

移动与嵌入式开发技术

使用动手实例创建2D和3D Android游戏app



Android 4 游戏实战编程

Practical Android 4 Games Development

[美] J.F. DiMarzio 著
张 龙 译



Apress®



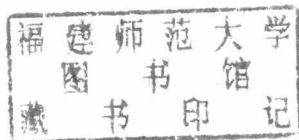
清华大学出版社

移动与嵌入式开发技术

Android 4 游戏实战编程

[美] J. F. DiMarzio 著

张 龙 译



1051447



T1051447

清华大学出版社

北 京

J. F. DiMarzio

Practical Android 4 Games Development

EISBN: 978-1-4302-4029-7

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 © 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: 01-2012-3838

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 游戏实战编程/(美) 蒂马尔奇奥(DiMarzio, J. F.) 著; 张龙 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: Practical Android 4 Games Development

ISBN 978-7-302-30498-2

I. ①A… II. ①蒂… ②张… III. ①移动终端—游戏程序—程序设计—教材 IV. ①TP929.53

中国版本图书馆 CIP 数据核字(2012)第 256981 号

责任编辑: 王 军 杨信明

装帧设计: 牛艳敏

责任校对: 成凤进

责任印制: 王静怡

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 刷 者: 清华大学印刷厂

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 18.75 字 数: 462 千字

版 次: 2013 年 1 月第 1 版 印 次: 2013 年 1 月第 1 次印刷

印 数: 1~3500

定 价: 48.00 元

译者序

目前的移动开发领域呈现出三足鼎立的态势，即苹果的 iOS、Google 的 Android 以及微软的 WP。其中 Android 的发展速度是相当迅猛的，据不久前 Google 发布的统计数据表明，到 2012 年中，全球已激活的 Android 设备高达 4 亿台。国内外各大电脑与手机厂商也在不断发力，如三星、摩托罗拉、中兴、华为、联想、HTC 等都发布了自己的基于 Android 系统的手机与平板电脑。

鉴于 Android 发展势头如此之猛烈，市场上 Android 相关的图书也如雨后春笋般涌现。本书也是在这个潮流下应运而生。与其他讲解 Android 开发基础知识的图书不同的是，本书重在介绍如何通过 OpenGL ES 等技术开发基于 Android 平台的游戏。

全书分为两大部分共 12 章，其中第一部分涵盖 1~9 章，第二部分涵盖 10~12 章。第一部分重点介绍如何通过 OpenGL ES 开发基于 Android 平台的 2D 游戏，同时也详细讲解了游戏规划、开发的一般过程和游戏背后的故事情节，这些内容对于一款成功的游戏来说是不可或缺的。第二部分则将之前所介绍的知识应用到 3D 游戏的开发当中，讲解了如何创建 3D 环境与 3D 游戏。读者在学习完本书后就可以开发属于自己的 2D 游戏与 3D 游戏并部署到实际设备上去。

除了游戏创建的知识外，本书还介绍了诸如碰撞检测、人工智能等游戏开发中的热门内容。这些知识的学习对于提升自己的游戏开发能力来说是至关重要的。除了理论知识的讲解外，本书还提供了大量的源代码供读者练习，读者可以在编写完代码后与书中代码进行比对，找出不一致的地方加以修改并最终成功运行游戏。

翻译技术书籍是一项艰苦的劳动，在这里我要将我最真挚的谢意送给我的妻子张明辉，正是她无微不至的关怀与照顾才能让我忘却生活中的琐事而专心于翻译。

感谢清华大学出版社的李阳老师，认识你是我的荣幸，你的专业与耐心都给我留下了深刻的印象，也非常感谢你的理解与包容。

本人从事 Android 与 iOS 相关开发工作已有 3 年时间，期间积累了不少经验，深谙移动开发的精髓。曾翻译过《Android Web 应用高级编程》、《iPhone SDK 编程入门经典：使用 Objective-C》、《Xcode 3 高级编程》、《iPhone 游戏开发》等书籍。本人目前担任 InfoQ 中文站翻译团队编辑、满江红开放技术研究组织成员，同时拥有 5 年以上的培训讲师经历。

对于译者来说，能将英文转换为中文并给读者带来切实的帮助是我最大的荣幸。因此，若读者在阅读过程中发现问题，请不吝赐教。鉴于译者水平有限，失误和遗漏之处在所难免，恳请读者批评指正。敬请广大读者提供反馈意见，读者可以将意见反馈到 zhanglong217@yahoo.com.cn，我会仔细阅读读者发来的每一封邮件，以求进一步提高今后译著的质量。译者的博客是：<http://blog.csdn.net/ricohzhanglong>，新浪微博是：<http://weibo.com/fengzhongye>，欢迎访问。

作者简介

J. F. DiMarzio 是资深的 Android 开发人员与作家。他早在 1984 年就开始在 TRS-80 彩色计算机上使用 Basic 开发游戏。从那时起，DiMarzio 曾在很多公司的技术部门工作过，如美国国防部与迪斯尼公司等。从 0.03 Beta 版开始，他就在 Android 平台上进行开发工作，并且在 Android 市场上发布了两款专业应用与一款游戏。

技术编辑简介

Yosun Chang 从 2009 年初就开始编写 iOS 与 Android 应用了，目前在一家名为 nusoy 的初创公司工作，该公司从事下一代 3D 游戏与增强现实移动游戏的开发。在这之前，她从 1999 年开始就从事 LAMP 上的 Web 开发与 Flash 工作。她还使用自己的艺名 Ina Centaur 在多个虚拟世界、剧院与增强现实大会上做过演讲。她在加州大学圣地亚哥分校与加州大学伯克利分校分别获得了物理学与哲学研究生学位。作为一名热心读者，她从众多的技术图书中学到了很多，比如现在这本书，她仔细阅读了自己审阅章节的每个单词并且审查了源代码。读者可以通过@yosunchang 在 Twitter 上找到她。

Tony Hillerson 是 EffectiveUI 公司的软件架构师。他毕业于大使大学并获得管理信息系统的文学学士学位。Tony 每天的工作会用到 Android、Rails、Objective-C、Java、Flex 及 Shell 脚本。从早期的 Beta 版开始他就对 Android 开发产生了浓厚的兴趣。Hillerson 录制过 Android 视频、在很多会议上做过 Android 专题演讲、并且作为技术审校者审校过多本 Android 书籍。他有时也会自己编写 Android 代码。他对于各种可用性与体验设计都很有兴趣，从数据库到服务端再到眼镜。在闲暇时间，Hillerson 喜欢演奏贝斯、玩魔兽世界和创作电子音乐。Tony 和他的妻子及两个儿子住在科罗拉多的丹佛市郊区。

游戏图形设计者简介

Ben Eagle 从事计算机图形与 Web 开发已有 14 年的时间，这是他在海军陆战队服役期间习得的技术。他曾在多家公司工作过，Ben 曾设计过数百个网站、公司标志、商标、商业广告和营销图案。目前他是一名高级程序员，住在佛罗里达州达文波特。在 34 岁时，他继续从事着自己的事业并在业余时间向学生传授图形学知识。他获得了数字媒体与 Web 开发这两个学位。Ben 还取得了 MCP 与 C++/Java 认证。在闲暇时间，他会继续从事计算机艺术与编程方面的工作，并在乐队中演奏。

致 谢

本书的面世得益于很多人的帮助：我的代理人 Neil Salkind 与 Studio B 的所有人；Steve Anglin、Corbin Collins、James Markham、Yosun Chang、Tony Hillerson 以及 Apress 的朋友；Ben Eagle 绘制了游戏中的图形；DWSS 的 MD、JS、CL、DL、MB、JK、CH、BB、DB 与 KK；此外还有那些这里可能忽略掉的一路以来帮助过我的朋友们。

前 言

欢迎阅读《Android 4 游戏实战编程》。本书将会从概念到代码一步步介绍两个不同的移动游戏的演变过程。读者会学习到如何从一个基本想法来构思出一个游戏并完成编写引擎的复杂任务以将想法转变为真正的游戏。

编写这本书的目的在于介绍在 Android 4 平台上创建 2D 游戏与 3D 游戏所需的技能。Android 4 将基于 Android 的移动电话与平板电脑上的操作系统统一为一个通用的 SDK。这意味着你所开发的游戏可以运行在最新的平板电脑与手机上,可以运行在最棒的硬件之上。现在,同样的游戏可以运行在这两种设备之上;只需要迈出第一步并创建一个引人入胜的游戏即可。

当完整支持 OpenGL ES 2D 与 OpenGL ES 3D 的 Android SDK 发布后,作者很快发现自己想要创建令人瞩目且好玩的游戏。那时就意识到创建这些游戏所需的技能(虽然掌握起来并不是那么困难)靠自己来探索并不是那么容易的事情。事实上,如果之前没有 OpenGL 特别是 OpenGL ES 方面的经验,那么进入到一般的 Android 游戏开发领域将是非常困难的。

本书决定利用之前在开发 Android 休闲游戏时所习得的技能,将其划分成核心的基本技能集,读者可以在游戏开发的过程中轻松掌握并扩展这些技能。可能无法通过这些技能在学习完本书后创建出下一个红色兵团:大决战游戏,但你可以通过其了解一些必备的知识,从而理解这些游戏是如何创建出来的,并且通过自己的奋斗和实践将其创建出来。

毫无疑问,读者的脑海中肯定已经有了第一个 Android 游戏的样子。你知道它的样子、知道怎么玩这个游戏。不知道的是如何将脑海中的想法变成现实,呈现在手机或平板电脑上。虽然拥有游戏的想法是很棒的,但将这个从想法阶段变成“可在移动设备上玩”这个阶段却是最棘手的部分。

在阅读本书时,作者的建议是保持想法的简单性。不要因为可以就将一款好游戏变得过于复杂。这里的意思是一些令人“上瘾”的游戏并不一定很复杂。这些游戏很容易上手,玩起来也很轻松,但却很难玩通关。在构思游戏概念时请谨记这一点。本书将会介绍一个简单的引擎,它用于实现一个卷轴射击游戏。卷轴射击游戏是个很简单的游戏类型,可以包含非常难且极具挑战性的游戏。长久以来,这类游戏都被认为是非常容易令人着迷的街机风格的游戏,因为它具备快速的动作和几乎无限量的玩法。重复进行卷轴射击游戏是很容易的,而且能够获得胜利的游戏体验。这正是本书选择这种风格的游戏作为第一个示例的原因所在。最后,如果读者想开发出自己想玩的游戏,那么经验就很有价值了。作者希望本书读者能充分享受 Android 游戏开发的旅程。

目 录

第 I 部分 规划与创建 2D 游戏

第 1 章 欢迎进入 Android 游戏世界	3
1.1 Android 游戏编程	3
1.2 从一个好故事开始	4
1.2.1 故事缘何重要	5
1.2.2 编写你的故事	6
1.3 本书的学习方式	8
1.4 获取 Android 开发工具	9
1.5 选择 Android 版本	11
1.6 小结	12
第 2 章 射击游戏“星空战士”	13
2.1 “星空战士”背后的故事	13
2.2 是什么构成了游戏	15
2.2.1 理解游戏引擎	16
2.2.2 理解具体游戏代码	17
2.2.3 探索“星空战士”引擎	19
2.3 创建“星空战士”项目	20
2.4 小结	22
第 3 章 启动：制作菜单	23
3.1 构建启动画面	23
3.1.1 创建活动	24
3.1.2 创建一个新类	24
3.1.3 将类转变为活动	25
3.1.4 创建启动画面图片	29
3.1.5 导入图片	30
3.1.6 处理 R.java 文件	31
3.1.7 创建布局文件	31
3.1.8 编辑 XML 文件	32
3.1.9 使用 FrameLayout	33

3.1.10 添加图片与文本	33
3.1.11 连接 StarfighterActivity 与 布局	36
3.1.12 创建渐变效果	38
3.1.13 线程化游戏	40
3.1.14 创建游戏线程	41
3.1.15 设置新的意图	43
3.1.16 终止活动	44
3.2 创建主菜单	46
3.2.1 添加按钮图片	46
3.2.2 设置布局	48
3.2.3 连接按钮	49
3.2.4 添加 onClickListener	51
3.3 添加音乐	53
3.3.1 创建音乐服务	54
3.3.2 播放音乐	59
3.4 小结	63
第 4 章 绘制环境	65
4.1 渲染背景	65
4.1.1 创建活动	66
4.1.2 创建渲染器	70
4.1.3 使用 OpenGL 加载图片	76
4.1.4 滚动背景	87
4.2 添加第二个层	94
4.2.1 加载第二个纹理	96
4.2.2 滚动第二个层	97
4.2.3 使用矩阵	98
4.2.4 完成 scrollBackground2() 方法	100
4.3 以每秒 60 帧的速度运行	102

4.3.1	暂停游戏循环	103	7.1.3	加载精灵图表	171
4.3.2	清除 OpenGL 缓冲	105	7.2	回顾 AI	172
4.4	修改主菜单	106	7.2.1	创建 moveEnemy()方法	172
4.5	小结	107	7.2.2	创建 enemies[]数组循环	173
第 5 章	创建角色	109	7.2.3	通过 AI 逻辑移动敌人	173
5.1	精灵动画	109	7.3	创建拦截机 AI	174
5.2	加载角色	111	7.3.1	调整顶点	175
5.2.1	创建纹理映射数组	112	7.3.2	锁定玩家位置	176
5.2.2	将纹理加载到角色上	116	7.3.3	实现斜率公式	179
5.2.3	创建游戏循环	119	7.4	创建侦察机 AI	186
5.3	移动角色	121	7.4.1	设置随机点以移动侦察机	188
5.3.1	绘制角色的默认状态	122	7.4.2	沿着贝塞尔曲线移动	189
5.3.2	编写 PLAYER_RELEASE 动作	124	7.5	创建战舰 AI	192
5.3.3	向左移动角色	127	7.6	小结	194
5.3.4	加载正确的精灵	128	第 8 章	防御	195
5.3.5	加载第二帧动画	132	8.1	创建武器精灵图表	195
5.3.6	向右移动角色	134	8.2	为武器设定轨道	199
5.3.7	加载右侧动画	136	8.2.1	创建武器数组	199
5.4	通过触摸事件移动角色	139	8.2.2	添加第 2 个精灵图表	199
5.4.1	解析 MotionEvent	140	8.2.3	初始化武器	200
5.4.2	捕获 ACTION_UP 与 ACTION_DOWN	142	8.2.4	移动武器炸弹	201
5.5	调整 FPS 延迟	144	8.2.5	检测屏幕边界	203
5.6	小结	145	8.2.6	调用 firePlayerWeapons() 方法	205
第 6 章	添加敌人	147	8.3	实现碰撞检测	206
6.1	中局管理	147	8.3.1	应用碰撞检测	206
6.2	创建纹理类	148	8.3.2	创建 detectCollisions()方法	207
6.3	创建敌人类	152	8.3.3	检测特定的碰撞	208
6.3.1	添加新的精灵图表	152	8.3.4	删除无效炸弹	209
6.3.2	创建 SFEnemy 类	154	8.4	扩展所学内容	211
6.3.3	贝塞尔曲线	157	8.5	小结	212
6.4	小结	162	8.6	回顾关键的 2D 代码	212
第 7 章	为敌人添加基本的人工智能	165	第 9 章	发布游戏	233
7.1	为敌人准备好 AI	165	9.1	准备清单	233
7.1.1	创建每个敌人的逻辑	167	9.2	准备签名、对齐及发布	234
7.1.2	初始化敌人	169	9.2.1	检测 AndroidManifest 文件	236
			9.2.2	创建密钥	237
			9.3	小结	239

第 II 部分 创建 3D 游戏

第 10 章 斑点猎人: 创建 3D 游戏 243

10.1 2D 游戏与 3D 游戏的比较 243

10.2 创建 3D 项目 244

10.2.1 BlobhunterActivity.java 244

10.2.2 BHGameView 245

10.2.3 BHGameRenderer 245

10.2.4 BHEngine 247

10.3 创建 3D 对象测试 247

10.3.1 创建常量 248

10.3.2 创建 BHWalls 类 248

10.3.3 实例化 BHWalls 类 251

10.3.4 映射图片 252

10.3.5 使用 gluPerspective()
方法 25310.3.6 创建 drawBackground()
方法 255

10.3.7 收尾工作 257

10.4 小结 259

第 11 章 创建身临其境的环境 261

11.1 使用 BHWalls 类 261

11.2 通过多个 BHWalls 实例创建
通道 262

11.3 使用 BHCORRIDOR 类 263

11.3.1 构建 BHCORRIDOR 类 263

11.3.2 使用 vertices[] 数组构建多个
墙体 265

11.3.3 创建 texture[] 数组 266

11.3.4 创建 draw() 方法 269

11.3.5 添加墙体纹理 273

11.4 调用 BHCORRIDOR 274

11.5 小结 275

第 12 章 在 3D 环境中进行导航 277

12.1 创建控制界面 277

12.1.1 编辑 BHEngine 278

12.1.2 编辑 BlobhunterActivity 279

12.1.3 向前移动玩家 280

12.2 穿过通道 281

12.3 调整玩家的视野 283

12.4 小结 284

12.5 回顾关键的 3D 代码 284

规划与创建 2D 游戏

本书的第 I 部分(第 1 章~第 9 章)将会介绍 2D Android 游戏星空战士的规划与创建过程。该游戏的创建将会遵循一个清晰且合乎逻辑的路径。首先需要规划并编写游戏背后的故事,接下来需要创建游戏的背景,然后会创建玩家与非玩家角色,最后会创建武器系统与碰撞检测。在第 9 章介绍将游戏部署到移动设备上的诸多步骤前,第 8 章结尾会提供第 I 部分中所创建或修改的大部分重要 2D 文件的完整程序清单。可以将这些程序清单与自己的代码进行比较以确保每个游戏都能正确运行。这样也会为第 II 部分“创建 3D 游戏”(第 10 章~第 12 章)中的 3D 开发阶段做好准备。

第 1 章

欢迎进入 Android 游戏世界

本书作者最早是在 2008 年初开始 Android 开发的，当时的平台还是 Beta 版。那时还没有针对这个新操作系统的手机，开发者都觉得自己身处一个令人兴奋的时代的起点。Android 充分具备开源项目开发早期那令人心潮澎湃的一切。在这个平台上开发令人回想起大家在凌晨两点围坐在空荡荡的学生休息室中，摇晃着可乐等待着 VAX 时间来运行最新的代码。它是一个令人兴奋的平台，本书作者有幸见证了它不断成型的过程。

随着 Android 的不断发展，Google 发布了很多更新来确定最终架构，有一件事情变得越来越明显：对于那些非游戏开发者来说，基于 Java 并包含了很多众所周知的 Java 包的 Android 使得他们可以很容易地转向游戏开发。Java 开发者所掌握的大多数知识都可以用到这个新平台上。众多的 Java 游戏开发者都可以利用这些知识平滑迁移到 Android 平台上。

那么 Java 开发者该如何在 Android 上开始游戏开发？需要哪些工具呢？本章旨在回答这些问题。本章将会介绍如何分解一个游戏故事，然后将其实现为游戏。我们还将探索用于完成接下来几章的任务所需的一些必要工具。

本章非常重要，因为它介绍了很多游戏图书并没有涉及的内容——关注于游戏的起源。虽然了解如何编写将游戏引入到实际生活中的代码是非常重要的，但如果游戏没有能够引入到实际生活中，那么游戏的代码也就没什么价值。知道如何以一种清晰的方式从脑海中获悉游戏想法是区分优秀的游戏与无人问津的游戏的重要差别。

1.1 Android 游戏编程

在 Android 上进行游戏开发利弊参半，在开始前就应该意识到这一点。首先，Android 游戏使用 Java 开发，但 Android 并不是一个完全的 Java 实现。使用 OpenGL 与其他图形库所需的众多包都位于 Android 软件开发包(SDK)中。但众多并不意味着所有，对于游戏开发者，特别是 3D 游戏开发者来说，一些颇具价值的包并未包含进来。之前用于构建游戏

所需的一些包可能并不在 Android 中。

随着每一款新版 Android SDK 的发布,越来越多的包被添加进来,旧版本的包可能不建议使用了。读者得清楚使用的是哪些包,本书将在后续章节中介绍这些内容。

另一个优势是人们对 Android 的熟悉程度,劣势则是它缺乏足够的能力。Android 简化了编程,但在速度与能力上却还很缺乏。大多数视频游戏,如为 PC 与控制台所编写的那些游戏都是使用低级语言如 C 语言,甚至是用汇编语言开发的。这样开发者能够最大限度地控制处理器执行代码的方式以及代码所运行的环境。处理器运行着非常底层的代码,距离处理器的原生语言越近,游戏运行所需的跳转解释器就越少。虽然 Android 在底层为代码提供了有限的能力,但它可以自己的执行系统来解释 Java 代码。这样开发者对游戏所运行环境的控制就很少了。

本书并不会介绍游戏开发的底层方式,为什么呢?因为 Java,特别是它可用于通用的 Android 开发,Java 已经成为广泛采用、易于使用的语言,可以创建一些好玩且回报丰厚的游戏。

本质上,如果你是一位经验丰富的 Java 开发者,那么你会发现在进行 Android 开发时,你掌握的那些技能依然可以派上用场。如果你不是一位熟练的 Java 开发者,那也不要担心。Java 是易于上手的优秀语言。出于这个原因,本书选择使用 Android 的原生 Java 开发环境来编写游戏。

这里已经介绍了不少在 Android 上进行游戏开发的优势与劣势。然而,对于独立游戏开发者与非游戏开发者来说,在 Android 平台上创建与发布游戏最大的一个优势在于发布游戏的自由度。虽然一些在线应用商店对于何种应用能够出售,售价是多少有着严格的规定,但 Android Market 却不是这样。每个人都可以列出和销售他们想要的任何应用。这样开发者就拥有了足够的创作自由度。

第 2 章将会创建首个 Android 游戏,这是一个非常简单的游戏。但首先我们得看看幕后,看看为游戏带来灵感的故事,这是非常重要的。

1.2 从一个好故事开始

每款游戏,从最简单的街机游戏到最复杂的角色扮演游戏(RPG)都是从一个故事开始的。故事可能仅仅就是一句话而已,比如说:想象一下我们有一个能够扫射的巨大的宇宙飞船。

然而,故事也可能像一本书一样长,能够描述游戏中的每块陆地、每个人以及每只动物。它甚至可以描述每件武器、挑战与成绩。

说明:

故事概览了动作、目的与游戏流。故事越详尽,代码开发工作就越容易。

看看图 1-1 所示的游戏,这幅图片能说明什么呢?这是来自于“星空战士”游戏的一个屏幕截图;这是从本书的一开始就要开发的一款游戏。这款游戏背后也有一个故事。



图 1-1 星空战士屏幕截图

我们很多人从来都不会去了解那些我们所喜欢的游戏背后的故事，因为故事只对创建游戏的人重要。假如开发者与创建者都能将自己的工作做好，那么玩家就会在游戏中理解游戏背后的故事了。

在小型、独立的开发公司中，除了主力开发者外其他人可能永远都不会了解故事。在大型的游戏开发公司中，故事在传递到主力开发者前会传播给众多的设计者、作者和工程师并由他们对故事进行加工处理。

每个人对想要制作的游戏背后的故事都有不同的编写与创建方式。对于游戏故事来说并没有对错之分，只能说在开始编写代码前需要有故事存在才行。下一节将会介绍为何说故事是非常重要的。

1.2.1 故事缘何重要

无可否认，在视频游戏早期，故事的重要性不如现在。发布一款令人感到愉悦而无须深入挖掘其目的的游戏是很容易的。

现在这种情况已经不复存在了。无论是愤怒的小鸟还是魔兽世界，玩家都希望每个动作都能有一个明确的目的。这种期望甚至都达到了潜意识的境界，但游戏需要吸引住玩家，这样他们才会继续玩下去。这种吸引力就是故事的驱动力。

出于各种各样的原因，游戏背后的故事是非常重要的。我们来看看为何说要在开始编写游戏代码前花些时间来编写故事。

游戏背后的故事之所以重要的第一个原因就是它赋予你一个在开始编码前完全了解游戏的机会，从游戏开始到游戏结束。无论你从事什么职业、是全职游戏开发者抑或仅仅是业余爱好，在这上面花的时间都是值得的。

对于全职游戏开发者来说，在编码与创建游戏的过程中是有薪水的。如果在业余时间创建独立游戏，那么时间可以通过你本可以做的事情来度量：钓鱼、与其他人交流等。无

论如何看待这一点，时间都是确定且有具体价值的，花在游戏编码上的时间越多，其代价也就越高。

如果在开始编写代码前游戏故事还未完全实现，那么读者肯定会陷入重重的问题当中，这些问题会导致你回过头来调整或是完全重写已经写好的代码。这会消耗读者的时间、金钱和精力。

说明：

请确保想法是完整的。想法的每个方面都要经过深思熟虑才行。

作为一名游戏开发者，最讨厌的事情就是被迫回过头来修改已经完成甚至是测试过的代码。理想情况下，代码应该具备足够的可扩展性，可以不费多少力气就能操纵它们——特别是在后面想要增加关卡或是大王的情况下更是如此。然而，可能需要重新编写一些相对简单的代码，比如说角色名或环境的名称，或是需要修改较多的代码。比如说，游戏开发者或许发现永远也不会将完成游戏所需的武器给予主角角色，因为在开始构建游戏时根本不清楚游戏该如何结束。

在编写代码时，如果游戏具有完整的情节，那么就可以将故事情节直接映射到代码上了。像这样映射游戏及其细节信息能够大大减少遇到的问题，这些问题可能会导致重新编写已经完成的游戏部分。这就是在开始编写代码前为何要有故事的第二个原因。

游戏所依据的故事也可以在编写代码时作为参考资料。无论是否需要回过头来检查角色名或敌对敌人的名称，抑或是改变城市街道的布局，都应该可以从故事中获得所需的信息。

如果多人一起开发游戏，那么能够参考故事以获得详细信息是至关重要的。游戏开发者可能尚未完成故事中某几个章节的编写。如果所编写的代码需要参考这些章节，那么完整的故事文档对于你来说就是无价的参考资料。

按照这种程度编写故事意味着多人可以参考同样的资料，他们对于所要完成的事情具有同样的标准。如果有多人在一个协作环境下一起工作，那么每个人都向同一个方向前进就是至关重要的了。如果每个人都按照自己脑海中游戏的样子来编写代码，那么每个人所编写的代码就都是不同的了。编写优秀的故事(可以被从事游戏开发的每个开发者所参考)将有助于让团队向共同的目标前进。

但如何将脑海中的想法形成故事并为你自己和他人提供参考呢？下一节将来回答这个问题。

1.2.2 编写你的故事

编写故事并没有什么窍门可言。只要你觉得适合，无论是精心制作还是较为粗糙都是可以的。从电脑旁的便签纸上的几个句子到格式良好的几页微软 Word 文档都行。要点在于不要像图书那样编写故事；而只是想将脑海中的想法变成一种清晰的格式，大家可以参考并希望不要修改。

故事在脑海中停留的时间越久，修改其细节的时间就越长。在修改故事中的任何细节时都会面临着重写代码的风险(我们已经介绍过这么做的危险之处了)。因此，如果只有一

个人，一台开发机器，觉得将故事写下来是没必要的，但是请三思而后行。将故事写下来可以确保不会忘记或是不小心修改了任何细节信息。

毫无疑问，读者的脑海中已经有一个游戏，想在学习完本书所介绍的知识后尽快开始开发。然而，可能并未考虑清楚游戏背后的故事到底是什么。现在来想想这个游戏吧。

提示：

如果脑海中有个游戏，那么现在请花点时间写下游戏的草稿。完成后，请与下面的模拟游戏进行对比。

现在来快速看看可用于游戏开发的一个故事示例。

John Black 从当地偷了一辆速度快且结实的轿车。一帮坏人很快就追上了他。现在，他不得不从 Villiansburg 逃出来，要带着钱、不能碰到警察并且要与被他偷了钱的那帮家伙火拼一场。这帮家伙的车速度更快，但幸好，John 可以在开车的时候开枪，希望他能逃过这一劫。

在这个故事中，虽然细节信息不多，但对于普通的开发者来说依然可以据此开发出简单的游戏。读者能从上面这段话中获悉到什么呢？

从这个简短的故事中我们首先能想到的是一个自上而下、街机风格的赛车游戏；想想最初的间谍猎人游戏。驾驶员或是汽车可以向敌人的车辆开火。当玩家抵达城镇的边缘、安全屋或是某个车库时，游戏宣告结束。

这个简短的故事甚至拥有足够的细节信息使得游戏更具可玩性。主角的名字是 John Black。需要避开两类敌人：警察与坏蛋。环境是由 Villiansburg 街道构成的，大多数敌人的车辆要比主角的快。毫无疑问，我们有足够多的好素材来开发出一个快速、常见的游戏。

脑海中隐喻的车轮应该开始构思对于这个游戏的想法了。上面这段简短的文字描述出了相当多优秀、街机风格游戏的动作。作为一名独立的普通开发者，如果能像这样通过一段话描述出想要制作的游戏，那就说明你已经走上了制作具备极佳可玩性的游戏之路了。

即便一小段文字都能拥有足够的细节信息来描述颇具吸引力的游戏，何况更长的故事。向游戏中添加的细节信息越多，编写代码时就会越轻松，游戏质量也会更棒。

花点时间完善故事，提供适当的细节信息。一段文字(就像本节上面的那样)就足以让读者继续前进，但毫无疑问，更多的细节信息可以在读者编写代码时提供更多的帮助。下面是阅读完这个故事后可能会问自己的一个问题列表：

- John 偷窃并驾驶的是什么样的轿车？
- 他为何要偷钱？
- 他有何种武器？
- 轿车上有何种武器？
- Villiansburg 是城市还是乡村？
- 最后是否有大王之战？
- 如果有的话，分数是如何累计的？

如果回过头去回答这些问题，那么故事可能会像下面这样。