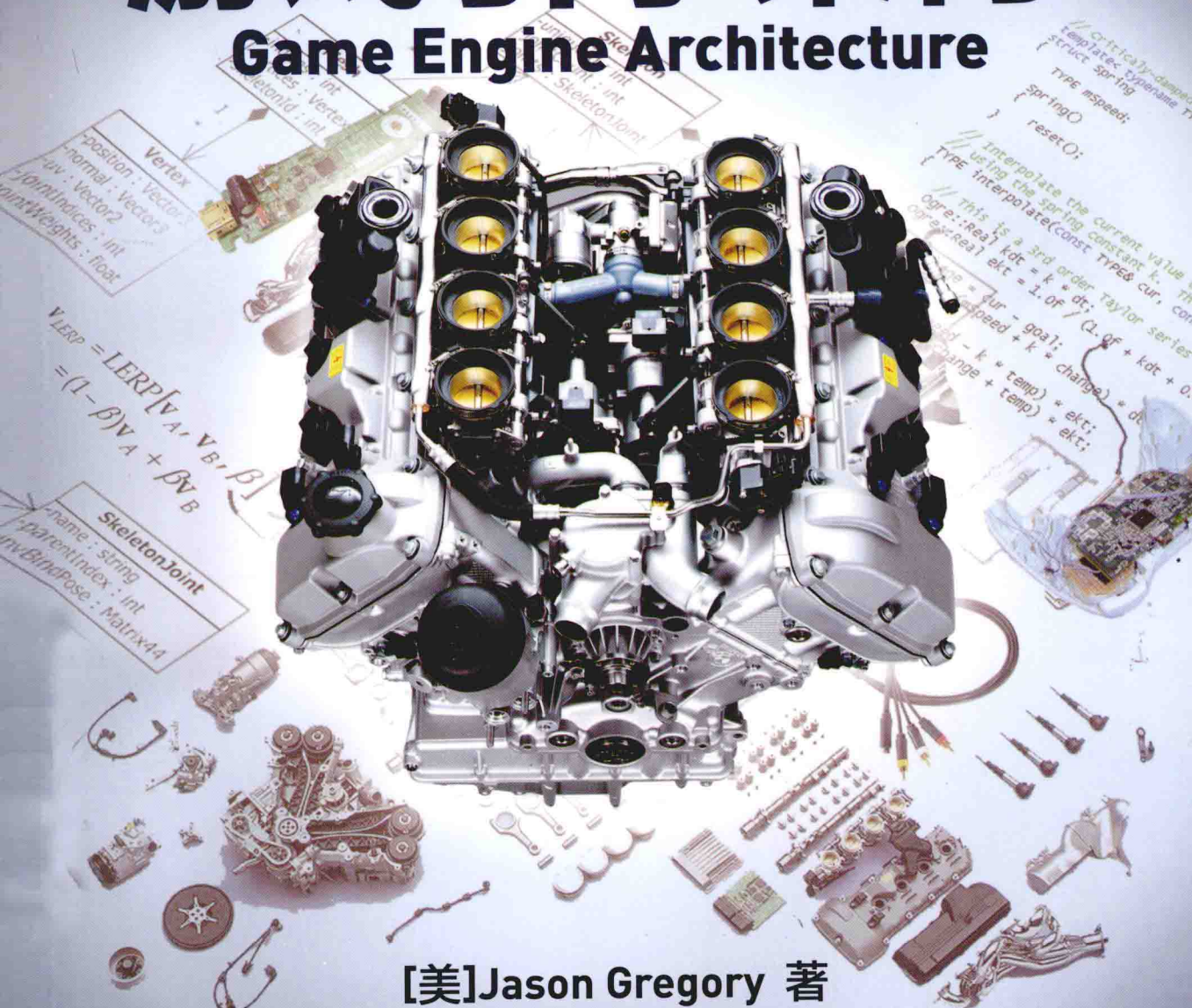


游戏引擎架构

Game Engine Architecture



[美] Jason Gregory 著

叶劲峰 译

游戏引擎架构

Game Engine Architecture

[美] Jason Gregory 著

叶劲峰 译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内容简介

本书同时涵盖游戏引擎软件开发的理论及实践，并对多方面的题目进行探讨。本书讨论到的概念及技巧实际应用于现实中的游戏工作室，如艺电及顽皮狗。虽然书中采用的例子通常依据一些专门的技术，但是讨论范围远超于某个引擎或API。文中的参考及引用也非常有用，可让读者继续深入游戏开发过程的任何特定方向。

本书为一个大学程度的游戏编程课程而编写，但也适合软件工程师、业余爱好者、自学游戏程序员，以及游戏产业的从业人员。通过阅读本书，资历较浅的游戏工程师可以巩固他们所学的游戏技术及引擎架构的知识，专注某一领域的资深程序员也能从本书更为全面的介绍中获益。

Jason Gregory: Game Engine Architecture, First Edition. ISBN: 978-1-5688-1413-1

Copyright ©2009 by A K Peter, Ltd.

Authorized translation from English language edition published by A K Peter, Ltd., part of Taylor & Francis Group LLC; All rights reserved.

Publishing House of Electronics Industry is authorized to publish and distribute exclusively the Chinese (Simplified Characters) language edition. This edition is authorized for sale throughout Mainland of China. No part of the 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.

Copies of this book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal.

本书原版由Taylor & Francis出版集团旗下，A K Peter出版公司出版，并经其授权翻译出版。版权所有，侵权必究。本书中文简体翻译版授权由电子工业出版社独家出版并在限在中国大陆地区销售，未经出版者书面许可，不得以任何方式复制或发行本书的任何部分。

本书封面贴有Taylor & Francis公司防伪标签，无标签者不得销售。

版权贸易合同登记号图字：01-2010-4326

图书在版编目(CIP)数据

游戏引擎架构/(美)格雷戈瑞(Gregory,J.)著;叶劲峰译.一北京:电子工业出版社,2014.2

书名原文:Game engine architecture

ISBN 978-7-121-22288-7

I. ①游… II. ①格…②叶… III. ①三维动画软件—游戏程序—程序设计 IV. ①TP391.41

中国版本图书馆CIP数据核字(2014)第002560号

策划编辑:张春雨

责任编辑:付睿

印刷:北京市新伟印刷有限公司

装订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开本:787×980 1/16 印张:49.75 字数:1093.4千字

印次:2014年3月第3次印刷

定价:128.00元



凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至zltz@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：(010) 88258888。

推荐序1

最初拿到《Game Engine Architecture》一书的英文版，是编辑侠少邮寄给我的打印版。他建议我接下翻译此书的合同。当时我正在杭州带领一个团队开发3D游戏引擎，我和我的同事都对这本书的内容颇有兴趣，两大本打印的英文书立刻在同事间传开。可惜那段时间个人精力顾及不来，把近千页的英文读物精读而后翻译成中文对个人的业余时间是个极大的挑战，不能担此翻译任务颇为遗憾。

不久以后听说Milo Yip（叶劲峰）已开始着手翻译，甚为欣喜。翻译此巨著，他一定是比我更合适的人选。我和Milo虽未曾蒙面，但神交已久。在网络上读过一些他的成长经历，和我颇为相似，心有戚戚。他对游戏3D实时渲染技术研究精深为我所不及，我们曾通过Google Talk讨论过许多技术问题，他都有独到的见解。翻译工作开始后，Milo是香港人，英文技术术语在香港的中文译法和大陆的有许多不同。但此书由大陆出版社出版，考虑到面对的读者主要是大陆程序员，Milo希望能更符合大陆程序员的用词习惯，所以在翻译一开始就通过Google Docs创建了协作页面，邀请大家共同探讨书中技术名词的中译名。从中我们可以一窥他作为译者的慎重。

三年之后，有幸在出版之前就拿到了完整的译本。这是一本用L^AT_EX精心排版的800页的电子书，我只花了一周时间，几乎是一口气读完。流畅的阅读享受，绝对不仅仅是因为原著精彩的内容，精美的版面和翔实的译注也加了不少分。

在阅读本书的过程中，我不只一次地获得共鸣。例如在第5章的内存管理系统的介绍中，作者介绍的几种游戏特有的内存管理方法我都曾在项目中用过，而这是第一次有书籍专门将这些方法详尽记录；又如第11章动画系统的介绍，我们也同样在3D引擎开发过程中改进原有动画片段混合方法的经历。虽然书中介绍的每个技术点，都可能可以在某篇论文，某本其他的书的章节，某篇网络blog上见过，但之前却无一本书可以把这些东西放在一起相互参照。对于从事游戏引擎开发的程序员来说，了解各种引擎在处理每个具体问题时的方案是相当重要的。而每种方案又各有利弊，即使不做引擎开发工作而是在某一特定游戏引擎上做游戏开发，从中也可以理解引擎的局限性以及可能的改进方法。尤其是第14章介绍的对游戏性

相关系统的设计，各个开发人员几乎都是凭经验设计，很少见有书籍对这些做总结。对于基于渲染引擎做开发的游戏程序员，这是必须面对的工作，这一章会有很大的借鉴意义。

本书作者是业内资深的游戏引擎开发人，他所参与的《神秘海域》和《最后生还者》都是我的个人最爱。在玩游戏的过程中，作为游戏程序员的的天性，自然会不断地猜想各个技术点是如何实现的，背后需要怎样的工具支持。能在书中一一得到印证是件特别开心的事情。作者反复强调代码实践的重要性，在书中遍布着C++代码。我不认为这些代码有直接取来使用的价值，但它们极大地帮助了读者理解书中的技术点。书中列出的顽皮狗工作室用lisp方言作为游戏配置脚本的范例也给我很大的启发，有了这些具体的代码示例以及作者本身的一线工程师背景，也让我确信书中那些关于主机游戏开发相关等，我所没有接触过的内容也都绝非泛泛而谈。

国内的游戏开发社区的壮大，主要是随最近十年的MMO风潮而生。而就在大型网络游戏在中国有些畸形发展，让这类游戏偏离电子游戏游戏性的趋势时，我们有幸迎来了为移动设备开发游戏的大潮。游戏开发的重心重新回到游戏性本身。我们更需要去借鉴单机游戏是如何为玩家带来更纯粹的游戏体验，我相信书中记录的各种技术点会变的更有帮助。

云风 @简悦云风

推荐序2

在我认识的许多游戏业开发同仁中，只有少数香港同胞，Milo Yip（叶劲峰）却正是这样一位给我印象非常深刻的优秀香港游戏开发者。我俩认识，是在Milo加入腾讯互动娱乐研发部引擎技术中心后，说来到现在也只是两年多时间。其间，他为人谦逊务实，对待技术问题的严谨求真态度，对算法设计和性能优化的娴熟技术，都为人所称道。Milo一丝不苟的工作风格，甚至表现在对待技术文档排版这类事情上（Milo常执著地用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 将技术文档排到完美），我想这一定是他在香港读大学、硕士及在香港理工大学的多媒体创新中心从事研究员，一贯沿袭至今的好作风。

我很高兴腾讯游戏有实力吸引到这样优秀的技术专家；即使在其已从上海迁回香港家中，依然选择到深圳腾讯互动娱乐总部工作。叶兄从此工作日每天早晚过关，来往香港和深圳两地，虽有舟车劳顿，但是兼顾了对家庭的照顾和在游戏引擎方面的专业研究，希望这样的状况是令他满意的。

认识叶兄当时，我便知道他在进行Jason Gregory所著《游戏引擎架构》一书的中译工作。因为自己从前也有业余翻译游戏开发有关书籍的经历，所以我能理解其中的辛苦和责任重大，对叶兄也更多一分钦佩。我以为，本书以及本书的中文读者最大的幸运便是，遇到叶兄这位对游戏有着如同对家对国般强烈责任感，犹如“游戏科学工作者”般的专业译者！

现在（2013年年末）无疑是游戏史上对独立游戏制作者最友好的年代。开发设备方便获得（相对过往仅由主机厂商授权才能获得专利开发设备，现在有一台智能手机和一台个人电脑就可以开发）、技术工具友好、调试过程简单方便，且互联网上有丰富的例程和开源代码参考，也有网上社区便于交流。很多爱好者能够很快地制作出可运行的游戏原型，其中一些也能发布到应用商店。

但是不全面掌握各方面知识，尤其是游戏引擎架构知识，往往只能停留在勉强修改、凑合重用别人提供的资源的应用程度上，难以做极限的性能改进，更妄谈革命式的架构创新。这样的程度是很难在成千上万的游戏脱颖而出的。我们所认可的真正的游戏大作，必定是在某方面大幅超越用户期待的产品。为了打造这样的产品，游戏内容创作者（策划、美术

等)需要“戴着镣铐跳舞”(在当前的机能下争取更多的创作自由度),而引擎架构合理的游戏可以经得起——也值得进行——反复优化,最终可以提供更多的自由度,这是大作出现的技术前提。

书的作者、译者、出版社的编者,加上读者,大家是因书而结缘的有缘人。因叶兄这本《游戏引擎架构》译著而在线上线下相识的读者们,你们是不是因“了解游戏引擎架构,从而制作/优化好游戏”这样的理想而结了缘呢?

亲爱的读者,愿你的游戏有一天因谜题巧妙绝伦、趣味超凡、虚拟世界气势磅礴、视觉效果逼真精美等专业因素取得业界褒奖,并得到玩家真诚的赞美。希望届时曾读叶兄这本《游戏引擎架构》译作的你,也可以回馈社会,回馈游戏开发的学习社区,帮助新人。希望你也可以建立微信公众号、博客等,或翻译游戏开发书籍,造福外语不好的读者,所以如果你的外语(英语、日语、韩语之于游戏行业比较重要)水平仍需精进,现在也可以同步加油了!

沙鹰 yingsha@qq.com

译序

数千年以来，艺术家们通过文学、绘画、雕塑、建筑、音乐、舞蹈、戏剧等传统艺术形式充实人类的精神层面。自20世纪中叶，计算机的普及派生出另一种艺术形式——电子游戏。游戏结合了上述传统艺术以及近代科技派生的其他艺术（如摄影、电影、动画），并且完全脱离了艺术欣赏这种单向传递的方式——游戏必然是互动的，“玩家”并不是“读者”、“观众”或“听众”，而是进入游戏世界、感知并对世界做出反应的参与者。

基于游戏的互动本质，游戏的制作通常比其他大众艺术复杂。商业游戏的制作通常需要各种人才的参与，而他们则需要依赖各种工具及科技。游戏引擎便是专门为游戏而设计的工具及科技集成。之所以称为引擎，如同交通工具中的引擎，提供了最核心的技术部分。因为复杂，研发成本高，人们不希望制作每款游戏（或车款）时都重新设计引擎，重用性是游戏引擎的一个重要设计目标。

然而，各游戏本身的性质以及平台的差异，使研发完全通用的游戏引擎变得极困难，甚至不可能。市面上出售的游戏引擎，有一些虽然已经达到很高的技术水平，但在商业应用中，很多时候还是需要因应个别游戏项目对引擎改造、整合、扩展及优化。因此，即使能使用市面上最好的商用引擎或自研引擎，我们仍需要理解当中的架构、各种机制和技术，并且分析及解决在制作中遇到的问题。这些也是译者曾任于上海两家工作室时的主要工作范畴。

选择翻译此著作，主要原因是在阅读中得到共鸣，并且能知悉一些知名游戏作品实际上所采用的方案。有感坊间大部分游戏开发书籍并不是由业内人士执笔，内容只足够应付一些最简单的游戏开发，欠缺宏观比较各种方案，技术与当今实际情况也有很大差距。而一些Gems类丛书虽然偶有好文章，但受形式所限欠缺系统性、全面性。难得本书原作者身为世界一流游戏工作室的资深游戏开发者¹，在繁重的游戏开发工作外，还在大学教授游戏开发课程以至编写本著作。此外，在与内地同事的交流中，了解到许多从业者不愿意阅读外文书籍。为了普及知识及反馈业界社会，希望能尽绵力。

¹原作者是顽皮狗（Naughty Dog）《神秘海域（Uncharted）》系列的通才程序员、《最后生还者（The Last of Us）》的首席程序员，之前还曾在EA和Midway工作。

或许有些人以为本著作是针对单机 / 游戏机游戏的，并不适合国内以网游为主的环境。但译者认为这是一种误解，许多游戏本身所涉及的技术是具通用性的。例如游戏性相关的游戏性系统、场景管理、人工智能、物理模拟等部分，许多时候也会同时用于网游的前台和后台。现时，一些动作为主、非MMO的国内端游甚至会直接在后台运行传统意义上的游戏引擎。至于前台相关的技术，单机和端游的区别更少。此外，随着近年移动终端的兴起，其硬件性能已超越传统掌上游戏机，开发手游所需的技术与传统掌上游戏机并无太大差异。还可预料，现时单机 / 游戏机的一些较高级的架构及技术，将在不远的未来着陆移动终端平台。

译者认为，本书涵括游戏开发技术的方方面面，同时适合入门及经验丰富的游戏程序员。书名中的架构二字，并不单是给出一个系统结构图，而是描述每个子系统的需求、相关技术及与其他子系统的关系。对译者本人而言，本书的第11章（动画系统）及第14章（运行时游戏性基础系统）是本书特别精彩之处，含有许多少见于其他书籍的内容。而第10章（渲染引擎）由于是游戏引擎中的一个极大的部分，有限的篇幅可能未能覆盖广度及深度，推荐读者参考[1]²，人工智能方面也需参考其他专著。

本译作采用L^AT_EX排版³，以Inkscape⁴编译矢量图片。为了令阅读更流畅，内文中的网址都统一改以脚注标示。另外，由于现时游戏开发相关的文献以英文为主，而且游戏开发涉及的知识面很广，本译作尽量以括号形式保留英文术语。为了方便读者查找内容，在附录中增设中英文双向索引（索引条目与原著的不同）。

本人在香港成长学习及工作，至2008年才赴内地游戏工作室工作，不黯内地的中文写作及用字习惯，翻译中曾遇到不少困难。有幸得到出版社人员以及良师益友的帮助，才能完成本译作。特别感谢周筠老师支持本作的提案，并耐心地给予协助及鼓励。编辑张春雨老师和卢鹤翔老师，以及好友余晟给予了大量翻译上的知识及指导。也感谢游戏业界专家云风、大宝和Dave给予了许多宝贵意见。此书的翻译及排版工作比预期更花时间，感谢妻子及儿女们的体谅。此次翻译工作历时三年半，因工作及家庭事宜导致严重延误，唯有在翻译及排版工作上更尽心尽力，希望求得等待此译作的读者们谅解。无论是批评或建议，诚希阁下通过电邮⁵、新浪微博⁶、豆瓣⁷等渠道不吝赐教。

叶劲峰 (Milo Yip)

2013年10月

²中括号表示引用附录中的参考文献。一些参考条目加入了其中译本的信息。

³具体是使用CTEX套装 (<http://www.ctex.org/>)，它是在MiKTeX (<http://www.miktex.org/>) 的基础上增加中文的支持。

⁴<http://inkscape.org/>

⁵miloyip@gmail.com

⁶<http://weibo.com/miloyip/>

⁷<http://www.douban.com/people/miloyip/>

奉献给

Trina、Evan及Quinn Gregory

纪念我们的英雄

Joyce Osterhus及Kenneth Gregory

序言

最早的电子游戏完全由硬件构成，但微处理器（microprocessor）的高速发展完全改变了游戏的面貌。现在的游戏是在多用途的PC和专门的电子游戏主机（video game console）上玩的，凭借软件带来绝妙的游戏体验。从最初的游戏诞生至今已有半个世纪，但很多人仍然认为游戏是一个未成熟的产业。即使游戏可能是个年轻的产业，若仔细观察，也会发现它正在高速发展。现时游戏已成为一个上百亿美元的行业，覆盖不同年龄、性别的广泛受众。

千变万化的游戏，可以分为从纸牌游戏到大型多人在线游戏（massively multiplayer online game, MMOG）等多个种类（category）和“类型（genre）”⁸，也可以运行在任何装有微芯片（microchip）的设备上。你现在可以在PC、手机及多种特别为游戏而设计的手持/电视游戏主机上玩游戏。家用电视游戏通常代表最尖端的游戏科技，又由于它们是周期性推出新版本，因此有游戏机“世代”（generation）的说法。最新一代⁹的游戏机包括微软的Xbox 360和索尼的PlayStation 3，但一定不可忽视长盛不衰的PC，以及最近非常流行的任天堂Wii。

最近，剧增的下载式休闲游戏，使这个多样化的商业游戏世界变得更复杂。虽然如此，大型游戏仍然是一门大生意。今天的游戏平台非常复杂，有难以置信的运算能力，这使软件的复杂度得以进一步提升。所有这些先进的软件都需要由人创造出来，这导致团队人数增加，开发成本上涨。随着产业变得成熟，开发团队要寻求更好、更高效的方式去制作产品，可复用软件（reusable software）和中间件（middleware）便应运而生，以补偿软件复杂度的提升。

由于有这么多种风格迥异的游戏及多种游戏平台，因此不可能存在单一理想的软件方案。然而，业界已经发展出一些模式，也有大量的潜在方案可供选择。现今的问题是如何找到一个合适的方案去迎合某个项目的需要。再进一步，开发团队必须考虑项目的方方面面，以及

⁸译注：Genre一词在文学中为体裁。电影和游戏里通常译作类型。不同的游戏类型可见1.2节。

⁹译注：按一般说法，2005年至今属于第7个游戏机世代。这3款游戏机的发行年份为Xbox 360（2005）、PlayStation 3（2006）、Wii（2006）。有关游戏机世代可参考http://en.wikipedia.org/wiki/List_of_video_game_consoles。

如何把各方面集成。对于一个崭新的游戏设计，鲜有可能找到一个完美搭配游戏设计各方面的软件包。

现时业界内的老手，入行时都是“开荒牛”。我们这代人很少是计算机专业出身（Matt的专业是航空工程、Jason的专业是系统设计工程），但现时很多学院已设有游戏开发的课程和学位。时至今日，为了获取有用的游戏开发信息，学生和开发者必须找到好的途径。对于高端的图形技术，从研究到实践都有大量高质量的信息。可是，这些信息经常不能直接应用到游戏的生产环境，或者没有一个生产级质量的实现。对于图形以外的游戏开发技术，市面上有一些所谓的入门书籍，没提及参考文献就描述很多内容细节，像自己发明的一样。这种做法根本没有用处，甚至经常带有不准确的内容。另一方面，市场上有一些高端的专门领域书籍，例如物理、碰撞、人工智能等。可是，这类书或者啰嗦到你难以忍受，或者高深到让部分读者无法理解，又或者内容过于零散而难于融会贯通。有一些甚至会直接和某项技术挂钩，软硬件一旦改动，其内容就会迅速过时。

此外，互联网也是收集相关知识的绝佳工具。可是，除非你确实知道要找些什么，否则断链、不准确资料、质量差的内容也会成为学习障碍。

好在，我们有Jason Gregory，他是一位拥有在顽皮狗（Naughty Dog）工作经验的业界老手，而顽皮狗是全球高度瞩目的游戏工作室之一。Jason在南加州大学教授游戏编程课程时，找不到概括游戏架构的教科书。值得庆幸的是，他承担了这个任务，填补了这个空白。

Jason把应用到实际发行游戏的生产级别知识，以及整个游戏开发的大局编集于本书。他凭经验，不仅融汇了游戏开发的概念和技巧，还用实际的代码示例及实现例子去说明怎样贯通知识来制作游戏。本书的引用及参考文献可以让读者更深入探索游戏开发过程的各方面。虽然例子经常是基于某些技术的，但是概念和技巧是用来实际创作游戏的，它们可以超越个别引擎或API的束缚。

本书是一本我们入行做游戏时想要的书。我们认为本书能让入门者增长知识，也能为有经验者开拓更大的视野。

Jeff Lander¹⁰

Matthew Whiting¹¹

¹⁰译注：Jeff Lander现时为Darwin 3D公司的首席技术总监、Game Tech公司创始人，曾为艺电首席程序员、Luxoflux公司游戏性及动画技术程序员。

¹¹译注：Matthew Whiting现时为Wholesale Algorithms公司程序员，曾为Luxoflux公司首席软件工程师、Insomniac Games公司程序员。

前言

欢迎来到《游戏引擎架构》世界。本书旨在全面探讨典型商业游戏引擎的主要组件。游戏编程是一个庞大的主题，有许多内容需要讨论。不过相信你会发现，我们讨论的深度将足以使你充分理解本书所涵盖的工程理论及常用实践的方方面面。话虽如此，令人着迷的漫长游戏编程之旅其实才刚刚启程。与此相关的每项技术都包含丰富内容，本书将为你打下基础，并引领你进入更广阔的学习空间。

本书焦点在于游戏引擎的技术及架构。我们会探讨商业游戏引擎中，各个子系统的相关理论，以及实现这些理论所需要的典型数据结构、算法和软件接口。游戏引擎与游戏的界限颇为模糊。我们将把注意力集中在引擎本身，包括多个低阶基础系统（low-level foundation system）、渲染引擎（rendering engine）、碰撞系统（collision system）、物理模拟（physics simulation）、人物动画（character animation），及一个我称为**游戏性基础层**（gameplay foundation layer）的深入讨论。此层包括游戏对象模型（game object model）、世界编辑器（world editor）、事件系统（event system）及脚本系统（scripting system）。我们也将接触游戏性编程（gameplay programming）的多个方面，包括玩家机制（player mechanics）、摄像机（camera）及人工智能（artificial intelligence, AI）。然而，这类讨论会被限制在游戏性系统和引擎接口范围。

本书可以作为大学中等级游戏程序设计中两到三门课程的教材。当然，本书也适合软件工程师、业余爱好者、自学的游戏程序员，以及游戏行业从业人员。通过阅读本书，资历较浅的游戏程序员可以巩固他们所学的游戏数学、引擎架构及游戏科技方面的知识。专注某一领域的资深程序员也能从本书更为全面的介绍中获益。

为了更好地学习本书内容，你需要掌握基本的面向对象编程概念并至少拥有一些C++编程经验。尽管游戏行业已经开始尝试使用一些新的、令人兴奋的编程语言，然而工业级的3D游戏引擎仍然是用C或C++编写的，任何认真的游戏程序员都应该掌握C++。我们将在第3章重温一些面向对象编程的基本原则，毫无疑问，你还会从本书学到一些C++的小技巧，不过C++的基础最好还是通过阅读[39]、[31]及[32]来获得。如果你

对C++已经有点生疏，建议你在阅读本书的同时，最好能重温这几本或者类似书籍。如果你完全没有C++经验，在看本书之前，可以考虑先阅读[39]的前几章，或者尝试学习一些C++的在线教程。

学习编程技能最好的方法就是写代码。在阅读本书时，强烈建议你选择一些特别感兴趣的`主题`付诸实践。举例来说，如果你觉得人物动画很有趣，那么可以首先安装OGRE，并测试一下它的蒙皮动画示范。接着还可以尝试用OGRE实现本书谈及的一些动画混合技巧。下一步你可能会打算用游戏手柄控制人物在平面上行走。等你能玩转一些简单的东西了，就应该以此为基础，继续前进！之后可以转移到另一个游戏技术范畴，周而复始。这些项目是什么并不重要，重要的是你在**实践**游戏编程的艺术，而不是纸上谈兵。

游戏科技是一个活生生、会呼吸的家伙，永远不可能将之束缚于书本之上。因此，附加的资源、勘误、更新、示例代码、项目构思等已经发到本书的网站¹²。

¹²<http://gameenginebook.com>

致谢

书籍从来不会凭空出现，本书亦然。没有家人、朋友、行业同僚的帮助，本书就不可能出版。衷心感谢他们及所有曾协助我完成本书的人们。

当然，受写作项目影响最严重的莫过于我的家庭。所以我想首先向我的妻子Trina致以特别的感谢。在这个艰难时期，她成为家庭的力量支柱。当我闭关完成一章又一章时，她昼夜无间照顾我们的两个儿子Evan（5岁）和Quinn（3岁）。我要多谢她，因为她放弃了自己的计划去适应我的时间表，她完成自己及我份内的家务（通常多于我想承认的），她经常在最需要的时刻给我鼓励。我也希望向儿子们表示感谢，长子Evan耐心容忍我不陪他玩他最爱的电视游戏，幼子Quinn在我每天长时间工作回家后给我拥抱和无尽的笑容。

我还要向我的编辑Matt Whiting及Jeff Lander致以特别的感谢。他们适时的反馈深具洞察力和针对性，常常恰到好处。此外，他们丰富的行业经验坚定了我将本书写得尽可能精准及前沿的信心。能跟这么老练的专家合作，既是愉快的经历，也是我的荣幸。感谢Jeff替我和Alice Peters¹³穿针引线，使这一项目起初能得以开展。

许多顽皮狗同事也对本书做出了贡献。他们要么提供反馈意见，要么帮助我制定某些章节的结构和主题内容。多谢Marshall Robin和Carlos Gonzalez-Ochoa对渲染一章的指导及监督，同时感谢Pål-Kristian Engstad对该章的文字及内容提供的建议。还要多谢Christian Gyrling对本书几个章节的反馈，包括动画一章（动画是他的专长之一）。感谢顽皮狗工程团队同仁制作了这么优秀的游戏引擎系统，成为本书的亮点。特别多谢艺电的Keith Schaeffer提供12.1节中关于物理对游戏影响的原始材料。也要多谢艺电的Paul Keet和圣达戈Midway的Steve Ranck（他是“Hydro Thunder”项目的主工程师）多年来的教导。虽然他们没有直接参与本书，但他们以不同形式影响了几乎本书每一页的内容。

这本书的素材源自我在南加州大学的信息技术课程（Information Technology Programme）中所开发的课堂“ITP-485：游戏引擎编程（Programming Game Engines）”的笔记，这门课到现在已经讲授了差不多3年。感谢Anthony Borquez博士，正是当时身为资

¹³译注：Alice Peters是本书原出版社A K Peters创始人之一。

讯技术计划总监的他，聘请我开发ITP-485这门课程。我也要多谢现在的课程总监Ashish Soni继续支持及鼓励我去发展ITP-485。

还要感谢亲友们的不断鼓励，包括他们常常在我写作时照顾我的妻子和两个儿子。我要感谢我的小姨子Tracy Lee、小舅子Doug Provins、姻表兄Matt Glenn，以及我们的好朋友Kim Clark和Drew Clark、Sherilyn Kritzer和Jim Kritzer、Anne Scherer和Michael Scherer、Kim Warner和Mike Warner。在我少年时，父亲Kenneth Gregory写了一本关于股票投资的书，让我萌生写作的念头。对此及其他事情，我永远都要感激他。也要感谢母亲Erica Gregory，一部分是因为她坚持希望我着手这一项目，一部分是因为在我小时候，她花费了很多心血教导我写作的艺术，我的写作技巧（以至于工作态度……和我古怪的幽默感）完全得自于她！

最后，我谨多谢Alice Peters、Kevin Jackson-Mead及A K Peters所有员工为出版本书做出的巨大努力。我很高兴可以与Alice和Kevin工作，衷心感谢他们在这么紧迫的时间下拼命工作，并多谢他们给予我这新作者的无限耐心。

Jason Gregory

2009年4月

目录

推荐序1	iii
推荐序2	v
译序	vii
序言	xvii
前言	xix
致谢	xxi
第一部分 基础	1
第1章 导论	3
1.1 典型游戏团队的结构.....	4
1.2 游戏是什么.....	7
1.3 游戏引擎是什么.....	10
1.4 不同游戏类型中的引擎差异.....	11
1.5 游戏引擎概观.....	22
1.6 运行时引擎架构.....	27
1.7 工具及资产管道.....	46
第2章 专业工具	53
2.1 版本控制.....	53
2.2 微软Visual Studio.....	61