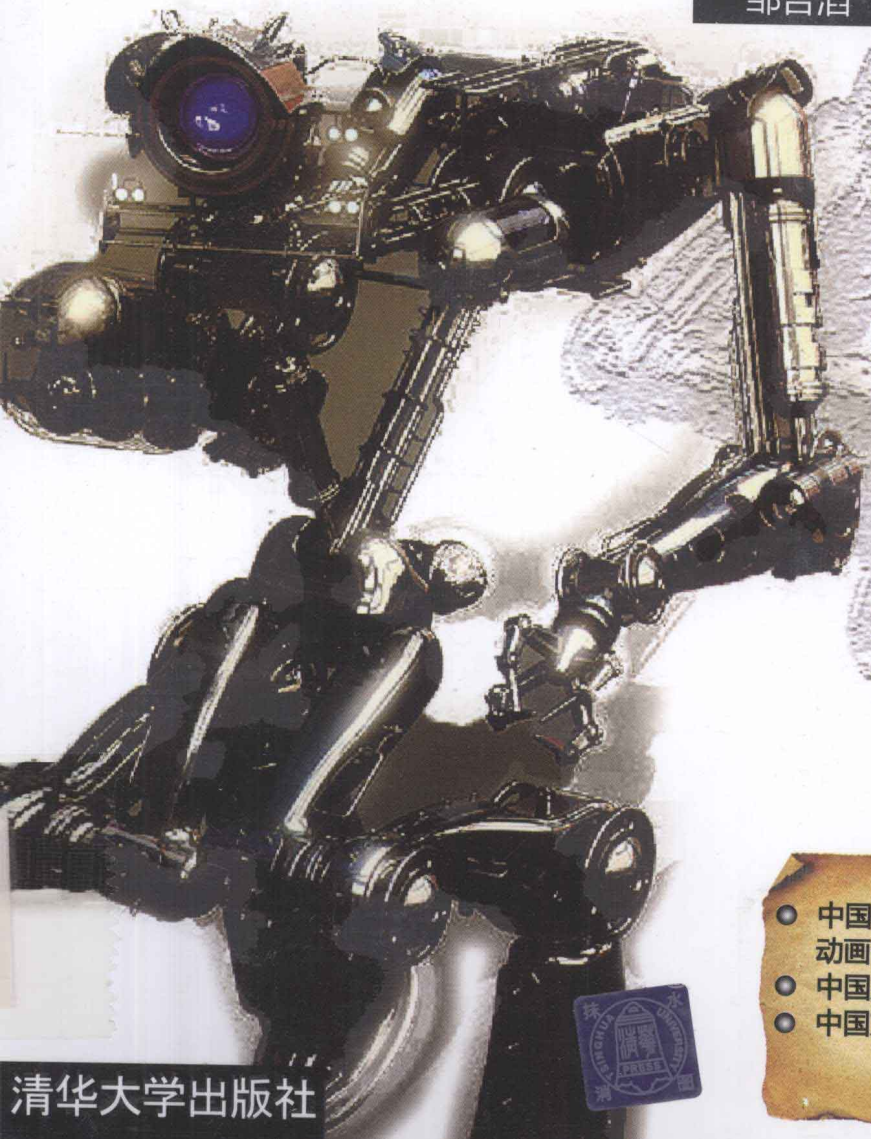
RGDC 游戏开发课程体系

第九艺术学院——游戏开发系列
COLLEGE OF THE NINTH ART

C++游戏编程

邹吉滔 姚 雷 易巧玲 编著



- 中国电影电视技术学会数字特效与三维动画专业委员会
- 中国系统仿真学会数字娱乐专业委员会
- 中国文化创意产业技术创新联盟

推荐教材

清华大学出版社

第九艺术学院——游戏开发系列

C++游戏编程

邹吉滔　姚　雷　易巧玲　编著

清华大学出版社

北　京

内 容 简 介

本书介绍如何用C++语言进行游戏程序开发。全书可分为C++语言的基础语法、面向对象编程技术、标准模板库的应用三个部分，共18章，主要内容包括：概观程序设计，开发环境简介，基本数据类型，运算符与表达式，程序的结构，宏和编译预处理，数组，函数与程序结构，指针和引用，结构、联合、枚举，类与对象，静态成员与友元，继承与多态，运算符重载，模板，标准模板库，I/O流，异常处理等。

本书适合游戏开发人员及游戏相关专业师生学习使用，也可供C++编程爱好者参考。

图书在版编目(CIP)数据

C++游戏编程/邹吉滔，姚雷，易巧玲编著. --北京：清华大学出版社，2011.1
(第九艺术学院——游戏开发系列)
ISBN 978-7-302-24591-9

Ⅰ. ①C… Ⅱ. ①邹… ②姚… ③易… Ⅲ. ①C语言—程序设计—高等学校—教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2010)第254498号

责任编辑：张彦青　桑任松
封面设计：杨玉兰
责任校对：周剑云
责任印制：杨　艳
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：北京市清华园胶印厂
装 订 者：三河市金元印装有限公司
经　　销：全国新华书店
开　　本：185×230　**印　张**：30.5　**字　数**：659千字
版　　次：2011年1月第1版　　**印　　次**：2011年1月第1次印刷
印　　数：1～4000
定　　价：50.00元

产品编号：023914-01

丛书编委会

前　言

在所有的编程语言中，C++可以说是最为复杂的，它既是一门传统的编程语言，也是一门新的编程语言。

说它是一门传统语言，是因为 C++已有 20 多年的历史了；特别是最近 10 年来 C++得到了快速的发展。C++是计算机软件领域中覆盖面最为广阔的编程语言，并且，与 C++相关的智力投入也是其他任何一门语言所无法比拟的。人们对于 C++的研究已经远远超出了对于一门编程语言所应有的关注。所以，现在的 C++已经非常成熟，有大量的资源(文档、书籍、源代码等)可供我们使用。

说它是一门新的编程语言，是因为在 1998 年 C++由国际标准化组织 ISO(International Standards Organization)完成了标准化，从此 C++领域有了统一的标准。所有的编译器都将向标准靠拢(或者说，与标准兼容)，这有利于我们写出可移植的 C++代码来。同时 C++标准也统一了 C++标准库，为 C++用户提供了最为基本的基础设施。C++经历了多年的发展，终于有了一个相对稳定的版本，所以，我们应该用一种新的眼光来看待 C++，而不再简单地把 C++认为是 C 语言的超集。通过本书，读者可以重新审视 C++语言。

随着编程技术在我国的普及，C++语言已经为众多程序设计人员所掌握，相关的书籍也层出不穷，但是现在国内 IT 企业拥有学士、硕士、博士文凭的软件开发人员比比皆是，可由于他们在接受大学教育时就没有对软件开发工作中有关的质量属性(如正确性、健壮性、可靠性、效率、易用性、可读性(可理解性)、可扩展性、可复用性、兼容性、可移植性等)进行深入的分析和探讨，并且在实践中运用相关的技术。“高质量”可不是通过读书就能学会的，那需要长期的实践工作以及对工作的成败进行不断的总结，才能实现！

因此有充分的理由疑虑：

(1) 编程老手可能会长期用隐含错误的方式编程(习惯成自然)，发现毛病后都不愿相信那是真的！

(2) 编程高手可以在某一领域写出极有水平的代码，但未必能从全局把握软件质量的方方面面。

事实证明确实如此。大多数“老”程序员的编程技能，质量合格率大约是 10%。很少有人能够写出完全符合质量要求的代码，很多程序员对指针、内存管理一知半解。

在这样的前提之下，笔者斗胆试着把工作中的一些经验心得以及从无数个前辈那里学到的知识整理出来，编写一本 C++的基础教学用书，不指望能够把 C++的知识全部说出来，

只希望让初学者能够很快看到 C++知识基础面貌以及编程的基本方法和风格。本书的一个重要特点是在讲解一些重要知识点的时候，用同一个例子进行说明，通过不断的升级改造同一例程，来说明不同知识点的运用技巧，这些例子主要是在游戏程序设计中用到的一些基础程序，因此对有志作为游戏程序设计者的帮助尤其显著。

本书主要分为 3 个部分，第一部分为 C++语言的基础语法部分，主要用来讲述语法知识以及程序的基本结构，运算符、数组、函数、预处理等知识。这部分知识中，有关内存处理部分，需要读者认真体会，第二部分是本书的真正重点，主要讲述面向对象编程技术，这部分的知识，主要通过几个贯穿这部分始末的例子来描述，其中一个重要的例子是一个简单的栈设计及应用，讲解不同知识点的时候，将利用相关的知识点对栈进行升级改造，读者可以充分体会到每个知识点在什么情况下运用可以发挥更大的作用。另外一部分重要的例子是如何在游戏程序设计中利用面向对象编程技术，可以通过设计游戏中的精灵类，来理解面向对象技术的强大作用。本书的第三部分，不作为重点进行讲述，但仍然需要读者认真学习，第三部分主要讲述标准模板库的应用。

本书由邹吉滔，姚雷，易巧玲编著。由于各方面的原因，书中疏漏之处在所难免，恳请广大读者批评指正。

编 者

目　录

第 1 章　概观程序设计

本章内容：

- 计算机程序发展历程。
- 程序设计思想变迁史。

重点：

面向对象编程思想。

目的：

了解程序设计发展史。

1.1　程序设计发展历程

1.1.1　什么是计算机程序

自 1946 年世界上第一台电子计算机问世以来，计算机科学及其应用的发展十分迅猛，计算机被广泛地应用于人类生产、生活的各个领域，推动了社会的进步与发展。特别是随着国际互联网(Internet)日益深入千家万户，传统的信息收集、传输及交换方式正被革命性地改变，人类已经难以摆脱对计算机的依赖，计算机已将人类带入了一个新的时代——信息时代。

一台计算机是由硬件系统和软件系统两大部分构成的。硬件是计算机的物质基础；而软件可以说是计算机的灵魂，没有软件，计算机只是一台“裸机”，什么也不能干，有了软件，才能灵活起来，成为一台真正的“电脑”。而所有的软件，都是使用计算机语言编写出来的。

软件程序是控制计算机完成特定功能的一组有序指令的集合，编写程序所使用的语言称为程序设计语言。电脑所做的每一次动作，每一个步骤，都是按照已经用计算机语言编好的程序来执行的。程序是计算机要执行的动作的集合，而程序全部都是用程序设计语言来编写的。所以人们要控制计算机一定要通过计算机语言向计算机发出命令。程序设计语

言经历了机器语言、汇编语言到高级语言的多个阶段。

随着计算机技术的不断进步和计算机图形技术的发展，计算机程序提供了对现实世界越来越逼真的模拟，在这种情况下一种新的计算机程序出现了，这种程序的目的不再是为了计算复杂的弹道数据，或是模拟宇宙的演化，它只是为了一个简单的目标：娱乐。就这样，电子游戏产生了，同时也产生了游戏程序员。

游戏程序员的程序设计能力决定着他能为游戏策划实现哪些功能，能做出什么样的游戏，可以说程序员的水平决定了游戏的成败。对于游戏程序员而言，掌握一门高级语言及其基本的编程技能是必需的。不仅如此，作为专业技术人员，除了掌握本专业系统的基础知识外，科学精神的培养、思维方法的锻炼、严谨踏实的科研作风的养成，以及分析问题、解决问题的能力的训练，都是日后工作的基础。学习计算机语言，正是一种十分有益的训练方式，而语言本身又是与计算机进行交互的有力工具。

1.1.2　计算机程序语言的发展历史

计算机程序设计发展的历史，也就是计算机程序设计语言的发展历史，它经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言

电子计算机所使用的是由“0”和“1”组成的二进制数，二进制是计算机的语言基础。在计算机发明之初，计算机所能识别的语言只有机器语言，即由 0 和 1 构成的代码。所以人们只能放弃自己的自然语言，用计算机的语言去命令计算机干这干那儿，也就是写出一串串由“0”和“1”组成的指令序列交由计算机执行，这种语言就是机器语言。

使用机器语言是十分痛苦的，特别是在程序有错需要修改时更是如此。而且，由于每台计算机的指令系统往往各不相同，所以，在一台计算机上执行的程序，要想在另一台计算机上执行，必须另编程序，这就造成了重复工作。但由于使用的是针对特定型号计算机的语言，且不需要经过翻译、编译等过程，而是直接执行的，故而它的运算效率是所有语言中最高的。机器语言，是第一代计算机语言。

2. 汇编语言

为了减轻使用机器语言编程的痛苦，提高程序开发的效率，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来替代一个特定指令的二进制串，比如，用“ADD”代表加法，“MOV”代表数据传递等。这样一来，人们就比较容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言，即第二代计算机语言。然而计算机是不认识这些符号的，这就需要一个专门的程序，专门负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编程序。

汇编语言同样十分依赖于机器硬件，移植性不好，但效率仍十分高，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精练而质量高，所以至今仍是一种常用且强有力的软件开发工具。

3. 高级语言

从最初与计算机交流的痛苦经历中，人们意识到，应该设计一种这样的语言，这种语言接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用。经过不懈的努力，在 1954 年，第一个完全脱离机器硬件的高级语言——FORTRAN 问世了，40 多年来，共有几百种高级语言出现，有重要意义的有几十种，影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、Delphi 和 Java 等。

高级语言的发展也经历了从早期语言到结构化程序设计语言，从面向过程到面向对象程序语言的过程。相应的，软件的开发也由最初的个体手工作坊式的封闭式生产，发展为产业化、流水线式的工业化生产。

过去的软件生产基本上是以个人为单位进行开发，缺乏科学规范的系统规划与测试、评估标准。当这样的工作方式应用于大型软件的开发时，恶果是大批耗费巨资建立起来的软件系统，由于含有错误而无法使用，甚至带来巨大损失，软件给人的感觉是越来越不可靠，以致几乎没有不出错的软件。这一切，极大地震动了计算机界，史称“软件危机”。人们认识到：大型程序的编制不同于写小程序，它应该是一项新的技术，应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性，也便于验证正确性。1969 年，荷兰计算机科学家迪克斯特拉(E.W.Dijkstra)提出了结构化程序设计方法；1970 年，第一个结构化程序设计语言——Pascal 语言出现，标志着结构化程序设计时期的开始。

从 20 世纪 80 年代初开始，在软件设计思想上又产生了一次革命，其成果就是面向对象的程序设计思想。在此之前的高级语言，几乎都是面向过程的，程序的执行如同流水线似的，在一个模块被执行完成前，人们不能干别的事，也无法动态地改变程序的执行方向，这和人们日常处理事物的方式是不一致的。对人而言，希望发生一件事就处理一件事。也就是说，不是面向过程，而应是面向具体的应用功能，也就是对象(Object)。其方法就是软件的集成化，如同硬件的集成电路一样，生产一些通用的、封装紧密的功能模块，称之为软件集成块，它与具体应用无关，但能相互组合，完成具体的应用功能，同时又能重复使用。对使用者来说，只需关心它的接口(输入量、输出量)及能实现的功能，至于是如何实现的，那是它内部的事，使用者完全不用关心，C++、Delphi 就是这种语言的典型代表。

随着技术的不断进步，高级语言的下一个发展目标是面向应用，也就是说，只需要告诉程序你要干什么，程序就能自动生成算法，自动进行处理，这就是非过程化的程序语言。

1.2　程序设计思想

1.2.1　结构化程序设计思想

结构化程序设计(Structured Programming，SP)的思想是由荷兰学者 E.W.Dijkstra 提出的，它规定了一套方法，使程序具有合理的结构，以保证和验证程序的正确性。这种方法要求程序设计者不能随心所欲地编写程序，而是要按照一定的结构形式来设计和编写程序。它的一个重要目的是使程序具有良好的结构，使程序易于设计，易于理解，易于调试修改，以提高设计和维护程序工作的效率。

结构化程序设计是 E.W.Dijkstra 等人在研究人的智力局限性随着程序规模的增大而表现出不适应的特点之后，于 1969 年提出的一种程序设计方法，这是一种在解决复杂问题时避免混乱的技术。它提出了把程序结构规范化的主张，要求对复杂问题的求解过程应按人们大脑容易理解的方式进行组织，而不是强迫人们的大脑去接受难以忍受的冲击。具体来说，结构化程序设计的思想包括以下三方面的内容。

- 程序由一些基本结构组成。任何一个大型的程序都由三种基本结构所组成，由这些基本结构顺序地构成一个结构化的程序。这三种基本结构为：①顺序结构，如图 1-1 所示；②选择结构(亦称分支结构)，如图 1-2 所示；③循环结构，如图 1-3 所示。

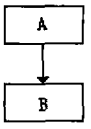


图 1-1　顺序结构

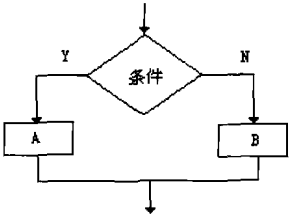


图 1-2　选择结构

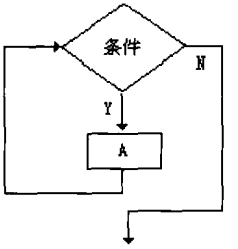


图 1-3　循环结构

同时结构化定理还进一步表明，任何一个复杂问题的程序设计都可以用顺序、选择和循环这三种基本结构组成，且它们都具有以下特点：只有一个入口；只有一个出口；结构中无死循环。程序中三种基本结构之间形成顺序执行关系。

- 一个大型程序应按功能分割成一些功能模块，并把这些模块按层次关系进行组织。
- 在程序设计时应采用自顶向下、逐步细化的实施方法。

用八个字来描述结构化程序设计的基本思想就是：自顶向下，逐步求精。

“自顶向下”是将大而复杂的问题划分为多个小问题，找出问题的关键、重点所在，然后用精确的思维定性、定量地去描述问题。

“逐步求精”是将现实世界的问题经抽象化处理转化为逻辑空间或求解空间的问题；复杂问题经抽象化处理变为相对简单的问题。经若干步抽象(精化)处理后，最后到求解域中的只是比较简单的编程问题。

在这个“自顶向下、逐步求精”的细化过程中，要对系统功能逐层分解出许多易于理解和实现的、逻辑上相对独立的子功能，形成一棵功能树。在程序实现时，功能树上的每一个结点对应一个函数。一个 C 程序是由一到多个函数组成；一个 C++程序不但可以包含多个函数，而且还可以包含多个类，而一个类就包含一到多个成员函数。

按结构化程序设计方法设计出的程序优点是：结构良好、各模块间的关系清晰简单、每一模块内都由基本单元组成。这样设计出的程序清晰易读、可理解性好、容易设计、容易验证其正确性，也容易维护。同时，由于采用了“自顶向下、逐步细化”的实施方法，能有效地组织人们的智力，有利于软件的工程化开发。

1.2.2　面向对象程序设计思想

随着计算机技术的不断发展，其软硬件之间的差距越来越大，这就造成了计算机发展的不均衡。当系统较为复杂时，常规的软件工具、技术和概念已不足以应付，从而使软件开发陷入了困境，即所谓的“软件危机”。尽管软、硬件发展的这种差距自计算机出现以来始终存在，但进入 20 世纪 90 年代后这种差距更加明显。在这一背景下，面向对象(Object-Oriented Programming，OOP)程序设计技术逐渐兴起，而 C++这种最为优秀的面向对象语言也应运而生，随着它的不断完善，其逐步进入实用阶段并受到广大软件开发者的青睐，吸引了众多的人士去研究和使用它。而面向对象的思想也在软件工程、人工智能等领域得到了十分广泛的应用。面向对象的程序设计语言的出现是计算机软件产业的一次革命。

1. 面向对象程序方法的基本思想及对象的含义

面向对象程序设计方法的基本思想是：认为世界由各种对象组成，任何事物都是对象，是某个对象类的实例；复杂的对象可以由比较简单的对象以某种方式组成。按照这个观点，整个世界也可以从一些最原始的对象开始，经过层层组合而成。因此，可以说整个世界就是一个最复杂的对象。

在面向对象程序设计中，对象是系统中的基本运行实体，是有特殊属性(数据)和行为