

游戏之旅——

我的编程感悟



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

游戏之旅——
我的编程感悟

云风 著

电子工业出版社·

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书忠实地记录了作者十余年来对游戏编程的所思、所感、所悟。全书按照作者本人学习和实践的过程,带着读者从基础的计算机知识到高级的编程技术,从非常专业的汇编优化到非常实际的项目管理进行了一次游戏开发的全景探索。

本书不仅适合游戏开发者阅读,也会给所有的开发者和程序爱好者带来启示。

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

游戏之旅——我的编程感悟 / 云风著. —北京: 电子工业出版社, 2005.12

ISBN 7-121-01609-5

I. 游… II. 云… 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 (2005) 第 086284 号

责任编辑: 朱沐红 zsh@phei.com.cn

印 刷: 北京天竺颖华印刷厂

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

经 销: 各地新华书店

开 本: 787×980 1/16 印张: 25.75 字数: 494 千字

印 次: 2005 年 12 月第 1 次印刷

印 数: 6000 册 定价: 46.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010) 68279077。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。

姚壮宪 序

PREFACE

云风，这名字在我 2000 年刚从台湾到北京之初就已经听闻了，当时觉得这名字真“武侠”，挺特别的。当时他已经是中国游戏程序界的名人了，那时我很想招募到这位高人，只可惜欠缺缘分未能如愿。现在他有更大的成就，我也很替他高兴。

在中国，我最佩服的网游公司是网易，不仅是因为其游戏运营得成功，更重要的是其赖以成功的几套网游完全是自制的。而云风是这些网游主力程序员之一，他也是我在中国最佩服的游戏开发者。看了云风的研发历程，我觉得就是一部中国的游戏程序史，从最早的 Z80，6502，PC8088，286，386... DOS，保护模式，Assembler 到 C++ 的整个发展轨迹。这本书可以说横跨了游戏程序的过去、现在和未来。书中传达的不仅是一些实用的技术经验，更是传达一种理念——虽然研发的环境随着时代而变，但研发的精神是不变的，那就是“在实践中积累”。

一些过去早期的技术手段，也许现在不一定派得上用场，但精神和原理是相通的。云风之所以有现在的成就，不是一朝一夕得来的，而是十余年的实践积累，从早期至今的每个阶段，面对各种当时的挑战，不断

地钻研。早期用一条条汇编指令跟运行速度斗争，64K，640K DOS 时代想尽办法跟内存空间斗争，和显卡斗争，到了网游时代，跟网速斗争，甚至跟外挂斗争。很辛苦，但是我觉得云风乐在其中，而且还能仔细地把这么长的过程中的这些细节记录下来并且写成书，实在非一般人可及，我就自叹不如。

像这样的书，在游戏界确实稀少，主要原因是，有空写书的往往不是在职的重要研发人员，而真正的在职高手工作负担都很重。像云风这批中国第一代的游戏程序高手目前几乎都在各企业担任重要工作，所以这类由本地从业人员所写，且内容实际在游戏中实践的书自然就很少。

云风花一年多写此书，可想而知是异常辛苦的，身体健康也付出不少代价。但是，我认为此书对整个中国游戏行业有承先启后的效果，可提供千千万万想成为未来游戏程序员的人很好的启蒙，对在职的工作者也能得到许多良好的经验。云风得到的只是一些稿费，而整个中国民族游戏产业得到的将是一次知识推动。

孟岩 序

PREFACE

我不是一个游戏程序员，对游戏开发可以说是一窍不通。因此，当云风邀请我给这本书写几句评论的时候，我是有些惶恐的——我能胜任吗？在读过这本书的初稿之后，我心里有了几分底气，因为我确信这本书不仅仅适合游戏开发者，同样也适合所有的开发者和程序爱好者。不同的人能从这本书里得到不同的启发，我愿意与大家分享我读这本书的几点体会。

这本书是我所看过的第一本以叙事散文体写作的技术作品，因此面目亲切成了我对它的第一印象。从作者记忆闸门里涓涓流淌而出的故事，本身就成为这本书的主线，也成为了最好的自序。本土技术图书的行文风格，历来以冰冷刻板为特征而自成一体，以至于我经常猜想，这种文体是否也是我们的以层层过滤为特点的精英选拔机制的一部分？

近年来，本版技术图书在“亲和性”方面已经做了很大的努力，很多作者，尤其是来自开发一线的作者，做了很多改进的努力，比如在文章里设定一个开发场景，讲一个故事，采用第一人称夹叙夹议，等等。

而云风的这本书，依我的有限的阅读来看，在可读性上达到了当前本土技术图书的一个新高度。行文亲切自然，态度谦和自信，记叙、议论、说明、抒情数体并用，转折无痕，恰到好处。尤其是在技术细节的表达上，非常细腻，非常深入，既无避重就轻、故弄玄虚之怯，又无居高临下、盛气凌人之骄。读着令人如此惬意的技术文字，我仿佛回到了三年前与云风品茗畅谈的那个晚上，看到了摇曳的烛光，闻到了茶气的清香，似乎他就坐在对面向我兴致勃勃而又慢条斯理地叙述着他的故事和心得。一本好书应该既是好老师又是好朋友，在这方面，我相信云风的这本书已经做了一个精彩的诠释。

当然，我并不是说技术书也应该按照文学的要求斧凿一番，本质上我们这些“理工科思维”的怪物还是会依据一本书的技术水准来给予它基本的评价。跟机器打交道的程序员自然有冷血的一面，哪怕你锣鼓喧天，如果没有真功夫，也休想赢得半声喝彩。而以“真功夫”论，本书在编程技术和游戏开发专业技术两方面都表现得极为出色。全书基本上按照作者本人学习和实践的过程，带着读着从基础的计算机知识到高级的编程技术，从非常专业的汇编优化到非常实际的项目管理进行了一次游戏开发的全景探索。

在技术上，本书既相当深入又很好地把握了重点，显然在选材上花了很多心思。同时作者非常清楚书本的局限性，很多功夫下在了书外，为读者提供了很多资源和学习方法的指导。从体裁上看，本书不是教材，但是由于作者将自己的学习经历忠实地记录了下来，所以读者如果能够在阅读本书的同时，根据自己的实际情况多动手、多思考、多实践，则完全能够走出自己的“高手”之路。因此，对于有心人来说，这本书也可以是最好的教材。更重要的是，全书完完全全来自实践，又经过了多年的沉淀，与时代同步，与读者零距离，没有食洋不化的弊端，也没有隔靴搔痒的尴尬，在目前国内游戏编程类图书中显得卓尔不群。

云风是国内著名的游戏程序员，也是网易公司著名网游《大话西游》的主力开发者，他的技术是高超的，他的事业也是非常成功的。当然，他的传奇经历不用我多说，这本书本身就是他最好的自传。从编程技术上讲，他是毫无疑问应该被归为“天才”一类的人，过人的天资加上惊人的勤奋，使他很早就成为国内游戏开发技术的佼佼者。我对于游戏开

发技术并不了解，但是在 C/C++ 开发上与他有过一些交流，深为他扎实的基本功、对系统的透彻理解、对程序的天然敏感和刻苦的钻研精神所折服。

特别让我感到钦佩的是，作为一个别人眼中的编程天才，他似乎很清楚地知道如何权衡地使用技术力量，在技术与实践中把握平衡。比如在 C++ 一些具体技术的使用问题上，我走过很多弯路，总是在几个极端上不断地自我否定，只是在数次挫折之后才逐渐有所体会。而云风不仅有着“先天”的敏锐嗅觉，而且能够比我快得多地达到“知”的彼岸。他既勇于探索又能够保持头脑清醒，既能够创新又能客观地看待自己的创新，从而在总体上始终能够向着最有利于实践的方向前进。这一点是非常难能可贵的，甚至对大多数人来说，必须经过长时间的实践才能在一次次的成功与挫折中悟道。

C++ 名人 Alexandrescu 说，十几岁的少年天才到处都有，三十多岁的优秀设计师凤毛麟角，掌握一种力量是容易的，学会恰当地使用这种力量却难得多，这就是聪明与智慧之间的差别。文武之道，有张有弛，知道什么是该做的，什么是不该做的，这是人生最大的学问。之前我经常奇怪，云风还非常年轻，他程序思想中的那种老练的智慧是从何处得来的呢？读完这本书之后，我终于明白，还是那句话：“无他，唯手熟耳”。可以说，云风的成功之路，就是一条实践之路。路上的每一步，都是他一点一点走过来的，任何一个读过本书的读者，恐怕都会对云风丰富的经历印象深刻。他付出比常人多得多的努力去实践，自然收获比常人多得多的智慧和经验，对此，我所能有的只是钦佩。

因此在我看来，这本书对我以及很多人来说更重要的启示在于，它揭示了一个程序员的成功之道——实践。由于工作的原因，我接触到很多的优秀开发者，他们在技术上各有特长，但在个人事业上却状态迥异。作为一个近距离的旁观者，固然常为成功者的意气风发所感染，却也时不时为众多高手的作茧自缚而一声叹息。是什么导致这样的差别？我觉得就是对于成功之“道”的体认，或者按照我的总结，对于“真正的实践”的体认。

这种实践不是传统文人式的闭门苦读，不是精神贵族式的居高临下的点点画画，也不是市侩式的唯利是图的混迹，而是有理想、有方向、有规划、有选择、有实干的主动的思考和行动。云风在这本书里用他自己的经历为这种实践作了出色的范例，他在自己成长的

每一个阶段都充分利用了客观环境所能提供的条件，既深入地学习，积极地实践，同时也毫不犹豫地参与社会工作，还能总结经验和教训并迈出新的步伐。这才能使他做到既深入技术细节又能把握工作全貌，既实现了技术梦想又实现了现实的事业。读者如果善于领悟，应该能够见微知著，体会作者的成功之“道”。

老子说，具体的知识越学习就积累越多，而“道”则是越追求便越简单，大道至简，至简的大道才能应变万端（为学日益，为道日损，损之又损，以至于无为，无为无不为）。知识和技术是学不完的，只有得到实践这个“道”，才能实现“无不为”的理想与抱负。这就是我从云风这本书里得到的最大的收获，愿与各位共勉。

写书是件非常辛苦的事情，尤其对云风来说，更是件非常得不偿失的事，他满可以用这宝贵的时间和精力为自己创造更多的“价值”，然而他选择了写书，把更大的价值奉献给读者。面对这沉甸甸的作品，我确实感到，这是云风用心写的书。用心写的书，当然出色。

前言

PREFACE

终于把整本书完成了，我几乎不敢相信能完成。陆陆续续地写了一年多，其中的辛苦恐怕只有写过技术书籍的朋友能够理解了。

很早以前就有朋友跟我提过，好好写本书，把自己理解的知识传授给他人，于己于人都是颇为有益的事情。写作技术书是很困难的事，它需要作者厚积薄发，把自己多年的知识浓缩归纳，才能用简洁的文字表达出来。而知识的传授绝对不同于讲故事，文字究竟能表达出多少作者自己已经理解的，我个人觉得是很难拿得准的。

先后有几个出版社的朋友来约稿，只是一直忙于项目开发，没有将其放在心上。直到《大话西游 II》和《梦幻西游》两款游戏相继开发完毕，并取得了市场上的成功，我的工作也相对闲了下来。这时，我终于可以动笔来写这本书了。

大家现在看到的这些内容，是我在一年多的时间里，断断续续整理出来的。看惯了国外严谨的技术书的读者，肯定会挑出许多毛病来。我几乎没有一章，把某个技术点由浅入深地阐述全面；也没有将游戏制作

的某个特定领域分章断节，细细道来；它绝对是依照我自己的偏好来选材，甚至，这本书不能算是一本游戏编程的专门书，我把重点放在了编程领域，而不是游戏。

我曾经对自己这本书的定位迷茫过。它看起来好像是一个大杂烩，整体的组织没有条理，细节的描述又不够彻底。终于有一天，我自己想明白了，得以心情舒畅地将之全部完稿。

首先，这里记录的诸多技术问题都曾经启发过我自己，那么它们一定也能启发他人，必然会和一些心有戚戚的读者引起共鸣。而另一方面，我不敢随意对游戏开发，甚至软件开发中哪些方面更重要一些下出论断。虽然我自己有自己的理解，但表现于具体的文章上，只好将自己觉得会非常重要的部分多写一些（因为自己觉得这些重要，所以研究得也多一些），希望读者读到那些枝节要点却没有点明的时候，能够理解我的用意。

其次，所有的基础知识都有前人留下的教导，和读者分享自己是怎样学到的足矣。授人以鱼，三餐之需；授人以渔，终生之用。大多数问题，如果把别的书中的章节生搬硬套过来，则是对我自己的精力和读者的时间的极大浪费。不以自己懂了什么就写出来给读者炫耀一番，是我后期删改书稿的重要原则。

终其整本书，虽然参考之书籍、网站、代码众多，但我自己都一直坚持一个原则：建立在充分理解的基础上，再用自己的语言表达出来。我相信只有这样，读者才能够看到一个从独立思考角度获得的知识体系，凌乱中不失一缕联系。

这样也带来许多遗憾。因为还有一些内容我在原计划里是想写的，它们对游戏软件的编写也是至关重要的，比如设计模式。可惜我在动手的时候，发现自己这些方面的修养在项目中用用还可以，离著书立说却差得很远，只好放弃。

曾经年少轻狂。但随着时间推移，越发发现自己所知所见，只是沧海一粟。学得越多，越是不敢轻言所知。所以，希望读者相信，我是非常谨慎地完成自己的第一本书的。孔子说过，无友不如己者。我真诚地尊重所有对本书提出批评的朋友。可以去我的个人主页（<http://www.codingnow.com>）留言发表你的看法，或给我发 E-mail: cloudwu@gmail.com，

我个人更倾向电子邮件这种交流方式。

希望听到大家的声音。

写书这件事情终归是告一段落了，我也将把精力投入到自己新的事业中去，或者很久都不可能再以这种形式将自己的技术人生做一个总结。身处当今热门行业中，可谓衣食无忧，而在国内的游戏制作圈子里混迹多年，业内也小有名气。而这些并非我去追求的，它们自然而生。这本书也一样，我没有刻意去追求表达些什么，也从没有想过本书会给我带来些什么。我现在是中国并不成熟的游戏制作行业中的一员，游戏给了我太多，我告诉自己需要做一点事情。分享知识和经验是我的义务，别无它。

云风

2005 年 5 月 23 日晚于广州

目 录

第 1 章 计算机, 游戏, 我———— 1

我, 云风, 从二十年前的计算机游戏萌芽阶段一路走来。计算机, 并不神秘。程序, 也只是表达计算机控制逻辑的符号而已。

1.1 计算机	2
1.2 计算机游戏	3
1.3 计算机与我	7
1.3.1 启蒙	7
1.3.2 编程	9

第 2 章 算法, 程序的灵魂———— 13

算法, 先于计算机存在于世, 比编程语言本身更为重要。语言只是工具, 算法才是灵魂。这里简单介绍了数据结构的基本知识、算法效率的评估方法、几种 NP 问题的定义。

这里对游戏中最常用的搜索算法问题之一寻路问题做了些许探讨, 并展开介绍了在以后可能被广泛使用的智能算法。最后用几个实例探讨了优化问题。

2.1 程序=算法+数据结构	14
2.1.1 算法	15
2.1.2 数据结构	17
2.2 搜索算法	23
2.2.1 地图寻路问题	23
2.2.2 博弈问题	27
2.2.3 更为广泛的运用	28
2.3 智能算法	29
2.3.1 遗传算法 (Genetic Algorithm)	29
2.3.2 模拟退火算法 (Simulated Annealing)	31
2.3.3 禁忌搜索 (Tabu Search)	33
2.3.4 人工神经网络 (Artificial Neural Network)	34
2.4 优化	36
2.4.1 质数问题	36
2.4.2 俄罗斯方块竞赛	37
2.5 Apple II 上的编程之路	39

至少要学会一种编程语言,才可能编写游戏软件。流传于世的编程语言数以百计,但真正流行的却不多。各种编程语言之间有它们的共通性,但也并非能够一通百通,甚至也不存在一种适合解决所有问题的语言。

我认为大部分游戏程序员都应该掌握工作平台的汇编语言。这对理解硬件,写出更好的代码,还有软件调试都有莫大的帮助。

3.1 C 语言	46
3.2 BASIC	50
3.3 C++	51
3.4 汇编语言	54
3.4.1 概述	55
3.4.2 程序的本质	57
3.4.3 寄存器	58
3.4.4 寻址方式	60
3.4.5 汇编指令	61
3.4.6 C/C++ 语言和汇编	62
3.4.7 小结	63
3.5 其他语言	63
3.5.1 Forth	63
3.5.2 Lisp	64
3.5.3 Java	64
3.5.4 Python、Lua、更多	65

现在,Windows 一统江山,它帮我们隐藏了硬件的许多东西。这一章是对10年前知识的一个回顾,它们如今已经沉寂在历史中,或是被Windows 掩盖。但是,作为喜爱刨根问底的程序员,我们有必要对Windows 之前的时代有一些了解。

4.1 386 保护模式	68
4.2 VGA 到 VESA	70
4.2.1 超越 BGI	70
4.2.2 VGA	72
4.2.3 VESA 标准	72
4.2.4 花絮	74
4.3 保护模式下的开发工具	75
4.4 闲话 Allegro	81
4.4.1 用C与汇编写成的程序库	81
4.4.2 BITMAP	82
4.4.3 Sprite	85
4.4.4 几何图形和3D	89
4.4.5 数据文件	91
4.4.6 声音	92
4.4.7 其他的部分	93

4.4.8 小结	94
4.5 cfido 中国惠多网	94

第5章 Windows 编程 101

如果我们在 Windows 下开发游戏,那么理解 Windows 系统就是尤为重要的事情。Windows 只是一个操作系统的特例,不属于计算机理论知识的范畴。这导致了许多已经掌握了编程知识,理论功底也很扎实的程序员,面对 Windows 下的开发无可适从。

5.1 Windows 编程入门	104
5.1.1 Windows 版本综述	105
5.1.2 操作系统的核心	107
5.1.3 Windows API 和 DLL	110
5.1.4 COM	111
5.1.5 Windows 的窗口和消息处理与传递	114
5.1.6 Windows GDI	125
5.2 控制游戏的速度	130
5.3 浅谈 MFC	132
5.4 小结	132

第6章 汇编优化 135

汇编优化,是在高性能游戏引擎设计中很难避免的一个环节。虽然不用汇编去做优化依然可以写出高性能的程序,但优化却可以变得更好。

但是,这一章并非从汇编开始谈起。相反,一开始我就花了不少笔墨谈汇编之外的知识。即使是一些技巧性的细节,也并非必须用汇编才可以做到。

这一章着重介绍了在 C/C++ 中和汇编进行混合编程的方法和技巧,然后才具体谈到汇编指令的高效使用策略。

6.1 浅谈代码优化	138
6.2 并不仅仅是汇编	139
6.2.1 在算法实现时减少上下文的依赖关系	139
6.2.2 低效的静态变量	140
6.2.3 数据的组织	141
6.2.4 消除除法	142
6.2.5 避免过大的循环	144
6.3 汇编和 C/C++ 的混合编程	146
6.3.1 使用外置的汇编编译器	146
6.3.2 使用内嵌汇编	148
6.3.3 C++ 的函数调用规则	150
6.3.4 寄存器的使用	152
6.3.5 _declspec(naked)	152
6.4 指令使用的技巧	153
6.4.1 LEA: 别样的内存寻址指令	153
6.4.2 BSWAP: 扩展寄存器的个数	154

6.4.3	缩短指令的字节数	155
6.4.4	避免跳转指令	159
6.5	Pentium 的汇编优化策略	161
6.5.1	UV 流水线	163
6.5.2	AGI 地址生成互锁 (Address Generation Interlock)	167
6.5.3	Pentium Pro, Pentium II 和 Pentium III 的汇编优化策略	168
6.5.4	Pentium 4 的汇编优化策略	178
6.5.5	如何测试代码的速度	181
6.5.6	汇编能干的事情	183

第 7 章 SIMD 指令 185

单指令多数据(SIMD, Single-Instruction-Multiple-Data)编程是现代 CPU 高速处理多媒体数据的发展方向。好的工具摆在一边不去用就是一种浪费。

这里介绍得多的是 MMX 使用技巧,这方面也是我用得比较多的,勉强算得上经验丰富。

7.1	单指令多数据编程	186
7.1.1	Alpha 混合	187
7.2	MMX	191
7.2.1	MMX 的指令概况	192
7.2.2	SSE 及其对 MMX 的扩展	194
7.2.3	SIMD 技术在 C 语言环境下的使用	196
7.2.4	MMX 编码技巧	197
7.2.5	SIMD 的应用举例	201
7.2.6	SIMD 编程注意要点	203
7.2.7	SIMD 的性能检测	204

第 8 章 引擎中的优化 205

单从内容来看,这一章是本书的独家内容。我得以成名的开源游戏引擎“风魂”的几乎所有优化策略都在这章公开出来。虽然引擎源码一直都是公开的,想研究的人自然可以通过阅读代码来学习,但是,由作者本人来解释其中的优化策略,这是第一次的文字记录。

8.1	Blit 的优化	207
8.2	脏矩形优化	209
8.2.1	“风魂”中改进的脏矩形算法	215
8.2.2	滚动优化	219
8.2.3	覆盖优化	222
8.2.4	脏矩形在 3D 游戏界面中的运用	224
8.2.5	时间和空间的权衡	229
8.2.6	关于估算	230

8.3 自己设计精灵格式	231
8.4 动态数据加载	234
8.5 小结	236

第9章 C 和 C++

237

C++ 是我这几年的最爱。如果程序员没有陷入工程项目的泥沼,单纯做理论学习研究的话,我相信很多人都会爱上博大精深的 C++ 语言。

C++ 被程序员圈子最大的误解就是效率低过 C 语言。这里,我想为 C++ 再次平反,已经有好多人干过这事,我多做一次也不为过。我想证明的是,至少在 PC 平台上,好的 C++ 程序不仅不会比同样的 C 语言实现慢,甚至会略快一些。

当然,我选择 C++ 不仅仅因为它的高效,还在于 C++ 语言的优雅。本章重点介绍两个枝节问题,一是内存的管理,二是 template 的使用,这都是广泛使用 C++ 后一定会遇到的问题。

9.1 从 C 到 C++	239
9.2 C vs C++, 效率至上	243
9.2.1 全局变量	243
9.2.2 函数调用的堆栈处理	243
9.2.3 函数调用的参数传递	244
9.2.4 临时变量的堆栈使用	244
9.2.5 代码生成	244
9.2.6 异常	244
9.2.7 inline	245
9.2.8 多态性	245
9.2.9 标准库的差异	245
9.2.10 动态内存分配	245
9.3 优雅的 C++	246
9.3.1 消失了的宏	247
9.3.2 const 修饰及类型转换	248
9.3.3 隐藏实现	248
9.3.4 引用而非指针	250
9.3.5 名字空间	250
9.4 C++ 的误区	251
9.4.1 类层次过细	251
9.4.2 滥用操作符重载	252
9.4.3 滥用标准容器	253
9.4.4 滥用动态类型识别和异常	253
9.4.5 滥用多重继承	254
9.4.6 忽视 C++ 高级特性的复杂度	255
9.4.7 学习 C++	256
9.5 动态内存分配的优化	257
9.5.1 另一种优化策略: 多次分配, 一次释放	266
9.5.2 垃圾回收	268