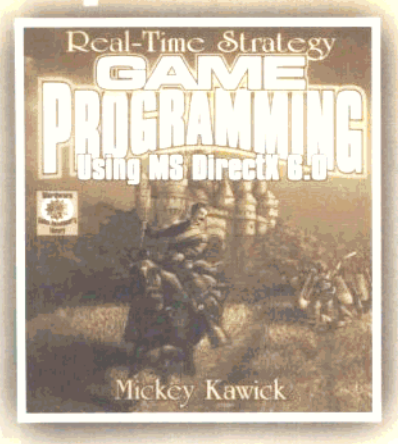
引 进 美 国 最 新 计 算 机 图 书 系 列



Real-Time Strategy

GAME PROGRAMMING

用DirectX6.0开发即时战略游戏

[美] Mickey Kawick 著

揭金良 龚 灏 周家纪 等译
朱 剑 喻秀英 刘 厉

内容新
编译精

电子科技大学出版社

用 DirectX 6.0

开发即时战略游戏

原著：[美] Mickey Kawick

揭金良　龚　灏　周家纪
朱　剑　喻秀英　刘　厉　等译

电子科技大学出版社

图书在版编目（CIP）数据

用 Direct X6.0 开发即时战略游戏 /（美）卡威克（Kawick，M.）著；揭金良等译.—成都：电子科技大学出版社，2000.11

（最新引进美国计算机系列图书）

ISBN 7-81065-546-9

Ⅰ.用… Ⅱ.①卡…②揭… Ⅲ.游戏卡-图形软件，Direct X 6.0-程序设计 Ⅳ.TP317.4

中国版本图书馆 CIP 数据核字（2000）第 54118 号

声　明

用 Direct X6.0 开发即时战略游戏

原著：[美] Mickey Kawick

揭金良　龚灏　周家纪　朱剑　喻秀英　刘厉　等译

出　版：电子科技大学出版社（610054　四川成都建设北路二段 4 号）

责任编辑：李建川

印　刷：西南冶金地质印刷厂

开　本：787×1092　1/16　印张：35.75　字数：828 千字

版　次：2000 年 11 月第一版

印　次：2000 年 11 月第一次印刷

书　号：ISBN 7—81065—546—9/TP・361

印　数：1—3000 册

定　价：70.00 元

前　言

很高兴此书能与读者见面。游戏是我的生命，我热爱它们，至今没有一种游戏能像实时战略游戏这样令我热血沸腾。伴着画面的展开，通过精心的谋略和布置，您逐步积累经验从而成为伟大的将军或国王。这些游戏各有特色，但每个游戏者首先需要做的就是学习。这个过程往往很长，而且通常在一个游戏中得心应手的战略不一定适用于其他游戏。所以游戏高手们通常都经历了上百个小时的苦战。还好，这并不困难，因为很少有实时战略游戏(特别是多人游戏)能够在两小时内完成。我曾经和一样玩友玩帝国时代，大概从早上十点一直玩到下午三点，将近五个小时。

在这本书中，我希望能够较为全面概要地介绍实时战略或实时策略游戏的开发过程。许多基本对象在一些细节上是统一的，书中还特别强调了图形接口 DirecDraw 的运用和画面的处理。书中提供了大量代码，同时在随书附的光盘上还附有 14 万多行原代码的库字典，读者尽可使用或修改。代码的行数当然不值一提，但却是经过测试的，非常之稳定，可以适用许多不同的环境。其中大量代码在书中都有详尽的注解，但更多的是留给读者自己去摸索。

书中我用了很多代码以简化理解我们所要讨论的概念，但这还不够。希望读者在阅读过程中试着对代码进行编译并随时运行，从而对代码的编写了然于胸。读者需要研究交叉的参考和多重文件以获得对代码的交互影响及所有代码如何工作的充分认识。

在书中列举了基于窗口和基于 DirectDraw 的代码，包括一种概括了 DirecDraw 所有要素的类，可以使您工作更加轻松自如。书中还收集了 DirecSound 中对于声音加载和管理体系方面的一些资料，使得声音的管理也轻而易举。本书涉及了所有关于线条描绘、初步管理、剪切、矩形的描绘和填充以及位图的绘制等基本概念。同时详尽介绍了如何完成图形的映射和压缩以消除冗余信息。书中还用较多文字探讨了实体的操纵和基本的 AI 概念。对接口管理、按钮的生成、窗口的管理和消息传递也有所提及。另外还广泛讨论了即时战略游戏地图的路径，包括地图自身的许多小细节。

附送的光盘中包含了许多具有个人特色的目录。另外，还附带了我个人的初始化编辑器，确实不错。读者还会看到一个 tile 系统范例和非常重要的文件库。这个库包括了本书所有的基本文件和大量的代码。别担心，它们都经过了可靠的验证并且可读性极好。千万记住将文库拷贝到您的 C 盘 Library 目录下。其所有对应的代码都在基本目录中。

读者还应该买一套 Visual C++作为工具。我也极力推荐选用 Borland C++。这一产品是不等同的，并且简化了工具的开发。所有我为各种游戏开发的基于窗口的工具都是用该产品做的。

祝您在开发实时策略游戏或实时战略游戏时好运。希望您会成功，因为那将意

味着我们会有又一个新的游戏可玩，它将使我们的游戏空间更加宽广。如有什么问题，我将会在 Mkawick@sprintmail.com 一一作答。

译 者 序

实时战略游戏是一种人见人爱的游戏。伴着画面的展开，通过精心的谋略和布置，就可掌握从编程到游戏使用的全过程。此书较为全面的介绍了实时战略或实时策略游戏的开发过程。许多基本对象有一些细节上是统一的，书中还特别强调了图形接口DirecDraw的运用和画面的处理。

作者在书中提供了大量代码，供读者在学习中使用，只要读者试着对代码进行编译并随时运行，就可学到并掌握大量的编程技能。此书可作为本科生研究生计算机科学及游戏爱好者的参考书。

揭金良副教授负责1、9、13章的译校，龚灏负责5、14、15章的译校，周家纪教授负责16、20章的译校，朱剑负责6、7、8章的译校，喻秀英负责2、3、4章的译校，刘厉负责17、18、19章的译校，罗震负责10、11、12章的译校，揭金良副教授对全书进行了审校。

原书风容严谨，具有较强的理论性和实用性。译者力求反映原书的特点和风貌，但由于时间关系及水平有限，不当和疏漏之处在所难免，敬请广大读者批评指正。

译 者

2000年4月

目　录

第一章

游戏就像是玩家们的血液，我们在游戏中生存和死亡。游戏是对我们意志力、灵活性、直觉、敏锐性、资金，甚至于时间驾御能力的挑战。我听说早在 1994 年旧纳西州的工人就曾热衷于 Windows 3.1 版本的纸牌游戏，以至竟组织起来帮助人们分享游戏乐趣。

游戏是很耗精力的 。您得掌握许多的规则，生成大量的键盘命令，学习有关游戏的顺序。最好的游戏是可以凭直觉来完成的，不必花太多时间去掌握规则；一个需要 60 个不同按键的游戏实在有点拿不出手。

大多数人都玩过一些电脑游戏。很多人热衷于微软的纸牌和挖雷游戏。还有人去买游戏来玩。鼠标在过去的 8～9 年中一直发挥重要作用，现在，鼠标已经成为主要的输入设备了。除了作家或程序员，一般人用键盘的时间决不会多过用鼠标的时间。

但很多人并不认同设计游戏是一项工作。其实即使是一个很平常的游戏也需要上万个小时，而像 Final Fantasy VII 这样的大游戏，则需要大约 600000 多个小时了。大多数美国人一年工作约 2000 个小时，而游戏设计师和有关人员却要工作 2500～3000 小时。著名的游戏设计师 John Carmak 每年工作近 4000 小时。许多人认为他是个工作狂，但他确实开发了大量游戏。几乎所有 Quake 和 Quake II 以及工具的代码都是他独自完成的。开发 Quake 用了近两年时间，也就是说他一个人为这个游戏投入了 8000 多个小时，当然，还有无数的平面设计师、美术师、制作人员等为之付出了辛勤工作。

概　述

游戏带给人们的是又一崭新的空间。现实中战争的世界永远无法像实时战略游戏般赋予我们如此大的权利：游戏中，你就是指挥官、将军、甚至上帝。战场上，网上另一端你的朋友——游戏中的对手可能会因指挥无方而被你打得尸横遍野。电脑代替了真枪实弹，怪兽随时冲进门来想要杀死最后活着的人类。与第一人称类型战略游戏有所不同的是，军队在玩家们的操纵下产生，在画面上自由行动。谁能够保证生产、训练好军队、有效利用资源，谁就能成为常胜将军。惟一要注意的就是控制好局面的同时记住自己的军队在哪里。

以上都是现在最富挑战性的游戏。它们中除少数非常出色外，大多是泛泛之作。一些成功的游戏有 Warcraft II ，Command and Conquer ，Dungeon Keeper (略有不同)，Age of Empires，Total Annihilation ，以及 Dark Reign。它们可令玩家立刻得偿所愿。各个游戏

的难度水平是不相同的，但大多是交给玩家一个发展领地。这些游戏都非常激烈，钢铁般的意志不可或缺。我的许多玩 Quake 的朋友一致认为实时战略/策略游戏确实更加激烈，难以掌握。

从创作的立场来看，设计这些游戏是非常具有挑战性的。首先，我们需要有好的构思。别的很多好游戏就几乎没有什么故事情节。绝大多数 RST/RTT 游戏（实时战略/策略游戏）需要很多故事，却没有一个精彩的，让人感觉他们各行其事而我们仿佛置身事外。游戏设计的第二要点是一定要有先见之明。要把它当作一项非常困难的，也许是最困难的项目之一，来进行周密的考虑。初步构思一个框架并不断完善是可能的。整个设计过程中，会不断有新的创意产生，还可以不断借鉴别的游戏的过人之处。最后一点，游戏产业最重要的因素当属人才。你得有高水平的美工师、设计师、乐师，以及优秀的程序员。

这本书会让你了解游戏制作的全过程，包括游戏设计文挡、技术支持文挡、基本的美工概念、图形处理系统、基于块的系统管理、布景、建筑、路径、各种行为、团体行动、界面设计，并提供大量可重用的代码。

C++语言在编程世界中正变得越来越流行。这本书中所有库都能在 C++中找到。所有代码都可以随意使用，所有介绍到的方法力求是最好的。它就像本技术手册，能帮助你熟悉游戏制作过程中的很多要点。当然，这本书并不是应有尽有，它并未涉及 3-D、网络、代码管理、代码传递和调试。有太多的书对这些方面进行了详尽的阐述，因此这里就不再多费笔墨。

这本书有一定难度，需要逐步摸索。要勇于尝试书中给出的例子，相信你会受益匪浅的。

什么是实时战略游戏?

这确实是个问题。从玩游戏来说，战略游戏的确可说是现在、也许不久的将来最为复杂的电脑游戏了。它让玩家不停地思考，玩家将在一个实时的战斗环境下指挥一支军队，或者至少是一小队士兵和坦克。所谓实时是指不论游戏者参加与否，行动都会继续进行。就像在所有动作游戏中一样，事件不断地突然发生，动物满屏地到处跑，危机随处可见，玩家们必须提高警惕。

玩家们必须懂得怎样修筑城池。建立一个城市并让它运作良好，大概是这类游戏中最难的一点。要考虑下一步修建什么、哪些建筑应该升级、应该开采哪些资源、每种建筑要修建多少，想做好这些，没有一点计划和直觉是绝对不行的。

玩家们还得保持资源平衡。大多数实时战略游戏对每项任务都有一定的资源限制，比如钢铁、能源、食物等等。玩家如果修建了太多的建筑，可能就无力让它们升级或应付战斗的需要。

玩家们还要会应付突发事件。比如敌军正大举发起攻击，而你的军队却正在侵略别的地方，这时你的步兵又被狮子吃掉了，该怎么办呢？

这类实时战略游戏都各有特色，使得它们与同类的其他游戏区别开来。它们都有一个

主题或重点：战争、外交、资源配置、建设、开采、技术改进等等。这类游戏也都有特定的环境：太空、水下、森林、西部风情、资本社会，从中世纪到遥远的未来，再到科幻世界（海怪、小妖精、巨龙、仙子）……真是包罗万象，应有尽有。

实时战略游戏与实时策略游戏

在实时战略游戏中，战略与策略是两个重要因素，二者缺一不可。你会发现当二者出现在同一个游戏中时，其实有着许多相似之处。“实时策略游戏”是一种新的说法，因此许多人在说到这类游戏时仍用更通常的说法——“实时战略游戏”。

战略涉及到了后勤学、建筑与构造以及资源管理。在 Webster 的 Web 页上对战略有如下解释：

①不论在战争或和平时期，一个政府或一些政府都需要用以为自身政权提供最大支持的政治、经济、心理、军事等方面所采用的知识和技巧；在现代战争环境下用于与敌人作战的方法和技巧。大量战术的运用。

②周详的计划或部署，即一条高明的策略。为达到某个目的而产生的计划或计谋。

③在争取进步与胜利过程中为完成一定的重要目标而作出的一些改进和适应。

战略就是运用一切必要手段来建立一个社会，这也就意味着税收、基本设施的建设、建造太空船基地、开采资源、进行基本的城市管理。

策略则涉及到建立一个社会的军事措施。Webster 的 Web 页上是这样说的：

1 ：完成任务的装置。

2 ：在战斗中运用武力的方式。

大多数策略涉及到进攻与征服敌人，破坏敌方建筑，击溃敌方装甲部队，三队人马同时从三个方向进攻敌人，进攻前集结大队人马，派遣拦截机等等。策略通常是一个进攻的概念，而不是防守，当然，能在整体进攻中构筑稳固的防守对策略来说也必不可少。

《PC Gamer》杂志首先对实时战略游戏（RTS）与实时策略游戏 (RTT)进行了比较。尽管这两类游戏之间相似点多于不同点，但二者强调的重点是不同的。如果你在游戏中建设和资源开采都非常简单快捷，玩家着重战斗和策略，那么这个游戏倾向于 RTS 游戏。反之，如果需要长时间进行建设，那么称之为 RTT 游戏则更为贴切。RTS 游戏的特点还有：各类建筑有一定的建造顺序，玩家必须随时对资源的开采进行调配，另外还有许多的琐事需要亲力亲为。如果游戏具备这些特点，那么毫无疑问，它是一个 RTS 游戏。

所有 RTS/RTT 游戏都有一些战略因素，比如建设、资源管理、开办工厂以及整体建设的本领。所有 RTS/RTT 游戏也都有一些策略因素。它们都有整体的行动和部署，对小队装甲集团的管理，以及针对增强的性能和防御能力而进行的单位升级。

三个阶段

RTS/RTT 游戏是实时的，所有玩家开始时都是平等的，而最终其中一个会消灭其他人。这类游戏一般分为三个阶段。

游戏开始阶段

游戏的开始阶段是最长的。玩家的资源只够建造最便宜的单元，而这些在游戏的后面阶段往往一点用都没有。还可以修建一个支持设施（一栋楼房或一个养殖场），雇佣一些散工（比如资源开采者），再建造一个兵营（一个训练中心或用来修建简单单元的工厂）。首要任务是立刻建造一个单元来开采资源，接着开始寻找新的资源，这些资源往往分布在每个玩家的基地附近。玩家需要对这些资源分派更多的工人，直到拥有 10～13 个资源开采者为止。读者也许会问："我怎么知道该分派多少个呢？每个游戏都是各不相同的呀！"当然不同，但它们似乎在一开始都需要相同数目的散工，我能想到的惟一真正的例外就是 Total Annihilation。

接下来，一旦拥有了 6～8 个工人，就开始修建兵营，这时玩家应该已经具备了足够的资源。只需要一个工人完成开采资源的过程即可。接着，开始考虑防守。保持所有工人的良好状态，再造一些士兵，将单元升级，还要修建所有剩下的基础设施（通常所有基础设施都是必不可少的）。这一过程一般为 10～20 分钟，视不同的游戏而定。

游戏的第二阶段

拥有了 4 或 5 个建筑、足够的资源、一些升级的单位和一支较小的军队之后，感觉到自己逐步强大，现在，该是全面集结部队的时候了。玩家可以根据自己的想法控制整个游戏下一步的走向。如果想采取防守，可以开始修筑城墙、塔楼、并构筑一条防线。如果想进攻，可以建立攻击性部队并让他们为即将发动的第一次进攻作准备。如果想要扩张，那么就派遣一些工人出去在地图的另一部分建立一个基地或别的什么。如果着重技术，你就筑较少的防御工事，而集中精力开展科学和研究工作以取得技术上的进步，以便建造更加强大的星船。

战斗大多发生在第二阶段，这一阶段，需要抢夺更多的资源来建造单元以击退来犯之敌并大举进攻对手。往往会有一个游戏者占有优势。在大多数游戏中，玩家必须将地图侦察清楚，而这也在第二阶段完成。

游戏决胜阶段

这是最困难的部分。胜负在此决出。资源和人马越来越少，一切都越发关键，要更加认真地考虑如何对敌。地图已经几乎完全可见，将有一些非常戏剧性的战斗，但大多数玩家都已没有多少资源了，因此所有玩家都将停留在他们在第二阶段所达到的技术水平。这个阶段所需时间与第一阶段基本相同。

游戏结束

肯定将会有一些单元残留下来，别的玩家几乎丧失了竞争资源和建设新单元的任何能力。要当心，一些玩家可能会把一些部队派到隐藏的角落并重新建立单元，从而赢得战争。

什么是 DirectX?

Microsoft 公司众多优点之一就是其接管新的领域（游戏 API 就是其中之一）的能力。API 是一种应用程序接口（application programming interface）或一组互相联系函数。

DirectX 系列有五种产品。DirectDraw 使曾一度非常困难的基础图形处理尽在掌握之中。由于 VESA2 的出现，屏幕的内存定位是一个连续的块，而不像以前的视频存储段。基本上，从 DirectDraw 返回的指针是指向可以开始描点的视频缓冲的地址。这是取得的一个主要的进步。

DirectSound 用于声音处理。只需要加载一个 WAV 文件到缓冲器中并让 DirectSound 播放。这只是一个小工具，在第 20 章中有所提及。

DirectPlay 是一个控制网络计算机的接口。自从 DirecX5 开始，它就是网络通讯的一个稳定手段。据说，6.0 版本中在打包尺寸方面已有很大改进，尽管 5.0 版本中每个包裹只需要 80 字节。另外，同步对象已经被引入并保证 IPX 协议（一个可能很快风靡的协议）所支持的包裹。DirectPlay 还没有平滑的接口而且工作起来可能有一点困难。

DirectInput 控制输入输出，但它最大优势在于对操纵杆的控制。在游戏的进行中输入输出可以同时进行，这是就键盘和鼠标输入而言。由于没有实时战略游戏使用过操纵杆，在本书中 DirectInput 并不十分有用。

Direct3D 也许是 DirectX 家族里继 DirectDraw 之后最为有用的产品。它为用以 3D 播放的所有视频卡（或者几乎所有）提供了一个相当好的接口。有人认为 OpenGL 更好一些，在大多数时候他们是对的。DirectX5 以及后面版本引入了一个面向 API 的新功能——DrawPrimitive，它使得 DirectX 从此可以画任何你想要的东西。曾经有一个噩梦般的对象：执行缓冲，使得 DirectX 编程变得非常乏味。现在大不相同了，当然 DirectX 仍然存在问题。但它的确非常有用，所有视频卡制造商都对其进行了支持。

所需的工具

开始工作以前需要有 DirectX6 以上版本，因为本书介绍了 DirectX 6 的一些功能，而这些功能在以前的版本中是没有的。另外，可以通过 http://www.microsoft.com/directx/default.asp 下载 DirectX 更高的版本。

还需要一个编译器。就编程和调试来说，Visual C++5.0 或更高版本就已经很好了，