

CENGAGE
Learning

2D游戏开发的优秀教程
内容全面，示例实用，覆盖必备核心技能



Game Design and Develop

游戏设计与开发



Java 2D游戏

编程入门

FUNDAMENTAL 2D GAME
PROGRAMMING WITH JAVA

[美] Timothy Wright 著 李强 裴云 译

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

CENGAGE
Learning

2D游戏开发的优秀教程
内容全面，示例实用，覆盖必备核心技能



Game Design and Develop

游戏设计与开发



Java 2D游戏

编程入门

FUNDAMENTAL 2D GAME
PROGRAMMING WITH JAVA

[美] Timothy Wright 著 李强 裴云 译

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目 (C I P) 数据

Java 2D游戏编程入门 / (美) 莱特 (Wright, T.) 著;
李强, 裴云译. — 北京: 人民邮电出版社, 2015. 7
ISBN 978-7-115-39277-0

I. ①J… II. ①莱… ②李… ③裴… III. ①游戏—
JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第106147号

版权声明

Fundamental 2D Game Programming with Java

Timothy Wright

Copyright © 2015 Course Technology, a part of Cengage Learning.

Original edition published by Cengage Learning. All Rights reserved.

本书原版由圣智学习出版公司出版。版权所有, 盗印必究。

Posts & Telecom Press is authorized by Cengage Learning to publish and distribute exclusively this simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书中文简体字翻译版由圣智学习出版公司授权人民邮电出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可, 不得以任何方式复制或发行本书的任何部分。

978-7-115-39277-0

Cengage Learning Asia Pte. Ltd.

151 Lorong Chuan, #02-08 New Tech Park, Singapore 556741

本书封面贴有 Cengage Learning 防伪标签, 无标签者不得销售。

-
- ◆ 著 [美] Timothy Wright
译 李强 裴云
责任编辑 陈冀康
责任印制 张佳莹 焦志炜
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 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: 35.25
字数: 672 千字 2015 年 7 月第 1 版
印数: 1-2 500 册 2015 年 7 月北京第 1 次印刷
著作权合同登记号 图字: 01-2015-2397 号
-

定价: 79.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

内容提要

学习 2D 游戏开发是快速积累游戏开发经验的关键。理解了 2D 环境的要素，就能够为游戏开发打下坚实的基础。

本书讲解使用 Java 进行 2D 游戏开发的基础知识和基本技能。本书共 18 章，分为基础知识、提高技能和完整游戏 3 个部分，详细介绍应用编程、全屏游戏、输入处理、矩阵变换、基础物理、相交测试、碰撞检测等知识点和技能，最后采用介绍的所有概念，从头到尾开发了一款完整的游戏。

本书适合游戏开发的初学者阅读，尤其适合想要学习 2D 游戏编程基础的人。本书假设读者理解核心编程概念、面向对象软件以及 Java 编程语言，但不需要读者具备任何游戏编程的知识。

致谢

首先，我要感谢 www.gamedev.net 的所有论坛版主。你们真诚而有帮助评论使得我有勇气展示自己并写好本书。

特别感谢 Drew Sikora 出版了我早期的教程，并且在时日艰难的时候发来了读者的邮件，鼓励我继续前行。感谢 John Hattan 和 Jason Zink 提供的出版信息。感谢审阅过本书、帮助调试代码示例，以及修正所有拼写错误的每一个人。我还想感谢 Emi Smith 以及 Cengage 团队的其他人，感谢他们帮助编辑、出版本书，并且努力搞清楚如何用他们的 Macs 和我的 Windows 协同工作。我还要感谢我的编辑 Kezia Endsley，他容忍了我所有的傻里傻气的问题，并且总是乐于解决这些问题。我还要感谢 Dustin Clingman，他做了很好的技术编辑工作，保证了代码示例的正确性和清楚明了。

最重要的，我要感谢我的妻子 Jimmi。20 年来，她是我最好的朋友，没有她的持续帮助和信任，这一切都不可能。我爱你，我的宝贝！感谢你在本书完成之前没有约我出去。

作者简介

Timothy Wright 是一名软件工程师，目前正在从事下一款美妙的独立游戏开发。在成人生涯的早期，他是一名军队的钢琴演奏师，从那时候起，他就发现自己热爱编程。他拥有计算机科学的学士学位，并且曾经在公司的研发部门工作了近 10 年时间。在开发游戏或玩游戏以外的时间，他喜欢打扑克、做饭和自己酿造啤酒。他和妻子、两个女儿、3 只狗以及一名外国交换学生住在一起，每一周的生活都充满了戏剧性。Timothy 几乎在美国各地居住过，并且花了 5 年时间在意大利、德国居住和在欧洲旅行。多年来，他是 www.gamedev.net 的一名论坛版主，你偶尔可以看到他以 Glass_Knife 为名字在论坛上冒泡。

前言

多年前，当我第一次将软件开发作为专业工作的时候，有人请我编写一个 applet。那时候，我对于 Java 语言知道得并不多。在整个上学期间，我很广泛地使用 C++。我确实用 Java 编写过一些代码，但认为它太慢并且是 C++ 的没落版。

同时，我购买和阅读了很多可以接触到的游戏编程图书。我通读了一本关于人工智能的书，其中包含很多不错的示例，但它们都是用 C++ 和 DirectX 编写的。由于忙着学习 Java 以便在工作中使用，我认为将示例转换为由 Java 编写可能是学习这门语言的一种好办法。

毕竟，Java 游戏编程又会有多难呢？

很快我发现，关于用 Java 编写游戏，可以获取的信息实在太少了。我找到的那些信息，都是关注于 apple 编程的，实际上并没有介绍如何制作应用程序类型的游戏。在将示例转换为 Java 版并发现这可能并不是很难之后，我开始搜索关于这一切内容的教程。我没有找到这样的教程，所以我自己编写了一本。

在 gamedev.net 上发布了两本教程并且从头开始用 Java 编写了软件渲染程序之后，我意识到 Java 这种语言给游戏社群提供了很多东西。在开发下一个系列教程的同时，我最终得到了很多的信息，使得编写一本书成为一种更好的解决方案。

这就是你现在所读到的书，它是一个开发者的代码工具集和说明，它的目的是帮助你加快学习 Java 游戏编程的过程。希望后面的内容可以解答所有必须的问题，并且帮助你学习重要的内容：开发游戏。

本书的组织结构

本书分为 3 个部分。第 1 部分是“基础知识”，覆盖了创建一个简单的原型游戏所需要的所有概念。第 2 部分是“提高技能”，介绍了编写一款完整的游戏所需要的其他概念。最后一个部分是“完整游戏”，采用介绍的所有概念，从头到尾开发了一款完整的游戏。内容的概览如下。

第 1 部分：基础知识

第 1 章 Hello World。本章创建了一个帧速率类，使用它来监控应用程序的速度。然后，介绍了窗口和全屏游戏循环。

第 2 章 输入。本章创建了类来监控键盘输入，以及绝对和相对鼠标输入。示例代码展示了一个游戏循环中的输入处理。

第3章 变换。本章介绍了向量、点和矩阵数学，以及点和向量的平移、旋转、缩放和切变；探讨了矩阵变换以及手动方法，还介绍了 Swing 的变换对象。

第4章 时间和空间。本章给循环添加了一个游戏时钟，以允许独立于帧速率的对象操作；还介绍了屏幕映射，允许从标准设备坐标映射任何的屏幕—空间对象。

第5章 简单游戏框架。本章使用前面展示的所有代码，创建了一个简单的游戏框架，以供在后面的示例中使用。

第6章 Vector2f 更新。本章更新了 Vector2f 类，这个类是在第3章中初次介绍的，它拥有进行相交测试所必需的所有线性代数代码。

第7章 相交测试。本章介绍了确定对象是否重叠或者是否在另一个对象之中所必需的数学和算法。点、圆、直线和矩形都被用来探讨这一话题。

第8章 游戏原型。本章使用目前为止给出的代码，配合简单框架，创建了一款原型游戏。尽管这还不是一款完整的游戏，但它展示了创建一款游戏所必需的所有代码的思路。

第2部分：提高技能

第9章 文件和资源。本章介绍了读取和写入 XML 文件，以及加载和保存 Java 属性文件。本章还介绍了访问类路径上的存档文件中所包含的资源。

第10章 图像。本章介绍了如何操作图像，从头创建图像，以及操作图像文件。本章还介绍了透明度的使用，以及用来绘制、旋转和缩放图像的不同方法。

第11章 文本。本章介绍了使用 Java 的 Font 库绘制文本所需的所有内容，还对输入库进行了更新，以支持录入。

第12章 线程。本章介绍了 Java 中可用的多线程库，并且展示了创建一个非阻塞的类、用一个阻塞类包装非阻塞的类，将多线程代码集成到一个单线程游戏循环中，以及使用一个线程库来加速资源加载。

第13章 声音。本章介绍了 Java 中支持声音的问题。既探讨了小的剪辑文件，也探讨了大文件的流播放，还有使用声音控件，例如平衡和音量等。本章中所开发的声音包装库，用于将声音集成到游戏循环之中，这要用到第12章所介绍的概念。

第14章 用 ANT 进行开发。本章对 ANT 构建工具提供了一个概览，并且带领你学习为游戏开发创建一个可扩展的构建脚本的过程。

第15章 碰撞检测。本章介绍了如何确定碰撞物体的交点，以及如何处理碰撞响应。本章通过学习很多的示例探讨了点、直线、矩形和圆。

第3部分：完整游戏

第16章 工具。本章介绍了制作完整游戏所需的工具。针对窗口游戏、全屏游戏开发了框架，还有 Swing 编辑器。工具库进行了更新，创建了一个简单的 Swing 编辑器来绘

制多边形并且将其保存为 XML。创建了一个 `sprite` 类，还开发了一个简单的粒子引擎。

第 17 章 太空火箭。本章使用所给出的所有代码，从头到尾开发了一款完整的游戏。尽管这款游戏很短并且很容易，但它很完备，带有加载屏幕、观赏模式、游戏进程、高分输入和保存以及一个部署 ANT 脚本。了解创建一款完整游戏所需的工作，将会激发你完成自己的游戏。

第 18 章 结论。本章列出了所有的代码示例，从而圆满结束了本书。本书给出了各章代码示例的列表，并且每个示例都解释了所学的内容，使你可以很容易地找到一个示例。

本书的目标读者

本书写给想要学习 2D 游戏编程基础的人。

尽管当前的图形发展非常快，但 3D 图形还只是迷宫的一块。如果一个程序员不理解游戏循环、输入处理、矩阵数学、相交测试、碰撞检测、图像渲染、线程、声音或者部署，只是 3D 图形也无法让游戏变得更好。即便 3D 渲染令人激动，但学习基础知识对于打下牢固基础来说仍很重要，并且处理任何情况都需要掌握工具。

本书假设读者理解核心编程概念、面向对象软件以及 Java 编程语言，但并不假设你需要有游戏编程的知识。

你应该已经熟悉 Java 语言了，本书并不会介绍它。

软件和版本

本书代码在 Windows 7 上、使用 Java SE 6.0 和 Java SE 7.0（编写本书时可用的最新版本）进行过测试。使用 Eclipse Helios IDE 进行代码的编译和执行。使用 1.8.2 ANT 版本进行部署。

示例源代码、声音、图像和资源文件，可以通过如下链接获取：

<http://www.indiegameprogramming.com>

也可以从本书的 Web 站点下载它们，可以使用书名或者 ISBN 来搜索：

<http://www.cengageptr.com/downloads>

本书体例

对于以下内容使用代码字体：

- Java 类；
- Java 方法；
- 变量名称；
- 数学变量和常量；
- 代码示例；
- 代码段；
- XML；
- 命令行输出。

问题和反馈

请将关于本书的任何评论、勘误或问题，发送到如下的 Email 地址：

indiegameprogramming@gmail.com

本书的 Web 站点上有相关的信息、更新、勘误和源代码，可以通过如下地址访问：

<http://www.indiegameprogramming.com>

目录

第 1 部分 基础知识

第 1 章 Hello World	2	4.3 调整视口高宽比	93
1.1 使用 <code>FrameRate</code> 类	2	4.4 大炮实例	100
1.2 创建 Hello World 应用程序	3	4.5 资源和延伸阅读	107
1.3 使用主动渲染	6	第 5 章 简单游戏框架	108
1.4 创建定制的渲染线程	8	5.1 屏幕到世界的转换	108
1.5 创建一个主动渲染的窗口	11	5.2 理解简单框架	110
1.6 修改显示模式	13	5.3 使用简单框架模板	116
1.7 全屏显示模式中的主动渲染	18	5.4 资源和延伸阅读	118
1.8 资源和延伸阅读	21	第 6 章 Vector2f 更新	119
第 2 章 输入	22	6.1 <code>inv()</code>	119
2.1 处理键盘输入	22	6.2 <code>add()</code>	119
2.2 键盘改进	26	6.3 <code>sub()</code>	120
2.3 处理鼠标输入	29	6.4 <code>mul()</code>	120
2.4 相对鼠标移动	37	6.5 <code>div()</code>	121
2.5 资源和延伸阅读	45	6.6 <code>len()</code> 和 <code>lenSqr()</code>	121
第 3 章 变换	46	6.7 <code>norm()</code>	122
3.1 使用 <code>Vector2f</code> 类	46	6.8 <code>perp()</code>	123
3.2 使用极坐标	54	6.9 <code>dot()</code>	123
3.3 理解点和向量	60	6.10 <code>angle()</code>	125
3.4 使用矩阵变换	62	6.11 <code>polar()</code>	125
3.5 行主序矩阵和列主序矩阵	63	6.12 <code>toString()</code>	126
3.6 理解 <code>Matrix3x3f</code> 类	65	6.13 资源和延伸阅读	128
3.7 仿射变换	76	第 7 章 相交测试	129
3.8 资源和延伸阅读	79	7.1 多边形中的点的测试	129
第 4 章 时间和空间	80	7.1.1 多边形特例	
4.1 计算时间增量	80	中的点	131
4.2 屏幕映射	86	7.1.2 多边形示例	

中的点	133
7.2 使用轴对齐边界框进行 相交测试	137
7.3 使用圆形测试相交	139
圆和 AABB 的重合	140
7.4 使用分隔轴方法	147
7.5 使用线段—线段的重叠方法	149
7.6 使用矩形—矩形的重叠方法	152
7.7 优化测试	156
7.8 资源和延伸阅读	157

第 2 部分 提高技能

第 9 章 文件和资源	198
9.1 理解 Java 如何处理文件和 目录	198
9.2 理解输入/输出流	201
9.3 创建 Resources.jar 文件 进行测试	206
9.4 将资源放到类路径上	208
9.5 制作资源加载工具包	211
9.6 利用 Java 属性	214
9.7 XML 文件概览	218
9.8 资源和延伸阅读	227
第 10 章 图像	228
10.1 学习 Java 中的颜色	228
10.2 了解不同的图像类型	229
10.3 执行颜色插值	235
10.4 使用 VolatileImage 提高速度	240
10.5 创建透明图像	244
10.6 使用 alpha 合成的规则	247
10.7 绘制精灵	254

第 8 章 游戏原型	158
8.1 创建一个多边形包装类	158
8.2 创建一个原型小行星	166
8.3 创建一个原型编辑器	171
8.4 用原型小行星工厂 生产小行星	175
8.5 原型 Bullet 类	181
8.6 原型 Ship 类	183
8.7 编写原型游戏	188
8.8 资源和延伸阅读	195
10.8 探索不同的缩放算法	264
10.9 资源与延伸阅读	280
第 11 章 文本	282
11.1 理解 Java 字体	282
11.2 制作绘制字符串工具	287
11.3 使用文本度量进行布局	291
11.4 支持线程安全的键盘输入	302
11.5 资源和延伸阅读	309
第 12 章 线程	310
12.1 使用线程实现 Callable 任务	310
12.2 使用线程加载文件	313
12.3 使用 FakeHardware 类测试	316
12.4 使用等待/通知方法	321
12.5 在游戏循环中使用线程	323
12.6 使用状态机	331
12.7 OneShotEvent 类	332
12.8 LoopEvent 类	334
12.9 RestartEvent 类	337
12.10 多线程事件示例	340
12.11 资源和延伸阅读	342

第 13 章 声音	343	14.2 理解构建脚本的格式	387
13.1 操作声音文件	345	14.3 学习常见 ANT 任务	389
13.2 使用声音库的问题	347	14.4 构建一个可扩展的 构建脚本	392
13.3 开发阻塞音频类	351	14.5 资源和延伸阅读	398
13.4 用阻塞的 Clip 类	353	第 15 章 碰撞检测	399
13.5 使用 AudioDataLine 类	357	15.1 带碰撞检测的弹球	399
13.6 BlockingDataLine 类	363	15.2 使用直线参数方程	404
13.7 创建一个 SoundEvent 类	366	15.3 查找直线—矩形相交	405
13.8 使用 OneShotEvent 类	368	15.4 查找圆—直线相交	412
13.9 使用 LoopEvent 类	369	15.5 查找直线—直线相交	418
13.10 使用 RestartEvent 类	370	15.6 计算一个反射向量	424
13.11 添加声音控件	376	15.7 在一个多边形中弹回 一个点	428
13.12 资源和延伸阅读	385	15.8 资源和延伸阅读	436
第 14 章 用 ANT 进行开发	386		
14.1 安装 ANT 软件	386		

第 3 部分 完整游戏

第 16 章 工具	438	17.9 ShipExplosion 类	497
16.1 创建一个游戏框架	438	17.10 添加游戏常量	498
16.2 更新多边形编辑器	450	17.11 Acme 类	499
16.3 绘制精灵	465	17.12 GameState 类	500
16.4 创建一个简单的粒子引擎	469	17.13 Score 类	501
16.5 资源和延伸阅读	474	17.14 QuickLooper 类	502
第 17 章 太空火箭	475	17.15 QuickRestart 类	503
17.1 Bullet 类	476	17.16 HighScoreMgr 类	505
17.2 PolygonWrapper 类	477	17.17 管理状态	509
17.3 Particle 类	480	17.18 StateController 类	510
17.4 Asteroid 类	482	17.19 CompleteGame 类	511
17.5 AsteroidFactory 类	485	17.20 GameLoading 状态	514
17.6 AsteroidExplosion 类	488	17.21 AttractState 类	520
17.7 Ship 类	490	17.22 PressSpaceToPlay 模式	522
17.8 ShipFactory 类	495	17.23 HighScore 状态	523

17.24	GameInformationState 类	524	第 6 章	Vector2f 更新	543
17.25	LevelStarting 状态	525	第 7 章	相交测试	544
17.26	LevelPlaying 状态	526	第 8 章	游戏原型	544
17.27	GameOver 状态	533	第 9 章	文件和资源	544
17.28	EnterHighScoreName 状态	534	第 10 章	图像	545
17.29	创建构建脚本	539	第 11 章	文本	545
17.30	资源和延伸阅读	540	第 12 章	线程	546
第 18 章	结论	541	第 13 章	声音	546
第 1 章	Hello World	541	第 14 章	用 ANT 进行开发	547
第 2 章	输入	542	第 15 章	碰撞检测	547
第 3 章	变换	542	第 16 章	工具	547
第 4 章	时间和空间	542	第 17 章	太空火箭	548
第 5 章	简单游戏框架	543		资源和延伸阅读	549

第 **1** 部分
基础知识

第 1 章

Hello World

大多数编程图书的第 1 章是创建一个 Hello World 应用程序，它执行一个简单的函数以确认一切都能工作。最简单的计算机游戏是一个黑色背景的窗口，还有一个紧凑的 while 循环，它负责尽可能快地清除屏幕并且重新绘制黑色的背景。添加一个帧速率计算器将可以度量应用程序的性能，并且验证窗口正在重新绘制。本书中的大多数示例都会测量帧速率，因此帧速率计算器是一个很好的起点。

1.1 使用 FrameRate 类

FrameRate 类位于 javagames.util 包中。本书中所开发的工具代码都会放到这个工具包中，随后，我们将把这个包变成一个工具库。这个类用来测量本书中所开发的应用程序的每秒的帧数（frames per seconds, FPS）。FPS 按照“FPS 100”的格式存储为一个字符串。每秒钟都会计算这个值。

在开始测量帧速率之前，需要先调用 initialize() 方法。这个方法把帧速率字符串初始化为 0，并且把最近时间初始化为当前时间（以毫秒为单位）。

System.currentTimeMillis() 调用返回了从 1970 年 1 月 1 日午夜开始的毫秒数。不同的操作系统，测量时间的精度可能不同。例如，一些 Windows 版本只能保证 10 毫秒的精度。

对于每一个渲染的帧，都应该调用一次 calculate() 方法。要计算帧速率，从最近时间减去当前时间，并且将其存储到 delta 变量中。每一帧中帧计数都会增加，并且当 delta 时间超过一秒的时候，会产生新的 FPS。delta 变量很少确切地等于 1 秒钟，因此从 delta 变量减去 1000 毫秒，以略去额外的毫秒数。一旦保存了新的帧速率，就会重置帧计数并且再次开始处理。

```
package javagames.util;  
public class FrameRate {
```



```
private String frameRate;
private long lastTime;
private long delta;
private int frameCount;

public void initialize() {
    lastTime = System.currentTimeMillis();
    frameRate = "FPS 0";
}

public void calculate() {
    long current = System.currentTimeMillis();
    delta += current - lastTime;
    lastTime = current;
    frameCount++;
    if( delta > 1000 ) {
        delta -= 1000;
        frameRate = String.format( "FPS %s", frameCount );
        frameCount = 0;
    }
}

public String getFrameRate() {
    return frameRate;
}
}
```

1.2 创建 Hello World 应用程序

图 1.1 所示的 Hello World 应用程序是第一个游戏窗口的示例。HelloWorldApp 位于 `javagames.render` 包中。除了清除和重新绘制背景，这个窗口中不再渲染其他内容。HelloWorldApp 扩展了 `JFrame` 类，这是 Java 的 `Swing` 库中一个顶级的窗口组件。这个应用程序包含一个 `FrameRate` 对象，该对象用来测量应用程序的帧速率。

由于 `Swing` 库不是线程安全的，因此你应该总是在 `Swing` 事件线程上创建并展示一个 `JFrame`。然而，该程序的 `main()` 方法并不是在事件线程上调用的，因

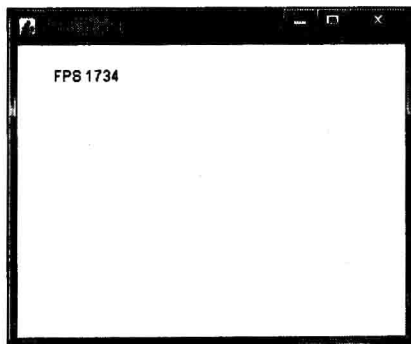


图 1.1 Hello World 应用程序