的准则 11.5.2 横竖屏布局设计 11.5.3 常见平板布局 11.6 习题第12章 Android UI综合应用 12.1 项目概述 12.2 用户界面设计 12.2.1 结构设计 12.2.2 交互设计 12.2.3 视觉设计 12.3 用户界面功能实现 12.3.1 主界面设计 12.3.2 歌曲列表界面设计 12.3.3 系统设置界面设计 12.3.4 桌面应用组件设计 12.4 UI测试 参考文献

[显示全部信息](#)

在线试读部分章节

第1章

Android开发基础

1.1 Eclipse中的Android开发

Eclipse是著名的跨平台开源集成开发环境，对开发Android应用提供了良好的支持。

1.1.1 创建项目

在Eclipse中创建Android项目的步骤如下：

启动Eclipse集成开发环境。

运行File New Android Application Project菜单命令，打开New Android Application向导，显示如图1-1所示界面。

在New Android Application向导中输入如下信息：

Application Name：HelloWorld

Project Name：HelloWorld

Package Name：com.liweiyong.helloworld（包的名称必须和所有安装在Android系统中的应用程序的包名不相同）

其他默认选择如下：

Minimum Required SDK：API 14

Target SDK：API 18

Compile With：API 19

Theme：Holo Light with Dark Action Bar

单击Next按钮，默认Configure Project设置和Configure the attribute of the icon set界面设置，选择BlankActivity，单击Finish按钮，完成Hello World项目的创建。第1章 Android开发基础 1.1 Eclipse中的Android开发

Eclipse是著名的跨平台开源集成开发环境，对开发Android应用提供了良好的支持。 1.1.1

创建项目 在Eclipse中创建Android项目的步骤如下： 启动Eclipse集成开发环境。

运行File New Android Application Project菜单命令，打开New Android

Application向导，显示如图1-1所示界面。 在New Android

Application向导中输入如下信息： Application Name：HelloWorld Project

Name：HelloWorld Package Name：com.liweiyong.helloworld（包的名称必须和所有安装在

Android系统中的应用程序的包名不相同）其他默认选择如下： Minimum Required

SDK：API 14 Target SDK：API 18 Compile With：API 19 Theme：Holo Light with Dark

Action Bar 单击Next按钮，默认Configure Project设置和Configure the attribute of the icon

set界面设置，选择BlankActivity，单击Finish按钮，完成Hello World项目的创建。图1-1

New Android Application向导 1.1.2 创建AVD Android Virtual Device（简称AVD）是运行An

droid项目的虚拟设备。AVD通过对硬件和软件的配置进行定义来模拟一个实际的设备。

1.2创建SD Card映像文件 Android模拟器自身已经具备了一个持久化存储空间，但这并不

够大，有时需要为应用程序和文件提供更大的存储空间。为了在模拟器上开发使用扩展

存储空间的程序，需要在PC上模拟一个SD Card（Secure Digital Memory Card，一种基于

半导体快闪记忆器的新一代记忆设备）的虚拟文件，然后加载到模拟器中。创建SD

Card虚拟文件的步骤如下： 在Windows中，运行“开始” “运行”菜单命令，在打

开的“运行”窗口中输入“cmd”并单击“确定”按钮，打开命令行窗口。

在窗口中输入如下命令： mksdcard -l mycard 500M C:\mysdcard.img

该命令的含义是在本地磁盘C盘创建一个500 MB大小的映像文件mysdcard.img。此时，查

看C盘可以看到一个名称为mysdcard.img的文件。 2.2创建AVD并关联SD Card

在Eclipse中创建AVD并关联SD Card的步骤如下： 在Eclipse中，运行Window Android

Virtual Device Manager菜单命令，打开Android Virtual Device

Manager对话框。单击对话框中的New按钮，弹出如图1-2所示的创建AVD对话框。在AVD Name文本框中输入AVD的名称（可以自定义）。Device用于设置模拟器的尺寸和分辨率。SDK提供的常见分辨率（sdk\platforms\android-#\skins\）包括：HVGA（320×480）、QVGA（320×240）、WQVGA400（400×240）、WQVGA432（432×240）、WVGA800（800×480）、WVGA854（854×480）和WXGA（1280×800）等。在Target中选择需要的SDK版本（平板电脑开发的最低版本是Android API 11）。SD Card的大小可以自定义输入数值；也可以选择File单选按钮后，单击Browse按钮，在打开的对话框中选择前面创建的SD Card映像文件。其他默认设置即可。

单击OK按钮，完成AVD的创建。

1.1.3 运行项目

下面介绍在Eclipse中运行Android项目的方法，步骤如下：

在Eclipse的项目窗口中，右击项目节点名称HelloWorld，运行Run as Run Conf?igurations菜单命令，打开Run Conf?igurations对话框。在对话框的左侧选择Android Application，并单击上方的New launch

conf?iguration按钮，在右侧的Android标签页中单击Browse按钮，打开Project

Selection对话框，选择HelloWorld项目，如图1-3所示。图1?3 Run

Conf?igurations对话框——选择项目 在对话框的Target标签页中的AVD列表中勾选合适的Android模拟器，如图1-4所示。单击Apply和Run按钮，运行程序。图1?4 Run

Conf?igurations对话框 图1-5显示了项目的运行结果。图1?5 项目运行结果

1.2 Android项目架构

在Eclipse的Package Explorer窗口中展开HelloWorld项目，项目架构如图1-6所示。一个Android工程包含了组成Android应用的所有源代码文件。Android工程主要由src、gen、assets、res文件夹和AndroidManifest.

xml等文件组成，下面分别对其进行解析。

1.2.1 Java代码解析

Android项目的Java代码主要存放在src文件夹和gen文件夹的包文件夹下。

1.?MainActivity解析

src文件夹用来存放项目的源代码。在Package Explorer窗口中，展开项目的src文件夹，打开通过向导生成的MainActivity.java，核心代码如下：

```
01 public class MainActivity extends Activity {
02     03 @Override
04     protected void onCreate(Bundle savedInstanceState) {
05         super.onCreate(savedInstanceState);
06         setContentView(R.layout.activity_main);
07     }
08     09 @Override
10     public boolean onCreateOptionsMenu(Menu menu) {
11         getMenuInflater().inflate(R.menu.main, menu);
12         return true;
13     }
14     15 }

```

01行的MainActivity是一个用户定义的Activity，继承自android.app.Activity类。Activity是应用程序的表示层，用于构建应用程序的UI界面。关于Activity的使用方法将在第3章介绍。03~07行重载的onCreate（Bundle）方法是Activity生命周期的组成部分，用于初始化Activity。例如界面的显示内容通过调用setContentView()方法来指定显示布局，如activity_main.xml。然后通过findViewById（int）方法在布局中检索需要交互的UI控件。

04行的Bundle类用于Activity之间传递数据。该类提供了公有方法containsKey（String key），如果给定的key包含在Bundle的映射中，则返回true，否则返回false。该类实现了Parcelable和Cloneable接口，所以它具有这两者的特性。05行的super.onCreate（savedInstanceState）方法的作用是调用父类中的onCreate()方法来实现对界面的绘制工作。注意，从savedInstanceState中读取保存到存储设备中的数据时，需要判断savedInstanceState是否为null，因为Activity第一次启动时并没有数据被存储在设备中。有关savedInstanceState的介绍参见3.4.4节。09~13行的onCreateOptionsMenu（Menu）方法是Activity中提供的用于创建菜单项

的回调方法。通过其中的getMenuInflater().inflate (int , Menu) 方法加载res/menu中定义的菜单资源。onCreateOptionsMenu (Menu) 方法通常和OnOptionsItemSelected (MenuItem) 回调方法配合使用，以响应菜单的选择事件。

2. R.java解析
打开gen/com.liweiyong.helloworld中的R.java文件，部分代码如下：
01 /* AUTO-GENERATED FILE. DO NOT MODIFY. 02 * 03 * This class was automatically generated by the 04 * aapt tool from the resource data it found. It 05 * should not be modified by hand. 06 */ 07 08 public final class R { 09 public static final class array { 10 public static final int action_file=0x7f040005; 11 ... 12 } 13 public static final class layout { 14 public static final int activity_main=0x7f030000; 15 } 16 ... 17 }
程序的第01 ~ 06行为注释说明，R.java文件由aapt工具根据res中的资源自动生成，不要手动修改该文件。R.java由ADT根据res中的资源自动生成drawable、layout、string等静态匿名内部类。不同的静态内部类分别根据其res中的资源定义一系列资源标识符，如“ public static final int activity_main=0x7f030000; ”对应的是layout目录下的activity_main.xml文件。每当res中的资源发生变化，aapt工具都会自动在R.java对应的内部类中生成一个静态int类型的常量，以对新添加的资源进行索引。如果从res中删除一个资源，R.java中对应的索引也会自动删除。通过R.java可以很快地查找需要的资源，另外编译器也会检查R.java列表中的资源是否被使用，没有被使用到的资源不会编译进apk中，这样可以减少应用在手机中占用的空间。

3. BuildConfig.java解析

打开gen/com.liweiyong.helloworld中的BuildConfig.java文件，代码如下：
01 /** Automatically generated file. DO NOT MODIFY */ 02 package cn.liweiyong.helloworld; 03 04 public final class BuildConfig { 05 public final static boolean DEBUG = true; 06 }
ADT允许开发者只在调试模式下运行某些代码。BuildConfig类包含一个DEBUG常量，该常量会根据Build类型自动设置值。可以通过BuildConfig.DEBUG常量来编写只在Debug模式下运行的代码。如果有些代码不想在应用发布后执行，也可以使用该功能。

[显示全部信息](#)

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

[更多资源请访问www.tushupdf.com](http://www.tushupdf.com)