



电脑报 东方工作室

PC

游戏编程

(人机博弈)

王小春 编著

轻松学编程
快乐写游戏



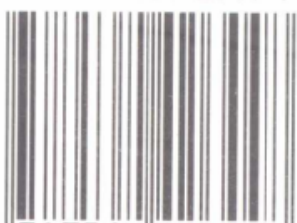
重庆大学出版社



总 策 划 陈 其

封面设计 金 华

ISBN 7-5624-2644-9



9 787562 426448 >

ISBN 7-5624-2644-9/TP·370 定价：38.00元（含1CD）

快乐写游戏 轻松学编程

PC 游戏编程(人机博弈)

王小春 编 著

重庆大学出版社

内 容 提 要



本书是一本专论机器博弈的作品。详细披露了编写人机对弈程序的原理,技术和各种相关内容。包含一个完整的中国象棋人机对弈程序和一个完整的五子棋人机对弈程序实例。毫无保留的展示了估值核心,走法产生,以及约十种不同的搜索引擎,彻底解析了高性能博弈程序的秘密所在。实用性是本书的最大特点,本书的目标是让一个粗通程序设计的人在一个月内写出令人惊讶的人机博弈程序。完全没有一般人工智能书籍晦涩难懂的感觉。

图书在版编目(CIP)数据

PC 游戏编程(人机博弈)/王小春编著. —重庆:重庆大学出版社,2002.5

(快乐写游戏 轻松学编程)

ISBN 7-5624-2644-9

I. P... II. 王... III. 游戏—应用程序—程序设计 IV. G899

中国版本图书馆 CIP 数据核字(2002)第 035762 号

快乐写游戏 轻松学编程

PC 游戏编程(人机博弈)

陈 其 总策划

王小春 编 著

责任编辑:陈 其

版式设计:吴庆渝

责任校对:廖应碧

责任印制:张永洋

*

重庆大学出版社出版发行

出版人:张鸽盛

社址:重庆市沙坪坝正街 174 号重庆大学(A 区)内

邮编:400044

电话:(023) 65102378 65105781

传真:(023) 65103686 65105565

网址:<http://www.cqup.com.cn>

邮箱:fxk@cqup.com.cn (市场营销部)

全国新华书店经销

重庆升光电力印务有限公司印刷

*

开本:787×1092 1/16 印张:16.25 字数:418 千

2002 年 6 月第 1 版

2002 年 6 月第 1 次印刷

印数:1~5 000

ISBN 7-5624-2644-9/TP·370 定价:38.00 元(含 1CD)

本书如有印刷、装订等质量问题,本社负责调换

版权所有 翻印必究

总 序 言

陈 其

《快乐写游戏 轻松学编程》丛书是重庆大学出版社为广大计算机编程爱好者和电脑游戏玩家送上的一份厚礼,是一套集学习、娱乐于一体的,全新教授模式的好书。全套书由陈其总策划,在多维图书策划中心以及各游戏工作室的鼎力协助下得以顺利出版。现就丛书的有关问题作出说明。

编程和游戏

程序是计算机的灵魂,掌握了编程技术就可以随心所欲地让计算机为你服务,让它实现你的梦想。但学习过程中大量的命令和语句又让人感到枯燥乏味,而每一个学编程的人都有过面对一大堆熟悉的命令却组织不起一个像样的程序的经历。于是我们联想到了让很多朋友都着迷的程序——电子游戏。

1) 第9艺术

电子游戏如同戏剧、电影一样,是一种综合艺术,并且是更高层次的综合艺术,它的出现代表了一种全新的娱乐方式——交互式娱乐(Interactive Entertainment)的诞生,而且从它的诞生到现在一直以其独特的魅力吸引了许多玩家,同时也激发了更多的人想写游戏的愿望。

一种事物,当它具有丰富而独特的表现力时,当它能给人们带来由衷的欢愉时,当它表现为许许多多鲜明生动的形象时,它就是一种艺术。电子游戏已经成为一门艺术,继绘画、雕刻、建筑、音乐、诗歌(文学)、舞蹈、戏剧、电影(影视艺术)之后人类历史上的第9艺术。20世纪70年代,出现了第一批简单的电子游戏;今天,它已经发展成为拥有亿万游戏迷的独立的新型艺术样式,向世人显示了其强大的艺术生命力。《文明》、《Doom》、《魔法门》……一个又一个奇迹在产生,进入这个行业成了很多人的梦想。娱乐界的大腕:卢卡斯、派拉蒙、华纳等都已致力于电子游戏产品的开发,并推出了一大批优秀的交互式电影(Interactive Movie)。在世界范围内,电子游戏业的利润已经超过了美国的电影工业和日本的汽车工业。相信不久的将来必然有一大批杰出的电影导演和真正的艺术家投身于电子游戏艺术作品的开发。而VR头盔与3D音效卡的诞生已使电子游戏远远跳出了一般电影所能达到的视听层次。可见,电子游戏已经将视听综合艺术推向了一个崭新的高度和崭新的领域。

在中国,电子游戏曾一度被称为是“电子海洛因”,一些教育界人士痛斥电子游戏是如何毒害青少年,如何损害人的健康。其实与其千方百计扼杀它,还不如共同想办法来扬其长、避其短。因为绝大多数反对电子游戏的人,并不是反对电子游戏本身,而是反对电子游戏中存在的消极面。正如水能载舟亦能覆舟的道理一样,任何事物都有其两面性,关键是怎样利用好的那一面为人类造福。

本丛书正是要利用电子游戏的积极面,将枯燥的学习融入轻松的游戏之中,达到寓教于乐的目的。

2) 培养全局观

许多刚学编程的朋友总是把大量的精力花在了命令和语句上,或是集中精力去学习那些复杂的函数。他们都忽略了怎样去实现一个完整的程序,所以有很多初学者到现在还没写过一个完整的程序。为了避免这种情况,在编写游戏实例时,各书都使用了简单而功能强大的游戏开发引擎,读者能非常轻松地学会如何显示图像文件、播放声音及控制输入设备等游戏中必备的功能,然后把注意力集中到如何实现一个完整游戏的过程及原理上来。

通过细致的讲解,读者朋友很快就能从实例中体会出程序全局观的作用和地位,并在一步步的学习后掌握它。

编程工具

作为一名程序员,要做的第一件事就是选择一把顺手的武器——编程工具。做程序的朋友都知道,比较流行的编程工具颇多,比如:VC,VB,DEPHI、汇编等等。由于本丛书是从编写游戏出发的,而为了能够完成一个完美的游戏,编程工具应具有贴近底层、代码运行速度快、便于优化等优点。于是 VC 成了不二之选。

初学 VC,会因为观念的改变而不知所措。其实,每个人时刻都面临着新知识的学习和旧知识的更新。这就好比,只有踏出新的一步才能前进。那么如何才能更快的学会程序(游戏)开发呢?很简单,那就是“边学边做”!所谓知识来源于实践,做做学学,学学做做,这样你很容易就能融汇贯通了。所以,首先了解一些 VC 使用常识,照着书中的一些简单的例子一步一步的实际操作,从中学会一些基本的游戏开发常识。然后学习一些 C++ 理论知识,选一些难一点的例子来学。之后再学一些游戏开发的高级技术,试着自己开发一个游戏出来。罗马不是一天建成的,饭不是一口吃得完的,游戏也不是一会就能做出来的。所以,每天砌一块砖,不久一座美丽的城堡就矗立在你面前了。

衷心祝愿每位读者能在本丛书中吸收到有用的知识。

本书的写作意图源自于作者自己学习博弈程序设计的过程。作者开始试图编写第一个人机博弈程序的时候,几乎找不到合适的参考资料。在国内的几本人工智能书籍中找到的仅有最基本的搜索理论。而这离编写出实用的人机博弈程序相距甚远。无奈之下,我将寻求资料的眼光转向了国外,开始阅读国外的书籍和论文以及技术报告。在此期间,也有很多朋友表现了对机器博弈的高度兴趣。在交流当中,大家也都表示可资参考的中文资料实在太少了。陆汝钫先生编著的《人工智能》一书,有40多页论述博弈搜索的内容,大概是国内市场上买得到的对博弈搜索论述最多的著作,这是我最初学习的主要的食粮。但该书由于论述的中心是搜索理论,故而对于具体的实现和博弈程序的其他重要技术并未着墨。除此之外,我在这个领域连一本翻译过来的中文书都未找到。

在参考了一些国外的资料之后,根据自己的设计实践,同时也对比了国内其他爱好者的设计,我觉得有必要编写一本介绍博弈程序设计的一般方法的书。因为博弈技术经历了这么多年的发展,有很多好的算法,技术已经为国外的设计者所熟知。而国内甚至还有爱好者在使用基本的极大极小算法设计人机对弈程序。因此,限制程序性能的不是国人的能力,而是汲取新知的管道。匮乏的中文资料成了学习人机博弈程序设计的一道障碍。当然,作者不想也没有能力将此书写成顶尖的理论书籍。但却相信本书中所介绍的实用可靠的内容一定可以让国内学习和编写博弈程序的人们在入门的道路上稍微地加快步伐。

作为人工智能领域的重要实验室,博弈技术催生了一大批技术成果。也正因为如此,博弈技术所涉及的内容极为广泛。一个人是不可能面面俱到的,本书的目标在于能够让一般读者在掌握人机博弈的基本原理之余,也能够完全了解此类程序的具体实现方法。而且对如何提高计算机的博弈性能也能做到心中有数、运用自如。跟随本书一步一步地实践下去,在很短的时间里读者就可以设计出自己的博弈程序。由于博弈技术与其他研究领域交叉的边缘问题,大多处在探索阶段,而且这需要读者有其他领域的知识背景,本书就不一一介绍。

词汇

如前所述,由于本书在写作时参考的资料大多数是国外的,所以其中有少量生僻的名词,作者在国内出版物上也找不到合适的译文,硬译过来则恐怕词不达意,反而给读者造成理解上的障碍,因此直接使用了原文。不过在书后的术语表中,作者给出了自己的翻译。对英语十分过敏的读者可以查阅对照。

读者对象

本书面向任何对机器博弈有兴趣的读者。作者相信不论学习任何技术,实践都是最快的方法。也正因为如此,本书最突出的特点就是实践。大

部分的重要内容都有程序范例。范例程序是用 Visual C++ 写成的。虽然对 MFC 熟悉的读者可以很快速地吃透例子的内容,但 MFC 的知识并不是读者所必备的。本书范例的核心与 MFC 毫无关系,有一点程序设计经验的读者可以轻易地将它移植到其他语言平台上,Java 或者 Pascal 程序员也不必退避三舍,因为本书并未用到任何 C++ 独有的技巧。为了能够降低读者在阅读例子时理解上的困扰,作者对所有例子都刻意做了简化。精简代码可以让读者看到更少与主题无关或次要的内容。

本书大多数的算法以伪代码的形式给出。之所以如此,是因为作者本人,还有别人的学习经历都表明:此种描述对于一个程序员可能是最容易理解的。在人类的语言发生歧义,表达含混的时候,代码也许是程序员之间最佳的沟通工具。

感谢

首先我要感谢我的父母。没有他们的支持和协助,我根本无法有时间和精力来撰写本书。

本书的问世是同许多朋友和同事讨论的结果,这些朋友在我学习博弈程序的时候给了我很大的帮助。我要感谢郭翡,她对机器的博弈原理有准确的理解,在我开始接触博弈的时候使我摆脱了死板的文字,她三言两语就给我讲清了博弈树搜索的来龙去脉。接下来应当感谢许少军、李德华两位朋友,同他们的讨论使我获益匪浅。并且许君正在研发的五子棋对弈软件也让人高度期待。还要感谢从未谋面的 Pham Hong Nguyen。就是在他放在多伦多相棋协会网站的对弈程序,使我在 Java 调试器中第一次见到了 Zobrist 哈希、历史启发、迭代深化、静止期搜索等不可或缺的技术。让我在日后接触到论述相关问题的资料时都能够有清晰透彻的理解。这个绝佳范例同机器博弈领域先行者们的文献仿佛是两面相互照亮的镜子,隔着时间和空间的河流,驱散了一个求知者眼中的迷茫。

编 者

2002 年 7 月

目 录

第一章	引 言	1
1.1	人机博弈的要点	3
1.2	棋盘表示(Board Representations)	3
1.3	走法产生(Move Generation)	3
1.4	搜索技术(Search Techniques)	3
1.5	估值(Evaluation)	4
1.6	本书的布局	4
1.7	其他话题	4
1.8	关于范例程序	5
第二章	棋盘表示	7
2.1	基本表示方法	8
2.2	比特棋盘(Bit Boards)	9
第三章	走法产生	11
3.1	如何产生	12
3.2	效率问题	13
3.3	逐个产生 VS 全部产生	14
3.4	内存的使用	15
第四章	基本搜索技术	17
4.1	博弈树	18
4.2	极大极小值算法(Minimax Algorithm)	20
4.3	深度优先搜索(Depth First Search)	20
4.4	负极大值算法(Negamax Algorithm)	22
第五章	估值基础	25
5.1	棋子的价值评估	26
5.2	棋子的灵活性与棋盘控制	26
5.3	棋子关系的评估	27
5.4	与搜索算法配合	27
第六章	实例研究	29
6.1	数据表示	30
6.2	走法产生器	32
6.3	搜索引擎	52
6.4	估值核心	59
6.5	操作界面	84
6.6	试用	98
第七章	搜索算法的改进	101
7.1	Alpha-Beta 搜索	102
	范例程序	105
7.2	Fail-soft alpha-beta	110



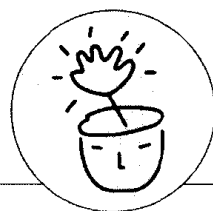
	范例程序.....	111
7.3	渴望搜索(Aspiration Search)	114
	范例程序.....	115
7.4	极小窗口搜索(Minimal Window Search/PVS)	117
	范例程序.....	119
7.5	置换表(Transposition Table)	122
7.6	哈希表(Hash Table)	123
	应用置换表的其他问题.....	125
7.7	Zobrist 哈希技术	127
	范例程序.....	128
7.8	迭代深化(Iterative Deepening)	138
	范例程序.....	139
7.9	历史启发(The History Heuristic)	142
	范例程序.....	144
7.10	杀手启发(Killer Heuristic)	152
7.11	SSS */DUAL * 算法	152
7.12	SSS * 与 Alpha-Beta	155
7.13	MTD(f)算法	156
	范例程序.....	159
7.14	综合运用.....	163
第八章	估值核心的优化.....	171
8.1	估值函数的速度	172
8.2	估值函数与博弈性能	173
8.3	估值函数的内容及其调试	174
第九章	其他重要的话题.....	179
9.1	水平效应(Horizon Effect)	180
	9.1.1 静止期搜索(Quiescence Search)	180
	9.1.2 扩展搜索(Search Extensions)	181
9.2	开局库(Opening Book)	182
9.3	残局库(Endgame Database)	183
9.4	循环探测(Repetition Detection)	184
9.5	代码的优化(Code Optimization)	185
	9.5.1 优化的阶段及地方	185
	9.5.2 函数调用的时间开销	186
	9.5.3 查表代替计算	186
	9.5.4 交叉递归	187
	9.5.5 内存管理的时间开销	189
9.6	其他方法	190
	9.6.1 轻视因子(Contempt Factor)	190
	9.6.2 机器学习(Machine Learning)	190
	9.6.3 并行搜索	191
	9.6.4 围棋	191

第十章 五子棋对弈的程序实例.....	193
10.1 概述.....	194
10.2 数据表示.....	194
10.3 走法产生器.....	196
10.4 搜索引擎.....	198
10.5 估值核心.....	217
10.6 操作界面.....	234
10.7 试用.....	247
附录 术语表.....	249
参考文献.....	253



第一章

引言





1997年,IBM公司的超级计算机“深蓝”与当时的国际象棋世界冠军卡斯帕罗夫进行了一场大肆渲染的比赛。这次被卡斯帕罗夫称作“终于来临的一天”的比赛以卡氏的失败而告终。IBM公司将“深蓝”的获胜称作是人工智能领域的一个里程碑。并且这场比赛取得的宣传效果好得出奇,大概没有多少人不知道“深蓝”了;而卡斯帕罗夫虽败尤荣,无数对国际象棋一无所知的人们因此知道了卡氏的大名。

实际上,远在1990年末,当诺曼·施瓦茨科普夫将军利用美军的超级计算机对“沙漠风暴”行动进行战略模拟的时候,机器博弈就已经作为高技术明星出现在那次有史以来最为成功的战争中了。这场战争博弈的成功当然比“深蓝”的成功更为巨大。多国部队以70余人伤亡的微小代价歼灭了伊拉克近10万部队,俘虏了6万余人。当地面战争在伊拉克境内展开的时候,土生土长的伊拉克部队表现得像一群外地人。在地面战争开始后几十个小时,伊拉克部队就与巴格达的领袖失去了联系。在战斗进行到第100h的时候,一切都结束了。布什宣布说,我们进行了一场“百时战争”。稍有军事常识的读者都不难看出,在军事行动当中,军队的部署,后勤物资的供应,乃至行动的时机等诸元的配合与协调远比棋盘上黑白方格上的问题要复杂得多。但两者在根本上,却存在着某种相似之处。

卡斯帕罗夫说“终于来临的一天”,意指人类棋手终于败给机器智能。实际上在这场著名的比赛之前,“深蓝”以及世界上许许多多的电脑就已能够击败绝大多数人类棋手了。海湾的战争博弈也许从一开始就不是一场公平的角逐。即使萨达姆·侯赛因的战争博弈水平同卡斯帕罗夫的棋艺一样高超,他也要面对美国军方无数超级计算机在战争的多个方面分别展开的大师级博弈。这或许是几十万部队在100h内溃不成军的一个重要原因吧。施瓦茨科普夫也向新闻界表明,此次军事行动的计划在提交总统之前就已在超级计算机上反复验过。

机器博弈在各种文献中被描述为人工智能的果蝇。就是说人类对机器博弈的研究衍生了大量的研究成果,这些成果对更广泛的领域产生了重要影响,与果蝇在遗传学中的研究所起的作用相似。所以IBM对“深蓝”成功的鼓吹也不算过分。毕竟,由博弈技术衍生而来的多种应用,如航空调度、天气预报、资源勘探、军事博弈,金融/经济调控等领域,也取得了大量引人瞩目的成果。

本书将介绍一些当今世界上成功的棋类博弈软件所广泛采用的主流技术。在介绍这些技术的同时,本书将以一个中国象棋的人机博弈程序实例贯穿其间,实际展示这些技术以期给读者更直接、更深刻的印象。

象棋是一种完全知识博弈(Games of Perfect Information),意思是指参与双方在任何时候都完全清楚每一个棋子是否存在,位于何处。只要看看棋盘,就一清二楚了。跳棋、围棋、象棋等都属于完全知识博弈。扑克游戏,还有麻将等大都不是完全知识博弈,因为你不清楚对方手中有什麼牌。

本书将要介绍的技术可或多或少地应用于完全知识博弈,尽管不同的游戏在细节上有很大差异,但在本书中所介绍的搜索算法上对特定的游戏知识依赖不多,不同于走法的生成,局面的评估,对具体的游戏规则有极大程度依赖。



1.1 人机博弈的要点

人机对弈的程序,至少应具备如下几个部分:

- ①某种在机器中表示棋局的方法,能够让程序知道博弈的状态。
- ②产生合法走法的规则,以使博弈公正地进行,并可判断人类对手是否乱走。
- ③从所有合法的走法中选择最佳的走法的技术。
- ④一种评估局面优劣的方法,用以同上面的技术配合做出智能的选择。
- ⑤一个界面,有了它,这个程序才能用。

本书将介绍上面列出的所有内容。对于界面,本书将示范一个简单的实例,但这方面并非本书的核心,并且读者也可以根据自己的需要做成更好的界面。

1.2 棋盘表示(Board Representations)

棋盘表示就是使用一种数据结构来描述棋盘及棋盘上的棋子,通常是使用一个二维数组。一个典型的中国象棋棋盘是使用 9×10 的二维数组表示。每一个元素代表棋盘上的一个交点。一个没有棋子的交点对应的元素是 0,一个黑帅对应的元素是 1,黑士则用 2 表示等等,依此类推。

棋盘的数据表示直接影响到程序的时间及空间复杂度。为了追求更高效率,研究人员针对不同棋类提出了多种不同的表示方法。

本书的第 2 章将探讨棋盘表示的方法及其细节。

1.3 走法产生(Move Generation)

博弈的规则决定了哪些走法是合法的。对有的游戏来说,这很简单,比如五子棋,任何空白的位置都是合法的落子点。但对于象棋来说,就有马走日,象走田等一系列复杂的规则。

走法产生是博弈程序中一个相当复杂而且耗费运算时间的方面。不过,通过良好的数据结构,可以显著地提高生成的速度。

本书的第 3 章将探讨走法产生的方法及其细节。

1.4 搜索技术(Search Techniques)

对于计算机来说,直接通过棋盘信息判别走法的好坏并不精确。除了输赢这样的局面可以可靠地判别外,其他的判断都只能做到大致估计。判别两种走法孰优孰劣的一个好方法就是察看棋局走下去的结果。也就是向下搜索若干步,然后比较发展下去的结果。为了避免差错,我们假定对手的思考也和我们一样,也就是,我们想到的内容,对手也想到了。这就是极大极小搜索算法的基本原则。极大极小搜索算法是本书中所有搜索算法的基础。

极大极小搜索算法的时间复杂度是 $O(b^n)$ 。这里 b 是分枝因子(branching factor),指棋局在各种情况下的合法走步的平均值; n 是搜索的最大深度,也就是向下搜索的博弈双方的走步



之和。例如向下搜索甲乙双方各走一步的情形, n 就是 2。显然对于象棋这种分枝因子在 40 左右的棋类游戏,时间开销随着 n 的增大会急剧的增长,不出几层就会超出计算机的处理能力,这将导致在有限时间内得不到令人满意的结果。

人们在开发高效的搜索算法上投入了大量的研究。在过去的几十年中,一些相当成功的改进大大提高了极大极小搜索的效率。Alpha-Beta 剪枝、迭代深化、置换表、历史启发等手段的综合应用将搜索效率提高了几个数量级。

本书将在第四章介绍博弈搜索的基本原理和方法,在第七章则讲述大量的搜索算法的优化及增强手段。

1.5 估值(Evaluation)

然而,现有的计算机的运算能力仍然十分有限。不可能一直搜索到分出输赢的那一步,在有限搜索深度的末端,我们用一些静态的方法,来估计局面的优劣。这些方法在很大程度上依赖于具体的游戏规则和我们对于该游戏的经验知识,其中相当一部分不完全可靠。例如,中国象棋的程序通常将一个炮赋予远高于一个兵的价值,但一个兵在高手的运用之下往往可以产生不次于炮的作用。

写出一个好的估值函数并不是一件轻松的事,它需要你对所评估的棋类相当了解,最好是一个经验丰富的高手。然后还要进行无数次的试验,经历几番失败后才可能得到一个令人满意的估值函数。

本书将在第二章介绍估值的基本方法,在第八章介绍一些进阶的估值方法及优化调试方法。

1.6 本书的布局

从本章 1.1 ~ 1.6 节的介绍可以看出,本书系由两大部分组成,即一至五章属基础知识部分。该部分内容涵盖了编写一个博弈程序所必备的最基本的原理性知识,这部分内容组合起来,可以构成包括象棋在内的各种棋类游戏的简单的博弈程序;七至九章属博弈技术的深化、优化部分,第十章为一个综合应用的五子棋实例。博弈有相当基础的读者可以略过基础部分,直接探寻深化、优化博弈技术的手段和方法。

1.7 其他话题

无论如何,编写一个出色的人机博弈程序不是一件简单的事情。除了上面介绍的一些要素之外,还有很多事情值得关注。

在人机博弈的原理背后,是解决具体问题的具体方法,一个博弈程序在实用化过程中,总会比理论研究要多碰到一些困难,在本书的第九章,包含了对一些博弈程序的缺陷以及补救手段的讨论。包括如下内容:

- ①水平效应的危害与预防手段。包括静止期搜索和扩展搜索。
- ②开局库。这可能是将人类经验赋予计算机的最简单的方法。



③残局库：利用计算机的庞大运算能力预先建立的残局数据库。

④循环探测：防止犯规和反复出现的愚蠢招数的方法。

⑤本书未提及的问题。限于作者的能力和见识,本书不可能涵盖所有的内容。但在本书的最后,列出了本书未能包括但在博弈领域又十分重要的内容,供读者进一步深入学习时参考。

1.8 关于范例程序

作为一本实用的指南性书籍,本书为读者提供了大量的相关程序范例。它们都被结合进一个中国象棋的人机博弈实例程序当中。包括数据表示、走法产生器、估值部件以及大约 10 种不同的搜索引擎。

范例程序的目的在于使读者尽可能具体地了解人机博弈的原理与方法。作为一个软件开发人员,范例程序将在很大程度上弥补文字表达上的不足。当然,作者最希望的还是读者能够自己动手撰写程序来实践本书所讲授的方法,这样一定会获益更多。

本书的范例是用 Visual C++ 6.0 编写的,主要是考虑到相比于 Java 和其他程序语言,C++ 的用户更多一些。为了实现一个简单的界面,在程序里还使用了一点 MFC 的内容,但仅仅用在界面部分。

范例程序被设计成了一个基于对话框的 Windows 应用程序。可以运行在 Windows98, Windows NT 及其后续平台上。

尽管作者在编写范例程序的时候希望其中仅包含刚刚所讲到的内容,但是在 Windows 程序里这几乎是不可能的。Visual C++ 的 Wizard 也会向其中加入大量的代码——读者自行编写时这倒是成了一个很好的优点。所以,这里建议读者仿照例子建立自己的程序而不是完全抄下来——让 Wizard 替你写大量代码将会是一个比较愉快的办法。

无论怎样,在机器上重建例子都会花费读者额外的时间。本书附带的光盘当中包含了本书全部的例子的源代码,读者只要将光盘上的例子拷贝到自己的电脑硬盘上就可一窥端倪。

如果读者碰到技术上或程序上的问题时,请直接与作者联系:hidebug@hotmail.com。