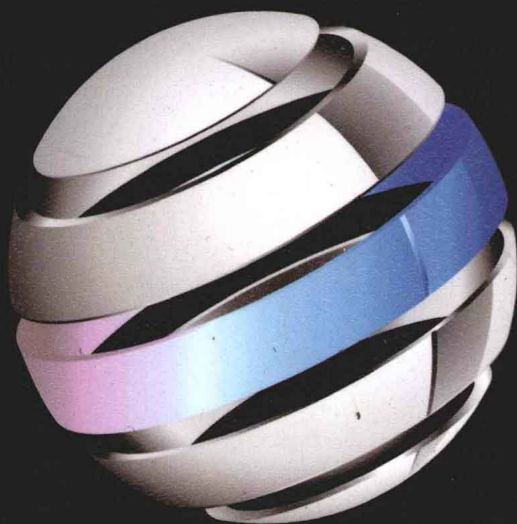


移动与嵌入式开发技术

构建更加成熟和令人着迷的 Android 游戏 app
充分发挥最新版 Android SDK 4 的强大功能



Android 4 游戏高级编程 (第2版)

Advanced Android 4 Games

[美] Vladimir Silva 著
杨丰盛 译
谢明辉 蔡达毅



ress®

清华大学出版社

移动与嵌入式开发技术

Android 4 游戏高级编程 (第 2 版)

常州大学图书馆
[美] Vladimir Silva 著
杨丰盛 谢明辉 蔡达毅 译

清华大学出版社

北 京

Vladimir Silva

Advanced Android 4 Games

EISBN: 978-1-4302-4059-4

Original English language edition published by Apress, 2855 Telegraph Avenue, #600, Berkeley, CA 94705 USA. Copyright © 2012 by Apress L.P. Simplified Chinese-language edition copyright © 2012 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: 01-2012-4744

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 游戏高级编程: 第2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, 2013.1

(移动与嵌入式开发技术)

书名原文: Advanced Android 4 Games

ISBN 978-7-302-30744-0

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

中国版本图书馆 CIP 数据核字(2012)第 284132 号

责任编辑: 王 军 李维杰

装帧设计: 牛艳敏

责任校对: 邱晓玉

责任印制: 李红英

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: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 16.5

字 数: 402 千字

版 次: 2013 年 1 月第 1 版

印 次: 2013 年 1 月第 1 次印刷

印 数: 1~4000

定 价: 39.00 元

产品编号: 048514-01

作者简介

Vladimir Silva 出生于厄瓜多尔基多。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

Vikram Goyal 是几本 Apress 书籍的作者，同时他也维护几个网站。他目前住在澳大利亚的布里斯班。

James Graham 出生于美国阿拉巴马州，在佛罗里达州的华尔顿堡滩和德克萨斯州的圣安东尼奥长大。他在 1988 年获得德克萨斯州 A&M 大学电子与电信专业学士学位。他一直是 Amoco 公司 Network Design 小组的助理网络工程师和美国空军特种作战司令部的智能系统分析师。他除了是一名 Android 开发人员外，还是高评价自由软件——飓风追踪程序 JStrack 的作者。

译者序

经过几年的发展，Android 已是今非昔比。从最初的智能手机逐渐进入人们生活的各个领域，而且这种迅猛发展从未停止过。相信在不久的将来，我们的生活中必将充满各种 Android 设备。试想一下，我们的手机、电脑、电视机、电冰箱、洗衣机、空调、照相机、餐桌、床等，一切尽是绿色机器人。作为开发者的我们，能看到自己的游戏运行在这些设备上，是多么的自豪啊！如果你已经有一些颇受欢迎的游戏在 PC 或其他设备上运行，如何完美地迁移到 Android 系统一定会成为你最关注的问题。

本书作者将帮助你以最快的速度将经典的 PC 游戏移植到 Android 平台，这也正是本书颇受欢迎的原因之一。有些书籍教你如何开发应用，有些书籍教你如何开发游戏，唯有本书教你如何最大限度地重用已有的代码。作者在本书中用了 3 个非常强大而且经典的 3D PC 游戏证明了这一点，只需要下载本书的源代码，就可以将 Doom、Quake I、Quake II 这 3 个曾在 PC 上风靡全球的 3D 游戏轻松地编译并部署到 Android 设备。至于如何编译，作者在本书中已经毫无保留地将方法写在了第 5~7 章中。

当然，除了这些精彩的实战演练外，本书还非常详细地介绍了 Android 游戏开发过程中的各个方面，包括图形硬件加速(OpenGL ES 1.x/2.0)、音视频处理、触摸屏技术、多点触摸技巧、外接蓝牙控制器的运用，以及针对不同硬件设备和系统版本进行的性能优化方案。作为游戏开发过程中最核心的内容，本书在第 2~4 章花了大量篇幅详细讲解，并且配有最合适的案例，让读者更深刻地理解每个知识点并加以运用。

引用作者的话说，Android 已经腾飞，并且后劲十足，我们没有理由不希望自己的作品运行在知名的操作系统上，向全世界用户展示我们的自豪。是时候行动了！

本书由杨丰盛、蔡达毅、谢明辉翻译，在翻译过程中得到了夏周全同事和清华大学出版社的大量帮助，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。

译者
2012 年 9 月

前言

欢迎来到《Android 4 游戏高级编程(第 2 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，但是本书将以一个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。每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，作为手机游戏开发人员，它们还可以帮助你节省大量的时间。

绿色机器人已经腾飞

难以置信，Android 进入智能手机市场仅两年时间就取得了如此成就。请看看美国智能手机平台市场份额，如图 0-1 所示。根据 Nielsen 的调查¹，2011 年 5 月，Android 占据了美国 36% 的智能手机市场份额，这对于诞生仅仅两年的操作系统来说并不算太差，并且这个状态还将越来越好。Distimo 是一家专注于应用程序商店分析的公司，它预测 Android Market 将在 2011 年 8 月超过苹果 App Store²。这为希望利用飞速发展的智能手机市场的开发人员打开了一个新的领域。如果你渴望将尖端游戏运行和编译到 Android 平台，《Android 4 游戏高级编程(第 2 版)》就正是你所需要的。

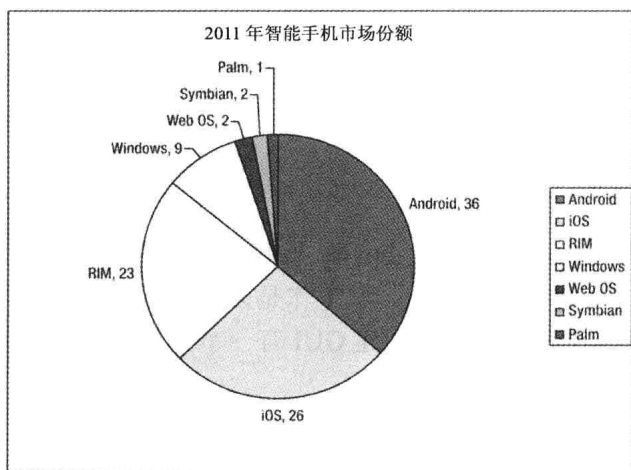


图 0-1 智能手机市场份额(2011 年 4 月, Nielsen)

1 《Android Leads 在美国智能机市场的份额和数据使用》，Nielsen Wire, <http://blog.nielsen.com/nielsenwire/?p=27793>。

2 《2011 年 8 月 Android 超越苹果 App Store》，由 Distimo 撰写的报告, <http://techcrunch.com/2011/05/05/android-to-surpass-apples-app-store-in-size-in-august-2011-report-exclusive/>。

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。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。第 3 章可以帮你实现目标。
- 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，因此本书可以在很多平台上帮助学生，包括 iPhone、BlackBerry 和 Meego。
- 任何对 Android 开发感兴趣的人。Android 已经在手机市场空间中加快了发展的步伐。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。

学习本书大部分内容需要的技能

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功能，但只有 C 能给予推进游戏开发所需要的力量。总而言之，你必须具备下面内容中描述的技能。

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 或是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着整个屏幕。该图像将作为背景，显示存储在项目 `res/drawable` 文件夹下名为 `doom.png` 的文件，并接收按键和触摸事件。在屏幕的左下方，重叠的 ID 为 `btn_upleft` 的图像按钮将被显示出来。

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。可以从下面的地址快速获取 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)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，将所有的游戏文件存储在设备 SD 卡中，你需要知道如何创建 SD 卡。例如，在 `home` 路径下创建 100MB 大小的 SD 卡文件，命名为 `sdcard.iso`，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杂项，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 路径已经被添加到系统变量 PATH。

Linux 与 shell 脚本编程的基础知识

对于处理混合游戏的章节，你将在 Ubuntu Linux 中完成工作，所以将重新使用这些老的 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

为充分利用这本书，你将需要具备提到的以下工具。

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 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 已被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，这需要一点点额外的配置。你可以从 www.eclipse.org/ 获取 Eclipse。

对于如何为其他 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>。

要正确安装 Android SDK，需要进行如下操作：

(1) 为 Eclipse 安装 Android SDK 插件：

- 在 IDE 主菜单中，单击 Help 菜单并选择 Install New Software。
- 单击 Add 按钮，添加新的站点并输入名称 Android SDK 和地址 <https://dl-ssl.google.com/android/eclipse/>。单击 OK 按钮。
- 从 Available Software 对话框中选择 Android SDK 并按照向导的简易安装说明进行安装。

(2) 安装 Android SDK，可以从前面提到的 Android 站点下载该工具。注意 Eclipse 必须知道 Android SDK 的位置。从 IDE 主菜单单击 Window 菜单并选择 Preferences 菜单项。在左边的导航菜单中，选择 Android 并输入 SDK 位置(如图 0-2 所示)。我使用的是 SDK 3.1，因为它是编写本书时最新版本的 SDK。不过，本书的代码已在 SDK 2.3 和 3.1 中测试通过(详细信息参见接下来的“SDK 兼容性”部分)。

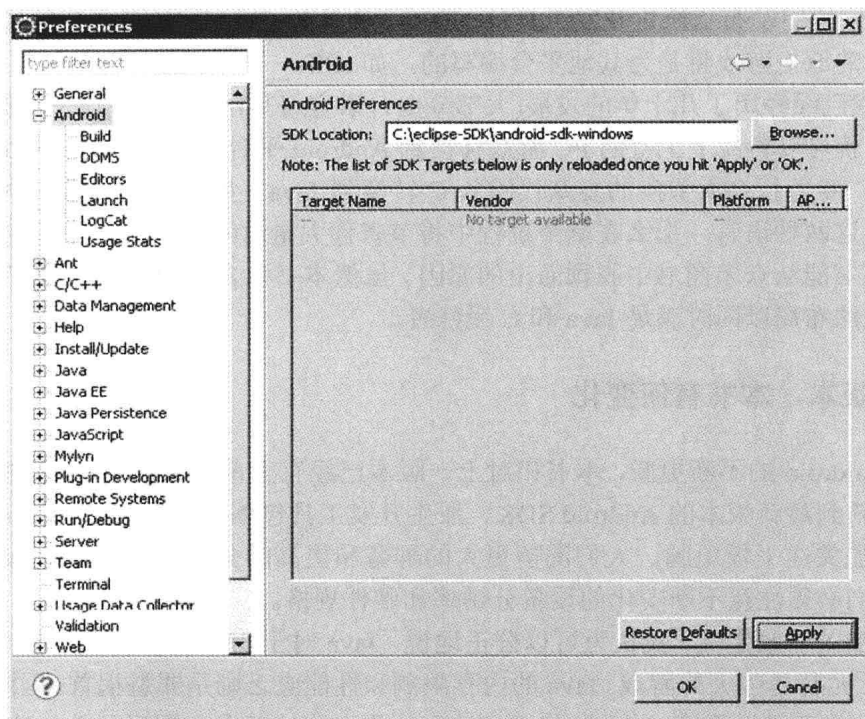


图 0-2 Eclipse Galileo 中的 Android SDK 配置对话框

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，开发人员可以逃离 Java 内存堆的束缚并激发他们在编译最强大的 C/C++ 引擎时的创作力，因为这仅限于硬件能够提供哪些支持。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。

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 Wolf 3D 和 Doom)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语言(Java 和 C)简单结合在一起。从 www.apress.com 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，他们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。为什么要将 10 万行令人头疼的复杂代码从 C 转换到 Java？如果可以采用一种优雅的方式简单地结合这两种语言，那么在整个过程中将节省你大量时间和金钱。这是我的目标，也是为什么本书能从众多图书中脱颖而出的原因。虽然本书包含纯 Java 游戏章节，但同样也呈现了合理的布局以同时满足 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被更新。这些改动如下：

- 更新到最新版本的 Android SDK、原生开发工具包(NDK)和 Eclipse IDE。
- 更加关注平板电脑。人们渴望更大的屏幕和更高的分辨率。平板电脑越来越多，我们必须注意不断变化的设备分辨率和硬件规格。
- 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则是罪魁祸首。因此，《Android 4 游戏高级编程(第 2 版)》将更加注重原生游戏的开发和图形硬件加速。
- 更大更好的现实世界引擎。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移植到你的移动设备上，如 Quake I 和 Quake II，而且还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

的频繁出现，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 4.0 版本和 3.1 版本上进行了充分测试。

底线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。

本书对强大的混合游戏有着很好的平衡布局，按章节划分如下：

第 1 章

本章提供了编译混合游戏的第一步，设置 Linux 系统，包括获取 Android 源代码、提取设备系统库、定制交叉编译工具链、定制编译脚本以及设置本书其余部分需要使用的 IDE。

第 2 章

在本章，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。你将学习关于 Java 原生接口(Java Native Interface, JNI)，以及结合 Java 和 C 所需要使用的 API 等，这都是一些令人兴奋的概念，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 OpenGL 的 3D 图形。其中展示了一种巧妙的方法，允许 Java 和 C 混合调用 OpenGL API。Google 提供的 3D 立方体实例对这一概念进行了很好的阐述，并展示了如何在纯 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了简短的描述，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 进行了更深入的了解，该项目使用 OpenGL ES 2.0 渲染一个二十面体。另外，还展示了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 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，也是毫无争议的最伟大的 3D 游戏，它开辟了 3D 图形的新领域。本章最终的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。你将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，额外添加的简易 Java 封装代码仅 500 到 1000 行。现在你可以在智能手机上玩 Quake 以展示荣耀。

第 7 章

本章在第 6 章的基础之上把 Quake II 引擎带到你的指尖。将向你介绍非常棒的名为 NanoGL 的工具，它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，同时也证明了 Java 语言伟大的可重用特性。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 封装。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。

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	5
1.4.1	安装 NDK	6
1.4.2	安装 Cygwin	6
1.5	创建 Android 模拟器	6
1.6	配置真实设备	9
1.7	本章小结	10
第 2 章	针对手机或平板电脑的 游戏技巧	11
2.1	在 Android 中编译原生代码	11
2.1.1	创建支持原生代码的 Android 项目	13
2.1.2	应用程序的体系结构	14
2.2	编译共享库	25
2.3	调试模拟器	26
2.4	Java 对 C/C++ 事件处理的 封装	27
2.4.1	独立处理音频	27
2.4.2	级联视频事件	32
2.5	多点触摸	37
2.5.1	MultiTouchGesture	37
2.5.2	MultiTouchScreen	39
2.5.3	TestActivity	41
2.6	蓝牙控制器(Zeemote)	42
2.7	本章小结	46

第 3 章	更多 OpenGL 与 JNI 游戏技巧	49
3.1	移动设备的动力	50
3.2	Java 方式的 OpenGL	51
3.2.1	创建项目	51
3.2.2	Java 版本的主 Activity	54
3.2.3	surface 视图	55
3.2.4	GL 线程	58
3.2.5	立方体渲染器	61
3.2.6	Cube 类	64
3.3	原生方式的 OpenGL	66
3.3.1	主 Activity	68
3.3.2	原生接口类	70
3.3.3	修改原示例	71
3.3.4	原生立方体渲染器	72
3.3.5	原生立方体类	78
3.3.6	编译和运行示例	79
3.4	通过 OpenGL ES 混合模式 缩放视频缓冲区	82
3.4.1	为什么使用混合缩放	83
3.4.2	初始化 surface	85
3.4.3	绘制到纹理	86
3.4.4	当图像的尺寸不是 2 的幂时 会发生什么	89
3.5	本章小结	94
第 4 章	OpenGL ES 2.0 的高效图形	95
4.1	OpenGL ES 2.0 与 Android	95
4.1.1	着色器	96
4.1.2	GLSL	96
4.1.3	剖析着色器	98

4.2	在 Android 中调用 OpenGL ES 2.0	101	6.4	Quake 的 Android 架构	179
4.3	二十面体项目	102	6.4.1	Java OpenGL 渲染器架构	180
4.3.1	定义形状	102	6.4.2	独立处理音频格式	185
4.3.2	处理项目	104	6.4.3	处理键盘事件	187
4.3.3	原生二十面体	112	6.4.4	处理触摸事件	189
4.4	添加“滑动”和多点触摸的“捏” 缩放功能	119	6.4.5	游戏的启动 Activity	192
4.5	编译与运行	122	6.5	原生 Quake 引擎必需的改动	193
4.6	本章小节	123	6.5.1	视频处理程序的改动	194
第 5 章	3D 射击游戏: Doom	125	6.5.2	处理 X 轴旋转(pitch)和 Y 轴 旋转(yaw)	196
5.1	组合使用 Java/C, 功能 无极限	126	6.5.3	处理前进和横向运动	197
5.2	将 Doom 移植到移动设备	127	6.5.4	音频处理程序的改动	197
5.3	Doom 游戏的体系结构	129	6.5.5	修复游戏循环	200
5.4	Java 层的主 Activity	130	6.6	在设备上运行 Quake	202
5.4.1	创建处理程序	131	6.7	本章小结	208
5.4.2	游戏布局	132	第 7 章	3D 射击游戏: Quake II	209
5.4.3	菜单和选择项处理程序	134	7.1	代码回顾	209
5.4.4	按键和触摸事件处理程序	135	7.2	逃离 Java 堆的束缚	210
5.4.5	原生回调处理程序	137	7.3	让 Quake II 引擎在 Android 中 工作	211
5.4.6	导航控件	140	7.3.1	代码的可重用性	211
5.5	独立处理音频格式	142	7.3.2	当发生致命错误时如何 处理	221
5.6	原生接口类	143	7.3.3	OpenGL 立即模式问题	223
5.6.1	回调监听器	143	7.3.4	视频处理程序	224
5.6.2	原生方法	144	7.4	使用 NDK 编译 Quake II	228
5.6.3	C 到 Java 的回调	144	7.5	在设备或模拟器中运行游戏	231
5.7	原生层	147	7.6	Quake II 在多种设备中的性能 表现	234
5.7.1	原生方法的实现	147	7.7	本章小结	234
5.7.2	对原始游戏进行修改	159	附录	部署和编译技巧	237
5.8	使用 NDK 编译 Doom	165			
5.9	在模拟器中测试 Doom	167			
5.10	本章小结	169			
第 6 章	3D 射击游戏: Quake	171			
6.1	一些关于 Quake 的术语	171			
6.2	代码重用的威力: Quake 中的 新代码少于 500 行	172			
6.3	软件渲染是合理解决方案吗	174			

第 1 章

欢迎来到 Android 世界

本章将通过介绍“如何配置环境来编译混合代码(C/Java)的游戏”这一问题来揭开序幕，其中包括对最新版本的开发 IDE(Eclipse)和 Android SDK，以及原生开发工具包(Native Development Kit, NDK)的介绍，这些都是构建完美 Android 游戏所必需的工具。如果你想学习如何结合 Java 优雅的面向对象功能和 C 超强的原生性能，那么这些内容将会非常重要，同时也是后面章节要构建所有原生游戏引擎功能所必需的。

假设在你的台式机上已经安装了以下软件：

- Eclipse：用于创建项目的开发 IDE，在本书编写时，Eclipse 的最新版为 3.7，代号为 Indigo，不过 Eclipse 3.6(Helios)或 Eclipse 3.5(Galileo)也照样可以使用。
- 正确配置的 Android SDK：在本书编写时，SDK 的最新版本为 4.0。
- Java JDK：这是运行 Eclipse 和 Android SDK 所必需的，要求 Java 5.0 以上的任意版本即可。

接下来将逐步介绍配置开发机的过程。

1.1 配置开发机

在构建 Android 游戏之前需要完成以下几个步骤：

- (1) 首先，也是最基本的，我们需要最新版本的 Java SDK/JRE(5.0 或后续版本)。在安装之前，请确保使用的是正确的版本。在此假设你已经完成了这一步。
- (2) 下载并安装 Android SDK：SDK 包含了用于 Android 开发的核心资源。
- (3) 配置 Eclipse：在进行任何操作之前需要为 Android 安装 Eclipse 插件。
- (4) 安装原生开发工具包(NDK)：这是任何一种使用类似 OpenGL 原生 API 的游戏的关键组件。在编写本书时，NDK 的最新版为 r6b，但是 Android 4 能够兼容老版本 NDK 编译出的二进制文件。这意味着老版本的 NDK 也能在 Android 4 上运行。不过，使用最新的