

SAMS
Teach
Yourself

- 全球销量逾百万册的系列图书
- 连续十余年打造的经典品牌
- 直观、循序渐进的学习教程
- 掌握关键知识的最佳起点
- “Read Less, Do More”（精读多练）的教学理念
- 以示例引导读者完成最常见的任务

每章内容针对初学者精心设计，**1**小时轻松阅读学习，
24小时彻底掌握关键知识

每章**案例与练习题**助你轻松完成常见任务，
通过**实践**提高应用技能，巩固所学知识

Android

游戏编程

入门经典

[美] Jonathan S. Harbour 著
宋松 王小飞 译

人民邮电出版社
POSTS & TELECOM PRESS

Android

游戏编程

入门经典

[美] Jonathan S. Harbour 著
宋松 王小飞 译



人民邮电出版社
北京

图书在版编目 (C I P) 数据

Android游戏编程入门经典 / (美) 哈伯
(Harbour, J. S.) 著 ; 宋松, 王小飞译. — 北京 : 人民
邮电出版社, 2013. 7

ISBN 978-7-115-31954-8

I. ①A… II. ①哈… ②宋… ③王… III. ①移动电
话机—游戏程序—程序设计 IV. ①TN929.53②TP311.5

中国版本图书馆CIP数据核字(2013)第099914号

版权声明

Jonathan Harbour: Sams Teach Yourself Android Game Programming in 24 Hours

ISBN: 0672336049

Copyright © 2013 by Pearson Education, Inc.

Authorized translation from the English languages edition published by Pearson Education, Inc.

All rights reserved.

本书中文简体字版由美国 Pearson 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以何种方式复制或抄袭。

版权所有, 侵权必究。



-
- ◆ 著 [美] Jonathan S. Harbour
 - 译 宋松 王小飞
 - 责任编辑 傅道坤
 - 责任印制 程彦红 焦志炜

 - ◆ 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>
北京铭成印刷有限公司印刷

 - ◆ 开本: 787×1092 1/16
印张: 20
字数: 490千字 2013年7月第1版
印数: 1-3000册 2013年7月北京第1次印刷
著作权合同登记号 图字: 01-2012-5020号

定价: 49.00元

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154

目 录

第 1 部分 简介

第 1 章 Android 4 简介	3
1.1 Android 4 简介	3
1.2 Android SDK 简介	6
1.3 Android NDK 简介	7
1.4 Android 开发系统要求	7
1.5 Android 平台的历史	7
1.6 Android 硬件规格	8
1.7 小结	10
1.8 问与答	10
1.9 测验	10
1.9.1 问题	10
1.9.2 答案	10
1.9.3 作业	10
第 2 章 安装开发工具	11
2.1 安装 JDK	11
2.2 下载 NetBeans 软件包	12
2.3 安装软件包	13
2.4 安装 Android SDK	14
2.5 下载 SDK	14
2.6 安装 SDK	15
2.7 运行 Android SDK Manager	16
2.8 安装 Eclipse 的 ADT 插件	18
2.9 小结	19
2.10 问与答	19
2.11 测验	20
2.11.1 问题	20
2.11.2 答案	20
2.11.3 作业	20
第 3 章 使用 Android SDK 配置 NetBeans 和 Eclipse	21
3.1 创建 Android 模拟器设备	21
3.2 创建新模拟器设备	22
3.3 运行模拟器	23
3.4 将 Android SDK 添加到 NetBeans 中	24
3.5 为 Eclipse 添加 Android SDK 支持	27
3.5.1 在 Eclipse 中创建新 Android 项目	27
3.5.2 选择 Android 构建目标	29
3.6 小结	30
3.7 问与答	30
3.8 测验	30
3.8.1 问题	30
3.8.2 答案	31
3.8.3 作业	31
第 4 章 创建第一个 Android 程序	32
4.1 创建新的 Android 项目	32
4.1.1 配置 NBAndroid 插件	34

4.1.2 完成新项目设置	35	6.1.2 绘制线条	64
4.2 构建新项目	35	6.1.3 绘制方框	65
4.2.1 打开输出窗口	36	6.1.4 绘制圆角矩形	65
4.2.2 在模拟器中运行程序	37	6.1.5 绘制三角形	66
4.3 编辑 Hello, Android! 程序	41	6.1.6 更改样式: 描边和填充	66
4.4 对比模拟器和 Android 设备	42	6.1.7 基本图形演示	66
4.4.1 AVD 限制	43	6.2 绘制文本	69
4.4.2 安装 Android Debug Bridge Device Driver	44	6.3 编写 Javadoc 的代码	72
4.4.3 在 Android 设备上运行代码	46	6.4 Android 屏幕密度和分辨率	73
4.5 小结	48	6.5 小结	77
4.6 问与答	49	6.6 问与答	77
4.7 测验	49	6.7 测验	77
4.7.1 问题	49	6.7.1 问题	77
4.7.2 答案	49	6.7.2 答案	77
4.7.3 作业	49	6.7.3 作业	77
 		第 7 章 加载和绘制图像	
第 2 部分 Android 硬件		7.1 双缓冲绘图	78
 		7.1.1 创建内存位图	78
第 5 章 开始使用图形		7.1.2 创建绘图画布	79
5.1 了解 Activity 类	53	7.1.3 使用后台缓冲	79
5.1.1 程序生命周期	54	7.2 加载位图文件	81
5.1.2 可视生命周期	55	7.2.1 添加位图资产	82
5.1.3 前台生命周期	55	7.2.2 使用 AssetManager	83
5.2 测试 Activity 状态	55	7.2.3 使用 InputStream	84
5.2.1 创建新项目	55	7.2.4 使用 BitmapFactory	84
5.2.2 源代码	58	7.2.5 关闭输入流	84
5.3 世界上最简单的 Android 图形演示	59	7.2.6 完整的位图加载代码	84
5.3.1 取消选中第一个图形演示	61	7.3 绘制位图	85
5.3.2 画布和视图简介	62	7.3.1 为透明度创建 alpha 通道	86
5.4 小结	63	7.3.2 完成的示例	88
5.5 问与答	63	7.4 小结	89
5.6 测验	63	7.5 问与答	89
5.6.1 问题	63	7.6 测验	90
5.6.2 答案	63	7.6.1 问题	90
5.6.3 作业	63	7.6.2 答案	90
 		7.6.3 作业	90
第 6 章 绘制基本形状和文本		第 8 章 使用循环赋予游戏活力	
6.1 绘制基本矢量形状	64	8.1 创建线程游戏循环	91
6.1.1 绘制圆形	64	8.1.1 查看非线性代码	91
		8.1.2 编写线程代码	92
		8.2 不使用 onDraw() 进行绘制	93

8.3	Runnable Animation Demo	95	11.1.1	初始化线性加速度传 感器	124
8.3.1	为行走的角色制作动画	95	11.1.2	关于暂停和恢复	124
8.3.2	项目源代码	96	11.1.3	读取传感器	124
8.4	小结	100	11.1.4	线性加速度演示	125
8.5	问与答	100	11.2	访问近距离传感器	129
8.6	测验	100	11.3	小结	130
8.6.1	问题	100	11.4	问与答	130
8.6.2	答案	100	11.5	测验	130
8.6.3	作业	101	11.5.1	问题	130
第 9 章	多点触摸用户输入	102	11.5.2	答案	131
9.1	单点触摸输入	102	11.5.3	作业	131
9.1.1	监听触摸事件	102	第 12 章	使用重力传感器和压力传 感器	132
9.1.2	单点触摸输入演示	103	12.1	使用重力传感器	132
9.2	多点触摸输入	106	12.1.1	初始化重力传感器	132
9.2.1	简单示例	107	12.1.2	读取重力传感器	133
9.2.2	封装多点触摸输入	108	12.1.3	测试重力传感器	134
9.2.3	多点触摸演示	108	12.2	使用压力传感器	138
9.3	小结	111	12.2.1	初始化压力传感器	138
9.4	问与答	111	12.2.2	读取压力传感器	139
9.5	测验	112	12.3	小结	139
9.5.1	问题	112	12.4	问与答	139
9.5.2	答案	112	12.5	测验	139
9.5.3	作业	112	12.5.1	问题	139
第 10 章	使用加速度计	113	12.5.2	答案	140
10.1	Android 传感器	113	12.5.3	作业	140
10.1.1	访问传感器	114	第 13 章	创建自己的“手机传感器”	141
10.1.2	禁用屏幕方向改变	115	13.1	封装 Android 传感器	141
10.1.3	加速度计初始化	115	13.1.1	BaseSensor	142
10.1.4	加速度计运动	116	13.1.2	加速度计	142
10.1.5	获取可用传感器列表	117	13.1.3	线性加速度	143
10.1.6	完整示例	118	13.1.4	近距离传感器	143
10.2	小结	121	13.1.5	重力传感器	143
10.3	问与答	121	13.1.6	压力传感器	144
10.4	测验	122	13.1.7	陀螺仪传感器	144
10.4.1	问题	122	13.1.8	指南针	144
10.4.2	答案	122	13.1.9	光检测器	145
10.4.3	作业	122	13.2	创建手机传感器项目	145
第 11 章	使用线性加速度传感器和 近距离传感器	123	13.2.1	包和导入	146
11.1	访问线性加速度传感器	123	13.2.2	主要类	146
			13.2.3	DrawView	147

13.2.4	暂停和恢复	147
13.2.5	线程进程	147
13.2.6	创建面板	148
13.2.7	更新传感器	149
13.2.8	绘制传感器面板	149
13.2.9	辅助方法	150
13.2.10	SensorPanel 类	150
13.2.11	输出文本行	151
13.2.12	Sensors 类	152
13.2.13	捕捉传感器事件	153
13.2.14	未使用的准确性事件	154
13.2.15	暂停和恢复应用程序	154
13.2.16	BaseSensor 类	156
13.2.17	各种传感器类	157
13.3	小结	158
13.4	问与答	158
13.5	测验	158
13.5.1	问题	158
13.5.2	答案	159
13.5.3	作业	159
第 14 章	使用音频系统	160
14.1	使用 MediaPlayer 播放音频	160
14.1.1	受支持的音频格式	160
14.1.2	初始化 MediaPlayer	161
14.1.3	将音频文件添加到项 目中	161
14.1.4	播放音频剪辑	163
14.2	使用 SoundPool 播放音频	163
14.2.1	初始化 SoundPool	164
14.2.2	加载音频资源	164
14.2.3	播放音频资源	164
14.2.4	Audio Demo 程序	165
14.3	小结	166
14.4	问与答	166
14.5	测验	166
14.5.1	问题	166
14.5.2	答案	167
14.5.3	作业	167
 第 3 部分 Android Gameplay		
第 15 章	构建 Android 游戏引擎	171
15.1	设计 Android 游戏引擎	171
15.1.1	设计目标	172
15.1.2	引擎组件	172
15.2	创建 Android 库项目	174
15.3	编写核心引擎类	178
15.3.1	Engine 类	178
15.3.2	Timer 类	185
15.3.3	TextPrinter 类	186
15.3.4	Texture 类	187
15.4	引擎测试演示项目	188
15.4.1	引擎演示源代码	190
15.4.2	记录引擎演示的日志	192
15.5	小结	193
15.6	问与答	193
15.7	测验	193
15.7.1	问题	193
15.7.2	答案	193
15.7.3	作业	193
第 16 章	创建 Sprite/Actor 类	194
16.1	静态子画面为道具	194
16.2	动态子画面为角色	195
16.3	封装基本子画面功能	196
16.3.1	拟定计划和设计	197
16.3.2	新 Sprite 类的简介	197
16.3.3	回顾 Texture	198
16.4	测试 Sprite 类	199
16.4.1	子画面演示源代码	200
16.4.2	测试 Sprite 类	202
16.5	小结	203
16.6	问与答	203
16.7	测验	203
16.7.1	问题	203
16.7.2	答案	203
16.7.3	作业	203
第 17 章	使用子画面表单/地图集制 作帧动画	204
17.1	使用一张图像制作动画	204
17.1.1	排列动画帧	204
17.1.2	从连续图像绘制帧	205
17.2	使用子画面表单(纹理地图集)	

制作动画	206	19.7.1 问题	244
17.3 动画演示	207	19.7.2 答案	244
17.4 小结	211	19.7.3 作业	245
17.5 问与答	211	第 20 章 实体分组	246
17.6 测验	211	20.1 实体分组	246
17.6.1 问题	211	20.1.1 引擎改进	246
17.6.2 答案	212	20.1.2 跳动动画更新	249
17.6.3 作业	212	20.1.3 扭曲行为更新	250
第 18 章 高级多动画技术	213	20.1.4 围栏行为	251
18.1 创建动画系统	213	20.1.5 实体分组演示	251
18.1.1 编写动画基类	213	20.2 小结	254
18.1.2 Sprite 类改进	214	20.3 问与答	255
18.1.3 alpha 动画	218	20.4 测验	255
18.1.4 帧动画	219	20.4.1 问题	255
18.1.5 旋转动画	220	20.4.2 答案	255
18.1.6 跳动动画	221	20.4.3 作业	255
18.1.7 圆周运动动画	222	第 21 章 碰撞检测	256
18.2 动画系统演示	222	21.1 碰撞检测技术	256
18.3 小结	226	21.1.1 边界矩形 (框碰撞)	256
18.4 问与答	226	21.1.2 边界圆形 (径向 碰撞)	257
18.5 测验	226	21.2 演示碰撞	259
18.5.1 问题	226	21.2.1 引擎改进	259
18.5.2 答案	227	21.2.2 子画面改进	262
18.5.3 作业	227	21.2.3 碰撞演示源代码	264
第 19 章 使用矩阵变换处理子画面	228	21.3 小结	267
19.1 矩阵平移	228	21.4 问与答	267
19.1.1 坐标系	229	21.5 测验	267
19.1.2 矩阵平移选项	231	21.5.1 问题	267
19.2 矩阵旋转	232	21.5.2 答案	268
19.3 矩阵缩放	233	21.5.3 作业	268
19.4 矩阵变换演示	233	第 22 章 使用线性速度实现实际运动	269
19.4.1 结合使用三个矩阵	234	22.1 从一个方向计算速度	269
19.4.2 将帧渲染到划痕位图	234	22.2 在移动方向上“定位”子 画面	271
19.4.3 获得屏幕分辨率 (Precanvas)	235	22.3 改进引擎	273
19.4.4 “扭曲”行为	236	22.3.1 Sprite 类的变化	273
19.4.5 更新的 Sprite 类	237	22.3.2 改进动画类	275
19.4.6 子画面变换演示	241	22.3.3 对速度行为进行分类	276
19.5 小结	244	22.3.4 更新圆周运动类	277
19.6 问与答	244		
19.7 测验	244		

22.3.5	更新围栏行为	278	23.5	测验	295
22.3.6	更新扭曲行为	279	23.5.1	问题	295
22.3.7	引擎改进	280	23.5.2	答案	295
22.4	小结	284	23.5.3	作业	295
22.5	问与答	284	第 24 章	球类游戏	296
22.6	测验	284	24.1	创建球类游戏	296
22.6.1	问题	284	24.1.1	自动移动球	297
22.6.2	答案	285	24.1.2	自动限制球拍	298
22.6.3	作业	285	24.1.3	球类游戏源代码	299
第 23 章	滚动背景	286	24.2	小结	303
23.1	背景滚动概述	286	24.3	问与答	303
23.1.1	翻转滚动矩形	286	24.4	测验	303
23.1.2	创建滚动纹理	287	24.4.1	问题	303
23.2	射击游戏	288	24.4.2	答案	303
23.3	小结	294	24.4.3	作业	304
23.4	问与答	294			

第1部分

简介

第1章 Android 4 简介

第2章 安装开发工具

第3章 使用 Android SDK 配置 NetBeans 和 Eclipse

第4章 创建第一个 Android 程序

第 1 章

Android 4 简介

本章介绍如下内容：

- Android 4 的新功能；
- Android 平台的历史；
- Android 硬件规格；
- 开发系统要求。

本书前 4 章都是非常实用的入门教程，介绍使用两种开发环境（NetBeans 和 Eclipse）设置 Android SDK 的步骤。我们将介绍配置 Eclipse 或 NetBeans（具体取决于您的选择）进行 Android 开发的步骤。

本章介绍 Google 的新智能手机平台 Android 4 上的游戏编程。在编写本书时，有两个版本可用：4.0（Ice Cream Sandwich，冰淇淋三明治）和 4.1（Jelly Bean，果冻豆），预计明年将推出更多版本。Android 4 是一个全新的平台，而不仅仅是 3.2 的升级版。令人兴奋的是，开发 Android 游戏拥有出色的软件开发工具包和开发工具。本章将介绍设备、功能集和硬件规格，首先将详细介绍开发工具。本章假设您已经会使用 Android 手机并且想要尽快开始编写代码。很快您就可以编写 Android 代码并正常运行这些代码了。本章旨在帮助您从众多的 Android 设备中选择一种适当的游戏开发设备。我们将查看最常见的手机和平板电脑，并比较其硬件规格，以了解大多数设备将运行的游戏类型。

1.1 Android 4 简介

最令人惊奇的是，仅用了 3 年时间，Android 就发展成为智能手机市场中强有力的竞争者。从提出构想到发布版本仅用了 3 年时间，这是一项了不起的壮举。之所以如此是有原因的：Android 是以 Linux core 3.0 为基础构建的。是的，Android 实际上是 Linux 操作系统的衍生物。

鉴于 Android 比 Apple 晚两年进入市场，因此，可以说 Android 在与 Apple 的明星产品 iPhone 的竞争中获得了成功。Microsoft 正在利用其 Windows Phone 平台从 Google 和 Apple 那里抢夺市场份额，在过去一年里，Windows Phone 平台也经历了类似的快速发展。在发展过程中，Google 的 Android SDK 和操作系统开发人员往往依靠创造力来克服通常需要更多时间的挑战（见图 1.1）。

图 1.1

Android 4.0, Ice Cream Sandwich。图
片来自 Google

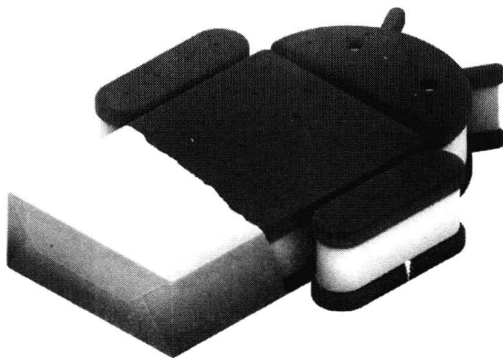
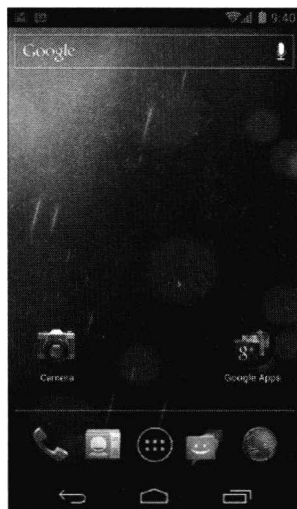


图 1.2 显示了 Android 4 智能手机直立时的主屏幕。注意界面的简单布局。底部是 3 个虚拟按键：返回、主页和最近任务。返回按键是通用的返回按键。主页按键返回到该屏幕。

图 1.2

Android 4 主屏幕。图
片来自 Google



最近任务按键显示如图 1.3 所示的屏幕，显示正在运行的应用程序（包括游戏）以进行快速任务切换（也称为“最近使用的应用程序”）。顶部是 Google 的搜索字段，带有语音识别功能。

在了解 Java SE 7、源代码项目、编译器配置、Android Market 和游戏开发者感兴趣的其他内容之前，我们需要首先了解此新平台。Android 是智能手机设备的操作系统。Android 也是非专有的，这意味着从 Google 获得操作系统许可是免费的。Android 的源代码是开源的。相反，Microsoft 和 Apple 则严格控制其操作系统（分别是 Windows Mobile 和 iOS）的源代码。这些公司严格控制其操作系统的发布。

Microsoft 向制造商提供 Windows Mobile OS（也称为 Windows Phone）的许可，然后制造商在其设备上使用该操作系统。这是一种中级许可方式：不像 Google 那样宽松，也不像 Apple 那样严格。

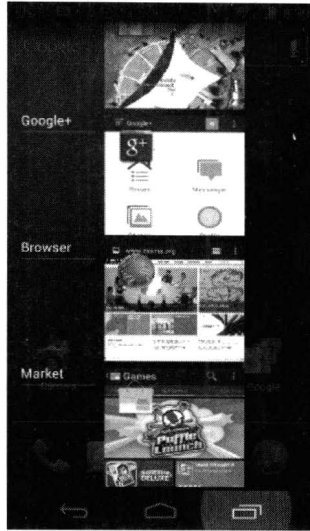


图 1.3

使用任务（最近使用的应用程序）按
键浏览正在运行的应用程序。图片
来自 Google

Apple 对其操作系统和硬件实施专有控制，对它们控制得非常好。尽管 Apple iOS 设备不像 Android 那样提供大量服务并具有通用性，但是他们提供了更加完善且一致的体验，因为是同一家公司设计的硬件和软件。

图 1.4 显示了 Android 4 智能手机上安装的应用程序。



图 1.4

Android 4 的 Apps
(应用程序) 屏幕。
图片来自 Google

开源是要付出代价的。发布操作系统的源代码并不意味着 Android 将占据市场的主导地位。相反，Android 游戏开发者面临的一个挑战是：大量非授权硬件制造商未经 Google 的许可自行构建 Android 手机和平板电脑。这些制造商按照指南或“硬件 API”构建其设备，以确保设备的广泛兼容性，但是也有例外情况。一些 Android 手机和平板电脑使用不同的 CPU，因此它们可能无法运行一些应用程序和游戏！

提示：如果您打算将智能手机游戏迁移到其他平台，请参阅我的另一本书籍，名为 *Sams Teach Yourself Windows Phone 7 Game Programming in 24 Hours*。

**By the
Way**

就在多年以前，拥有一台 PC 就可以完成您的工作：编程、软件工程、计算机辅助设计

(CAD)、文字处理、会计。即使是在 20 世纪 80 年代，每个员工的桌子上都有一台 PC 也是很少见的，家中拥有 PC 的情况就更少见。许多孩子可能有一个任天堂游戏机 (Nintendo Entertainment System, NES)、世嘉八位机 (Sega Master System、SMS) 或者是更老的雅达利 2600 (Atari 2600)，这些游戏机使用的是卡盒式游戏。比这些电子游戏机更进一步的是当时真正的 PC，比如 Apple II、Commodore 64、Amiga、Atari 400/800、Atari ST。

那时没有电脑爱好者在家里使用 IBM PC。与其他许多用户友好的操作系统相比，MS-DOS 是一种可怕的操作系统。如果您想编程，则通常倾向于选择消费者 PC，而不是面向企业的 IBM PC。那时，Apple Macintosh 非常昂贵，普通的孩子更喜欢 Apple II，但在 20 世纪 80 年代 Apple II 是 Mac 的第一代产品（经过了几次重新设计，它才成为现代的苹果操作系统）。

而今天的世界已经跟那时候大不相同了。即使我们忽略如今功能强大的计算机，但是看看所有的手持型游戏机，任天堂的 DS 系列和索尼的 PSP 系列是手持型游戏机的两个主要竞争对手，它们可以完成其上一代产品（任天堂 Wii 游戏机和索尼 PS3）能够做的所有事情，包括联机模式。这些情况随处可见。走进商店或购物中心时，您肯定会看到一些孩子随身带着某种便携式游戏机，更不用说智能手机了。此外，不仅仅是孩子，成人也有自己的玩具，比如 iPhone、iPad、Windows Phone 和 Android 设备。

我最喜欢的一款游戏是宝开游戏 (PopCap Games) 开发的《植物大战僵尸》。此游戏已经被移植到了大多数系统，包括 Xbox 360、Mac OS X、Windows、任天堂 DSi、iPhone、Android 和许多其他系统。此外，一些高水准的游戏开始从 Windows、Xbox 360 和 iPhone 等系统移植到 Android 4 中。

Android 4 到底是什么呢？您正在阅读本书，这说明您对此设备的游戏编程感兴趣。但是，此平台的开发是什么样子的？它是什么呢？我们必须询问自己这些问题，因为如果您想开发一款被认真对待的游戏，不是要花费大量金钱，就是要投入大量时间。最有可能的情况是，希望了解 Android 4 游戏开发的人已经了解了 Java SE 7。如果您从未使用过此开发工具，则接下来的两章内容对您来说非常重要，因为我们会创建项目，并频繁地使用 NetBeans 和 Eclipse。我将假设您并未使用过此开发工具。

1.2 Android SDK 简介

Android SDK 是以 Java 语言为基础进行开发的。Java 程序是使用 Java 开发工具包 (Java Development Kit, JDK) 编译的。所有应用程序和游戏都是使用 Java 语言编写的，并且在使用 Java 运行时环境 (Java Runtime Environment, JRE) 的 Android 设备上运行。Java 程序的扩展名是 .class，而源代码文件的扩展名是 .java。使用 Android SDK 构建的程序被编译到了一个可以在 Android 平台上运行的程序包。有两个主要的 Java 开发环境：Eclipse 和 NetBeans。二者各有所长亦各有所短，并不是一种比另一种更好，只是个人喜好的问题。稍后两章将介绍如何安装和配置 Android SDK。

如果您刚接触 Java 和 Android SDK，则可以按照此说明开始编写 Android 代码。但是，如果您之前没有使用 Java 等语言（比如 C#、C++ 或 Lua）进行编程的经验，则可能会发现代码难以理解。在使用 Android SDK 之前，编程新手可能会想要阅读有关基本 Java 编程的入门书籍。由于所有编程语言会共享一些概念，因此了解至少一种语言会对您有所帮助。拥有一些编程经验对充满信心地理解本书中的代码非常重要，因为我们不会介绍基本编程概念，那样会有太多的内容需要介绍。

如果您已经是一名经验丰富的 Java 开发人员，则仍然需要仔细阅读接下来两章的内容，以掌握 Android SDK 和开发工具的结构和安装。

1.3 Android NDK 简介

Android NDK（与 SDK 不同）支持您以本机 C++ 代码而非 Java 语言编写应用程序或游戏的组件。NDK 是一个单独的工具，可在 Android SDK 安装以后再安装它。NDK 不允许以 C++ 语言编写整个应用程序或游戏。更确切地说，它用来补充 SDK，支持 C++ 代码和库，并旨在充当许多没有 Java 库的硬件设备的桥梁。我们可以使用 NDK 优化游戏代码。除了许多 C++ 库外，NDK 还支持实现 3D 渲染的 OpenGL ES 2.0、针对音频的 OpenSL ES 和像素缓存访问。Android 开发人员不推荐对大多数应用程序和游戏使用 NDK，但游戏开发者总是会利用各种可能性！

1.4 Android 开发系统要求

Android SDK 支持下列操作系统：

- Windows XP、Vista、7（32 位或 64 位）；
- Mac OS X 10.4.8 或更高版本（仅 x86）；
- Linux（32 位或 64 位）；具有 Glibc 2.7 或更高版本的任意发行版。

我们将使用 Windows 版的 NetBeans 和 Eclipse，您可以选择使用哪一种开发集成环境（IDE）。接下来的两章将介绍 IDE 和 Android SDK 的安装和配置。

1.5 Android 平台的历史

Android 4 存在的时间非常短，第一个 Android SDK 版本可追溯到 2008 年。Android 直接与 Apple 的 iOS（包括 iPhone、iPod 和 iPad 设备）展开竞争。尽管 Windows Phone 7 的市场份额还较小，但是它仍然是一个有力的竞争对手。

有趣的是，我不会将 Apple 的 iPhone 视为 Palm Pilot（第一台 PDA）的飞跃。iPhone 并没有遵守“便携式计算机”（可追溯到 Palm Pilot 和掌上电脑）的传统，iPhone（即 iOS）来自 Apple 的 iPod。iPod 由索尼公司发明，索尼公司是一家负责便携式音乐播放器 Walkman 系列的公司。在 20 世纪 80 年代或 90 年代，所有人都曾经听说过 Walkman。

您肯定听过“播客（podcast）”这一术语。如今该术语的含义相当广泛，指在网络或便携式播放器上播放的数字音频流。此概念是 Apple 针对 iPod 和 iTunes 开发的，iTunes 现在也支持播放音频。当我们大多数人被卷入关于音乐共享的 Napster 诉讼案时，Apple 正忙着开发 iTunes，并开始以一种变革性的新方式销售音乐：按每首歌曲而不是整个专辑销售音乐。

您是否曾经在广播中听到了一首好听的歌，并且想要购买它以在 iPod、Android 音乐播放器或类似的媒体设备上播放？在过去 10 年里，您需要买下整张 CD，然后使用 Windows Media Player 或 Winamp 等软件将歌曲转换为 MP3 格式。这一点存在争议，但是我想说，Apple iTunes 证明数字音乐销售能取得商业成功，对唱片艺术家和服务提供商来说，都能获得高收益。Amazon 也表明数字媒体是一个成功的行业。

iPod 很成功，因此它演变为 iPhone 和 iPad，在过去几年里，竞争公司试图在这些市场追赶上

Apple。iOS 操作系统工作得很好。它提供用户需要的内容，而不是软件工程师认为用户需要的内容，这是主要差别。尽管 Android 来自 iOS，但它有自己的创新，例如它是基于 Linux 核心构建而成的。

客户需要什么？不是一个面面俱到的拙劣设备，而是一个做最有用事情的出色设备。相反，许多公司雇用“专家”进行客户调研，然后花费数百万试图说服客户他们确实需要该产品。这可能是进入不太知名的市场或根据客户兴趣调整产品功能集的一种好方式。2007 年，Apple 找到了自己的定位，发布了 iPhone，随之而来的是一系列的模仿。

Google 将其标志放在许可中来吸引硬件制造商。大多数出色的 Google 网络服务，例如 Earth、Maps、Picasa 和 Documents (Android 用户所熟知的 Google Drive)，作为免费的应用程序提供给 Android。免费许可对设备制造商极具吸引力，这使 Android 在智能手机和平板电脑市场拥有 1/3 的市场份额。

注意：Android OS 1.0 于 2008 年 9 月发布。鉴于当前操作系统的成熟度和稳定性，这是非常快的上市时间。

*By the
Way*

表 1.1 列出了这一出色操作系统的简史。

表 1.1 Android 操作系统的历史

日期	版本	代号	Linux 内核
2007 年 11 月	1.0 测试版		2.6
2008 年 9 月	1.0		
2009 年 2 月	1.1		
2009 年 4 月	1.5	Cupcake (纸杯蛋糕)	2.6.27
2009 年 9 月	1.6	Donut (甜甜圈)	2.6.29
2009 年 10 月	2.0	Eclair (闪电泡芙)	
2010 年 1 月	2.1		
2010 年 5 月	2.2	Froyo (冻酸奶)	2.6.32
2010 年 12 月	2.3	Gingerbread (姜饼)	2.6.35
2011 年 2 月	3.0	Honeycomb (蜂巢)	2.6.36
2011 年 5 月	3.1		
2011 年 7 月	3.2		
2011 年 10 月	4.0	Ice Cream Sandwich (冰淇淋三明治)	3.0.1
2012 年 8 月	4.1	Jelly Bean (果冻豆)	

1.6 Android 硬件规格

许多设备类型都使用了 Android 操作系统，而不仅仅是智能手机，但智能手机是目前为止最大的市场。图 1.5 显示了三星 Galaxy Nexus 智能手机。如图 1.6 所示，平板电脑的设计与三星 Galaxy Tab 十分相似。这绝不是任何一种品牌的代言，只是为了展示同一制造商生产的智能手机和平板电脑之间的差别。

表 1.2 展示了在编写本书时最受欢迎的 Android 手机型号的常见硬件规格。最值得注意的是，规格并未遵守所有制造商的基本标准。Apple 已经证明太多的开放性和灵活性在移动硬件中并不总是可取的。Android 开发人员面临的困难之一是，需要使用一个代码库支