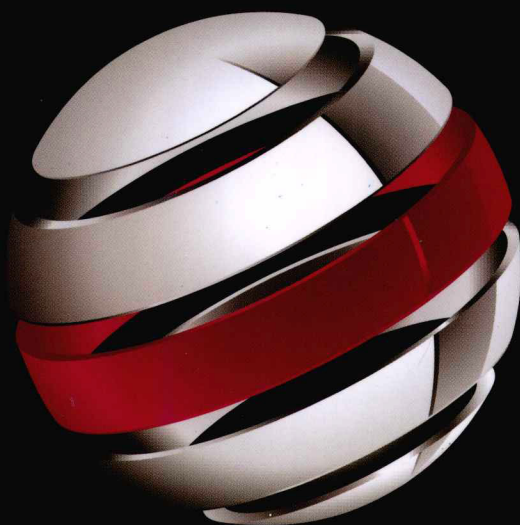


- 赢取移动开发新视野
- 从容步入Android平板电脑编程殿堂
- 由浅入深，循序渐进



Beginning Android Tablet Programming

# Android平板电脑编程 基础教程

[澳] Robble Mattnes 著  
袁国忠 译



人民邮电出版社  
POSTS & TELECOM PRESS

**TURING** 图灵程序设计丛书 移动开发系列



Beginning Android Tablet Programming

# Android平板电脑编程 基础教程

[澳] Robble Matthews 著

人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

Android平板电脑编程基础教程 / (澳) 马修斯  
(Matthews, R.) 著 ; 袁国忠译. -- 北京 : 人民邮电出  
版社, 2012. 6

(图灵程序设计丛书)

书名原文: Beginning Android Tablet Programming  
ISBN 978-7-115-28201-9

I. ①A… II. ①马… ②袁… III. ①移动终端—应用  
程序—程序设计—教材 IV. ①TN929.53

中国版本图书馆CIP数据核字(2012)第104750号

## 内 容 提 要

本书以 Android Honeycomb 为基础, 全面介绍了如何针对 Android 平板电脑编程, 包括关于 Android 程序的基本结构的知识、Android 操作系统的工作原理、2D/3D 编程, 以及如何部署自己的系统、如何从零开始编写 Android 平板电脑应用、如何利用触摸屏界面和 Honeycomb SDK 等。

本书适合 Android 程序员、Java 程序员以及其他使用开源编程语言 (特别是 Python) 的程序员阅读。

图灵程序设计丛书

## Android 平板电脑编程基础教程

◆ 著 [澳] Robbie Matthews

译 袁国忠

责任编辑 朱 巍

执行编辑 李 瑛

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京艺辉印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 15.25

字数: 372千字

2012年6月第1版

印数: 1-3 500册

2012年6月北京第1次印刷

著作权合同登记号 图字: 01-2011-8028号

ISBN 978-7-115-28201-9

定价: 49.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

# 版 权 声 明

Original English language edition, entitled *Beginning Android Tablet Programming* by Robbie Matthews, published by Apress , 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2011 by Robbie Matthews. Simplified Chinese-language edition copyright © 2012 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

谨以此书献给我的家人，如果没有你们的支持及偶尔的吹毛求疵，本书就不可能完成。

# 示例代码使用说明

本书提供了丰富的示例代码，这些代码可从 Apress 网站下载（在本书的页面中有一个下载链接）。

访问 Apress 网站，搜索本书，本书的网址为 [www.apress.com/9781430237839](http://www.apress.com/9781430237839)<sup>①</sup>。页面上有一个下载源代码的链接，这些源代码被压缩到了一个 zip 文件中。

## 安装并配置 Eclipse

除 Python 示例外，本书所有的编码示例都是使用 Eclipse 编写的。第 1 章将引导你安装并配置 Eclipse 环境，请阅读该章并按指示操作。

## 导入示例项目

要导入示例项目，请启动 Eclipse 并执行如下操作。

选择菜单 **File**►**Import**，在打开的对话框中，依次选择 **General** 和 **Existing Projects Into Workspace**，再单击 **Next** 按钮。然后，选中单选按钮 **Select Archive File**，浏览选择 **BA3TPSource.zip**。

此时将出现一个可选项目的列表，你可以选择感兴趣的项目，也可选择所有项目，然后单击 **Finish** 按钮。

## 工具库

对于工具库 **BA3TPUtils**，可能需要做额外的设置。这个库包含一组工具函数，本书的很多项目都要用到它们。**BA3TPUtils** 工具库需要知道到哪里去寻找 Android 开发文件，要指定该路径，可右击 **BA3TPUtils** 并选择 **Build Path** ► **Add External Archive**，再选择如下文件夹中的文件 **android.jar**：

自定义的<android-sdk 安装目录>\platforms\<自定义的平台>。

例如，在我的 Windows 计算机中，路径为 **C:\Program Files (x86)\Android\android-sdk\platforms\android-11**。

你可能遇到的另一个问题是，**BA3TPGps** 或 **BA3TPContacts2** 会抱怨缺少 **BA3TPUtils** 或找不到 **FileUtils**、**ListPicker** 或 **MessageBox**。这可能表明你没有先导入 **BA3TPUtils** 或这些项目找不到它。

---

<sup>①</sup> 也可以从图灵社区（[www.ituring.com.cn](http://www.ituring.com.cn)）本书相关页面下载。——编者注

解决方案是导入 BA3TPUtils，并在必要时告诉抱怨的项目到哪里去寻找它。具体做法如下：

右击抱怨的项目并选择 Build Path►Configure Build Path，在打开的对话框中选择 Projects 并单击 Add 按钮，再选择 BA3TPUtils 并单击 OK 按钮。

这样项目就能正常运行了。

## Python 示例

为减少输入量，下载包提供了第 4 章的所有 Python 示例文件。请在下载包中寻找文件夹 Python，其中的文件 examples.txt 指出了每个程序清单对应的源代码文件。

## 致谢

感谢 SL4A 的全体工作人员，如果没有你们，我就不可能编写本书；尤其要感谢 Damon Kohler，是他最初创立了 SL4A，本书介绍的很多技法都借鉴了他的创意；还要感谢 Paul Ferrill 将我引荐给 Apress 出版社，要不然我周末就闲着了！

# 目 录

|                           |    |                                  |    |
|---------------------------|----|----------------------------------|----|
| 第 1 章 Android 入门.....     | 1  | 2.3 Android 与 Linux.....         | 44 |
| 1.1 我的便携式编程之路.....        | 1  | 2.4 总结.....                      | 46 |
| 1.2 Android 横空出世.....     | 3  | 第 3 章 Android 平板电脑的功能.....       | 47 |
| 1.3 准备好计算机.....           | 4  | 3.1 大量传感器.....                   | 48 |
| 1.3.1 搭建开发环境.....         | 4  | 3.2 光线、相机和操作.....                | 55 |
| 1.3.2 创建模拟器实例.....        | 5  | 3.3 浏览网页.....                    | 57 |
| 1.3.3 设置路径.....           | 5  | 3.3.1 管理资产.....                  | 58 |
| 1.4 编写第一个 Android 程序..... | 6  | 3.3.2 WebView 的强大功能.....         | 59 |
| 1.5 如果你害怕 Java.....       | 9  | 3.4 你的个人通讯录：管理联系人.....           | 61 |
| 1.6 Java 简介.....          | 10 | 3.4.1 访问联系人.....                 | 62 |
| 1.6.1 Java 的基本结构.....     | 10 | 3.4.2 列表简介.....                  | 63 |
| 1.6.2 基本数据类型.....         | 11 | 3.4.3 访问联系人信息的其他方式.....          | 65 |
| 1.6.3 流程控制.....           | 12 | 3.5 个人经验分享：收发数据.....             | 66 |
| 1.6.4 对象.....             | 13 | 3.5.1 补充内容：选项菜单.....             | 67 |
| 1.6.5 构造函数、初始化块和重载.....   | 16 | 3.5.2 发送电子邮件.....                | 67 |
| 1.6.6 何时释放对象.....         | 18 | 3.6 喧闹的音画世界：管理多媒体文件.....         | 68 |
| 1.6.7 字符串处理.....          | 18 | 3.7 总结.....                      | 69 |
| 1.6.8 包.....              | 19 | 第 4 章 超越 Java：Python 编程.....     | 70 |
| 1.6.9 列表和映射.....          | 20 | 4.1 为何使用另一种语言.....               | 71 |
| 1.6.10 泛型.....            | 21 | 4.2 准备.....                      | 71 |
| 1.6.11 继承和接口.....         | 22 | 4.3 管理脚本.....                    | 71 |
| 1.6.12 注解.....            | 23 | 4.3.1 Help! I Need Somebody..... | 72 |
| 1.6.13 其他内容.....          | 24 | 4.3.2 Python 帮助.....             | 72 |
| 1.7 总结.....               | 24 | 4.3.3 Facade 是什么.....            | 73 |
| 第 2 章 Android 的工作原理.....  | 25 | 4.3.4 再谈意图.....                  | 73 |
| 2.1 Android 程序的基本结构.....  | 25 | 4.3.5 使用意图的不同方式.....             | 75 |
| 2.2 意图简介.....             | 38 | 4.3.6 用户交互.....                  | 76 |
| 2.2.1 意图过滤器.....          | 43 | 4.3.7 事件.....                    | 78 |
| 2.2.2 常见意图.....           | 44 | 4.3.8 多媒体文件.....                 | 80 |
| 2.2.3 结束活动.....           | 44 |                                  |    |

|                             |            |                           |            |
|-----------------------------|------------|---------------------------|------------|
| 4.3.9 控制手机 .....            | 83         | 6.1.7 MIME 类型 .....       | 118        |
| 4.3.10 我身处何方 .....          | 84         | 6.1.8 获悉文件的日期信息 .....     | 118        |
| 4.3.11 电池 .....             | 86         | 6.1.9 入乡随俗 .....          | 118        |
| 4.3.12 让设备处于唤醒状态 .....      | 87         | 6.1.10 设置首选项 .....        | 122        |
| 4.3.13 编辑技巧 .....           | 88         | 6.1.11 读取首选项 .....        | 124        |
| 4.3.14 联系人和电话号码 .....       | 89         | 6.1.12 排序技巧 .....         | 125        |
| 4.3.15 SQL .....            | 90         | 6.1.13 活动本身 .....         | 126        |
| 4.3.16 其他接口 .....           | 91         | 6.1.14 对用户选择列表项做出响应 ..... | 127        |
| 4.4 总结 .....                | 91         | 6.1.15 onCreate 方法 .....  | 127        |
| <b>第 5 章 多媒体播放器</b> .....   | <b>92</b>  | 6.1.16 另一个菜单 .....        | 128        |
| 5.1 片段 .....                | 92         | 6.1.17 响应上下文菜单 .....      | 128        |
| 5.1.1 片段是什么 .....           | 92         | 6.1.18 对话框 .....          | 129        |
| 5.1.2 研究示例 .....            | 93         | 6.1.19 一些清理工作 .....       | 131        |
| 5.1.3 操作栏 .....             | 96         | 6.1.20 改进 .....           | 132        |
| 5.1.4 片段管理器 .....           | 98         | 6.2 总结 .....              | 132        |
| 5.2 多媒体播放器 .....            | 98         | <b>第 7 章 联系人管理器</b> ..... | <b>133</b> |
| 5.2.1 以列表方式显示游标的内容 .....    | 100        | 7.1 完全不同的领域 .....         | 134        |
| 5.2.2 查询多媒体文件 .....         | 101        | 7.1.1 最终创建的应用程序 .....     | 134        |
| 5.2.3 响应单击 .....            | 101        | 7.1.2 指示牌剖析 .....         | 134        |
| 5.2.4 再谈操作栏 .....           | 102        | 7.1.3 获悉视图何时可用 .....      | 137        |
| 5.2.5 菜单 .....              | 102        | 7.1.4 改进空间 .....          | 138        |
| 5.2.6 获取多媒体文件的 URI .....    | 103        | 7.1.5 向后兼容性 .....         | 138        |
| 5.2.7 播放多媒体文件 .....         | 103        | 7.1.6 列表处理 .....          | 139        |
| 5.2.8 各种多媒体 .....           | 104        | 7.1.7 保存列表 .....          | 139        |
| 5.2.9 对话框 .....             | 104        | 7.1.8 重用库 .....           | 140        |
| 5.2.10 修改文件列表的游标 .....      | 106        | 7.1.9 导入和导出 .....         | 140        |
| 5.2.11 高级多媒体播放 .....        | 106        | 7.1.10 有关对话框的最后说明 .....   | 141        |
| 5.2.12 显示进度 .....           | 107        | 7.2 管理联系人 .....           | 142        |
| 5.2.13 控制进度 .....           | 108        | 7.2.1 Loader .....        | 142        |
| 5.3 总结 .....                | 108        | 7.2.2 启动 Loader .....     | 143        |
| <b>第 6 章 资源管理器</b> .....    | <b>109</b> | 7.2.3 Loader 的工作原理 .....  | 143        |
| 6.1 文件资源管理器剖析 .....         | 109        | 7.2.4 为何使用 Loader .....   | 144        |
| 6.1.1 使用 ListActivity ..... | 110        | 7.2.5 搜索栏 .....           | 144        |
| 6.1.2 资源 .....              | 111        | 7.2.6 联系人详细信息 .....       | 146        |
| 6.1.3 创建自定义列表适配器 .....      | 112        | 7.2.7 编辑数据 .....          | 148        |
| 6.1.4 填充 .....              | 113        | 7.2.8 添加电话号码 .....        | 150        |
| 6.1.5 填充列表 .....            | 114        | 7.2.9 修改电话号码类型 .....      | 151        |
| 6.1.6 获取有关文件的信息 .....       | 115        | 7.2.10 删除数据 .....         | 151        |
|                             |            | 7.3 总结 .....              | 152        |

|                        |     |                          |     |
|------------------------|-----|--------------------------|-----|
| 第 8 章 定位               | 153 | 9.3.1 一个盒子               | 187 |
| 8.1 位置管理器              | 153 | 9.3.2 其他进程间通信方式          | 192 |
| 8.1.1 位置提供器            | 154 | 9.4 总结                   | 193 |
| 8.1.2 BAT3PGps         | 154 | 第 10 章 提醒                | 194 |
| 8.1.3 一个操作位置管理器的类      | 155 | 10.1 闹钟                  | 194 |
| 8.1.4 位置               | 158 | 10.1.1 PendingIntent     | 195 |
| 8.2 应用程序本身             | 159 | 10.1.2 停止闹钟              | 195 |
| 8.2.1 指南针              | 159 | 10.1.3 匹配的 PendingIntent | 196 |
| 8.2.2 方位和距离            | 161 | 10.2 发送短信                | 196 |
| 8.3 到达目的地了吗            | 162 | 10.3 语音合成                | 198 |
| 8.3.1 叫醒服务             | 162 | 10.4 SQLite              | 200 |
| 8.3.2 服务               | 162 | 10.4.1 创建 SQLite 数据库     | 200 |
| 8.3.3 通知               | 163 | 10.4.2 获取数据库             | 201 |
| 8.3.4 接近提醒             | 166 | 10.4.3 使用数据库             | 201 |
| 8.3.5 “接近提醒”的优点和缺点     | 167 | 10.5 示例应用程序              | 202 |
| 8.4 目的地列表              | 168 | 10.5.1 使用该应用程序           | 204 |
| 8.4.1 这是什么地方           | 172 | 10.5.2 联系人               | 205 |
| 8.4.2 接下来怎么办           | 172 | 10.5.3 设置闹钟              | 205 |
| 8.5 总结                 | 173 | 10.5.4 闹钟列表              | 208 |
| 第 9 章 游戏编程             | 174 | 10.5.5 检查 TTS            | 211 |
| 9.1 线程                 | 174 | 10.5.6 改进该应用程序           | 212 |
| 9.1.1 Java 线程          | 175 | 10.6 总结                  | 212 |
| 9.1.2 Android 线程       | 178 | 第 11 章 其他                | 213 |
| 9.2 移动的点               | 178 | 11.1 定时器                 | 213 |
| 9.2.1 SurfaceView      | 179 | 11.2 在后台执行任务             | 217 |
| 9.2.2 使用 SurfaceHolder | 181 | 11.3 下载管理器               | 223 |
| 9.2.3 Handler          | 181 | 11.4 动画                  | 227 |
| 9.2.4 使用 Handler       | 182 | 11.4.1 定义动画              | 228 |
| 9.2.5 运行游戏             | 182 | 11.4.2 竟然明目张胆地插值         | 228 |
| 9.2.6 处理输入             | 183 | 11.4.3 加载动画              | 229 |
| 9.2.7 自定义字体            | 183 | 11.5 USB                 | 229 |
| 9.2.8 资产是什么            | 184 | 11.5.1 USB 编程            | 230 |
| 9.2.9 字体示例             | 185 | 11.5.2 响应 USB 事件         | 232 |
| 9.2.10 另一种创建菜单的方式      | 186 | 11.6 总结                  | 233 |
| 9.3 进入三维领域             | 187 |                          |     |

## 第 1 章

## Android 入门



欢迎进入 Android 平板电脑编程领域。本章将简要地介绍如何搭建编程环境，引领你开发自己的第一个 Android 程序并简要地介绍 Java。

但在此之前，先简单地说说我自己在手持计算领域的经历吧。

## 1.1 我的便携式编程之路

手持电脑的历史可能比你想的还要悠久。iPhone 不久前才登上历史舞台，并成了衡量时尚与否的标准，很多人都认为它开启了手持电脑的历史。

但从 20 世纪 70 年代末起，笔者就一直在使用类似的手持设备。

Sharp PC-1210 是我接触到的第一台可称之为手持电脑的设备，它装备了单行显示器和 QWERTY 键盘，支持 BASIC 编程。这是我一位表亲的，当时我很想有一台这样的设备尝尝鲜。

我不确定我想使用它来做什么——也许是在 D&D<sup>①</sup>中随机选择战役，但从那时起，我开始着迷于娇小可爱的手持设备，这种爱好一直保持到现在。

Apple Newton 于 1987 年面世，骄傲地宣称支持手写识别，我饶有兴致地关注着，也见证了它的没落，它成了广告宣传的牺牲品——宣传与实际功能严重不符。然而，这可能是我见到的第一款平板电脑。

大约在 1996 年，PalmPilot 面世了，我买了一部，它至今还在我家里。Palm 也支持一定的手写识别，但实现的功能远没有 Apple Newton 那么多，因此性能非常好。

我很喜欢 Palm。设计师从 Newton 的彻底失败中吸取了教训，将重点放在使用当时的技术可实现的功能上。相对于笔记本电脑，手持电脑必然是一种功能有限的设备。当今智能手机的计算能力超过了 NASA 将人类送上月球时拥有的计算能力，但在内存、处理能力、屏幕大小等方面，根本不可与台式机同日而语。鉴于智能手机的特性，也许永远都将如此。

因此，设计 Palm OS 时，并未试图使其全能，而是假定有 PalmPilot 的人都有计算机，因此将需要大量处理能力和存储空间的功能都留给了计算机，并重点优化用户实际使用的功能。PalmPilot 还带来了新名词“个人数字助理”（PDA），它表示所有类似于 PalmPilot 的设备。

<sup>①</sup> D&D 是世界上第一款商业化角色扮演游戏。我以前是技术狂，现在仍然是，并为此感到自豪！

进行移动设备编程时，需要牢记的重点是，确保简洁、避免面面俱到并将重点放在可用性上。请注意这一点，后面的小测验将涉及这方面。

随后出现了 Psion 等设备，我购买了几部。Psion 很好，它配置的键盘很小，但功能齐备，几乎可以盲打。Psion 使用一种独特而简单的编程语言，运行的是 Symbian 操作系统。有趣的是，虽然 Psion 早已被淘汰，Symbian OS 却活了下来，还在 Nokia 手机中活得很滋润。

随后，我购买了另一款手持设备——LifeDrive，它是一种 PDA，几乎与 iPhone 一样薄。它支持 Wi-Fi 和蓝牙，采用触摸屏，自带丰富的应用程序，配置 4GB 存储器。事实上，除了没有内置电话，它与现代智能手机几乎差不多，而面世时间却早了 5 年。

移动设备的另一个分支是手机。

20 世纪 80 年代末，公司在我的车上安装了车载电话，旨在将无法联系到我的时间限定在 5 分钟内。

20 世纪 90 年代初，我购买了第一部手机。从此以后，我每隔几年就换部新的，而手机也越来越智能，屏幕越来越大，功能越来越多：地址簿、日历、计算器、游戏等。

再后面的情况你都知道了。从那以后，手机和 PDA 一直在相互借鉴，直到智能手机面世，它们才融为一体。

当然，从 20 世纪中叶开始，就有其他常用的小型移动设备；如果算上相机，移动设备的历史就更长了（如果将手表也算上，其历史有几个世纪）。

晶体管收音机、计算器、相机、手表、随身听、MP3 播放器、DVD 播放器、GPS 导航器、电子图书阅读器，它们的功能逐渐被融入到 PDA 或手机中，而现在 PDA 和手机也融为一体了。

同样，这种趋势也不像很多人认为的那样是新出现的。十年前，Kyocera 6035（一款使用 Palm OS 的手机）就将手机和 PDA 融为一体了，Treo、Blackberry 和各种 Windows CE 平台亦如此。

但这些设备的用户群主要是时髦的年轻主管和坚定的技术狂。归根结底，这些设备太贵了，功能却无法与专用设备媲美。

随着 Apple 发布 iPhone，一切都融合在了一起。智能手机的功能、价位和市场推广使其成了必备的大众热销品。

还有一类移动计算设备没有说到，那就是笔记本电脑。我使用的第一台笔记本是 KayPro II，那是 1982 年的事了。这款笔记本电脑采用 CP/M Z80 处理器，配置了两个软驱，自带阴极射线管显示器。它不是笔记本电脑，而是“手提”电脑，因为它有把手，只要你足够强壮，就可以随身携带它。

我使用的第二款笔记本电脑是 Commodore-SX64，它装在盒子内，备有软驱、5 英寸显示器和把手，在我看来非常轻巧。

随后出现了真正的笔记本电脑，且功能日益强大、体积越来越小、电池续航时间越来越长。大约五六年前，出现了采用触摸屏的笔记本电脑，它们虽然漂亮，却并未引起轰动。

几年前，我购买了第一台上网本，这是一款采用固态驱动器的超小型笔记本电脑。我一直将其作为移动设备使用，直到不久前购买了 HTC Desire Android 手机。

随后，Apple 发布了 iPad，它基本上是 iPhone 的放大版，配置了更大的屏幕，填补了笔记本电脑和智能手机之间的市场空白。

这将移动设备分成了相互补充的两类：智能手机和平板电脑，前者可装入口袋，后者也算轻便、易于携带，而且其屏幕和外围设备让你能够完成重要的工作。

## 1.2 Android 横空出世

从技术上说，iPhone 和 iPad 无可挑剔，但它们的缺点是不支持开源。事实上，很多人都对 Apple 智能手机不开源一事颇有微词。

另外，正如前面说过的，智能手机的历史并不短，且至少有 3 款移动操作系统得到了市场的认可，因此 iPhone 不久就面临着竞争。

2005 年，Google 收购了创建于 2003 年的 Android 公司。iPhone 发布仅两个月后，即 2007 年 11 月，开放手机联盟（Open Handset Alliance）成立了，Android 平台也于当日发布。开放手机联盟由众多公司组成，成员包括 Google、摩托罗拉、HTC 以及其他许多手机制造商和运营商。

Android 基于 Linux 内核，基本上是开源的（有些例外）。Android 应用主要是使用 Java 编写的。Android 版 Java 运行在专门针对手持设备进行了优化的 Dalvik 虚拟机上。

时间飞逝，到 2011 年（我称为现在），出现了一些 Android 平板电脑，但仅就操作系统而言，它们基本上只是放大版的移动电话。直到 Android 3.0（Honeycomb）发布后，Android 才能充分利用平板电脑丰富的功能。

这正是笔者编写本书的意义所在。

言归正传，阅读完本章后，你将能够运行自己的第一个 Android 程序。

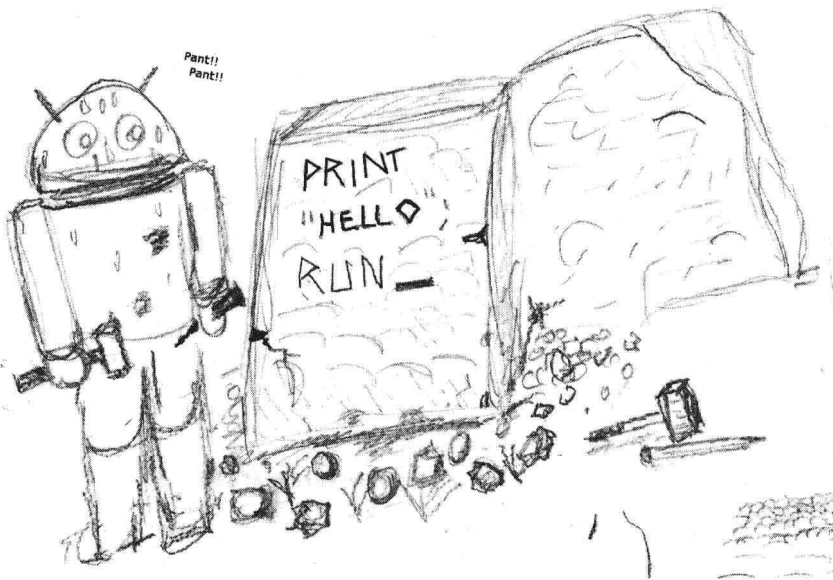


图 1-1 可用于编程的平板电脑曾如此遥不可及……

## 1.3 准备好计算机

要进行 Android 平板电脑编程，需要搭建编程环境，这包括下载 JDK（Java Development Kit，Java 开发包）、Eclipse IDE（Integrated Development Environment，集成开发环境）和 Android SDK（Software Development Kit，软件开发包）。

Android SDK 虽然有一些本机代码扩展，但主要是使用 Java 编写的。如果你不喜欢 Java，也有其他选择。实际上，如果你选择使用 Python 之类的语言，将更快、更容易上手。本书后面专辟了一章（第 4 章）来讨论 Python，如果只想快速上手，使用 Python 可能更合适。它还有一个优势，那就是可直接在 Android 设备中编程，而无需将其连接到计算机。

然而，Android 主要是用 Java 编写的，因此大部分文档都是针对 Java 的。使用 Java 可最大限度地挖掘设备的潜力，因此即使你最终在其他地方进行应用程序开发，也有必要了解 Java。

### 1.3.1 搭建开发环境

下面是简化的流程。

(1) 访问 <http://developer.android.com/sdk/installing.html>。

(2) 按指示操作。

(3) 开始编程！

实际上，就这么简单。而且，鉴于 Android 更新频繁，绝对应该采用这种流程。

然而，该流程虽然很简单，但对新手来说，还是有些陷阱和圈套，下面将详细介绍。

(1) 确保计算机满足需求，有关最新需求，请参阅 <http://developer.android.com/sdk/requirements.html>。Android SDK 有 Windows、Linux 和 Mac 版本，这几乎涵盖了当今所有的个人电脑，你应该能够找到适合自己的版本。

(2) 确保安装了最新的 JDK，当前可从 [www.oracle.com/technetwork/java/javase/downloads](http://www.oracle.com/technetwork/java/javase/downloads) 下载它。

(3) 从 [www.eclipse.org/downloads](http://www.eclipse.org/downloads) 下载 Eclipse 并安装它。

(4) 从 <http://developer.android.com/sdk/index.html> 下载 Android SDK 并安装它。

(5) 启动 Eclipse。如果你按正确的顺序做，Eclipse 会自动找到 JDK，否则可能就需要告诉 Eclipse 到哪里去寻找。

(6) 安装 ADT（Android Developer Tools，Android 开发者工具）。

(a) 在 Eclipse 中，选择菜单 Help ► Install New Software ► Add。

(b) 在文本框 Name（名称）中，输入 ADT Plugin。

(c) 在文本框 Location（位置）中，输入 <https://dl-ssl.google.com/android/eclipse/>。

(d) 单击 OK（确定）按钮。

(e) 这将打开 Available Software 对话框。选择 Developer Tools，不断单击 Next 按钮，直到出现 Finish 按钮，再单击它。

(f) 让系统加载 ADT 并重新启动 Eclipse。

(7) 告诉 Eclipse 到哪里寻找 Android SDK。

(a) 选择菜单 Window ► Preferences, 再选择 Android。

(b) 指定 Android SDK 安装目录(即 c:\program files\android\android-sdk 或/Developer/ android-sdk-mac\_x86)。

(c) 依次单击 Apply 和 OK 按钮。

(8) 加载平台。

(a) 选择菜单 Window ► Android SDK and AVD Manager。

(b) 单击 Available Packages。没有理由不下载 Android Repository 下列出的所有内容,但至少应下载 Android SDK Platform-Tools 以及 Android Platform 3.0 或 Android Platform 3.1。另外,还应下载文档。

就这么简单,现在可以开始编程了!但在此之前,还有必要完成另外两项工作。

### 1.3.2 创建模拟器实例

如果有 Android 设备,可以使用 USB 连接线将其连接到计算机,并使用它来测试程序,这样速度会更快。但有时没有这样的设备。在 Eclipse 中,选择菜单 Window ► Android SDK and AVD Manager,再选择 Virtual Devices 并新建一个模拟器。选择 API level 11,并指定一个有意义的名称,如 Honeycomb。指定较大的外部存储空间:默认设置为 16MB,这太小了,我通常起码将其设置为 512 MB。

---

**警告** Android 模拟器的速度非常慢,而 Honeycomb 模拟器更慢。务必要有耐心,在模拟器启动期间去喝杯咖啡,让它一直开着,直到使用完毕。

---

下面是一些提高模拟器速度的技巧。

- ❑ 模拟器是个单线程进程,不会同时使用多个 CPU。在规格相同的情况下,如果系统装备的是双核处理器,而不是四核处理器,模拟器的性能将更佳,因为这样模拟器可使用的 CPU 处理能力更高。
- ❑ 也可将 Dynamic Ram Size (动态内存)增大到 1024 MB,但必须确保计算机的物理内存超过 1024MB,否则将适得其反。

### 1.3.3 设置路径

Android SDK 自带了一些功能强大的工具,其中你很可能用到的是 adb,这是一个命令行工具。要使用它,必须将 platform-tools 加入到环境变量 PATH 中,具体情况如下。

- ❑ Windows: PATH=%PATH%;C:\Program Files\Android\Android-SDK\platform-tools。
- ❑ Linux: export PATH=\$PATH:~/android-sdk/platform-tools。
- ❑ Mac: export PATH=\$PATH:~/Developer/android-sdk-mac\_x86/platform-tools。

具体路径取决于 Android SDK 的安装位置。

## 1.4 编写第一个 Android 程序

下面进入正题，开发第一个 Android 程序。

启动 Eclipse，并创建一个 Android 的 Hello World 程序。这很简单。首先，选择菜单 File►New►Android Project，如图 1-2 所示。

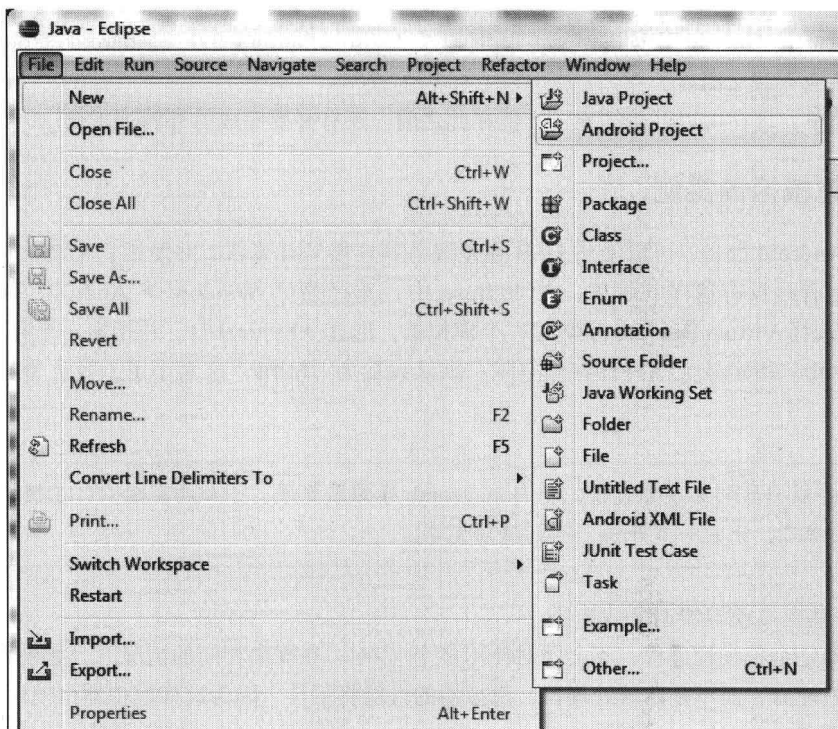


图 1-2 在 Eclipse 中新建 Android 项目

**注意** 在 Mac 中，选择菜单 File ► New ► Project ► Android Project。

接下来，给项目指定名称（我指定的名称为 BA3TPHello）并选择目标环境。

这可能是 Android 3.0（Honeycomb），但就这里而言，任何目标环境都行。另外，以后可修改目标环境。

你需要指定项目名、应用程序名和包名。

如果你查看我使用的设置（如图 1-3 所示），将发现我没有取消选择 Create Activity，这样将创建一个默认活动（activity）。

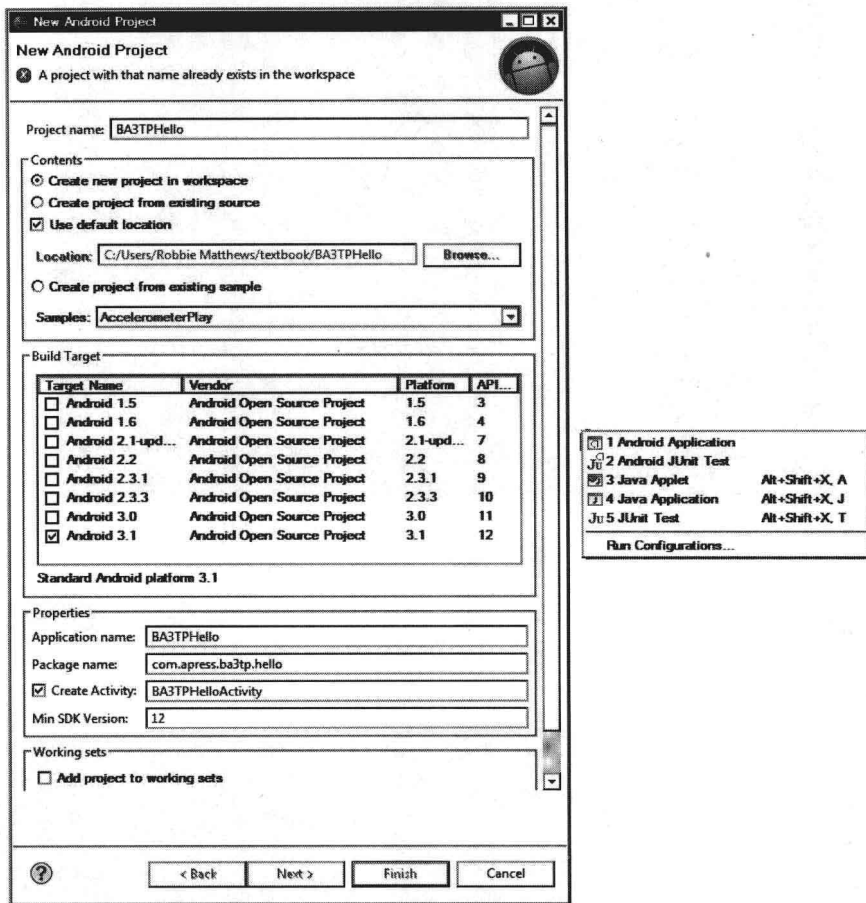


图 1-3 Android 应用程序设置

稍后将更详细地介绍活动。这里我们先让 Android SDK 为我们完成所有的工作。

确保创建了模拟器或将 Android 设备连接到了计算机。单击 Finish 按钮，再单击工具栏按钮 Run。如果 Eclipse 不知道以什么方式运行该程序，请右击刚创建的项目，再选择 Run As，并在 Eclipse 询问时指定要以 Android 应用程序的方式运行。

结果如图 1-4 所示。

你可能会问，应用程序怎么知道我要显示 Hello World 呢？答案是，应用程序会自动在屏幕布局中添加 Hello World。

这可能不那么令人满意。在包资源管理器（Package Navigator）中，双击目录 res/layout 下的文件 main.xml，这将打开布局编辑器，如图 1-5 所示。