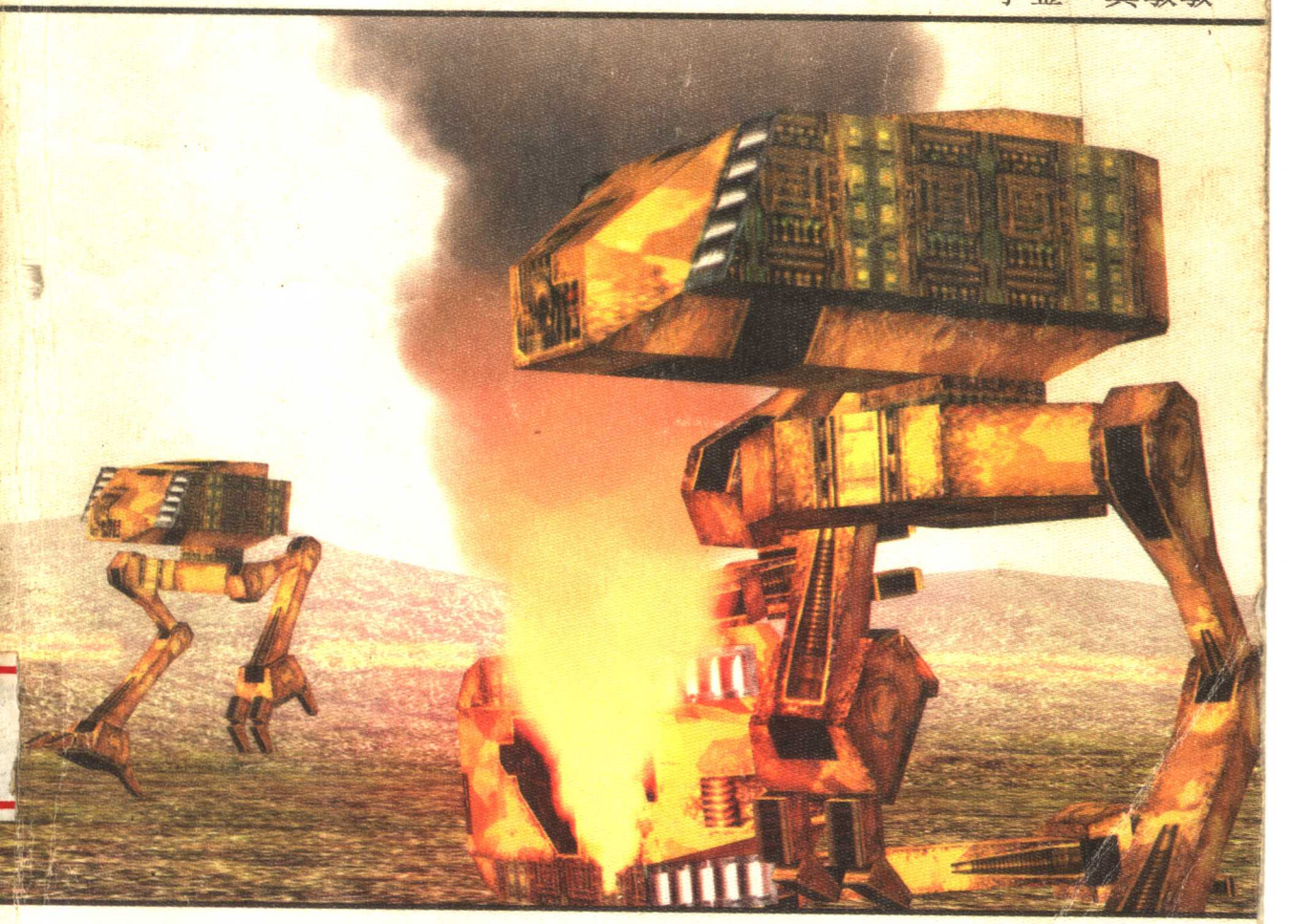
电脑报 东方工作室

快乐写游戏
轻松学编程

PC 游戏编程（基础篇）

PC Youxi Biancheng Jichupian

编著 徐丹 谭文洪
李鉴 龚敏敏



重庆大学出版社

PC 游戏编程(基础篇)

徐 丹 谭文洪 李 鉴 龚敏敏 编著

重庆大学出版社

内容提要

本书全面介绍了游戏编程的初级入门知识，包括数据结构、算法、文件打包以及C ++。并详细介绍了一些在游戏策划中所需要的知识。另外，作为本书的一个侧重点，书中用较大篇幅介绍了计算机图形学与Direct3D，这些章节为读者进入游戏三维编程殿堂提供了入门知识，也能够给一些有经验的三维编程人员提供一些宝贵的经验。同时介绍了一些在实时三维游戏美术中需要注意的地方（涉及到面简化等技术）。最后一章，大部分内容是为了巩固前几章所学的理论知识所编写的例子程序，每个例子程序都能够说明一定的问题，并且每一个例子都比上一个例子涉及的知识更深入，读者可以按照顺序渐进学习。

图书在版编目(CIP)数据

PC游戏编程(基础篇)/徐丹等编著. —重庆：重庆大学出版社，2003.1
(快乐写游戏轻松学编程系列)
ISBN 7-5624-2776-3

Ⅰ.P... Ⅱ.徐... 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097755号

快乐写游戏 轻松学编程

PC游戏编程(基础篇)

徐 丹 谭文洪 李 鉴 龚敏敏 编著
责任编辑：陈 其 版式设计：吴庆渝
责任校对：何建云 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
万州日报印刷厂印刷

*

开本：787×1092 1/16 印张：19 字数：474千
2003年1月第1版 2003年1月第1次印刷
印数：1—5 000
ISBN 7-5624-2776-3/TP·404 定价：36.50元(赠1CD)

序

编写程序就像是在盖房子——看着“房子”一天一天盖起来，作为程序员的我们的心里，那份喜悦是不能用言语表达的。尤其是编写游戏程序，当你看到自己所写的游戏带给大家快乐的时候，你的心里更是快乐无比。但是，这份快乐是有代价的，是与无数的不眠之夜分不开的。所以，每一个想成为游戏程序员的你们，在做这件事之前，一定要考虑清楚，你真的愿意把你的时间，你的青春奉献在这里吗？你愿意当别人在酒吧潇洒的时候，你却独自坐在计算机的面前，面对着一行行的程序吗？你愿意长时间地忍受失败，仅仅为了那一次成功吗？

如果你愿意，那么好，现在就开始，跟随这本书开始一段旅程，一段追逐梦想的旅程，你很快会发现，其实写游戏的程序很简单，只要你学会一定的规则，按照一定的规律，一个演示版很快就会出现在你的面前。这也是本书的目的，这本书只给了你一把钥匙，一把进入游戏制作大门的钥匙。而只有你进了这个大门之后，你才能看到一片广阔的天地，等待你去驰骋。

游戏的程序就是一行行的代码，需要你的灵魂才能赋予它以生命，有了生命的游戏才能带给大家快乐，也才能给你所想要的。你的游戏是你的热情，是你的汗水，是你的喜怒哀乐，是你的梦想，你的游戏其实就是你自己。

剑　刚

于目标软件

2002 年 11 月 25 日

前言

小时候，我总喜欢沉迷在虚拟的世界里：安徒生童话、海底两万里……但那些令人着迷的人物和故事似乎离我们太过遥远。直到游戏的出现，那个虚拟的世界变得不再遥远，你也许就是故事的主人，按照剧情的发展和你的意愿去做你想要做的每一件事情。而我立志做游戏开发者的最大的原动力就是能够成为这个世界规则的缔造者。

有人会问，做游戏程序员的乐趣究竟在哪里，我要告诉你的是：在每一次突破性的成功后的巨大喜悦：在接到9个投诉电话之后，突然接到1个称赞电话时的飘飘然；在学习和"窃取"开发资源并应用自如后的欣喜若狂……这些不仅仅是游戏程序员的乐趣，重要的是它们是你在困难中前进的动力。

本书和这个系列的其他书一样，重点讲述了一些游戏编程的技术以及技巧，但它和前作不同的是，它的侧重点这次放在了三维游戏程序上。它不仅适合想进入游戏领域的编程爱好者，同样也适用于有一定经验的游戏开发者。除了介绍了一些在游戏中常用的编程技术以外，本书的大部分篇幅都在介绍三维程序的开发，同时介绍了大量的游戏开发的经验，这些都会对你的开发有一定的帮助。

本书并没有完整的介绍一个三维游戏的开发过程——因为一个完整的三维游戏的工程量不是几天几个月就能够做完的；本书也没有基于某一个特定的游戏开发库，这样做是为了给读者一定的自由度。当然，本书的图形程序例子是基于DirectX编写的，如果你认为这还是太"高级"，或是你根本对DirectX没有兴趣。那么，本书的第七章着重讲述的图形学原理有可能能够吸引你的注意，对，它完全没有涉及特定的API。

阅读本书需要的预备知识

本书是假定你有一定的C++语言的编程基础知识，以及一些基本的数据结构基础知识。如果不是这样，建议你去阅读一些基础书籍，目前专述C++与数据结构的好书为数不少，必有一些能够适合你。

如何阅读本书

本书的基本结构是渐进式的，也是分章独立式的，如果你对其中某个章节非常熟悉，完全可以跳过阅读。并且，章节后的习题对你巩固知识也是非常重要的，千万不要忽略它们。

深切感谢重庆大学出版社陈其先生和拓智文化陈治刚先生以及其他对本书编写鼎立支持的朋友们。

感谢和我一起编写本书的李鉴、龚敏敏、谭文洪，他们对工作的认真以及对技术的精通，让本书的内容更有价值。

最后，让我们一起来感谢一直支持我们的家人和朋友，他们是我们坚持的动力！

徐　丹

2002年12月于北京

目录

第一章 游戏编程基础

1.1 PC 游戏文化概述

PC 游戏已经有十几年的发展历史了,而游戏真正成为产业却只是在这几年。这些年来,新的软、硬件技术不断地推动这个行业的发展,并使之越来越蓬勃。而且,随着网络的出现,又为这个产业注入了新鲜的血液。

回顾人类文化的发展历史,任何一种新的娱乐方式的兴起,都不可避免地包含着想像力和技术这两种因素。

幻想是人类永恒的需要,它从不放过每一个展开翅膀的机会。一种为实用而创造出来的工具往往会被人们转变为一种想像的工具,人类的想像力面对种种不同的技术手段,似乎始终保持着惊人的适应和创造潜能。顽石化为庄严凝重的雕像,这是一种想像力创造的奇迹;那么当由"0"与"1"构成的枯燥无味的计算机语言转化为屏幕上变幻无穷的、曲折往返的电子游戏时,这不也是一种奇迹吗?不妨说,电子游戏是用最逻辑的东西创造感性,是用最抽象的东西创造直观。而实现这对矛盾的转换的正是人类的想像力。

然而,在赞叹人类想像力的同时,应当注意技术的因素在人类文化发展中的位置。正是技术手段更新带来的信息载体的更新,为新的幻想方式的展开,为新的艺术门类与娱乐形式的兴起提供了可能。而这一点往往是最容易被研究者所忽视的。

比如说史诗,它为什么会繁盛在早期人类社会,恰恰也是基于特定的技术因素。史诗繁盛的时代是一个口语的时代,在那个时代,文字虽然已经出现,但由于书写工具的限制,它不可能大量的记录,传播人类文献、文明依赖的是口耳相传,依赖的是人的记忆。所以口语的时代是一个韵文的时代,有韵的文字可以歌唱或吟诵,这对于人的记忆来说要容易一些。

而长篇小说的兴起无疑同印刷术的突破性进展有关。印刷术从技术上解决了限制文献大规模复制、传播的难题,为长篇大论的写作创造了条件。书写取代了吟诵,散文(广义的散文,指所有无韵的文字)取代了韵文。小说,这种更为自由随意的文学形式在文学创作中成为了最重要的门类。

今天的游戏,已经从之前的单纯用于娱乐的程序,发展成为一种"互动艺术"的表现形式。游戏本身并不预示着任何一种文化,它可以表达东方的东西,也可以表达西方的东西,换句话说游戏已经和音乐、建筑一样具备了作为文化传输媒介的一切特征。对文化传播媒介比较敏感的人或许会有这样的体会:音乐那种从听觉上令人心旷神怡,思想随之飞扬的感觉,就和欣赏建筑那种从线条上带来的美感不一样,王小波先生将这种感觉称之为韵律。我借用一下这个比喻,把话说下去——不同的媒介之间的区别,在于其韵律的不一。音乐的韵律是贴近听觉的韵律,绘画的韵律是贴近视觉的韵律;而游戏也拥有它自己的韵律,且它的韵律最贴近这个经典表达形式与科技相互冲击、交融的时代。就根据韵律,贯注以心灵的声音,人们创作出各种不同的游戏节奏,再让不同的节奏相互交错成曲,成为一个个优秀的作品,例如《轩辕剑》系列(图 1.1)。不难发现,脱离了游戏韵律,游戏节奏以及游戏中的人文材料便会顿失它们原本在游戏里面的意义,甚至沦为乏味的聒噪。回过头来看看许多人眼中的游戏文化,大抵是局部提取游戏题材,异化游戏韵律,刻意要它与另一种媒介的韵律相结合,不过这种异化以后的东西将不再属于游戏。

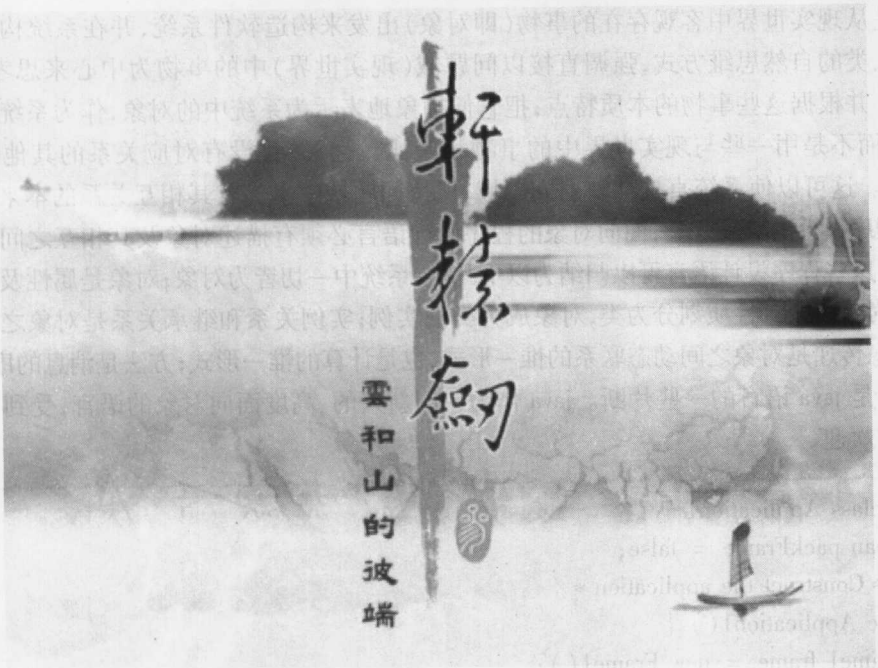


图 1.1　游戏已经成为了一种文化

1.2　程序设计基础

无疑，程序设计是一个软件的灵魂。而且随着计算机程序设计语言的越来越智能化，程序设计也变得越来越容易，也有越来越多的人在尝试用一行一行的代码来实现自己的梦想。同样，也有越来越多的人准备投身到这个行业中来。

一直以来，有一个这样的观念，即“软件 = 数据结构 + 算法 + 代码”，虽然随着技术的进步以及不断扩大的软件规模，这样的观念受到了很大的冲击。但不可否认的是，目前大多数软件的设计还是没有突破这样一个公式。

1.2.1　程序设计语言

程序设计语言是随着计算机的诞生而诞生的。最初的程序就是打孔机在纸上打的一个一个的小孔，它们代表了计算机能够识别的二进制的代码。为了抛弃这种不直观而且容易出错的方法，人们逐渐的使用一些符号来代替难于记忆的二进制编码，并随之产生了程序设计语言。

早期的开发方法主要是面向过程，或者说面向处理的。这种对过程的强调是由于当时使用的程序设计语言主要是面向过程的语言，主要应用是批量的和基于文件的应用。开始于 20 世纪 70 年代的结构化方法是最有代表性的方法。Yourdon 和 Constantine 提出的一种符合软件工程模块化原则的结构化设计方法，至今仍在使用。

面向对象是一种新兴的程序设计方法，或者说它是一种新的程序设计模型，其基本思想是使用对象、类、继承、封装、消息等基本概念来进行程序设计。

它是从现实世界中客观存在的事物(即对象)出发来构造软件系统,并在系统构造中尽可能运用人类的自然思维方式,强调直接以问题域(现实世界)中的事物为中心来思考问题、认识问题。并根据这些事物的本质特点,把它们抽象地表示为系统中的对象,作为系统的基本构成单位(而不是用一些与现实世界中的事物相关比较远,并且没有对应关系的其他概念来构造系统)。这可以使系统直接地映射问题域,保持问题域中事物及其相互关系的本来面貌。

从程序设计的角度来看,面向对象的程序设计语言必须有描述对象及其相互之间关系的语言成分。这些程序设计语言可以归纳为以下几类:系统中一切皆为对象;对象是属性及其操作的封装体;对象可按其性质划分为类,对象成为类的实例;实例关系和继承关系是对象之间的静态关系;消息传递是对象之间动态联系的惟一形式,也是计算的惟一形式;方法是消息的序列。

下面是 java 语言的一些片断。java 作为一门新兴的、高度面向对象的语言,受到了越来越多用户的欢迎。

```
public class Application1 {
  boolean packFrame = false;
  /** Construct the application */
  public Application1() {
    Frame1 frame = new Frame1();
    //Validate frames that have preset sizes
    //Pack frames that have useful preferred size info, e.g. from their layout
    if (packFrame) {
      frame.pack();
    }
    else {
      frame.validate();
    }
    //Center the window
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
    Dimension frameSize = frame.getSize();
    if (frameSize.height > screenSize.height) {
      frameSize.height = screenSize.height;
    }
    if (frameSize.width > screenSize.width) {
      frameSize.width = screenSize.width;
    }
    frame.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height -
frameSize.height) / 2);
    frame.setVisible(true);
  }
  /** Main method */
  public static void main(String[] args) {}
}
```

就目前来说,比较流行的语言有 C/C ++ , Java, Object Pascal 和 Visual Basic 等。它们中的大部分都是面向对象的语言。非面向对象的语言目前主要应用在 UNIX 环境中,在其他的领域已经应用得不多了。

1.2.2 集成开发工具

软件开发发展到今日,已经不完全是单纯的源代码的积累了,为了更好地进行程序设计,人们开发了一系列的辅助工具。而集成开发工具正是把这些工具集合起来,成为一个单独的软件包。一般的集成开发工具都集合了语言编译器、代码编辑器、程序调试器等等有利于程序设计的工具,如图 1.2。

图 1.2 Microsoft . net 集成开发环境

随着集成开发工具的出现,程序设计的效率大大地提高了。以下列出了流行语言的集成开发工具。

C/C ++ : Microsoft Visual C ++ , Borland C ++ Builder。
Pascal: Borland Delphi。
Basic: Microsoft Visual Basic。
Java: Borland Jbuilder, Visual cafe。

以上列举的工具都是同类语言中开发工具的佼佼者,具有一定的普遍性。

1.2.3 数据结构与算法

毫不夸张地说,数据结构的选择很大程度上决定了程序的复杂度。对于初学者来说,不应该被大学教材上那些看似高深的知识而吓倒。很多人认为只有掌握了树、图这样复杂的数据结构才算真正掌握了数据机构。其实不然,实际上在常规的程序设计中(不涉及到复杂的底层),树和图应用得很少。而且,深入地了解基础知识对理解树和图也是大有帮助的。

一般来说,复杂的数据结构都能被相对简单的数据结构所模拟,这样会带来一些灵活度,但也会导致控制难度的增大。

最基本的数据结构就是线性表、堆栈以及队列。而根据在内存中存放的方式不同也分为顺序存储和链式存储两种方式。学习数据结构,一定要对这些基础的东西有一个很深入的了解,才能在应用中游刃有余。

而算法的好坏是衡量一个程序性能的重要标准。尽管计算机硬件技术的飞速发展使得算法优化所能够起到的作用越来越小,但良好的算法素养一定能够在开发大型程序,特别是游戏中使你获益颇多。

最通用的两类算法就是排序和搜索,在实际的编程中,这两类算法的应用也比较广泛。根据实际情况的不同选择不同的算法能够达到最好的效率。

值得注意的是,目前的很多程序设计语言都加入了类似的算法库。C ++ 甚至把这类的算法库加入到标准中来,称为 standard template library,即 STL。下面就是一段 STL 的程序,它能够在 Microsoft Visual C ++ 中编译通过。

```
// The debugger can't handle symbols more than 255 characters long.
// STL often creates symbols longer than that.
// When symbols are longer than 255 characters, the warning is disabled.
#pragma warning(disable:4786)
#include <iostream>
#include <vector>
using namespace std;
typedef vector<int> INTVECTOR;
const ARRAY_SIZE = 10;
void ShowVector(INTVECTOR &theVector);
void main()
{
    // 动态分配一个向量
    INTVECTOR theVector;
    // 初始化向量,并把0-9这几个数字插入到向量中
    for (int cEachItem = 0; cEachItem < ARRAY_SIZE; cEachItem ++)
        theVector.push_back(cEachItem);
        // 使用迭代子来删除第6个元素
    theVector.erase(theVector.begin() + 5);
```