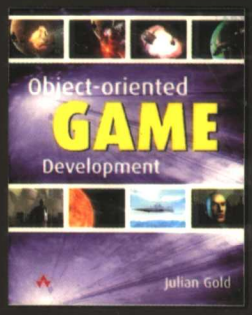
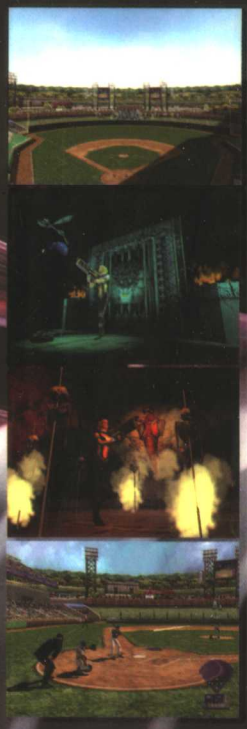
Broadview
www.broadview.com.cn



OBJECT-ORIENTED GAME DEVELOPMENT

面向对象的游戏开发

[美] Julian Gold 著
陈为 周骥 杨珂 邹林灿 等译



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

OBJECT-ORIENTED
GAME DEVELOPMENT

面向对象的游戏开发

[美] Julian Gold 著
陈为 周骥 杨珂 邹林灿 等译

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书以作者 10 余年游戏开发的经验为基础，用生动易懂的语言，介绍了面向对象游戏开发中的程序设计要素，特别是游戏引擎的团队开发的基本知识。内容包括：以商业软件代码为例说明实用的面向对象设计方法、实用的设计模式、必要时编写可重用代码、使用组件技术编写游戏代码、使用迭代技术进行编程开发和进度安排等。

本书不仅适合于攻读游戏开发方面学位的学生、对游戏引擎设计感兴趣的研究生、从事游戏开发的制作人和艺术家以及广大电脑爱好者阅读，而且还可以了解游戏开发的理念，掌握游戏开发的基本技巧和流程，从而有助于用不断更新的技术开发越来越复杂的游戏软件。

版权贸易合同登记号　图字：01-2004-5285

图书在版编目（CIP）数据

面向对象的游戏开发 /（美）戈德（Gold, J.）著；陈为等译. —北京：电子工业出版社，2005.6
书名原文：Object-oriented GAME Development
ISBN 7-121-01222-7

Ⅰ. 面…　Ⅱ. ①戈…　②陈…　Ⅲ. ①面向对象语言－程序设计　②游戏－应用程序－程序设计
Ⅳ. ①TP312　②G899

中国版本图书馆 CIP 数据核字（2005）第 047737 号

责任编辑：顾慧芳
印　　刷：北京智力达印刷有限公司
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：26　字数：136 千字
印　　次：2005 年 6 月第 1 次印刷
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。

译者序

人生如戏。每个人的一生都充满未知因素，然而，游戏可以重来，人生却无法存盘。从某种程度上说，计算机游戏可以在短短的数十小时之内，模拟人在一生中几乎所有的探索、活动、思维、幻想等等一切，玩家可以在游戏提供的虚拟世界中体验丰富多彩的生活，不可谓不动人。对计算机游戏的策划、开发和设计者而言，提供玩家这种体验无疑是巨大的挑战。而要成为一个合格的计算机游戏程序设计人员，不仅需要掌握程序设计技巧和多种专业领域知识，还需要对计算机游戏的基本内涵、开发过程和游戏产业发展状况有个基本的了解，才能有所作为。

在国外，游戏开发经历了20余年的风风雨雨，现已经成长为一个高技术、高利润和高速发展的行业。在中国，一方面，游戏特别是网络游戏，构成了IT行业中新兴的利润增长点，国际数据公司和中国出版工作者协会游戏工作委员会的2003年中国游戏产业报告指出：2002年中国游戏市场的规模为10.2亿元人民币，2003年的市场规模为17.8亿元，2004年上半年已达15.5亿元，全年可达到35亿元，并预计到2007年，中国游戏市场的规模将达到67亿元，年复合增长率将达到49.2%，如此之高的增长率充分表明了中国的游戏产业正处于快速发展时期。

另一方面，由于历史的原因，我国的游戏产业尚处于起步阶段，国内游戏开发水平低，人才匮乏，而优秀叫座的游戏都是舶来品，本土市场的利润大部分流向国外。因此，着手游戏开发的教育、研究和开发工作势在必行。2004年，浙江大学、四川大学、山东大学等高校的计算机学院相继开出了计算机游戏程序设计方面的课程，浙江大学、中国美术学院等高校还设立了与游戏开发相关的专业。然而，目前优秀的计算机游戏程序设计教程、参考资料在市面上仍不多见。

计算机游戏程序设计涉及的知识面极广，包括软件工程、面向对象的程序设计、数据结构、游戏引擎、图形动画、音频、网络、人工智能等方面。优秀的计算机游戏程序员不仅需要具有计算机编程方面的知识，还需要掌握图形、美术、物理等方面的相关内容。本书以生动易懂的语言，以作者10余年的游戏开发经验为背景，介绍了面向对象游戏开发中的程序设计要素，特别是游戏引擎的团队开发的基本知识。本书主要介绍的内容包括：以商业软件代

码为例说明实用的面向对象设计方法、实用的设计模式、必要时编写可重用代码、使用组件技术编写游戏代码、使用迭代技术进行编程开发和进度安排。本书的潜在读者，攻读游戏开发方面学位的学生、对游戏引擎设计感兴趣的硕士生，已经在游戏公司工作的程序员、制作人、设计师和艺术家，都将在本书中阅读到大量满足现实需要的实用技术，了解游戏开发的基本理念，熟悉游戏开发的基本技巧和流程，从而有助于其在使用不断更新的技术开发越来越复杂的娱乐软件时按时按预算完成任务。

本书的翻译工作由陈为博士组织，由陈为（浙江大学计算机学院）、周骥（上海威盛公司）、杨珂（浙江大学计算机学院）、董兆华（浙江大学计算机学院）和邹林灿（浙江大学计算机学院）负责翻译。其中第1到第3章由陈为、邹林灿负责翻译，第4、5章由周骥负责翻译，第6、7章由董兆华负责翻译，第8、9、10和11章由杨珂负责翻译，附录、序言和封底等由陈为负责翻译，全书由陈为负责定稿、审校，感谢朱沐红、顾慧芳、张龙、赵爱东等在翻译过程中的大力协助，在此一并致谢。

本书原版英文比较口语化，并夹杂了大量的俚语和网络用语，这给翻译工作带来了一定困难，但是我们经过多次审校，克服了这些困难。由于译者水平和学识所限，译本中难免存在错误和不妥之处，在此我们恳请读者批评指正。译者联系电子邮件：shearwarp@hotmail.com。

译　者

2005年1月28日于浙江大学计算机学院

致　谢

本书的编写得到了许多睿智的专家和学者的帮助。

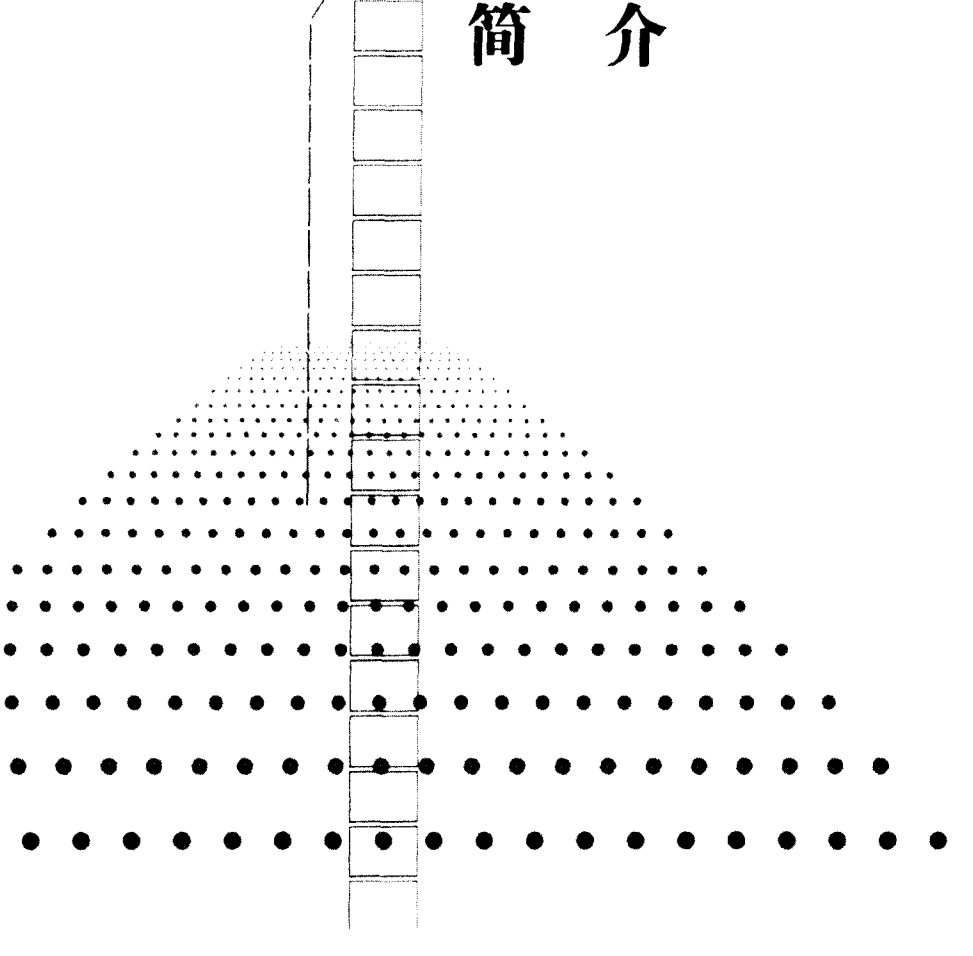
Six By Nine 公司的 Sam Brown、Peter Bracher、Adrian Hirst、Graeme Baird 及 Beverley Shaw 对本文许多想法提供了宝贵的意见。他们不愧是我真正的朋友。

Emma 对本书进行了校对。他修改了我的许多过于修饰性的语句，使它们变得直白而严谨。最重要的是要感谢他对我的许多不良生活习惯（例如一边打字一边吃东西及喝酒）的容忍。

目　录

第1章

简介



1.1 关于本书

当你从书店的书架上拿起这本书的时候，多半你会怀疑这又是一本有关所谓“内核秘密”的书。市场上有很多书声称将讲述叫好的游戏开发过程中的真经，事实证明却只是一些改头换面、勉强拼凑的免费手册。我自己也曾买过此类书籍，如果你掩卷之后未曾如我一般失望之极并由此引发疑问，那我也无话可说。我只是希望这本书与那些书不同，它应当给予你从别处看不到的真知灼见，而当你读完后，会有一种豁然开朗之感。

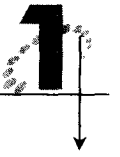
本书的目的是讨论可行的游戏开发方法。书中将借鉴一些传统的开发术语，同时增加少量的软件工程原则、面向团队的开发进程、项目管理和一些与之相关的常识。每个专题都已有人单独做过详尽的描述，其中一些将被本书引用。但据我所知，还没有一本书将这些专题就游戏开发中的应用细节进行详细的阐述，因此这本书会展示一些与众不同的内容。

我希望本书会改变读者的一些观念，也欢迎读者对书中的内容提出质疑。当然，本书无法在每个技术或管理细节上提供完美的解决方法，因为如果这些问题容易解决，早就人所共知了。我写本书的思路是以一些典型的问题为例，提供行之有效的解决方法，这样，比空谈理论要强。

1.2 背景

当我们步入 21 世纪的时候，商业游戏开发也到了一个十字路口。当我写下这些文字的时候，三个新的游戏平台正在欧洲兴起：索尼（SONY）公司的 PlayStation™2，任天堂（Nintendo）公司的 GameCube™ 和微软（Microsoft）公司的 X-BOX™。这些平台的出现，代表了游戏技术的又一次飞跃；无论这些平台的实际能力如何，宣传它们所用的夸大广告和随之而来的公众高涨的期待必将呼唤更好、更复杂、更具沉浸感的游戏诞生。

判断一个游戏的开发是否可管理的标志是游戏开发的时间与游戏的规模相适应程度。事实上，在游戏产业中游戏项目延期已成为家常便饭。开发者如果不想令公众失望，按时根据预算完成游戏开发，便恨不得将一天当做 30 个小时来过。



那么，如何缩短开发周期呢？当然，指望增加工作时间是行不通的。因此，我们必须求助于传统的管理手段，使得游戏项目的开发更为有效，而不是一味地苦干。

本书简要地介绍了一些如何巧干的方法。这些方法将使我们摆脱现有技术的局限，从容应对下一个新的技术平台带来的不断增长的游戏开发的复杂性。

1.3　本书作者

在过去的 10 年中，我一直致力于或大或小的游戏项目开发。在此之前，我获得了天文学专业的学士学位并打下了坚实的数学功底。我花费了几年时间用 FORTRAN 语言为英格兰皇家格林尼治天文台（Royal Greenwich Observatory）编写了一个图像处理软件。此后，我从 1993 年开始开发视频游戏。我在不同规模的游戏公司工作过，做过各种各样开发平台上的游戏程序员，一直做到主程序员。其间我发现这些公司都会或多或少地重复同样或类似的错误。其中一个重要的教训是，分析并处理这些错误，比仅仅获得员工或其他资源，具有更重要的借鉴意义。

1.4　本书读者

尽管这本书谈的不仅仅是程序，但它的潜在读者还是游戏程序员。原因很简单：当今的游戏开发已经发展为一个包含多个学科方向的综合过程：

- 编程
- 艺术
- 设计
- 音乐
- 其他

当然，还包括对上述要素的管理。

一个程序员的日常工作不仅包括与其他程序员的交流，也包括与其他专业背景的人的沟通。简而言之，在游戏开发过程中，个人的工作是无法与团队的工作割裂开来的，因此单单考虑编程技术而不参考其他专业学科是极度错误的。不失一般性，在这本书中我坚持认为编程、艺术和设计是游戏开发过程中神圣的三位一体。

如果你是一个程序员，或程序员主管，或与程序员一道工作的开发者，你肯

定处于一个团队或者说商业环境中。在业余时间与朋友们一道开发程序是一件很愉快的事情，但处于商业压力下开发程序则完全是另一回事，我们将在本书中探讨其中涉及到的许多问题。即便你喜欢单干，不管你是一个程序员、艺术家、设计师、音乐设计师或策划师，你仍然可以在书中找到一些有益的东西。

1.5 内容概要

本书可以分成两个主要部分。第一部分讨论在商业游戏开发中遇到的主要问题。我将尽量在足够抽象的层面上讨论问题，使得问题具有普遍意义和实际应用价值。同时，我还会保持一定的可操作性，使之可以在现实中具备一定的指导意义。事实上，我将尽量抽象化这些问题，因为我无法预测读者将会遇到什么具体的技术问题。

第二部分是关于游戏编程的，或者说是游戏开发中的软件工程。我会谈论如何选择游戏开发语言以及选择的理由。我还将讨论高层的游戏架构设计以及如何将之转换成最终的代码，并将介绍一种在交叉平台上分析和实施游戏开发的方法。最终，我将在前面探讨的原理基础上开发某些特定的游戏系统。

如果你是一个主程序员或技术主管，你可能更关心游戏开发过程。如果你是一个上进心强的程序员，你也许会发现其他有关开发的章节更为实用。事实上，本书各个章节之间的关联性较少，读者完全可以随意阅读本书中的各个章节而不需考虑前后顺序。

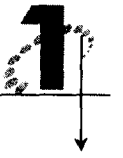
闲话少说，让我们开始吧。

1.6 游戏发展的简要历史

回首过去，我们非常有必要对游戏发展的历史作一个简要的介绍，从中可以看到那些令人激动的技术是来自哪个天才的大脑，而这些技术又是如何推动游戏向前发展的。

1.6.1 遗忘的历史

计算机游戏的历史比许多人想像的要古老得多。有趣的是，人们总是乐于将人类之间互玩的游戏搬到计算机上。制作诸如国际象棋（chess）和西洋跳棋



（checkers）这样的游戏业已花费并且还将花费世界各地很多学院研究者和商业公司的巨大努力，它们的开发历史可以追溯到 20 世纪 50 年代。那时，电子计算机，不仅仅被当做军方计算导弹轨迹的工具，而且已经被用于民众教育基础设施和商业领域。Samuel Jackson 的跳棋算法就是一个典型的由游戏推动软件技术发展的例子。这个在 1956 年推出的算法可以判断在给定条件下某局游戏的结局，这令跳棋游戏成为一类为数不多的从技术上已经彻底解决的游戏。

令人惊讶的事情发生在 1958 年。一名在纽约 Brookhaven 国家实验室工作的研究员 William Higinbotham，利用一个真空管系统开发了人类历史上第一个图形视频游戏。这个简单的网球游戏，用一个小像素斑点代表网球，一个倒置的 "T" 字母表示球网。它是一个完全基于硬件的游戏，以至于当这个硬件系统被拆卸了以后，这款游戏也就从此消失了。

在那个年代，电脑硬件依然尺寸非常庞大且价格高昂，游戏技术只能在学院的环境中发展。这种情况一直沿袭到 20 世纪 70 年代，由于半导体元件的大批量生产带来的降价，情况才发生变化。

如果电脑硬件不实际而且不可靠，那么与之相伴的软件也同样如此。从手工二进制数的开关，到打孔卡，到可以把类似英语的文字翻译成相应的二进制数的指令程序（编译器、汇编语言和解析器），计算机语言发生了翻天覆地的变化。不过，在那个年代，程序员还是直接使用二进制数的代码，这极大地加重了程序和算法开发的难度和痛苦程度，导致了无数恼人的错误出现。

1962 年，一位马萨诸塞州坎布里奇 Hingham 学院的研究者 Steve Russell，开发了一款二人对战的太空船星战游戏 Spacewar（见图 1.1）。

Russell 于 1969 年访问斯坦福大学，Spacewar 游戏给该校的众多学生留下了极深的印象，其中的一位就是 Nolan Bushnell。Bushnell 在学习电子工程的同时还为一家主题公园工作。他想把电脑变成欢乐的一部分。于是他花费了大部分的业余时间设计出一个 Spacewar 游戏的商业版本，Computer Space。1971 年，Bushnell 的游戏成为世界上第一个投币电脑游戏。继而，Bushnell 成立了 Atari 公司，后来这家公司主宰了视频游戏长达 10 年之久。

1.6.2　学院派的天下

与此同时， 20 世纪 60 年代末，在位于美国加利福尼亚州的 AT&T 贝尔实

验室中诞生了划时代的多用户操作系统 UNIX。尽管 UNIX 本身与游戏关系不大，但 UNIX 的众多副产品却对游戏的发展产生了将近 10 年的巨大影响。其中的一个代表是 C 语言，它设计的本意是用于编写操作系统。很有意思的是，用于操作系统的编写仅发挥了 C 语言的部分能力，它的强大能力却使其成为了游戏开发产业的首选语言。当然，这是后话。

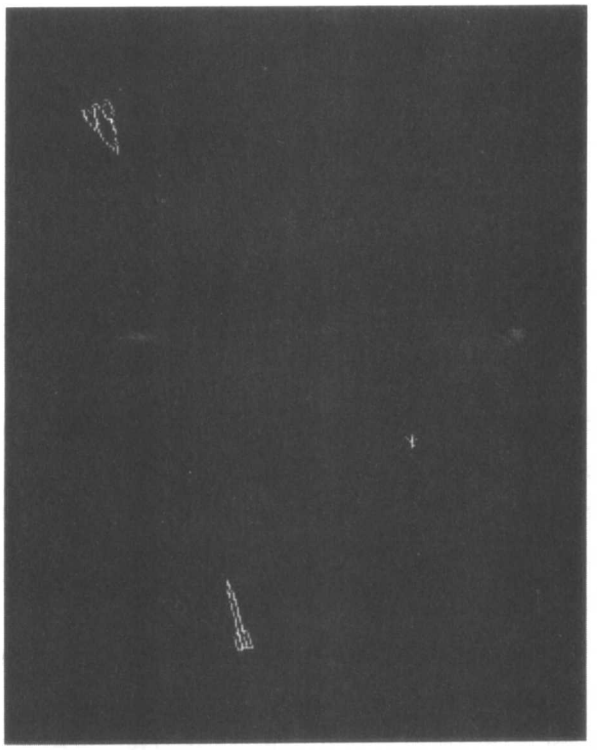


图 1.1
Spacewar：所有视频游戏之母

在 20 世纪 70 年代早期，Intel 开发了第一个具有现代中央处理器架构的商用单芯片 4004。随着大规模集成电路（VLSI）的发展，电脑配件的价格逐渐低廉，电脑走进大众的时代终于降临。一些商业的游戏控制台逐渐出现，它们通常配有只读存储器（ROM），一般只支持一些小游戏，如 Pong（William Higinbotham 的网球游戏的后续产品，见图 1.2）。

在此之前，开发游戏是硬件厂商们的版图。随着半导体技术的发展，个人电脑开始走进千家万户，电脑产业发展的瓶颈开始被突破。Sinclair 研究所的 ZX80 为电脑进入民用领域铺平了道路。值得注意的是，大部分的早期电脑都带有 BASIC。

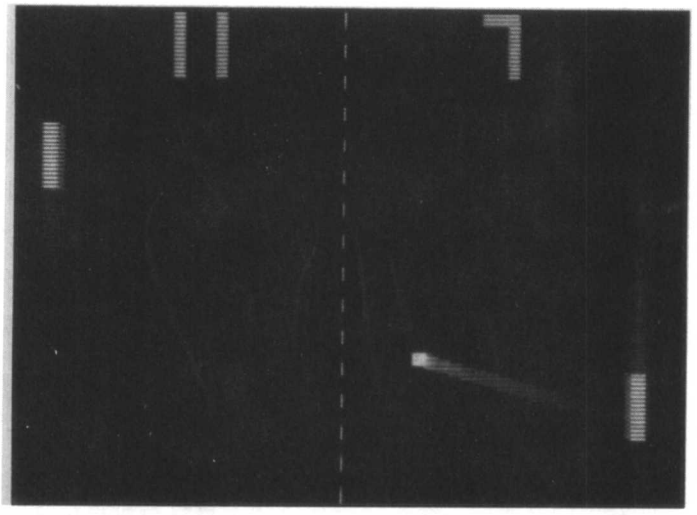


图 1.2
Pong- 网球游戏的后续产品

BASIC 的设计初衷并不是一种游戏开发工具。由于 BASIC 是一种解释性的语言，因此用 BASIC 语言编写的国际象棋或西洋跳棋速度缓慢。它无法流畅地刷新游戏画面，因此也不适用于 Pong 这样的图形视频游戏。然而，BASIC 的语法非常简单易学，这使得它成为许多人学习编程基础的启蒙语言。尽管有些人认为 BASIC 足够使用，但另外一些人则对 BASIC 的速度和能力感到不满。于是他们开始尝试利用 PEEK 和 POKE 命令修改数据，定制自己的图形环境，甚至编写机器指令。我自己也曾天真地花费了大量的时间直接输入二进制数据，结果往往令我失望。有时候，一个小小的打字错误就会使我所有的输入全部作废。

当人们拥有 Atari800，Commodore 64，BBC Micro，Sinclair Spectrum 等第二代计算机系统时，电脑游戏逐渐走向正轨。这些系统支持改进版的 BASIC，更重要的是，它们还支持汇编语言。这使得程序员可以抛开 BASIC 操作系统，直接对 CPU 进行编程。盒式磁带存储的发明使得软件能被迅速地发布开去，游戏产业从此诞生。

1.6.3　成长中的游戏

在 20 世纪 80 年代，随着游戏的规模逐渐扩大，游戏的复杂性也与日俱增，从 Pong 游戏中的块状精灵，到线形框架的 3D 游戏，如 David Braben 的 Elite。公众胃口越来越大，程序员也开始变得雄心勃勃，游戏技术也随之变得更加成熟。

最初，单个人可以自行设计游戏、编写代码、绘制游戏中所需图片以及编辑游戏音乐。游戏中所有的一切，他都可以亲力亲为。一般的游戏项目需要开发 1 个月到 6 个月的时间。当游戏的规模和复杂性逐渐变大时，两个问题产生了。

首先，一个游戏项目可能需要两个或更多的人参与，以便游戏可以按时完工。这两个人可以都在编程，也可以有一个编写代码，另一个负责艺术加工。不难发现，两个人一起工作，并不意味着工作进程可以加快一倍。两个人工作的效率与工作的分工息息相关。无论如何，商业游戏开发已不可避免成为一个团队项目。

其次，软件开发技术越来越繁杂，汇编语言已远远不能满足算法实现的要求。汇编语言的开发是一个耗时的工作，同时它也无法满足游戏公司开发跨平台游戏的需求。原因是，汇编语言的运行与平台相关，而平台移植[1]工作几乎就意味着重写整个游戏。

因此，游戏开发中真需要一个运行得足够快、且具有可移植性的编程语言，这将使得大部分代码可以与机器无关，从而加速开发进程。最初为了 UNIX 而设计的 C 语言，恰好符合了游戏产业界的所有要求，因此 C 语言突然变得热门。在 16 位平台（SNES，Sega Megadrive）和 32 位平台（Commodore 的 Amiga 和 Atari 的 ST）家用电脑共存于市场的时候，C 语言逐渐成为了商业游戏开发中最得力的工具。

C 语言是一个奇怪的综合体。一方面，它是与 BASIC 语言一样的高级语言；另一方面，它又是一种编译型的语言。在 20 世纪 80 年代，C 语言编译器编译产生的代码还很丑陋。但 C 是一种结构化的语言，它可以方便地被研究人员用来编写操作系统。BASIC 则完全不同，它是一种非结构的教学语言。而任何汇编代码也都没有结构的概念。在那时，转型到 C 语言对很多程序员是一件很痛苦的事情，这也导致了大量效率低下的 C 代码的产生。

1.6.4 从卧室到会议室

接下来的事情就顺理成章了。整个 20 世纪 80 年代是游戏发展的黄金时期，在 16 位平台或 32 平台上销售的游戏使一些人淘到了第一桶金。游戏开发的场所

1 “移植”的意思是要在两个系统上进行几乎相同的游戏开发（相对于在一个系统上工作而言），那就意味着要制作的程序既能在一台机器上执行又能在另一台机器上执行。