“十二五”普通高等教育本科国家级规划教材
国家精品课程教材

计算机游戏程序设计

（提高篇）（第3版）

◎耿卫东　陈　为　梁秀波　王　锐　主　编
◎张　帆　郑文庭　李启雷　张　顺　副主编

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

教学资源

“十二五”普通高等教育本科国家级规划教材
国家精品课程教材

计算机游戏程序设计

（提高篇）（第3版）

耿卫东　陈　为　梁秀波　王　锐　主　编
张　帆　郑文庭　李启雷　张　顺　副主编

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书为“十二五”普通高等教育本科国家级规划教材。

本书着重介绍计算机游戏程序设计所需的专业领域知识，包括三维图形学基础、高级图形学编程、计算机动画技术、人工智能技术、音频处理技术和网络技术、VR/AR 游戏开发等，基本涵盖了计算机游戏编程的各个主要方面。全书共 12 章，取材于国内外的最新资料，强调理论与实践相结合，通过游戏实例来启发性地说明游戏编程的各种原理和方法。

本书教学资源包括三部分内容：示例代码、集成示例和绘制引擎（读者可以通过扫描二维码进行下载）。

本书面向的读者对象是那些已掌握基本的程序设计技能，但立志于从事计算机游戏软件开发的程序员和游戏开发爱好者。本书既可作为计算机、数字媒体技术和游戏专业的本科生（研究生）的教材，也可用于游戏学院和各类游戏编程人员培训班的参考资料，对正在从事游戏开发和制作的相关人员也具有重要参考价值。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目(CIP)数据

计算机游戏程序设计 . 提高篇 / 耿卫东等主编. —3 版. —北京：电子工业出版社，2018.8
ISBN 978-7-121-31938-9
Ⅰ. ⑮ 计… Ⅱ. ① 耿… Ⅲ. ① 游戏程序－程序设计－高等学校－教材 Ⅳ. ① TP317.61
中国版本图书馆 CIP 数据核字（2017）第 139476 号

策划编辑：章海涛
责任编辑：章海涛　　特约编辑：何　雄
印　　刷：北京七彩京通数码快印有限公司
装　　订：北京七彩京通数码快印有限公司
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：787×1092　1/16　　印张：22.5　　字数：570 千字
版　　次：2005 年 1 月第 1 版
　　2018 年 8 月第 3 版
印　　次：2018 年 8 月第 1 次印刷
定　　价：56.00 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）88254888，88258888。

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

本书咨询联系方式：192910558（QQ 群）。

前　言

在电子工业出版社的大力支持下，本书分别于2005年和2009年推出第1版和第2版。其中，第2版成功入选**普通高等教育“十一五”国家级规划教材**。近年来，随着游戏开发技术的不断演进，特别是VR/AR游戏的迅猛发展，应广大读者的要求，对原书稿中的内容做了大幅更新和补充，形成了第3版书稿，并入选**“十二五”普通高等教育本科国家级规划教材**。

在修订本书时，原本希望增加“游戏中的软件工程”“游戏策划”“游戏脚本编程”“游戏开发综合实例”等内容，使得本书变成一本能够覆盖不同层次、不同领域人员的“大而全”的游戏开发专业教材。但在编写过程中，我们发现涉及的内容实在太多，后来听取了采用该教材的大多数任课老师和同学的反馈意见，考虑到学时等限制因素，最终决定面向不同层次和水准的开发人员将书稿分为基础篇、提高篇两册。**“基础篇”**面向入门级的游戏开发人员，以介绍二维游戏开发的专业知识为主，通过丰富的实例使得读者可快速上手。**“提高篇”**面向有一定经验的游戏开发人员，以介绍三维游戏开发的专业知识为主，深入探究游戏开发的基本原理和技术，并针对当前游戏产业热点，补充了虚拟现实与增强实现游戏的开发流程和技术介绍，让读者的游戏开发能力更上一层楼。

在本书修订过程中，我们把本书编写的指导原则定位为以下几方面：

（1）把游戏开发涉及的多领域共性知识点的讲授和实践技能培训相融合，强调知识学习和动手实践的有机结合。

（2）在内容组织上，系统介绍游戏开发的知识点，并提供完整的示例代码和实验手册。

（3）采用“目标学习、案例导航”的组织方式，让读者能够快速掌握知识和技能要点。

（4）整合和吸收多教材的框架结构，结合作者的最新研究和开发工作进展，使得修订内容在技术和知识方面与时俱进。

本书提供的游戏场景融合了本书涵盖的游戏开发过程中涉及的主要技术和流程，目的是希望通过对这一场景的实现和学习，读者可以更直观、更深刻地感受和掌握游戏开发的主要技术和过程。本书从内容到文字反复修订、修改数次，力求达到减少错误。其中，王锐等重新修订编写了第1～5章的内容。第6章由丁治宇修订撰写，内容涉及力学和物理，他参考、翻阅了大量的物理教材，设计和制作了所有的插图和游戏场景。朱标修订了第7章，梅鸿辉修订了第8章，新增了习题实例并给出了代码。第9～11章由梁秀波等修订。第12章及附录A中的虚拟现实游戏开发等内容，由郑文庭、张帆等参与修订。此外，丁治宇还负责书稿的

通稿和校对，设计了全部的游戏场景；黄家东参与了全书的最后校对，在此一并致谢。

再次感谢浙江大学计算机科学与技术学院、软件学院、CAD&CG 国家重点实验室为作者提供的优良科研条件和各种便利，使得本书的撰写得以顺利完成。感谢为本书提供图片和示例代码的众多相关公司和人员，限于篇幅，不一一列出。

感谢所有使用本书的任课老师和同学的反馈建议。感谢电子工业出版社的各位编辑，谢谢他们的鼓励与协作。最后，感谢家人对本书的撰写和再次出版的无私支持和付出。

由于作者知识和水平有限，即使本书是再版，其中也肯定会存在一些问题，恳请读者提出宝贵建议。

本书为任课教师提供配套的教学资源（包含电子教案和示例代码），需要者可**登录华信教育资源网**（http://www.hxedu.com.cn），注册之后进行免费**下载**，或者**扫描下面的二维码进行下载**。

作　者

此为试读，需要完整PDF请访问：www.ertongbook.com

目 录

此为试读，需要完整PDF请访问：www.ertongbook.com

第 1 章　三维游戏引擎技术简介

在游戏开发界有这样一种说法，如果把游戏比作一个人，那么游戏技术是核心的支撑骨架，游戏策划是决定游戏逻辑的内在血肉，美术则是包裹游戏内容的外表皮肤。一款游戏好玩与否，是技术、策划、美术等因素共同作用的结果。本书讲述的是这三者中支撑起整个游戏的骨架——技术。而本章要讨论的三维游戏引擎则是驱动游戏运行的技术核心。总之，游戏引擎把游戏中的所有程序与资源元素捆绑在一起，通过指挥它们同时、有序地工作，让玩家感受到游戏希望传达的剧情、关卡、美术、音乐等外在形式，体验到操作游戏内容带来的可玩性。因此，从某种意义上来说，三维游戏引擎的开发是整个游戏开发的技术核心和开发基础，它为开发者提供的是一系列可视化开发工具和可重用组件。

游戏引擎是随着游戏开发技术的进步而发展起来的。早在电子游戏行业诞生初期，并没有游戏引擎这个概念，所有图形操作以及游戏本身都是由开发者独立从头开始完成游戏开发的。但是随着游戏开发规模的扩大以及开发者人数的增加，如果每次开发一款游戏，都要重新单独编写游戏的不同模块，并完成与硬件相关底层代码的相关接口程序，这将十分地耗时耗力。那么，有没有办法能够将这个独立出来，缩短游戏开发周期，降低对游戏开发者的要求，减少成本呢？答案是肯定的。一些有经验的开发者逐渐摸索出了一条偷懒的方法，他们借用上一款类似题材的游戏中的部分代码作为新游戏的基本框架，以节省开发时间和开发费用。根据生产力学说，单位产品的成本因生产力水平的提高而降低，自动化程度较高的手工业者最终将把那些生产力低下的手工业者淘汰出局。引擎的概念就是在这种机器化作业的背景下诞生的。世界上第一个具有通用性的游戏引擎是由 id Software 工作室基于游戏 DOOM 的开发而提出的，它在架构上非常合理地定义并区分了底层软件组件系统、游戏世界构造、游戏规则制定等。这样做的好处在于，游戏开发者开发新产品的时候，不需要从零开始，可以充分利用现有的软件模块实现内容不同的游戏，因此大大降低了游戏开发的门槛，让玩家 DIY（Do it by yourself）游戏成为一种可能。正是由于 DOOM 引擎的高度可定制，MOD（Modifying existing games, using free toolkits provided by the original developers）一词也随之诞生，并出现了大量优秀的由玩家 DIY 或其他工作室开发的第三方游戏内容（MOD 作品），大大丰富了 DOOM 框架下游戏的可玩性，并为 DOOM 赢得了极佳的口碑。之后随着技术的进步，游戏引擎及其相关技术已经发展成为一个庞大的产业。

无论是 2D 游戏还是 3D 游戏，无论是角色扮演游戏、即时策略游戏、冒险解谜游戏还是动作射击游戏，都有一段内核功能代码。游戏引擎只是将这些内核功能代码抽象化、模块化，并不断进化，发展为一套由多个子系统共同构成的复杂系统，从建模、动画到光照、粒子特效，从物理系统、碰撞检测到文件管理、网络特性，还有专业的编辑工具和插件，几乎涵盖了开发过程中的所有重要环节，以下对三维游戏以及三维游戏引擎的一些关键部件进行简单介绍。

1.1　三维游戏的基础架构

三维游戏的基础架构可以分为 4 层，底层是硬件层，往上依次是操作系统、硬件驱动、一系

列高级图形 API 等，如 Direct3D 或 OpenGL，这些 API 封装了 GPU 以及一些硬件的部分功能，顶层是游戏开发者或是艺术家，游戏引擎的位置就是次顶层，不但要对游戏画面进行绘制，还需要对事件、I/O、AI、音效等进行处理，并为开发者提供可视化和舒适的开发编辑环境。图 1-1 给出了三维游戏一个典型基础架构。

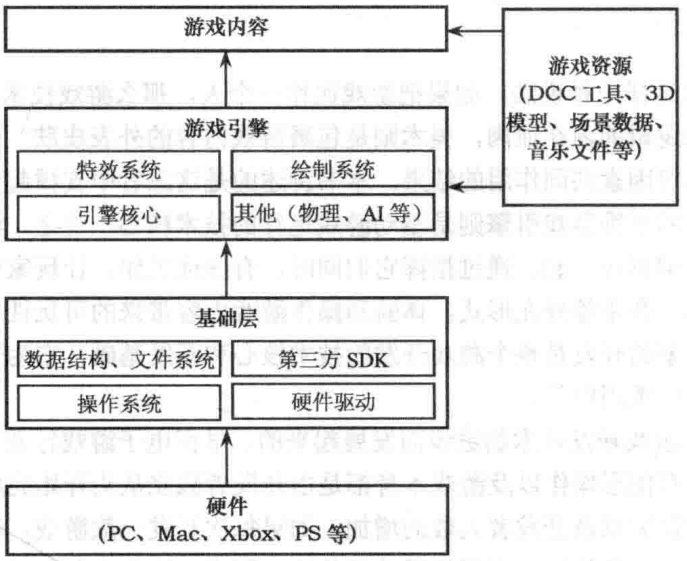

图 1-1　三维游戏的基础架构

1.1.1　硬件层

硬件层与其他诸多计算机软件架构类似，是整个游戏引擎体系架构的底层，决定着游戏引擎能够运行的游戏硬件组成。目前，由于游戏硬件架构的不统一，不同平台开发游戏的难度和质量以及目标游戏类型也会有较大区别，以下列举部分目前较为主流的游戏硬件平台并作简单介绍。

1．个人计算机游戏平台

从第一款电子游戏诞生到现在，利用计算机来进行游戏一直是电子游戏发展的重要推动力量。个人计算机（PC）是目前数量上最大的游戏硬件载体，也是当前最大的游戏平台。同时，个人计算机平台也是民用领域硬件更新最快的平台，提供的游戏硬件性能往往要高出家用游戏主机不少。但是由于 PC 硬件本身的体系架构问题以及基于 PC 平台的游戏优化问题，虽然 PC 的单一硬件性能往往远超游戏主机，但是实际游戏性能的提升并不明显。

2．家用主机游戏平台

家用游戏机又称为游戏主机，主要针对一部分在家里客厅使用电视进行游戏的人群。早期，由于 PC 硬件体系架构差异巨大以及普及程度的差异，家用主机平台采用了定制的硬件架构、CPU 芯片与 GPU 显卡。随着硬件技术的不断发展，无论主机硬件生产厂商还是游戏厂商都被不断整合。从 Xbox 面世以后，游戏主机硬件体系架构开始越来越接近于 PC 的体系架构。后续的 Xbox360、PS3 均采用了 PC 平台标准的图形处理芯片，Xbox360 的 GPU 是基于 ATI R500 架构的显卡，PS3 则采用了基于 NVIDIA G73 架构的 GPU。近年来，家用主机平台的最新代表 Xbox One、PS4 均采用了与 PC 平台几乎完全一样的体系结构，无论 CPU、GPU 还是内存，均采用了 PC 的标准。因此，在游戏开发成本大幅提升以及硬件架构越来越趋同的大背景下，主机平台和 PC 平台的游戏

开始渐渐渗透，跨平台游戏越来越多，很多原先只有主机独占的游戏开始登陆 PC，同时很多经典计算机游戏也相应地推出了运行在游戏主机上的版本。

3. 移动游戏平台

早期移动游戏平台是任天堂的天下，任天堂 1989 年推出的 GameBoy 游戏机一度成为掌上游戏机的代名词。后来，从 SONY 在 2004 年推出 PSP 开始，移动游戏机市场开始出现竞争。最近随着智能手机性能的不断强化，俨然形成了由任天堂、SONY、智能手机多种平台相互角逐的局面。不同于主机和 PC，智能手机不单单在硬件架构上和传统掌上游戏机有较大区别，操作方式上也有显著不同。但是随着智能手机游戏处理性能的迅速增长，掌上游戏机和智能手机在性能、架构及指令集上的差异也开始变得更小，越来越多的厂商开始发布跨掌上游戏机和智能手机平台的游戏。

需要指出的是，除了上面介绍的几种游戏硬件平台外，还有一大类定制的游戏硬件，如在专业游戏厅中的各种游戏机。由于这些游戏硬件具有独特的硬件组成，游戏开发缺乏通用性，往往需要针对定制的硬件规格进行开发，因此本书对这些游戏硬件平台不再赘述。

1.1.2 基础层（驱动、操作系统及 API）

基础层可以说是连接软件和硬件的桥梁，驱动程序则是电脑体系架构中直接与硬件接口的代码。通过驱动程序，计算机的其他应用程序（包括游戏程序）就可以与设备进行通信，操控硬件提供的各种功能和特性。操作系统作为计算机软、硬件管理者，需要均衡的分配各种资源（比如处理器时间片）给各个应用程序，其中包括游戏程序。目前，随着游戏产业的发展，一个成熟的游戏引擎需要支持的硬件平台与操作系统平台越来越多样化，从 PC 平台到家用主机平台再到移动计算平台，从 Windows 操作系统到 Linux、Mac OS 操作系统再到 IOS 或 Android 操作系统，甚至到完全运行于浏览器之上的网页平台。为了降低在这些多种多样硬件平台上应用开发的难度，人们开发出了一系列通用的软、硬件接口库，如针对音效开发的 OpenAL、DirectSound，针对图形绘制开发的 OpenGL、DirectX 等。

1.1.3 游戏引擎

游戏引擎并不是一个模块或两个模块那么简单，完整的游戏引擎包含各种模块，涉及绘制、物理、模型，特效、资源等，以及游戏内容创建器（Digital Content Creator），为游戏开发人员提供多种游戏资源和逻辑的编辑工具和环境，并将各类场景模型以及资源导入到游戏引擎在线运行内核中（Runtime Engine）。游戏引擎在线运行内核是整个游戏程序架构中最重要的部件所在，涉及绘制、物理、人工智能、音乐等一系列的组件，以及为游戏提供具体的管理服务等。

1. 游戏内容创建器

游戏在本质上是一个多媒体应用，不仅仅是算法程序，因此除了游戏引擎外，与游戏相关的各种数据资源也非常重要。对一个游戏来说，数据资源可以是 3D 模型，也可以是贴图纹理，甚至是音乐文件等游戏内容。随着游戏产业与技术的发展，游戏所需的数据资源往往变得极为庞大，远远超过了控制游戏运行的逻辑代码本身，因此游戏引擎通常将这部分负责创建并管理这些数据资源的任务作为一个工具独立出来，这就是游戏内容创建器。

游戏内容创建器工具可以分为两类。一类是已经非常成熟的第三方数据内容创建器工具，目前使用较为广泛的第三方数据内容创建器工具有：① Autodesk 的 Maya 和 3ds MAX，是目前十分流行的创建 3D 网格模型和动画数据的游戏内容创建器工具；② Adobe 的 Photoshop，主要用来生成和编辑图像、纹理数据；③ SoundForge 是一个流行的用来创建声音片段的工具。另一类是游戏引擎自己开发数据内容创建器来供使用该游戏引擎的开发人员使用。当前的商用游戏引擎 Unreal、CryEngine、Unity3D 都有自己的数据内容创建器。

2. 游戏引擎基础库

游戏引擎基础库一般包括如下内容。

① 容错管理：常常用来在游戏引擎建立阶段进行错误检测，帮助开发者找到错误并完善产品，但是在最终推出的产品中往往不会包含。

② 内存管理：一般来说，每个游戏引擎都需要建立自己独立的内存分配系统，来保证高效的内存调用和管理。

③ 数学库：必不可少，游戏往往高度依赖于数学计算，因此每个游戏引擎都包含一个甚至更多的数学库，这类数学库提供向量、矩阵等线性代数的运算以及计算交点、距离等几何运算。

④ 传统数据管理与算法：在设计一款游戏引擎的时候，除非开发者打算完全依赖于一些第三方的包（如 STL），否则需要自行开发一个合适的用来管理数据和算法的工具。这类算法和数据结构往往采用最优的性能、最小的内存占用，以满足游戏对性能和存储空间的苛刻要求。

3. 在线运行内核

三维游戏通常是一个非常庞大的系统，通常包含了若干子系统，如时间与消息处理、输入/输出子系统、三维场景管理、游戏状态维护与更新、人工智能、音效系统、绘制系统等。这些子系统基于三维数据，对数据进行调度与管理，进行逻辑计算，实现游戏内容的优美呈现。这些内容将在本书的后面章节中逐一介绍。

1.2 三维游戏引擎发展简史

经过多年发展，游戏技术涌现了很多游戏引擎，但真正能获得他人认可并成为标准的游戏引擎并不多。纵观几十年的发展历程，游戏引擎最大的驱动力来自于 3D 游戏，尤其是 3D 射击游戏。因此，下面对三维游戏引擎的历史回顾将主要围绕动作射击游戏的变迁展开。动作射击游戏同 3D 引擎之间的关系相当于孪生兄弟，一同诞生，一同成长，互相为对方提供着发展的动力。

在游戏引擎出现之前，开发游戏就像在开发单一的软件，针对特定游戏平台从下到上设计，以充分利用硬件性能。即使在类似的平台上，代码也很少能在游戏之间重复使用。虽然在 20 世纪 80 年代出现了几种用于独立视频游戏开发的 2D 游戏创建系统，但是这些系统仅仅具有了当前游戏引擎的部分要素。

这种情况直到 1993 年，id Software 的游戏公司发布了一款新游戏——DOOM《毁灭战士》。这款游戏在游戏历史上非常重要，首先它定义了现代第一人称视角射击游戏的游戏类型；其次，这款游戏第一次定义并实现了一个游戏引擎 DOOM Engine（后来被称为 id Tech 1），并将游戏引擎引入到了游戏开发中，开启了游戏开发的一个新时代。

id Software 游戏公司的首席程序员·卡马克（John Carmack）在开发《毁灭战士》时就不仅仅

创造一个新游戏，而是构思并执行了一种组织计算机游戏各种组件的新方法——将游戏引擎执行核心功能与游戏内容的资源分开。在游戏发布的前一年（1992 年），id Software 专门召开了一个新闻发布会，介绍即将到来的《毁灭战士》包含的创新游戏技术和设计：一个卓越的图形引擎，使用当时最先进的 256 色 VGA 图形，支持多人游戏的点对点网络，以及 P2P 的游戏模式等。这些技术为当时以及未来的游戏引擎给出了示范性的架构设计与接口规范，也给出了在 PC 平台上或者以后跨平台上进行游戏开发的新范例。

用革命来描述这样的开发改变丝毫不为过。Lev Manovich [15]将《毁灭战士》的影响描述为，为游戏开发创造了“新文化经济”。他把游戏分解为引擎和资源的软件模型用经济学的挂点来描述：“生产者定义了一个对象的基本结构并发布了一些例子，同事允许消费者建立自己版本的工具，与其他消费者共享。”卡马克以放弃对创造性进行控制的方式，解放了游戏美术、设计对游戏的想象力。id Software 鼓励玩家社区并与第三方开发人员合作，允许他们修改游戏，或者在 id Software 的引擎上创建新的游戏。通过游戏引擎创建强大的内容创建模型，允许游戏开发者或者玩家做他想做的事情：改变游戏并与其他玩家分享变化。这使得程序开发人员可以将注意力更多地放到改进技术上，而不是游戏设计。

这一共享的思想无疑取得了很大成功。《毁灭战士》系列本身相当成功，大约卖了 350 万套，而授权费又为 id Software 公司带来了一笔可观的收入。有趣的是，这种共享的思想或许与卡马克青少年时黑客的背景有关。在此之前，引擎只是作为一种自产自销的开发工具，从来没有哪家游戏商考虑过依靠引擎赚钱，《毁灭战士》引擎的成功无疑为人们打开了一片新的市场。

1994 年，id Software 发布了 Quake《雷神之锤》，同时将《毁灭战士》使用的 Doom Engine 引擎升级为 Quake Engine（id Tech 2）。这是当时第一款完全支持多边形模型、动画和粒子特效的真正意义上的 3D 引擎，而不是 2.5D 引擎。此外，id Tech 1 引擎支持的网络功在 id Tech 2 上得到进一步发扬，这使得在《雷神之锤》上进行多人对战成为一种风潮，也直接促进了电子竞技产业的发展。

一年后，id Software 公司进一步推出 Quake II《雷神之锤 2》，在 Quake 引擎基础上改进，升级为 Quake II 引擎。Quake II 引擎能够支持 OpenGL 实现更好、更快的 3D 渲染，加上 Lightmap 技术，使得 Quake II 引擎在图形效果上实现了质的飞越。后期有大量游戏使用 Quake II 的引擎，如 Raven Software 公司的 Heretic II（1998 年）和 Soldier of Fortune（2000 年）、Ritual Entertainment 公司的 SiN（1998 年）、Ion Storm 公司的 Daikatana（2000 年）和 Anachronox（2001 年）等。2001 年，id Software 公开了 Quake II 的源码，所以一直有利用 Quake II 引擎基于 GPL 授权开发的游戏，如 Thirty Flights of Loving（2012 年）和 Alien Arena: Warriors of Mars（2017 年）。

1999 年，id Software 发布了 Quake 系列的第三款游戏 Quake III Arena《雷神之锤 III 竞技场》。这款游戏使用的引擎是在 Quake II 引擎基础上进行大量改进的 Quake III Arena 引擎，后又被称为 id Tech 3 引擎。id Tech 3 引擎舍弃了使用 CPU 渲染的方法，而强制使用 OpenGL 支持的 GPU 来进行渲染。此外，id Tech 3 引入了样条曲面和体素来表示物体，改进了阴影算法，使用了新的骨骼系统，并且能够支持大范围的室外场景等。id Tech 3 引擎获得了广泛的使用，除 id Software 自己开发 Quake III 的资料篇外，还被 Raven 软件公司用来开发《星际迷航》《星球大战》系列的后续作品。在 id Tech 3 引擎之上演化出来的 IW 引擎（IW engine）则成为 Infinity Ward 工作室《使命召唤》系列的支持引擎，而 IW engine 已经发展到了第 7 代。

除 IW 引擎，另一款改进 Quake 引擎来开发游戏的是 Valve 公司。Valve 公司广受好评的游戏 Half Life《半条命》采用的游戏引擎由授权的 Quake 引擎大量修改而来，以致 Valve 公司创始人

此为试读，需要完整PDF请访问：www.ertongbook.com