CENGAGE Learning

• 游戏开发经典丛书 •

(美)Allen Sherrod 著
狄东宁 译

DirectX 游戏开发终极指南

Ultimate Game Programming with DirectX

CENGAGE Learning

清华大学出版社

DirectX游戏开发终极指南

(美) Allen Sherrod 著

狄东宁 译

清华大学出版社

北 京

Allen Sherrod
Ultimate Game Programming with DirectX
EISBN: 1-58450-458-7

981-978-7-302-17286-4

Cengage Learning Asia Pte. Ltd.
5 Shenton Way, # 01-01 UIC Building, Singapore 068808

北京市版权局著作权合同登记号 图字 01-2006-7224 号

图书在版编目(CIP)数据

DirectX 游戏开发终极指南/(美) 谢里德(Sherrod, A.) 著；狄东宁 译. —北京：清华大学出版社，2008.6
书名原文：Ultimate Game Programming with DirectX
ISBN 978-7-302-17286-4

Ⅰ. D… Ⅱ. ①谢… ②狄… Ⅲ. ①多媒体—软件工具，DirectX ②游戏—应用程序—程序设计

Ⅳ. TP311.59 G899

责任编辑：王 军 李 阳
装帧设计：孔祥丰
责任校对：成凤进
责任印制：王秀菊
出版发行：清华大学出版社 **地 址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn **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：北京四季青印刷厂
装 订 者：三河市李旗庄少明装订厂
经 销：全国新华书店
开 本：185×260 **印 张**：44 **字 数**：1017 千字
附光盘 1 张
版 次：2008 年 6 月第 1 版 **印 次**：2008 年 6 月第1 次印刷
印 数：1～4000
定 价：88.00 元

序

3D 游戏开发已经成为一个热门的研究领域。对于那些想要涉足这方面学习、研究的人而言，拥有一本这方面的教材已经迫在眉睫。如今市面上这类书很多，但在众多书籍之中这本《DirectX 游戏开发终极指南》却有它的独到之处，它围绕一个实例由浅入深地将游戏开发这个高深的主题娓娓道来。

本书主要介绍在 Microsoft 公司的 DirectX 软件开发包上开发 3D 游戏的全过程。作为一本抛砖引玉的游戏编程书籍，本书尽可能避免对高深的游戏编程理论的介绍，这有利于引起初学者的兴趣。同时整本书结合了一个游戏开发实例，这也可以很大程度地提高读者的学习兴趣。读者学完本书后，应该对如何在 Windows 平台上开发 3D 游戏有一个完整的了解，这对于今后从事游戏开发工作或是深入研究都大有裨益。

本书分为 3 个主要部分，从头至尾介绍了整个 3D 游戏的开发。第一部分重点介绍了使用 Direct3D API 生成 DirectX 图形的内容。第二部分重点介绍了游戏数学、碰撞检测、输入检测和游戏声音等内容。第三部分重点介绍了模型加载和 3D 场景中的动画与场景管理。本书还在附录中介绍了游戏开发的相关书籍和推荐的一些资源。

本书既可作为游戏开发爱好者的入门书籍，也可作为游戏开发人员的培训教材，对那些高级游戏程序员而言，本书也可作为他们的案头参考书。

欢迎各位读者对本书提供反馈意见。我希望读者能从本书中受益，也希望通过读者的反馈意见来了解自己的不足，以求在今后的译作中更多更切实际地考虑读者的需要。请将您的反馈信息发送至 wkservice@tup.tsinghua.edu.cn，我将不胜感激。

译　者

2007 年 10 月

前　言

欢迎阅读《DirectX 游戏开发终极指南》一书。这是一本介绍在 Windows 平台上使用 C++和 DirectX 开发游戏的书籍。我希望每个阅读本书的读者都能够从本书中找到有价值的内容，并且希望读者能够享受到初次开发视频游戏所带来的乐趣。本书旨在为那些没有 DirectX 使用经验的读者提供帮助。如果读者已经有了一定的 DirectX 开发经验，那么也希望您能从中发掘出对您有价值的内容。我希望读过此书的每一位读者都能够与我为伍，并且能够加入到 www.UltimateGameProgramming.com 网站上的其他程序员社区中来。

DirectX API 由图形、输入设备、网络、音效及其他部分组成。本书从头至尾介绍一个名为 Stranded 的第一人 3D 游戏的开发。这样读者可以通过研究该游戏的开发和运行完成本书的学习。相对于当今市面上的专业游戏而言，该游戏比较简单，但这是游戏开发程序员的极为重要的入门步骤。

本书的组织结构

本书主要分为三部分，介绍了游戏开发的整个流程。第一部分重点介绍了使用 Direct3D API 生成 DirectX 图形。其内容涵盖了向屏幕上显示几何图形、图像、文本和图形界面以及设定 3D 光照、创建和显示特效、为游戏编写脚本的所有知识。这部分涉及到本书 1~7 章的内容，为读者创建最终的游戏以及从一个入门级程序员转变为中级程序员打下基础。本部分有很多需要花费读者大量时间阅读的内容，因此读者要确保在进一步深入学习之前理解这部分材料。对这部分材料理解得越深刻，深入学习后面的高级专题就越容易。

本书的第二部分主要介绍了游戏数学、碰撞检测、输入检测和游戏音效。游戏中使用的数学知识可能非常高深，尤其是实现更高一级的物理和图形效果时更能体现出这一点。即使数学功底不是很好，通过耐心学习和不断实践也可以掌握游戏开发所需的数学知识。读者并不需要有数学方面的专业知识，但要乐意于深入学习这方面的内容，因为游戏开发和它紧密相关。

本书的第三部分介绍了模型加载和在 3D 场景中的动画和场景管理。模型加载过程就是将模型的各个组成部分显示到屏幕上。在此将介绍不同的文件格式以及用于驱动这些模型并使其在 3D 空间中移动的技术。本部分最后介绍了读者在深入学习游戏编程中可能遇到的不同的场景管理技术和主题。

本书的最后将完成整个游戏的开发。虽然这是本书最激动人心的内容，但是读者不要为了迫切得到最终的游戏而忽略前面的章节直奔本书最后内容。踏实地学好本书前两部分的基础内容将有助于您超越本书所讨论内容的限制从而开发出自己的游戏。

学习本书的要求

为了最大限度地发掘本书的价值，您需要掌握一些 C++编程知识。本书旨在介绍游戏编程，而不是计算机编程，所以并不需要读者是 C 或 C++专家，但需要读者已经掌握了编写代码的方法。只要读者基本理解类和结构就足够了，而不需要掌握任何 DirectX 知识。

为了编写 DirectX 应用程序，读者要有一套 DirectX 软件开发包或是 DirectX SDK。该软件可以从 Microsoft 的网站下载或是通过本书的附带光盘获得。读者至少要在自己的机器上安装 DirectX 9.0c。虽然所有的 Windows 操作系统都提供了 DirectX，但是读者要确保在自己的机器上安装了最新版本的 DirectX。本书编写的时候，DirectX 的最新版本是 2005 年 12 月更新的 DirectX 9.0c。

CD-ROM 上所有的演示程序都是用 Microsoft 公司的 Visual Studio 2003 编写的。为了打开项目文件，读者机器上至少要安装.NET 2003。读者可以使用以前版本的 Visual Studio 或是完全不同的集成开发环境，但是为了创建应用程序就必须创建自己的项目文件。这并不是很困难的事情。所以，读者要牢记的是，是否已安装了另一个自己喜欢使用的开发环境。

补充材料

Internet 上有很多关于游戏开发和业界发展动态的信息。这里推荐一个值得访问的网站：http://www.UltimateGameProgramming.com。除了有一个访问量很大的社区之外，该网站还提供了大量关于 DirectX 编程、OpenGL 以及游戏引擎编程的信息。同样，读者要牢记的是，如果在运行过程中遇到问题的话，可以使用自己机器上或 MSDN 网站上的 MSDN 和 DirectX 帮助文档。附录 A 中包含了作者向读者推荐的大量书籍和网站。

目　录

1 DirectX 导 论

本章内容

- 本书概述
- 游戏规划
- DirectX 背景
- 手动设置窗口
- 演示程序

从某种程度上讲，游戏编程和开发是极具潜力且激动人心的行业。它发展迅猛、富有挑战性并带给那些为之努力工作并且超越极限的人非常丰厚的回报。游戏开发需要一批极富创造力的人协同工作，从而开发出新的游戏产品。这些人包括软件开发人员、作家、音效师、数学家、物理学家、艺术家以及许多其他人员。每个人面临的职责、应对的挑战和得到的回报也各不相同。游戏产业自诞生以来，规模已得到长足发展。现在它已经可以和诸如电影、音乐、电视这样的产业竞争。这个每年可以带来数百亿美元利润的产业随着时间的推移、技术的强大和效率的提高也在不断地增长。视频游戏已经存在数十年了，而且还在快速增长。随着越来越多游戏玩家的成长，游戏产业带来的经济利润就越来越大。大部分购买游戏的人都是成年人。这没有什么值得奇怪的，因为大多数孩子没有很多的钱去购买游戏，而等到他们长大成人后，就有钱买游戏了。

游戏开发并不总是充满乐趣和令人兴奋。开发交互式软件是一件艰苦且充满挑战的事情，对此从来都不要掉以轻心。开发游戏像玩游戏一样有趣，但同时需要做大量的工作和付出很多的心血。许多人一想到这一点就会因气馁而放弃。游戏开发中最困难也是最让人有挫折感的部分是学习如何开发游戏的方法。学习基础知识就会为你今后提高能力和技术水平打下坚实的基础。如同在任何领域中一样，通常需要耗费几年的时间才能开发出专业级别的游戏，只要有耐心、冲劲和决心，就能做到这一点。

本书为读者介绍使用 DirectX 开发游戏的方法。本章将介绍 DirectX，更确切地说是 Direct3D 图形知识。随着本书的深入，读者同样会学到绘图技巧、输入设备检测和响应、声音回放、场景管理、动画、模型/角色的加载和绘制等内容。

1.1 本书概述

本书将使用 DirectX 软件开发包(SDK)开发一个名为 Stranded 的第一人称射击游戏。本书从头至尾将为该游戏编写代码，这样可以给读者一个清晰的概念，即在 DirectX 基础上开发游戏所需掌握的内容。某些章节的结尾处有一些名为“游戏项目”的内容。这样可以将在当前或前面章节中学到的知识运用到本书的游戏项目中。到学完本书时，读者会在开发自己的第一个 3D 游戏方面取得进步。

本书分为 5 个部分。第一部分涉及到 DirectX 图形学的基础知识，包括 Direct3D 的安装和使用，DirectX 中控制屏幕渲染等内容。读者具备这些基础知识后，就可以立即开发出一个短小而且包含 Direct3D 场景的演示程序。这些基本技能包括绘制基本图形和对象，在几何图形表面显示图像，在屏幕上创建和显示文本以及学会 3D 场景中的游戏开发方法。

本书第二部分涉及到游戏中共用的编程数学知识和碰撞检测。这部分内容非常重要，因为游戏以及其他交互娱乐平台都涉及了大量的数学知识。从渲染图形到移动人物等，所有这些都需要数学知识。使用本书或学习游戏开发时，读者不必具备很高的数学水平，但经验越多，事情做起来就越容易。

本书第三部分将介绍输入和声音这部分内容。在此，将介绍 DirectX 家族中的两部分内容：Direct Input 和 Direct Sound。这些章节同样也很重要，因为能够和软件进行交互(输入)非常重要，这会让产品更像一款真正的游戏。声音增加了游戏真实感，并通过影响场景中的状态和气氛来增强玩家的体验。

本书第四部分讨论了模型加载和动画制作。这几章集中讨论了从文件中加载复杂模型和在 3D 中将数据用动画形式逼真地显示出来的内容。了解游戏数学和 DirectX(Direct3D)渲染知识，就可以让人物逼真地在环境中移动。

本书最后一部分将完成整个游戏的开发工作。此时，读者已经做了大量的工作。这些章节将“整理”在前面章节中学习过的知识，并最终开发出成品游戏。尽管从编码角度而言，该游戏非常简单，但它将会让读者终身获取非常宝贵的学习经验。

读者学习完本书后，一定要查看附录 A 中推荐的书籍和网站。从中可以获取将来在游戏开发中加深学习而要用到的补充资源和参考资料。

1.1.1 编写本书的目的

本书旨在使用入门级游戏程序员知识，开发一个简单的第一人称射击游戏。读者通过编写该游戏，将了解到游戏设计和开发的整个过程。此外，读者同样可以开发一个让自己的朋友和家人留下深刻印象的游戏。

虽然读者在学习本书时，不需要有任何的游戏开发和 DirectX 使用经验，但是应该有一些编程经验。本书是一块垫脚石，它将带领读者进入到游戏开发的广阔空间中。本书并不是一本涉及到游戏开发全部内容的“终极指南”，它只是一个开端。读者可以将其作为一本参考资料加以收藏，在将来需要时可以在其中查找对自己有用的信息。

1.1.2 读者对象

本书面向的读者是那些希望在 Windows 操作系统上使用 DirectX 进行游戏开发的 C++程序员。虽然读者不一定要是 C++大师，但掌握这方面的知识对读者大有裨益，例如掌握面向对象编程。另外，读者不必有任何游戏编程或图形编程经验，就可以享受到游戏开发的乐趣，并从本书获取大量有用的信息。本书仅涉及面向对象编程和软件开发的一般主题，非常适合那些以此为爱好的程序员、学生以及希望在这个行业中找到工作的人阅读。

如果读者想要学习 DirectX，但是又没有任何编程经验，则可以参阅附录 A 中列出的一些书籍。这些书籍有助于读者在编程方面的入门。同样，附录中还推荐了一些该领域的书籍和资源，它们有助于提高读者的技能水平，这些资源已超出了本书的讨论范围，包括编程、软件开发、数学、人工智能、物理学以及高级图形学和引擎编程等。

1.1.3 工具和资源

本书基于 DirectX 9.0 进行开发工作，所以读者务必要有一套最新版的 DirectX 9.0 软件开发包(DirectX 9.0 SDK)。读者可以从 Microsoft 公司的 www.Microsoft.com/DirectX 网站上下载该软件开发包。本书的代码示例同样可以使用 Microsoft 公司的 Visual Studio .NET 2003 集成开发环境编译。读者也可以使用以前版本的 Visual Studio 集成开发环境，但要在该环境中创建自己的项目，以便对代码示例进行编译。如果读者有最新版本的 Visual Studio，就可以直接打开用 Visual Studio .NET 2003 开发的项目而不会出现任何问题。

除了附录 A 中提到可以帮助读者提高编程能力的资源外，读者还可以登录到 www.UltimateGameProgramming.com 网站，使用该网站提供的资源。读者在此可以找到一个提供大量游戏开发知识和文章的社区，这些知识和文章涉及多个不同专题，供读者学习使用，以帮助读者解决学习过程中遇到的问题。像可以随意下载代码示例一样，读者可以自由参与网站上的各种讨论。

1.2 游戏规划

如前所述，本书旨在开发一款简单的 DirectX 第一人称射击游戏。通过本书大部分章节中最后一部分“游戏项目”的渐进学习，读者就可以完成这款游戏的开发工作。读者通过本书前面四分之三内容的学习，可以编写简单的引擎代码，它可以作为最终游戏的开发基础。剩下四分之一的内容将编译和完成最终的游戏。随书光盘提供了本书中所有的开发代码。

1.2.1 项目 Stranded 概述

游戏 Stranded 的风格与《雷神之锤》(Quake)和《虚幻》(Unreal)这类游戏很相似。游戏是在名为 Stranded 的游戏引擎基础上开发的。本书通篇都是同时进行游戏引擎和游戏的开发工作。每章将会解开游戏开发的一部分难题，直到完成一个简单的 3D 游戏为止。当读者学完本书时，如果想要将自己的工作提高一个层次，那么书中提供的任何代码都很容易扩展，或是可以毫不费力地更换其中的代码。

可以使用键盘和鼠标或游戏垫(game pad)等输入设备玩游戏。如果喜欢的话，读者可以添加其他设备，因为 DirectInput 作为 DirectX 的输入，可以方便高效地检测所有输入设备。同样，游戏可以为武器提供 3D 声效，如同在点击游戏主菜单时出现的环境音乐。3D 声音似乎是从游戏中某个位置发出的声音。声音文件主要采用.wav 格式，可以使用如 Reason 3.0 这样的声音/音乐开发软件编制。

Stranded 在环境和环境物体的重力和碰撞中用到了简单的物理学知识。由于物理学是一个不可能详细介绍的庞大领域，所以只能对简单的内容加以介绍。书中同样还介绍了简单的脚本系统，这样初学者可以涉足脚本系统这一领域的学习。同样，附录 A 还推荐了可以帮助读者开发更复杂脚本系统的资源，当然也可以使用现有的脚本系统。在学完本书后，有些程序员可能希望开发出更复杂的脚本系统。

从图形角度而言，游戏非常简单，因为本书目标只是对游戏编程做一个入门级介绍。如果希望深入开发工作，读者在学完本书后，添加更高级的图形也是不难实现的。实际上，可以使用本书的源代码创建更高级的图形，一方面由于本书空间所限，另一方面因为这也超出了本书的讨论范围，所以本书删除了这部分内容。

游戏中使用的人工智能(AI)非常简单。人工智能是一个巨大的研究领域，它能真正让游戏给人留下深刻印象，让人愿意去观看和进行交互。人工智能使读者可以和游戏自身进行交互，但它并不能模仿最新最好的游戏。任何学完本书的人都可以在编写代码的基础上，继续开发更高级和效率更高的人工智能。

游戏本身很简单。主角是一个陷落在地球上的外星人。第一个任务是主角为了返回家乡，要找到一个像飞船一样的物体。在寻找该物体的过程中，主角会遭遇试图抓他并获取其技术的敌人。这意味着有两类角色：敌人和外星人。每一类角色可以共用一种武器。同样，游戏只有单场景：一座荒岛。一旦开发完这个基本游戏，读者就可以为其添加更多的武器、角色、场景等。

1.2.2 设计概述

如果读者计划涉足游戏开发领域，那么游戏设计是要面对的一个重要而广泛的领域。关于这一主题，有很多文章和书籍可以引导读者进入该领域的学习。然而，本书只是开发一个短小而简单的游戏，因此不需要很多的游戏设计知识就可以完成开发工作。这并不意味着无需进行游戏设计。实际上，由于本书关注的是 DirectX 和游戏编程，因此它采用了一种非常规的方法替代专业的游戏设计。附录 A 推荐了许多游戏设计资源。因为

游戏设计本身也可以用一本书来诠释，所以本书像对待其他部分一样，并未用太多的篇幅来讨论这个主题。第 2 章将讨论游戏规划，但目前仅对该主题稍作介绍。

在开始真正的游戏开发前，需要编写大量的代码来构建本书的项目游戏。所有现代游戏都运行在一个称为游戏引擎的平台上。游戏引擎指的是可以确保在较高层面上运行游戏的代码集。例如，可以编写负责渲染工作的代码，而不是直接和 Direct3D(或 OpenGL 等)打交道，这样可以构建一个封装在场景后面的渲染引擎，而任何使用该代码的人都只需考虑渲染引擎即可，而不用考虑它要用到的图形用户界面(GUI)。这样在底层改动代码时，例如用 OpenGL 替代 Direct3D，那些使用游戏引擎的人会对此毫无察觉。同样，在学习过程中，本书中的项目需要一些可以使编译、调试和纠错等工作简化的功能。这包括错误日志记录系统、简单的脚本运行等。

1.2.3　引擎设计概述

如前所述，游戏引擎是一系列高级代码，读者可以以它为基础开发自己的游戏。现代游戏引擎已经对那些使用它的人隐藏了底层实现的细节和规范。例如，可以在 OpenGL 和 Direct3D 的基础上开发渲染引擎，这样，引擎用户就不需要知道使用的是哪一个渲染引擎，尽管用到了一些底层的东西。读者可能会说，那样游戏引擎会变得很复杂。游戏引擎包括：渲染引擎、物理引擎、声音引擎等。游戏引擎本身只是一个由更小的引擎组成的集合。游戏引擎或它涵盖的内容并没有一个精确的定义。对视频游戏而言，它的游戏引擎至少要包含渲染引擎和输入引擎，这是必需的，否则，就不能称其为交互式游戏。

由于游戏引擎开发这个主题很复杂，所以本书并不会试图去开发一个很庞杂的游戏引擎。实际上，本书将在构成 DirectX 的不同 API 基础上开发出一个简短的游戏引擎。学完本书足以让读者了解引擎，读者不只是可以编写出一个简单的游戏引擎，还可以在这个引擎上继续开发工作或对其改动以满足自己的需求。对于那些讨论引擎体系结构和开发的全部资源而言，这不是问题。虽然没有一本书讨论了类似的游戏引擎的开发，但它们都会给读者提供宝贵的提示、想法，并引领读者深入考虑开发代码的方法。

本书的游戏引擎包括构建在 Direct3D 基础上的渲染引擎、构建在 DirectInput 基础上的输入系统、构建在 DirectSound 基础上的声音系统、处理碰撞和运动的物理系统、动画系统和一个简单的人工智能系统。该引擎代码还包括创建错误日志、使用命令和属性脚本的基本脚本能力和图形用户界面(GUI)系统。GUI 系统用于创建主菜单、加载场景等。

Stranded 游戏引擎中并未涉及网络系统，这是因为网络是个很庞大的主题，它超出了本书为初学者介绍的知识范围。其他一些系统至少可以做一些简化，而网络系统编写起来并不容易。当然并非所有的网络系统代码都很高深，或是很庞大，但要了解的内容已经超出了本书的讨论范围。由于本书不是一本讨论引擎开发的书籍，所以只关注 DirectX 的不同组成，并将其集成在一起开发出一个简单的游戏。游戏程序员和游戏引擎程序员完成的是两种不同的工作。游戏程序员只与开发游戏的游戏引擎打交道，而游戏引擎程序员要开发出其他人用于开发游戏的引擎。

1.2.4 渲染系统

渲染系统将会是 Stranded 游戏引擎中最大的系统。和专业级的渲染系统相比较，该系统虽小却具备了本书完成游戏开发任务所需的各种功能。该系统可以在屏幕上绘制几何图形，显示文本，渲染图形用户界面(GUI)，实现硬件光照。同样，它还要将图像添加到场景中的几何图形表面上。对以前没有开发过渲染系统的人而言，这似乎要完成大量的工作，但实际上工作量很小而且很简单。

本书从头至尾将介绍渲染系统的开发，而不只是在某一章中介绍它的开发。随着本书的深入学习，读者在学习一些涉及图形学的新知识后，就可以将其添加到渲染系统中。尤其对 DirectX 新手而言，把每样东西包含进开发的渲染系统中就变得简单多了。学完本书后，读者就会开发出一个渲染系统，读者可以在这个基础上继续开发或进行修改以满足未来的游戏项目要求。只要规划仔细、考虑清楚，那么为渲染系统添加代码的工作就简单多了。如果在开始编码前没有做任何规划，那么在项目开发过程中和项目结束后就会产生许多令人头痛的问题和缺陷。

1.2.5 输入系统

检测输入是游戏开发中非常重要的主题。使用 DirectX 的 Direct Input API 可以实现对不同设备的输入检测，这里不会出现任何问题。如果是输入设备，那么可以对它做输入检测。如果是键盘、鼠标、手柄、游戏垫、方向盘等，就不会出现任何问题。DirectInput 有一个接口，可以查找和检测目前市场上或将来作为输入的设备。

当前的主要输入设备是键盘、鼠标和游戏控制器。游戏控制器指的是任何既非键盘也非鼠标的输入设备，如手柄、游戏垫、方向盘等。本书代码主要针对键盘、鼠标和游戏垫进行检测。当然，由于游戏控制器采用相同的处理方式，所以如果有手柄或方向盘，那么可以编写代码对其进行测试，并在最终的游戏中使用该输入设备。如果只有键盘和鼠标，就不能使用游戏控制器了，但是可以编写、编译使用游戏控制器的代码。游戏控制器可以非常便宜，如果计划做这方面的测试或在代码中支持游戏控制器的话，推荐购买一个游戏控制器。例如，有些游戏控制器只要花费 20 美元就可以买到。

1.2.6 声音系统

开发人员可以使用 DirectSound 处理游戏引擎中的声音。DirectSound 是 DirectX 内核中的一个 API，负责处理全部的声音。声音系统是游戏引擎中较小的系统，因为它只要很少的代码就可以将声音调用和播放出来。引擎中的声音用于播放背景音乐和 3D 音效。这些声音包括武器音效和游戏玩耍过程中播放的其他不同音效。所有的声音都从文件中加载，要么是完整加载该文件，要么是从硬盘上以流形式播放，例如：声音回放。这全都可以用 DirectSound 处理，本书将以它为基础开发一个声音系统。

1.2.7　物理系统

物理系统中涉及的内容很多，在视频游戏中它可以构成现实，也可以分解现实。物理学正逐渐成为 3D 游戏的重要组成内容，并且它是所有有抱负的游戏开发人员都要学习的知识。对本书而言，物理学将很简单并且只涉及到重力和碰撞检测。对一个简单的第一人称射击游戏，这已经足够了，但如果需要更多物理方面的知识时，可以查看附录 A 获取补充资料。物理学不止涉及数学，还要了解工作环境及其运作方式。

《半条命 2》(Half-Life 2)以及其他现在广为流行的游戏都将物理学带到了新的高度，并且似乎已经提高了整个行业门槛。物理学可以为游戏增加许多内容，通过增加的物理特性，可以改变不同的人与环境交互的方式，并增加回放功能。这样的物理系统可以非常复杂，并且开发耗时，但一旦完成该系统的开发，它就会让开发出来的游戏成为不朽的传奇。它会让游戏玩家对游戏充满兴趣，并提供多重选择，这正是游戏好坏的差别之分。

1.2.8　动画系统

屏幕上看到的内容非常重要，这可以增加玩家对游戏的真实体验感。物体和人物的移动方式会对游戏软件的真实感程度产生巨大的影响。不流畅的动画或是缺少动画都会分散游戏玩家的注意力，尤其动画是游戏(例如，搏击游戏)中将要发生的内容时，更是如此。

有几种不同类型的动画。过去，游戏主要采用关键帧动画。这类动画与卡通动画非常相似，因为人物动画中的每一帧代表不同的姿势。在游戏中，这意味着它要尽可能的保存游戏中每一个动画实例的几何形状。帧数越多，动画就显得越平滑，但这样做要消耗大量的内存。当开始使用非常详细的多边形角色和大量的动画时，该问题尤为突出。同样，由于动画数据不能共享，为了让每个人物充满生命力，就要做大量的工作。

本书使用一种名为“骨骼动画”的技术。本书稍后将详细讨论该技术。目前，只要知道它可以根据动画数据信息和通过骨架(skeleton)获取单一模型(例如，人物、物体等)并移动该模型中的每一片即可。骨架会告诉模型，模型的哪一片与哪种骨骼关联。为了让动画显得平滑，骨骼会告诉模型移动不同部分的方法。骨骼动画显得极为真实。使用骨骼可以节省大量的内存，因为这样不必为动画中的每一帧保存几何图形。实际上，可以为每个关键帧保存骨骼位置，这样只需要更少的模型几何图形即可实现动画。

1.2.9　人工智能(AI)系统

游戏玩家玩视频游戏的时候，会受到挑战。当游戏玩家与计算机作战时，他们必须找到对抗，否则很快会失去注意力。如果一款游戏难度系数太低，那么它的重玩值会相当低。在此之上，如果游戏过于简单或过于困难，那么游戏玩家也许不会对其有乐趣。在视频游戏中，AI 控制计算机对下一步做出响应。AI 形式多种多样，可以用多种方法实现，诸如脚本系统、硬编码行为等。