

趣味电脑游戏程序

王颂赞 费德宝 编著 科学技术文献出版社



趣味电脑游戏程序

王颂赞 费德宝 编著

科学技术文献出版社

(京)新登字130号

内 容 简 介

本书通过一系列Apple电脑游戏程序设计思想的剖析,由浅入深地介绍各种绘画方法、制作动画的技巧,以及编制图文并茂、生动活泼、丰富多采的计算机游戏程序的方法。

本书适合学过BASIC语言的电脑爱好者阅读、参考。

趣味电脑游戏程序

王颂赞 费德宝 编著

科学技术文献出版社出版

(北京复兴路15号 邮政编码100038)

北京市燕山联营印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

787×1032毫米 32开本 5.125印张 110千字

1992年9月第1版 1992年9月第1次印刷

印数:1—5 200册

科技新书目:273—119

ISBN 7-5023-1704-X/TP·91

定 价: 3.70元

前 言

初次接触计算机的青少年，常常被那变幻莫测、富有情趣的计算机游戏所吸引，沉湎于游戏之中，流连忘返，乐此不疲，因而耗费了大量的时间与精力。兴趣是入门的向导，在计算机知识的普及教育活动中，若能因势利导，以编制游戏程序为活动内容，不仅能使青少年的游戏兴趣得以满足，而且，游戏程序中涉及到的动画、音响及程序控制的技巧与方法，将触发青少年丰富的想象力和创造力，将引导他们去设计更富有智慧型的计算机程序，从而使他们从兴趣走向专业，从业余爱好者行进到计算机专门人才的行列之中。

本书以青少年为读者对象，从简单的游戏程序开始，由浅入深，介绍各种绘画方法、动画技巧、游戏程序的设计方法。由于将编制游戏程序作为计算机活动的中级内容，因此要具有基础的程序设计能力，这是以学会了基本BASIC语言及掌握了APPLE计算机操作技能为起点的。而书中很多程序是用汇编语言编写的，因而，涉及基本的6502汇编语言的程序设计基础。

何冰钢、王强、张中杰、周晓文、章子凌等同学为本书编制打印了部分程序并对原有程序作了有益的改进。在此，我们致以谢意。

目 录

前 言

第一章 简单的游戏程序	(1)
第一节 从猜数开始.....	(1)
第二节 剪刀、石头和布.....	(5)
第三节 激光炮打飞机.....	(7)
第四节 猫捉老鼠.....	(9)
第五节 蛇吃蛋.....	(16)
第二章 用低精度绘图做的动画	(25)
第一节 参考点绘图法.....	(25)
第二节 利用监控子程序绘图.....	(29)
第三章 用高精度绘图做的动画	(38)
第一节 高精度图形的移动.....	(38)
第二节 两页图画的切换.....	(39)
第三节 造型表.....	(42)
第四节 APPLESOFT中的绘图子程序.....	(48)
第五节 H PLOT造型表及动画.....	(50)
第六节 飞吧,小飞机! 飞吧,大鸟!.....	(56)
第四章 动画技术	(71)
第一节 图形元动画.....	(71)
第二节 预移图序动画.....	(80)
第三节 背景旋转动画.....	(91)

第五章 编一个更好的游戏程序	(98)
第一节 游戏中的键控.....	(98)
第二节 游戏摇杆.....	(102)
第三节 音响效果.....	(106)
第四节 碰撞与爆炸.....	(112)
第五节 《打击入侵者》.....	(114)
第六章 利用工具软件制作动画	(130)
第一节 TAKE1的起动及通用键.....	(131)
第二节 画面和背景的制作.....	(133)
第三节 角色文件的制作和镜头的编辑.....	(139)
第四节 动画片的完成.....	(145)
第五节 TKDOS的使用	(148)
第六节 利用TKDOS编写动画	(152)

第一章 简单的游戏程序

第一节 从猜数开始

本书要向青少年读者介绍许多编制图文并茂、生动活泼、丰富多采的计算机游戏程序的方法。但是，万丈高楼平地起，总得从最简单的开始。这里介绍的猜数程序，也许可从诸多BASIC课本中找到，但它确实能体现一些计算机游戏的基本思想，就让我们从这里开始。

最简单的猜数程序的框图如图1.1。

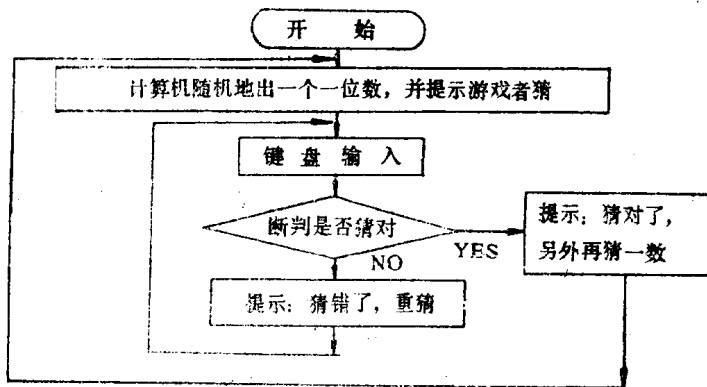


图 1.1

在这个程序中，仅用了一个判断语句来判断是否猜中，因此是最最简单的了。游戏者只需依照某种次序（从大到小或从小到大）一一猜测，最终总会猜着，而且猜对猜错一个样。

程序 1.1

```
10 PRINT "请你猜一个一位整数!"
20 A=INT(RND(1)*10):N=0
30 INPUT B:N=N+1
40 PRINT
50 IF A=B THEN 120
60 IF N=3 THEN 160
70 IF B<A THEN 100
80 PRINT "太大了!猜个小点的数。"
90 GOTO 30
100 PRINT "太小了!猜个大点的数。"
110 GOTO 30
120 IF N=1 THEN PRINT "一猜就中":GOTO 150
130 IF N=2 THEN PRINT "二次猜中":GOTO 150
140 PRINT "第三次才猜中"
150 PRINT "再猜一个":GOTO 20
160 PRINT "三次猜不中, 另猜一个":GOTO 20
```

这就使游戏者的兴趣大为减退, 玩上二次也就不再玩了。如果在程序中适当增加些限制(例如一个数字最多猜三次), 而且对猜对不同次数者予以不同的评价(给以不同的奖分), 在每次猜错后, 计算机还要提示一下, 让你有个方向, 这就大大

提高了游戏者的兴趣。你看到的程序1.1就是这样一个程序。

读者可以在上述程序基础上再加些花样, 如规定猜五个数为一局, 每猜一个数根据所用次数的多少记上不同的分数, 一局结束给出总分。这样就可以几个人轮流玩, 比一比谁的总分最高。下面是一个猜数程序。在这个游戏中, 计算机用0—9之间三个互不相同的数字组成一个三位数让你来猜, 一共可猜20次。每猜1次, 计算机会提示你猜得的正确程度:

说明 在APPLE机中还有一种产生随机数的方法: 由于在\$4E, \$4F(十进制为78, 79)两个内存单元中存放着的数与最近两次击键的时间间隔有关, 由于两次击键之间间隔是随机的, 所以其中保留的数实际上是随机的。我们可用下列程序来验证: 10 PRINT 256*PEEK(79)+PEEK(78):GET A\$:GOTO 10

(1) 若猜中一个数字但位置不对, 计算机显示出一个“/”;
 (2) 如果某个数字的大小和位置都对了, 计算机显示出一个“*”;
 (3) 如果一个数字也没对, 计算机显示出一个“×”。

于是, 游戏者可以根据计算机给出的信息, 推想出正确的结果的一部分, 多猜几次就会逐渐与正确答案渐渐接近, 以求得最终猜对。由于每猜1次要扣去5分, 因此要求越早猜中越好。可以用数学方法证明, 如果你掌握了比较好的方法, 最多用8次就可猜中, 因此积分应大于60, 假如你不动脑筋瞎猜, 那就会不及格, 甚至连一分都没有。

在程序中〔10〕—〔45〕* 由计算机出一个三位数。〔110〕—

```

5 REM pro1-2          程序 1.2
10 FOR I=1 TO 3
20 A(I)=INT(10*RND(1))
25 IF I=1 THEN 45
30 FOR J=1 TO I-1 : IF A(I)=A(J) THEN 20
40 NEXT J
45 NEXT I
50 PRINT "我有一个三位数让你猜"
60 FOR I=1 TO 20
65 PRINT "第";I;"次猜数";
70 INPUT A$
75 IF LEN(A$)<>3 THEN 65
80 FOR J=1 TO 3
85 B(J)=ASC(MID$(A$,J,1))-48
90 NEXT J
100 C=0:D=0: REM C 记录大小对位置不对的数字个数
      D 记录大小位置全对的数字个数
  
```

* 〔10〕表示第10语句, 〔10〕—〔45〕表示10至45语句, 下同。

```

110 FOR J=1 TO 2
115 IF A(J)=B(J+1) THEN C=C+1
120 IF A(J+1)=B(J) THEN C=C+1
125 NEXT J
130 IF A(1)=B(3) THEN C=C+1
135 IF A(3)=B(1) THEN C=C+1
140 FOR J=1 TO 3
145 IF A(J)=B(J) THEN D=D+1
150 NEXT J
160 IF D=3 THEN 260
170 IF C=0 THEN 200
180 FOR J=1 TO C
185 PRINT "/";
190 NEXT J
200 IF D=0 THEN 230
210 FOR J=1 TO D
215 PRINT"*";
220 NEXT J
230 IF C+D=0 THEN PRINT"x"
235 PRINT
240 NEXT I
250 PRINT "次数超过. 这个数是: ";A(1);A(2);A(3)
255 GOTO 270
260 PRINT "猜对了"
265 PRINT "得分: ";105-5*I
270 END

```

[150]对输入的数字进行分析,判断猜对了多少。[160]—
[230]输出信息处理。[250]—[265]总结。

第二节 剪刀、石头和布

下面我们再介绍大家常玩的剪刀、石头和布相克的游戏程序(见程序 1.3),从而谈谈游戏的智能化问题。

程序中用 1 表示布, 2 表示剪刀, 3 表示石头。那么 3 胜 2, 2 胜 1, 1 胜 3。〔60〕随机地出计算机想出的花样, 记入 A, 〔80〕由游戏者输入希望出的花样, 记入 B。〔90〕—〔140〕是判断输赢的处理, 以 X 记计算机胜的次数, Y 记游戏者胜的次数。每盘

程序 1.3

```
10 REM prot-3
20 M$(1)="布":M$(2)="剪刀":M$(3)="石头"
30 FOR I=1 TO 3
40 NEXT I
50 FOR J=1 TO 10:PRINT
60 A= INT(RND(1)*3)+1
70 PRINT "我已出好了,请你出"
80 INPUT B:IF B<0 OR B>3 THEN 80
90 A1=A:B1=B:IF A=B THEN 210
100 IF ABS(A-B)=2 THEN 140
110 C=A
120 A=B:B=C
140 IF A>B THEN 180
150 Y=Y+1
180 PRINT "我胜了,你出";M$(A1)
165 PRINT "        你出";M$(B1)
170 GOTO 220
180 X=X+1
190 PRINT "你胜了,我出";M$(A1)
195 PRINT "        你出";M$(B1)
200 GOTO 220
210 PRINT "平手,大家都是";M$(A)
220 NEXT
230 PRINT "你的总分是:";X
240 PRINT "我的总分是:";Y
250 END
```

共10次，10次完成给出总分。

由于整个程序都是随机进行的，计算机和游戏者都是盲目地出数，大家碰运气获胜。换句话说，这种游戏是没有任何智力因素的，所以说这是一种低层次的游戏，玩几次也就乏味了。下面介绍有简单智能的剪刀、石头、布游戏程序的编程方法：

游戏者在石头、剪刀、布三者中选一，而计算机独立地作出自己的选择（在这方面应是绝对公平的）。但计算机记住游戏者每次选择的动向，并把它与游戏者的前1次的行动联系起来。这样，计算机就能揣摩游戏者的心理，从而考虑自己的对策。在计算机内程序为自己建立一张表格（用 3×3 数组构成），称之为目标表。计算机把游戏者每种选择紧跟着选哪一种的次数全记录下来，轮到计算机出数时，先看一下游戏者上次出的什么，再查一下表格看其下次最可能出什么，从而作出选择。这样，游戏进行了足够次后，计算机取胜的可能性也会大起来。

表 1.1 紧跟着的选择

前次选择		石头	剪刀	布
	石 头			
	剪 刀			
	布			

当然，大多数游戏者会意识到这一点，并随机地改变自

己的策略，这样就不会总是败给计算机了。为了让计算机有更多的得分机会，可以建立更大的数组，使计算机记住的不只是前1次而是前2次、3次甚至更多次游戏过程，从中找出游戏者的规律。至于如何建立表格，计算机又如何从这些表格中找到规律，需要涉及到“人工智能”的问题，它已超出了我们这本书的范围。因为我们打算着重讨论的是如何设计一个生动、高速、有趣的电子游戏程序。

第三节 激光炮打飞机

本节介绍的“激光炮打飞机”游戏，利用键盘上现有的字符来模拟飞碟、炮和炮弹：#代表飞碟，Y代表激光炮，I表示激光炮射击之炮弹。由于游戏的画面是由一些字符组成的，所以称之为TEXT游戏。我们只要学会使用HTAB，VTAB，PRINT等语句，即可达到定位打印字符的目的。

游戏一开始，屏幕上会出现一座激光炮和一个不断改变位置的飞碟，我们可以通过按进格键和退格键来移动激光炮的位置，可以通过按空格键来命令向飞碟开炮。程序还会记录发射子弹之多少，击中飞碟游戏即告结束。可以几个人轮流玩这个游戏，比一比谁能用最少的子弹就能击中飞碟。见程序1.4。

在程序中，用变量A来记录飞碟的所在行，B、C记录飞碟所在列。I记录激光炮所在行，J、K记录激光炮所在列。W记录炮弹所在列，M记录射击次数。

〔20〕画出一条界线，作为炮兵阵地，并对飞碟、激光炮

程序 1.4

ILIST

```

10 HOME : VTAB 14: HTAB 10: PRINT
   "HIT ANYKEY TO BEGIN!": WAIT
   - 16384,120: POKE - 16368,
   0
20 CALL - 936: FOR I = 15 TO 35
   : VTAB 23: HTAB 1: PRINT "="
   : NEXT I: A = 2: C = 2: B = 25:
   J = 25: I = 22: K = 2: M = 0: Q =
   0
30 P = PEEK ( - 16384): POKE -
   16368,0
40 VTAB A: HTAB C: PRINT " ": A =
   INT ( RND (1) * 9 + 11): VTAB
   A: HTAB B: PRINT "#": C = B: VTAB
   I: HTAB K: PRINT "Y"
50 IF K < > J THEN VTAB I: HTAB
   K: PRINT " ": VTAB I: HTAB J
   : PRINT "Y"
60 B = B + ( RND (1) > 0.5 AND B <
   30) - ( RND (1) < 0.5 AND B >
   20): K = J: W = 21: VTAB 2: HTAB
   37: Q = Q + 1: PRINT Q
70 IF P < 127 THEN 30
80 J = J + (P = 149 AND J < 30) -
   (P = 136 AND J > 20): IF P <
   > 160 THEN 30
90 W = W - 1: IF W < > A + 1 THEN
   FOR N = 1 TO 3: VTAB W: HTAB
   J: PRINT "I": NEXT N: VTAB W
   : HTAB J: PRINT " ": GOTO 90

100 M = M + 1: VTAB 1: HTAB 38: PRINT
   M
110 IF J = B THEN PRINT CHR$ (
   7): FOR L = 1 TO 350: NEXT L
   : HOME : VTAB 12: HTAB 17: FLASH
   : PRINT "GAME OVER": NORMAL
   : WAIT - 16384,128: END

120 GOTO 30

```

1

的初始位置作了设定。〔30〕接受键盘信息。这是因为在-16384单元中，存放着最后一次被击键的代码。〔30〕就把该键盘信息记入变量P中，以备下面程序查用。

〔40〕〔50〕在屏幕上显示出字符#以示飞碟，显示出字符Y以示激光炮。〔60〕是飞碟坐标的变换公式。当由RND(1)得到的随机数大于0.5时让飞碟向右移 ($B = B + 1$)；反之，让飞碟向左移 ($B = B - 1$)。〔80〕是激光炮位置更换公式，它根据从键盘上得到的移位信息，计算出激光炮的新坐标。〔90〕是开炮语句，它用字符I表示炮弹而飞向飞碟。其他一些语句的功能是不难理解的，只要读者把程序输入计算机且执行几遍，再回来读读程序就可以明白了。

第四节 猫 捉 老 鼠

“猫捉老鼠”是一个人机对抗的游戏。在低精度绘图形式下，我们用黄颜色的小方块铺成可通行的道路，用紫色小方块表示沿途的粮食。计算机会指挥代表猫的小方块向代表老鼠的小方块逼近，而游戏者可以利用I、J、K、L四个键控制老鼠的逃跑方向。游戏者应尽量控制老鼠向有粮食的位置逃去，这样老鼠就能边吃边逃。当老鼠吃满400粒粮食就算逃逸了猫的追逐，游戏者取得了胜利。一个老鼠被抓住以后又会出现一只新的老鼠。程序中有五只老鼠，五鼠全灭则全局结束。在屏幕的下方标出现有老鼠数和老鼠吃到的粮食数供游戏者参考。

程序的大致框图如图1.2。

在介绍程序前，必须有二个 问题 要弄清楚：(1) 如何

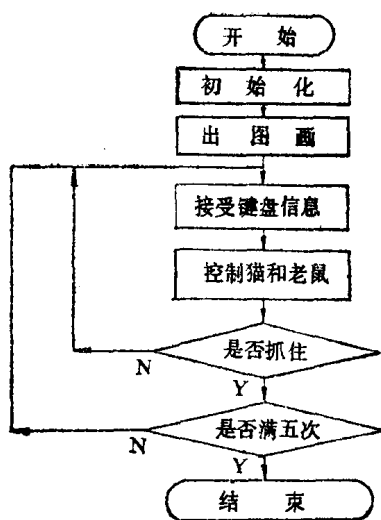


图 1.2

让代表猫和鼠的亮点在屏幕上动起来，还要考虑到不能越出屏幕所规定的范围；(2) 猫是如何逼近老鼠的。

大家知道，在低精度绘图状态下，屏幕被分成 40×40 个色点，左上角为原点，坐标是 $(0, 0)$ ，从左到右为X轴正向，从上向下为Y轴正向（这与数学中的规定不一样）。

假设一开始有一亮点在

$X0 = 20$ ， $Y0 = 20$ 处，那么它有上、下、左、右4个方向*，每个方向坐标的增量表如表1.2所示。

表 1.2

方向	X增量	Y增量
上	0	-1
左	-1	0
下	0	1
右	1	0

我们可以预先把这个增量表放在一个称之为工作数组的两维数组 $W(4, 2)$ 中，见表1.3。

* 实际上有上、下、左、右、左上、左下、右上、右下8个方向，读者可按所述原理自己加以扩充。

表 1.3

W	1	2	3	4
1	0	-1	0	1
2	-1	0	1	0

然后再按 键盘输入之信息(I,J,K,L之一)让亮点坐标加上相应的增量,这个亮点就随之动了起来: I→向上, J→向左, K→向下, L→向右。读者不妨试验一下程序1.5。

程序 1.5

```
1LIST
```

```
10 DATA 0,-1,-1,0,0,1,1,0
20 FOR I = 1 TO 4
30 READ W(I,1),W(I,2)
40 NEXT I
50 GR :X0 = 20:Y0 = 20
60 COLOR= 7: PLOT X0,Y0
70 X1 = X0:Y1 = Y0
80 Z = PEEK ( - 16384) - 200
90 IF Z < 1 OR Z > 4 THEN 110
100 X0 = X0 + W(Z,1):Y0 = Y0 + W(
    Z,2)
110 :
120 COLOR= 0: PLOT X1,Y1
130 GOTO 60
```

```
1
```

在执行程序1.5时, 你总按某一个键, 那么被你控制的亮点就向一个固定的方向走去, 很快就使坐标值超出了计算机所允许的范围而死机。为了避免这一情况, 我们在亮点运动之前先判断一下亮点下一步是否会出界。如果将要出界就