游戏引擎原理与应用

曹华 编著

WUHAN UNIVERSITY PRESS
武汉大学出版社

游戏引擎原理与应用

曹华 编著

WUHAN UNIVERSITY PRESS
武汉大学出版社

图书在版编目(CIP)数据

游戏引擎原理与应用/曹华编著. —武汉:武汉大学出版社,2016.8
ISBN 978-7-307-18570-8

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

中国版本图书馆 CIP 数据核字(2016)第 203415 号

责任编辑:谢文涛　　责任校对:李孟潇　　版式设计:马　佳

出版发行:武汉大学出版社　(430072　武昌　珞珈山)
(电子邮件:whu_publish@163.com　网址:www.stmpress.cn)
印刷:荆州市鸿盛印务有限公司
开本:787×1092　1/16　印张:22.75　字数:538 千字　插页:1
版次:2016 年 8 月第 1 版　2016 年 8 月第 1 次印刷
ISBN 978-7-307-18570-8　定价:45.00 元

前　言

游戏引擎是当今游戏产业发展的核心技术驱动力。目前各类火爆的游戏如手机游戏或网络游戏大多基于各类游戏引擎来进行开发，极大地提高游戏的开发效率和运行质量。基于游戏引擎框架平台的游戏开发、基于工程化方法构建和维持高质量游戏软件等游戏开发中普遍采用的技术理念越来越符合软件工程的概念范畴。

经过近几年的快速发展，游戏引擎进入一个蓬勃发展、百家争鸣的时代，出现了很多高水准的商业化游戏引擎，包括 Unreal、Unity3D 等。但遗憾的是，由于游戏引擎的高度复杂和快速发展，以及早期国外引进游戏引擎的高价格和技术门槛，国内高校的游戏引擎技术原理的教学资料相对还比较缺乏，这不利于学生应用游戏引擎开发高质量的游戏，进而开发高质量的游戏引擎本身。这使得我国游戏产业的自主创新、持续发展受到很大的挑战和制约。

作者近几年一直在华中科技大学软件学院从事数字媒体技术专业游戏程序设计的教学工作，一方面不断分析国外游戏引擎相关理论技术的发展，另一方面对基于游戏引擎的开发实践追踪总结，因此萌生了撰写游戏引擎原理与应用相关书籍的想法。

基于游戏引擎的游戏软件开发是一个复杂的系统工程，需要遵循软件工程的规范，本书在一定程度上从软件工程的视角论述三维游戏引擎原理及相关应用技术，期望为从事游戏软件开发的本科生、研究生和工程技术人员提供关于游戏引擎原理的全面知识和应用方法，为我国的游戏产业向更高层次发展形成强劲的后续动力。全书共 12 章，内容包括三个部分：基础内容，主要技术原理和应用实例。基础内容侧重游戏引擎的基础知识，概述了目前主要的游戏种类及其对游戏引擎的技术需求，游戏开发中的数学基础，从软件工程的角度呈现游戏引擎的架构以及游戏开发的软件工程基础，包括面向对象概念在游戏开发中的体现，游戏开发中用到的主要软件设计模式。主要技术原理围绕游戏引擎架构体系中各主要方面展开，系统地介绍了游戏循环、渲染引擎、场景管理、动囤系统、物理引擎、GUI 及输入输出模块、游戏人工智能和网络引擎的实现机制。为了使读者对三维游戏引擎原理的应用设计有更切身的体会，以目前国内流行的 Unity3D 商业引擎为例，结合原理进行应用剖析，并通过完整的应用案例，阐述应用游戏引擎实现游戏功能。

本书的撰写得到华中科技大学教材建设项目的资助。姜涛、刘思尧、李珊珊参与了编写工作。由于作者的知识面和实践知识有限，书中肯定存在很多片面和错误之处，恳请读者批评指正。

作　者

2016 年 7 月

目　录

第1章 引 言

1.1 游戏和游戏发展史

电子游戏产业是一个综合性强的多学科交叉领域，其涉及的相关技术包括：数字图像处理技术、数字视频和音频处理技术、计算机动画技术和虚拟现实技术等。电子游戏产业是科学技术和人文艺术的结合，它是科技高度发展和人类社会进步的产物。当代的游戏融合了人工智能、计算机图形图像技术、音乐和网络通信等技术，现代大型游戏的设计也逐步开始被大众所认可而成为了一门综合的设计艺术。

1. 电子游戏的萌芽期

1971 年，美国麻省理工学院学生设计了一款名为“Computer Space”(电脑空间)的游戏机，游戏主题是两个玩家各自控制一艘太空战舰向对方发射导弹进行相互攻击。1972 年，Atari 公司推出的以乒乓球为题材的游戏机“Pong”，标志着电子游戏产业化的开始。1970 年苹果公司的 Apple 系列机推出，机器中附带了许多现在看来十分初级的动作、射击类的游戏，引起了人们极大的兴趣，从此电脑游戏迅速发展起来。

2. 电子游戏业的发展期

20 世纪 80 年代，电子游戏开始广泛进入家庭娱乐领域。美国和日本的游戏设计师们设计了各种类型的被称为 TV Game 的电视游戏，以及掌上游戏机。

在电子游戏迅速发展的同时，个人电脑的普及也促进了电脑游戏业的蓬勃发展。1989 年，Broderbund 公司在 Apple Ⅱ上开发了风靡一时的电脑游戏《波斯王子》。这款动作与冒险相结合的电脑游戏，在游戏业获得了巨大的成功。而电脑游戏真正的发展和强大是在接下来的 90 年代，电脑游戏开始逐渐为人们所熟悉和接受。

3. 电子游戏业的成熟期

20 世纪 90 年代，电子游戏进入了个人电脑领域并形成了丰富的游戏形式。Intel 处理器芯片运行速度的不断提高和高性能图形显示卡的出现，使游戏的画面质量大幅提高。1992 年 3D Realms 公司与 Apogee 公司公布了 3D 游戏“德军总部”(Wolfenstein 3D)，开创了 3D 电子游戏的新时代，并率先在游戏开发过程中使用了游戏引擎。3D 游戏为游戏玩家提供了一个更加逼真、自由的三维立体空间，使玩家在体验游戏的过程中有“身临其境”的感受。

20 世纪 90 年代后期，随着网络技术的进步，游戏业的发展也开始从单人游戏的娱乐模式向多人在线娱乐模式的方向转变。网络游戏将人机对战转为人与人之间的较量，成千上万的玩家通过互联网一起游戏，极大增强了游戏的挑战性和趣味性。

自20世纪70年代发展至今，电子游戏业经历了萌芽期、发展期和成熟期三个阶段，从电视到个人电脑再到网络这三种媒介的转变可以看出技术起到了决定性和推动性的作用。

1.2 常见的游戏类型

以下介绍一些常见的游戏类型，并探讨每个类型的技术需求。

1.2.1 第一人称射击游戏(FPS)

第一人称射击游戏FPS(First Person Shooting)的典型例子是《雷神之锤》(Quake)《虚幻竞技场》(Unreal Tournament)《半条命》(Half-Life)《反恐精英》(Counter-Strike)和《使命的召唤》(Call of Duty，图1-1)。这些游戏中的角色移动相对缓慢，但可以漫游在主要为走廊的巨大场景中，也可以发生在各种各样的虚拟环境中，包括开阔的室外区域以及狭窄的室内，FPS角色移动机制包括行走、散放的操控器具，如地面车辆、气垫船、船只和飞机。

图1-1 使命召唤(Xbox 360/PS3)

第一人称射击游戏最具有技术挑战性，包含如下关键技术：

(1)大型三维虚拟世界的高效渲染；玩家的虚拟手臂和武器的逼真动画；非玩家角色(如玩家的敌人及同盟)有逼真的动画和智能；

(2)快速响应的摄像机控制/瞄准机制；

(3)玩家角色运动和碰撞模型，并让这些游戏有“浮动”的感觉；

(4)小规模在线多人游戏的能力。

1.2.2 平台及其他第三人称游戏

“平台游戏Platformer”指基于任务角色的第三人称动作游戏(Third Person Game)，这类游戏主要的机制是在平台之间跳跃。典型的有二维游戏时代的《太空惊魂》(Space

Panic)，三维游戏时代的《超级玛丽奥 64》(Super Mario 64)《杰克与达斯特》(Jack and Daxter)系列(图 1-2)和《战争机器》(Gears of War，图 1-3)等。

图 1-2 《杰克与达斯特》(Jack and Daxter)

图 1-3 《战争机器》(Gears of War)

第三人称游戏和第一人称射击游戏有很多共同之处，但第三人称游戏更强调主角的能力(Ability)和运动模式(Locomotion Mode)。第三人称游戏专注于以下技术：

(1)移动平台、梯子、绳索、棚架和其他各种各样的运动模式。

(2)摄像机视线追踪技术，使玩家用手柄或鼠标旋转摄像机，跟随相机注视玩家的角色。

(3)复杂的摄像机防碰撞系统，以保证视点不会穿过背景几何物体或动态的前景物体。

1.2.3 格斗游戏

格斗游戏通常是控制两个玩家在一个擂台上互相对打。典型的例子有《灵魂能力》(Soul Calibur)和《铁拳》(Tekken，图 1-4)。

图 1-4 《铁拳 3》(Tekken 3)PlayStation

由于这些游戏的三维世界比较小，而且摄像机一直位于动作的中心，因此只有很少甚至不需要世界细分(World Subdivision)或遮挡剔除。最新的格斗游戏，比如EA的《拳击之夜3》(Fight Night Round 3，图1-5)，具有以下技术特点：

(1)高清角色图形，包括仿真的皮肤着色器(Shader)。着色器模拟表面下散射和冒汗效果。

(2)逼真的角色动画。

(3)基于物理的布料和头发模拟。

图1-5 《拳击之夜3》(Fight Night Round 3)

1.2.4 竞速游戏

竞速游戏(Racing Game)包括了所有主要任务是在赛道上驾驶汽车或其他车辆的游戏，移动速度通常比FPS快得多。通常关注车辆的图形细节、赛道和周围环境。图1-6是知名竞速游戏系列《跑车浪漫旅5》(Gran Turismo 5)的屏幕截图。典型的竞速游戏包括以下技术特性：

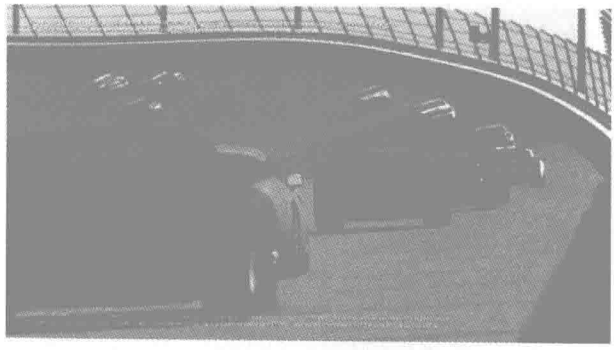

图1-6 《跑车浪漫旅5》(Gran Turismo 5)

(1)使用各种“窍门”去渲染遥远的背景，如使用二维纸板形式的树木、丘陵和山脉。

(2)赛道通常分解为相对简单的二维区域，称为“分区(Sectors)”。这些数据结构用来优化渲染、可见性判断(Visibility Determination)，帮助非玩家操作车辆的人工智能及路径搜寻，以及解决许多其他技术问题。

(3)第三人称视角摄像机通常追随在车辆背后，第一人称摄像机有时候会置于驾驶舱里。

(4)当赛道经过隧道和其他较狭窄的空间，必须努力确保摄像机不与背景几何体相撞。

1.2.5 实时策略游戏

现代即时战略 RTS(Real-Time Strategy)包括《沙丘魔堡 2》(Dune Ⅱ: The Building of Dynasty)《魔兽争霸》(Warcraft)《命令与征服》(Command & Conquer)《帝国时代》(Age of Empires)和《星际争霸》(Starcraft)等。游戏世界通常会是一个自上而下的斜视角。图 1-7 是《帝国时代》的游戏截图。

图 1-7 《帝国时代》(Age of Empires)

RTS 游戏中包括以下常用技术：

(1)每个作战单元使用相对低解析度的模型，这样游戏就可以同时支持显示大量的单位。

(2)游戏的设计和进行是在高度场地形(Height Field Terrain)画面上展开的。

(3)基于栅格(Grid-based 或基于单元/Cell-based)构建游戏世界，并使用正射投影(Orthographic Projection)简化渲染系统。

(4)使用透视投影和真三维世界，确保作战单位和背景元素(如建筑物)能适当地对齐，如图 1-8 所示的《命令与征服 3》。

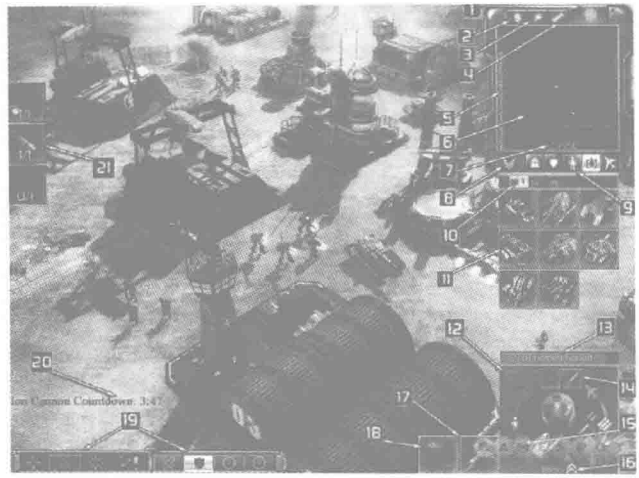

图 1-8 《命令与征服 3》(Command & Conquer 3)

1.2.6 大型多人在线游戏

大型多人在线游戏 MMOG(Massively Multiplayer Online Game)的典型例子是游戏《魔兽世界》《无冬之夜》《无尽的任务》《星球大战》等，支持大量玩家同时在线(从数千到数十万)，通常都在一个非常大的持久世界(Persistent World)里进行游戏(持久世界即世界的内部状态持续很长一段时间，远远超出特定玩家每次玩的时间)。图 1-9 显示的是《魔兽世界》的一个屏幕截图。

图 1-9 《魔兽世界》(World of Warcraft)

MMOG 的核心是一组非常强大的服务器，这些服务器维护游戏世界的权威状态，管理用户登录/退出，也提供网络用户聊天或 IP 电话服务等。由于游戏场景规模和玩家数量都很大，因此图形逼真度通常低于其他类型的游戏。

1.2.7 其他游戏类型

还有很多其他类型的游戏，不在此详述，例如：

(1)体育游戏(sport)，包含每个主要运动如 football，baseball，soccer，golf，etc.；

(2)角色扮演游戏(Role Playing Game，RPG)；

(3)上帝模拟游戏(God Game)，如《上帝也疯狂》《黑与白》；

(4)环境与社会模拟游戏，如《模拟城市》或《模拟人生 Sims》；

(5)解密游戏 puzzle，如《俄罗斯方块》；

(6)非电子游戏的移植，如象棋、围棋、卡牌游戏等；

(7)基于网页的游戏；

(8)其他游戏类型。

每个游戏都有自己的特定的技术要求，游戏引擎因游戏类型不同而存在不同的风格流派。随着硬件性能越来越强大，流派之间的差异因考虑优化涉及的差异会越来越小，因此越来越可能在不同的游戏类型甚至在不同的硬件平台上重用相同的引擎技术。

第 2 章　游戏引擎概观

目前常见的游戏主要由 2D/3D 虚拟世界组成，并有少量的玩家(1~16 个)，也包括应用到互联网上的 flash 游戏、纯解谜游戏或 MMOG 大型多人在线(Massively Multiplayer Online Games)。本章主要讨论游戏引擎开发游戏的原理和游戏引擎的架构，这些引擎可以用来开发第一人称射击、第三人称动作/平台游戏、赛车游戏、格斗游戏等。

2.1　游戏开发和游戏引擎

二十几年前的游戏以现在的眼光来看都很简单，容量大小都是以兆(M)计，通常一款游戏的开发周期为 8~10 个月，最主要的是，每款游戏开发都需要重头编写代码，其间存在着大量的重复劳动，耗时耗力。慢慢地，开发人员总结出一个规律，某些游戏总是有些相同的代码，可以在同题材的游戏中应用，这样就可以大大减少游戏开发周期和开发费用，慢慢地这些通用的代码就形成了引擎的雏形，伴随着技术的发展，最终演变成今天这样的游戏引擎。

同样的，游戏引擎出现之后，也反过来促进了游戏的开发。随着显卡性能越来越强，游戏的画质越来越高，游戏开发周期也越来越长，通常都会达到 3~5 年，若是自行开发游戏引擎则时间还会更长，所以大多数游戏公司选择购买现成的游戏引擎，简化游戏的开发过程。

实际的游戏开发过程中，游戏引擎是把游戏与显卡连接在一起的，游戏中的各种特效通过调用显卡来实现。显卡是游戏的物理基础，所有游戏效果都需要一款性能足够的显卡才能实现，在显卡之上是各种图形 API，目前主流的是 DirectX 和 OpenGL，而游戏引擎则是建立在这种 API 基础之上，控制着游戏中的各个组件以实现不同的效果。

在引擎之上，则是引擎开发商提供给游戏开发商的 SDK 开发套件，这样游戏厂商的程序员和美工就可以利用现成的 SDK 为自家的游戏加入自家建立的模型、动画以及画面效果，而最终的成品则是各种游戏。整个关系可用图 2-1 来表示。

图 2-1 游戏引擎与 GPU 的关系图

2.2 游戏引擎的功能

简单地说，游戏引擎就是用于控制所有游戏功能的主程序，从计算碰撞、物理系统和物体的相对位置，到接受玩家的输入，以及按照正确的音量输出声音等等，都是游戏引擎需要负责的事情。它扮演着中场发动机的角色，把游戏中的所有元素捆绑在一起，在后台指挥它们有序地工作。可以说，游戏引擎虽然有着“动力(Engine)”之名，但其实是行“大脑(Brain)”之实，指挥控制着游戏中各种资源。因此，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，哪怕是一个小游戏，都有一段起控制作用的代码。

经过不断地进化，如今的游戏引擎已经发展为由多个子系统共同构成的复杂系统，从建模、动画到光影、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所有的重要环节。一个典型的游戏引擎一般包含以下几个组件：

(1)实时渲染：实时 2D/3D 图像处理是引擎最重要的功能，也是引擎所有组件中最复杂的，它的强大与否直接决定了最终游戏画面质量。

(2)光影计算：决定场景中的光源对处于其中的人和物的影响方式。游戏的光影效果完全是由引擎控制的，折射、反射等基本的光学原理以及动态光源、彩色光源等高级视觉效果都是通过引擎中的光影技术来实现的。

(3)动画技术：可以分为骨骼动画系统和模型动画系统。前者用内置的骨骼带动物体产生运动，后者则是在模型的基础上直接进行变形。引擎把这两种动画系统预先植入游戏，以方便动画师为角色设计丰富的动作造型。

(4)物理系统：它可以使游戏世界中的物体运动遵循客观世界规律。例如，当游戏人物跳起的时候，系统内定的重力值将决定它能跳多高，以及它下落的速度有多快；子弹的飞行轨迹、车辆的颠簸方式等也是由物理系统决定的。另外，碰撞检测也是物理系统的另

一个核心部分，它可以检测游戏中各物体的动态交互情况。

(5)场景管理：负责游戏空间的划分和排序、可见性判断和裁剪，加快渲染速度，在物体空间进行碰撞检测与反馈，提高响应速度。

(6)人工智能：赋予游戏中非玩家角色以可信的智能，增强游戏的可玩性。

(7)人机交互：负责游戏玩家与电脑之间的沟通，处理来自键盘、鼠标、摇杆和其他外设的信号。

(8)网络接口：如果是网络游戏，则网络处理代码也会被集成在引擎中，用于管理客户端与服务器之间的通信。

通过上面的介绍可以了解到：游戏引擎相当于游戏的框架，框架搭好后，关卡设计师、建模师、动画师只要往里面填充内容就可以了。

游戏引擎的作用可以通过图 2-2 所示的一个例子描述：在游戏的一个场景中，玩家控制的角色躲藏在屋子里，敌人正在屋子外面搜索玩家。突然，玩家控制的一个穿迷彩服的士兵碰倒了桌子上的一个杯子，杯子坠地发出破碎声，敌人在听到屋子里的声音之后聚集到玩家所在位置，玩家开枪射击敌人，子弹引爆了周围的易燃物，产生爆炸效果。在这个简单而常见的过程中，游戏引擎便在后台起着作用，控制着游戏中的一举一动。

图 2-2　引擎在游戏场景的后台作用

把这个过程用专业的语言分解一下就会变成这样：首先出场的是可以行动的士兵，也就是人物模型，模型由引擎中的动画系统赋予运动能力，游戏中角色能做出什么动作便取决于动画系统有多强大，人物的真实程度则取决于 3D 模型渲染引擎的能力，这也是游戏引擎最重要的功能之一，游戏的画质高低便由它来决定。其次，士兵碰倒了杯子，这个过程涉及引擎的碰撞检测，它决定不同的物体在接触的时候会产生什么样的结果，有的游戏能穿墙有的则不能便是不同的碰撞检测控制的。例子中设定的是把杯子碰倒了，杯子发出了破碎声。在发生某种动作的同时发出相应的声音则属于引擎中的音效处理了，杯子破碎的声音吸引了敌人的注意，这是引擎中的 AI 智能运算在起作用。最后双方交火引发爆炸，爆炸产生的烟雾、爆炸物飞散则是引擎中的粒子系统在起作用了。

一款游戏能实现什么样的效果，很大程度上取决于所使用引擎的性能，优秀的游戏引擎一定具有如下优点：

(1)完整的游戏功能。随着游戏要求的提高，现在的游戏引擎不再单纯是一个 3D 图