SPOT

The Society of PET Owners and Trainers

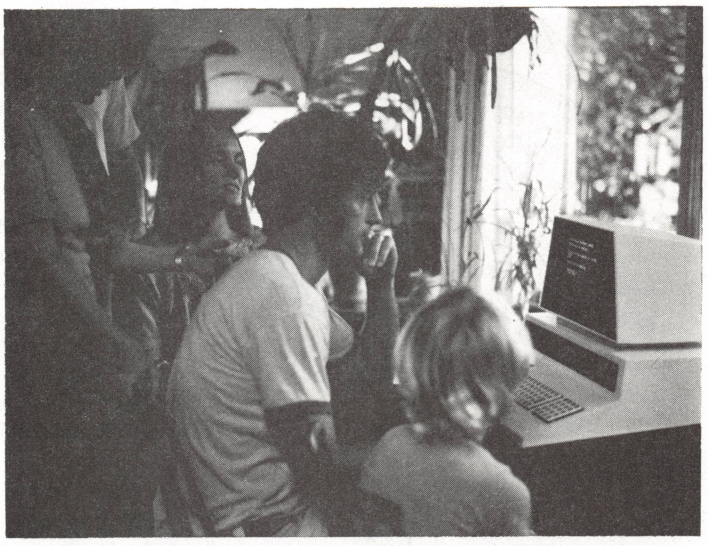


Photo courtesy of Visualscope.

PET SOCIETIES & NEWSLETTERS

Here's additional information about PET users' groups and newsletters. In the San Francisco Bay area, those interested in the East Bay's SPHINX society should contact Neil Bussey (415) 451-6364. Those in the San Jose-San Francisco area should call the Palo Alto Mr Calculator store at (415) 328-0740 for details on the next local users' group meeting.

Great news: the newsletter from the Bay Area groups is now available. It's packed with info that's available nowhere else. The most recent issue, for example, contained articles on a simple way to add a standard keyboard to the PET (while retaining PET graphics), an article on using the PET's 8-bit parallel I/O port, info on the PET's character set, memory map, and lots more, including announcements of many PET-related products. The two back issues are available at $.75 each; $4.50 will get the monthly newsletter for the next six months. Send orders to Pete Rowe, Lawrence Hall of Science, U.C. Berkeley, Berkeley, CA 94720.

THE PET PAPER is being published by Terry Laudereau, formerly Software coordinator for Commodore, and Rick Simpson, KIM Product Manager at MOS Technology, a Commodore company. It's scheduled to include articles to interest both beginners and experts, news of User Groups, software reviews, and hardware how-to's. For a year's subscription (number of issues not specified) send $15 to THE PET PAPER, PO Box 43, Audubon, PA 19407.

SOFTWARE

See our PET software review under 'Reviews'. Many distributors of PET software are springing up. Most offer royalty contracts for programs running from 2% of wholesale to 20% of retail. Many deal in both TRS-80 and PET programs. As of early April, these companies are marketing software:

Don Alan Enterprises, PO Box 401, Marlton, NJ 08053.

Peninsula School Computer Project, Peninsula Way, Menlo Park, CA 94025.

Personal Software, PO Box 136-B4, Cambridge, MA 02183; (617) 783-0694.

Silver State Enterprises, PO Box 27111, Lakewood, CO 80227.

The PET Paper, PO Box 43, Audubon, PA 19407.

As of early April, these companies are gearing up to sell PET software:

Commodore, 901 California Ave, Palo Alto, CA; (415) 326-4000. Contact Adrian Byram.

Creative Computing, PO Box 789-M, Morristown, NJ 07960; (201) 540-0445.

Kilobaud, Peterborough, NH; (603) 924-3873.

Mind's Eye Personal Software, PO Box 354, Palo Alto, CA 94301; (415) 326-4039. (Run by Greg Yob, formerly of Commodore).

TEACHERS' (& KIDS') PET

We would like to communicate with other schools who are using the Commodore PET for educational purposes. We are a small K-10 school. We are currently teaching BASIC to some 7-10 graders and they are using the language to develop programs for their mathematics classes. We would be interested in sharing methods and programs with other schools who are attempting the same sort of thing.

Charles Ebert
The Midwestern Academy of
the New Church
73 Park Drive
Glenview, IL 60025

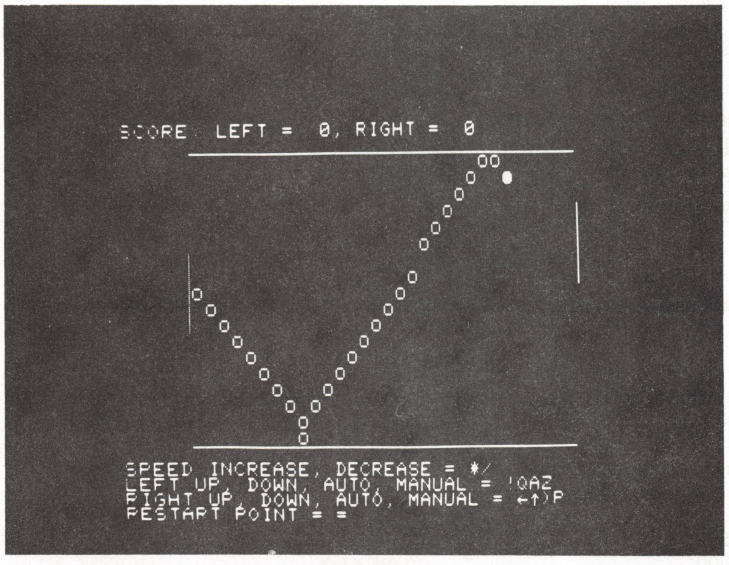
LISTING CONVENTIONS

Program listings employ the following conventions to represent characters that are difficult to print on a standard printer: Whenever square brackets appear in the listing, neither the brackets nor the text they enclose should be typed literally. Instead, the text between the brackets should be translated to keystrokes. For example, [CLR] means type the CLR key, [3 DOWN] means [DOWN, DOWN, DOWN] ie press the first CRSR key three times.

PONG for the PET

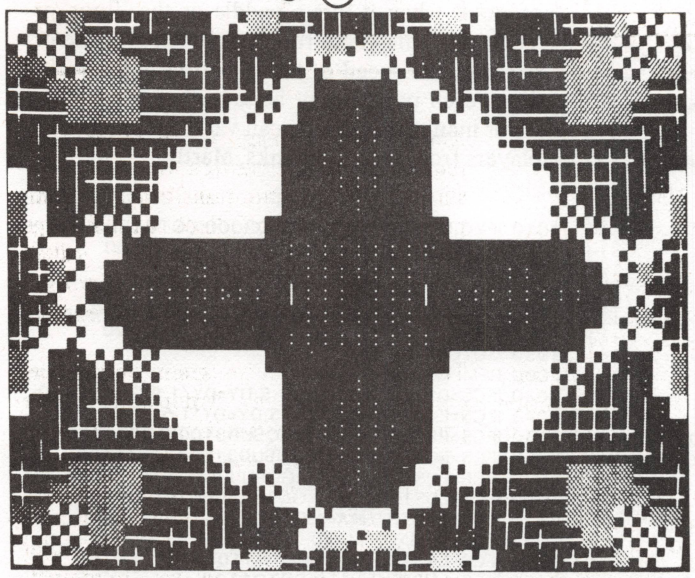
Martin Cohen, of Technology Service Corporation in Santa Monica, CA, has written a fine PONG game for the PET. The ball actually *squashes* when it hits a paddle or the 'floor' or 'ceiling' of the game room. You can increase or decrease the speed of the game to suit yourself. Since each paddle may be set either to automatic or manual mode you can vary the number of players from 0 to 2. Thanks, Martin!

```
500 READ DS$,IS$,LU$,LD$,RU$,RD$,LA$,LM$,RA$,RM$,AG$
510 DATA "/","*","!","Q","←","↑","A","Z",")","P","="
515 Z3=32768:Z4=40
520 X0=5:X1=35
530 Y0=2:Y1=19
540 L0=INT((Y0+Y1)/2)-2:L1=L0+4
550 R0=L0:R1=L1:DM=0
560 PRINT"[CLR]";
570 Y2=Y1+1:Z2=99
580 FORX2=X0TOX1:GOSUB900:NEXTX2
590 Y2=Y0-1:Z2=100
600 FORX2=X0TOX1:GOSUB900:NEXTX2
620 X2=X:Y2=Y
630 XP=X2:YP=Y2
640 GOSUB3000:GOSUB3100
650 LN=0:RN=0
660 DX=1.5:DY=0
670 PRINT"[HOME]";:FORI=1TOY1+2:PRINT"[DOWN]";:NEXTI
680 PRINT"SPEED INCREASE, DECREASE = ";IS$;DS$
690 PRINT"LEFT UP, DOWN, AUTO, MANUAL = ";LU$;LD$;LA$;LM$
700 PRINT"RIGHT UP, DOWN, AUTO, MANUAL = ";RU$;RD$;RA$;RM$
710 PRINT"RESTART POINT = ";AG$;
790 LA=0:RA=0
890 GOTO2400
900 REM PUT Z2 AT (X2,Y2)
910 POKEZ4*Y2+X2+Z3,Z2
920 RETURN
950 REM WAIT TJ SEC
960 TJ=TI+60*TJ
965 IFTI<TJGOTO965
970 RETURN
1000 REM TOP OF LOOP
1005 ZB=81:REM STANDARD BALL
1010 XX=X:YY=Y:X=X+DX:Y=Y+DY
1020 IFX<X0GOTO1200
1030 IFX>X1GOTO1300
1040 IFY<Y0GOTO1400
1050 IFY>Y1GOTO1500
1060 REM TEST FOR KEY HIT
1070 GETC$:IFLEN(C$)>0GOTO2000
1080 IFLAANDDX<0GOTO1800
1090 IFRAANDDX>0GOTO1900
1100 REM DISPLAY AT(X,Y)
1120 XQ=XP:YQ=YP
1130 XP=INT(X):YP=INT(Y)
1140 ZQ=Z4*YP+XP+Z3
1170 IFZR<>ZQTHENPOKEZR,32:ZR=ZQ
1180 POKEZQ,ZB
1190 GOTO1000
1200 REM AT LEFT
1210 Y=Y-DY*(X-X0)/DX
1220 DX=-DX:X=X0:ZB=97
1230 IF(Y<Y0)OR(Y>Y1)GOTO1040
1240 IF(Y<L0-.75)OR(Y>L1+.75)GOTO2200
1290 GOTO1600
1300 REM AT RIGHT
1310 Y=Y-DY*(X-X1)/DX
1320 DX=-DX:X=X1:ZB=225
1330 IF(Y<Y0)OR(Y>Y1)GOTO1040
1340 IF(Y<R0-.75)OR(Y>R1+.75)GOTO2300
1390 GOTO1600
1400 REM AT TOP
1410 X=X-DX*(Y-Y0)/DY
1420 DY=-DY:Y=Y0:ZB=226
1430 GOTO1530
1500 REM AT BOTTOM
1510 X=X-DX*(Y-Y1)/DY
1520 DY=-DY:Y=Y1:ZB=98
1530 IFX<X0THENX=X0
1540 IFX>X1THENX=X1
1550 GOTO1060
1600 REM MAKE BOUNCE FUNNY
1610 NB=NB+1:IFNB<5GOTO1040
1630 DD=SQR(DX*DX+DY*DY)
1640 DX=DX*(1.5*RND(1)+.5)
1644 AD=ABS(DY/DX)
1645 IFAD>2ORAD<.2THENDY=DX*(RND(1)+.5)
1650 DY=DY*((2-5/NB)*RND(1)+.7)
1660 DD=DD/SQR(DX*DX+DY*DY)
1670 DX=DX*DD
1680 DY=DY*DD
1690 GOTO1040
1800 REM AUTO LEFT
1810 IFY<L0THENDM=-2:GOSUB3000:GOTO1100
1820 IFY>L1THENDM=2:GOSUB3000:GOTO1100
1830 GOTO1100
1900 REM AUTO RIGHT
1910 IFY<R0THENDM=-2:GOSUB3100:GOTO1100
1920 IFY>R1THENDM=2:GOSUB3100:GOTO1100
1930 GOTO1100
2000 REM KEY HIT
2030 IFC$=DS$THENDX=DX/1.5:DY=DY/1.5:GOTO1100
2040 IFC$=IS$THENDX=DX*1.5:DY=DY*1.5:GOTO1100
2050 IFC$=RU$THENDM=-3:GOSUB3100:GOTO1100
2060 IFC$=RD$THENDM=3:GOSUB3100:GOTO1100
2070 IFC$=LU$THENDM=-3:GOSUB3000:GOTO1100
2080 IFC$=LD$THENDM=3:GOSUB3000:GOTO1100
2090 IFC$=RA$THENRA=1:GOTO1100
2095 IFC$=RM$THENRA=0:GOTO1100
2100 IFC$=LA$THENLA=1:GOTO1100
2105 IFC$=LM$THENLA=0:GOTO1100
2110 IFC$=AG$GOTO2400
2190 GOTO1100
2200 REM PASSED LEFT
2210 RN=RN+1
2220 GOTO2350
2300 REM PASSED RIGHT
2310 LN=LN+1
2350 REM SHOW WHERE SCORED
2360 X2=XP:Y2=YP:Z2=32:GOSUB900
2370 X2=INT(X):Y2=INT(Y):Z2=42:GOSUB900:TJ=TI+60
2380 IFTI<TJGOTO2380
2390 Z2=32:GOSUB900:GOTO2420
2400 REM A SCORE - DISPLAY AND START A POINT
2410 X2=XP:Y2=YP:Z2=32:GOSUB900
2420 PRINT"[HOME]SCORE: LEFT = "+STR$(LN)+",
     RIGHT = "+STR$(RN)+"[3 SPACE]";
2425 DD=SQR(DX*DX+DY*DY)
2440 R=RND(1):S=RND(1)+.5:DY=RND(1)
2450 IFR>.5THENX=X0:DX=S:Y=(L0+L1)/2
2460 IFR<=.5THENX=X1:DX=-S:Y=(R0+R1)/2
2470 XP=INT(X):YP=INT(Y):
2480 ZB=81
2490 DD=DD/SQR(DX*DX+DY*DY)
2500 DX=DX*DD:DY=DY*DD
2510 NB=0
2520 ZR=999
2540 TJ=1:GOSUB950
2550 X2=XP:Y2=YP:Z2=ZB:GOSUB900
2560 TJ=1:GOSUB950
2590 GOTO1100
3000 REM MOVE LEFT PADDLE DM
3010 X2=X0-1:ZP=103
3020 YA=L0:YB=L1:GOSUB3200
3030 L0=YA:L1=YB
3040 RETURN
3100 REM MOVE RIGHT PADDLE DM
3110 X2=X1+1:ZP=101
3120 YA=R0:YB=R1:GOSUB3200
3130 R0=YA:R1=YB
3140 RETURN
3200 REM MOVE A PADDLE
3210 Z2=32:Y8=Z4*YA+X2+Z3:Y9=Y8+Z4*(YB-YA)
3220 FORY2=Y8TOY9STEPZ4:POKEY2,Z2:NEXTY2
3230 YA=YA+DM:YB=YB+DM
3240 IFYA<Y0THENYB=YB+Y0-YA:YA=Y0
3250 IFYB>Y1THENYA=YA+Y1-YB:YB=Y1
3260 Z2=ZP:Y8=Z4*YA+X2+Z3:Y9=Y8+Z4*(YB-YA)
3270 FORY2=Y8TOY9STEPZ4:POKEY2,Z2:NEXTY2
3280 RETURN
9000 REM LINEARITY CHECK
9010 PRINT"[CLR, RVS]";
9020 FORI=1TO999
9030 PRINT"[shiftLBRACK]";
9040 NEXTI
9050 GOTO9050
```



To indicate motion in this photo we modified the program so the ball, shown as a white dot would leave a trail of 'hollow' dots. The 'trail' is not part of the program listed here.

KALEIDOSCOPE



```
4 ZC=0:C=0:ZQ=59456:ZW=32
5 PRINT "[CLR]";
6 CL(0)=ASC(" ")+128
7 CL(1)=ASC("[?]")-64
8 CL(7)=ASC(" ")
9 CL(3)=ASC("[@]")-128
10 CL(4)=ASC("[shiftLBRACK]")-128
11 CL(5)=ASC("[shiftRBRACK]")-128
12 CL(2)=ASC("[&]")-64
13 CL(6)=ASC(":")
18 N1=32768: N2=40: N3=.625: N4=39.9999
20 FOR W=3 TO 50
30 FOR I=1 TO 19
40 FOR J=0 TO 19
50 K=I+J
60 C=CL((J*3/(I+3)+I*W/12) AND 7)
70 Y1=N1+N2*INT(N3*I)
80 Y2=N1+N2*INT(N3*K)
90 Y3=N1+N2*INT(N3*(N4-I))
100 Y4=N1+N2*INT(N3*(N4-K))
110 POKEI+Y2,C: POKEK+Y1,C: POKEN2-I+Y4,C
120 POKEN2-K+Y3,C: POKEK+Y3,C: POKEN2-I+Y2,C
130 POKEI+Y4,C: POKEN2-K+Y1,C
140 NEXT J
150 NEXT I
160 NEXT W
170 GOTO 20
```

Changes that prevent twinkling, but slow down display...

```
2 GOTO 4
3 WAITZQ,ZW:WAITZQ,ZW,ZW:POKEZC,C:RETURN
110 ZC=I+Y2:GOSUB3: ZC=K+Y1:GOSUB3: ZC=N2-I+Y4:GOSUB3
120 ZC=N2-K+Y3:GOSUB3: ZC=K+Y3:GOSUB3: ZC=N2-I+Y2:GOSUB3
130 ZC=I+Y4:GOSUB3: ZC=N2-K+Y1:GOSUB3
```

Kaleidoscope is a simple program that runs continuously while drawing interesting patterns on the screen of a Commodore PET computer. It is adapted from a program written by Rod Holt on a different computer.

You will probably want to try each of the two variations of the program. As originally written, you may see 'glitches' flashing on the screen while the program executes. You can get rid of these 'glitches' with a variation of this program provided by Larry Tesler. By replacing the POKEs with GOSUBs, as indicated, the program will slow down considerably, but the picture will be cleaner. I personally prefer the visual effects of motion that appear in the original, faster version.

You will notice that there are no PRINT statements in the program: instead, the program POKEs the ASCII equivalent of the graphic characters into the area of memory where the PET stores its current picture display. This area starts at memory location 32768. The first 40 locations of this area are for the first row of characters on the screen. The next 40 locations are for the next row, and so on.

The built-in function, ASC, does not quite give you the numbers you need for doing POKEs instead of PRINTs. If you are interested in experimenting with different graphics characters, the following statement, when executed, will tell you the integer that needs to be added or subtracted from the value ASC computes:

[CLR] ? –ASC("x")+PEEK(32775)

To assure that this statement will work, be sure not to include any spaces after you press the CLR key. Try different graphics characters in place of the x above. You are now prepared to change lines 6-13 with your own graphics characters.

If you want to change the *shapes* of the patterns created, replace

"J*3/(I+3)+I*W/12"

in line 60 with anything you please, and see what happens.

If you are interested in experimenting further, you can change PET's character set by executing POKE 59468,14. To restore the regular character set, POKE 59468,12. While the alternate character set is in effect, the characters generated by shift–) and shift–← make for interesting kaleidoscope patterns.

Dave Offen
Menlo Park, CA

Tiny GRAPHICS

My children have enjoyed running the attached graphic programs on my PET. I am offering them in the hope others may enjoy them.

M C Hofheinz
Stockton, CA

A
```
10 POKE (32768 + 1000*RND(1)),
   255*RND(1)
20 GOTO 10
   (Or substitute any number from 1 to
   255 for the expression after the comma.)
```

B
```
10 FOR X=1 to 255
20 FOR Y=1 to 1000
30 POKE (32, 767+Y), X
40 NEXT Y
50 NEXT X
```

C
```
20 FOR X=1 to 255
30 PRINT X: POKE (32767+X), X
40 NEXT X
   (Best to hold RVS during this one)
```

D
```
20 FOR Y=1 to 1000
30 POKE (32767+Y); INT (Y/4+1)
40 NEXT Y
```

E
```
   Add to any of the above
5  POKE 32768, 14
```

HAM PET OWNERS

Would you like to take part in experiments to transmit programs, etc by Ham Radio? Please get in touch with the undersigned. To arrange a schedule give frequency, time, date, and call letters and perhaps a telephone number.

Orin K Batesole – W6HJE
150 Shady Lane
Walnut Creek, CA 94596
Telephone (415) 934-8661

PET PRINTERS

A rumor of interest to all PET owners who have access to a Versatec printer: We hear that for $100 Versatec (Santa Clara, CA) will sell a Versatec printer/PET interface.

Commodore's $595 printer will allow you to print out graphic characters as they appear on the screen. It sure will be nice to be able to print graphics, but listings will still be confusing if a graphic character prints when cursor control is done in print strings. A sample of the print quality is shown below.

ABCDEFGHIJKLMNOPQRSTUVWX
ABCDEFGHIJKLMNOPQRSTUVWX
ABCDEFGHIJKLMNOPQRSTUVWX

DRAW UPDATE

1) A bug got into our last version: lines 7000 and 7040 should read

```
7000 PRINT "[CLR, DOWN]"
7040 PRINT "[HOME]";: NEW
```

2) A number of readers found line 5535 puzzling:

```
5535 : : V=C>BY: IF V=RV GOTO 5545
```

The leading colons are to force an indentation to make structure clearer. 'C>BY' is a Boolean condition. If C is greater than BY then the expression is TRUE, and so evaluates to –1: therefore V is set equal to –1. If C is not greater than BY, then the expression is false, and evaluates to 0: so V is set equal to 0. In the second command of the line, V is compared to RV in the usual manner.

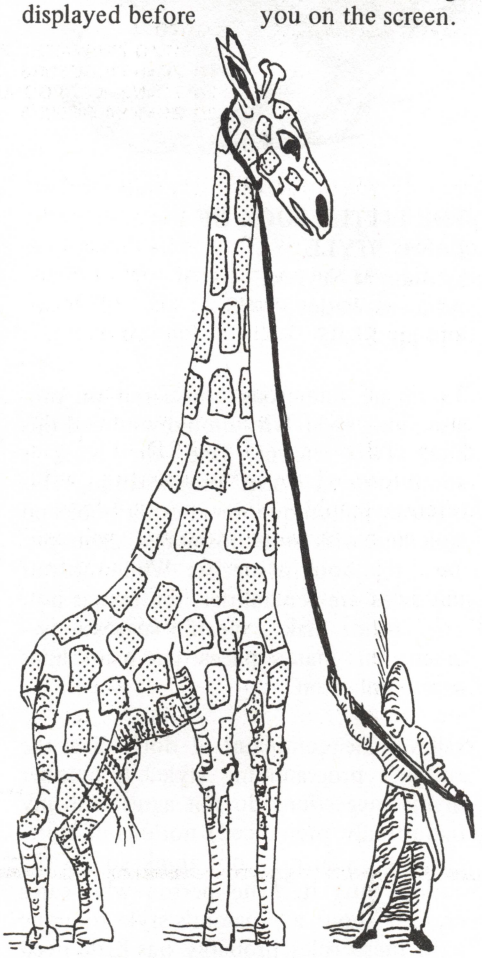
□

REVIEWS

PET SOFTWARE REVIEW

Don Alan Enterprises
P.O. Box 401, Marlton, NJ 08053
10 programs on a cassette, $19.95

Don Alan Enterprises is selling a PET cassette containing ten programs for $19.95. I am generally disappointed with the quality of these programs. However, since there is not yet available a wide selection of programs for the PET, there undoubtedly will be those of you who would rather play with these programs, than stare at the '7167 BYTES FREE' message displayed before you on the screen.



If you're a computer hobbyist who is just learning to program, and you are unfamiliar with the capabilities of small computers, you might appreciate this product. Most of the programs included are short. They provide readable examples of working programs written in the PET's particular dialect of BASIC.

Among the supplied programs are two requiring no intervention once they are started. One program transforms the computer into a digital clock with a large numerical display. The other, called WORM, draws a delightful criss-crossed maze of lines all over PET's display screen.

The remaining eight programs on the tape are interactive games. I find none of the games particularly inspiring. In addition, the authors do not devote nearly enough attention to the needs of the game-player. To me, this is a serious flaw because an important test of any good computer game is that it should be easy to interact with and pleasant to use.

In particular, the math practice program has no facility for letting you determine the difficulty level of the problems. How much value can there be in practicing on problems that may be either too easy or too difficult for you? Also, when you provide an invalid response to the initial question, the program prints out a confused and inappropriate message. This indicates a sloppy programming job.

I found shortcomings with some of the other programs as well. Some of the games unnecessarily require the player to press the return key after each single-letter response. Why should the programmer require that you press two keys when one is adequate? Other games would be considerably improved if they were to automatically repeat when a key is held down, rather than requiring twenty or thirty keystrokes on the same key.

The creators of this package of programs advertise that their product should be used to 'house-break your PET'. Unfortunately, you may discover that if you have an interest in moving beyond the toilet training stage, the Don Alan programs are not for you.

Reviewed by Dave Offen
Computer Software Consultant
Menlo Park, CA

STIMULATING SIMULATIONS
60 pp, $5.00
THE DEVIL'S DUNGEON
15 pp, $3.50
by C William Engel
Box 16612, Tampa, FL 33687

At school or at home, what do you do with your personal computer? Why, you write programs to make it do things, of course. But what things? One approach is to tackle problems directly related to school or work. You can learn a powerful lot of programming skills by developing software to multiply matrices or balancing end-of-month checking account statements. But this applications approach is less than edifying to the developing programmer who lacks meaningful applications suitable to his/her level of skill. An often overlooked alternative is the game-simulation. If the objective is learning to program, why not have fun doing it? There are lots of game-simulations available to be copied. *101 Computer Games*, edited by David Ahl, comes to mind. But this is a canned approach which emphasizes the recreational aspect of personal computing rather than skill development. C William Engel, in writing *Stimulating Simulations* has done a nice job of getting away from the copy-a-game approach. In this booklet, he offers ten game-simulations of varying difficulty. Judging from the accompanying scenarios, they are all exceedingly interesting. Dr Engel's contribution is to fully document each program with a scenario, a sample run, a very readable flow chart, a listing, and suggestions for minor and major changes. So what's new? Well, these programs are understandable. They can be decoded and modified by the learner-programmer. They can be rewritten for different systems or to do different things. In short, they teach!

The Devil's Dungeon is a more sophisticated game-simulation of the same genre as *Stimulating Simulations*. The game seems to be a variation of Caves. The objective of the game is to obtain a maximum amount of gold from the dungeon in the face of many hazards including monsters, and poisonous gas. It appears to have a high potential for interest and challenge. I can't say the *Devil's Dungeon* is in the same league as Star Trek, nor can I say that it isn't. What makes a computer game popular is often obscure. A reading of the scenario, however, and the high quality documentation more than warrants putting the *Devil's Dungeon* high on your things-to-try list.

Reviewed by Peter S Grimes
Curriculum Supervisor
San Jose Unified School District

Personal Software, (PO Box 136-B4, Cambridge MA 02183) offers Stimulating Simulations *on tape with Engel's book for $14.95. On one side of the tape are PET programs, on the other side TRS-80 programs.*

THE LITTLE BOOK OF BASIC STYLE
by John M Nevison
Addison-Wesley, 1978
147 pp, $5.95

There are numerous books out on programming style. Why should you read this one? Two reasons. One, this book is about style in BASIC programming. This is somewhat unique: most other books on style deal with more hospitable languages. Two, this book is specific. While most of the rules are generalities, the text is not. The author makes specific suggestions—indent this many spaces, put blank lines here—and so on.

I have one complaint—I don't like the author's programming style. A number of his suggestions do not agree with my (admittedly prejudiced) notions of style. However, you may not think so. As the author puts it, 'The person who cares enough about a program's style to argue with these rules probably has little need of them. On the other hand, an argument against a rule should be advanced for the same reason the rule itself was suggested: because there is a better way to make the program read.' I agree.

Reviewed by Eryk Vershen.

8080A/8085: ASSEMBLY LANGUAGE PROGRAMMING
by Lance A Leventhal
Osborne & Associates, Inc., 1978
400 pp, $7.50

This book comes as highly recommended as did Osborne and Associates' *An Introduction to Microcomputers, Volume 0: The Beginner's Book* (see Tom Williams' review in the March-April 1978 issue).

8080A/8085 is written in the same style as *Volume 0*, and it is everything I had hoped for in an instructional text on assembly language, as well as on how to use assembly language to program a microcomputer. It begins with a brief discussion of the meaning of instructions—the programming problem (program understandability, debuggability, entry speed, readability, and length), using octal versus hexidecimal, instruction code mnemonics, and advantages and disadvantages of high-level (as well as assembly) language. Next, there is a 'basic-literacy' discussion of assemblers and loaders, followed by thorough and concise definitions, descriptions, and examples of each instruction of the entire 8080A and 8085 instruction sets.

8080A/8085 goes one step further than *Volume 0* in that not only is it a primer, in the classical sense, full of examples and samples, but it is also an excellent reference manual, with sample macros, programs (one's complement, 8- and 16-bit addition/subtraction, word dis/assembly, sum of squares, and more), simple program loops, character-coded data, code conversion, arithmetic problems, tables and lists, subroutines, I/O devices and programs, interrupts—the list goes on and on. Chapters 14 and 15, on debugging, testing, documentation, and re-design, are, in themselves, worth the price of the book.

If you have (or plan to have) an 8080 microprocessor, and you want to program it in assembly language, *8080A/8085* is written especially for you. In short, the first twelve chapters concentrate on the writing of short programs; the rest describes how to formulate tasks as programs and how to put short programs together to form a *working* system.

Reviewed by Vicki Parish. □