

移动开发经典丛书

Pro Android Games



Android 4.1

游戏高级编程(第3版)

[美] Vladimir Silva 著
杨丰盛 谢明辉 蔡达毅 译



Apress®

清华大学出版社

移动开发经典丛书

Android 4.1 游戏高级编程

(第 3 版)

[美]	Vladimir Silva	著
	杨丰盛 谢明辉	
	蔡达毅	译

清华大学出版社

Vladimir Silva

Pro Android Games

EISBN: 978-1-4302-4797-5

Original English language edition published by Apress Media. Copyright © 2012 by Apress Media. Simplified Chinese-Language edition copyright © 2013 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字: 01-2013-6036

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Android 4.1 游戏高级编程(第3版)/(美) 席尔瓦(Silva, V.) 著; 杨丰盛, 谢明辉, 蔡达毅 译.

—北京: 清华大学出版社, 2013.9

(移动开发经典丛书)

书名原文: Pro Android Games

ISBN 978-7-302-33410-1

I. ①A… II. ①席… ②杨… ③谢… ④蔡… III. ①移动终端—游戏程序—程序设计
IV. ①TN929.53 ②TP311.5

中国版本图书馆 CIP 数据核字(2013)第 180865 号

责任编辑: 王 军 于 平

装帧设计: 牛艳敏

责任校对: 邱晓玉

责任印制: 杨 艳

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 19.75

字 数: 481 千字

版 次: 2013 年 9 月第 1 版

印 次: 2013 年 9 月第 1 次印刷

印 数: 1~3000

定 价: 49.00 元

译者序

当我第一次看到本书原版英文书 *Pro Android Games, Second Edition* 时，脑海中只出现了一句话——“逆水行舟，不进则退！”。这也正是我毫不犹豫地决定要翻译本书的原因；为什么这么说呢？

初看本书时觉得与上一版似乎没有多大变化，但你是否感觉到智能手机硬件在不停更新？Android 系统版本在不断演进？新技术、新设备也在相继诞生？这正是本书需要解决的问题，本书将上一版中使用的 Android SDK 更新到当前最新的版本，且将重点放在 OpenGL ES 2.x 上。同时，本书介绍了如何将 Android 系统与主流游戏机手柄进行整合，带来更好的游戏体验；而笔者认为更重要的是，本书最后的章节为我们带来了两项全新的技术——增强现实和 Google TV，笔者也曾经关注过这两项技术，但惭愧的是仅仅只有关注，并没采取任何行动。而通过对本书的翻译，也让笔者重新认识到了这两项新技术所拥有的潜力。它们必将进入我们的生活，作为程序员的我们，也是时候有所行动了吧。

接下来请允许我为大家介绍一下永远走在技术前沿的作者 Vladimir Silva，他不仅拥有系统分析师学位和计算机科学硕士学位，同时还是 Oracle 和 Microsoft 等知名公司的认证专家；并且毕业后就加入了 IBM WebAhead 实验室，其研究内容也非常广泛，包括网格计算、神经网络和人工智能；这也许是众多 IT 工程师都希望拥有的经历吧。这样的人生在大家眼里自然是很成功了。但从本书的内容看来，Vladimir Silva 并没有止步不前，而是不停地在研究、学习和分享新的技术。难道这不能体现“逆水行舟，不进则退！”吗？

因此，我们翻译本书就是因为：一方面，本书介绍了运用在 Android 平台上的新技术；另一方面，作者在业界中永不停止的探索精神；更重要的是作者并不将自己限制在一条技术路线上，而是全面发展。我本不想写序，也不擅长写序，所以本序并没有非常华丽的词句，但确实是笔者在翻译本书的过程中最真实的感受，希望能对大家有所帮助。

本书由蔡达毅、谢明辉、杨丰盛翻译，在翻译过程中得到了夏周全同事和清华大学出版社李阳编辑的大力支持，在此表示衷心感谢。作为本书中英文的传递者，我们希望尽可能准确地传达原作者想要表达的意思，使更多读者获益，但由于水平有限，书中难免会出现疏漏，恳请广大读者批评指正。

作者简介

Vladimir Silva 出生于厄瓜多尔的基多市。他在 1994 年获得厄瓜多尔军队理工学院系统分析师学位。同年，他作为交换生来到美国，在中田纳西州州立大学攻读计算机科学硕士学位。毕业后，他加入了 IBM WebAhead 实验室。他的兴趣包括网格计算、神经网络和人工智能。他还拥有许多 IT 认证，包括 Oracle 认证专家(Oracle Certified Professional, OCP)、微软认证解决方案开发专家(Microsoft Certified Solution Developer, MCSD)和微软认证专家(Microsoft Certified Professional, MCP)。他在 IBM developerWorks 社区编写了大量关于安全和网格计算方面的技术文章。

技术审校者简介

Jim Graham 在 1989 年(88 级)毕业于德克萨斯州 A&M 大学，获得了电子与电信专业学士学位。他在国际通信协会(International Communications Association, ICA)1988 年刊的 ICA Communique 上发表过文章(*Fast Packet Switching: An Overview of Theory and Performance*)。他曾担任过 Amoco 公司(位于伊利诺伊州芝加哥市)网络设计小组的助理网络工程师；Tybrin 公司(位于佛罗里达州 Fort Walton Beach)的高级网络工程师；第 16 特种作战信息联队和美国空军特种作战司令部(位于佛罗里达州 Hurlburt Field)的智能系统分析师。他于 2001 年 12 月 18 日收到了第 16 特种作战信息联队的正式表彰信。

前言

欢迎来到《Android 4.1 游戏高级编程(第3版)》，本书将帮助你创建精彩的 Android 游戏。有很多书籍都在解决这个问题，但是本书会以一种独特的视角向你展示如何轻而易举地将原生 PC 游戏移植到 Android 平台中。每一章都将使用真实的示例和源代码。记住在深入研究这本书之前，需要具备 Java 和 ANSI C 方面的坚实基础。我已尝试尽可能清晰、简洁并结合图形和示例代码来解释这些最复杂的概念。每一章提供的源代码都将帮助你详细地理解这些概念，让你成为一名出色的移动游戏开发人员。

绿色机器人已经腾飞

难以置信，Android 进入智能手机市场仅几年时间就取得了如此大的成就。请看看 IDC¹ 进行的智能手机平台市场份额调查，如图 0-1 所示。相比于 2011 年的 36%，现在 Android 已经占据了整个世界上智能手机市场份额的 59%，这是一个惊人的数字！对于诞生仅仅几年的操作系统来说还不坏，并且这个状态还将越来越好。这为希望利用飞速发展的智能手机市场的开发人员展开了一个新的领域。如果你渴望将尖端游戏运行和编译到 Android 平台，《Android 4.1 游戏高级编程(第3版)》就正是你所需要的书籍。

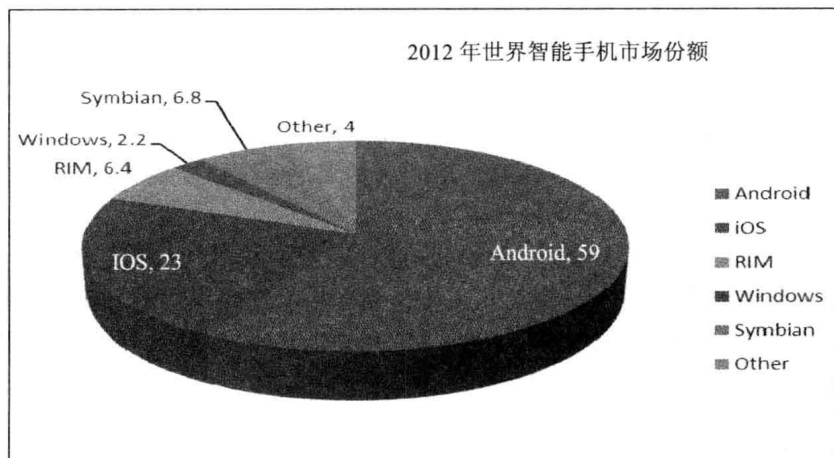


图 0-1 2012 年世界智能手机市场份额

¹ “IDC: Android has a heady 59 percent of world smartphone share,” www.engadget.com/2012/05/24/idc-q1-2012-world-smartphone-share/.

本书的目标读者

本书的目标读者是经验丰富的游戏开发人员，不仅包括 Java 开发人员，还包括 C 开发人员。这很合理，因为性能是游戏开发的关键。其他读者包括：

- 商业应用程序开发人员。如果你从事原生应用程序开发，那么本书将是十分有价值的工具。
- 科学开发人员。解决科学世界中的原始性能问题。介绍 JNI 和 OpenGL 的章节可以帮你实现目标。
- 学习新移动平台的计算机科学专业的学生。Android 开放且相当容易移植，因此本书可以帮助使用很多不同平台的学生，包括 iPhone、BlackBerry 和 Meego 等。
- 任何对 Android 开发感兴趣的人。Android 已经在手机市场空间中加快发展的步伐。你需要扩展自身的技能，包括游戏和图形，否则将会被甩在后面。

学习本书内容需要的技能

本书需要的技能包括 C/C++ 和 Java，以及一些基本的 Linux shell 脚本编程知识。Java 提供了优雅的面向对象特性，但只有 C 能给予推进游戏开发所需要的力量。

坚实的 Android 基础

本书假定你已经了解基本的 Android 开发，例如，你需要知道什么是 Activity、视图和布局。如果你仅仅通过查看下面的代码片段就能理解它们的意思，那么你的基础还不错。

```
public class MainActivity extends Activity
{
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

该代码片段定义了应用程序的主 Activity 或控制应用程序生命周期的类。当应用程序启动时，onCreate 方法将被调用一次，其工作是为应用程序设置内容布局或 GUI。

同时，你还需要对如何使用 XML 创建 GUI 有一个基本的了解。看看下面的代码片段，你能告诉我它做了些什么吗？

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <ImageView android:id="@+id/doom_iv"
```

```

        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="@drawable/doom"
        android:focusableInTouchMode="true" android:focusable="true"/>

<ImageButton android:id="@+id/btn_upleft"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:src="@drawable/img1" />
</RelativeLayout>

```

它定义了一个相对布局。在相对布局中，组件彼此相对放置(有时重叠在一起)。在本例中，有一个图像视图填充整个屏幕。该图像将作为背景，显示存储在项目 `res/drawable` 文件夹下的名为 `doom.png` 的文件，并接收按键和触摸事件。在屏幕的左下角，ID 为 `btn_upleft` 的重叠图像按钮将显示出来。

需要 Android 教程吗？

这里有许多与 Android 开发相关的概念，要记住关于 Activity、视图和布局的每一个细节是不可能的。可以从下面的网址快速获取 Android 教程：<http://developer.android.com/>。

有关 Android 开发人员的终极指南——最新发布、下载、SDK 快速入门、版本说明、原生开发工具、以前发布版本，可以在以下站点中找到：http://developer.android.com/sdk/1.6_r1/index.html。

由于本书(特别是在处理原生代码的章节中)频繁使用 Android 软件开发工具包(Software Development Kit, SDK)命令工具(系统管理员任务)，因此你需要清楚地理解这些工具，尤其是 Android 调试桥(Android Debug Bridge, ADB)。你需要知道如何完成以下操作：

- 创建 Android 虚拟设备(Android Virtual Device, AVD)。AVD 封装了详细的移动设备配置信息，如固件版本和 SD 卡路径。创建 AVD 非常简单，可以使用 AVD Manager(可通过单击工具栏中的黑色手机图标来访问)在集成开发环境(Integrated Development Environment, IDE)中完成。
- 创建 SD 卡文件。在后面的章节中，有些游戏有大文件(5MB 或更多)。为了节省空间，会将所有的游戏文件存储在设备 SD 卡中，因此你需要知道如何创建 SD 卡文件。例如，为了在 `home` 路径下创建 100MB 大小的 SD 卡文件并命名为 `sdcard.iso`，可以使用如下命令：

```
$ mksdcard 100M $HOME/sdcard.iso
```

- 连接到模拟器。这样做可以完成一些系统管理，如库提取。打开设备 shell，使用如下命令：

```
$ adb shell
```

- 将文件上传到模拟器和从模拟器导出文件。该任务对于将游戏文件存储到设备和从设备中提取出游戏文件非常有帮助。使用如下命令：

```
$ adb push <LOCAL_FILE> <DEVICE_FILE>
$ adb pull <DEVICE_FILE> <LOCAL_FILE>
```

注意：

在运行命令创建 SD 卡文件、连接模拟器或者上传和导出文件前，都要确保 SDK_HOME/tools 路径已经添加到系统变量 PATH 中。

Linux 与 shell 脚本编程的基础知识

除第 1 章外，本书所有章节均使用了 Java/C 混合编程，所以需要 you 了解原生开发的知识。对于这些章节，你将在一个类似于 Linux 的 shell(叫做 Cygwin)中完成工作，所以将重新使用这些早期的 UNIX 技能。你应该知道基本的 shell 命令，例如列举文件、安装软件组件(这可能非常棘手，需要根据 Linux 发行的版本而定)以及基本的系统管理。在本书中有一些非常简单的 shell 脚本，大致了解一下 bash shell 总是有帮助的。

提示：

如果你需要复习一下 Linux 和 shell 脚本，请参阅下面由 Ashley J.S Mills 编写的教程：
<http://supportweb.cs.bham.ac.uk/documentation/tutorials/docsystem/build/tutorials/unixscripting/unixscripting.html>。

学习本书所需要的硬件/软件

为充分利用本书，你将需要具备以下提到的工具。

已经正确安装 Java SDK 的 Windows 或 Linux PC

我想这很容易理解，因为大多数 Android 开发都是由 Java 完成的。注意我提到的是 Java SDK，而不是 JRE。必须是 SDK，因为后面的章节会大量使用 JNI 头文件和命令行工具。

已经正确安装的 Eclipse IDE 和 Android SDK

Eclipse 几乎是 Android 开发的标准 IDE。我使用 Eclipse Galileo 为本书的示例创建工作区。不过，Eclipse Ganymede 应该也能正常工作。

需要 IDE 吗？

尽管 Eclipse Helios 已用于创建代码工作区，但是你仍可以使用自己喜欢的 IDE。当然，这需要一些额外的配置。你可以从 www.eclipse.org/ 获取 Eclipse。

对于如何使用其他 IDE(如 IntelliJ)或基本编辑器设置 Android SDK 的说明,请参见 <http://developer.android.com/guide/developing/other-ide.html>。第 1 章详细介绍如何在开发机中配置 Android SDK。

原生开发工具包(NDK)

NDK 是任何正式游戏开发人员必不可少的工具。它提供了编译器链、头文件以及将尖端游戏移植到移动平台所需的文档。使用 NDK,开发人员可以逃脱 Java 内存堆的束缚,利用他们自己的创造力开发出强大的 C/C++引擎,而仅受到硬件能力的限制。在本书中,你将广泛地使用 NDK,因此要充分理解每章中提出的概念,就需要具备坚实的 C 程序基础。第 1 章详细介绍如何在开发机中设置最新的 NDK。

章节源代码

这是一个可选工具,但是它对你理解概念有着极大的帮助。我将尽可能简单地描述每个章节。然而,一些游戏(特别是 Quake I 和 Quake II)使用了 C 编写的庞大核心引擎(Doom 有近 10 万行代码),而且缺少注释,难以理解。总而言之,你将看到如何轻而易举地将两种强大的语言(Java 和 C)结合在一起。从本书出版商的站点可以获取本书的配套源代码,它们都使用最新的 Eclipse SDK 创建。

本书独特之处

我认为重要的是让读者理解本书的写作目标和独特之处。尽管 Java 是 Android 的主要开发语言,但 Google 已经意识到 Android 要想成为成功的游戏平台,就必须混合 Java/C 开发。因此,Google 发布了原生开发工具包(NDK)。我认为 Google 支持 C 开发是明智的,否则会有大量的原生游戏将是为其他平台编写的,如 iOS。

PC 游戏已经经历了几十年的发展(大部分由 C 语言编写),只需要使用一个简单的 ARM C 编译器,就可以将成千上万的 PC 游戏移植到 Android 平台。这也正是本书的独特之处。如果可以采用一种优雅的方式简单地结合这两种语言,并在整个过程中节省你大量时间和金钱,为什么要将 10 万行令人头疼的复杂代码从 C 转换到 Java? 这是我的目标,也是本书能从众多图书中脱颖而出的原因。但是,本书也包含纯 Java 游戏章节,以同时满足 Java 和 C 爱好者。

相对上一版本,本书有何变化

随着 Android 的不断更新,本书相对上一版本已经有大量内容更新。这些改动如下:

- 更新:更新到最新版本的 Android SDK、原生开发工具包(NDK)和 Eclipse IDE。

- 更加关注原生端：我认为可以公正地说，Java 对于 3D 游戏开发人员已经没有吸引力，尤其是强大的游戏。Java 的内存限制和性能匮乏是罪魁祸首。因此，《Android 4.1 游戏高级编程(第 3 版)》将更加注重原生游戏的开发和图形硬件加速。
- 更广大更好的真实引擎。我的目的不是简单地提供一些开发游戏的技巧，而是提供更真实、强大的实例。本书将展示如何将优秀的游戏引擎几乎不做任何改动地移植到你的移动设备上，如 Quake I 和 Quake II，而且包括上一版本中旧的内容——Doom。

Android SDK 兼容性

作为开发人员，你可能很想知道本书代码的 SDK 兼容性。随着 Android SDK 新版本的频繁发布，这一问题显得十分重要。在编写本书时，Google 刚发布了 Android SDK 3.2 版本。本书代码已在 Android SDK 的以下版本上进行了充分测试。

- SDK 版本 4.1
- SDK 版本 3.x

重要的是本书代码可以运行在任何版本的 SDK 中，这也是我一直以来的目标。

第 1 章

本章提供编译混合游戏的第一步，即设置 Windows 系统，包括

- 获取 Android 源代码
- 设置 Eclipse IDE 以进行开发
- 安装最新版本的 NDK
- 创建模拟器以进行测试以及配置真实设备
- 将本书源代码导入工作区中，这对于理解一些复杂主题很重要

第 2 章

在本章中，你将学到如何通过一种优雅的方式结合 Java 和 C 代码，以便在原生库的上层建立简单的 Java 应用程序。你将学习 Java 原生接口(Java Native Interface, JNI)，以及结合 Java 和 C 所需要使用的 API 等，这些都是令人兴奋的概念，包括如何加载原生库、如何使用原生关键字、如何生成 JNI 头文件，以及方法签名、Java 数组与 C 数组、调用 Java 方法、编译和打包程序等内容。

第 3 章

本章讨论使用 OpenGL 处理 3D 图形。其中将展示一种巧妙的方法，允许 Java 和 C 混合调用 OpenGL API。本章使用 Google 提供的 3D 立方体实例对这一概念进行很好的阐述，并展示如何在纯 Java 和混合模式下使用 OpenGL。该技巧将为 Android 的 3D 开发开辟新的领域，并可能将大量的 3D PC 游戏带到这个平台，同时还能节约开发成本和时间。

第4章

第4章采用 OpenGL ES 2.0 处理高效图形。本章首先对 OpenGL ES 2 所能提供的最重要特性进行简短的描述,包括着色器、GLSL 以及它们如何影响 Android 平台。然后,巧妙地通过一个 Android 项目对 GLSL 进行更深入的了解,该项目使用 OpenGL ES 2.0 渲染一个二十面体。另外,本章还展示如何使用单点和多点触摸函数来控制二十面体的旋转速度,以及通过“捏”操作实现放大和缩小。

第5章

第5章随着引入 Doom 将本书内容带入一个新的层次,Doom 是 PC 游戏的一个里程碑,是最伟大的 3D 游戏之一,它开辟了 3D 图形的新领域。本章的最终目的不是描述游戏本身,而是向你展示将诸如 Doom 这样复杂的 PC 游戏移植到 Android 平台是多么容易。如何才能证明呢?Doom 有超过 10 万行的 C 代码,而把它移植到 Android 平台(包括创建移动 UI 的 Java 代码)却只需要不到 200 行的新增 JNI API 调用代码。本章向你展示不需要把 10 万行 C 代码转换为 Java 代码,而只需要在一个优雅的应用程序中结合这两种强大的语言。考虑到可能节省开发时间和成本,这将是必读的一章。

第6章

这是开始让我们真正感到兴奋的部分。第6章将为你带来第一人称射击游戏(First Person Shooter, FPS)的精品:Quake。你会学习如何轻而易举地将强大的 PC 引擎移植到 Android 平台。你可以保持原始 C 代码 95% 的完整性,额外添加的轻量级 Java 包装代码仅有 500~1000 行。现在你可以在智能手机上玩 Quake 了。

第7章

本章在第6章的基础上把 Quake II 引擎带到你的指尖。你将看到,通过一个非常棒的名为 NanoGL 的工具,可以使 Quake II 引擎高度复杂的 OpenGL 渲染器原始代码保持完整性。NanoGL 允许开发人员将复杂的 OpenGL 立即绘制模式转换到 OpenGL ES 上,并保持原始代码的完整性。你还将学习如何创建定制音频和视频处理程序,以便让 Quake II 引擎能够在 Android 中正常工作,同时也证明 Java 语言出色的可重用特性。总而言之,本章将保持原始 Quake II 的 C 代码 99% 的完整性,并且不做任何改动地重用第6章中的轻量级 Java 包装。本章将向你展示如何对强大的工具简单地进行组合来驯服 Quake II 强大的 OpenGL 渲染器。

第8章

本章介绍蓝牙控制器。用触摸屏界面或者很小的键盘玩游戏是很困难的(例如第一人称射击游戏)。一些游戏必须用游戏键盘才能玩。如果你很热爱游戏,通过本章就会很高兴地

看到在游戏中集成两种流行的游戏控制器 Wiimote 和 Zeemote 是很容易的。在学习本章的过程中,你将掌握蓝牙 API 的用法,了解 Wiimote 和 Zeemote 的内部工作机制,以及关于 JNI 和异步线程等方面的知识。

第 9 章

本章介绍的新主题很有意思,也许很快就会出现你的移动设备和卧室中:增强现实技术和 Google TV。目前看来,休闲游戏正在朝着增强现实的方向发展。其他平台(如 PS Vista)已经迈出了第一步,为实现增强现实游戏提供了坚实的基础。增强现实技术已经引起了很多人的注意,Android 开发人员也在其中。本章将介绍如何使用流行的 ARToolKit 以及 OpenGL 和 JNI 创建支持增强现实的应用程序/游戏。另外还将关注智能电视的兴起,主要是 Google TV。Google TV 采用了 Android 3.1 版本,所以与你可能正在准备创建的游戏是完全兼容的。

联系作者

如果你有任何疑问或者想要发表一些评论,或者在本书中发现了一些错误,认为应该让作者知道,那么可以通过以下方式联系作者:vladimir_silva@yahoo.com。

目 录

第 1 章 欢迎来到 Android 世界	1
1.1 配置开发机	1
1.2 下载并安装 SDK	2
1.3 配置 Eclipse	2
1.4 安装原生开发工具包	6
1.4.1 安装 NDK	6
1.4.2 安装 Cygwin	6
1.5 创建 Android 模拟器	7
1.5.1 安装 Android 4.1	7
1.5.2 创建 AVD	8
1.6 配置真实设备	9
1.7 将源代码导入 Eclipse	11
1.8 本章小结	12
第 2 章 针对手机或平板电脑的游戏技巧	13
2.1 在 Android 中编译原生代码	13
2.2 应用程序的体系结构	16
2.3 编译共享库	25
2.4 在设备上测试应用程序	26
2.5 Java 对 C/C++ 事件处理的包装	28
2.5.1 独立处理音频	28
2.5.2 级联视频事件	33
2.6 多点触摸	38
2.6.1 MultiTouchGesture	38
2.6.2 MultiTouchScreen	40
2.6.3 TestActivity	41
2.7 本章小结	43
第 3 章 更多 OpenGL 与 JNI 游戏编程技巧	45
3.1 移动设备的能力	46
3.2 Java 方式的 OpenGL	47

3.2.1	创建项目	47
3.2.2	Java 版本的主 Activity	50
3.2.3	表面视图	51
3.2.4	GL 线程	54
3.2.5	立方体渲染器	57
3.2.6	Cube 类	60
3.3	原生方式的 OpenGL	62
3.3.1	主 Activity	64
3.3.2	原生接口类	65
3.3.3	修改原立方体示例	66
3.3.4	原生立方体渲染器	68
3.3.5	原生立方体类	73
3.3.6	编译和运行示例	75
3.4	通过 OpenGL ES 混合模式缩放视频缓冲区	77
3.4.1	为什么使用混合缩放	78
3.4.2	初始化表面	80
3.4.3	绘制到纹理中	81
3.4.4	当图像的大小不是 2 的次幂时会发生什么	83
3.5	本章小结	88
第 4 章	OpenGL ES 2.0 的高效图形和可移植性	89
4.1	可移植性	89
4.1.1	处理直接模式	90
4.1.2	加载纹理	92
4.1.3	显示列表、服务器属性和其他	93
4.1.4	使用库包装	93
4.2	OpenGL ES 2.0 的高效性	94
4.2.1	着色器	94
4.2.2	GLSL	95
4.2.3	剖析着色器	97
4.3	在 Android 中调用 OpenGL ES 2.0	100
4.4	二十面体项目	100
4.4.1	定义形状	100
4.4.2	处理项目	102
4.4.3	原生二十面体	110
4.5	添加“滑动”和多点触摸的“捏”缩放功能	116
4.6	编译与运行	120
4.7	本章小结	120

第 5 章	3D 射击游戏: Doom	121
5.1	组合使用 Java/C, 功能无极限	122
5.2	将 Doom 移植到移动设备	123
5.3	Doom 游戏的体系结构	125
5.4	Java 层的主 Activity	126
5.4.1	创建处理程序	127
5.4.2	游戏布局	128
5.4.3	菜单和选择项处理程序	130
5.4.4	按键和触摸事件处理程序	131
5.4.5	原生回调处理程序	132
5.4.6	导航控件	136
5.5	独立处理音频格式	138
5.6	原生接口类	139
5.6.1	回调监听器	139
5.6.2	原生方法	139
5.6.3	C 到 Java 的回调	140
5.7	原生层	143
5.7.1	原生方法的实现	143
5.7.2	对原始游戏进行修改	154
5.8	使用 NDK 编译 Doom	160
5.9	在模拟器中测试 Doom	162
5.10	本章小结	164
第 6 章	3D 射击游戏: Quake	165
6.1	一些关于 Quake 的术语	165
6.2	代码重用的威力: Quake 中的新代码少于 500 行	166
6.2.1	OpenGL 不同于 OpenGL ES	166
6.2.2	软件渲染是否是合理解决方案	168
6.2.3	NanoGL: 救助器	168
6.3	Android 架构的 Quake	173
6.3.1	Java OpenGL 渲染器架构	174
6.3.2	独立处理音频格式	178
6.3.3	处理键盘事件	181
6.3.4	处理触摸事件	183
6.3.5	游戏的启动 Activity	185
6.4	原生 Quake 引擎必需的改动	186
6.4.1	视频处理程序的改动	187
6.4.2	处理 X 轴旋转(pitch)和 Y 轴旋转(yaw)	189
6.4.3	处理前进和横向移动	190

6.4.4	音频处理程序的改动	190
6.4.5	修复游戏循环	193
6.5	在设备上运行 Quake	195
6.6	本章小结	201
第 7 章	3D 射击游戏: Quake II	203
7.1	代码回顾	203
7.2	逃离 Java 堆的束缚	204
7.3	让 Quake II 引擎在 Android 中工作	205
7.3.1	代码的可重用性	205
7.3.2	当发生致命错误时如何处理	214
7.3.3	OpenGL 立即模式问题	217
7.3.4	视频处理程序	217
7.4	使用 NDK 编译 Quake II	221
7.5	在设备或模拟器中运行游戏	224
7.6	Quake II 在多种设备中的性能表现	227
7.7	本章小结	228
第 8 章	蓝牙控制器的乐趣	231
8.1	剖析 Wiimote	231
8.1.1	查询过程	232
8.1.2	设备连接	233
8.1.3	其他小技巧	235
8.2	Wiimote 控制器应用程序	236
8.2.1	Wiimote Android 项目	236
8.2.2	WiiC 库	245
8.2.3	WiiC JNI 接口	250
8.2.4	编译与测试	258
8.3	Zeemote	261
8.4	本章小结	265
第 9 章	展望未来: 增强现实技术与 Google TV	267
9.1	什么是 AR	267
9.2	AR 与 Android	268
9.2.1	寻找合适的 AR 库	269
9.2.2	认识 ARToolKit(ARTK)	269
9.2.3	知识汇总	274
9.2.4	故障排除	275
9.3	在 Android 平台上运行 ARToolKitRTK	275
9.3.1	获取视频载体	276