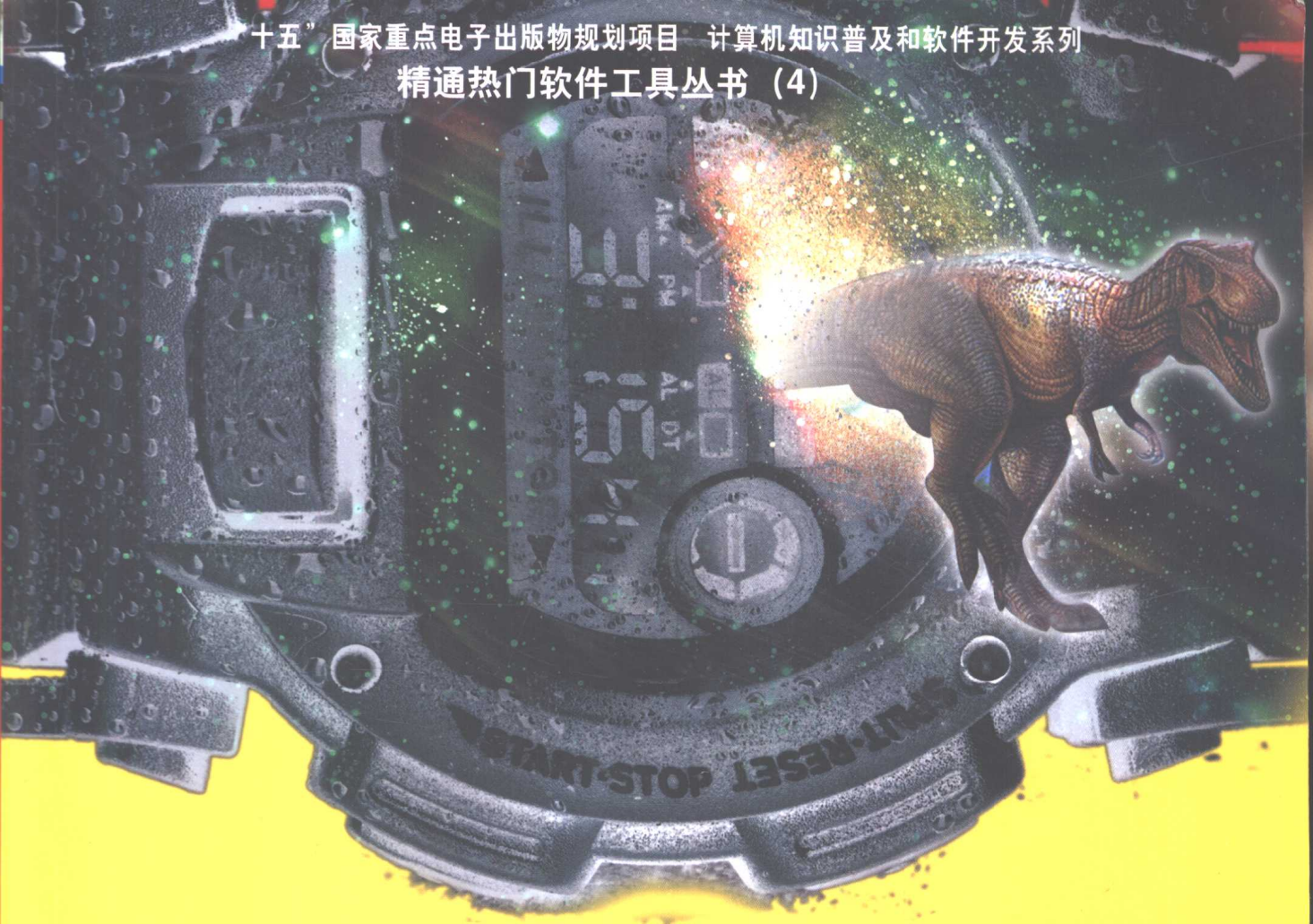


“十五”国家重点电子出版物规划项目 计算机知识普及和软件开发系列  
精通热门软件工具丛书 (4)



北京希望电子出版社 总策划  
浦滨 编 著

# 游戏编程 从入门到精通



北京希望电子出版社  
Beijing Hope Electronic Press  
[www.bhp.com.cn](http://www.bhp.com.cn)

十五 国家重点电子出版物规划项目 计算机知识普及和软件开发系列  
精通热门软件工具丛书 (4)



北京希望电子出版社 总策划  
浦滨 编 著

# 游戏编程 从入门到精通



北京希望电子出版社  
Beijing Hope Electronic Press  
[www.bhp.com.cn](http://www.bhp.com.cn)

## 内 容 简 介

本书以 C 语言游戏编程入手, 以 **102** 个实例, 近 **200** 个函数较为系统地介绍了 C 基于游戏编程与开发的方法与技巧, 内容丰富并相互包容, 相互渗透。以实际的基于不同平台的游戏制作为背景, 知识阐述与实际案例结合, 深入浅出, 具体、直观、全面, 可操作性强; 是一本难得的集入门、深入到精通 C 游戏编程的综合书籍。

该书将大学计算机及大量相关专业课程的知识运用到 C 语言游戏编程的实践中, 并从 C 语言游戏编程实践角度来诠释计算机及许多其他专业课题, 内容涉及计算机硬件、软件等方面的编程技术。具体内容包括, 图形绘制、中文显示、动画实现、文件调用、内存使用、声卡调用、中断、内存驻留技术、接口技术、数据库实现、简单病毒、界面技术等进行了详尽的介绍, 并且配以大量的源程序以及程序分析对所涉及的理论进行充分的讲解和支撑。

本书根据大学 C 语言教学需要, 适合于高校计算机和数学相关专业的学生以及所有 C 语言爱好者。此外, 对于 C/C++ 语言的初、中级用户, 业余爱好者学习与培训, 以及有一定软件开发经验的程序员和专业技术人员也有很好的借鉴和参考价值。

本版 CD 内容为游戏实例的源代码、编译程序、游戏函数及数据库等。

系 列 盘 书 : “十五”国家重点电子出版物规划项目 计算机知识普及和软件开发系列  
精通热门软件工具丛书 (4)

盘 书 名 : C 游戏编程从入门到精通

总 策 划 : 北京希望电子出版社

文 本 著 者 : 浦 滨

C D 制 作 者 : 希望多媒体开发中心

C D 测 试 者 : 希望多媒体测试部

责 任 编 辑 : 但明天

出 版、发 行 者 : 北京希望电子出版社

地 址 : 北京中关村大街 26 号, 100080

网址: [www.bhp.com.cn](http://www.bhp.com.cn) E-mail: [lwm@bhp.com.cn](mailto:lwm@bhp.com.cn)

电话: 010-62562329, 62541992, 62637101, 62637102, 62633308, 62633309

(图书发行和技术支持)

010-62613322-215 (门市) 010-62547735 (编辑部)

经 销 : 各地新华书店、软件连锁店

排 版 : 希望图书输出中心 邓蛟龙

C D 生 产 者 : 北京中新联光盘有限责任公司

文 本 印 刷 者 : 北京媛明印刷厂

开 本 / 规 格 : 787 毫米×1092 毫米 16 开本 25.5 印张 591 千字

版 次 / 印 次 : 2002 年 5 月第 1 版 2002 年 5 月第 1 次印刷

本 版 号 : ISBN 7-900101-01-2

印 数 : 0001-4000

定 价 : 39.00 元 (本版 CD)

说明: 凡我社产品如有残缺, 可执相关凭证与本社调换。

# 前 言

本书从 C 语言游戏编程入手，对图形绘制、中文显示、动画实现、文件调用、内存使用、声卡调用、中断、内存驻留技术、接口技术、数据库实现、简单病毒、界面技术等进行了详尽的介绍，并且配以大量的源程序以及程序分析对所涉及的理论进行充分的讲解和支撑。

该书将大学计算机及相关专业大量课程的知识运用到 C 语言游戏编程的实践中。这些课程包括：C 语言程序设计、C++ 语言程序设计、汇编语言、高等数学、计算机硬件、接口技术、数据库原理、数据结构、算法设计、软件工程。此外，还涉及到图形技术、动画技术、病毒与病毒防治编程等一系列专业知识。

本书共包括 17 章，具体内容分别为：

第 1 章 猜数字游戏：通过一个简单的游戏引导大家认识游戏编程。

第 2 章 语言函数库画图：介绍 C 语言自身提供的绘图函数。

第 3 章 简单动画：利用 C 语言绘图函数制作简单的动画效果。

第 4 章 简单图形游戏：在动画的基础上实现简单的图形游戏。

第 5 章 图形模式：介绍在制作游戏前如何将屏幕从文本模式设置为图形模式。

第 6 章 二维图形：介绍如何自行设计基本图形函数。

第 7 章 中文显示：主要介绍在图形中如何实现中文。

第 8 章 图形文件：图形文件的调用和显示，介绍 bmp、pcx 和 ico 文件的算法和显示。

第 9 章 动画原理：全面介绍各种动画实现的方法，并给出例程。

第 10 章 子画面技术：介绍最流行的子画面动画技术。

第 11 章 文件操作：文件的建立、写入和读取，主要用于游戏进度保存和游戏数据读取。

第 12 章 声音技术：主要介绍扬声、声卡的使用，以及游戏中的实现背景音乐。

第 13 章 内存技术：主要介绍 xms、EMS 内存技术及其在游戏中的应用。

第 14 章 接口技术：主要介绍键盘、鼠标和串口的驱动和使用方法。

第 15 章 界面技术：为游戏提供具有面向对象思路的图形、操作界面。

第 16 章 其它问题：通过加密和病毒保证游戏的版权。

第 17 章 游戏例程：通过一个实际的游戏编写来整合前面各章节学到的内容。

从构建一个全方位的 C 语言游戏实用函数库，使得函数数量达到 200 个左右。函数类别包括：图形函数、动画函数、图形文件调用函数、数据文件处理函数、中文显示函数、内存函数、声音函数、计算机基本功能设置函数、接口函数、界面函数、数据库函数等。需要说明的是，本游戏函数库是在对 Andre LaMothe 等编写的运行于 Microsoft C7.0 平台下的函数库进行平台改写、函数大量修改和扩充以及增加了 100 多个其他方面函数的基础上实现的。此外，本书还包括 108 个例程。

内容全面。游戏编程全方位阐述；涉及计算机各相关专业课程；适于编写各类型游戏，包括桌面智力游戏、视频战斗游戏、简单 RPG 游戏、简单联机游戏等。

在创新方面，将 TC 作为游戏编程平台。保护模式（TC 没有）等更适合写游戏，所以很少有人用 TC 写游戏；

其次，从游戏角度学习 C 语言，从一个新的角度看待编程教学。教师通常从专家角度进行教学，于是在基础教学过程中往往和学生脱节（学生无法理解其很多表达含义）。本书的内容和程序有不少是从大一到大四过程中的笔记修改而来的，与读者学习进度非常接近，从而更容易吸收。

在过程性语言中更多加入面向对象思路。在游戏对象结构定义的时候融入更多属性、事件函数指针和链表指针，在子画面处理和界面对象处理中更多采用了事件驱动的思路，从而为今后转向 C++ 游戏编程铺平道路。

此外，将计算机专业知识通过 C 语言游戏编程的视角进行了解、学习或巩固。平时学习各科目内容较为独立或者仅仅基于理论上的联系，通过游戏编程我们可以将它们贯穿起来学习，触类旁通。

本书读者对象：

大学计算机、数学及相关专业学生；游戏爱好者。

另外，本书适合大学 C 语言教学需要，内容衔接和难度上由浅入深，着重衔接 C 语言基础教材；也适合于高校计算机和数学相关专业的学生以及所有 C 语言爱好者。

游戏编程其实就是编程！如果你接触过程序设计，那么你打算写一个游戏已经不再是什么困难的事情。如果你认为自己虽然学过 C 语言，然而对于编程只有一点初步的了解，根本谈不上编写游戏的问题。我们是否有编游戏的基础呢？不过相信在看这本书的时候，你一定具备了：

- (1) 编程的初步基础；
- (2) 对编写游戏的强烈欲望。

### 1. 游戏编程的几大要素

一个成功的游戏通常有如下价值和特点。

- (1) 极高的美工水平；
- (2) 逼真的动画效果；
- (3) 精巧的构思；
- (4) 简便的操作。

可以说这与编程的关系不算很大，只有部分动画和用户操作是需要编程中下苦功夫的；而游戏成功的更重要原因是构思的独特和美工制作的赏心悦目。

以下是根据游戏的要素和编程的特点提出的游戏编程的五大要素：

(1) 动画效果。动画的效果一定要流畅，没有散动和尽量逼真，最好能够符合“物理学”和“生物学”运动规律。

(2) 人机交互性。游戏易于操作，并且能够快速响应，必要的时候能够设计出适合游戏的输入设备和驱动程序。

(3) 媒体多样性。提高图像、动画、声音和操作的同步性和混合性。

(4) 数据结构高效性。用于实现游戏的数据结构一定要经过规划，尽量简便、高效，以及适用面和扩充性强。

(5) 代码的通用性和对象化。引入面向对象的思路将对理清程序员思路、简化程序设计以及程序扩充等问题带来相当大的帮助。

这些编程要素事实上涉及到了包括直接写屏、中断、多任务、内存技术、动画技术、显示技术和优化算法等各种编程技术；涉及到了内存、PC 喇叭、声卡、键盘、鼠标、手柄、显卡等各种硬件原理和硬件编程；也涉及了包括《C 语言程序设计》、《C++语言程序设计》、《汇编语言》、《数据结构》、《数据库》、《计算机硬件》、《接口技术》、《算法》、《高等数学》、《概率论》、《人工智能》、《软件工程》和《多媒体技术》等计算机专业科目；此外，还涉及到《动画技术》、《三维游戏编程》、《加密、解密技术》和《病毒与病毒防治编程》等一些当前流行的计算机专业技术。

具备良好的计算机专业知识是实现游戏编程各大要素的前提。当然也不用太担心许多专业知识你还不太懂。因为这里毕竟是游戏编程，编程能力才是真正的关键，一切运用到的相关专业知识也都是围绕游戏编程的，在之后的章节中会根据相应编程的需求有所阐述。

其实，游戏编程可以看作动画编程、输入输出设备编程和多媒体编程的集合，自然在写程序前应当设计好数据结构及算法。

## 2. 游戏的流程

别看游戏有那么复杂的情节和玩法，从开始到发展再到结束的流程中，归根结底就只有四样东西：

动画 根据变量运算结果对屏幕中变化的对象进行重画，从而实现动画效果；

响应 对于来自键盘、鼠标等输入设备和计算机内部如时间等中断进行响应；

运算 根据各类具体响应通过计算和重新赋值来改变程序内变量数值；

循环 反复进行动画、响应和运算的操作来实现游戏的真正进程。

实现一个游戏的基本流程如右（流程图 A）。

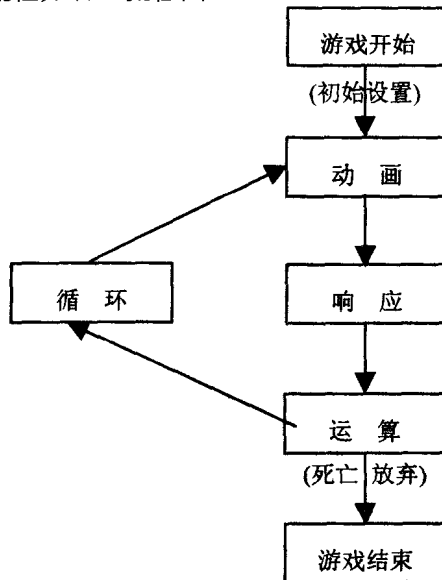


图 A 流程示意

下面以俄罗斯方块为例，动画流程分为以下几个步骤（图 B、图 C、图 D、图 E 及图 F）：

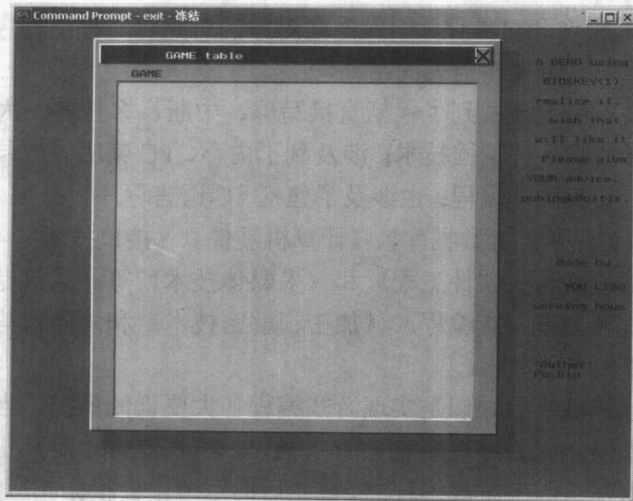


图 B

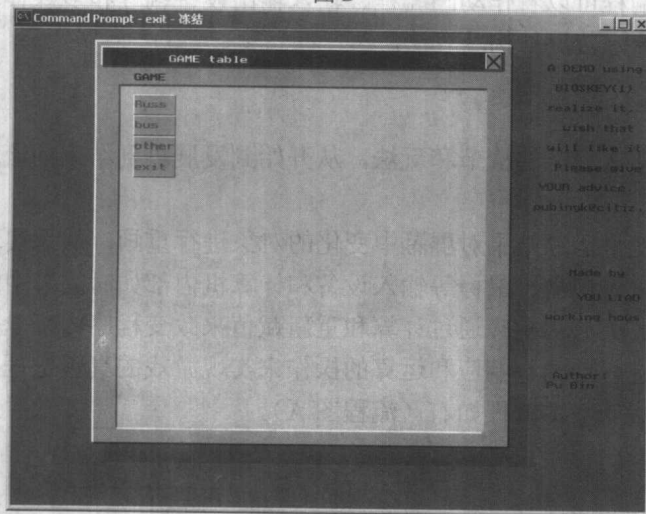


图 C

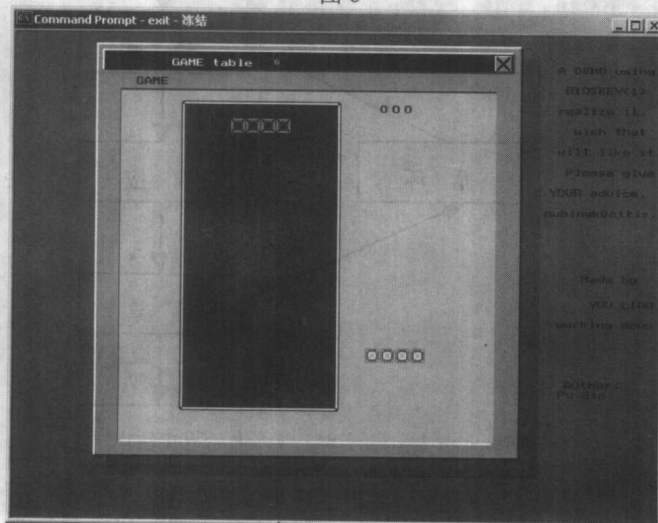


图 D

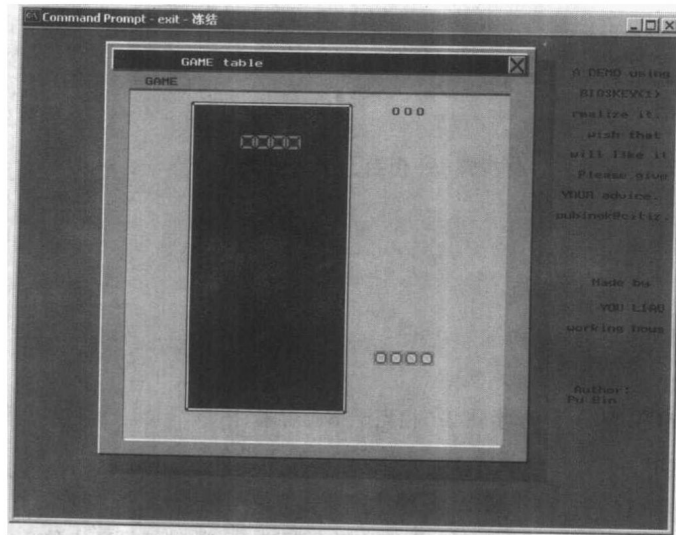


图 E

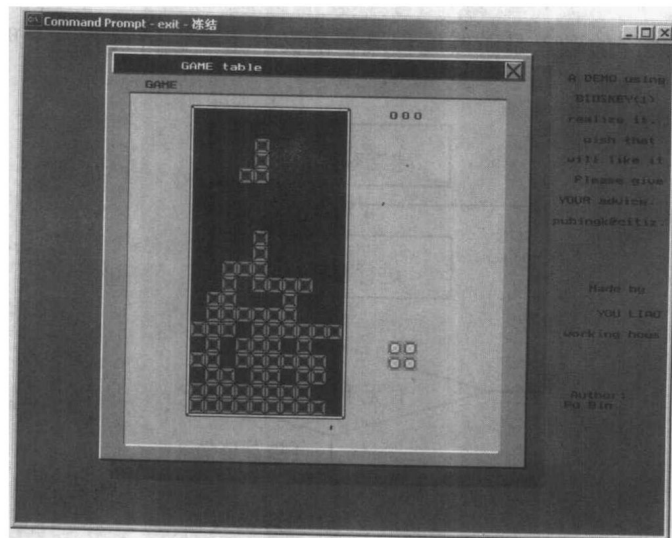


图 F

- (1) 游戏开始，初始设置（这个过程设置了所有的变量初始值）；
- (2) 进行一个方块下落或者消去的动画（这个过程只负责重画屏幕）；
- (3) 然后察看是否有用户发出的旋转、下落、左右移动或退出游戏的指令（这个过程接收信息，同时存入输入变量）；
- (4) 此后如果有用户指令根据其指令计算出块的下一个形状、位置，同时也按照游戏本身设置的下落速度进行叠加计算并且判断是否到达底部，如果到达底部再判断是否可以消去和是否死亡（这个过程改变各种变量的值）；
- (5) 计算完毕后，根据结果进行下一次下落或消去的动画（回到步骤 2），如此循环往复直到游戏结束（这个过程就是一个 while 语句之类的循环）。

对应上面 5 个步骤，再以一个简单的 C 语言伪程序为例：

```
#include...
```



```

#define...
void main(void)
{
int a,b,c;          //步骤 1: 相当于设置初始值
a=1;
b=2;
c=3;
while(a!='q')
{
printf("%d",c); //步骤 2: 相当于重画屏幕
a=getch();      //步骤 3: 相当于响应输入设备
c=a*b;          //步骤 4: 相当于重新运算变量值
}                //步骤 5: 相当于如果 a 不等于 q 就继续循环步骤 2-4
}

```

面向对象编程思路的融入是游戏编程流程的又一重点。再来看一下程序设计，原先的流程可以被理解成结构图 G。

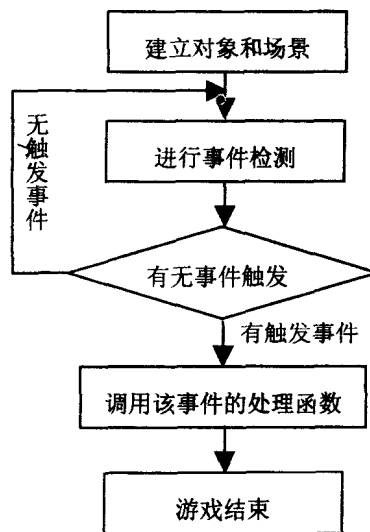


图 G 结构图

这和前面游戏流程并不矛盾，只是换个角度看问题。表 A 是对象思路中的每个环节和游戏流程的四大部分进行对应关系。

表 A 流程与对象

游戏流程	对象思路
动画	对象事件函数中的对象动画处理
响应	事件检测（也有的加上对象函数中的响应部分）
运算	对象事件函数中的运算
循环	事件检测循环

一个是流程明晰，一个是流行的对象化思路，那么到底我们按照哪种思路来编写程序呢？

由于我们使用的是 TC 平台，所以本书的思路是，在流程化语言的基础上加入面向对象思路。这样的好处是能让流程化语言也拥有一定的面向对象思路容易扩充、维护的优点。

本节就是给了你一套写游戏最简单的模版。无论你要写怎样的游戏，都将它分成动画、响应、运算和循环四部分，外加一个初始设置。只要记住给你的那个 C 语言伪程序，套用之后再一步一步改写，一个游戏就出来了。怎么样，解牛的顺序知道了吧！

### 3. 游戏的设计步骤

软件工程课告诉我们任何一个软件的实现都要经过以下几个步骤：

构思：你的创意，具体的内容规划；

设计软件结构：用简洁高效的数据结构实现你的创意；

编写程序：将你的数据结构用程序语言来表达出来；

调试、测试：对程序的系统可用性、模块功能进行测试，检验程序中错误和边缘问题，同时修改程序不符人意的部分；

完善、改进和推出：使游戏更加好玩、更便于操作；

游戏设计事实上也是软件的设计，然而它又有自身的一些特点。比如，游戏强调有极其新颖的创意，要对游戏的每个细节都有准确、详尽的描述和表现等。游戏编程的步骤可以归纳为如下几点。

#### (1) 创意阶段

良好的创意。对创意进行完善，设计出游戏的完整过程并画出流程图，并将游戏分解为若干层次。

对每个层次进行详尽的描述，包括其中的人和物以及每个层次的实现目标。

对每个层次设定规则，包括人物的移动、操作者的权利、游戏的奖励和惩罚以及周边的一切环境、音效和帮助。

#### (2) 规划阶段

使用的图形模式。比如 BIOS 中断 10H 的 13H 模式，即 200\*320 像素 256 色。

确定图形、动画的复杂程度。比如用二维实现还是三维。

考虑使用游戏函数库中的哪些函数来实现游戏创意中的那些要求。

#### (3) 周边准备

制作人物、环境的图像文件。用绘画软件制作\*.JPG 或\*.BMP 文件。

设计人物、环境的声音文件。用录音机制作\*.WAV 文件。

建立地点、发展层次，人、物品、对话和奖励惩罚的数据库文件。例如用 Foxbase、dBase 或者 Foxpro 数据库软件建立\*.DBF 文件。

#### (4) 细部实现

图像实现函数、图像动画函数、声音播放函数、输入装置驱动和功能函数以及存盘、读盘函数。

#### (5) 模块实现

数据结构系统、图像、动画系统、输入/输出系统、人工智能系统、游戏循环系统、用

户界面系统以及声音系统。

#### (6) 整体完善

美工、音效的提高；游戏动画、运算效率的提高；游戏功能和完整性的扩展；游戏的文件大小的缩减。

这只是一个大的方向性原则，具体的内部设计步骤根据不同游戏的情况和个人风格进行增减或调整。然而关键的是要编写一个好的游戏一定要按照软件工程的步骤，从构思到结构到编程一步一个脚印；切不可贪快——一上来就写，到头来程序写到一半写不下去，前功尽弃。好的程序员才可能成为好的游戏编程人员。庖丁要顺骨解牛，不可乱来！

# 目 录

<b>第 1 章 猜数字游戏</b> .....	1	<b>第 5 章 图形模式</b> .....	44
1.1 游戏创意.....	1	5.1 显示适配器与显示模式.....	44
1.2 游戏规划.....	2	5.1.1 显示适配器.....	44
1.3 程序实现.....	4	5.1.2 显示模式.....	45
1.4 游戏调试.....	6	5.2 图形模式 13H.....	45
1.5 文本模式游戏制作.....	8	5.3 调用 BIOS 中断 10H.....	47
1.5.1 文本窗口函数.....	9	5.4 用汇编设置模式.....	47
1.5.2 INT10 中断功能.....	11	5.4.1 使用汇编文件.....	47
1.6 本章小结.....	11	5.4.2 行内汇编.....	48
<b>第 2 章 用 C 语言函数库画图</b> .....	12	5.5 本章小结.....	49
2.1 设置图形模式.....	12	<b>第 6 章 二维图形</b> .....	51
2.2 在图形模式下绘图.....	14	6.1 基本图形.....	51
2.2.1 点.....	14	6.1.1 直接写屏.....	51
2.2.2 线.....	14	6.1.2 直接画点.....	52
2.2.3 填充.....	15	6.1.3 直接画线.....	53
2.3 在图形模式下写字.....	15	6.1.4 直接画多边形.....	56
2.3.1 文本属性设置.....	15	6.2 图形函数优化.....	57
2.4 独立图形程序的建立.....	16	6.3 更多图形.....	59
2.5 赛车的完整图画.....	17	6.4 本章小结.....	63
2.6 本章小结.....	18	<b>第 7 章 中文显示</b> .....	64
<b>第 3 章 简单动画</b> .....	20	7.1 文字显示原理.....	64
3.1 实现动画思路.....	20	7.2 西文显示.....	65
3.2 屏幕保存与恢复.....	22	7.2.1 使用 ROM 字符集.....	65
3.3 重画动画实例.....	24	7.2.2 使用西文字库.....	67
3.4 简单动画实现.....	26	7.3 中文平台下文字显示.....	69
3.5 用异或实现赛车动画.....	29	7.3.1 汉字显示方法.....	69
3.6 本章小结.....	32	7.3.2 中文平台判别.....	69
<b>第 4 章 简单图形游戏</b> .....	33	7.4 西文平台下中文调用.....	70
4.1 从动画到游戏.....	33	7.4.1 hzk16 中文字库文件.....	70
4.2 简单用户响应.....	34	7.4.2 hzk24 中西文共显.....	73
4.3 接收用户信息.....	36	7.5 小字库、无字库技术.....	73
4.4 配上其它东西.....	39	7.5.1 小字库技术.....	74
4.4.1 配上声音.....	39	7.5.2 无字库技术.....	80
4.4.2 加入片头和片尾.....	39	7.6 中文特效.....	83
4.4.3 使用随机数.....	41	7.6.1 多字体显示.....	83
4.5 赛车游戏.....	42	7.6.2 文字格式显示.....	83
4.6 本章小结.....	43	7.7 本章小结.....	84

<b>第 8 章 图形文件</b> .....	<b>85</b>	11.2 游戏进度文件 .....	155
8.1 bmp 文件调用 .....	85	11.2.1 两种方法 .....	156
8.1.1 bmp 文件结构 .....	86	11.2.2 保存进度文件 .....	157
8.1.2 256 色 bmp 文件显示 .....	88	11.2.3 读取进度文件 .....	159
8.2 pcx 文件调用 .....	93	11.3 游戏数据文件 .....	161
8.2.1 pcx 文件结构和编码 .....	93	11.4 dbf 文件 .....	163
8.2.2 pcx 文件显示 .....	96	11.4.1 dbf 文件结构 .....	163
8.2.3 播放 pcx 文件 .....	98	11.4.2 dbf 文件读取 .....	164
8.3 ico 文件显示 .....	99	11.5 本章小结 .....	165
8.3.1 ico 文件结构 .....	99	<b>第 12 章 声音技术</b> .....	<b>166</b>
8.3.2 ico 文件显示 .....	102	12.1 PC 喇叭发声 .....	166
8.4 本章小结 .....	105	12.1.1 发声系统 .....	166
<b>第 9 章 动画原理</b> .....	<b>106</b>	12.1.2 PC 喇叭播放歌曲 .....	167
9.1 动画技术分类 .....	106	12.1.3 扬声器背景音乐 .....	168
9.2 重画技术 .....	107	12.2 声卡技术 .....	169
9.2.1 直接重画 .....	107	12.2.1 DSP 简介 .....	169
9.2.2 缓冲技术 .....	108	12.2.2 DSP 端口寻找 .....	170
9.3 异或技术 .....	110	12.2.3 写 DSP .....	170
9.4 调色板技术 .....	112	12.3 播放 wav 文件 .....	171
9.4.1 调色板寄存器 .....	112	12.3.1 WAV 文件格式 .....	171
9.4.2 调色板动画原理 .....	115	12.3.2 WAV 文件播放 .....	172
9.4.3 调色板动画举例 .....	116	12.4 WAV 背景音乐 .....	173
9.5 拉屏技术 .....	120	12.5 本章小结 .....	173
9.6 适用环境和效率 .....	125	<b>第 13 章 内存技术</b> .....	<b>175</b>
9.7 本章小结 .....	127	13.1 常规内存 .....	175
<b>第 10 章 子画面技术</b> .....	<b>129</b>	13.2 内存结构 .....	176
10.1 子画面概述 .....	129	13.3 XMS 技术 .....	177
10.1.1 子画面 .....	129	13.3.1 XMS 基本知识 .....	177
10.1.2 子画面结构 .....	131	13.3.2 XMS 基本函数 .....	177
10.1.3 面向对象 .....	133	13.3.3 XMS 调用基本程序 .....	177
10.2 显示子画面 .....	135	13.3.4 将中文字库调入 XMS .....	178
10.3 子画面运动 .....	140	13.4 EMS 技术 .....	180
10.4 背景问题 .....	141	13.4.1 EMS 基本知识 .....	180
10.5 子画面游戏 .....	144	13.4.2 EMS 调用基本程序 .....	181
10.6 子画面绘制 .....	151	13.4.3 将中文字库调入 EMS .....	183
10.7 本章小结 .....	152	13.4.4 全方位拉屏 .....	184
<b>第 11 章 文件操作</b> .....	<b>154</b>	13.5 本章小结 .....	196
11.1 文件基本操作 .....	154	<b>第 14 章 接口技术</b> .....	<b>197</b>
11.1.1 建立、打开和关闭 .....	154	14.1 键盘 .....	197
11.1.2 读取和写入 .....	155	14.1.1 键盘读取 .....	197

14.1.2 同时按下问题.....	200	16.4 各类游戏编程思路.....	256
14.1.3 模拟按键.....	201	16.4.1 桌面游戏编程思路.....	256
14.1.4 清空键盘缓冲.....	201	16.4.2 视频对战游戏编程思路.....	257
14.2 鼠标.....	202	16.4.3 魂斗罗类游戏编程思路.....	259
14.2.1 鼠标基本函数.....	202	16.4.4 玛丽、赛车类游戏编程思路.....	260
14.2.2 改变鼠标形状.....	204	16.4.5 RPG 游戏编程思路.....	262
14.2.3 用 pcx 图像做鼠标.....	209	16.5 本章小结.....	263
14.3 串口.....	211	<b>第 17 章 游戏例程.....</b>	<b>265</b>
14.3.1 串口基础.....	211	17.1 建立通用游戏函数库.....	265
14.3.2 利用串口传输文件.....	216	17.2 游戏创意.....	265
14.3.3 两机坦克对打例程.....	219	17.3 游戏规划.....	268
14.4 本章小结.....	226	17.3.1 详细设计.....	268
<b>第 15 章 界面技术.....</b>	<b>228</b>	17.3.2 程序流程设计.....	271
15.1 界面对象的结构.....	228	17.4 程序编写.....	272
15.1.1 对象的结构分析.....	228	17.4.1 文件清单.....	272
15.1.2 对象的初始化.....	230	17.4.2 进度文件.....	272
15.1.3 界面设计与分析.....	231	17.4.3 图片文件.....	272
15.2 对象绘制函数.....	232	17.4.4 数据文件.....	273
15.2.1 填充矩形绘制函数.....	232	17.4.5 代码文件.....	276
15.2.2 立体按钮绘制.....	233	17.5 游戏场景.....	276
15.2.3 窗体、按钮和菜单绘制.....	233	17.6 本章小结.....	281
15.3 使用链表.....	238	<b>附录 A 游戏函数库.....</b>	<b>282</b>
15.4 对象事件函数.....	239	<b>附录 B 简单数据库.....</b>	<b>297</b>
15.4.1 按钮的基本动作.....	239	B.1 数据库要求.....	297
15.4.2 菜单的基本动作.....	240	B.2 详细设计.....	298
15.5 进行事件检测.....	243	B.3 模块设计.....	300
15.6 界面例程.....	244	B.3.1 输入.....	300
15.7 游戏实例.....	245	B.3.2 检查.....	303
15.7.1 DOS 游戏界面设计.....	245	B.3.3 显示.....	307
15.7.2 将界面插入游戏.....	247	B.3.4 删除.....	309
15.7.3 构建个性化界面.....	250	B.3.5 插入.....	317
15.8 本章小结.....	251	B.3.6 查找.....	322
<b>第 16 章 其他问题.....</b>	<b>252</b>	B.3.7 修改.....	329
16.1 TSR 驻留.....	252	B.3.8 排序和交换节点.....	337
16.1.1 TSR 基本知识.....	252	B.3.9 保存.....	343
16.1.2 时钟驻留.....	253	B.3.10 读取.....	344
16.1.3 热键驻留.....	253	B.3.11 清空.....	346
16.2 简单病毒.....	253	B.4 程序代码.....	346
16.3 OOP 应用.....	254	B.5 通用数据库设计.....	397

# 第 1 章 猜数字游戏

## 本章导读

猜数字的游戏规则是，电脑在 0~9 这 10 个数字中，任意不重复地选择四个排列成四位数，然后让玩的人猜使用的是哪四个数字和数字在第几位；A 代表数字对了但位置不对；B 表示数字和位置都对了，A、B 前的数字表示处于两种情况下的数字个数；举例，电脑给你猜的数字是 2943，你猜 2893，电脑就显示 2A1B。

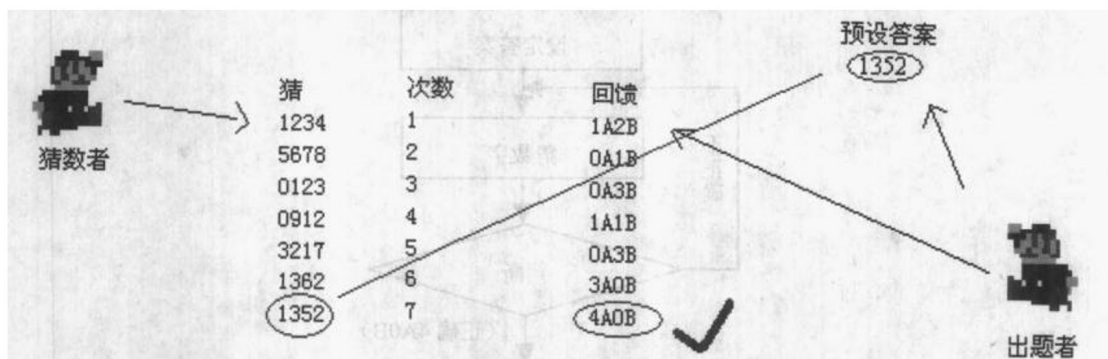
## 本章重点

通过实例了解游戏设计的完整过程：

- (1) 游戏创意、游戏规划、程序实现、游戏调试；
- (2) 使用文本窗口函数和 INT10 中断（视频服务程序）来帮助文本游戏制作。

## 1.1 游戏创意

首先我们来模仿以下游戏的进行过程，如图 1-1 所示。



如果 1 个数字对了但位置不对就用 1B 表示，如果 1 个数字和位置都对了就用 1A 表示

图 1-1 猜数字游戏示意图

前提：

- (1) 两个游戏者：一个出题（以下称为出题者），一个解题（以下称为猜数者）；
  - (2) 两张白纸：一张写答案数字（给出题者写），一张写猜测的数字（给猜数者写）；
- 出题者预设答案：1352

猜数过程见表 1-1。

表 1-1 猜数字游戏过程

次数	猜（猜数者）	回馈（出题者）
1	1234	1A2B
2	5678	0A1B

(续表)

次数	猜 (猜数者)	回馈 (出题者)
3	0123	0A3B
4	0912	1A1B
5	3217	0A3B
6	1362	3A0B
7	1352	4A0B (正确)

事实上猜数者是根据每次出题者给出的反馈来猜测下一个数字,从而越来越靠近答案。而猜数者使用的次数越少则猜数循环次数越少,积分越高。

猜数字:出题者在0~9中任选不重复四个数排列成四位数,然后让猜数者每次通过从0~9中任选四个数字来猜使用的是哪四个数字和这四个数字各在哪个位置?如果1个数字对了但位置不对就用1B表示,如果1个数字和位置都对了就用1A表示;比如,给你猜的数字是2943,你猜2893,回答是2A1B,因为2和3的数字和位置都对了,而9的数字对了位置没有对。这样循环往复,在每次猜数和猜的正确性信息基础上,得出最终答案。看谁猜的次数最少谁就积分越高。

通常,出题者会在一张小纸片上写好他想给别人猜的数字。而猜数者在另一张纸上写上每次的答案并给出题者看,出题者把每次猜的情况回馈在答案后面。如此循环往复直到得出正确答案。游戏进行流程如图1-2所示。

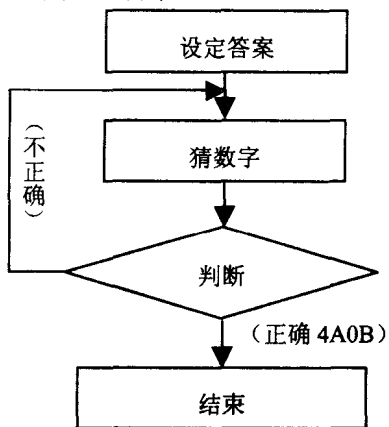


图 1-2 猜数字执行流程

由此可见,创意阶段说到底就是搞清楚到底要做一个什么游戏和这个游戏最基本的进展流程。比如这个游戏用口授教别人如何玩并不很难,可使用文字表达真的不容易,倘若我们连自己要做什么游戏都不是非常清楚,那么到真写程序的时候一定会出很多问题。事实上对游戏运行的流程描述,一方面进一步帮助我们了解游戏规则,另一方面也为今后的程序设计流程作了铺垫。

## 1.2 游戏规划

由于当时我写这个游戏时候的情况和大家差不多,都是一个编程初学者。于是,我只



可能选择用文本模式；并且由于当时的能力所限，又决定不使用任何例如图形、菜单之类的界面支持；换句话说，当时我的规划很简单，就是将这个游戏用最简朴但仍然易懂的方法表达出来。

此外，为了尽量简化游戏编写难度我还对游戏作了以下调整：

- (1) 由程序固定被猜数字；
- (2) 由玩者给出猜测答案，程序给出反馈；
- (3) 不包括由玩者出题；
- (4) 暂不进行非数字和非四位数字的判别等周边工作；
- (5) 不提供任意终止游戏的功能，必须将游戏进行到底；
- (6) 不使用子函数（当时也不会写）；

这样一来，所要做的那部分程序更加清楚了，对游戏的编写进行一个简单的、不全面的分析之后，我当时列出了以下几个难点：

- (1) 几 A 几 B 的判断问题之算法；
- (2) 玩者所猜数字从键盘输入的函数调用；
- (3) 游戏循环出口的设定；
- (4) 被猜数字随机数产生的函数调用；（可选）

针对以上 3 个难点，我在思路中给出了一点解决方案：

(1) 对于几 A 几 B 问题的算法实现，我以为不会非常简单，必须到程序中不断调试才可能完善。基本思路是，将给出的被猜数字变量的每个位和每次玩者输入的猜测答案变量的每个位进行一一比较，结合给双方各设定一个当前位标志给出每次 A、B 的数量。具体如下图 1-3 所示。

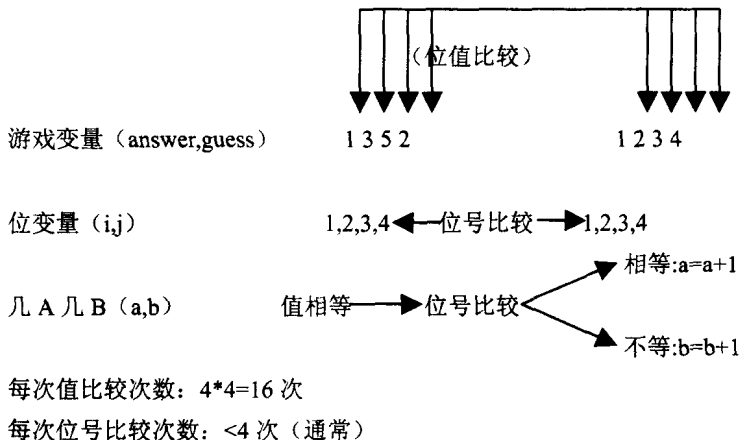


图 1-3 数字循环与判断

(2) 对于输入函数的选用，一个是 `getchar` 函数（字符输入函数），一个是 `scanf` 函数（格式输入函数）。这里我们要的是一个数字，当然使用允许返回数字类型的 `scanf` 函数。

(3) 游戏循环出口的问题，其实可以看作 `while` 语句中设定什么表达式的问题。我之前已经忽略了“任意终止游戏的工作”，于是这里只要判断玩者是否猜对了答案就可以了。问题一下子便简单了，因为猜对答案就意味着 4A0B 的情况，也就是说那个变量 `a=4` 就