COMPUTE!'s

ATARI Collection

VOLUME 2

More than two dozen never-before-published games, applications, utilities, and educational programs for the Atari 400, 800, XL, and XE home computers.

A **COMPUTE!** Books Publication

$14.95

COMPUTE!'s ATARI Collection

VOLUME 2

COMPUTE! Publications, Inc.
One of the ABC Publishing Companies
Greensboro, North Carolina

The following articles and programs were first published in *COMPUTE!* magazine, copyright Small System Services, Inc.:

"Atari Data Management/Database System," November 1981;
"Machine Language Sort Utility," March 1982.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-87455-029-7

COMPUTE! Publications, Inc., Post Office Box 5406, Greensboro, NC 27403, (919) 275-9809, is one of the ABC Publishing Companies and is not associated with any manufacturer of personal computers. Atari is a trademark of Atari, Inc.

Contents

Foreword

COMPUTE!'s Atari Collection, Volume 2 collects more than two dozen articles and programs never before published, and includes three excellent utilities which appeared in out-of-print issues of *COMPUTE!* magazine from the early 1980s.

Divided into five chapters, *COMPUTE!'s Atari Collection, Volume 2* includes something for every Atari user. With action games like "Front Attack" and "Electronic Football," strategy games such as "Heroes" and "The Catacombs of Your Mind," and nine other games, Atari arcade lovers will be busy for hours.

Programmers will enjoy the convenience of using "Block Line Deleter" and "Variable Changer" and the easy creation of vivid graphics with "Graphics Plus." And, as with most COMPUTE! collections, there are application programs that turn your Atari from a game machine into a workhorse. All the programs in *COMPUTE!'s Atari Collection, Volume 2* are ready to type in and enjoy. We've even included the error-checking utility, "The Automatic Proofreader," which makes program entry errors obsolete.

Atari personal computers have been vital for a long time—and so have COMPUTE!'s Atari books. When Atari users look for quality programs and clear explanations of the most difficult techniques, they look to COMPUTE!. Our second Atari collection will be no exception.

> If you prefer to purchase a disk containing all the programs in this book rather than type them in, just use the convenient coupon in the back, or call toll-free 1-800-334-0868 (in North Carolina, call 1-919-275-9809).

Chapter 1

Applications

The Magic Directory

Stuart Goldenberg

Access many of the DOS commands from this easy-to-use BASIC utility. Requires a disk drive.

How often have you been in BASIC and found that you needed to do some disk cleanup—perhaps you wished to rename some programs or even delete them? Perhaps you wished you knew what was on the disk, but because you didn't write everything down, were forced to go to DOS to get the directory?

Going back and forth between DOS and BASIC is time-consuming. And when you return to BASIC, you're not sure if the filename had one *T* in it or two. Besides, spelling is not one of your strong points, and remembering that funny name, which made sense when you named the program, just doesn't make sense now.

Wouldn't it be nice to be able to do most of your disk management without having to type in the program names, and be able to load and run programs the same way?

"The Magic Directory" will let you display the disk directory on the monitor and get a hardcopy of it on your printer. You can exit the program to BASIC, you can delete a program, you can lock and unlock, rename, load, run, or enter a program. All these options can be done from this utility program. The table below is a list of commands.

Just select the appropriate file by using the normal cursor controls, which move a + cursor in front of the names of programs. Select the desired program, and then press the appropriate option and follow instructions as displayed.

Magic Directory Commands

Description	Command	Notes
List of files on disk	CTRL-A	Lists first 44 files
	CTRL-S	Lists remaining files
	CTRL-H	Hardcopy of display
Exit program to BASIC	CTRL-B	Clears memory (NEW)
Delete file	CTRL-D	Deletes locked or unlocked programs, providing prompt as to whether the file is locked or not

Description	Command	Notes
Rename files	CTRL-N	Renames locked or unlocked files, indicating if the file is locked and preventing the use of an existing filename
Lock files	CTRL-P	
Unlock files	CTRL-U	
LOAD file from directory	CTRL-L	
ENTER file from directory	CTRL-E	ENTER does not clear memory of prior programs
RUN file from directory	CTRL-R	Runs file by cursor

Any attempt to do anything to DOS.SYS, DUP.SYS, or MEM.SAV will give you the option of entering DOS. If these options are different on your DOS, change line 440 as appropriate.

Magic Directory

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
LO 10 OPEN #2,4,0,"K":OPEN #4,12,0,"S":OPEN #1
      ,6,0,"D:*.*"
MN 20 REM READING IN THE DIRECTORY
KE 30 GRAPHICS 0:PRINT "{13 SPACES}DIRECTORY"
CO 40 DIM A$(20),A1$(39),T$(20),T1$(39),F$(12)
      ,RENAME$(12):F$="FREE SECTORS"
FO 50 LINE=0:POKE 752,1
DA 60 TRAP 120
NA 70 INPUT #1;A$,A1$:LINE=LINE+1
EH 80 IF A1$(5,16)=F$ THEN GOTO 110
DJ 90 PRINT A$;" ";A1$:IF LINE=22 THEN GOTO 13
      0
DB 100 GOTO 70
KH 110 PRINT A$:PRINT A1$;:IF LINE=22 THEN LIN
       E=23
PD 120 IF A$(5,16)=F$ THEN PRINT A$:IF LINE=22
        THEN LINE=23
HI 130 REM
IG 140 IF LINE=22 THEN INPUT #1,A$:IF A$(5,16)
       =F$ THEN POSITION 2,23:LINE=100:PRINT A
       $;:GOTO 150
KH 150 XT=PEEK(90):LR=3:X=2
GE 160 POKE 84,X-(LINE=1)-(LINE=23):POKE 85,3:
       PUT #4,43
DG 170 K=0:GET #2,K
HO 180 POKE 85,LR:POKE 195,0:PUT #4,32
```

```
GG 190  IF K=28 AND X>=2 THEN X=X-1:POKE 84,X:P
        OKE 85,LR
DL 200  IF K=29 AND X<XT-1+(LINE=22) THEN X=X+1
        :POKE 84,X:POKE 85,LR
HP 210  IF K=30 THEN POKE 85,3:LR=3
OB 220  IF K=31 THEN POKE 85,21:LR=21
AB 230  IF K=4 OR K=5 OR K=12 OR K=14 OR K=16 O
        R K=18 OR K=21 THEN GOTO 300
AC 240  IF K=1 THEN RUN
LE 250  IF K=2 THEN GRAPHICS 0:NEW
PA 260  IF K=8 THEN GOTO 890
BM 270  POKE 85,LR:PUT #4,43
OF 280  IF K=19 AND LINE=22 THEN GOSUB 1030:INP
        UT #1,A1$:GOTO 80
GM 290  GOTO 170
IC 300  IF K=16 THEN POKE 85,LR-1:GET #4,TX:PUT
         #4,43:IF TX=42 THEN GOTO 170
HQ 310  IF K=21 THEN POKE 85,LR-1:GET #4,TX:PUT
         #4,43:IF TX=32 THEN GOTO 170
AB 320  REM READING NAME AND PACKING IT
IP 330  T$(1)="D:"
DP 340  FOR I=3 TO 14
KH 350  GET #4,TX
FL 360  T$(I,I)=CHR$(TX)
CC 370  NEXT I
KK 380  POKE 85,12+LR:PUT #4,32
MF 390  POKE 752,0
PO 400  IF T$(11,11)<>" " THEN T$(14,14)=T$(13,
        13):T$(13,13)=T$(12,12):T$(12,12)=T$(11
        ,11):T$(11,11)="."
DL 410  FOR I=1 TO 14
IM 420  IF T$(I,I)<>" " THEN T1$(LEN(T1$)+1)=T$
        (I,I)
BP 430  NEXT I
NP 440  IF T1$="D:DOS.SYS" OR T1$="D:DUP.SYS" O
        R T1$="D:MEM.SAV" THEN GOSUB 850:GOTO 8
        40+(T=35)*40
AF 450  TRAP 460:XIO 13,#1,0,0,T1$
AN 460  IF PEEK(195)=170 THEN GRAPHICS 0:PRINT
        "ERROR 170 - ILLEGAL FILE NAME OR FILE
        NOT FOUND":GOSUB 860:RUN
GO 470  TRAP 480
GH 480  IF PEEK(195)=21 THEN GRAPHICS 0:PRINT "
        ERROR 21 - NON LOAD FILE":GOSUB 860:RUN
HK 490  IF PEEK(195)=165 THEN GRAPHICS 0:PRINT
        "ERROR 165 - FILE NAME ERROR":GOSUB 860
        :RUN
DJ 500  REM LOAD, RUN OR ENTER
```

```
LA 510  IF K=12 THEN GRAPHICS 0:PRINT "LOADING
        ";T1$:LOAD T1$
KP 520  IF K=18 THEN GRAPHICS 0:PRINT "RUNNING
        ";T1$:RUN T1$
EA 530  IF K=5 THEN GRAPHICS 0:PRINT "ENTERING
        ";T1$:ENTER T1$
HA 540  REM UNLOCKING PROCEDURE
EO 550  POKE 85,LR-1:GET #4,S
GB 560  IF S<>42 THEN GOTO 660
EO 570  GRAPHICS 0:PRINT "TYPE # TO CONTINUE, P
        ROGRAM IS LOCKED,CONTINUING WILL UNLOCK
         THE PROGRAM"
MM 580  PRINT "AND CONTINUE YOUR REQUEST TO ":P
        RINT " "
EP 590  IF K=4 THEN PRINT "DELETE ";T$
IP 600  IF K=21 THEN PRINT "UNLOCK ";T$
HO 610  IF K=14 THEN PRINT "RENAME ";T$
LG 620  GOSUB 860
IC 630  IF T<>35 THEN RUN
AM 640  XIO 36,#1,0,0,T1$:GOTO 720
II 650  REM UNLOCK, LOCK, DELETE PROCEDURE
LL 660  GRAPHICS 0:PRINT "TYPE # TO CONTINUE YO
        UR REQUEST TO":PRINT " "
PH 670  IF K=16 THEN PRINT "LOCK ";T$
EP 680  IF K=4 THEN PRINT "DELETE ";T$
IG 690  IF K=14 THEN PRINT "RENAME ";T$
LF 700  GOSUB 860
IB 710  IF T<>35 THEN RUN
KF 720  IF K=4 THEN XIO 33,#1,0,0,T1$:RUN
NL 730  IF K=16 THEN XIO 35,#1,0,0,T1$:RUN
NJ 740  IF K=21 THEN XIO 36,#1,0,0,T1$:RUN
AB 750  REM RENAME PROCEDURE
MI 760  PRINT " ":PRINT "newname.ext":INPUT REN
        AME$:PRINT :PRINT "RENAME D:OLD NAME,NE
        W NAME"
JH 770  T$="{16 SPACES}":T$="D:":T$(3)=RENAME$
OA 780  TRAP 790:XIO 13,#1,0,0,T$
GM 790  IF PEEK(195)<>170 THEN GRAPHICS 0:PRINT
         "FILE NAME ALREADY ON DISK OR ILLEGAL
         FILE NAME":POKE 195,0:GOTO 760
PG 800  T1$(LEN(T1$)+1)=",":T1$(LEN(T1$)+1)=REN
        AME$
NK 810  PRINT " ":PRINT T1$
PM 820  XIO 32,#1,0,0,T1$
GN 830  CLOSE #2:CLOSE #4
JB 840  RUN
GH 850  GRAPHICS 0:PRINT "BY TYPING #, DOS WILL
         BE ENTERED, OR"
CK 860  PRINT "TYPE ANY KEY EXCEPT # TO RETURN
        TO THEDIRECTORY"
```

```
NC 870 GET #2,T:POKE 195,0:RETURN
IG 880 DOS
HK 890 TRAP 990
KF 900 OPEN #7,8,0,"P"
DA 910 POKE 85,3:PUT #4,32:POKE 85,21:PUT #4,3
       2
KD 920 PRINT #7:PRINT #7
JI 930 FOR Y=0 TO XT
DF 940 POSITION 1,Y:FOR I=1 TO 39:GET #4,S:T1$
       (I,I)=CHR$(S):NEXT I
GJ 950 PRINT #7;T1$
DH 960 NEXT Y
KI 970 PRINT #7:PRINT #7
JG 980 RUN
IB 990 GRAPHICS 0:PRINT "ERROR 138 - CHECK PRI
       NTER/INTERFACE TOSEE IF THEY ARE ON"
NP 1000 GOSUB 860
JC 1010 CLOSE #7
LI 1020 RUN
GK 1030 INPUT #1,A1$
OG 1040 GRAPHICS 0:POKE 752,1:PRINT "
        {6 SPACES}DIRECTORY{3 SPACES}-
        {4 SPACES}PAGE 2"
LD 1050 LINE=LINE+1:POP :GOTO 80
```

Atari Bibliography

Dana Noonan

Index all your computer magazines conveniently and easily and store the data in files on disk. Requires 48K RAM and a disk drive.

If you're like me, you have piles of computer magazines. Often, I know there's an article somewhere in that pile about a particular subject if only I could remember where.

To solve this problem I wrote "Atari Bibliography" to keep track of all those magazine articles I had sitting around the house. The program had to meet several criteria. It had to be able to retrieve entries by title, source, category, and programming language, but not by author, year, or issue. Since hundreds of articles containing Atari programs are published each year, the program had to be able to handle a very long string in RAM memory. And because it's easier and faster to find something in an alphabetized list, it had to include a simple title sort. As the program developed, I added features so that entries could be changed, deleted, or printed.

The sort was first written in BASIC, but it was very slow. I adapted Ron Marcuse's machine language sort routine, which first appeared in the March 1982 issue of *COMPUTE!* and was reprinted in *COMPUTE!'s Third Book of Atari.* It appears below as Program 1.

One Long String

The Atari Bibliography program (Program 2) requires 48K RAM. Because of the program's large memory requirements, frequently used constants are read in as variables to save memory. In line 210, Z$ is DIMensioned to 23500—or the equivalent of 500*47. Each substring—A$ or Z$(#*K−J,#*K)—is 47 characters long and contains the following information:

A$(1,1)	Category
A$(2,2)	Language
A$(3,3)	Source
A$(4,5)	Year
A$(6,7)	Issue or month
A$(8,10)	Starting page
A$(11,47)	Title and comments if necessary

If you receive an Error 2 message at line 210, you're using a DOS other than 2.0S, or you have less than 48K RAM.

Lower the number in the DIM statement for Z$. Try DIM Z$(18500). Now you'll be able to enter only 400 articles instead of 500.

The title (T$) fits on one line (37 characters), so you can see just how much room you have for title and comments. If you near the end of the line, you can go back and shorten the title to fit. If you want to add a comment about an article, and if the title is short, you can include it here. If a title begins with the words *Atari* or *The,* begin with the second word. Shorten titles any way that makes sense to you. If you add a short comment to series titles, you'll be able to find just the right article faster.

Typing In the Programs

Atari Bibliography is composed of two programs. Using "The Automatic Proofreader" from Appendix C, type in Program 1, "Machine Language Sort," written by Ron Marcuse. ***Save it before you run it.*** Next, type in Program 2, Atari Bibliography, and save it using the filename **BIBL.BAS.** Each time you want to use the program, you must load and run Program 1. Program 1 will install the machine language, and then load and run Program 2 for you.

Using the Program

The first time you use the program, you'll, of course, have no data saved on disk. You should start entering data by year. You'll have nine choices: the first for magazines prior to 1983, and then for each year through 1990. Pick the year you want to start with and press the corresponding key. The disk drive will spin and a new menu will appear indicating that there are no records in this file.

From this menu you can choose to enter data; sort it; search by title, source, category, or programming language; save the data; load a new data file; or quit the program. At this point you'll want to enter data. The maximum number of articles permitted for one file is 500. Entering data is as simple as following the prompts and answering the questions. Entry has been made faster by predefined answers such as a list of magazines to pick from.

Once you have entered all the data, be sure to return to the menu and save the data. Now, whenever you want to search for an article, just load the year you think the article

appeared and search for it. When you search by title, only the article title will appear. Press the space bar to see the next title or another key to see all the information for the last title.

After entering a few articles, save the data and experiment with the different options; this is the best way to become familiar with all the features.

Program 1. Machine Language Sort

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
DC 10  REM ATARI BIBLIOGRAPHY
IP 20  REM PROGRAM 1
EH 30  REM
CG 40  REM THIS PROGRAM LOADS AND RUNS
MB 50  REM PROGRAM 2.
AJ 60  REM YOU MUST SAVE PROGRAM 2
BA 70  REM USING THE FILENAME BIBL.BAS
EM 80  REM
EN 90  REM
CK 100 REM MACHINE LANGUAGE SORT UTILITY
CH 110 REM FROM COMPUTE! MARCH 1982
PI 120 REM REPRINTED IN COMPUTE!S THIRD BOOK O
       F ATARI
BC 130 REM BY RON MARCUSE
HJ 140 REM
NE 800 ? CHR$(125):POSITION 2,10:? "ENTERING M
       ACHINE LANGUAGE...":? "{DOWN} PLEASE WA
       IT"
MI 810 FOR I=1568 TO 1693:READ A:POKE I,A:NEXT
        I
JE 820 TRAP 840:? CHR$(125):POSITION 2,10:? "L
       OADING D:BIBL.BAS"
GP 830 RUN "D:BIBL.BAS"
CF 840 ? CHR$(125);CHR$(253):POSITION 2,10:? "
       BIBL.BAS NOT ON THIS DISK"
HB 850 ? "{DOWN}INSERT PROPER DISK AND PRESS A
        KEY":POKE 764,255
OK 860 IF PEEK(764)=255 THEN 860
EE 870 POKE 764,255:GOTO 820
FB 1568 DATA 104,104,133,217,104,133
FD 1574 DATA 216,104,133,209,104,133
PH 1580 DATA 208,169,0,133,218,133
PJ 1586 DATA 207,162,1,165,216,133
CJ 1592 DATA 214,165,217,133,215,24
FF 1598 DATA 165,214,133,212,101,205
FA 1604 DATA 133,214,165,215,133,213
OD 1610 DATA 105,0,133,215,164,203
CF 1616 DATA 165,206,240,10,177,214
```

```
OL 1622 DATA 209,212,144,44,240,12
DF 1628 DATA 176,19,177,214,209,212
IK 1634 DATA 144,13,240,2,176,30
FI 1640 DATA 200,196,204,240,227,176
ME 1646 DATA 23,144,223,169,1,133
GB 1652 DATA 218,164,205,136,177,214
CN 1658 DATA 72,177,212,145,214,104
PD 1664 DATA 145,212,192,0,208,241
IE 1670 DATA 232,224,0,208,2,230
GJ 1676 DATA 207,228,208,208,172,165
GP 1682 DATA 209,197,207,208,166,165
MN 1688 DATA 218,201,0,208,144,96
```

Program 2. Atari Bibliography

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
AC 1 REM ATARI BIBLIOGRAPHY
NB 2 REM SAVE THIS PROGRAM WITH
CF 3 REM SAVE "D:BIBL.BAS"
BI 4 REM
CM 10 GOTO 200
GP 15 POKE Z,Z1:FOR C=Z1 TO Z10+Z2:? ,CHR$(C+Z
      64);". ";D$(C*Z20-Z19,C*Z20):NEXT C:POKE
       Z,Z0:RETURN
GN 20 POKE Z,Z1:FOR C=Z1 TO Z10+Z3:? ,CHR$(C+Z
      64);". ";E$(C*Z20-Z19,C*Z20):NEXT C:POKE
       Z,Z0:RETURN
IJ 25 POKE Z,Z1:FOR C=Z1 TO Z11:? ,CHR$(C+Z64)
      ;". ";F$(C*Z10-Z9,C*Z10):NEXT C:POKE Z,Z
      0:RETURN
DA 30 N=LEN(Z$)/K:RETURN
CN 35 ? S$:POSITION Z11,Z10:POKE Z,Z1:? "ONE M
      OMENT PLEASE":RETURN
AG 40 CLOSE #Z2:? :? "{4 SPACES}TO SCREEN OR P
      RINTER?";:GET #Z1,R:IF R<>80 THEN OPEN #
      Z2,Z4,Z0,"S:":POKE Z,Z1:P=Z0:RETURN
MG 45 TRAP 50:OPEN #Z2,Z4,Z0,"P:":GOSUB Z35:S=
      Z0:P=Z1:RETURN
BO 50 TRAP X:CLOSE #Z2:? S$:? :? "{5 SPACES}CH
      ECK PRINTER - HIT RETURN":GET #Z1,R:GOTO
       Z40+Z5
HE 60 CLOSE #Z2:POSITION Z2,Z20+Z2:? "
      {13 SPACES}END OF FILE{12 SPACES}":? L$;"
      {UP}":FOR C=Z1 TO Z900:NEXT C:GOTO Z700
PA 70 E=Z5+Z10:H$="SOURCE":B=72:M=77:GOTO Z100
      -Z10
EF 72 IF R$=B$(Z3,Z3) THEN GOSUB Z112
EL 74 RETURN
```

```
NA 76  E=Z20:H$="CATEGORY":B=78:M=77:GOTO Z100-
       Z10
EH 78  IF B$(Z1,Z1)=R$ THEN GOSUB Z112
EI 80  RETURN
GH 82  E=Z20+Z5:H$="LANGUAGE":B=84:M=74:GOTO Z1
       00-Z10
EG 84  IF B$(Z2,Z2)=R$ THEN GOSUB Z112
EO 86  RETURN
EG 90  ? S$:GOSUB E:? :? ,"Y. ALL":? :? ,"Z. NO
       NE":? :? H$;:INPUT R$:IF R$="Z" THEN GOT
       O Z700
CG 92  GOSUB Z40:IF R$="Y" THEN U=Z1:FOR W=Z64+
       Z1 TO M:R$=CHR$(W):RR=W-Z64
HP 100  IF P=Z0 THEN GOSUB 144
PB 102  IF Q>0 THEN GOTO Z100+Z8
IM 104  GOTO Z40
NJ 106  GOSUB Z35
MD 108  FOR I=Q TO N:B$=Z$(I*K-J,I*K):GOSUB B:N
        EXT I:IF U=Z0 THEN GOTO Z30+Z30
DL 110  ? #Z2:S=S+Z1:NEXT W:GOTO Z30+Z30
OB 112  IF P=Z1 THEN ? #Z2;CHR$(15),B$(Z11,K);:
        GOTO Z100+Z20
ED 114  ? #2;B$(Z11,K):GET #Z1,R:L=L+Z1:IF L>Z1
        0+Z8 AND R=Z30+Z2 THEN GOSUB 144
DJ 116  IF R=Z30+Z2 THEN RETURN
HP 118  IF R=77 THEN GOTO Z700
BC 120  RR=ASC(B$(Z3,Z3))-Z64:? #Z2;D$(RR*Z20-Z
        19,RR*Z20);
HD 122  ? #Z2;"19";B$(Z4,Z5);"{3 SPACES}#";B$(6
        ,7);"  p.";:IF P=Z1 THEN ? #2;B$(Z8,Z10
        ),:GOTO 126
LF 124  ? #Z2;B$(Z8,Z10):L=L+Z1
ID 126  RR=ASC(B$(Z1,Z1))-Z64:? #Z2;E$(RR*Z20-Z
        19,RR*Z20);:L=L+Z2:IF B$(Z2,Z2)=" " THE
        N ? #Z2:GOTO Z100+Z30
KM 128  RR=ASC(B$(Z2,Z2))-Z64:? #Z2;F$(RR*Z10-Z
        9,RR*Z10)
AD 130  IF P=Z0 THEN 136
BD 132  S=S+Z1:IF S<Z64-Z2 THEN RETURN
LD 134  FOR C=Z1 TO Z4:? #2:NEXT C:S=Z0:RETURN
ON 136  ? #Z2:GET #Z1,R:IF R=77 THEN GOTO Z700
IB 138  IF R=67 THEN GOTO Z800
LL 140  IF L>Z10+Z8 THEN L=Z0:GOSUB 144
HH 142  RETURN
GD 144  L=Z0:? S$:POSITION Z2,Z20+Z2:? "
        {5 SPACES}SPACE BAR FOR FAST LISTING
        {5 SPACES} "
EA 146  ? "■CHANGE ENTRY{4 SPACES}MENU
        {5 SPACES}NEXT ITEM{UP}":POSITION Z2,Z
        0:RETURN
```

```
HM 200 READ Z0,Z1,Z2,Z3,Z4,Z5,Z8,Z9,Z10,Z11,Z1
       9,Z20,Z30,Z35,Z40,Z64,Z100,Z112,Z300,Z5
       00,Z600,Z700,Z800,Z900,J,K,Z,X
CP 205 DATA 0,1,2,3,4,5,8,9,10,11,19,20,30,35,
       40,64,100,112,300,500,600,700,800,900,4
       6,47,752,40000
JM 210 DIM Z$(23500),D$(240),E$(260),F$(Z100+Z
       10),DD$(Z8),H$(Z8),T$(K-Z10),S$(Z1),Y$(
       Z10-Z3),M$(Z1),R$(Z3),C$(Z2)
PP 215 DIM B$(K),Q$(K),A$(K),L$(Z40-Z4):L$="
       {M}":L$(Z40-Z4)=L$:L$(Z2)=L$:S$="
       {CLEAR}":OPEN #Z1,Z4,Z0,"K:"
BA 220 ? S$:POKE Z,Z1:POSITION Z10,Z10:? "COMP
       UTER BIBLIOGRAPHY":? :? :? ," BY DANA N
       OONAN"
EA 222 POSITION 2,20:? "COPYRIGHT COMPUTE! PUB
       LICATIONS, INC."
OH 225 POKE 203,Z10:POKE 204,J:POKE 205,K:POKE
        206,Z0
MH 230 D$=" ":D$(240)=D$:D$(Z2)=D$:FOR I=Z1 TO
        Z10+Z2:READ B$:D$(I*Z20-Z19,I*Z20)=B$:
       B$="":NEXT I
NF 235 E$=" ":E$(260)=E$:E$(Z2)=E$:FOR I=Z1 TO
        Z10+Z3:READ B$:E$(I*Z20-Z19,I*Z20)=B$:
       B$="":NEXT I
OH 240 F$=" ":F$(110)=F$:F$(Z2)=F$:FOR I=Z1 TO
        Z11:READ B$:F$(I*Z10-Z9,I*Z10)=B$:B$="
       ":NEXT I
EK 245 Z$="":DD$="D:BIB.D ":? S$:? "{DOWN}  CO
       MPUTE!'S ATARI BIBLIOGRAPHY{DOWN}"
HA 246 ? ,"1. BEFORE 1983":? ,"2. 1983":? ,"3.
        1984"
MP 247 ? ,"4. 1985":? ,"5. 1986":? ,"6. 1987"
FC 248 ? ,"7. 1988":? ,"8. 1989":? ,"9. 1990":
       POKE Z,Z0
OI 250 ? "{DOWN}  ENTER THE NUMBER OF YOU CHOI
       CE?";:GET #1,R:? CHR$(R):IF R<49 OR R>5
       7 THEN ? CHR$(253):GOTO 245
AK 260 DD$(Z8,Z8)=CHR$(R):F=Z0:GOSUB Z30:GOSUB
        Z35:TRAP Z300-Z30:CLOSE #Z2:OPEN #Z2,Z
       4,Z0,DD$
BD 265 FOR I=Z1 TO Z500:INPUT #Z2;B$:Z$(LEN(Z$
       )+Z1)=B$:NEXT I
OH 270 TRAP X:CLOSE #Z2:GOTO Z300
CL 280 ? S$:? "{DOWN}{TAB}SAVE ";DD$;" (Y/N)?"
       ;:GET #Z1,R:IF R=89 THEN GOTO Z300-Z10
LO 282 ? :? ,"RETURN TO MENU (Y/N)";:INPUT R$:
       IF R$<>"N" THEN GOTO Z300
```

```
IN 284 ? :? "SAVE FILE (";DD$;")";:INPUT DD$:I
       F DD$="" OR LEN(DD$)<Z8 THEN ? "{2 UP}"
       :GOTO 284
JI 290 GOSUB Z35:CLOSE #Z2:OPEN #Z2,Z8,Z0,DD$:
       FOR I=Z1 TO N:? #Z2;Z$(I*K-J,I*K):NEXT
       I:CLOSE #Z2:GOTO Z300
HH 295 END
FP 300 ? S$:? ,"  ARTICLE FILE":GOSUB Z30:? :?
        ,"  ";N;" RECORDS":? :? :? ,"1. ENTER
       DATA":? :? ,"2. SORT DATA"
FB 305 ? :? ,"3. RETRIEVE DATA":? :? ,"4. SAVE
        DATA":? :? ,"5. LOAD NEW FILE(MAIN MEN
       U)":? :? ,"6. QUIT"
JC 310 POKE Z,Z0:? :? ,"CHOICE?";:GET #1,R:IF
       R<49 OR R>54 THEN ? "{BELL}{2 UP}":GOTO
        310
KH 315 R=R-48:ON R GOTO 350,Z500,Z700,280,245,
       Z300-Z5
NE 350 ? S$:GOSUB Z30:IF N=Z500 THEN ? ,DDD$;"
        FULL":FOR I=1 TO Z900:NEXT I:GOTO Z300
EM 352 B$="":POKE Z,Z0:? "TITLE (TYPE END TO Q
       UIT)":INPUT T$:? :IF T$="" OR LEN(T$)>Z
       40-Z3 THEN 350
FA 354 IF T$="END" THEN GOTO Z300
GH 360 IF LEN(T$)<Z40-Z3 THEN T$(LEN(T$)+Z1)="
        ":GOTO 360
DD 365 IF F=Z1 THEN ? :? "SAME ISSUE = RETURN
       - OTHER = ANY KEY":GET #Z1,R:IF R=155 T
       HEN Y$=Y$(Z1,Z4):GOTO 390
AI 370 GOSUB Z5+Z10:?
GB 375 ? "SOURCE";:INPUT M$:IF ASC(M$)>76 OR A
       SC(M$)<Z64-Z1 THEN ? "{BELL}{2 UP}":GOT
       O 375
DI 380 Y$="":? :? "LAST 2 DIGITS OF YEAR";:INP
       UT R$:Y$(Z1,Z2)=R$
NM 385 ? :? "MONTH/ISSUE ";:INPUT R$:Y$(LEN(Y$
       )+Z1)=R$:F=Z1:IF LEN(Y$)<Z4 THEN Y$(LEN
       (Y$)+Z1)=" "
EO 390 ? :? "PAGE";:INPUT R$:Y$(LEN(Y$)+Z1)=R$
EF 395 IF LEN(Y$)<7 THEN Y$(LEN(Y$)+Z1)=" ":GO
       TO 395
FL 400 ? S$:GOSUB Z20:C$="{3 SPACES}":?
MI 405 ? "CATEGORY";:INPUT R$:C$(Z1,Z1)=R$:IF
       ASC(R$)>77 OR ASC(R$)<65 THEN ? "{BELL}
       {2 UP}":GOTO 405
OF 410 ? S$:GOSUB Z20+Z5:? :? "LANGUAGE";:INPU
       T R$:IF R$="" THEN C$(Z2,Z2)=" ":GOTO 4
       20
AE 412 IF ASC(R$)>75 OR ASC(R$)<65 THEN PRINT
       CHR$(253):GOTO 410
```

```
EJ 415  C$(Z2,Z2)=R$
DI 420  A$="":A$(LEN(A$)+Z1)=C$:A$(LEN(A$)+Z1)=
        M$:A$(LEN(A$)+Z1)=Y$:A$(LEN(A$)+Z1)=T$:
        Z$(LEN(Z$)+Z1)=A$
NM 430  POKE Z,Z1:? :? ," CONTINUE OR MENU":GET
         #Z1,R:IF R<>67 AND R<>77 THEN ? "
        {BELL}":GOTO Z430
CH 435  IF R=67 THEN GOTO 350
HJ 440  IF R=77 THEN GOTO Z300
CP 500  GOSUB Z35:IF N>Z1 THEN A=USR(1568,ADR(Z
        $),N):GOTO Z300
CB 600  IF ASC(T$)<Z64+Z1 THEN Q=Z1:GOTO Z100
BI 605  Q=N:Q=INT(Q/Z2):Q$="":R=N:GOTO Z600+Z20
KP 610  Q$=B$
FC 615  IF Q=Z0 THEN RETURN
PK 620  B$=Z$(Q*K-J,Q*K):IF B$=Q$ THEN Q=Q-Z1:R
        ETURN
AI 625  IF B$(Z11,TT)=A$(Z11,TT) THEN Q=Q-Z1:GO
        TO Z600+Z40
FL 630  IF B$(Z11,TT)>A$(Z11,TT) THEN R=Q:Q=INT
        (Q/Z2):GOTO Z600+Z10
KB 635  IF B$(Z11,TT)<A$(Z11,TT) THEN M=(R-Q)/Z
        2:Q=INT(Q+M):GOTO Z600+Z10
OA 640  B$=Z$(Q*K-J,Q*K):IF B$(Z11,TT)=A$(Z11,T
        T) THEN Q=Q-Z1:GOTO Z600+Z40
HP 645  RETURN
LC 700  U=Z0:? S$:? "{5 SPACES}YOU CAN RETRIEVE
         ARTICLES BY:":? :? ,"1.  ALL ARTICLES"
DK 705  ? :? ,"2.  TITLE":? :? ,"3.  SOURCE"
DB 710  ? :? ,"4.  CATEGORIES":? :? ,"5.  LANGU
        AGE":? :? ,"6.  MAIN MENU"
PJ 715  POKE Z,Z0:POSITION Z10,15:? "YOUR CHOIC
        E?";:GET #Z1,R:? CHR$(R):IF R<49 OR R>5
        4 THEN ? "{BELL}{4 UP}":GOTO 715
HP 720  Q=Z1:L=Z0:S=Z0:GOSUB Z20+Z10:R=R-48:ON
        R GOTO Z700+Z30,Z700+Z40,70,76,82,Z300
BE 730  ? :GOSUB Z40:GOSUB 144:FOR I=Z1 TO N:B$
        =Z$(I*K-J,I*K):GOSUB Z112:NEXT I:GOTO Z
        30+Z30
IM 740  ? :? "TITLE";:INPUT T$:IF T$="" THEN ?
        "{BELL}{2 UP}":GOTO Z700+Z40
NO 745  TT=LEN(T$)+Z10:A$="  ":A$(K)=A$:A$(2)=A$
        :A$(Z11,TT)=T$:GOSUB Z40:B=750:GOSUB Z6
        00:GOTO Z100
JC 750  IF B$(Z11,TT)=A$(Z11,TT) THEN GOSUB Z11
        2
AI 755  IF B$(Z11,TT)>A$(Z11,TT) THEN GOTO Z30+
        Z30
HN 760  RETURN
```

```
HI 800 ? S$,"{3 SPACES}CHANGE ENTRY":? :? L$
NH 805 ? B$(Z11,K):H=ASC(B$(Z11,Z11)):RR=ASC(B
       $(Z3,Z3))-Z64:? D$(RR*Z20-Z19,RR*Z20);
HG 810 ? "19";B$(Z4,Z5);"{3 SPACES}#";B$(6,7);
       "  p.";B$(Z8,Z10)
DA 815 RR=ASC(B$(Z1,Z1))-Z64:? E$(RR*Z20-Z19,R
       R*Z20);
OH 820 IF B$(Z2,Z2)=" " THEN GOTO Z800+Z30
MG 825 RR=ASC(B$(Z2,Z2))-Z64:? F$(RR*Z10-Z9,RR
       *Z10)
CI 830 ? :? L$:? "{DOWN}{4 SPACES}CHANGE OR DE
       LETE OR MENU";
DO 835 GET #Z1,R:IF R<>Z64+Z3 AND R<>Z64+Z4 AN
       D R<>77 THEN ? "{3 UP}":GOTO Z800+Z35
IB 840 IF R=77 THEN GOTO Z700
HG 845 IF R=Z64+Z3 THEN ? "{2 UP}":GOTO Z900-Z
       40
ON 850 ? :? ,"DELETE ITEM (Y/N)?";:GET #Z1,R:I
       F R<>89 THEN GOTO Z700
KH 855 Z$(I*K-J,LEN(Z$))=Z$((I+Z1)*K-J,LEN(Z$)
       ):Z$=Z$(Z1,LEN(Z$)-K):GOTO Z700
CE 860 ? :? ,"  1. TITLE{8 SPACES}":? ,"  2. S
       OURCE":? ,"  3. YEAR":? ,"  4. ISSUE":?
        ,"  5. PAGE":? ,"  6. CATEGORY"
NG 865 ? ,"  7. LANGUAGE":? ,"  8. NO CHANGE":
       ? :? ,"YOUR CHOICE?";:GET #Z1,R:R=R-48:
       ?
AM 870 ON R GOTO Z900-Z20,Z900+Z5,Z900+Z10,Z90
       0+15,Z900+Z20,Z900+25,Z900+Z30,Z700
NM 880 ? :? "TITLE":INPUT T$:IF LEN(T$)>Z40-Z3
        THEN ? "{3 UP}":GOTO Z900-Z20
KD 890 IF LEN(T$)<37 THEN T$(LEN(T$)+Z1)=" ":G
       OTO Z900-Z10
OE 900 GOSUB Z35:B$(Z11,K)=T$:Z$(I*K-J,I*K)=B$
       :GOTO Z800
BC 905 ? S$;:GOSUB Z30:? :? "MAGAZINE";:INPUT
       M$:B$(Z3,Z3)=M$(Z1):GOTO Z900+Z35
PE 910 TRAP X:TRAP Z900+Z10:? :? "YEAR (2):";:
       INPUT Y$:B$(Z4,Z5)=Y$(Z1,Z2):GOTO Z900+
       Z35
GH 915 TRAP X:TRAP 915:? :? "ISSUE (2):";:INPU
       T Y$:B$(6,7)=Y$(Z1,Z2):GOTO Z900+Z35
BE 920 TRAP X:TRAP Z900+Z20:? :? "PAGE (3):";:
       INPUT Y$:B$(Z8,Z10)=Y$(Z1,Z3):GOTO Z900
       +Z35
ME 925 ? S$;:GOSUB Z20:? :? "CATEGORY";:INPUT
       C$:B$(Z1,Z1)=C$(Z1,Z1):GOTO Z900+Z35
GC 930 ? S$;:GOSUB Z20+Z5:? :? "LANGUAGE";:INP
       UT C$:B$(Z2,Z2)=C$(Z1,Z1):GOTO Z900+Z35
```

```
JH 935 Z$(I*K-J,I*K)=B$:GOTO Z8ØØ
DL 1ØØØ DATA ACE,ANALOG,ANTIC,ATARI CONNECTION
        ,BYTE,COMPUTE!,CREATIVE COMPUTING,MICR
        O,MICROCOMPUTING,SOFTLINE
AP 1Ø1Ø DATA SOFTSIDE,OTHER SOURCES
FO 11ØØ DATA ADVENTURE,ARCADE GAME,WORD GAME,O
        THER GAME,EDUCATION,UTILITY,SOUND DEMO
        ,GRAPHICS DEMO
AB 111Ø DATA GRAPHICS/SOUND DEMO,ML DEMO,APPLI
        CATION,REVIEW,OTHER
GG 12ØØ DATA BASIC,BASIC-ML,MICROSOFT,ML,FORTH
        ,PILOT,PASCAL,LOGO,LISP,VISICALC,OTHER
```

Personal Net Worth Statement

John L. Nuss

"Personal Net Worth Statement" will calculate your net worth easily and quickly. The built-in editing functions permit easy changes in any entry. And since the data is saved as part of the program, it will work with tape or disk.

Financial advisors recommend the practice of calculating your personal net worth annually or even more frequently if possible. Doing so enables you to measure the extent to which you are attaining your savings goals for retirement or other purposes. It can also be helpful for estate planning.

This program allows you to calculate your net worth with relative ease. It also summarizes your holdings by categories which you have set up. This should allow you to determine if the mix of your investments is in line with the investment strategy you have set. Best of all is the ease with which you can update your financial data by using an onscreen editing technique.

Changing Categories

Type in the program as listed except for any changes or additions you might want to make to the DATA statements which contain the category names. Remember that you're limited to one category for each letter of the alphabet, so you may need some imagination to come up with meaningful category descriptions. The program as written allows for only 17 categories, because 17 is all that will fit on one screen of the selection menu or the net worth statement. You could change this by providing for a second page for these screens, but since 17 categories are probably more than adequate, no change should be required.

When running the program, you'll first be asked for the current date. The program accepts only numerals for the date input and expects to see two digits for each item (month, day, and year), so you should preface single-digit months and dates with a zero. After receiving an acceptable date, the program will present a menu of possible routines. All that's required is a one-key response, since I tried to minimize the need for the RETURN key as much as possible.

Type 1 to begin entering your assets and liabilities. You will be presented with the category menu from which to select. Type the first letter of your category choice. The data entry/edit screen will then appear for that category. Type the description of your first entry and hit RETURN. An item number and category letter will appear in the REF column, and the description will appear in the description column as entered. You will then be prompted to enter the value and valuation date in the same manner. The valuation date default is the current date, so just hit RETURN if that is satisfactory.

Liabilities (debts) should be entered as negative numbers. For instance, you might enter the market value of your car or home as an asset and the balance of your auto loan or mortgage as a liability. When you have made all the entries for a category, just hit RETURN and you'll be offered the choice of either going to another category or returning to the main menu. Note that because of the amount of room on the screen, you are limited to 13 entries per category.

Editing

If you make an error on an entry or if you later wish to change an amount or description, you need only move the cursor to that item, edit it, and hit RETURN. The corresponding DATA statement will be revised without disturbing any other items on the screen. A minor drawback is that you cannot completely delete items in this manner. However, the value of an item can be set to 0, thereby effectively eliminating it. Or both the value and description can be changed, which, in effect, will replace it with a new item.

There is also the option to delete all your entries. This routine can be accessed from the main menu.

After entering your assets and liabilities, you can review your net worth by requesting the proper routine from the menu screen. The program will then total your entries and display your net worth by category as well as give an overall total. The proportional relationship of each category to the total is also displayed as a percentage.

Personal Net Worth Statement

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
PD 110  OPEN #2,12,0,"E:"
JG 120  DIM P$(8),R$(2),X(2)
```

```
AA 130 READ P
CF 140 READ P$
CH 150 GOSUB 350:REM - GOTO SCREEN CLEAR & TIT
       LE ROUTINE
ML 160 CONT
GN 180 TRAP 180:CLOSE #1
OL 190 POSITION 2,6:? "What is today's date (M
       M/DD/YY)?"
NA 200 GOSUB 550:REM - GOTO DATE ENTERING ROUT
       INE
CH 210 FOR I=1 TO 100:NEXT I
AA 220 DIM CAT$(20),C$(1),D$(13),V$(9),DT$(8),
       S$(38),A$(9),CM$(1),I$(38),V(26),PC(26)
IN 230 S$="{38 SPACES}"
PH 240 CM$=","
MB 250 R$=S$
EI 260 CAT$=S$
LE 270 C$=S$
LG 280 D$=S$
MJ 290 V$=S$
AD 300 DT$=S$
KN 310 A$=S$
LG 320 I$=S$
OM 330 GOTO 660:REM - GOTO TO MAIN MENU
IL 340 DATA 0,00/00/00
PB 350 GRAPHICS 0:SETCOLOR 2,11,0:SETCOLOR 4,1
       1,0:SETCOLOR 1,10,10:POKE 752,1
CO 360 ? "* NET WORTH STATEMENT AS OF ";P$;" *
       {DOWN}":RETURN
IC 370 REM - DOLLAR AND CENTS FORMAT
MA 380 IF ABS(A)>999999 THEN A$=STR$(A):RETURN
BB 390 A$="{6 SPACES}.  ":B=7-LEN(STR$(INT(A))
       )+(A<0)
PF 400 IF A<0 THEN A=-A:A$(B-1,B-1)="-"
BH 410 A$(B,6)=STR$(INT(A))
HA 420 A$(7,9)=STR$(100+INT((A-INT(A))*100+0.0
       5))
JG 430 A$(7,7)="."
EH 440 IF VAL(A$)<0 THEN A=-A
HJ 450 RETURN
AI 460 REM - FORCED READ ROUTINE PART 1
DM 470 POSITION 2,18:? "{6 DEL LINE}"
PD 480 POSITION 2,19:RETURN
AM 490 REM - FORCED READ ROUTINE PART 2
EM 500 ? "CONT"
NC 510 POSITION 2,18
HE 520 POKE 842,13:STOP
GC 530 POKE 842,12:GOSUB 460:RETURN
GM 540 REM - ENTER DATE ROUTINE
PL 550 OPEN #1,4,0,"K":POKE 752,1
```

```
LF 560 POSITION 24,8:? "■■/■■/■■"
PL 570 P$="  /  /  ":TRAP 650
IK 580 FOR I=1 TO 6:J=I+(I>2)+(I>4)
OK 590 IF PEEK(764)=255 THEN 590
PE 600 GET #1,X:P$(J,J)=CHR$(X)
JD 610 IF X<48 OR X>57 THEN 650
GO 620 POSITION 23+J,8:? CHR$(X)
CB 630 NEXT I
HO 640 CLOSE #1:RETURN
JG 650 POP :GOTO 180
LE 660 GOSUB 350
HE 670 CLOSE #1:TRAP 660
KC 680 POSITION 5,3:? "Please choose one of th
      e following:"
CB 690 POSITION 10,5:? "1 - Update or Review D
      ata{4 SPACES}"
KD 700 POSITION 10,7:? "2 - Net Worth Report
      {9 SPACES}"
MB 710 POSITION 10,9:? "3 - Store Data and End
      "
IF 720 POSITION 10,11:? "4 - Erase Data"
NH 730 POSITION 5,15:? "What is your selection
      ?"
JI 740 OPEN #1,4,0,"K"
OG 750 IF PEEK(764)=255 THEN 750
NL 760 GET #1,OP:OP=VAL(CHR$(OP)):CLOSE #1
OC 770 ON OP GOSUB 790,1590,1880,2020
HE 780 GOTO 660
IO 790 REM -UPDATE DATA
LA 800 GOSUB 350
GM 810 POSITION 2,3:? "WHICH CATEGORY?"
IH 820 TRAP 870:RESTORE 8010:I=0
IM 830 READ CAT$:IF CAT$="□" THEN 870
MH 840 I=I+1
MO 850 POSITION 7,5+I:? CAT$;
HC 860 GOTO 830
JM 870 OPEN #1,4,0,"K"
OP 880 IF PEEK(756)=255 THEN 880
KP 890 GET #1,X:CLOSE #1:GOSUB 350
IC 900 POSITION 1,2:? "{=}REF {=} DESCRIPTION
      {=}  VALUE  {=}  DATE  {=}"
FJ 910 COLOR 14:PLOT 1,1:DRAWTO 39,1
FN 920 COLOR 13:PLOT 2,3:DRAWTO 38,3
IK 930 COLOR 124:PLOT 1,2:DRAWTO 1,17
EG 940 PLOT 6,3:DRAWTO 6,17
JP 950 PLOT 20,3:DRAWTO 20,17
KC 960 PLOT 30,3:DRAWTO 30,17
LF 970 PLOT 39,2:DRAWTO 39,18
JN 980 RESTORE 9000:I=0:J=0:T=0:TRAP 1100
MN 990 I=I+1
```

```
OD 1000 J=J+1
GH 1010 READ C$,D$,V$,DT$
PA 1020 IF C$="[E]" THEN 1100
MO 1030 IF C$<>CHR$(X) THEN 1000
PI 1040 POSITION 2+(J<100)+(J<10),3+I:? J
CD 1050 POSITION 5,3+I:? C$
CH 1060 POSITION 7,3+I:? D$
GG 1070 POSITION 21,3+I:? V$
KK 1080 POSITION 31,3+I:? DT$
KF 1090 GOTO 990
LA 1100 REM - DATA UPDATE
MP 1110 POKE 752,1:COLOR 13:PLOT 1,18:DRAWTO 3
        9,18:PLOT 39,18
OH 1120 TRAP 1220:RESTORE 8010
FO 1130 READ CAT$:IF CAT$(1,1)=CHR$(X) THEN 11
        50
ME 1140 GOTO 1130
FP 1150 POSITION 2,17:? CAT$:POSITION 31,17:?
        " TOTAL"
JO 1160 A=0:RESTORE 9001
LE 1170 READ C$:IF C$="[E]" THEN 1200
OA 1175 READ D$,V$,DT$
GO 1180 IF C$=CHR$(X) THEN A=A+VAL(V$)
MN 1190 GOTO 1170
NN 1200 GOSUB 370
AO 1210 POSITION 21,17:? A$
DA 1220 TRAP 1220:POSITION 2,20
CD 1230 ? "TYPE DESCRIPTION OF NEW ITEM OR MOV
        E  CURSOR TO EDIT AN ITEM ON THE SCREE
        N.{14 SPACES}{=}HIT [RETURN]{=}"
FI 1240 POSITION 1,22:POKE 752,0:? " ";
MF 1250 OPEN #1,4,0,"K"
GK 1260 GET #1,R:CLOSE #1:IF R=155 THEN 1530
GK 1270 IF (R>64 AND R<91) THEN 1380
GA 1280 IF (R<28 AND R>31) THEN 1150
KO 1290 INPUT #2,I$:POKE 752,1
NF 1300 R=VAL(I$(1,3))
PD 1310 D$=I$(6,18)
DD 1320 V$=I$(20,28)
HH 1330 DT$=I$(30,37)
LO 1340 C$=I$(4,4)
OE 1350 GOSUB 470
JB 1360 ? "{DOWN}";9000+R;" DATA ";C$;CM$;D$;C
        M$;V$;CM$;DT$:GOSUB 500
MG 1370 GOTO 1100
MB 1380 POSITION 2,22:PUT #2,R:D$=S$:D$(1,1)=C
        HR$(R)
FO 1385 IF I>13 THEN POSITION 2,19:? "
        {3 [SPACES]}[NO MORE ROOM IN THIS CATEGOR
        Y ]";:GOTO 1100
```

```
KM 1390 FOR K=2 TO 13:GET #2,R:IF R=155 THEN P
        OP :GOTO 1410
OF 1400 D$(K,K)=CHR$(R):NEXT K
EL 1410 TRAP 1410:GOSUB 1580
KH 1420 R=P+1:POSITION 2+(R<100)+(R<10),I+3:?
        R;CHR$(X);
CI 1430 POSITION 7,I+3:? D$
GB 1440 POSITION 2,20:? "ENTER VALUE:";:INPUT
        #2,A:GOSUB 370:V$=A$
GI 1450 POSITION 21,I+3:? V$
JD 1460 GOSUB 1580:POSITION 2,20:? "ENTER VALU
        ATION DATE (MM/DD/YY):";:INPUT #2,DT$
DE 1470 IF DT$="" THEN DT$=P$
KO 1480 POSITION 31,I+3:? DT$
CF 1490 P=P+1:I=I+1:POKE 752,1:GOSUB 470
MO 1500 ? "{DOWN}";9000+P;" DATA ";CHR$(X);CM$
        ;D$;CM$;V$;CM$;DT$:GOSUB 500
PO 1510 POSITION 2,22
DC 1520 IF I<14 THEN 1100
CN 1530 GOSUB 1580:POSITION 2,20:? "PRESS [SEL
        ECT] TO CHANGE CATEGORIES."
LO 1540 POSITION 2,22:? "PRESS [OPTION] WHEN A
        LL DATA ENTERED."
BD 1550 IF PEEK(53279)=5 THEN 800
FK 1560 IF PEEK(53279)=3 THEN RETURN
NB 1570 GOTO 1550
KO 1580 POKE 752,1:POSITION 2,20:? "
        {4 DEL LINE}":RETURN
IK 1590 REM - NET WORTH CALCULATION
NP 1600 GOSUB 350
JN 1610 FOR I=0 TO 26:V(I)=0:PC(I)=0:NEXT I
LH 1620 RESTORE 9001
MB 1630 READ C$:IF C$="[E]" THEN 1680
OB 1635 READ D$,V$,DT$
HB 1640 C=ASC(C$)-64
LG 1650 V(C)=V(C)+VAL(V$)
JB 1660 V(0)=V(0)+VAL(V$)
NB 1670 GOTO 1630
BB 1680 POSITION 4,2:? "CATEGORY{12 SPACES}AMOU
        NT{3 SPACES}PERCENT";
KN 1690 RESTORE 8010:I=0
MI 1700 TRAP 1800
OE 1710 READ CAT$:IF CAT$="[E]" THEN 1800
OE 1720 C=ASC(CAT$(1,1))-64
CC 1730 I=I+1:IF I>16 THEN STOP
LL 1740 POSITION 2,3+I:? CAT$
IJ 1750 A=V(C):GOSUB 370
JD 1760 POSITION 22,3+I:? A$;
BK 1770 PC(C)=INT(100*V(C)/V(0))
```

```
KO 1780 POSITION 34+(PC(C)<100)+(PC(C)<10),3+I
        :? PC(C);"%";
ND 1790 GOTO 1710
MG 1800 POSITION 7,5+I:? "TOTAL";
HD 1810 A=V(0):GOSUB 370
FH 1820 POSITION 22,5+I:? A$
KO 1830 PC(0)=100
OK 1840 POSITION 34,5+I:? PC(0);"%";
AK 1850 POSITION 2,22:? "PRESS [OPTION] TO RES
        UME."
IJ 1860 IF PEEK(53279)<>3 THEN 1860
LA 1870 RETURN
OJ 1880 GOSUB 350
ND 1890 ? "SAVE PROGRAM AND DATA FOR NEXT SESS
        ION"
GK 1900 POSITION 5,5:? "1.  TURN OVER TAPE AND
         REWIND."
GB 1910 POSITION 5,7:? "{5 SPACES}SAVING COUNT
        AND DATE..."
OG 1920 GOSUB 460
AB 1930 ? "{DOWN}";340;" DATA ";P;CM$;P$
OD 1940 GOSUB 500
PM 1950 POSITION 5,7:? "2.  PRESS [RECORD] AND
         [PLAY]."
CL 1960 POSITION 5,9:? "3.  PRESS [RETURN] AFT
        ER BUZZER{11 SPACES}RINGS TWICE."
NL 1970 TRAP 1990
KL 1980 LPRINT
EF 1990 CSAVE
DL 2000 POSITION 5,12:? "4.  REWIND TAPE."
PN 2010 POKE 752,0:END
PP 2020 REM - OBSOLETE DATA DELETE
NN 2030 GOSUB 350
OF 2040 POSITION 2,10:? "Do you REALLY want to
         erase your data?"
ME 2050 OPEN #1,4,0,"K"
DP 2060 IF PEEK(756)=255 THEN 2060
HP 2070 GET #1,R:CLOSE #1
CD 2080 IF CHR$(R)<>"Y" THEN RETURN
AM 2090 POSITION 2,10:? S$,"ERASING DATA..."
MA 2100 FOR LINE=9001 TO 9000+P
NO 2110 GOSUB 460
MI 2120 ? "{DOWN}";LINE
NL 2130 GOSUB 500
CD 2140 NEXT LINE
JG 2150 P=0:GOSUB 460:? "{DOWN}";340;" DATA ";
        P;CM$;P$:GOSUB 500
KJ 2160 RETURN
LP 8000 REM -CATEGORIES
```

```
FJ 8010 DATA BONDS
AD 8020 DATA CASH
EO 8030 DATA EQUITIES
PJ 8040 DATA FIXED TERM DEPOSITS
GJ 8050 DATA OTHER
NB 8060 DATA PERSONAL PROPERTY
ND 8070 DATA REAL ESTATE
LJ 8990 DATA 
LK 9990 DATA 
MI 10000 END
```

Data Management System

An Atari Database with Application Generation Features

Ronald Marcuse

This generic database system can be used for many applications. The next article, "Mailing Label Utility," is an example of its adaptability. Requires 32K RAM and a disk drive.

My initial excursion into the world of microcomputers began soon after I purchased my Atari 800. It took perhaps a month for my data processing (meaning file processing) background to emerge from the myriad of games that I was coding and playing on the Atari, but this was inevitable. Don't misunderstand me—the games are fun, but I personally wanted more from the computer than just zapping Zylon Raiders or hitting a quarter-inch baseball across a 25-inch screen.

In the ensuing month, information processing systems materialized in all shapes and sizes, covering such important data as telephones/addresses, appointments, and the Star Wars figure collection of my offspring (which turned out to be the largest of my numerous files). When I saw myself coding a catalog system to keep track of all the other catalog/information systems, I vowed to find a better way.

This leads us to the subject of the article—an Atari data management/database system. There is a great deal of similarity between different systems that are designed to track different data; in fact, the similarities usually far outweigh the differences. The variance in file/record attributes doesn't require a markedly different approach in getting to and from the storage medium. If you were to store these file/record attributes in, say, a data dictionary, you could use the dictionary to supply the parameters to drive generalized file/screen/printer IO routines. You'd need only to specify the attributes to generate the application. An information system needed to track paper clips could be implemented in minutes. That, in a nutshell, is the concept.

The Dictionary

The data dictionary function, as the front-seat driver of the entire vehicle, deserves clarification first. Its primary function is to create and monitor the individual dictionary records containing the attributes of the database on that particular disk. Within this scheme, the ADD logic resides in lines 8000–8630, the INDEX begins at line 6500 and continues with the subroutine in 9200–9320, the INQUIRY or LIST is in lines 7000–7080, and the DELETE in 7500–7640. Subroutines shared by one or more of these modules are PAUSE, 6300; CREATE FILESPEC, 6310–6320; PRINT DIC RECORD, 8750–8870; LOAD VARIABLES FROM DIC RECORD, 9000–9050; and READ DIC RECORD, 9200–9320. Additionally, located from 9801 to 9846 are short subroutines of the variety GOSUB NNNN+I utilized in moving headers and data elements to and from larger strings. Tables 1 and 2 list the format of the dictionary record (REC$) and other variables.

In operation, when you're creating a new application/database, you would specify a title (25 characters maximum), data set name (8 chararacters maximum), and the number of data elements (6 maximum) within the proposed record. A loop (lines 8110–8200) would step you through the process of entering the heading (12 characters), size (30 maximum), and editing requirements of each data element.

Editing criteria of N (Numeric), D (Date), and $ (Dollar) may be requested. I've imposed a nominal limit of six data elements with 30 characters maximum for any one element and a total record length of 120 bytes (for no other reason than I happened to like those numbers on that day). If the data dictionary has *not* been initialized on the disk, you'll be prompted during the add routine for permission to create it (lines 8610–8630).

There is a conspicuous absence of any UPDATE functions on the data dictionary menu. This was intentional. Changing the attributes of a given database *after* it has been created and fed huge mounds of raw data may be detrimental to your health. (Picture the File Manager going after a 50-byte record that it now believes is 80 bytes long.)

The Manager

The File Management system, with its menu residing in lines 500–670, requires that the operator first select one of the databases on that disk (lines 750–886); an index is made available by entering a single character I. The selection process reads the data dictionary file and, upon finding your request (filename is the key), moves the dictionary record into the applicable variables. Once loaded, these variables control the execution of the other File Management modules. These are ADD RECORD (lines 1000–1200), LIST RECORD/INQUIRY (2000–2440), and UPDATE RECORD (3000–3310). Note that typing E on any function menu will return program control to the module one step higher in the network. Tables 1 and 2 list the variables used in these procedures. A SEARCH procedure is generated in lines 4000–4210 that can be utilized by both the INQUIRY/LIST and UPDATE functions. You need only to specify the field number and enter the value of the characters to be used. By entering fewer characters than the size of the field, you can perform a generic search, extracting, in turn, all records that satisfy those requirements.

The previously mentioned input editing is performed in lines 4400–4590. Selecting the output medium (screen or printer) is done in lines 680–730. ASCII control characters (for an EPSON printer), based on the attributes of the resident Database, are generated and sent to the device, thereby setting character size and tabs. The EPSON MX-80 has software-selectable print lines of from 40 to 132 positions. Other printers without this feature may require a modified approach or a limitation in your record size.

The Sort

The SORT routine (Program 2) is an example of the "selection and exchange" variety. Logic similar to that contained in the database manager allows you to select the file to sort or list the index of that particular disk (lines 4000–4340). The choice of sort key and whether "ascending" or "descending" occurs is in lines 5000–5070. The file is input and stored in the DIMensioned string X$ in lines 5100–5160. The variables I, J, and K are used as pointers during the loops through X$ as follows: I represents the sorted/not sorted boundary X$, J is the current comparison with the previously selected lowest or

highest value (DS$), and K is the location of this previous value.

An exchange between X$(K,K+L) and X$(I,I+L) occurs if an unsorted condition is detected on any loop. The sort terminates when the sorted boundary is equal to the size of X$. Then, X$ is written to the disk, the original file is deleted, and the new file is renamed.

At this point it would probably be helpful for us to single-step through the code, but we have neither the time nor the space to do this. But we can examine at least one aspect of the structure, notably the interaction of the subroutines that perform the input and handling of an existing data dictionary record. Entry to the READ DICTIONARY routine (9200–9310) is from several possible locations: LIST INDEX, both FMS and DIC (via line 6500); DICTIONARY INQUIRY/LIST (line 7080); and SELECT DATABASE (line 800). Two variables (loaded by the calling module) enable this one subroutine to perform several functions. T represents the "type" of request (0—index; 1—loop through to find and load data dictionary record with key equal to value in FILD$; 2—list single data dictionary record with key FILD$; 9—browse through all data dictionary records). The variable R returns control back to a specific line number in the event of a DISK I/O error being TRAPped to line 9500.

As the program executes each line within the routine, program control is modified based on the value of T during that call. Options include loading the full record into all of the dictionary variables (9000–9050), listing the dictionary record to the screen (8750–8870), chatting with the operator (9280–9290), and a "quickie" index. The subroutines mentioned above, as well as the balance of the program, work in a similar manner.

Typing It In

Both program listings contain unprintable ASCII characters used for screen and printer control. Be sure to check the Appendices for how to enter these characters. Because the two programs call each other by name, you must save the two as "D:DMSDB" and "D:DMSSORT".

Table 1. Data Dictionary Record (REC$)

Variable Name	Position Within REC$	Description
FILD$	1–8	Application filename
APP$	9–33	Application title
DL(1) : DL(6)	34–45	Lengths of 6 (max) data elements within record (2 char ea)
HD1$: HD6$	46–117	Heading titles (12 char)
DE(1) : DE(6)	118–123	Editing criteria for elements 0=Alphanumeric 1=Numeric (N) 2=Date(D) 3=Dollar($)
	124	Delimiter (''*'')

Table 2. Other DMS/DB Variables

Name	Size	Description
RL	(var)	Application data set record length
NE	(var)	Number of data elements
DM$	8	DMX/DB filespec (D:DMS.DB)
FIL$	14	Application filespec (D:filename.DB)
I$	1	Operator response to questions
IN$	31	General input of data and temp stor
SF		
SL		
SS	(var)	Used by search function
SE		
SV$		
FD1$: FD6$	(var)	Application data elements
T	—	Tran type passed to I/O routines
EOF	—	End of file counter
I,J,K,L,N	—	Temp stor for looping, length, etc.
ERR	—	Input data error flag
R	—	Error message return line #
P	—	Printer/screen indicator (1=Print)
S$,X$,P$,N$	—	Messages, prompts

Program 1. DMSDB

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
MJ 40 POKE 82,0:POKE 83,39:GOTO 100
KG 50 DIM APP$(26),FIL$(14),FILD$(9),HD1$(12),
      HD2$(12),HD3$(12),HD4$(12),HD5$(12)
IL 60 DIM HD6$(12),I$(1),REC$(132),IN$(31),DM$
      (8),DL(6),DE(6)
GN 65 DIM S$(15),X$(5),P$(24),N$(16):N$="RECOR
      D NOT FOUND":DM$="D:DMS.DB"
DE 70 P$="PRESS RETURN TO CONTINUE":S$="{DOWN}
      SELECT OPTION:":X$="ERROR"
EA 75 FOR I=1 TO 6:DL(I)=0:DE(I)=0:NEXT I
BJ 80 REC$(1)=" ":REC$(132)=" ":REC$(2)=REC$:R
      ETURN
JB 100 GRAPHICS 0:? "{DOWN} DATA MANAGEMENT SY
       STEM - VER 1.2":CLR :DIM I$(1)
ND 110 ? "{DOWN}{5 SPACES}PRIMARY OPTION MENU"
       :POKE 16,64:POKE 53774,64
IE 120 ? "{DOWN} D - DATA DICTIONARY FUNCTIONS
       ":? "{5 SPACES}(DEFINE NEW APPLICATION
       & DATA)"
BK 130 ? "{DOWN} R - EXEC FILE MANAGEMENT SYST
       EM":? "{5 SPACES}(ACCESS EXISTING DATA
       BASE)"
DD 150 ? "{DOWN} S - CALL DMS SORT UTILITY":?
       "{DOWN} E - END (TERMINATE DMS)"
KF 160 ? :? "SELECT OPTION:";:INPUT I$:IF I$="
       R" THEN 500
EP 180 IF I$="D" THEN 6000
HE 200 IF I$="S" THEN RUN "D:DMSSORT"
BG 210 IF I$="E" THEN GRAPHICS 0:END
KI 220 ? "INVALID":GOTO 160
FO 490 ? "{DOWN}SELECT DATA SET 1ST":GOSUB 630
       0
MI 500 ? "{CLEAR}{DOWN}{3 SPACES}DMS FILE MANA
       GEMENT MENU"
GN 520 IF NE<>0 THEN ? :? ,APP$:IF P=1 THEN ?
       ,"PRINTER SELECTED"
KC 540 ? "{DOWN} S - SELECT DATA BASE":? " A -
        ADD"
DH 580 ? " L - LIST / INQUIRY":? " U - UPDATE"
ML 590 ? " P - PRINTER (SCREEN DEFAULT)":? " E
        - END (RETURN TO DMS MENU)"
JD 600 ? :? "SELECT OPTION:";:INPUT I$:IF I$="
       E" THEN 100
DD 620 IF I$="S" THEN 750
EH 630 IF I$="A" THEN 1000
FE 640 IF I$="L" THEN 2000
```

```
FP 650  IF I$="U" THEN 3000
DG 660  IF I$="P" THEN 680
LA 670  ? "[illegible]":GOTO 600
PJ 680  IF NE=0 THEN 490
GH 690  ? :? "TYPE: P-PRINTER S-SCREEN":INPUT I
        $:IF I$<>"P" THEN P=0:GOTO 500
AF 700  P=1:IN$="{R}{ESC}D{5 SPACES}{,}":IF RL>
        55 OR N>4 THEN IN$(1,1)="{0}"
HF 710  L=1:FOR I=2 TO NE:J=DL(I-1):IF J<12 THE
        N J=12
JL 720  L=L+J+2:IN$(I+2,I+2)=CHR$(L):NEXT I
GF 730  R=690:TRAP 9510:LPRINT IN$:TRAP 40000:G
        OSUB 80:GOTO 500
DE 750  ? "{CLEAR}{DOWN}{4 SPACES}DATA BASE SEL
        ECTION":CLR :GOSUB 50:R=760
OI 760  ? "{DOWN}ENTER DATA SET NAME (I FOR IND
        EX)":? :? P$;" (END)"
NO 770  INPUT FILD$:IF LEN(FILD$)=0 THEN 500
HJ 780  IF LEN(FILD$)=1 AND FILD$="I" THEN GOSU
        B 6500:GOTO 760
LD 800  T=1:GOSUB 9200:IF EOF=0 THEN 500
GM 820  ? "{DOWN}DATA SET LOADED":GOSUB 6300
GB 830  RL=1:FOR N=1 TO NE:GOSUB 880+N:RL=RL+DL
        (N):NEXT N
DG 850  DIM DS$(RL),SV$(30):FOR I=1 TO RL:DS$(I
        ,I)=" ":NEXT I:GOTO 500
PE 881  DIM FD1$(DL(N)):RETURN
PG 882  DIM FD2$(DL(N)):RETURN
PI 883  DIM FD3$(DL(N)):RETURN
PK 884  DIM FD4$(DL(N)):RETURN
PM 885  DIM FD5$(DL(N)):RETURN
PO 886  DIM FD6$(DL(N)):RETURN
BM 1000  IF NE=0 THEN 490
DC 1005  R=500:TRAP 9500:XIO 36,#2,0,0,FIL$:OPE
         N #2,9,0,FIL$:TRAP 40000
AI 1010  ? "{CLEAR}{DOWN}{3 SPACES}";APP$:? "
         {DOWN} TO ADD RECORD, ENTER:"
JO 1040  FOR N=1 TO NE
FK 1050  GOSUB 9820+N:IF DE(N)<>2 THEN ? " (";D
         L(N);" CHAR MAX)":GOTO 1070
PA 1060  ? " (6 CHAR - MMDDYY)"
ML 1070  INPUT IN$:L=LEN(IN$):IF N=1 AND L=0 TH
         EN 1180
GI 1075  GOSUB 4400:IF ERR<>0 THEN 1070
NJ 1080  GOSUB 9810+N:NEXT N:GOSUB 4000
KM 1170  ? #2;DS$:? "{DOWN}TRANSACTION ACCEPTED
         "
OA 1180  ? "{DOWN}TYPE E FOR FMS MENU":? P$
PC 1190  INPUT I$:IF I$<>"E" THEN 1010
IM 1200  CLOSE #2:XIO 35,#2,0,0,FIL$:GOTO 500
```

```
BN 2000 IF NE=0 THEN 490
NB 2020 ? "{CLEAR}{DOWN}  ";APP$:? "{DOWN}  IN
        QUIRY / LIST":IF P=1 THEN ? ,,"{UP}PRI
        NTER SELECTED"
HD 2030 ? "{DOWN} A - LIST ALL RECORDS":? " S
        - SEARCH ON KEY"
KA 2050 ? " E - END (RETURN TO MENU)":T=9
LA 2060 ? S$;:INPUT I$:IF I$="E" THEN 500
HM 2080 IF I$="A" THEN 2300
HN 2090 IF I$="S" THEN T=1:? "{CLEAR}":GOSUB 4
        100:GOTO 2300
LJ 2100 ? X$:GOTO 2060
GO 2300 I=0:EOF=0:R=2000:TRAP 9500:OPEN #2,4,0
        ,FIL$
LK 2310 TRAP 2420:INPUT #2,DS$:TRAP 40000:IF T
        >8 THEN 2330
AI 2320 IF SV$<>DS$(SS,SE) THEN 2310
MO 2330 EOF=EOF+1:IF P<>1 THEN 2400
PA 2340 L=1:J=1:IF I>0 THEN 2370
KP 2350 R=2440:TRAP 9510:LPRINT "{I}{N}";APP$;
        "{T}{K}"
FC 2360 LPRINT HD1$;"{I}";HD2$;"{I}";HD3$;"
        {I}";HD4$;"{I}";HD5$;"{I}";HD6$;"{K}"
GD 2370 FOR N=1 TO NE:K=DL(N):IF K<12 THEN K=1
        2
GL 2375 REC$(J,J+DL(N)-1)=DS$(L,L+DL(N)-1):L=L
        +DL(N):J=J+K+2:NEXT N
OO 2380 TRAP 9510:LPRINT REC$(1,J):I=I+1:IF I<
        55 THEN 2310
CA 2390 I=0:LPRINT "{L}":GOTO 2310
HB 2400 GOSUB 4300:? :? P$;" (E TO STOP)"
PC 2410 INPUT I$:IF I$<>"E" THEN 2310
HO 2420 IF EOF=0 THEN ? N$
PH 2430 IF EOF>0 THEN ? "REC COUNT- ";EOF
HD 2435 IF P=1 THEN LPRINT "{L}"
EP 2440 CLOSE #2:GOSUB 6300:GOTO 2020
BO 3000 IF NE=0 THEN 490
FL 3020 ? "{CLEAR}{DOWN}  ";APP$:? "{DOWN} UPD
        ATE / DELETE";:GOSUB 4100
PJ 3040 R=500:TRAP 9500:OPEN #2,4,0,FIL$:OPEN
        #3,8,0,"D:TEMP":EOF=0
DF 3060 TRAP 3250:INPUT #2,DS$:TRAP 40000
AN 3080 IF SV$<>DS$(SS,SE) THEN ? #3;DS$:GOTO
        3060
AL 3100 GOSUB 4300
BC 3120 ? "{DOWN}ENTER: FIELD # TO UPDATE; D T
        O DELETE":? "PRESS RETURN TO WRITE REC
        "
BE 3130 EOF=EOF+1:INPUT I$:IF LEN(I$)=0 THEN ?
         #3;DS$:GOTO 3060
```

```
IF 3135 IF I$="D" THEN 3060
MJ 3140 TRAP 3120:N=VAL(I$):TRAP 40000:IF N<1
        OR N>NE THEN 3120
MP 3150 ? "{DOWN}ENTER NEW ";:GOSUB 9820+N:? "
         "
JK 3170 INPUT IN$:L=LEN(IN$):GOSUB 4400:IF ERR
        <>0 THEN 3150
EM 3180 GOSUB 9810+N:GOSUB 4000:GOTO 3100
HI 3250 FOR I=1 TO LEN(IN$):IN$(I,I)=" ":NEXT
        I:IN$(1,7)="D:TEMP,"
OL 3260 IN$(8,7+LEN(FIL$)-2)=FIL$(3)
II 3270 CLOSE #2:CLOSE #3:XIO 36,#2,0,0,FIL$:X
        IO 33,#2,0,0,FIL$
DK 3280 XIO 32,#3,0,0,IN$:XIO 35,#3,0,0,FIL$
IE 3290 IF EOF=0 THEN ? N$
MA 3300 ? "{DOWN}MORE UPDATES? (Y OR N)":INPUT
         I$:IF I$="Y" THEN 3000
JF 3310 GOTO 500
DH 4000 FOR I=1 TO LEN(DS$):DS$(I,I)=" ":NEXT
        I
FE 4010 L=1:FOR N=1 TO NE:GOSUB 9840+N:DS$(L,L
        +DL(N)-1)=IN$
GF 4020 L=L+DL(N):NEXT N:DS$(RL)="*":RETURN
KN 4100 ? " SELECT KEY:":? :FOR I=1 TO NE:? "
        ";I;" ";
EM 4110 GOSUB 9820+I:? " ":NEXT I:? :IF SF=0 T
        HEN 4140
HB 4120 ? "PRESS RETURN FOR KEY:":? " 1ST ";SL
        ;" POS OF ";
PG 4130 GOSUB 9820+SF:? " (";SV$;")":? " ":? "
         OR, ";
HF 4140 ? "ENTER KEY FIELD #";
JA 4160 TRAP 4200:INPUT SF:TRAP 40000:IF SF<1
        OR SF>NE THEN ? X$:GOTO 4140
AL 4170 ? "ENTER VALUE OF ";:GOSUB 9820+SF:? "
         (1-";DL(SF);")"
JB 4180 INPUT SV$:SL=LEN(SV$):IF SL<1 OR SL>DL
        (SF) THEN ? X$;" LEN":GOTO 4170
PF 4190 SS=1:IF SF>1 THEN FOR I=2 TO SF:SS=SS+
        DL(I-1):NEXT I
ID 4200 IF SF=0 THEN ? X$:GOTO 4160
IE 4210 SE=SS+SL-1:RETURN
EO 4300 L=1:FOR N=1 TO NE:IN$=DS$(L,L+DL(N)-1)
MD 4310 GOSUB 9810+N:L=L+DL(N):NEXT N
LC 4320 ? "{CLEAR}{DOWN}  ";APP$:? :FOR N=1 TO
         NE
LD 4330 ? N;" ";:GOSUB 9820+N:GOSUB 9840+N:? "
         ";IN$:NEXT N:RETURN
IA 4400 ERR=0:IF L<1 OR L>DL(N) THEN ERR=1
```

```
AP 4410 ON DE(N) GOSUB 4500,4520,4560:IF ERR<>
        0 THEN ? X$
KK 4420 RETURN
NP 4500 FOR I=1 TO L:J=ASC(IN$(I,I)):IF J<48 O
        R J>57 THEN ERR=1
GM 4510 NEXT I:RETURN
FP 4520 GOSUB 4500:IF L<>6 THEN ERR=1
HO 4530 IF IN$(1,2)<"01" OR IN$(1,2)>"12" THEN
         ERR=1
II 4540 IF IN$(3,4)<"01" OR IN$(3,4)>"31" THEN
         ERR=1
KO 4550 RETURN
DA 4560 IF IN$(L-2,L-2)<>" " THEN ERR=1
OH 4570 TRAP 4590:I=VAL(IN$(1,L)):TRAP 40000
LB 4580 RETURN
GN 4590 ERR=1:GOTO 4580
EG 6000 ? "{CLEAR}{DOWN}{3 SPACES}DMS DATA DIC
        TIONARY MENU":CLR :GOSUB 50
CE 6050 ? "{DOWN} A - ADD NEW APPLICATION / DA
        TA BASE"
JE 6060 ? " I - LIST DICTIONARY INDEX":? " L -
         LIST CURRENT DB DESCRIP"
JN 6080 ? " D - DELETE DATA BASE":? " E - END
        (RETURN TO DMS MENU)"
GI 6100 ? :? S$;:INPUT I$
HO 6120 IF I$="A" THEN 8000
KK 6130 IF I$="I" THEN R=6000:GOSUB 6500:? :?
        P$;:INPUT I$:GOTO 6000
IK 6140 IF I$="L" THEN 7000
II 6150 IF I$="D" THEN 7500
EP 6160 IF I$="E" THEN 100
MD 6170 ? X$:GOTO 6100
HM 6300 FOR I=1 TO 150:NEXT I:RETURN
BK 6310 FOR L=1 TO LEN(FILD$):IF FILD$(L,L)<>"
         " THEN NEXT L
KF 6320 L=L-1:FIL$(1,2)="D:":FIL$(3,2+L)=FILD$
        (1,L):FIL$(3+L)=".DB":RETURN
HJ 6500 ? "{CLEAR}{DOWN}  DATA SET INDEX":? :T
        =0:GOSUB 9200:RETURN
LN 7000 ? "{CLEAR}{DOWN}  DATA DICTIONARY INQU
        IRY":? "{DOWN} A - ALL FILES":T=9
LN 7020 ? " S - SINGLE FILE":? " E - END (RETU
        RN TO MENU)"
OD 7030 ? S$;:INPUT I$:IF I$="E" THEN 6000
IH 7040 IF I$="A" THEN 7080
EL 7050 IF I$="S" THEN ? "ENTER FILE NAME":T=2
        :GOTO 7070
MF 7060 ? X$:GOTO 7030
EG 7070 INPUT FILD$:IF LEN(FILD$)=0 THEN 7060
OE 7080 R=7000:GOSUB 9200:GOTO 7000
```

```
NJ 7500 ? "{CLEAR}{DOWN} TO DELETE DICTIONARY
        ELEMENT AND":? " RELATED FILE, TYPE FI
        LE NAME:"
KB 7510 ? P$;" (CANCEL)":INPUT FILD$:IF LEN(FI
        LD$)=0 THEN 6000
ON 7520 R=7500:TRAP 9500:OPEN #2,4,0,DM$:OPEN
        #3,8,0,"D:TEMP":EOF=0
IB 7530 TRAP 7600:INPUT #2,REC$:TRAP 40000
HK 7540 IF FILD$<>REC$(1,LEN(FILD$)) THEN ? #3
        ;REC$:GOTO 7530
CP 7550 EOF=EOF+1:GOSUB 9000:GOSUB 8750
EK 7560 ? "TYPE D TO DELETE";:INPUT I$:IF I$<>
        "D" THEN ? "SAVED":? #3;REC$:GOTO 7530
HO 7570 ? "DELETED":XIO 36,#4,0,0,FIL$:XIO 33,
        #4,0,0,FIL$:GOTO 7530
ID 7600 IF EOF=0 THEN ? N$
PG 7610 CLOSE #2:CLOSE #3:XIO 36,#2,0,0,DM$:XI
        O 33,#2,0,0,DM$
PD 7620 XIO 32,#3,0,0,"D:TEMP,DMS.DB":XIO 35,#
        3,0,0,DM$
NA 7640 GOTO 6000
AG 8000 PRINT "{CLEAR}{DOWN}  ADD TO DATA DICT
        IONARY":? "{DOWN} ENTER APPLICATION NA
        ME (1 TO 25 CHAR)"
JA 8010 INPUT APP$:L=LEN(APP$):IF L<1 OR L>25
        THEN ? X$:GOTO 8010
IO 8020 ? "ENTER FILE NAME (1 TO 8 CHAR)":? "
         FILESPEC WILL BE 'D:XXXXXXXX.DB'"
OB 8030 INPUT FILD$:L=LEN(FILD$):IF L<1 OR L>8
         THEN ? X$:GOTO 8030
AF 8040 IF FILD$(1,1)<"A" OR FILD$(1,1)>"Z" TH
        EN ? X$:GOTO 8030
BI 8060 GOSUB 6310
LM 8080 ? "{DOWN} (MIN REC LEN=10, MAX=120)":?
         "{DOWN}ENTER # OF DATA ELEMENTS IN RE
        C (2-6)"
JK 8090 TRAP 8080:INPUT NE:TRAP 40000
CO 8100 IF NE<2 OR NE>6 THEN ? X$:GOTO 8080
HO 8110 RL=0:FOR I=1 TO NE:? "{CLEAR}{DOWN}  F
        OR DATA ELEMENT # ";I;" OF ";NE;" ENTE
        R:":?
LL 8120 ? "HEADING (1-12)":INPUT IN$:L=LEN(IN$
        ):IF L<1 OR L>12 THEN ? X$:GOTO 8120
JF 8125 GOSUB 9800+I
FF 8130 ? "ELEMENT LENGTH (1 TO 30)":? "  (TOT
         REC LEN IS ";RL;")"
MA 8140 TRAP 8130:INPUT L:TRAP 40000:IF L<=0 O
        R L>30 THEN ? X$:GOTO 8130
KL 8150 DL(I)=L:RL=RL+L
PL 8160 ? "EDITING? (N:NUMERIC, D-DATE, $:DOLL
        AR)":? "RETURN TO SKIP"
```

```
FM 8170 INPUT I$:DE(I)=0:IF I$="N" THEN DE(I)=
        1
FD 8180 IF I$="D" THEN DE(I)=2
DF 8190 IF I$="$" THEN DE(I)=3
FC 8200 NEXT I
KA 8220 IF RL<10 OR RL>120 THEN ? "REC LEN=";R
        L;"  ";X$:GOTO 8080
MF 8300 REC$(1,8)=FILD$:REC$(9,33)=APP$
DJ 8320 FOR I=1 TO 6:REC$(32+I*2,33+I*2)=STR$(
        DL(I))
PE 8340 REC$(117+I,117+I)=STR$(DE(I)):NEXT I
JF 8360 REC$(46,57)=HD1$:REC$(58,69)=HD2$:REC$
        (70,81)=HD3$
PC 8380 REC$(82,93)=HD4$:REC$(94,105)=HD5$:REC
        $(106,117)=HD6$:REC$(124)="*"
CJ 8390 GOSUB 8750:? "{DOWN}TYPE Y TO CREATE D
        ATA BASE"
AL 8400 INPUT I$:IF I$<>"Y" THEN 6000
OD 8420 TRAP 8600:XIO 36,#2,0,0,DM$:OPEN #2,9,
        0,DM$
LC 8430 ? #2;REC$:CLOSE #2:XIO 35,#2,0,0,DM$:T
        RAP 40000
IB 8450 ? "DATA BASE CREATED"
HL 8460 OPEN #3,8,0,FIL$:CLOSE #3:XIO 35,#3,0,
        0,FIL$:GOTO 6000
CP 8600 STATUS #2,ST:IF ST<>170 THEN R=8390:GO
        TO 9500
OI 8610 ? DM$;" NOT ON DISK":? "TYPE Y TO INIT
        IALIZE"
AP 8620 INPUT I$:IF I$<>"Y" THEN 6000
DL 8630 OPEN #2,8,0,DM$:GOTO 8430
OP 8750 ? "{CLEAR}{CLR TAB}{TAB}{CLR TAB}{TAB}
        {CLR TAB}{TAB}{CLR TAB}{TAB}{CLR TAB}
        {TAB}{CLR TAB}{DOWN}  DATA DICTIONARY
        RECORD"
LP 8760 ? "{DOWN} FILE NAME - ";FILD$:? " APPL
        ICATION - ";APP$
MJ 8780 ? "{DOWN} EL{SET TAB}EM #  {SET TAB}HE
        ADING{6 SPACES}LE{SET TAB}NGTH
        {SET TAB} EDIT?":?
HI 8800 FOR I=1 TO NE:? "{TAB}";I;"{TAB}";:GOS
        UB 9820+I:? "{TAB}";DL(I);"{TAB}";
KD 8820 IF DE(I)=0 THEN ? " "
LI 8830 IF DE(I)=1 THEN ? "NUMERIC"
MF 8840 IF DE(I)=2 THEN ? "DATE"
GH 8850 IF DE(I)=3 THEN ? "DOLLAR"
EG 8870 NEXT I:? :? " RECORD LENGTH = ";RL:RET
        URN
MD 9000 FILD$=REC$(1,8):APP$=REC$(9,33)
MA 9010 RL=0:NE=0:FOR I=1 TO 6
```

```
KP 9020 DL(I)=VAL(REC$(32+I*2,33+I*2)):DE(I)=V
        AL(REC$(117+I,117+I))
OP 9030 IF DL(I)=0 THEN 9050
BC 9040 NE=NE+1:RL=RL+DL(I):IN$=REC$(34+I*12,4
        5+I*12):GOSUB 9800+I
PE 9050 NEXT I:GOSUB 6310:RETURN
KP 9200 TRAP 9500:OPEN #2,4,0,DM$:EOF=0
AP 9210 TRAP 9300:INPUT #2,REC$:TRAP 40000:IF
        T>8 THEN 9250
AB 9220 IF T=0 THEN ? " ";REC$(1,8);" ";REC$(9
        ,33):GOTO 9210
DE 9230 IF FILD$<>REC$(1,LEN(FILD$)) THEN 9210
KP 9250 EOF=EOF+1:GOSUB 9000:IF T>1 THEN GOSUB
         8750
MG 9280 IF T>1 THEN ? :? P$:IF T>8 THEN ? " TY
        PE E TO END"
CD 9290 IF T>1 THEN INPUT I$:IF T>8 AND I$<>"E
        " THEN 9210
JH 9300 CLOSE #2
JL 9310 IF T>0 AND EOF=0 THEN ? N$:GOSUB 6300
KO 9320 RETURN
PO 9500 STATUS #2,K:? "CHECK DISK DRIVE",X$;K:
        ? P$;:CLOSE #2:INPUT I$:POP :GOTO R
LF 9510 STATUS #7,K:? "CHECK PRINTER",X$;K:? P
        $;:INPUT I$:GOTO R
MF 9801 HD1$=IN$:RETURN
MH 9802 HD2$=IN$:RETURN
MJ 9803 HD3$=IN$:RETURN
ML 9804 HD4$=IN$:RETURN
MN 9805 HD5$=IN$:RETURN
MP 9806 HD6$=IN$:RETURN
ME 9811 FD1$=IN$:RETURN
MG 9812 FD2$=IN$:RETURN
MI 9813 FD3$=IN$:RETURN
MK 9814 FD4$=IN$:RETURN
MM 9815 FD5$=IN$:RETURN
MO 9816 FD6$=IN$:RETURN
EJ 9821 ? HD1$;:RETURN
EL 9822 ? HD2$;:RETURN
EN 9823 ? HD3$;:RETURN
EP 9824 ? HD4$;:RETURN
FB 9825 ? HD5$;:RETURN
FD 9826 ? HD6$;:RETURN
MH 9841 IN$=FD1$:RETURN
MJ 9842 IN$=FD2$:RETURN
ML 9843 IN$=FD3$:RETURN
MN 9844 IN$=FD4$:RETURN
MP 9845 IN$=FD5$:RETURN
NB 9846 IN$=FD6$:RETURN
```

Program 2. DMSSORT

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
BN 20 ? CHR$(253):GRAPHICS 0:POKE 82,0:POKE 16
   ,64:POKE 53774,64:GOTO 4000
ON 50 DIM X$(8500),DS$(124),APP$(25),FIL$(14),
   FILD$(8),HD1$(12),HD2$(12)
BP 60 DIM HD3$(12),HD4$(12),HD5$(12),HD6$(12),
   I$(1),IN$(12),DL(6)
HB 70 FOR I=1 TO 6:DL(I)=0:NEXT I
EI 80 RETURN
KP 4000 PRINT "{CLEAR}{DOWN}  DMS DATA BASE SO
   RT":GOSUB 50
NN 4010 ? :? " SELECT DATA SET;":? " (I - INDE
   X), D - DMS, E - END)"
HO 4030 INPUT FILD$:L=LEN(FILD$):IF L=1 AND FI
   LD$="D" THEN RUN "D:DMSDB"
AA 4040 T=1:IF L=0 THEN 4010
KP 4050 IF L=1 AND FILD$="E" THEN GRAPHICS 0:E
   ND
HP 4060 IF L=1 AND FILD$="I" THEN PRINT "
   {CLEAR}{DOWN} DATA SET INDEX":? :T=0
NM 4200 R=4000:TRAP 9500:OPEN #2,4,0,"D:DMS.DB
   ":EOF=0
DA 4210 TRAP 4300:INPUT #2,DS$:TRAP 40000
HB 4220 IF T=0 THEN ? " ";DS$(1,8);" ";DS$(9,3
   3):GOTO 4210
OH 4230 IF FILD$<>DS$(1,LEN(FILD$)) THEN 4210
DO 4240 FILD$=DS$(1,8):APP$=DS$(9,33)
EJ 4250 EOF=EOF+1:RL=1:NE=0:FOR I=1 TO 6
GI 4260 DL(I)=VAL(DS$(32+I*2,33+I*2)):IF DL(I)
   =0 THEN 4290
NA 4280 NE=NE+1:RL=RL+DL(I):IN$=DS$(34+I*12,45
   +I*12):GOSUB 9800+I
FH 4290 NEXT I
BA 4300 CLOSE #2:IF T=0 THEN 4010
JF 4310 IF EOF=0 THEN ? "NOT FOUND":GOTO 4010
BK 4330 FOR L=1 TO LEN(FILD$):IF FILD$(L,L)<>"
    " THEN NEXT L
IL 4340 L=L-1:FIL$(1,2)="D:":FIL$(3,2+L)=FILD$
   (1,L):FIL$(3+L)=".DB"
CA 5000 IF NE=0 THEN 490
KP 5010 ? "{CLEAR}{DOWN}  ";APP$:? "{DOWN}SELE
   CT SORT KEY:":? :FOR I=1 TO NE
PD 5020 ? I;" ";:GOSUB 9820+I:? :NEXT I:? "PRE
   SS RETURN TO CANCEL";
NJ 5030 TRAP 4000:INPUT SF:TRAP 40000:IF SF<1
   OR SF>NE THEN ? "INVALID":GOTO 5030
FO 5040 T=2:? "ASCENDING OR DESCENDING? (A OR
   D)";:INPUT I$:IF I$="D" THEN T=1
```

```
MI 5050 SL=DL(SF):? "LOADING ";FIL$:? "SORT ON
         ";SL;" CHAR OF ";:GOSUB 9820+SF:?
PC 5060 SS=1:IF SF>1 THEN FOR I=2 TO SF:SS=SS+
        DL(I-1):NEXT I
DD 5070 SE=SS+SL-1:R=5000:EOF=0:N=1:L=RL-1
HD 5100 TRAP 9500:OPEN #2,4,0,FIL$
DE 5110 TRAP 5150:INPUT #2,DS$:TRAP 40000
GJ 5120 EOF=EOF+1:X$(N,N+L)=DS$:N=N+RL:GOTO 51
        10
DG 5150 CLOSE #2:? "{DOWN}REC LOADED= ";EOF;",
         RAM (BYTES)= ";N-1:? "BEGIN SORT"
GB 5160 I=1:N=N-RL
IG 5200 K=0:DS$=X$(I,I+L):J=I+RL
LC 5210 ON T GOTO 5220,5240
BO 5220 IF X$(J+SS-1,J+SE-1)<DS$(SS,SE) THEN D
        S$=X$(J,J+L):K=J
MP 5230 GOTO 5250
CA 5240 IF X$(J+SS-1,J+SE-1)<DS$(SS,SE) THEN D
        S$=X$(J,J+L):K=J
DH 5250 J=J+RL:IF J<=N THEN 5210
PL 5280 IF K<>0 THEN X$(K,K+L)=X$(I,I+L):X$(I,
        I+L)=DS$
PK 5290 I=I+RL:IF I<N THEN 5200
CM 5300 ? "{▤} SORT COMPLETED"
FI 5320 TRAP 9500:XIO 36,#2,0,0,FIL$:OPEN #2,8
        ,0,FIL$:TRAP 40000:EOF=0
II 5330 FOR I=1 TO N STEP RL:? #2;X$(I,I+L):EO
        F=EOF+1:NEXT I
LM 5340 CLOSE #2:XIO 35,#2,0,0,FIL$:? "RECOUNT
        = ";EOF
JP 5350 ? "SORT THIS FILE AGAIN? (Y OR N)";:IN
        PUT I$:IF I$="Y" THEN 5010
ML 5360 GOTO 4000
HO 9500 STATUS #2,K:? "{▤}CHECK DISK DRIVE","▤
        RROR ";K:CLOSE #2:? "PRESS ENTER";:INP
        UT I$:GOTO R
MF 9801 HD1$=IN$:RETURN
MH 9802 HD2$=IN$:RETURN
MJ 9803 HD3$=IN$:RETURN
ML 9804 HD4$=IN$:RETURN
MN 9805 HD5$=IN$:RETURN
MP 9806 HD6$=IN$:RETURN
EJ 9821 ? HD1$;:RETURN
EL 9822 ? HD2$;:RETURN
EN 9823 ? HD3$;:RETURN
EP 9824 ? HD4$;:RETURN
FB 9825 ? HD5$;:RETURN
FD 9826 ? HD6$;:RETURN
```

Mailing Label Utility

Vernon M. Daly

The "Mailing Label Utility" is an excellent example of how the "Data Management System" can be used for other applications. Requires the use of the Data Management System (previous article) and a disk drive.

This mailing label utility requires the use Ron Marcuse's "Data Management System" (DMSDB) to create the database necessary for printing labels. I have used his program (unchanged since I first typed it in) to keep track of several things around the house. Even though it's of limited size, it fits lots of applications. It's easy to use and has good data format, both on the screen and the printer.

Doesn't Do It All

I soon realized that there was one thing I would like to do with DMSDB that it didn't do: print mailing labels. DMSDB's capabilities were perfect. I could enter new names, make deletions or corrections, and do searches and sorts. But the printer format, good for making a list, was wrong for labels. I decided to write a program that would use the data files created by Ron Marcuse's program, but output the data in the right format for the job I wanted to do.

A Few Extras

While I was at it, I decided to add a few extras that would make label printing easier. First, I wanted to be able to control the number of fields that would be printed. This would allow me to have other data in the file than what I might want to print on the label. It would also be nice to be able to restart the label printing process anywhere in the data file. Finally, since my printer has friction feed, I wanted to be able to do single forms (to print right on the envelopes). This means that the program would need to wait for me while I insert each envelope. All these features are in fact incorporated in this "Mailing Label Utility."

To make the program easy to use, I kept the format of operation very similar to the Data Management System program. Prompts and error messages are handled in much the same way as the original program, as are the format for getting into the proper database and the use of selective searches. This

makes it easy for someone familiar with the original program to use this utility. Let me note here that I don't claim originality for most of the file handling routines in this program. They're Ron Marcuse's from the original article.

Using the Program

All data entry and editing are done through the original DMSDB program. A typical example of a file setup would be to use field 1 for name, field 2 for address, field 3 for city and state, and field 4 for zip code.

With the data entered into the database, you can run the mail utility. Load the printer with the mailing labels. Any printer should work since no special control codes are used. Select the database the same way as in the DMSDB program. Next, select the printer option. The data-to-screen option is included as an operator convenience. It displays all of the data to the screen and has no data control options.

You may now select among the various printout options. You must tell the computer if you want all records or only selected ones. This is done in exactly the same fashion as the DMSDB program. If you choose selective printing, you must select the key field and give the selection criteria.

Next, input the number of fields to print, starting entry number, and form length (number of lines that the label is long). If you want to print single forms as single-feed labels or printing on the envelopes, enter 0 here. As soon as you enter the form length, printing will begin. Data will be shown on the screen as it is printed. On continuous forms, printing will continue until all data has been printed. On single forms, you will be prompted to press RETURN to allow printing to continue after each record is printed.

Mailing Label Utility

This program must be used with DMSDB program in the previous article, "Data Management System."

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
KB 40 POKE 82,0:POKE 83,39:GOTO 90
KG 50 DIM APP$(26),FIL$(14),FILD$(9),HD1$(12),
      HD2$(12),HD3$(12),HD4$(12),HD5$(12)
GL 60 DIM HD6$(12),I$(1),J$(1),REC$(132),IN$(3
      1),DM$(8),DL(6)
GN 65 DIM S$(15),X$(5),P$(24),N$(16):N$="RECOR
      D NOT FOUND":DM$="D:DMS.DB"
```

```
DE 70  P$="PRESS RETURN TO CONTINUE":S$="{DOWN}
       SELECT OPTION:":X$="ERROR"
HG 75  FOR I=1 TO 6:DL(I)=0:NEXT I
BJ 80  REC$(1)=" ":REC$(132)=" ":REC$(2)=REC$:R
       ETURN
DC 90  DIM I$(1)
GF 100  GRAPHICS 0:? "{DOWN}  DMS MAILING LABEL
        UTILITY"
GE 110  ? "{DOWN}{4 SPACES}PRIMARY OPTION MENU"
JJ 120  ? "{DOWN} D - SELECT DATA BASE"
LO 130  ? "{DOWN} S - PRINT DATA TO SCREEN"
CA 140  ? "{DOWN} P - PRINT DATA TO PRINTER"
CC 150  ? "{DOWN} E - END (EXIT PROGRAM)"
IB 160  ? :? "{DOWN} SELECT OPTION";:INPUT I$:I
       F I$="D" THEN 750
FJ 170  IF I$="S" THEN 2000
GB 180  IF I$="E" THEN END
FH 190  IF I$="P" THEN 1000
GC 200  GOTO 160
LA 490  PRINT "{DOWN}SELECT DATA SET 1ST":GOSUB
        6300:GOTO 100
DE 750  ? "{CLEAR}{DOWN}{3 SPACES}DATA BASE SEL
       ECTION":CLR :GOSUB 50:R=760
OI 760  ? "{DOWN}ENTER DATA SET NAME (I FOR IND
       EX)":? :? P$;" (END)"
NK 770  INPUT FILD$:IF LEN(FILD$)=0 THEN 100
HJ 780  IF LEN(FILD$)=1 AND FILD$="I" THEN GOSU
       B 6500:GOTO 760
KP 800  T=1:GOSUB 9200:IF EOF=0 THEN 100
GM 820  ? "{DOWN}DATA SET LOADED":GOSUB 6300
DC 850  DIM DS$(RL),SV$(30):FOR I=1 TO RL:DS$(I
       ,I)=" ":NEXT I:GOTO 100
BM 1000  IF NE=0 THEN 490
GO 1020  PRINT "{CLEAR}{DOWN}{3 SPACES}";APP$
JI 1030  PRINT "{DOWN}A - PRINT ALL RECORDS"
BL 1040  PRINT "{DOWN}S - PRINT SELECTED RECORD
        S"
AG 1050  PRINT "{DOWN}E - END (RETURN TO MENU)"
PL 1060  PRINT S$;:INPUT J$:IF J$="E" THEN 100
LE 1070  IF J$="A" THEN T=2:GOTO 1160
EI 1080  IF J$="S" THEN T=1:PRINT "{CLEAR}":GOT
       O 1100
AN 1090  PRINT X$:GOTO 1060
HP 1100  ? " SELECT KEY:":FOR I=1 TO NE:PRINT "
         ";I;" ";
EM 1110  GOSUB 9820+I:NEXT I
LO 1120  PRINT "ENTER KEY FIELD #";
DH 1130  INPUT SF:IF SF<1 OR SF>NE THEN PRINT X
       $:GOTO 1120
```

```
KB 1140 PRINT "ENTER VALUE OF ";:GOSUB 9820+SF
        :PRINT " (1-";DL(SF);")"
ND 1150 INPUT SV$:SL=LEN(SV$):IF SL<1 OR SL>DL
        (SF) THEN PRINT X$;" LEN":GOTO 1140
PD 1155 SS=1:IF SF>1 THEN FOR I=2 TO SF:SS=SS+
        DL(I-1):NEXT I
HA 1156 SE=SS+SL-1
CD 1160 PRINT :PRINT "ENTER NUMBER OF FIELDS T
        O PRINT"
DA 1170 INPUT NF:IF NF<1 OR NF>NE THEN PRINT X
        $:GOTO 1160
JP 1180 PRINT :PRINT "START PRINTOUT WITH ENTR
        Y NUMBER ";:INPUT START
FH 1190 PRINT :PRINT "WHAT IS THE FORM LENGTH
        IN LINES":PRINT "ENTER '0' FOR INDIVID
        UAL FORMS":INPUT FORMLEN
ND 1200 R=1000:TRAP 9600:OPEN #7,8,0,"P:"
MC 1220 GOTO 2200
BN 2000 IF NE=0 THEN 490
CB 2020 ? "{CLEAR}{DOWN}{3 SPACES}";APP$
CI 2050 T=2:NF=NE
GN 2200 I=0:EOF=0:R=2000:TRAP 9500:OPEN #2,4,0
        ,FIL$
ME 2210 TRAP 2420
JC 2220 INPUT #2,DS$
PL 2225 L=1:TRAP 40000:EOF=EOF+1:IF T=2 THEN 2
        328
AH 2230 IF SV$<>DS$(SS,SE) THEN 2210
KF 2328 FOR J=1 TO NF
NK 2340 PRINT DS$(L,L+DL(J)-1)
AN 2345 IF I$="P" THEN GOSUB 3000
PH 2352 L=L+DL(J)
FI 2355 NEXT J
BC 2356 IF I$="P" THEN GOSUB 3300
CG 2358 PRINT :PRINT
MJ 2360 GOTO 2210
MM 2420 IF EOF=0 THEN PRINT N$
EF 2430 IF EOF>0 THEN PRINT "REC COUNT- ";EOF
BF 2440 CLOSE #2:CLOSE #7:PRINT P$:INPUT I$:GO
        TO 100
EE 3000 REM PRINT SUBROUTINE
AG 3100 IF EOF<START THEN RETURN
GM 3120 PRINT #7;DS$(L,L+DL(J)-1)
KF 3200 RETURN
AI 3300 IF EOF<START THEN RETURN
FM 3305 IF FORMLEN=0 THEN PRINT P$:INPUT J$:RE
        TURN
EJ 3310 IF NF=FORMLEN THEN RETURN
NO 3320 FOR J=1 TO FORMLEN-NF
LA 3330 PRINT #7
```

```
FD 3340 NEXT J
KL 3350 RETURN
HM 6300 FOR I=1 TO 150:NEXT I:RETURN
BK 6310 FOR L=1 TO LEN(FILD$):IF FILD$(L,L)<>"
         " THEN NEXT L
KF 6320 L=L-1:FIL$(1,2)="D:":FIL$(3,2+L)=FILD$
        (1,L):FIL$(3+L)=".DB":RETURN
HJ 6500 ? "{CLEAR}{DOWN}  DATA SET INDEX":? :T
        =0:GOSUB 9200:RETURN
MD 9000 FILD$=REC$(1,8):APP$=REC$(9,33)
MA 9010 RL=0:NE=0:FOR I=1 TO 6
EM 9020 DL(I)=VAL(REC$(32+I*2,33+I*2))
OP 9030 IF DL(I)=0 THEN 9050
BC 9040 NE=NE+1:RL=RL+DL(I):IN$=REC$(34+I*12,4
        5+I*12):GOSUB 9800+I
PE 9050 NEXT I:GOSUB 6310:RETURN
KP 9200 TRAP 9500:OPEN #2,4,0,DM$:EOF=0
HN 9210 TRAP 9300:INPUT #2,REC$:TRAP 40000
AB 9220 IF T=0 THEN ? " ";REC$(1,8);" ";REC$(9
        ,33):GOTO 9210
DE 9230 IF FILD$<>REC$(1,LEN(FILD$)) THEN 9210
KP 9250 EOF=EOF+1:GOSUB 9000:IF T>1 THEN GOSUB
         8750
IA 9280 IF T>1 THEN ? :? P$
FC 9290 IF T>1 THEN INPUT I$
JH 9300 CLOSE #2
JL 9310 IF T>0 AND EOF=0 THEN ? N$:GOSUB 6300
KO 9320 RETURN
PH 9500 STATUS #2,K:? "{⊟}CHECK DISK DRIVE{⊟}"
        ,X$;K:? P$;:CLOSE #2:INPUT J$:POP :GOT
        O R
OB 9600 STATUS #7,K:? "{⊟}CHECK PRINTER{⊟}",X$
        ;K:? P$;:CLOSE #7:INPUT J$:POP :GOTO R
MF 9801 HD1$=IN$:RETURN
MH 9802 HD2$=IN$:RETURN
MJ 9803 HD3$=IN$:RETURN
ML 9804 HD4$=IN$:RETURN
MN 9805 HD5$=IN$:RETURN
MP 9806 HD6$=IN$:RETURN
FM 9821 PRINT HD1$:RETURN
FO 9822 PRINT HD2$:RETURN
GA 9823 PRINT HD3$:RETURN
GC 9824 PRINT HD4$:RETURN
GE 9825 PRINT HD5$:RETURN
GG 9826 PRINT HD6$:RETURN
```

Chapter 2

Programming Aids

User-Defined Function Keys

John Hebrink

"Userkeys" allows you to define any key on the Atari keyboard to represent any character string or BASIC function of up to 255 characters, and have it displayed or executed with the push of a button. It takes up 1K of memory and lets you define up to 64 keys for a total of 512 characters. For Atari 400 and 800 only (does not work on XL or XE models).

Although the keyboard doesn't have 64 individual keys, some can represent multiple character strings. For example *A*, *a*, inverse-video *A*, inverse-video *a*, CTRL-*A*, and inverse CTRL-*A* are all legitimate, definable characters. Uppercase letters are naturally the easiest to use since they involve only a single keystroke.

Defining a Key

To define a key, run the program and press the SELECT button (before you run this program, be sure to save it). A question mark will appear, prompting you to enter the character to be defined. Enter the string and RETURN. To make a BASIC functional string, press ESC at the end of the character string before hitting RETURN. To exit the program, hit RETURN when the question mark appears. From that point on, anytime you want to define a key, just press the SELECT button. To have a string displayed or executed, press the definition character and then press the START button.

For example, if you want to have a key reset the left and right margins, clear the screen, and list your BASIC program, try this:

?L=POKE 82,3:POKE 83,37:PRINT CHR$(125):LIST{ESC}

To execute this function, press L and then START.

One of the most useful functions I have run across is autonumbered DATA statements:

?D=PRINT A;" DATA ":A=A+10{ESC}
?F={UP}{SHIFT-DELETE}{3 UP}{SHIFT-DELETE}{10 RIGHT}

This is a two-stage function, however, and is executed by pressing D, START, F, and START. Variable A would have to

be defined in immediate mode to equal the starting line number before using this function. The second string, F, is used to erase the first one, D, and the unwanted READY, and to reposition the cursor behind DATA. When entering this string and strings like it, don't press ESC before entering the characters. The computer will think you want to RETURN. Just enter the functions as is. This may create a somewhat confusing display, but will be executed properly when printed by the program. Anything that will work in immediate mode will also work this way.

Line 60 uses the RETURN key mode to erase the BASIC program. This means that you must be sure to save the program before running it, or you will lose it. The program then resets the BASIC program pointers (128–145) up 1K by passing MEMLO down to LOMEM.

If you want to use the console buttons in a program, POKE 548,35. (Why not define a key to do it?) SYSTEM RESET will resume program execution.

Userkeys

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program. The program will not work on XL and XE models.

```
IO 10 FOR I=1536 TO 1777:READ A:POKE I,A:NEXT
      I
CB 20 FOR I=256 TO 325:READ A:POKE I,A:NEXT I
MB 30 A=PEEK(129):IF PEEK(128)<>0 THEN A=A+1
MN 40 POKE 743,0:POKE 744,A+4:POKE 204,A
CL 50 POKE 2,0:POKE 3,A:POKE 9,2
LL 60 GRAPHICS 0:POSITION 2,4:? "NEW":? :? :?
      "D=USR(256)":POSITION 2,0:POKE 842,13
HJ 70 DATA 173,232,2,24,105,4,141,232,2,162,0,
      160,21,169,7,32,92,228,76,77,160,173,31,
      208
ED 80 DATA 201,5,240,101,173,31,208,201,6,240,
      3,76,98,228,169,43,141,36,2,173,31,208,2
      01
NN 90 DATA 6,208,3,76,98,228,166,205,202,202,2
      02,202,189,0,0,205,251,2,240,6,224,0,208
CO 100 DATA 240,240,36,189,1,0,141,97,0,189,2,
       0,141,98,0,189,3,0,141,124,0,120,162,0,
       138
OG 110 DATA 72,189,0,0,201,27,208,14,169,12,14
       1,252,2,104,169,21,141,36,2,76,98,228,3
       2
```

```
ID 120 DATA 164,246,104,170,232,224,0,208,223,
       240,236,169,28,141,36,2,166,205,165,207
       ,157
ED 130 DATA 1,0,165,204,157,2,0,169,1,133,206,
       169,63,32,164,246,32,226,246,201,155,20
       8,3
CN 140 DATA 76,9,0,166,205,157,0,0,32,164,246,
       169,61,32,164,246,164,207,169,126,145,2
       03
LM 150 DATA 230,207,165,207,208,2,230,204,165,
       204,197,129,208,8,169,0,141,27,0,76,9,0
OC 160 DATA 32,226,246,201,155,240,11,164,207,
       145,203,32,164,246,230,206,16,216,32,16
       4
LH 170 DATA 246,166,205,165,206,157,3,0,232,23
       2,232,232,134,205,16,148,104,169,12,141
OF 180 DATA 74,3,165,204,160,6,190,55,1,157,0,
       6,136,16,247,24,105,1,160,7,190,62,1,15
       7
MO 190 DATA 0,6,136,16,247,160,0,185,0,6,145,2
       03,169,0,153,0,6,200,208,243,230,204,23
       0
EE 200 DATA 204,76,27,241,10,78,84,90,164,202,
       205,61,75,81,87,140,145,169,233
```

Variable Changer and Block Deleter

Robert Dolan

With these two short routines, instantly change the name of a variable each time it appears in your program or delete a series of lines that are no longer needed.

Here are two aids for the BASIC programmer. The first is a line block deleter that allows you to delete a series of lines quickly and easily. The second is a variable name change utility.

Block Line Deleter

This simple utility will delete a series of lines. Use the routine with caution, especially if you're using Revision B of Atari BASIC. If you're not sure which version of BASIC you have, enter this line in immediate mode:

PRINT PEEK (43234)

If your Atari responds with 234, then you have Revision C and this routine should work just fine. If it responds with 96, watch out. You have Revision B and it is full of bugs—especially the famous lockup bug. Be careful in deleting lines. To be safe, always save a copy of your BASIC program on disk or tape before attempting to use Program 1, especially if you don't have Revision C of BASIC.

To use the program, simply type it in using "The Automatic Proofreader" and LIST it to tape (LIST "C:") or disk (LIST "D:DELETER.LST"). Now, whenever you're working on a BASIC program and want to delete lots of lines, just ENTER the program from disk (ENTER "D:DELETER") or tape (ENTER "C:", being sure to position the tape correctly). The program will append itself to the BASIC program in memory. (Note that since the "Deleter" starts at line 32000, it will work only with programs that do not use these same line numbers.) To delete lines from your program, type GOTO 32000 and answer the prompts.

When all the deleting is done, you'll want to save your program without lines 32000 and above. Use the LIST command:

LIST "D:*filename*",1,31999 (Disk)

or

LIST "C:",1,31999 (Tape)

Variable Changer

This utility will peruse your variable name table and list all variables it finds in the order in which they were entered into your program. It also assigns a number to each variable, which is used to identify the variable when you want to change it. This makes the program smaller and less complicated.

To use the utility, simply follow the same procedure described above. First, type in and LIST the program to disk or tape. Load in your BASIC program and append this utility to it. To use the "Changer" type GOTO 32000, press RETURN, and follow the prompts. Again, once the changes have been made, LIST the program to disk or tape, being sure to indicate the line number range so that the Changer is not saved with your BASIC program.

When the utility is executed, you'll be asked if you want to send the list to a printer. If so, enter Y and RETURN. This is handy since you don't have to remember or write down the number of any variable names you wish to change. If you don't have a printer or if your paper supply is low, just hit RETURN, and the list will be displayed on the screen in blocks of 20 names.

After the listing is completed, you'll be prompted for the number of the variable name to change. Be sure to enter a number here or the program will stop. You're limited in your replacement name to the same number of characters as the original name. Also be careful of string and array names. You must include the dollar sign for strings and the parentheses for arrays.

Once you enter the new name, all references to it throughout your program will be changed instantly. At this point the progam will reprompt you for another name change. If you are finished, hit RETURN and the program will end.

Program 1. Block Line Deleter

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
KL 32000 REM ** LINE BLOCK DELETE PROG
NL 32002 DIM LINE(200):L=0:PR=31999:SETCOLOR 2
         ,10,2:SETCOLOR 4,10,2:SETCOLOR 1,10,1
         4:TRAP 32004
```

```
GL 32003 LADR=PEEK(136)+256*PEEK(137)
NE 32004 ? "{CLEAR}":? "***** LINE BLOCK DELET
         E UTILITY *****"
JL 32005 ? :? :? "ENTER START LINE ";:INPUT LS
         T
GE 32006 ? :? "ENTER END LINE{3 SPACES}";:INPU
         T LNEND:TRAP 40000:IF LST>PR OR LNEND
         >PR THEN ? :? "INVALID LINE NUMBERS":
         END
PL 32007 IF LST>LNEND THEN ? :? "END LINE MUST
          BE GREATER THAN START":END
CO 32008 ? :? "WORKING......"
KK 32009 THS=PEEK(LADR)+256*PEEK(LADR+1)
AJ 32010 IF THS>=LST THEN 32012
AM 32011 LADR=LADR+PEEK(LADR+2):GOTO 32009
EO 32012 IF THS>LNEND THEN 32014
LP 32013 L=L+1:LINE(L)=THS:GOTO 32011
OI 32014 ? "LINES FOUND IN THIS BLOCK:"
PK 32015 FOR I=1 TO L:? I,LINE(I):NEXT I
CC 32016 ? :? CHR$(253):? "WARNING IF YOUR PRO
         GRAM IS LONG,":? "THEN CARE SHOULD BE
          USED IN THIS PROCEDURE";
EG 32017 ? ".WAIT AWHILE AFTER EACH ":? "BLOCK
          TO AVOID FATAL LOCKUPS (#%$@?#)"
LH 32018 ? :? :? "HIT START TO DELETE LINES"
OM 32019 IF PEEK(53279)<>6 THEN 32019
DI 32020 ? "{CLEAR}":POSITION 2,4
AL 32021 FOR D=1 TO 20:Q=Q+1
DB 32022 ? LINE(Q):IF Q=L THEN POP :GOTO 32024
HN 32023 NEXT D
OP 32024 ? "CONT":POSITION 2,0
NJ 32025 POKE 842,13:STOP
FJ 32026 POKE 842,12
JD 32027 IF Q<L THEN 32018
JC 32028 ? "ALL DONE":END
```

Program 2. Variable Changer

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
NJ 32000 REM
EO 32001 REM * VARIABLE CHANGE UTILITY *
HE 32003 XYZ=PEEK(130)+256*PEEK(131):QQQ=0:DIM
          QQQQ$(10),VPTR(129)
PD 32004 ? "{CLEAR}":POKE 710,146:POKE 712,146
         :? "SEND VARIABLES TO PRINTER ";:INPU
         T QQQQ$:GOSUB 32023
LF 32005 ? #2;"VAR NO.=";QQQ;"{3 SPACES}";:? #
         2;"VAR NAME=";
```

```
ND 32006 VPTR(QQQ)=XYZ
OO 32007 IF PEEK(XYZ)<127 THEN ? #2;CHR$(PEEK(
         XYZ));
LJ 32008 IF PEEK(XYZ)>127 THEN ? #2;CHR$(PEEK(
         XYZ)-128);
DC 32009 IF PEEK(XYZ)<128 THEN XYZ=XYZ+1:GOTO
         32007
DC 32010 XYZ=XYZ+1:QQQ=QQQ+1:Q=Q+1:IF Q=20 AND
          QPQ=0 THEN ? #2:GOSUB 32020
CA 32011 ? #2:IF XYZ<(PEEK(132)+256*PEEK(133))
          THEN 32005
BC 32012 ? #2:? #2:? #2;"END OF VARIABLE NAME
         TABLE":? #2;"NUMBER OF VARIABLES FOUN
         D=";QQQ:CLOSE #2
EI 32013 TRAP 32025:? :? "ENTER VARIABLE NO. T
         O CHANGE ";:INPUT VN:TRAP 40000
JD 32014 Q=VPTR(VN+1)-VPTR(VN)
HG 32015 ? "VAR. NO.";VN;" IS ONLY ";Q;" CHARS
         . LONG"
DE 32016 ? :? "ENTER NEW VARIABLE NAME ";:INPU
         T QQQQ$
LH 32017 IF Q=1 THEN POKE VPTR(VN),ASC(QQQQ$(1
         ,1))+128:GOTO 32013
CI 32018 FOR QQQ=0 TO Q-2:POKE VPTR(VN)+QQQ,AS
         C(QQQQ$(QQQ+1)):NEXT QQQ
BL 32019 POKE VPTR(VN)+QQQ,ASC(QQQQ$(QQQ+1))+1
         28:GOTO 32013
HL 32020 Q=0:POKE 764,255:? :? "HIT ANY KEY TO
          CONTINUE"
JO 32021 IF PEEK(764)=255 THEN 32021
FC 32022 ? :RETURN
PE 32023 CLOSE #2:IF QQQQ$="Y" THEN OPEN #2,8,
         0,"P:":QPQ=1:RETURN
OE 32024 OPEN #2,8,0,"E:":QPQ=0:RETURN
JN 32025 ? :? "Okay":END
```

Varlist

Brian Totty

A bug in a program got you completely baffled? Perhaps you're unknowingly using the same variable name twice. "Varlist" will help you keep track of your variables and also introduce you to how the Atari stores variables.

Atari BASIC is a useful and flexible language, especially in its usage of numeric variables. Atari BASIC allows for 128 different variable names. This BASIC does not, however, limit you to names of only a few characters. Where some machines may allow only N, G2, or SP, Atari's BASIC allows NUMBEROFBOMBS, GRADES2, or SHIPS. It's easy to get carried away with this convenience and start tossing about variables haphazardly. This makes the program flow harder to follow and the program harder to revise as the functions of some variables are forgotten.

There are times when you might want to know if you've used a certain variable and have to search the entire program line by line to try to find it.

Variable Storage

Atari BASIC holds all variable names that have been used during your programming in the Variable Name Table (VNT). The location of the VNT will depend on the configuration.

The Atari uses two memory locations (called pointers) to point to the VNT. It uses two locations because each location (called a byte) can hold a number from 0 to 255. The VNT is going to start at a location much greater than 255, so the second byte holds the number of multiples of 256. For example, if there is a 3 in the first byte (called the low byte) and a 0 in the next byte (called the high byte), you would have 3 plus 0 multiples of 256, or 3+(0*256), or 3. If there is a 67 in the low byte and a 6 in the high byte, you would have 67 plus 6 multiples of 256, or 67+(6*256), or 1603.

The two-byte pointer to the VNT starts at location 130 decimal (82 hexadecimal). In other words, 130 is the low byte and 131 is the high byte. Knowing this, you can find where the Variable Name Table starts with this BASIC line:

```
START=PEEK(130)+256*PEEK(131):PRINT "TABLE STARTS
  AT";START
```

Each variable is stored character by character in the bytes after the start of the table. Atari BASIC does this by placing the ASCII value of each character in subsequent bytes. At the end of the variable name, the last character is stored in inverse video (with 128 added to the ASCII value). For instance, ABD would be stored as 65, 66, 196 (the last number being the ASCII value of D with 128 added); VG$ would be stored as 86, 71, 164. And a G array would be the ASCII values of G(with the parenthesis in inverse video. Such variables as arrays and matrices are stored only up to the first parenthesis. The end of the table is signified by a zero.

Variable Listers

Program 1 is a short BASIC program that lists variables.

Line 30 creates some variables, so they can be found inside the VNT.

Line 40 finds the start of the VNT by the two-byte pointer at addresses 130 and 131 decimal.

Line 50 PEEKs the character in the table.

Line 60 checks if that byte is a zero (the end of the table).

Line 70 PRINTs the character value of that byte.

Line 80 checks for the end of the variable; if so, then jump to the start of next PRINT line.

Line 90 adds 1 to the place to get next byte and goes back to line 50.

There are two inherent problems with this BASIC lister, however. First, to use this program with another program, you will have to merge the two with the ENTER command, being careful of overlapping lines—an inconvenience at the least. Second, even without line 30, the variable table will get cluttered with the variables that the lister uses.

Program 2 is a short BASIC loader that creates a machine language program and stores it in page 6, starting at address 1536. Type in Program 2 and save it before you run it.

When Program 2 is executed, there will be a short pause and then the READY prompt will reappear. Now type PRINT USR(1536) and press RETURN. Voilà. There's your VNT, but what is that number at the end? That is the number of variables currently in your table.

Now erase the BASIC loader program with the NEW command. Again, type PRINT USR(1536) and watch what

happens. NEW erases your BASIC program, clears the VNT, but doesn't harm the machine language subroutine.

Program 1. Variable Lister, BASIC

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
PI 30 DIM ARR(1),F(1),B$(1):C1=10
KA 40 VNT=PEEK(130)+256*PEEK(131)
EE 50 BYTE=PEEK(VNT)
FG 60 IF BYTE=0 THEN 100
LF 70 PRINT CHR$(BYTE);
LP 80 IF BYTE>127 THEN PRINT
MK 90 VNT=VNT+1:GOTO 50
GI 100 END
```

Program 2. Variable Lister, Machine Language

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
IH 10 FOR I=1536 TO 1608:READ A:POKE I,A:NEXT
      I
OA 20 DATA 104,169,000,133,212,133
OI 30 DATA 213,165,130,133,204,165
OD 40 DATA 131,133,205,160,000,177
OC 50 DATA 204,240,034,133,206,032
OP 60 DATA 056,006,165,206,016,007
PD 70 DATA 169,155,032,056,006,230
OG 80 DATA 212,230,204,165,204,240
OH 90 DATA 003,076,015,006,230,205
BG 100 DATA 165,205,240,003,076,015
BG 110 DATA 006,096,162,000,142,072
AJ 120 DATA 003,142,073,003,160,011
BN 130 DATA 140,066,003,032,086,228
EO 140 DATA 096
GN 150 END
```

Scroll Protect

Jim Bender

Sometimes it would be desirable to print text to the top of a GRAPHICS 0 screen and allow the rest of the screen to scroll. All it takes is a short machine language routine.

At first I thought the only way to protect part of the GRAPHICS 0 screen from scrolling off the top was to open a text window at the bottom of the screen.

Since I really wanted to protect four or five lines on the top of the screen from scrolling, I decided to try to write a machine language routine that would protect the top four lines of the screen and that would interface with BASIC. Since the routine would have to be running all the time to catch any text that might stray into the protected area, the logical solution turned out to be the *Vertical Blank Interrupt* (VBI).

The routine would run during the time that the electron beam in the TV is returning from the bottom to the top of the screen, during the vertical blank period. BASIC is normally interrupted during this time as the operating system goes off to update certain registers. A disassembly of the operating system VBI routine revealed that it does little more than increment the realtime clock (decimal locations 18, 19, and 20). Given this and the fact that the Atari can execute upward of 30,000 cycles during the VBI, my short routine had found its interface.

The result appears in this short demonstration program. The DATA lines contain the actual routine. You set the routine in motion by first POKEing location 203 (decimal) with the number of lines you want to protect. You can call the routine with X=USR(1536). To turn off the scroll protect routine, use X=USR(1653) to call the GETOUT routine.

Scroll Protect

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
OJ 10 RESTORE :FOR I=0 TO 127:READ A:POKE 1536
     +I,A:NEXT I:REM LOAD THE ML ROUTINE
PP 20 PRINT CHR$(125):REM CLEAR THE SCREEN
MK 30 POSITION 2,0:REM START ON TOP LINE
II 40 PRINT CHR$(127);"THESE LINES WILL"
CN 50 PRINT CHR$(127);"NOT BE SCROLLED"
KC 60 PRINT CHR$(127);"OFF THE TOP OF"
```

```
CO 70 PRINT CHR$(127);"THE SCREEN."
HK 80 POKE 203,4:REM PROTECT 4 LINES
LE 90 A=USR(1536):REM START VBI ROUTINE
IM 95 REM OTHER LINES SCROLL NORMALLY
DK 100 FOR I=1 TO 80
ED 102 POKE 85,INT(I/4)
HN 104 PRINT "BUT THESE LINES WILL"
BP 106 NEXT I
KM 108 REM ML PROGRAM FOLLOWS-TYPE CAREFULLY!!
IE 110 DATA 104,173,48,2,133,205
JE 120 DATA 173,49,2,133,206,198
LF 130 DATA 206,169,0,133,204,105
FC 140 DATA 1,133,207,162,6,160
DD 150 DATA 31,169,7,32,92,228
FN 160 DATA 96,24,165,203,201,7
DA 170 DATA 48,4,169,6,133,203
LO 180 DATA 169,0,133,204,166,203
OK 190 DATA 165,204,105,40,202,208
BI 200 DATA 251,133,204,165,203,208
FG 210 DATA 4,169,0,133,204,165
IE 220 DATA 207,240,22,160,0,169
LE 230 DATA 0,145,205,200,196,204
JL 240 DATA 208,249,160,0,177,88
CA 250 DATA 145,205,200,196,204,208
LM 260 DATA 247,162,0,134,207,160
JB 270 DATA 0,177,205,145,88,200
CP 280 DATA 196,204,208,247,234,234
CH 290 DATA 234,234,234,234,234,234
KA 300 DATA 76,98,228,104,160,98
GG 310 DATA 162,228,169,7,32,92
OG 320 DATA 228,96
```

Data Dumper

Clifford Engels

Turn machine language routines in BASIC DATA statements into an object file that can be loaded from the Assembler/Editor for disassembly and modification.

Ever wonder how those great little subroutines in BASIC programs work? Have you ever wanted to easily load those ML routines with your Assembler/Editor?

Here is a small utility that will convert the DATA statements in a BASIC program to machine language and store the data on tape or disk. Later, this data can be loaded by the Assembler/Editor cartridge using the LOAD#C: or LOAD#D: *filename* command. The code can then be disassembled.

First, type and LIST the program to tape or disk. Then load in the program with the DATA statements. Since the utility program uses lines 0–11, be sure that no DATA statements are at those line numbers. If there are, temporarily adjust your BASIC program (remember, for the moment you're only interested in the DATA statements). Next, enter the utility. It should be the first 12 lines of the program in memory.

When you run the program, the first prompt will ask for the starting DATA line number. Enter the line number of the DATA statement at which you wish to start disassembling. The utility will stop automatically after the last data element has been read. The next prompt asks for the starting address. This address is where the machine language data will load using the Assembler/Editor cartridge. This address does not have to be the same as the load address in the BASIC program. You are going to disassemble the program, not run it. In fact, I recommend using 4096 as the load address so that you can easily remember the address when you disassemble the data. Decimal 4096 is hexadecimal 1000. You must use hexadecimal address when using the Assembler cartridge. Finally, you'll be asked for a filename. Enter C: to save to tape or D:*filename* (substitute a legal filename for *filename*) to save to disk. No quote marks are necessary.

DATA Dumper

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
PC Ø  DIM FI$(15):? "{CLEAR}ENTER LINE # OF FIR
      ST DATA LINE TO BE DISASSEMBLED ";:INPUT
      R:RESTORE R:B=Ø:C=Ø:TRAP 7
IJ 1  READ A:B=B+1:GOTO 1
EL 2  ? :? "ENTER STARTING LOAD ADDRESS ";:INPU
      T ADDR:ADRH=INT(ADDR/256):ADRL=ADDR-(256*
      ADRH):GOSUB 1Ø
PA 3  POKE 53775,35:POKE 53768,4Ø:POKE 53764,Ø:
      POKE 53766,Ø:POKE 53773,255:OPEN #3,8,Ø,F
      I$:RESTORE R
EI 4  PUT #3,255:PUT #3,255:PUT #3,ADRL:PUT #3,
      ADRH
NM 5  ADDR=ADDR+B-1:ADRH=INT(ADDR/256):ADRL=ADD
      R-(256*ADRH):PUT #3,ADRL:PUT #3,ADRH:C=1
DC 6  READ A:PUT #3,A:GOTO 6
CN 7  TRAP 7:IF PEEK(195)<>6 THEN ? :? "*** ERR
      OR# =";PEEK(195):CLOSE #3:END
NI 8  IF C=Ø THEN 2
BG 9  CLOSE #3:END
JA 1Ø ? "{DOWN}ENTER FILENAME.":? "{DOWN}BE SU
      RE TO INCLUDE [D:] PLUS THE{9 SPACES}FILEN
      AME FOR DISK.":? "{DOWN}OR [C:] FOR TAPE."
LP 11 INPUT FI$:RETURN
```

Chapter 3

Graphics

Atagraph

Edward Chaney

Create your own animated displays and save them for future display. Requires one disk drive and a joystick.

"Atagraph" is designed to help you create animation routines on your Atari. It allows you to create animation routines by drawing a series of pictures using a joystick. You can then select the sequence in which these drawings are to be displayed. If you like, you can save the routine for use in your own programs.

When Atagraph is run, a drawing frame will be displayed. Above this frame are the COLOR and SETCOLOR commands which are in effect. Below the frame is a brief menu of options:

View	Displays completed drawings below the frame
Clear	Clears the drawing area
Scan	Allows the user to scan completed drawings
End	Ends the current drawing
Exchange	Changes the drawing in the frame area
Directory	Lists the disk files

Options available but not listed are the OPTION key, which changes the plotting speed, and the down, right, and left arrow keys. These keys allow drawings displayed in the frame to be positioned onscreen for reference.

Changing Colors

To use Atagraph, follow the directions below to enter and run the program. When the frame is displayed, press the joystick trigger button. The color will change. Press the OPTION key and again press the trigger. Now, the color changes at a slower speed. Press and hold the OPTION key and the trigger until the desired speed is reached. Then, push the joystick in any direction. The cursor will draw or erase, depending on the color chosen. Color 0 erases. The colors are the same as with BASIC color statements.

The setcolor is also the same as BASIC's setcolor commands. SETCOLOR commands are issued by pressing R until the desired color register is displayed. Pressing H will change the hue, and pressing L will change the luminance.

When you have selected the colors and setcolor, you are ready to do your drawings. Use the joystick to draw a picture.

When your picture is completed, press the E key. The computer will scan the drawing frame and encode your drawing (you may have to press the E key more than once). You'll be asked if you are ready to save this series. If you plan more than one drawing, answer by pressing N and you'll be returned to the drawing board. Press the V key and the drawing will be displayed below the frame. Press X and the drawing will be displayed inside the frame where it can be moved or modified by using the arrow keys.

Putting It in Order

When you've completed several drawings, press S. All drawings will be displayed, and you will be asked to enter the sequence. The drawings are designated A to Z. Enter the appropriate letters for the order in which you wish your drawings to be displayed. Separate each letter with a comma and press RETURN after the whole sequence has been entered. You'll be asked to enter the delay. This is the length of time each drawing will remain on the screen. A high number gives a longer delay. When you've entered the delay, you'll be asked to enter the graphics mode. Your drawings may be displayed in either graphics mode 5 or 7. Enter your choice. The drawings will be displayed in the order you selected. This scan will continue, repeating itself, until you press a key.

When you press a key, you'll be asked if you're ready to save the series. If your animation routine is completed, press Y and answer the filename question by entering the correct filename. Enter a standard DOS filename beginning with D:. No quotes are necessary.

When the save is completed, you may do another series by pressing the Y key when asked if you wish to do another series.

Loading Pictures

To load a series of pictures you previously saved with Atagraph, select D for directory. You'll presently see a directory of the disk in drive 1. Then you'll be asked which file you wish to load. Be sure to use D: in front of the filename; quotes are not needed.

Typing In the Program

When typing in Atagraph, enter and save Program 1; then run it. After a few seconds the screen will clear, and new lines 50, 60, and 70 will appear on the screen. The program will have been erased from memory. Hold down the CONTROL key and press the up arrow key (the key immediately to the right of the P key) as many times as necessary so that the cursor is on top of the 5 of the number 50 (this will be at the very top left of the screen). Once the cursor is positioned correctly, press the RETURN key three times. You have now entered the first three lines of the Atagraph program. Now type in Program 2 so that lines 50, 60, and 70 become part of the program.

The string variables LINE$, SIDE$, and MINE$ (lines 50, 60, and 70) contain characters which are difficult to enter correctly. The method used here makes them much easier to enter.

When all of Program 2 and the addition of lines 50, 60, and 70 have been entered correctly, save the entire program (you might want to save it a few times along the way just in case). Now, each time you want to use Atagraph, just load and run this final version.

Program 1. Atagraph String Maker

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
KJ 10 DIM MINE$(96),LINE$(24),SIDE$(39)
PN 20 FOR X=0 TO 23:READ A:LINE$(LEN(LINE$)+1)
      =CHR$(A):NEXT X
PO 30 FOR X=0 TO 38:READ A:SIDE$(LEN(SIDE$)+1)
      =CHR$(A):NEXT X
AK 40 FOR X=0 TO 95:READ A:MINE$(LEN(MINE$)+1)
      =CHR$(A):NEXT X
AM 50 GRAPHICS 0:? "50 LINE$=";CHR$(34);LINE$
EN 60 ? "60 SIDE$=";CHR$(34);SIDE$
FH 70 ? "70 MINE$=";CHR$(34);MINE$
AD 80 ? :? :? "ENTER THESE LINES THEN ":? "ENT
      ER PROGRAM 2"
FD 90 NEW
HF 100 DATA 104,104,133,204,104,133,203,160,0,
       132,205,177,203,240,3,230,205,96,200,19
       2,10,208,244,96
FN 110 DATA 104,104,133,204,104,133,203,160,0,
       132,205,162,0,177,203,240,3,230,205,96
NA 120 DATA 232,224,26,240,250,165,203,24,105,
       40,133,203,144,2,230,204,24,144,230
```

```
FC 130  DATA 104,104,162,4,104,149,203,202,16,2
        50,165,203,56,233,7,133,203,176,2,198,2
        04,160,0,162,4
OH 140  DATA 177,205,153,196,2,200,202,16,247,1
        77,205,133,212,200,177,205,133,213,165,
        208,24,101,213,133,209
HD 150  DATA 200,166,213,165,207,240,2,177,205,
        145,203,202,240,4,200,24,144,241,198,21
        2,165,212,240,21,230
PA 160  DATA 205,165,205,208,2,230,206,165,203,
        56,229,209,133,203,176,216,198,204,144,
        212,96
```

Program 2. Atagraph

Be sure to read the directions for entering this program before beginning.

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
DB 30  GOSUB 1380:DIM MINE$(96),LINE$(24),SIDE$
       (39)
IK 40  DIM SEQ$(26),FILE$(20),A$(1),VIEWS(26):V
       IEWCNT=0:VCNT=1:XVIEW=1:GR=7:MEM=15000:V
       IEWS(0)=MEM
AK 80  POKE 82,1:GRAPHICS 7:POKE 764,255:POKE 7
       52,1:RESTORE 90:F=0
CN 90  DATA 3,70,6,6,85,70,88,6,89,6,90,6
JE 100  DLIST=PEEK(560)+256*PEEK(561):HOME=PEEK
       (88)+256*PEEK(89):FRAME=HOME+255
NJ 110  FOR X=0 TO 5:READ A:READ B:POKE DLIST+A
       ,B:NEXT X
NN 120  POSITION 3,0:POKE 87,0:? #6;"SETCOLOR R
       ,H,L{6 SPACES}COLOR{4 SPACES}";REG
KN 130  ? "VIEW CLEAR SCAN END  EXCHANGE DIRECT
       ORY"
FC 140  POKE 87,7:FOR Y=0 TO 2
KJ 150  COLOR 3:PLOT 59-Y,8-Y:DRAWTO 100+Y,8-Y
KO 160  DRAWTO 100+Y,36+Y:DRAWTO 59-Y,36+Y
BJ 170  DRAWTO 59-Y,8-Y:NEXT Y
JN 180  POKE 87,7:POKE 208,39:X=70:Y=20
DE 190  IN=STICK(0):IF IN=15 THEN 320
NG 200  IF IN=14 THEN Y=Y-1
NE 210  IF IN=13 THEN Y=Y+1
ND 220  IF IN=11 THEN X=X-1
FK 230  IF IN=10 THEN Y=Y-1:X=X-1
DB 240  IF IN=9 THEN X=X-1:Y=Y+1
KJ 250  IF IN=7 THEN X=X+1
DA 260  IF IN=6 THEN X=X+1:Y=Y-1
CO 270  IF IN=5 THEN X=X+1:Y=Y+1
FJ 280  IF X<60 THEN X=99
```

```
FM 290  IF X>99 THEN X=60
BN 300  IF Y<9 THEN Y=35
CA 310  IF Y>35 THEN Y=9
PP 320  FOR T=0 TO DEL:IF COL=0 THEN COLOR 2:PL
        OT X,Y
IL 330  COLOR 0:PLOT X,Y:COLOR COL:PLOT X,Y:NEX
        T T:IF PEEK(53279)=3 THEN DEL=DEL+1:IF
        DEL+1>20 THEN DEL=0
LD 340  IF STRIG(0)=0 THEN COL=COL+1:IF COL=4 T
        HEN COL=0
CI 350  POKE 87,0:POSITION 32,0:? #6;COL:POKE 8
        7,7
OB 360  IF PEEK(764)=255 THEN 190
IF 370  REM ■CLEAR{3 SPACES}
LH 380  IF PEEK(764)=18 THEN GOTO 80
OH 390  IF PEEK(764)<>40 THEN 440
BN 400  REM ■REGISTER
HN 410  REG=REG+1:IF REG>4 THEN REG=0
PH 420  A=PEEK(708+REG):B=INT(A/16):A=A-B*16
NC 430  POKE 87,0:POSITION 12,0:? #6;REG;",";B;
        ",";A;"  ":POKE 87,7:POKE 764,255:GOTO
        190
OP 440  IF PEEK(764)<>57 THEN 480
JP 450  REM ■HUE{4 SPACES}
AB 460  A=PEEK(708+REG)+16:IF A>240 THEN A=A=0
OI 470  POKE 708+REG,A:GOTO 420
LC 480  IF PEEK(764)<>0 THEN 520
NN 490  REM ■LUMINANCE
NG 500  A=PEEK(708+REG):B=INT(A/16):A=A-B*16:A=
        A+1:IF A>15 THEN A=0
NK 510  A=A+B*16:POKE 708+REG,A:GOTO 420
LG 520  IF PEEK(764)<>6 THEN 550
JC 530  REM ■LEFT ARROW
NJ 540  IF VIEWCNT>0 THEN NOW=XNOW:XNOW=XNOW-1:
        GOTO 610
LN 550  IF PEEK(764)<>7 THEN 580
GI 560  REM ■RIGHT ARROW
NK 570  IF VIEWCNT>0 THEN NOW=XNOW:XNOW=XNOW+1:
        GOTO 610
BI 580  IF PEEK(764)<>15 OR VIEWCNT=0 THEN 620
KF 590  REM ■DOWN ARROW
AI 600  NOW=XNOW:XNOW=XNOW+40:IF XNOW>HOME+2655
         THEN XNOW=HOME+2655
OB 610  U=USR(ADR(MINE$),0,VIEWS(XVIEW),NOW):U=
        USR(ADR(MINE$),1,VIEWS(XVIEW),XNOW):POK
        E 764,255:GOTO 190
ON 620  IF PEEK(764)<>16 THEN 690
LI 630  REM {3 SPACES}■TEN{5 SPACES}
FA 640  IF VIEWCNT=0 THEN ? CHR$(125):? "NO VIE
        WS COMPLETED!":FOR A=0 TO 500:NEXT A:GO
        TO 80
```

```
EA 650  NOW=HOME+2655:U=USR(ADR(MINE$),0,VIEWS(
        VCNT),NOW)
OF 660  VCNT=VCNT+1:IF VCNT>VIEWCNT THEN VCNT=1
LC 670  U=USR(ADR(MINE$),1,VIEWS(VCNT),NOW)
GN 680  GOTO 420
OO 690  IF PEEK(764)<>22 THEN 750
NO 700  REM {3 SPACES}EXCHANGE
HN 710  IF VIEWCNT=0 THEN 640
DP 720  XNOW=HOME+1415:U=USR(ADR(MINE$),0,VIEWS
        (XVIEW),XNOW)
ED 730  XVIEW=XVIEW+1:IF XVIEW>VIEWCNT THEN XVI
        EW=1
GJ 740  U=USR(ADR(MINE$),1,VIEWS(XVIEW),XNOW):G
        OTO 420
BK 750  IF PEEK(764)<>62 THEN 1060
CG 760  REM SCAN
ID 770  IF VIEWCNT=0 THEN 640
HP 780  GRAPHICS 7:NOW=HOME+1042:POKE 764,255
HM 790  FOR X=1 TO VIEWCNT:U=USR(ADR(MINE$),1,V
        IEWS(X),NOW)
GB 800  NOW=NOW+PEEK(VIEWS(X)+6)+1:IF NOW>HOME+
        1070 AND F=0 THEN NOW=HOME+2563:F=1
NG 810  NEXT X:? "ENTER SEQUENCE";:INPUT SEQ$
EN 820  TRAP 820:? :? :? "ENTER DELAY";:INPUT A
LI 830  TRAP 830:? :? :? "GRAPHICS 5 OR 7";:INP
        UT GR:IF GR=5 THEN 850
DP 840  IF GR<>7 THEN 830
ED 850  GRAPHICS GR+16:NOW=PEEK(88)+PEEK(89)*25
        6
DB 860  IF GR=7 THEN NOW=NOW+2215:POKE 208,39:G
        OTO 880
CL 870  NOW=NOW+685:POKE 208,19
EE 880  OLDX=0
GD 890  TRAP 890:B=1
PL 900  X=ASC(SEQ$(B,B))-64
DJ 910  IF X>VIEWCNT OR X<0 THEN X=1
MC 920  IF OLDX=X THEN 940
BO 930  U=USR(ADR(MINE$),0,VIEWS(OLDX),NOW):U=U
        SR(ADR(MINE$),1,VIEWS(X),NOW):OLDX=X
CH 940  IF PEEK(764)<>255 THEN 960
GE 950  FOR Y=0 TO A:NEXT Y:B=B+1:GOTO 900
CA 960  POKE 208,39:POKE 764,255:GRAPHICS 1+16
NM 970  POKE 764,255:? #6:? #6:? #6;" ARE YOU R
        EADY TO ":? #6;" SAVE THIS SERIES?"
OJ 980  A=PEEK(764):IF A=255 THEN 980
LM 990  POKE 764,255:IF A<>43 THEN 80
HA 1000  TRAP 1010:GRAPHICS 0:? "  ENTER FILENA
         ME ";:INPUT FILE$:GOTO 1020
MC 1010  CLOSE #3:GOTO 1000
FA 1020  OPEN #3,8,0,FILE$:PUT #3,VIEWCNT
```

```
CL 1030 FOR X=VIEWS(0) TO MEM:PUT #3,PEEK(X):N
        EXT X:CLOSE #3
EB 1040 ? "THIS SERIES SAVED":? :? "DO YOU WIS
        H TO DO ANOTHER SERIES";:INPUT A$:IF A
        $<>"Y" THEN GRAPHICS 0:END
HO 1050 CLR :GOTO 40
EB 1060 IF PEEK(764)<>42 THEN 1220
AP 1070 REM END DRAWING
LL 1080 LINE=HOME+375:TOP=0
PD 1090 U=USR(ADR(LINE$),LINE):IF PEEK(205)=0
        THEN LINE=LINE+40:TOP=TOP+1:GOTO 1090
PK 1100 LINE=HOME+1415:BOTTOM=26
KN 1110 U=USR(ADR(LINE$),LINE):IF PEEK(205)=0
        THEN LINE=LINE-40:BOTTOM=BOTTOM-1:GOTO
         1110
OO 1120 LINE=HOME+375:LEFT=0
CD 1130 U=USR(ADR(SIDE$),LINE):IF PEEK(205)=0
        THEN LEFT=LEFT+1:LINE=LINE+1:GOTO 1130
LA 1140 LINE=HOME+375+9:RIGHT=9
NB 1150 U=USR(ADR(SIDE$),LINE):IF PEEK(205)=0
        THEN RIGHT=RIGHT-1:LINE=LINE-1:GOTO 11
        50
LL 1160 HGT=BOTTOM-TOP:WIDE=RIGHT-LEFT:VIEWCNT
        =VIEWCNT+1:VIEWS(VIEWCNT)=MEM
IK 1170 TRAP 1220:FOR X=0 TO 4:POKE MEM,PEEK(7
        08+X):MEM=MEM+1:NEXT X:POKE MEM,HGT+1:
        POKE MEM+1,WIDE+1:MEM=MEM+2
PP 1180 A=BOTTOM*40:LINE=HOME+375+A+LEFT
LA 1190 FOR Y=0 TO HGT:FOR X=0 TO WIDE
KD 1200 A=PEEK(LINE+X):POKE MEM,A:MEM=MEM+1:NE
        XT X:LINE=LINE-40:NEXT Y
JM 1210 GOTO 960
EJ 1220 IF PEEK(764)<>58 THEN 1340
LP 1230 REM DIRECTORY
ML 1240 TRAP 1270:POKE 764,255:POKE 82,2
LP 1250 GRAPHICS 0:? :? "  THIS DISK CONTAINS
        THE FILES:":OPEN #3,6,0,"D:*.*"
NC 1260 FOR X=0 TO 5000:GET #3,A:? CHR$(A);:NE
        XT X
LF 1270 CLOSE #3:? "DO YOU WISH TO LOAD AN OLD
         FILE";:INPUT A$:IF A$<>"Y" THEN 80
LI 1280 TRAP 1290:? "ENTER FILENAME";:INPUT FI
        LE$:OPEN #3,4,0,FILE$:GOTO 1300
NG 1290 CLOSE #3:GOTO 1280
MB 1300 GET #3,A:VIEWCNT=VIEWCNT+A:TRAP 1320
OG 1310 GET #3,A:POKE MEM,A:MEM=MEM+1:GOTO 131
        0
II 1320 CLOSE #3:MEM=15000:FOR X=1 TO VIEWCNT
BN 1330 VIEWS(X)=MEM:MEM=MEM+PEEK(MEM+5)*PEEK(
        MEM+6)+7:NEXT X:GOTO 80
```

```
CD 1340 IF PEEK(764)<>61 THEN POKE 764,255:GOT
        O 190
NN 1350 IF GR=5 THEN GR=7:GOTO 1370
KA 1360 IF GR=7 THEN GR=5
HE 1370 POKE 764,255:GOTO 770
EA 1380 REM OPENING
CF 1390 GRAPHICS 2+16:POSITION 5,3
CE 1400 ? #6;"ATAGRAPH"
IH 1410 POSITION 1,5:? #6;"THE BASIC VERSION"
PN 1420 POSITION 3,6:? #6;"OF GRAPHICS"
EC 1440 POSITION 3,8:? #6;"by ed chaney"
JB 1450 FOR T=0 TO 500:NEXT T:RETURN
```

Superplot

Mike Portuesi

With this plotting and drawing utility you'll be able to create displays easily and save them for use in your own programs. Requires at least 24K RAM and a joystick.

COMPUTE!'s Second Book of Atari contains an article called "Plotting Made Easy." The program that accompanies the article allows you to draw on the screen and have the computer write a series of PLOT-DRAWTO statements for use in another program. I've added some improvements to this program and call the new program "Superplot."

How to Use Superplot

Type in Superplot using "The Automatic Proofreader" in Appendix C. Make a few backup copies just in case. Be sure not to run the program until you've saved it a few times. Not only is this program self-modifying, but the computer could also lock up if you fail to type in the machine language data correctly.

When you run the program, there will be a short pause as it initializes. Then you'll be asked a series of questions dealing with color and luminance for the cursor, background, and each of the color registers. Enter the standard Atari color and luminance values for each of these prompts (the same values used in the SETCOLOR statement). When the questions are over, the program sets up its display.

The Indicator Line

You should notice a set of indicators on the bottom of the display. The first two indicators (from left to right) are X= and Y=. These tell the current location of the cursor. The coordinates you see there are the same as the X,Y values of the PLOT statement, and they point to the center blank spot of the cursor. The next indicator, C=, tells you what color you're drawing in. Note that this indicator is the same value as the Atari COLOR statement, and *not* a color register number. The next indicator, LAST=, tells you the last thing you did on the display, whether it was plotting a point or drawing a line. If you just plotted a point on the display, the indicator will read P. If you just drew a line, it will read D. The last indicator is M. It tells you the amount of free memory in bytes. This

serves as a warning in case you're running out of memory. There are also four color bars under the text window. These will be discussed below.

The Cursor

At the upper left of the display is a small cross. This is the cursor. If you don't see it, or if you see only part of it, use the joystick to move it into view.

To plot a point, move the cursor to the desired spot and press the fire button. To draw a line, move the cursor to the endpoint of your intended line and press the fire button again. The computer will draw a line to connect these two. To plot a single point, simply press the fire button twice. This will actually PLOT and DRAWTO a line to the same spot, but when the program writes the BASIC subroutine, it will show up as only one PLOT statement. If you want to connect lines, make the endpoint of the first the beginning of the second by pushing the fire button again after you draw the first line. For example, if you want to draw a square, follow these steps:

1. Move the cursor to the first corner of the square and press the fire button.
2. Move the cursor to the second corner and press the fire button twice. This draws the first side of the square and plots the beginning of the second.
3. Move the cursor to the third corner and press the fire button twice.
4. Continue this process until the square is completed.

When the program writes the BASIC subroutine, the instructions for the square will come out as one PLOT statement and a series of DRAWTOs.

The Color Bars

The color bars are used to change the color you wish to draw with. To change the drawing color, simply move the cursor over the appropriate bar and push the fire button. The next line you draw will be in the color selected. Note that the blank space in the middle of the cursor must be over the color bar to make the change.

The Console Keys

Each of the console keys performs a different function.

OPTION. When you press this key, the program will ask for an output device. Enter **C:** to save your BASIC subroutine to cassette, **D:*filename.ext*** to save your routine to disk, or **P:** to list your routine out to the printer. This program saves the routine in LIST format. To enter it back in, use the ENTER command. If you saved it on disk, use **ENTER "D:*filename.ext*"** to retrieve it from disk, and if you saved it to cassette, use **ENTER "C:"**.

After the program saves your routine, it will ask if you want to clear it out in memory. Answering Y will result in the computer deleting all the lines it has added to the program. It will then return you to graphics mode 7 with a clean display, ready for a new picture. If you don't want to save your subroutine, but simply want to get rid of it, push the RETURN key in response to the "Enter output device?" prompt. It will then ask if you want to clear out the lines.

SELECT. This key is used to change the colors in each color register. The computer will ask for the color register number (see the chart above to pick the right one), the Atari color value, and the luminance of that color. It will then change the color in that register.

START. Pressing this key will cause the computer to add statements to the program. All the lines you have drawn will automatically be converted to PLOT-DRAWTO statements. They will be stored with line numbers beginning at 20000, continuing upward by increments of ten. If you don't press START, the program will automatically update itself after every 20 lines you draw on the screen. Note that this option will not work if you have just plotted a point on the display. You must have just completed drawing a line for this option to work.

The Space Bar

If you make a mistake drawing a line, hit the space bar. The program will delete the last line drawn on the display. Each time you hit the space bar, the computer will erase the previous line drawn. You can erase as many lines as you wish this way.

There are a few restrictions on its use, however. The first is that you cannot delete a line whose coordinates have already been made part of the program. If you want to do this, just edit the finished subroutine. The second limitation is that the program uses the background color to erase the offending line from the display. In doing this, it may inadvertently erase parts of other lines as well. This will not affect program operation in any way, and when the program modifies itself, it will restore the erased parts as it redraws the display.

Like the START key, this option will not work if you have just plotted a point. If you plotted a point in the wrong spot, finish the line before using the space bar. The last thing to note about this option is the way it handles the current drawing color.

Superplot

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
OI 30  DIM MEM$(1),CURSOR$(7),PX(20),PY(20),DRX
       (20),DRY(20),C(20),COL(3),DEVICE$(15),AN
       $(1),JOY$(29)
BC 40  LINENO=20000:SETPM=9020:INTARRAY=7020:SE
       TDIS=8020:SAVLIN=11020
NM 50  PLTDR=2020:CHNGCOL=1020:SWITCH=3020:MODP
       ROG=6020:DELCOMND=4020:INCOLOR=5020:INIT
       ML=10020:UNTRAP=40000
NA 60  FLAG=0:P=1:C=1
NI 70  GOSUB INTARRAY:GOSUB INITML
HI 80  GRAPHICS 0:? :? :POKE 201,9:? ,"*** SUPE
       R PLOT ***":POKE 201,10
PA 90  ? :TRAP 90:? "ENTER CURSOR COLOR";:INPUT
        CC:TRAP UNTRAP
HF 100  IF CC<0 OR CC>15 THEN 90
OP 110  TRAP 110:? "ENTER CURSOR LUMINANCE";:IN
       PUT CL:TRAP UNTRAP
LB 120  IF CL<0 OR CL>14 THEN 110
MO 130  TRAP 130:? "ENTER BACKGROUND COLOR";:IN
       PUT BC:TRAP UNTRAP
KC 140  IF BC<0 OR BC>15 THEN 130
PI 150  TRAP 150:? "ENTER BACKGROUND LUMINANCE"
       ;:INPUT BL:TRAP UNTRAP
LH 160  IF BL<0 OR BL>14 THEN 150
AG 170  COL(3)=16*BC+BL
AL 180  FOR I=0 TO 2
```

```
IB 190 TRAP 190:? "ENTER COLOR FOR REG. ";I;:I
       NPUT RC:TRAP UNTRAP
MF 200 IF RC<0 OR RC>15 THEN 190
JJ 210 TRAP 210:? "ENTER LUMINANCE FOR REG. ";
       I;:INPUT RL:TRAP UNTRAP
NB 220 IF RL<0 OR RL>14 THEN 210
DJ 230 COL(I)=16*RC+RL
BO 240 NEXT I
NH 250 BASE=PEEK(106)-16:POKE 106,BASE
OF 260 GRAPHICS 7:COLOR C:GOSUB SETDIS:GOSUB S
       ETPM
IC 265 FOR I=0 TO 2:POKE 708+I,COL(I):NEXT I:P
       OKE 712,COL(3)
MG 280 DX=USR(JOY,0,0)-1
MJ 290 DY=USR(JOY,0,1)-1
DI 300 X=X+DX:Y=Y+DY
EM 310 IF X>159 THEN X=0
EL 320 IF X<0 THEN X=159
AK 330 IF Y>95 THEN Y=0:MEM$(1)="{,}":MEM$(255
       )="{,}":MEM$(2)=MEM$
AJ 340 IF Y<0 THEN Y=95:MEM$(1)="{,}":MEM$(255
       )="{,}":MEM$(2)=MEM$
CG 350 MEM$(Y+14,Y+20)=CURSOR$
OH 360 POKE 53248,X+46
OE 370 POKE 656,0:POKE 657,5:? X;"  ";:POKE 65
       7,13:? Y;"  ";
EH 380 IF Y>91 AND ABS(STRIG(0)-1) THEN GOSUB
       CHNGCOL:GOTO 280
OP 390 IF ABS(STRIG(0)-1) THEN GOSUB PLTDR
BO 400 KEY=PEEK(53279):IF KEY<>7 THEN GOSUB SW
       ITCH
KJ 405 IF PEEK(764)=33 THEN POKE 764,255:GOSUB
        DELCOMND
GI 410 GOTO 280
KD 1000 REM CHNGCOL
GN 1010 REM Changes current drawing color
GK 1020 C=INT(X/40)
IG 1030 COLOR C
NA 1040 POKE 656,0:POKE 657,21:PRINT C;
CK 1050 C(P)=C
KH 1060 RETURN
CM 2000 REM PLTDR
OF 2010 REM PLOTs, DRAWTOs, and saves coordina
       tes
PB 2020 IF FLAG THEN DRAWTO X,Y:POKE 656,0:POK
       E 657,30:? "D";:DRX(P)=X:DRY(P)=Y:P=P+
       1:GOTO 2050
BD 2030 PLOT X,Y:PX(P)=X:PY(P)=Y
NE 2040 POKE 656,0:POKE 657,30:? "P";
LN 2050 FLAG=ABS(FLAG-1)
```

```
DH 2060 IF P=21 THEN P=20:GOSUB MODPROG
MP 2065 IF STRIG(0)=0 THEN 2065
NM 2070 POKE 77,0:RETURN
HJ 3000 REM SWITCH
FG 3010 REM Checks console switches
CJ 3020 IF KEY=3 THEN GOSUB SAVLIN
HF 3030 IF KEY=5 THEN GOSUB INCOLOR
OO 3040 IF KEY=6 THEN P=P-1:GOSUB MODPROG
KI 3050 RETURN
OO 4000 REM DELCOMND
FG 4010 REM Deletes last PLOT-DRAWTO and COLOR
         (if one was made)
FI 4020 IF P=1 THEN RETURN :REM just finished
        saving screen
NB 4025 IF FLAG THEN RETURN :REM just PLOTted,
         must DRAWTO first
HM 4030 COLOR 0:PLOT PX(P-1),PY(P-1):DRAWTO DR
        X(P-1),DRY(P-1)
IK 4040 COLOR C
LJ 4050 IF C(P-1)=-1 THEN P=P-1:RETURN
KM 4055 IF P=2 THEN C=C(1):COLOR C:POKE 657,21
        :? C;:P=P-1:RETURN
KH 4060 C(P-1)=-1
FE 4070 FOR I=P-1 TO 1 STEP -1
HL 4080 IF C(I)<>-1 THEN COLOR C(I):C=C(I):POK
        E 657,21:? C;:P=P-1:POP :RETURN
OE 4090 NEXT I:P=P-1:RETURN
LP 5000 REM INCOLOR
PO 5010 REM Change value of user-specified col
        or register
LJ 5020 GOSUB 5095:POKE 656,1:POKE 657,2:? "En
        ter register to change";:TRAP 5020:INP
        UT REG:TRAP UNTRAP
HH 5030 IF REG<0 OR REG>4 OR REG=3 THEN 5020
OF 5040 GOSUB 5095:POKE 656,1:POKE 657,2:? "En
        ter color";:TRAP 5040:INPUT RC:TRAP UN
        TRAP
CM 5050 IF RC<0 OR RC>15 THEN 5040
IP 5060 GOSUB 5095:POKE 656,1:POKE 657,2:? "En
        ter luminance";:TRAP 5060:INPUT RL:TRA
        P UNTRAP
EB 5070 IF RL<0 OR RL>14 THEN 5060
KB 5080 SETCOLOR REG,RC,RL
CF 5085 GOSUB 5095
IJ 5087 IF REG=4 THEN COL(3)=16*RC+RL:RETURN
AO 5088 COL(REG)=16*RC+RL
KO 5090 RETURN
PP 5093 REM Short routine to clear prompt line
HL 5095 POKE 656,1:POKE 657,2:? "{36 SPACES}";:
        RETURN
```

```
LM 6000 REM MODPROG (The biggie!)
CH 6010 REM Modifies program to add statements
FB 6020 IF P=0 OR FLAG THEN P=P+1:RETURN
AF 6025 POKE 53248,0:POKE 559,34:POKE 54286,64
EK 6030 GRAPHICS 0
MP 6040 POSITION 2,2
FE 6050 FOR I=1 TO P:? LINENO;" ";
JB 6055 IF C(I)<>-1 THEN PRINT "C. ";C(I);":";
AD 6060 IF I=1 THEN 6070
ON 6065 IF PX(I)=DRX(I-1) AND PY(I)=DRY(I-1) T
        HEN ? "DR. ";DRX(I);",";DRY(I):GOTO 60
        85
KH 6070 IF PX(I)=DRX(I) AND PY(I)=DRY(I) THEN
        ? "PL. ";PX(I);",";PY(I):GOTO 6085
LJ 6075 ? "PL. ";PX(I);",";PY(I);":";
EH 6080 ? "DR. ";DRX(I);",";DRY(I)
CG 6085 LINENO=LINENO+10
EP 6100 NEXT I
DP 6110 GOSUB 11500
IF 6140 GRAPHICS 7:GOSUB SETDIS:GOSUB 20000
NH 6145 ATAB=PEEK(140)+256*PEEK(141):AD=BASE*2
        56+512:OFFSET=AD-ATAB
JD 6147 HIGH=INT(OFFSET/256):LOW=OFFSET-256*HI
        GH:POKE VTAB+2,LOW:POKE VTAB+3,HIGH
MD 6148 POKE 559,46:POKE 53248,X+46
HE 6150 GOSUB INTARRAY:P=1:FOR I=0 TO 2:POKE 7
        08+I,COL(I):NEXT I:POKE 712,COL(3):JOY
        =ADR(JOY$)
KN 6160 RETURN
BF 7000 REM INTARRAY
KM 7010 REM Initializes PX(, PY(, DRX(, DRY(,
        C( arrays
GM 7020 FOR I=1 TO 20
GD 7030 PX(I)=0:PY(I)=0
PA 7040 DRX(I)=0:DRY(I)=0
EE 7050 C(I)=-1
FF 7060 NEXT I
BD 7070 C(1)=C
KP 7080 RETURN
HI 8000 REM SETDIS
OG 8010 REM Sets up display at bottom of scree
        n
FK 8020 POKE 656,0:POKE 752,1
DJ 8030 POKE 657,2:? "X=";X;"   ";
GL 8040 POKE 657,10:? "Y=";Y;"   ";
MJ 8050 POKE 657,18:? "C=";C;
CP 8060 POKE 657,33:? "M=";FRE(0);
CF 8065 POKE 656,0:POKE 657,24:? "L=5TE=";
FI 8070 POKE 656,2:POKE 657,11:? "*** SUPER PL
        OT ***";
```

```
KC 8080 DLIST=PEEK(560)+256*PEEK(561)
HI 8090 POKE DLIST+90,8
DB 8095 TXTMEM=PEEK(DLIST+86)+256*PEEK(DLIST+8
        7):RESTORE 8107
NF 8100 FOR I=TXTMEM+120 TO TXTMEM+120+9:READ
        BYTE:POKE I,BYTE:NEXT I
OK 8107 DATA 0,0,5,85,85,170,170,175,255,255
BO 8108 POKE DLIST+89,130:POKE DLIST+84,141
IL 8110 POKE 512,0:POKE 513,6:POKE 54286,192
KL 8120 RETURN
DG 9000 REM SETPM
CB 9010 REM Initializes PM graphics
EE 9020 VTAB=PEEK(134)+256*PEEK(135)
CK 9030 ATAB=PEEK(140)+256*PEEK(141)
IJ 9040 BASE=BASE+12:POKE 54279,BASE
AN 9050 OFFSET=(BASE*256+512)-ATAB
DP 9060 HIGH=INT(OFFSET/256):LOW=OFFSET-256*HI
        GH
OH 9070 POKE VTAB+2,LOW:POKE VTAB+3,HIGH
DL 9080 POKE VTAB+4,0:POKE VTAB+5,1
EA 9090 POKE VTAB+6,0:POKE VTAB+7,1
II 9095 MEM$(1)="{,}":MEM$(255)="{,}":MEM$(2)=
        MEM$
JB 9100 POKE 704,16*CC+CL
GO 9110 RESTORE 9130:FOR I=1 TO 7
OM 9120 READ BYTE:CURSOR$(I,I)=CHR$(BYTE):NEXT
         I
HM 9130 DATA 0,32,32,216,32,32,0
FM 9140 MEM$(Y+14,Y+20)=CURSOR$
BN 9150 POKE 53248,X+46
AI 9160 POKE 559,46:POKE 53277,3
LB 9170 RETURN
KC 10000 REM INITML
OL 10010 REM Initializes m/l interrupt routine
         s & joystick driver
AO 10020 RESTORE 10500:Z=0
FO 10025 IF PEEK(1536)=72 AND PEEK(1597)=64 TH
         EN 10060
PM 10030 READ BYTE:IF BYTE=-1 THEN POKE 512,0:
         POKE 513,6:GOTO 10060
NI 10040 POKE 1536+Z,BYTE
KK 10050 Z=Z+1:GOTO 10030
GL 10060 JOY=ADR(JOY$):RESTORE 10600:FOR Z=1 T
         O 29:READ BYTE:JOY$(Z,Z)=CHR$(BYTE):N
         EXT Z:RETURN
MD 10500 DATA 72,138,72,169,10,162,148,141,10,
         212,141,23,208,142,24,208,169,6,162,3
         0,141,1,2,142,0,2,104,170,104
```

```
GF 10510 DATA 64,72,138,72,173,197,2,174,198,2
         ,141,10,212,141,23,208,142,24,208,169
         ,6,162,0,141,1,2,142,0,2,104,170
DN 10520 DATA 104,64
HB 10530 DATA -1
AO 10600 DATA 104,104,104,170,104,189,120,2,40
         ,176,2,74,74,41,3,56,233,2,16,2,169,2
         ,133,212,169,0,133,213,96
KD 11000 REM SAVLIN
PI 11010 REM Saves lines to output device
IC 11020 POKE 656,1:POKE 657,2:? "Enter output
          device";:INPUT DEVICE$:IF DEVICE$=""
          THEN 11040
FL 11030 POKE 54286,64:LIST DEVICE$,20000,3276
         7:POKE 512,0:POKE 513,6:POKE 54286,19
         2
LH 11040 GOSUB 5095:POKE 656,1:POKE 657,2:? "C
         lear lines (Y/N)";:INPUT AN$
KE 11050 IF AN$="N" THEN GOSUB 5095:RETURN
HG 11060 IF AN$<>"Y" THEN 11040
DB 11070 POKE 53248,0:POKE 559,34:POKE 54286,6
         4
IC 11080 GRAPHICS 0:POSITION 2,2:COUNTER=1
HA 11090 FOR Z=20000 TO LINENO STEP 10
DH 11100 ? Z:COUNTER=COUNTER+1:IF COUNTER=20 T
         HEN COUNTER=1:GOSUB 11500:? CHR$(125)
         :POSITION 2,2
DO 11110 NEXT Z:GOSUB 11500
EE 11120 GRAPHICS 7:GOSUB SETDIS:ATAB=PEEK(140
         )+256*PEEK(141):AD=BASE*256+512:OFFSE
         T=AD-ATAB
LH 11130 HIGH=INT(OFFSET/256):LOW=OFFSET-256*H
         IGH:POKE VTAB+2,LOW:POKE VTAB+3,HIGH
OL 11135 POKE 559,46:POKE 53248,X+46
MN 11140 GOSUB INTARRAY:P=1:LINENO=20000:FOR I
         =0 TO 2:POKE 708+I,COL(I):NEXT I:POKE
          712,COL(3):JOY=ADR(JOY$)
NI 11150 RETURN
BC 11160 REM Short routine to do modifications
AA 11500 ? "CONT":POSITION 0,0:POKE 842,13:STO
         P
GO 11510 POKE 842,12:RETURN
```

Graphics Plus

James Luczak

"Graphics Plus" is a utility that lets you design playfield graphics quickly and easily. It also produces the BASIC code so that you can use your creations in your own programs.

"Graphics Plus" is a feature-packed utility. Here is a list of its features:

- Graphics Plus is menu-driven, so there are no complicated sequences of operations to remember.
- Pictures can be drawn in graphics modes 3–8.
- Circles and ellipses or any part of a circle or ellipse can be drawn—just give size and location.
- It draws lines.
- All positioning is done by using cursor movement. Position the cursor with the arrow keys and press the appropriate key (the menu will keep you informed) and you're on your way.
- Text can be added with graphics in any graphics mode.
- Each drawing stored is assigned different line numbers. This allows many drawings to be stored without losing previously stored pictures.
- Graphics Plus saves as many pictures as memory allows.
- Any or all drawings can be erased from storage.
- Pictures drawn in one mode can be displayed in any other graphics mode.
- Graphics Plus can be used just as easily with cassette as with disk.

Easy to Use

Everything you need to know to use Graphics Plus appears in the text window at the bottom of the screen. The only exception is the color command. By pressing 0, 1, 2, or 3 you will change the color you are drawing with (modes 3, 5, 7). The menu will show what color is being used. Remember, color 2 will not show up in modes 4, 6, and 8.

As an example, let's draw a square in graphics mode 5 and save it. As Graphics Plus initializes, it puts you into graphics mode 7. So to draw in mode 5, press G. Now choose mode 5 by pressing 5. You will return to the main menu. Notice that the mode has changed in the text window. To draw a line, choose option L. This will put you into the line draw

mode. A flashing cursor should be at the upper-left corner of the screen. Move the cursor around by using the arrow keys. Notice that the X,Y coordinates are shown in the text window. Also, the cursor will wrap around the screen vertically and horizontally.

Let's draw the square. Place the cursor wherever you like. Then press P. You will hear a beep and the word PLOT will appear in the text window. Now move the cursor to where you want the line to end and press D. You will hear another beep and the word DRAWTO will take the place of PLOT in the text window. At the same time a line will be drawn from where you pressed P to where you pressed D. Press P again, move the cursor to the next corner of the square, and then press D. (The DRAWTO command will always draw the line from the position of the cursor when you last pressed P to the cursor's present position.) Continue in this manner until you have a square.

If you want to remove the entire picture and start over, return to the main menu (press RETURN), select the graphics option (press G), and press 5 (for graphics mode 5). This will clear the screen. You can reenter the line draw mode by pressing L.

If you want to remove only part of the picture, press 0. This will cause your next PLOT and DRAWTO to be in the background color, erasing the line (this works with circles and ellipses also).

Saving Your Masterpiece

Now that you have a picture, let's save it. Return to the main menu (press RETURN), then press P (picture submenu). The picture option will appear in the text window. Since we want to save the picture, press S. Watch the text window for instructions. Using the arrow keys, position the cursor in the upper-left corner of the square. Make sure you place the cursor there. If you place it in the upper-left corner of the screen, you will save the entire screen (modes 3, 4, and 5 only). After the cursor is positioned, press S. You will hear a double beep. Now, using the arrow keys, move the cursor to the lower-right corner of the square, and press S again.

The program will start to store the picture into a string (A$). There will be a delay, depending on how big the picture is and what graphics mode you're in. After the picture has been saved, you'll be asked to assign it a number. Assign it

the number 3. Press 3, then RETURN. The program will now enter the RETURN key mode and add the picture to the program starting at line 10300. If you assigned the picture the number 5, the saved picture would start at line 10500. Assigning the picture the number 0 would start saving the picture at line 10000. After the picture has been saved, you'll be returned to the main menu. Saving the picture only saves it to memory; you'll need to use the quit option to resave the entire program (more on this below).

If you want to save a picture and assign it the number of an already existing picture, don't worry. Graphics Plus will automatically check to see if there is a picture stored with that number and will erase it. Save the new picture in its place.

This may sound a bit complicated, but once you try it, it becomes easy. Remember, the text window will always display all the information you need to work with.

Some Options

The program is quite simple to use since instructions appear on the screen as needed. Let's just review a few other options.

In the picture submenu (press P in the main menu) the view option allows you to view the pictures that you have stored. For instance, if you've stored pictures 1, 2, 3, 4, and 5, and you enter the number 1, all of your pictures will be displayed one after the other. If you choose 3 instead of 1, pictures 3–5 will be displayed. After the last picture is displayed, the main menu returns.

Main menu options C and E (circle and ellipse) are easy to use. Position the cursor and provide the program with the information it requests. Keep an eye on the text window. When you position the cursor, you'll be asked if the position is correct. Answer yes or no. After you've entered all the necessary information, a summary of the circle or ellipse information will appear. Again you'll be asked if it's correct or not. If you answer yes, the circle or ellipse will be drawn. It's a good idea to write down this information. Later, if you want to erase the circle or ellipse, all that you'll need to do is use color 0 and enter the same parameters.

The comments option (K) allows text in any graphics mode. "Textplot" is used for text in graphics modes 3–7. Text in graphics mode 8 will be normal size. However, text in modes 3–7 will be larger than normal, depending on which

mode you're in. For instance, text in graphics 3 is really big. The article "Textplot" (*COMPUTE!'s First Book of Atari Graphics*) will give you some ideas of what can be done with Textplot.

The Quit option (Q) is very important. All pictures saved with Graphics Plus are only temporarily stored. The only time pictures are permanently saved is when you use the Quit option. When Quit is used, the entire program is saved to disk or cassette, including the pictures you have saved. So, remember, if you want your drawings permanently saved, use the Quit option. As printed here, the program will be saved to disk when you use the Quit option. If you are a cassette user, you'll need to substitute this line for line 3700:

```
3700 IF CHR$(A)="Q" THEN CSAVE:GRAPHICS 0:CLR:END
```

Adding Pictures to Programs

Program 2 is an example of how to use the data created by Graphics Plus in your own programs.

Start by listing the line numbers that contain the picture data you want into a temporary storage file. If you want to use, say, picture 5, the picture data will start with line 10500, and the last line with data for picture 5 will be the line number with a return.

Example:

LIST " D: PICT 5 ", 10500, 10507 (Disk)
LIST "C:",10500, 10507 (Cassette)

Type in or load Program 2 and then enter the picture data.

Example:

ENTER " D: PICT 5" (Disk)
ENTER "C:" (Cassette)

That's all there is to it. By adjusting the X,Y coordinates, you can place your picture anywhere on the screen, as many times as you like.

Program 1. Graphics Plus

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
PM 500 GRAPHICS 2:POSITION 5,2:? #6;"GRAPHICS"
DI 520 POSITION 7,4:? #6;"PLUS"
LJ 540 CLR :GOTO 3180
BH 560 FOR Y=N TO Q:FOR X=M TO P:LOCATE X,Y,R:
       IF R=0 THEN R=4
```

```
PL 580 FL1=FRE(0)
CN 600 S=S+1:IF FL1<800 THEN 3060
DO 620 A$(S,S)=CHR$(R+64):NEXT X:NEXT Y:T1=P-M
       +1:U1=Q-N:RETURN
EA 640 SOUND 0,SD,10,10:FOR DELAY=1 TO 15:NEXT
        DELAY:SOUND 0,0,0,0:RETURN
GC 660 FOR I=1 TO AQ:? "{RIGHT}";:NEXT I:RETUR
       N
BP 680 ? CHR$(125):? "{UP}";D$;" [GR.]";B;" ";G
       RPX;"X";GRPY;" [X=]";X;" [Y=]";Y:RETURN
NP 700 ? CHR$(125):? "{UP}USE ARROWS TO MOVE C
       URSOR":? "[P]=PLOT [D]=DRAWTO [RETURN]=
       MENU":TI=1:TIM=0
EF 720 IF TIM THEN SOUND 0,45,10,8:FOR DELAY=1
        TO 5:NEXT DELAY:SOUND 0,0,0,0
PF 740 IF X>GRPX THEN X=0
PF 760 IF X<0 THEN X=GRPX
PM 780 IF Y>GRPY THEN Y=0
PD 800 IF Y<0 THEN Y=GRPY
AG 820 LOCATE X,Y,QZ
FO 840 IF QZ=0 THEN F=F5:G=0
AI 860 IF QZ=1 OR QZ=2 OR QZ=3 THEN F=0:G=QZ
NH 880 TI=1:IF TIM>8 AND B>=6 THEN TI=4
MJ 900 COLOR F:PLOT X,Y:FOR DELAY=1 TO BJ:NEXT
        DELAY
LH 920 COLOR G:PLOT X,Y
KC 940 ? "X=";X,"Y=";Y,"[COLOR] ";F5;" ";C$;"
       {UP}":IF DRP=1 THEN COLOR F:PLOT LPX,LP
       Y
NM 960 IF PEEK(764)=50 THEN POKE 764,255:F5=0:
       GOTO 720
NO 980 IF PEEK(764)=31 THEN POKE 764,255:F5=1:
       GOTO 720
PO 1000 IF PEEK(764)=30 THEN POKE 764,255:F5=2
        :GOTO 720
AG 1020 IF PEEK(764)=26 THEN POKE 764,255:F5=3
        :GOTO 720
EI 1040 IF PEEK(764)=7 THEN POKE 764,255:X=X+T
        I:TIM=TIM+1:GOTO 720
EL 1060 IF PEEK(764)=6 THEN POKE 764,255:X=X-T
        I:TIM=TIM+1:GOTO 720
HN 1080 IF PEEK(764)=15 THEN POKE 764,255:Y=Y+
        TI:TIM=TIM+1:GOTO 720
HH 1100 IF PEEK(764)=14 THEN POKE 764,255:Y=Y-
        TI:TIM=TIM+1:GOTO 720
FL 1120 IF K=1 OR K=2 THEN 1220
LK 1140 IF PEEK(764)=10 THEN POKE 764,255:LPX=
        X:LPY=Y:DRP=1:C$="[PLOT]":SD=120:GOSU
        B 640:TIM=0:GOTO 720
```

```
LA 1160 IF PEEK(764)=58 THEN POKE 764,255:COLO
        R F5:PLOT X,Y:DRAWTO LPX,LPY:DRP=0:C$=
        " DRAWTO ":SD=80:GOSUB 640:TIM=0:GOTO
        720
BH 1180 IF PEEK(764)=12 THEN POKE 764,255:GOTO
         3460
PI 1200 POKE 764,255:TIM=0:GOTO 720
LJ 1220 IF PEEK(764)=62 THEN POKE 764,255:SD=6
        0:GOSUB 640:RETURN
KB 1240 IF PEEK(764)=12 AND K=2 THEN POKE 764,
        255:GOTO 3460
PO 1260 POKE 764,255:TIM=0:GOTO 720
NE 1280 K=1:? CHR$(125):? "{UP}POSITION CURSOR
         TO UPPER LEFT CORNER OF PICTURE TO BE
         SAVED.THEN PRESS "S"":GOSUB 720
IP 1300 M=X:N=Y:SD=45:GOSUB 640
CM 1320 ? CHR$(125):? "{UP}POSITION CURSOR TO
        LOWER RIGHT CORNER OF PICTURE TO BE SA
        VED.THEN PRESS "S"":GOSUB 720
KD 1340 P=X:Q=Y:SD=25:GOSUB 640:S=0:K=0:A$=""
KI 1360 IF P<M OR Q<N THEN 1280
MK 1380 IF (P-M)*(Q-N)>3299 THEN 3060
CM 1400 ? CHR$(125):? "{8 SPACES}*** SAVING PI
        CTURE ***":GOSUB 560
EK 1420 GRAPHICS 0:? CHR$(125):POKE 84,10:POKE
         85,6:? " SAVE AS PICTURE NUMBER "
BK 1440 ? :POKE 85,6:? "PICTURES SAVED ";
PN 1460 R1=1480:TE=1:R=0:R2=0:GOTO 2380
MM 1480 TE=0:R=0
OC 1500 ? :POKE 85,6:? "ENTER PICTURE NUMBER "
        ;:TRAP 1520:INPUT SR:TRAP 40000:GOTO 1
        540
NI 1520 ? "{3 UP}":GOTO 1500
GN 1540 SM=1:R1=1560:TE=1:SS=0:GOTO 2660
AI 1560 T=T1:U=U1:TE=0:SM=0:SS=0
LH 1580 POKE 82,2:SR1=100:SR2=(SR*100)+10000:S
        R3=LEN(A$):SR4=0:SS=0:SR5=SR1-99:SR6=0
ON 1600 GRAPHICS 0:? CHR$(125):POSITION 2,5
PJ 1620 ? SR2+SR4;" REM":SR7=SR2+SR4:SR4=SR4+1
IO 1640 ? SR2+SR4;" IF TE=1 THEN RETURN":SR4=S
        R4+1
PK 1660 IF SR3>100 THEN 1700
KC 1680 ? SR2+SR4;" A$=";CHR$(34);A$;CHR$(34):
        SS=1:SR4=SR4+1:GOTO 1720
CI 1700 ? SR2+SR4;" A$(";SR5;")=";CHR$(34);A$(
        SR5,SR1);CHR$(34):SR1=SR1+95:SR4=SR4+1
        :SR5=SR1-94:SR6=SR6+1
DC 1720 IF SS=1 THEN ? SR7;" T=";T;":U=";U;":B
        1=";B;":SU=";SR4;":SQ=";SR:? SR2+SR4;"
        RETURN":GOTO 1800
```

```
FK 1740 IF SR6>3 THEN SR6=0:GOTO 1800
EA 1760 IF SR3>SR1 THEN 1700
EE 1780 SR1=SR3:SS=1:GOTO 1700
MD 1800 POSITION 0,0:POKE 842,13:POSITION 2,22
        :? "CONT"
FA 1820 POSITION 2,2:STOP
CJ 1840 POKE 842,12
DK 1860 ? CHR$(125):POSITION 2,5
FP 1880 IF SS=1 THEN A$="":TRAP 3100:GRAPHICS
        B:TRAP 40000:POKE 82,0:GOTO 3460
NB 1900 GOTO 1760
LE 1920 ? CHR$(125):? "{UP}{6 SPACES}ENTER PI
        CTURE NUMBER TO LOAD"
FE 1940 ? "PICTURES AVAILABLE ARE ";
AH 1960 R1=1980:TE=1:R=0:R2=0:GOTO 2380
OD 1980 TE=0:R=0:? :? "ENTER PICTURE NUMBER ";
        :TRAP 2000:INPUT SR:TRAP 40000:GOTO 20
        20
NO 2000 ? "{3 UP}":GOTO 1980
KE 2020 SR1=(SR*100)+10000
LG 2040 TRAP 2200:GOSUB SR1:TRAP 40000
GO 2060 IF SET=1 THEN B=B1:X=0:Y=0:TRAP 3100:G
        RAPHICS B:TRAP 40000:GOTO 2100
OP 2080 K=2:? CHR$(125):? "{UP}USE CURSOR TO P
        OSITION LOCATION OF PICTURE.THEN PRESS
         S ":GOSUB 720:K=0
JI 2100 S=1:FOR W=Y TO Y+U:POSITION X,W
DM 2120 TRAP 3460:? #6;A$(S,S+T-1):S=S+T:TRAP
        40000
FN 2140 NEXT W
OB 2160 IF SET=1 THEN GOTO 2500
NB 2180 GOTO 3460
DF 2200 IF SET=1 THEN SET=0
MM 2220 GOTO 3460
FC 2240 ? CHR$(125):? "{UP}SELECT PICTURE # T
        O VIEW FROM TO LAST"
FA 2260 ? "PICTURES AVAILABLE ARE ";
PG 2280 R1=2300:TE=1:R=0:R2=0:GOTO 2380
ME 2300 TE=0:R=0
AH 2320 ? :? "ENTER PICTURE # ";:TRAP 2340:INP
        UT SR:TRAP 40000:GOTO 2360
NK 2340 ? "{3 UP}":GOTO 2320
IN 2360 SET=1:R2=0:GOTO 2440
EI 2380 TRAP 2400:GOSUB 10000+(R*100):TRAP 400
        00:? SQ;",";:R=R+1:R2=0:GOTO 2380
AC 2400 IF R2<15 THEN R2=R2+1:R=R+1:GOTO 2380
IE 2420 GOTO R1
BL 2440 TRAP 2460:GOSUB 10000+(SR*100):TRAP 40
        000:R2=0:GOTO 2020
```

```
KL 2460 IF R2<15 THEN R2=R2+1:SR=SR+1:GOTO 244
        Ø
GH 2480 SET=Ø:GOTO 3460
KD 2500 SR=SR+1:A$="":SD=50:GOSUB 640:SD=75:GO
        SUB 2520:GOSUB 640:FOR DELAY=1 TO 400:
        NEXT DELAY:TRAP 3100:GRAPHICS B:TRAP 4
        0000:GOTO 2440
ID 2520 ? CHR$(125):? "{UP}{7 SPACES}*** GRAP
        HICS MODE ***= ";B:? "{7 SPACES}*** PI
        CTURE NUMBER ***= ";SQ:RETURN
OH 2540 ? CHR$(125):? "{UP}{9 SPACES}** ERASE
        OPTION **":SD=40:GOSUB 640:SD=80:GOS
        UB 640:SD=160:GOSUB 640
OJ 2560 ? "PICTURES AVAILABLE ";:R1=2580:R=Ø:R
        2=Ø:TE=1:GOTO 2380
PF 2580 R=Ø:TE=Ø:? :? "PICTURE # TO *** ERASE
        ***":? "( A FOR ALL)( R FOR MENU) ";
        :INPUT SR$:IF SR$="A" THEN SR=Ø:GOTO 2
        980
OP 2600 IF SR$="R" THEN 3460
CA 2620 TRAP 2640:SR=VAL(SR$):TRAP 40000:SM=Ø:
        SS=Ø:GOTO 2660
NA 2640 GOTO 2540
KL 2660 SR1=10000+(SR*100):SR2=Ø:SR3=Ø
HP 2680 TRAP R1:GOSUB SR1:TRAP 40000
CA 2700 GRAPHICS Ø:? CHR$(125):POKE 82,2:POSIT
        ION 2,5
MO 2720 IF SR2>15 THEN 2800
JK 2740 IF SR3>SU THEN SS=1:GOTO 2800
OH 2760 ? SR1+SR3
NJ 2780 SR2=SR2+1:SR3=SR3+1:GOTO 2720
ME 2800 POSITION Ø,Ø:POKE 842,13:POSITION 2,22
        :? "CONT"
FB 2820 POSITION 2,2:STOP
CK 2840 POKE 842,12
DL 2860 ? CHR$(125):POSITION 2,5
BF 2880 IF SS=1 AND SM=1 THEN GOSUB 2960:GOTO
        1560
FC 2900 IF SS=1 AND SN=1 THEN 2980
MG 2920 IF SS=1 THEN GOSUB 2960:TRAP 3100:GRAP
        HICS B:TRAP 40000:POKE 82,Ø:GOTO 3460
FD 2940 SR2=Ø:GOTO 2740
LF 2960 R2=Ø:SR2=Ø:SR3=Ø:SS=Ø:SM=Ø:SN=Ø:RETURN

PI 2980 R2=Ø:SR2=Ø:SR3=Ø:SS=Ø:IF SN=1 THEN SR=
        SR+1
HO 3000 SR1=(SR*100)+10000:SN=1:TRAP 3020:GOSU
        B SR1:TRAP 40000:R2=Ø:GOTO 2700
JN 3020 IF R2<15 THEN R2=R2+1:SR=SR+1:GOTO 300
        Ø
```

```
OO 3040 GOSUB 2960:POKE 82,0:TRAP 3100:GRAPHIC
        S B:TRAP 40000:GOTO 3460
KN 3060 REM
BD 3080 POKE 752,1:? CHR$(125):? "{UP}■■■ INS
        UFFICIENT RAM TO SAVE PICTURE ■■■":GOT
        O 3140
NM 3100 IF FM=1 THEN B=B-1:GOTO 3400
BO 3120 POKE 752,1:? CHR$(125):? "{UP}■■■ INS
        UFFICIENT RAM FOR GR.MODE ■ ";B;" ■■■":
        B=B-1:FM=1
CP 3140 FOR DELAY=0 TO 150:? "{14 SPACES}TIMEOU
        T=  {3 LEFT}";150-DELAY;"{UP}":NEXT DE
        LAY
OL 3160 A$="":GOTO 3320
FI 3180 CLOSE #1:OPEN #1,4,0,"K:":? CHR$(125):
        CLR :DIM C$(12),D$(25),B$(1):C$="":D$=
        "":POKE 752,1:POKE 82,0:POKE 83,39
HG 3200 DIM R$(168),T$(40):T$=" ":T$(40)=T$:T$
        (2)=T$:R$=" ":R$(168)=R$:R$(2)=R$:F5=1
        :K=0:DIM SR$(2)
BN 3220 DIM A$(3300)
EF 3240 B=7:GOSUB 6380:GOTO 3320
LA 3260 ? CHR$(125):? "{UP}{9 SPACES}■CHOOSE G
        RAPHICS MODE■":? "{12 SPACES}CHOOSE 3 T
        HRU 8"
NO 3280 ? :GET #1,A:IF CHR$(A)<"3" OR CHR$(A)>
        "8" THEN 3260
EP 3300 B$=CHR$(A):B=VAL(B$)
HN 3320 IF B=3 THEN GRPX=39:GRPY=19:BJ=5
NL 3340 IF B=4 OR B=5 THEN GRPX=79:GRPY=39:BJ=
        5
BD 3360 IF B=6 OR B=7 THEN GRPX=159:GRPY=79:BJ
        =4
HB 3380 IF B=8 THEN GRPX=319:GRPY=159:GOSUB 53
        20:BJ=4
PE 3400 TRAP 3100:GRAPHICS B:TRAP 40000
LO 3420 POKE 82,0
DJ 3440 FM=0:IF B=8 THEN POKE 710,156:POKE 712
        ,156:POKE 709,0
DG 3460 REM **** MENU ****
IC 3480 SD=200:GOSUB 640:SD=250:GOSUB 640
PC 3500 POKE 752,1
CF 3520 ? CHR$(125):? "{UP}■C■=CIRCLE
        {3 SPACES}■E■=ELLIPSE ■L■=LINE":? "■K■
        =COMMENTS ";
JG 3540 ? "■G■=GR.MODE ■P■=PICT MENU":? "■Q■=Q
        UIT{5 SPACES}■COLOR■=";F5;"{3 SPACES}■
        GR.MODE■=";B
LC 3560 REM
BM 3580 C$="":POKE 752,1
```

```
MD 3600 GET #1,A:IF CHR$(A)="C" THEN D$="■CIRC
        LE■":GOTO 4120
MA 3620 IF CHR$(A)="E" THEN E=1:D$="■ELLIPSE■"
        :GOTO 4120
NA 3640 IF CHR$(A)="L" THEN D$="":K=0:GOTO 700
LD 3660 IF CHR$(A)="G" THEN 3260
MC 3680 IF CHR$(A)="P" THEN 3840
OD 3700 IF CHR$(A)="Q" THEN SAVE "D:GRAPLUS":G
        RAPHICS 0:CLR :END
PG 3720 IF CHR$(A)="0" THEN F5=0:GOTO 3460
PK 3740 IF CHR$(A)="1" THEN F5=1:GOTO 3460
PO 3760 IF CHR$(A)="2" THEN F5=2:GOTO 3460
AC 3780 IF CHR$(A)="3" THEN F5=3:GOTO 3460
LO 3800 IF CHR$(A)="K" THEN 5980
ND 3820 GOTO 3460
OD 3840 ? CHR$(125):? "{UP}{6 SPACES}■** PICTU
        RE OPTION MENU **■"
BL 3860 ? "{6 SPACES}■S■=SAVE{3 SPACES}■L■=LOA
        D{5 SPACES}■V■=VIEW":? "{6 SPACES}■E■=
        ERASE  ■R■=RETURN TO MENU":GET #1,A
MD 3880 IF CHR$(A)="S" THEN 1280
LG 3900 IF CHR$(A)="L" THEN 1920
LA 3920 IF CHR$(A)="E" THEN 2540
MA 3940 IF CHR$(A)="V" THEN 2240
MD 3960 IF CHR$(A)="R" THEN 3460
NM 3980 GOTO 3840
LE 4120 K=2:? CHR$(125):? "{UP}USE CURSOR TO P
        LACE X Y COORDINATES":? "PRESS ■S■ TO
        SET POSITION ■RETURN■=MENU":GOSUB 720:
        K=0
JK 4140 GOSUB 4980:IF E=1 THEN 4580
CL 4160 TRAP 4160:GOSUB 680:? "RADIUS=";:INPUT
         R:TRAP 40000:GOSUB 5180
AG 4180 TRAP 4180:AQ=11:GOSUB 660:? "{UP}STEP=
        ";:INPUT STP:TRAP 40000:IF STP=0 THEN
        STP=1
FG 4200 TRAP 4200:? "DEGREES START={4 SPACES}
        {4 LEFT}";:INPUT ST:TRAP 40000:GOSUB 5
        220
PC 4220 TRAP 4220:AQ=18:GOSUB 660:? "{UP}DEGRE
        ES STOP={4 SPACES}{4 LEFT}";:INPUT SP:
        TRAP 40000:GOSUB 5220
DC 4240 GOSUB 680:? "■RADIUS=";R;" ■STEP=";STP
        ;"{8 SPACES}CORRECT"
KJ 4260 ? "■DEG START=";ST;" ■DEG STOP=";SP;"
        (Y OR N)":GET #1,A:IF CHR$(A)="Y" THEN
         4300
MO 4280 GOTO 4120
JC 4300 FL=0:FOR N=ST TO SP STEP STP
JG 4320 SOUND 0,0,0,0
```

```
BC 4340 X1=X-R*COS(N/180*3.1416):Y1=Y+R*SIN(N/
        180*3.1416):COLOR F5
FP 4360 SOUND 0,100,10,12
FD 4380 IF X1<0 THEN X1=0+ABS(X1)
ED 4400 IF X1>GRPX THEN X1=GRPX-ABS(X1):IF X1<
        0 THEN X1=GRPX-ABS(X1)
FB 4420 IF Y1<0 THEN Y1=0+ABS(Y1)
FA 4440 IF Y1>GRPY THEN Y1=GRPY-ABS(Y1):IF Y1<
        0 THEN Y1=GRPY-ABS(Y1)
BP 4460 IF E=1 THEN RETURN
FJ 4480 IF FL=1 THEN 4520
IB 4500 PLOT X1,Y1:FL=1
NL 4520 DRAWTO X1,Y1
JK 4540 NEXT N:SOUND 0,0,0,0:FL=0
NF 4560 GOTO 3460
JK 4580 GOSUB 680:TRAP 4580:? "LENGTH=";:INPUT
         C:TRAP 40000
AK 4600 TRAP 4600:AQ=12:GOSUB 660:? "{UP}HEIGH
        T=";:INPUT D:TRAP 40000
AH 4620 TRAP 4620:AQ=23:GOSUB 660:? "{UP}STEP=
        ";:INPUT STP:TRAP 40000:IF STP=0 THEN
        STP=1
GG 4640 TRAP 4640:? "DEGREES START={4 SPACES}
        {4 LEFT}";:INPUT ST:TRAP 40000:GOSUB 5
        220
AD 4660 TRAP 4660:AQ=19:GOSUB 660:? "{UP}DEGRE
        ES STOP={4 SPACES}{4 LEFT}";:INPUT SP:
        TRAP 40000:GOSUB 5220
DJ 4680 GOSUB 680:? "■LENGTH=";C;" ■HEIGHT=";D
        ;" ■STEP=";STP;"CORRECT"
JD 4700 ? "■DEG START=";ST;" ■DEG STOP=";SP;"
        (Y OR N)"
KG 4720 GET #1,A:IF CHR$(A)="Y" THEN 4780
LG 4740 IF CHR$(A)="N" THEN 4120
NM 4760 GOTO 4680
KD 4780 DEG
JH 4800 FL=0:FOR N=ST TO SP STEP STP
JL 4820 SOUND 0,0,0,0
JA 4840 X1=X-((C/2)*COS(N)):Y1=Y-((D/2)*SIN(N)
        )
CB 4860 GOSUB 4380
GB 4880 IF FL=1 THEN 4920
IF 4900 PLOT X1,Y1:FL=1
NP 4920 DRAWTO X1,Y1
DM 4940 SOUND 0,80,10,10:NEXT N:SOUND 0,0,0,0:
        E=0:FL=0:RAD
NJ 4960 GOTO 3460
LE 4980 IF X=GRPX THEN X=X-1
KI 5000 IF Y=GRPY THEN Y=Y-1
BA 5020 IF X=0 THEN X=1
```

```
BE 5040 IF Y=0 THEN Y=1
LF 5060 COLOR F5:PLOT X-1,Y:DRAWTO X+1,Y:PLOT
        X,Y-1:DRAWTO X,Y+1
LB 5080 REM
00 5100 FOR S=1 TO 29:? "{RIGHT}";:NEXT S:? "
        {UP}■CORRECT■":FOR S=1 TO 29:? "
        {RIGHT}";:NEXT S:? "■(Y OR N)■":GET #1,
        A
AH 5120 IF CHR$(A)="Y" THEN COLOR F5:RETURN
CA 5140 IF CHR$(A)="N" THEN COLOR 0:PLOT X-1,Y
        :DRAWTO X+1,Y:PLOT X,Y-1:DRAWTO X,Y+1:
        POP :COLOR F5:GOTO 4120
MP 5160 ? "{2 UP}":GOTO 4980
OJ 5180 IF R>GRPY-1 THEN POP :GOTO 4160
KH 5200 RETURN
KD 5220 IF ST<0 OR ST>360 AND D$="■CIRCLE■" TH
        EN POP :? "{UP}";:GOTO 4200
IJ 5240 IF ST<0 OR ST>360 AND D$="■ELLIPSE■" T
        HEN POP :? "{UP}";:GOTO 4640
IN 5260 IF SP<0 OR SP>360 AND D$="■CIRCLE■" TH
        EN POP :GOTO 4220
HD 5280 IF SP<0 OR SP>360 AND D$="■ELLIPSE■" T
        HEN POP :GOTO 4660
KI 5300 RETURN
EK 5320 IF B2=1 THEN RETURN
NH 5340 RESTORE 5420:B2=1
OM 5360 FOR I=1 TO 168:READ A:R$(I,I)=CHR$(A):
        SOUND 0,A,10,6:NEXT I:SOUND 0,0,0,0
EE 5380 ML=ADR(R$)
KJ 5400 RETURN
IE 5420 DATA 104,201,4,240,9,170
OG 5440 DATA 240,5,104,104,202,208
CF 5460 DATA 251,96,104,133,215,104
FD 5480 DATA 133,214,104,104,168,104
EL 5500 DATA 133,217,104,133,216,104
BK 5520 DATA 104,240,236,133,212,24
CH 5540 DATA 165,214,101,88,133,214
CM 5560 DATA 165,89,101,215,133,215
CF 5580 DATA 152,240,15,165,214,105
CD 5600 DATA 64,133,214,165,215,105
00 5620 DATA 1,133,215,136,208,241
OB 5640 DATA 132,221,160,0,132,220
MF 5660 DATA 177,216,160,0,170,16
MB 5680 DATA 1,136,132,213,138,41
HA 5700 DATA 96,208,4,169,64,16
IP 5720 DATA 14,201,32,208,4,169
CK 5740 DATA 0,16,6,201,64,208
MN 5760 DATA 2,169,32,133,218,138
JF 5780 DATA 41,31,5,218,133,218
DA 5800 DATA 169,0,162,3,6,218
```

```
CE 5820 DATA 42,202,208,250,109,244
PL 5840 DATA 2,133,219,164,221,177
DC 5860 DATA 218,69,213,164,220,145
FD 5880 DATA 214,200,132,220,196,212
CI 5900 DATA 208,182,24,165,214,105
LI 5920 DATA 40,133,214,144,2,230
AD 5940 DATA 215,230,221,169,8,197
AP 5960 DATA 221,208,159,96,207,96
DA 5980 ? CHR$(125):? "{UP}■ENTER OR DELETE CO
        MMENT■":? "■E■=ENTER ■D■=DELETE ■R■=RE
        TURN TO MENU ":GET #1,A
DB 6000 IF B=8 THEN K=1:GOTO 6040
IC 6020 K=2
KK 6040 IF CHR$(A)="E" THEN 6120
KP 6060 IF CHR$(A)="D" THEN 6160
LP 6080 IF CHR$(A)="R" THEN 3460
NG 6100 GOTO 5980
LO 6120 ? "{UP}■ENTER COMMENT TO BE ADDED■":IN
        PUT T$:IF T$="" THEN 5980
MM 6140 GOTO 6200
GH 6160 ? "{UP}■ENTER COMMENT TO BE DELETED■":
        INPUT T$:IF T$="" THEN 5980
LM 6180 FOR I=1 TO LEN(T$):T$(I,I)=" ":NEXT I
CN 6200 ? CHR$(125):? "{UP}USE ARROWS TO POS C
        URSOR":? "PRESS S TO ENTER OR DELETE C
        OMMENT"
BC 6220 IF K=2 THEN 6280
BO 6240 GOSUB 720:COL=INT(X/8):ROW=INT(Y/8):U=
        USR(ML,COL,ROW,ADR(T$),LEN(T$))
MG 6260 K=0:GOTO 3460
OJ 6280 GOSUB 720
BL 6300 IF B=7 THEN COL=INT(X/8):ROW=Y
EK 6320 IF B=5 THEN COL=INT(X/16):ROW=Y
GJ 6340 FOR I=1 TO LEN(T$):R=ASC(T$(I,I)):V=US
        R(1536,R,F,COL+I,ROW):NEXT I
MH 6360 K=0:GOTO 3460
GE 6380 RESTORE 6400:MIL=1536:FOR I=0 TO 252:R
        EAD A:POKE MIL+I,A:SOUND 0,A,10,6:POKE
         708,A:NEXT I:SOUND 0,0,0,0:RETURN
KJ 6400 DATA 104,240,10,201,4,240
BE 6420 DATA 11,170,104,104,202,208
DK 6440 DATA 251,169,253,76,164,246
FE 6460 DATA 104,133,195,104,201,128
NF 6480 DATA 144,4,41,127,198,195
MD 6500 DATA 170,141,250,6,224,96
MO 6520 DATA 176,15,169,64,224,32
MI 6540 DATA 144,2,169,224,24,109
IM 6560 DATA 250,6,141,250,6,104
OM 6580 DATA 104,141,251,6,104,104
```

```
FL 6600 DATA 141,252,6,14,252,6
OL 6620 DATA 104,104,141,253,6,133
AK 6640 DATA 186,166,87,169,10,224
GD 6660 DATA 3,240,8,169,20,224
GC 6680 DATA 5,240,2,169,40,133
DG 6700 DATA 207,133,187,165,88,133
PJ 6720 DATA 203,165,89,133,204,32
GJ 6740 DATA 228,6,24,173,252,6
OH 6760 DATA 101,203,133,203,144,2
BP 6780 DATA 230,204,24,165,203,101
EJ 6800 DATA 212,133,203,165,204,101
PB 6820 DATA 213,133,204,173,250,6
AN 6840 DATA 133,187,169,8,133,186
MJ 6860 DATA 32,228,6,165,212,133
PC 6880 DATA 205,173,244,2,101,213
IM 6900 DATA 133,206,160,0,162,8
PK 6920 DATA 169,0,133,208,133,209
AL 6940 DATA 177,205,69,195,72,104
GH 6960 DATA 10,72,144,8,24,173
JL 6980 DATA 251,6,5,208,133,208
CE 7000 DATA 224,1,240,8,6,208
JL 7020 DATA 38,209,6,208,38,209
CB 7040 DATA 202,208,228,104,152,72
PA 7060 DATA 160,0,165,209,145,203
FB 7080 DATA 200,165,208,145,203,104
BP 7100 DATA 168,24,165,203,101,207
OC 7120 DATA 133,203,144,2,230,204
MN 7140 DATA 200,192,8,208,183,96
JB 7160 DATA 169,0,133,212,162,8
JA 7180 DATA 70,186,144,3,24,101
EM 7200 DATA 187,106,102,212,202,208
FH 7220 DATA 243,133,213,96,0,1
FB 7240 DATA 28
```

Program 2. Graphics Plus Demo

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
OH 60 GRAPHICS 0
AF 70 ? "ENTER X COORDINATE ";:INPUT X
AI 80 ? "ENTER Y COORDINATE ";:INPUT Y
HF 90 GRAPHICS 5:REM ** TRY OTHER GR.MODES **
LM 100 DIM A$(300):REM ** DIM A$ TO THE AMOUNT
       YOUR PICTURE USES **
PM 110 S=1:GOSUB 10300:REM ** USE THE LINE NUM
      BER FOR YOUR PICTURE HERE **
OD 120 FOR W=Y TO Y+U
OE 130 POSITION X,W
MN 140 ? #6;A$(S,S+T-1)
```

```
PI 150 S=S+T
CN 160 NEXT W
GP 170 END
FF 10300 T=17:U=16:B1=5:SU=5:SQ=3:REM ** B1=GR
         .MODE SU=# OF LINES USED FOR PICTURE
         SQ=PICTURE NUMBER **
BA 10301 IF TE=1 THEN RETURN :REM ** THIS LINE
          CAN BE DELETED **
HM 10302 A$(1)="DDDDDAAAAAAADDDDDDDDAAADDDDDAA
         ADDDDDAADDBBBBBDDAADDDAADBBDDDDDBBDAA
         DDADBBDCCCCCDBBDADAADBDCCDDDCCDBD"
IM 10303 A$(101)="AAADBDCCDBBBDCCDBDAADBDCDBBD
         BBDCDBDAADBDCDBDDDBDCDBDAADBDCDBBDBBD
         CDBDAADBDCCDBBBDCCDBDAAADBDCCD"
FJ 10304 A$(196)="DDCCDBDAADADBBDCCCCCDBBDADDA
         ADBBDDDDDBBDAADDDAADDBBBBBDDAADDDDDAA
         ADDDDDAAADDDDDDDDAAAAAAADDDDD"
NJ 10305 RETURN
```

Turtle Graphics in Atari BASIC

Prabhudeva Kavi

By adding just a short subroutine to your BASIC programs, you can simulate some turtle graphics commands.

Here is a BASIC subroutine (modified from Charles Brannon's "Turtle Pilot for the Atari," *COMPUTE!*, October 1982) that allows you to add turtle graphics to Atari BASIC. Turtle graphics will not affect the BASIC graphics statements in any way, and both turtle and BASIC graphics statements can be used at the same time.

Here are descriptions of how to use turtle graphics.

Turn the turtle:

TURN=X:GOSUB 31010 Turns the turtle clockwise X degrees if X is positive, and counterclockwise X degrees if X is negative.

Turnto:

TURTA=90−X Changes the angle of the turtle to the value of X. North is 0 degrees; south is 180 degrees; east and west are 90 and 270 degrees respectively.

Draw:

LINEL=X: GOSUB 31020 Moves turtle X units in the direction it is facing; leaves a trail.

Move the turtle:

LINEL=X: GOSUB 31060 Like Draw, but doesn't leave a trail.

Move to:

TURTX=X:TURTY=Y Moves turtle to X,Y location without leaving a trail. The range of X coordinates is from −79 to 80; for Y it is −48 to 47. The origin is at the center of the screen.

Programs 2 and 3 use the subroutines in Program 1. Type in Program 1 and save it. Add each of the other programs to Program 1.

Program 1. Turtle Subroutine

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
AM Ø PI=3.1415927/18Ø:TURTA=9Ø
GM 31ØØØ REM TURTLE GRAPHICS SUBROUTINES
KD 31Ø1Ø TURTA=TURTA-TURN:TURTA=TURTA-36Ø*(TUR
        TA>36Ø)+36Ø*(TURTA<Ø):RETURN :REM TUR
        N TURTLE
NH 31Ø2Ø TRAP 31Ø3Ø:PLOT TURTX+79,48-TURTY:REM
         LAST TURTLE LOCATION
PH 31Ø3Ø TURTX=TURTX+LINEL*COS(TURTA*PI):TURTY
        =TURTY+LINEL*SIN(TURTA*PI):REM FIGURE
         NEW COORDINATES
DA 31Ø4Ø TRAP 31Ø5Ø:DRAWTO TURTX+79,48-TURTY:T
        RAP 4ØØØØ:REM DRAW IT
NJ 31Ø5Ø RETURN
KE 31Ø6Ø TURTX=TURTX+LINEL*COS(TURTA*PI):TURTY
        =TURTY+LINEL*SIN(TURTA*PI):RETURN :RE
        M MOVE WITHOUT DRAWING
```

Program 2. Spiral

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
DE 2Ø GRAPHICS 8+16:SETCOLOR 2,Ø,Ø:COLOR 1:TUR
     TX=8Ø:TURTY=-48
LI 3Ø FOR I=1 TO 14Ø:LINEL=I:GOSUB 31Ø2Ø:TURN=
     91:GOSUB 31Ø1Ø:NEXT I
AB 4Ø GOTO 4Ø
```

Program 3. Snowflake

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
CI 2Ø GRAPHICS 7+16:COLOR 1:SETCOLOR Ø,Ø,14
BP 3Ø FOR I=1 TO 18:TURN=2Ø:GOSUB 31Ø1Ø:FOR J=
     1 TO 5:LINEL=35:GOSUB 31Ø2Ø:TURN=144:GOS
     UB 31Ø1Ø:NEXT J:NEXT I
AB 4Ø GOTO 4Ø
```

Chapter 4

Action Games

Guardian

Heath Lawrence

Can you save the scientists from the invading aliens in this fast-action shooting game? Requires one joystick.

The year is 1997. Alien starships have abducted most of Earth's key scientists for their own diabolical purposes. Those remaining have been taken to a powerful armed bunker for special protection. The base is shielded by an enormous defense field projected above the main compound. However, the aliens are not easily discouraged and launch deadly attacks against the force-field generators. Your mission as commander of the guardian bunker is to use the three ground-based turrets to destroy the incoming energy thieves. The fate of our scientists is in your hands.

Playing the Game

The enemy attack ships make continuous runs across the screen, getting lower with each complete pass. When they finally make a successful run below your energy field, the protective shield will drop momentarily. This period of time is just long enough for the alien mothership to kidnap one of your scientists with its transport beam.

The height of the force field is indicated by the two arrows at the sides of the playfield. When an enemy drone inflicts a power loss, these arrows will point outward showing that the shield is down. After a scientist has been taken, a successor will run to take his place. Five points are given for each ship that is destroyed. Unfortunately, your force shield rises when certain scores are reached (depending on your skill level). This allows the aliens to reach the shield much faster.

Your only weapons against the intruders are the three turrets that are stationed at the bottom of the screen. Each gun is fired by pushing the joystick to the side of the desired turret. Pushing the joystick forward will fire the center gun. The red trigger button is not used for firing. You can have only one shot on the screen at once, so all your shots should be very carefully timed.

Guardian

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
PD 1  T=1
LE 5  GOSUB 245:IF T THEN GOSUB 350
IA 10  GOSUB 285
FC 15  GOSUB 75
MH 25  REM CHECK STICK
ND 30  IF FLAG2 THEN 60
PC 40  S=STICK(0):IF S=11 THEN DIR=-LN+1:POS=SC
       R+LN*18+LN-36:FLAG=1:FLAG2=1:SHOT=70
NC 45  IF S=7 THEN DIR=-LN-1:POS=SCR+LN*18+LN-5
       :FLAG=1:FLAG2=1:SHOT=71
KD 50  IF S=14 THEN DIR=-LN:POS=SCR+LN*18+LN/2:
       FLAG=1:FLAG2=1:SHOT=86
HN 55  IF FLAG=1 THEN FOR X=1 TO 255 STEP 12:SO
       UND 1,X,10,8:NEXT X:SOUND 1,0,0,0
FC 60  IF FLAG THEN GOSUB 130
KD 65  REM UPDATE SCORE
EN 70  POSITION 2,1:? "SCORE:";VALUE:POSITION 1
       7,1:? "HI:";HI:POSITION 28,1:? "SAFE MEN
       :";GUYS
PC 72  IF VALUE=HVAL THEN HVAL=HVAL+INC:GOSUB 2
       02
JG 73  IF VALUE=BONUS THEN FOR X=1 TO 100 STEP
       5:SOUND 1,X,10,8:NEXT X:GUYS=GUYS+1:VALU
       E=VALUE+10:BONUS=BONUS+100
BC 74  SOUND 1,0,0,0:GOTO 15
IG 75  REM MOVE ATTACK SHIPS
MF 80  POKE LOC,BLA:LOC=LOC+VEC:P1=PEEK(LOC):IF
        P1=SHOT THEN 0=LOC:GOSUB 155:RETURN
HA 85  IF P1=BOU THEN N1=N1+2:LOC=SCR+LN*N1+LN-
       39
IJ 95  POKE LOC,EN1:IF N1>ALT THEN POKE LOC,BLA
       :N1=4:LOC=SCR+LN*N1+LN-39:GOSUB 210
PM 100  POKE MOT,BLANK:MOT=MOT+ZON:P2=PEEK(MOT)
        :IF P2=SHOT THEN 0=MOT:GOSUB 155:RETURN

HF 105  IF P2=BOU THEN N2=N2+2:MOT=SCR+LN*N2+LN
       -2
LC 115  POKE MOT,EN2:IF N2>ALT THEN POKE MOT,BL
       A:N2=4:MOT=SCR+LN*N2+LN-2:GOSUB 210
HD 120  RETURN
BN 125  REM UPDATE SHOT
EO 130  P3=PEEK(POS+DIR):IF P3=BOU THEN FLAG=0:
       FLAG2=0:POKE POS,BLA:RETURN
MH 135  IF P3=EN2 THEN 0=MOT:GOSUB 155:RETURN
LA 140  IF P3=EN1 THEN 0=LOC:GOSUB 155:RETURN
PK 145  POKE POS,BLA:POS=POS+DIR:POKE POS,SHOT:
       RETURN
```

```
HP 150 REM SUCCESSFUL HIT
AN 155 POKE POS,BLA:POKE O,EXP:FLAG=0:FLAG2=0:
       VALUE=VALUE+5
MF 160 FOR X=1 TO 10:SOUND 1,50,10,15:SOUND 1,
       0,0,0:NEXT X
BB 161 POKE O,BLA
AL 162 IF O=LOC THEN N1=4:LOC=SCR+LN*N1+LN-39:
       POKE LOC,EN1:RETURN
MP 170 N2=4:MOT=SCR+LN*N2+LN-2:POKE MOT,EN2:RE
       TURN
MN 175 REM DRAW SHIELD MARKERS
CO 180 POKE SCR+LN*SH+1,SHIE1:POKE SCR+LN*SH+3
       8,SHIE2:RETURN
HI 185 REM GAME OVER
HI 190 FOR J=1 TO 30:SOUND 0,64-J,10,10:SOUND
       1,F+J,10,10:FOR K=1 TO 30-J:NEXT K:SOUN
       D 0,0,0,0:SOUND 1,0,0,0
GO 192 FOR K=1 TO 10:NEXT K:NEXT J:POSITION 15
       ,10:? "GAME OVER":POSITION 37,1:? "0":I
       F VALUE>HI THEN HI=VALUE
BN 195 GUYS=T1:HVAL=T2:POSITION 7,11:? "PRESS
       START TO PLAY AGAIN":IF PEEK(53279)<>6
       THEN 195
CM 200 GOTO 10
HD 201 REM INCREASE SHIELD HEIGHT
AG 202 POKE SCR+LN*SH+1,BLA:POKE SCR+LN*SH+38,
       BLA:ALT=ALT-2:SH=SH-2:IF SH<5 THEN SH=S
       H+2:ALT=ALT+2
MJ 204 GOSUB 180:RETURN
CH 205 REM KIDNAP HUMANOID
FG 210 POKE SCR+LN*SH+1,SHIE2:POKE SCR+LN*SH+3
       8,SHIE1:FOR I=1 TO 10:SOUND 1,100,10,8:
       SOUND 1,0,0,0:NEXT I
KC 215 FOR I=1 TO 20:POKE SCR+LN-15+I*LN,BOU:S
       OUND 1,I,8,15:NEXT I:FOR I=20 TO 1 STEP
        -1:POKE SCR+LN-15+I*LN,BLA
AB 220 NEXT I:GUYS=GUYS-1:IF GUYS<1 THEN GOTO
       190
AC 225 SOUND 1,0,0,0:FOR I=35 TO 26 STEP -1:PO
       SITION I,20:? "h":SOUND 1,90,10,15:FOR
       D=1 TO 10:NEXT D
HC 230 POSITION I,20:? "i":SOUND 1,0,0,0:FOR D
       =1 TO 10:NEXT D:POSITION I,20:? " ":NEX
       T I:POSITION 25,20:? "i"
MN 235 GOSUB 180:RETURN
IF 240 REM DISPLAY TITLES
BM 245 POKE 106,PEEK(106)-5:GRAPHICS 0:POKE 75
       2,1:POSITION 12,1:? "{N}{F}{M}GUARDIAN
       {M}{G}{N}":POSITION 9,3
```

```
HL 246 ? "BY >> HEATH LAWRENCE":FOR X=1 TO 10:
       SOUND 0,95,10,15:FOR K=1 TO 5:NEXT K:SO
       UND 0,0,0,0:FOR K=1 TO 5:NEXT K
KK 247 NEXT X:POSITION 1,6:? "ELIMINATE ENERGY
       THIEVES":POSITION 1,8:? "PROTECT YOUR
       HUMANOIDS":SOUND 1,0,0,0
KJ 248 POSITION 1,10:? "SELECT SKILL LEVEL(1-4
       )>>>";:INPUT SKL:HVAL=SKL*10:INC=HVAL:G
       UYS=SKL+1
CG 252 T1=GUYS:T2=HVAL:FOR X=1 TO 10:SOUND 0,9
       5,10,15:FOR K=1 TO 5:NEXT K:SOUND 0,0,0
       ,0:FOR K=1 TO 5:NEXT K:NEXT X
DI 253 POSITION 9,13:? "<<< REDEFINING >>>":RE
       TURN
KG 280 REM INITIALIZATION
LN 285 GRAPHICS 0:POKE 756,ST/256:SETCOLOR 2,0
       ,0:POKE 752,1
OK 290 SCR=PEEK(88)+256*PEEK(89):ALT=16:SH=15:
       LN=40:BOU=128:BR1=103:BLA=0:EN1=101:EN2
       =102:FLAG2=0:EXP=83:SHOT=86
EG 295 LOC=SCR+LN*4+LN-39:MOT=SCR+LN*4+LN-3:N1
       =4:N2=4:VEC=1:ZON=-1:SHIE1=30:SHIE2=28:
       VALUE=0:FLAG=0:BONUS=100
FP 300 REM DRAW PLAYFIELD
GB 305 POSITION 19,19:? "ab{DOWN}{3 LEFT}gggg"
       :POSITION 3,19:? "c{DOWN}{LEFT}g":POSIT
       ION 36,19:? "d{DOWN}{LEFT}g":POSITION 1
       0,10:?
MI 315 FOR I=0 TO LN-1:POKE SCR+I,BOU:NEXT I:F
       OR I=0 TO LN-1:POKE SCR+LN*22+I,BOU:NEX
       T I
PE 320 FOR I=0 TO 22:POKE SCR+I*LN,BOU:NEXT I:
       FOR I=0 TO 22:POKE SCR+LN-1+I*LN,BOU:NE
       XT I
NJ 325 FOR I=1 TO LN-2:POKE SCR+LN*21+I,BR1:NE
       XT I:POSITION 25,20:? "i":GOSUB 175:RET
       URN
JJ 345 REM REDEFINE CHAR SET
EM 350 Z=776:T=0:ST=(PEEK(106)+1)*256:FOR X=0
       TO 1023:POKE ST+X,PEEK(57344+X):NEXT X
DF 370 FOR N=1 TO 9:FOR X=0 TO 7:READ A:POKE S
       T+Z+X,A:NEXT X:Z=Z+8:NEXT N:RETURN
IN 385 DATA 1,1,1,255,129,255,129,255
MH 390 DATA 128,128,128,255,129,255,129,255
JB 395 DATA 3,6,12,24,126,195,129,255
DD 400 DATA 192,96,48,28,126,195,129,255
AN 405 DATA 192,88,243,95,95,243,88,192
ND 410 DATA 60,100,207,26,26,207,100,60
BL 415 DATA 145,118,132,17,128,42,128,41
JN 420 DATA 24,219,126,60,24,24,60,102,24,24,2
       4,255,24,24,60,102
```

Mt. Rock

Steven M. Low

Quick reflexes are what you need to save the city from being covered by mud. Joystick required.

There's been a lot of rain lately at Mt. Rock. Now the mountain is sliding into the city. Each mud rock that falls on the city means less time to evacuate the residents. More time is needed and your role is clear. Flying above the city in your copter, you must fire at each mud ball and turn it into dust and hope that the high winds will prevent it from falling on the city.

Game Play

The game requires a 30-second initialization. Push the joystick trigger or computer START key to begin. Overlapping waste masses are indestructible. A collision with a mud ball, the mountain, city, upper borders, or side borders returns the copter to the landing pad. Points are accrued by turning a mud ball into dust (use the joystick button). The game ends when the city is completely buried. Restart the game by pressing the START key. Replay setup takes only five seconds.

Be sure to save a copy of the program before you try to run it.

Mt. Rock

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
GG 160  GOTO 510
HM 170  REM
OA 180  BURY=BURY+1:LINE=85-(BURY-1)
MO 190  POKE 87,7:PLOT 53,LINE:DRAWTO 159,LINE
JK 200  IF LINE<>69 THEN FALLEN=0:RETURN
OI 210  FOR I=0 TO 3:SOUND I,0,0,0:NEXT I
JI 220  GOTO 1910
HJ 230  REM
LI 240  JS=PEEK(632):YX=YX+STX(JS):YY=YY+STY(JS
        ):IF YX>208 THEN YX=208:REM Calculates
        players moves
KJ 250  X=USR(ADR(POK$),PLX,YX,PLY,YY)
KJ 260  IF PEEK(644)=1 THEN GOTO 350
ME 270  REM {DOWN} Player shoots {DOWN}
BO 280  X=USR(ADR(POK$),PMBASE+768+YY+4,3,PMBAS
        E+768+YY+5,3):FOR I=6 TO 20 STEP 2:POKE
         53252,YX-I:NEXT I
```

```
EK 290  X=USR(ADR(POK$),PMBASE+768+YY+4,0,PMBAS
        E+768+YY+5,0)
KF 300  REM {DOWN} Did you hit any nuke waste?
IE 310  POW=PEEK(53256):IF POW=2 THEN WX=55:WY=
        60:S=S+1:POKE 87,2:POSITION 17,1:? #6;S
        ;"   "
CO 320  IF POW=4 THEN W2X=73:W2Y=100:S=S+1:POKE
         87,2:POSITION 17,1:? #6;S;"   "
DP 330  IF POW=8 THEN W3X=90:W3Y=165:S=S+1:POKE
         87,2:POSITION 17,1:? #6;S;" "
CA 340  REM {DOWN} Calculate nuke moves {DOWN}
AK 350  IF WX<135 THEN WX=WX+4:WY=WY+4
HH 360  IF WX>134 THEN WY=WY+8:IF WY>209 THEN W
        X=55:WY=60:FALLEN=FALLEN+1
AF 370  IF W2X<125 THEN W2Y=W2Y-2:W2X=W2X+4
GE 380  IF W2X>124 THEN W2Y=W2Y+6:W2X=W2X+3:IF
        W2Y>209 THEN W2X=73:W2Y=100:FALLEN=FALL
        EN+1
AK 390  IF W3X<122 THEN W3X=W3X+2:W3Y=W3Y-5
GI 400  IF W3X>121 THEN W3X=W3X+1:W3Y=W3Y+4:IF
        W3Y>209 THEN W3X=90:W3Y=165:FALLEN=FALL
        EN+1
MK 410  IF FALLEN>=1 THEN GOSUB 180
IG 420  REM {DOWN} Coll. Reg. keep plyr in
        {DOWN}{13 SPACES}bounds
MD 430  IF PEEK(53260)<>0 THEN YX=164:YY=174
MF 440  IF PEEK(53252)<>0 THEN YX=164:YY=174
NE 450  REM {DOWN} Actual ML moves players
        {DOWN}
JA 460  X=USR(ADR(POK$),PLX+1,WX,PLY+1,WY,PLX+2
        ,W2X,PLY+2,W2Y,PLX+3,W3X,PLY+3,W3Y)
HP 470  SOUND 1,WY,2,8:SOUND 2,W2Y,2,8:SOUND 3,
        W3Y,2,8
BE 480  POKE 53278,0:REM HITCLR
GM 490  GOTO 240
KB 500  REM INITIALIZATION
FH 510  RESTART=0:DIM POK$(25),LEDR$(2),STX(15)
        ,STY(15):GOSUB 1850:GOSUB 1780:GOSUB 21
        70
HL 520  REM
LC 530  GRAPHICS 7+16
OG 540  SETCOLOR 0,1,9:SETCOLOR 1,0,4:SETCOLOR
        4,3,2:SETCOLOR 2,12,6
EH 550  POKE 559,0:GOSUB 1310
GL 560  REM BACKGROUND DRAWING
DO 570  REM MOUNTAINS
PM 580  COLOR 1:PLOT 7,1:DRAWTO 7,4
ED 590  COLOR 2:PLOT 0,5:DRAWTO 10,5:DRAWTO 10,
        12:COLOR 1:PLOT 0,12:DRAWTO 15,12
GD 600  DRAWTO 20,23:DRAWTO 22,23:DRAWTO 27,27
```

```
JI 610 COLOR 2:PLOT 0,27:DRAWTO 28,27:DRAWTO 2
       8,34
IJ 620 COLOR 1:PLOT 0,35:DRAWTO 32,35
EL 630 DRAWTO 34,41:DRAWTO 36,46:DRAWTO 40,50:
       DRAWTO 43,58:DRAWTO 0,58
NC 640 COLOR 2:PLOT 44,59:DRAWTO 44,67
JK 650 COLOR 1:PLOT 0,68:DRAWTO 52,68
EB 660 DRAWTO 52,85:DRAWTO 0,85
IB 670 REM
MI 680 COLOR 1:PLOT 0,1:POKE 765,1:POSITION 0,
       4:XIO 18,#6,0,0,"S:"
PN 690 COLOR 2:PLOT 0,5:POKE 765,2:POSITION 0,
       11:XIO 18,#6,0,0,"S:"
CH 700 COLOR 1:PLOT 0,12:POKE 765,1:POSITION 0
       ,26:XIO 18,#6,0,0,"S:"
CP 710 COLOR 2:PLOT 0,27:POKE 765,2:POSITION 0
       ,34:XIO 18,#6,0,0,"S:"
DK 720 COLOR 2:PLOT 0,58:POKE 765,2:POSITION 0
       ,67:XIO 18,#6,0,0,"S:"
DE 730 COLOR 1:PLOT 0,35:POKE 765,1:POSITION 0
       ,58:XIO 18,#6,0,0,"S:"
FB 740 PLOT 0,59:POKE 765,1:POSITION 0,85:XIO
       18,#6,0,0,"S:"
OM 750 COLOR 2:DRAWTO 0,1:DRAWTO 159,1
OP 760 REM This is the city
IC 770 REM
LP 780 COLOR 3:FOR X=60 TO 66 STEP 2:FOR Y=75
       TO 85:PLOT X,Y:NEXT Y:NEXT X:REM Artifa
       cting color
PN 790 COLOR 3:FOR X=70 TO 74:FOR Y=73 TO 85:P
       LOT X,Y:NEXT Y:NEXT X:COLOR 1:PLOT 71,7
       5:PLOT 72,80:PLOT 73,77
KD 800 COLOR 1:FOR X=78 TO 84:FOR Y=70 TO 85:P
       LOT X,Y:NEXT Y:NEXT X
KL 810 COLOR 0:PLOT 79,71:PLOT 80,80:PLOT 81,7
       5:PLOT 82,77:PLOT 83,73
BM 820 COLOR 3:FOR X=89 TO 95 STEP 2:FOR Y=73
       TO 85:PLOT X,Y:NEXT Y:NEXT X
BE 830 COLOR 1:FOR X=90 TO 96 STEP 2:FOR Y=73
       TO 85:PLOT X,Y:NEXT Y:NEXT X
IE 840 COLOR 2:FOR X=90 TO 96 STEP 2:FOR Y=74
       TO 84 STEP 2:PLOT X,Y:NEXT Y:NEXT X
PD 850 COLOR 3:FOR X=100 TO 110:FOR Y=80 TO 85
       :PLOT X,Y:NEXT Y:NEXT X
JM 860 X=1:Y=80
GP 870 X=X+1:Y=Y-1
ON 880 PLOT 100+X,Y:DRAWTO 110-X,Y:IF Y=77 THE
       N GOTO 900
HJ 890 GOTO 870
```

```
EC 900  COLOR 1:PLOT 101,81:PLOT 103,84:PLOT 10
        3,82:PLOT 104,82:PLOT 106,84:FOR X=107
        TO 109:PLOT X,84:NEXT X
AK 910  COLOR 2:FOR X=105 TO 108:PLOT X,81:NEXT
         X
BK 920  COLOR 2:PLOT 115,85:DRAWTO 115,69:PLOT
        125,69:DRAWTO 125,85
DF 930  COLOR 1:PLOT 115,69:DRAWTO 125,69
DJ 940  PLOT 120,70:DRAWTO 120,85
BN 950  COLOR 2:PLOT 115,85:DRAWTO 125,69:PLOT
        125,85:DRAWTO 115,69
AG 960  COLOR 3:FOR X=127 TO 131:FOR Y=75 TO 77
        :PLOT X,Y:NEXT Y:NEXT X
DK 970  COLOR 2:PLOT 127,85:DRAWTO 127,75
BL 980  COLOR 1:FOR X=128 TO 129:PLOT X,75:NEXT
         X
HJ 990  COLOR 1:FOR X=135 TO 159 STEP 2:FOR Y=8
        0 TO 85:PLOT X,Y:NEXT Y:NEXT X
CD 1000  COLOR 0:PLOT 137,81:PLOT 139,84:PLOT 1
         45,82:FOR X=150 TO 153:FOR Y=83 TO 85:
         PLOT X,Y:NEXT Y:NEXT X
MF 1010  COLOR 1:PLOT 136,83:PLOT 144,83
DH 1020  COLOR 2:FOR X=146 TO 149:FOR Y=77 TO 7
         9:PLOT X,Y:NEXT Y:NEXT X
HJ 1030  PLOT 147,77:DRAWTO 159,73
NN 1040  IF RESTART=1 THEN 1060
JB 1050  GOSUB 1450:GOSUB 1150
BF 1060  POKE 559,62:POKE 53277,3:REM Enable P/
         M Gr.
JC 1070  POKE 87,2:POSITION 0,1:? #6;"[INV] ROCK":
         POKE 87,2:POSITION 11,1:? #6;"SCORE
         {4 SPACES}"
GG 1080  FOR VOL=15 TO 0 STEP -0.1:SOUND 0,10,1
         2,VOL:NEXT VOL:REM Ding!
BB 1090  IF PEEK(53279)=6 THEN 240
JH 1100  IF PEEK(644)=0 THEN 240
MG 1110  GOTO 1090
JL 1120  END
KJ 1130  REM
CK 1140  REM CUSTOMIZED CHARACTER SET
EH 1150  CS=256*(PM-2):POKE 756,CS/256
EN 1160  FOR I=128 TO 207:POKE CS+I,PEEK(57344+
         I):NEXT I:REM .{3 SPACES}ROM NUMBER SE
         T
HD 1170  RESTORE 1180:FOR I=1 TO 10:READ NC:FOR
          J=0 TO 7:READ CC:POKE CS+((NC*8)+J),C
         C:NEXT J:NEXT I
IN 1180  DATA 45,130,198,238,254,254,238,238,23
         8
OI 1190  DATA 51,126,192,192,62,2,254,252,248
```

```
BG 1200 DATA 35,62,126,192,192,192,254,126,62
CM 1210 DATA 47,124,124,230,230,230,230,126,62
HC 1220 DATA 50,252,196,252,252,254,206,206,20
        6
HP 1230 DATA 37,254,224,248,224,254,254,254,25
        4
GA 1240 DATA 52,254,254,56,56,56,56,56,56
HI 1250 DATA 46,198,230,246,254,222,206,206,20
        6
JL 1260 DATA 53,198,198,198,198,254,254,254,12
        4
HC 1270 DATA 43,198,204,216,240,248,222,222,22
        2
KL 1280 RETURN
HK 1290 REM ALTER DISPLAY LIST
AM 1300 REM GR. 2 ABOVE GR. 7
KH 1310 DL=PEEK(560)+PEEK(561)*256
HH 1320 POKE DL+3,71
KL 1330 SCR=PEEK(88)+256*PEEK(89)
GE 1340 SCR=SCR-20
OK 1350 REM Line 1370 re-aligns the{15 SPACES}G
        R. 7 screen to left edge{13 SPACES}of s
        creen by altering
IH 1360 REM Display List.{25 SPACES}Ever see th
        is before ?
BM 1370 X=USR(ADR(POK$),88,SCR):REM The
        {7 SPACES}ML program handles HI byte L
        OW{8 SPACES}byte problem
FK 1380 POKE DL+94,65:POKE DL+95,PEEK(560):POK
        E DL+96,PEEK(561)
KN 1390 RETURN
JM 1400 END
PH 1410 REM INITIALIZE VBLANK PM
NG 1450 IF PEEK(1696)=104 THEN RESTORE 1750:GO
        TO 1470
BJ 1460 RESTORE 1640:FOR I=1536 TO 1706:READ A
        :POKE I,A:NEXT I
FG 1470 FOR I=1774 TO 1787:POKE I,0:NEXT I
BG 1480 PM=PEEK(106)-32:POKE 54279,PM:PMBASE=2
        56*PM
FB 1490 FOR I=PMBASE+1023 TO PMBASE+2047:POKE
        I,0:NEXT I:FOR I=PMBASE+768 TO PMBASE+
        1022:POKE I,0:NEXT I
IL 1500 FOR I=0 TO 7:READ A:POKE PMBASE+1025+I
        ,A:NEXT I
JA 1510 FOR I=0 TO 7:READ A:POKE PMBASE+1281+I
        ,A:NEXT I
MB 1520 RESTORE 1760:FOR I=0 TO 7:READ A:POKE
        PMBASE+1537+I,A:NEXT I
```

```
MG 1530 RESTORE 1760:FOR I=0 TO 7:READ A:POKE
        PMBASE+1793+I,A:NEXT I
HG 1540 POKE 704,16*10+6:POKE 705,16*3+8:POKE
        706,16*2+10:POKE 707,16*14+6
HL 1550 PLX=53248:PLY=1780:PLL=1784:POKE 623,4
        :POKE 1788,PM+4:POKE 54279,PM
FK 1560 X=USR(1696):REM > VBlank PM <
        {9 SPACES}This only needs executed onc
        e
JM 1570 YX=164:YY=174:WX=55:WY=60:W2X=73:W2Y=1
        00:W3X=90:W3Y=165
HP 1580 POKE PLL,8:POKE PLL+1,8:POKE PLL+2,8:P
        OKE PLL+3,8
MF 1590 POKE PLX,YX:POKE PLY,YY:POKE PLX+1,WX:
        POKE PLY+1,WY:POKE PLX+2,W2X:POKE PLY+
        2,W2Y
CB 1600 POKE PLX+3,W3X:POKE PLY+3,W3Y
KI 1610 RETURN
FO 1620 REM VBLANK INTERRUPT ROUTINE
GC 1630 REM For moving P/M Graphics
HH 1640 DATA 162,3,189,244,6,240,89,56,221,240
        ,6,240,83,141,254,6,106,141
DO 1650 DATA 255,6,142,253,6,24,169,0,109,253,
        6,24,109,252,6,133,204,133
EK 1660 DATA 206,189,240,6,133,203,173,254,6,1
        33,205,189,248,6,170,232,46,255
EL 1670 DATA 6,144,16,168,177,203,145,205,169,
        0,145,203,136,202,208,244,76,87
PM 1680 DATA 6,160,0,177,203,145,205,169,0,145
        ,203,200,202,208,244,174,253,6
LE 1690 DATA 173,254,6,157,240,6,189,236,6,240
        ,48,133,203,24,138,141,253,6
ND 1700 DATA 109,235,6,133,204,24,173,253,6,10
        9,252,6,133,206,189,240,6,133
GK 1710 DATA 205,189,248,6,170,160,0,177,203,1
        45,205,200,202,208,248,174,253,6
CF 1720 DATA 169,0,157,236,6,202,48,3,76,2,6,7
        6,98,228,0,0,104,169
ON 1730 DATA 7,162,6,160,0,32,92,228,96
DA 1740 REM {DOWN} P/M Data {DOWN}
HP 1750 DATA 252,32,35,255,248,240,240,224
IG 1760 DATA 0,24,124,62,62,126,59,24
KN 1770 REM {DOWN} Initialize stick values
        {DOWN}
MB 1780 STX(5)=6:STX(6)=6:STX(7)=8:STX(9)=-6:S
        TX(10)=-6:STX(11)=-8:STX(13)=0:STX(14)
        =0:STX(15)=0
ML 1790 STY(5)=6:STY(6)=-6:STY(7)=0:STY(9)=6:S
        TY(10)=-6:STY(11)=0:STY(13)=8:STY(14)=
        -8:STY(15)=0
```

```
KJ 18ØØ RETURN
KO 1810 REM
JD 1850 RESTORE 1860:FOR W=1 TO 25:READ P:POK$
        (W,W)=CHR$(P):NEXT W:RETURN
EE 1860 DATA 104,74,170,160,0,104,133,255
HA 1870 DATA 104,133,254,104,240,4,200,145
ID 1880 DATA 254,136,104,145,254,202,208,237,9
        6
AM 1890 REM GAME OVER
PB 1900 REM *** G ***
AO 1910 COLOR 1:PLOT 73,73:DRAWTO 65,73:PLOT 6
        5,74:DRAWTO 65,81:PLOT 66,74:DRAWTO 66
        ,81
MC 1920 PLOT 65,82:DRAWTO 74,82:DRAWTO 74,79:P
        LOT 73,82:DRAWTO 73,79:DRAWTO 70,79
OO 1930 REM *** A ***
DI 1940 PLOT 76,82:DRAWTO 79,73:PLOT 80,73:DRA
        WTO 84,82:PLOT 77,78:DRAWTO 83,78
PM 1950 REM *** M ***
GL 1960 PLOT 86,82:DRAWTO 86,73:PLOT 87,82:DRA
        WTO 87,73:DRAWTO 91,77:DRAWTO 95,73:DR
        AWTO 95,82:PLOT 96,73:DRAWTO 96,82
PG 1970 REM *** E ***
KM 1980 PLOT 98,82:DRAWTO 98,73:PLOT 99,82:DRA
        WTO 99,73:DRAWTO 106,73:PLOT 100,77:DR
        AWTO 104,77:PLOT 100,82:DRAWTO 106,82
AC 1990 REM *** O ***
AM 2000 PLOT 115,82:DRAWTO 115,73:DRAWTO 123,7
        3:DRAWTO 123,82:DRAWTO 115,82
KF 2010 REM ** V **
KA 2020 PLOT 126,73:DRAWTO 130,82:DRAWTO 134,7
        3
JG 2030 REM ** E **
JA 2040 PLOT 136,82:DRAWTO 136,73:PLOT 137,82:
        DRAWTO 137,73:DRAWTO 144,73
FN 2050 PLOT 138,77:DRAWTO 142,77:PLOT 138,82:
        DRAWTO 144,82
KG 2060 REM ** R **
MH 2070 PLOT 147,82:DRAWTO 147,73:PLOT 148,82:
        DRAWTO 148,73:DRAWTO 155,73:DRAWTO 155
        ,78:PLOT 154,73:DRAWTO 154,78
JP 2080 DRAWTO 148,78:PLOT 150,78:DRAWTO 154,8
        2:PLOT 151,78:DRAWTO 155,82
IE 2090 IF PEEK(53279)<>6 THEN 2090
AD 2100 POKE 559,0:RESTART=1:BURY=0:FALLEN=0:S
        =0:YX=164:YY=174:WX=55:WY=60:W2X=73:W2
        Y=100:W3X=90:W3Y=165
LK 2110 POKE PLX,YX:POKE PLY,YY:POKE PLX+1,WX:
        POKE PLY+1,WY:POKE PLX+2,W2X:POKE PLY+
        2,W2Y
```

```
BP 2120 POKE PLX+3,W3X:POKE PLY+3,W3Y
FB 2130 COLOR 4:POKE 87,7:FOR Y=69 TO 85:PLOT
        53,Y:DRAWTO 159,Y:NEXT Y
JN 2140 GOTO 760
JP 2150 END
EP 2160 REM INTRODUCTION
GG 2170 GRAPHICS 1+16:SETCOLOR 0,1,9:SETCOLOR
        1,3,2:SETCOLOR 2,12,6:SETCOLOR 4,3,2
OG 2180 BEGIN=PEEK(560)+PEEK(561)*256+4:SCROLL
        =PEEK(88)+256*PEEK(89)
LN 2190 RESTORE 2330:X=6
BM 2200 X=X+1:IF X=9 THEN X=X+1
GC 2210 IF LEDR$="K" THEN 2240
AH 2220 READ LEDR$:POSITION X,0:? #6;LEDR$
PF 2230 FOR Y=1 TO 5:POSITION X,Y-1:? #6;" ":P
        OSITION X,Y:? #6;LEDR$:NEXT Y:GOSUB 23
        10:GOTO 2200
KB 2240 FOR X=1 TO 8:POSITION X,9:? #6;" B":NE
        XT X:GOSUB 2320
LF 2250 FOR X=19 TO 10 STEP -1:POSITION X,9:?
        #6;"Y ":NEXT X:GOSUB 2320
HF 2260 FOR Y=22 TO 11 STEP -1:POSITION 6,Y+1:
        ? #6;"{5 SPACES}":POSITION 6,Y:? #6;"S
        TEVE":NEXT Y
BB 2270 GOSUB 2310
DM 2280 FOR Y=22 TO 11 STEP -1:POSITION 12,Y+1
        :? #6;"{3 SPACES}":POSITION 12,Y:? #6;
        "LOW":NEXT Y
BD 2290 GOSUB 2310
MH 2300 GOTO 2340
DB 2310 FOR I=1 TO 5:SOUND 0,125,14,6:FOR D=1
        TO 5:NEXT D:NEXT I:SOUND 0,0,0,0:RETUR
        N
CL 2320 FOR I=1 TO 5:SOUND 0,100,14,6:FOR D=1
        TO 5:NEXT D:NEXT I:SOUND 0,0,0,0:RETUR
        N
IO 2330 DATA M,T,R,O,C,K
FE 2340 FOR D=1 TO 200:NEXT D
BB 2350 REM {DOWN} Scroll intro to left {DOWN}
IJ 2360 X=0:FOR I=1 TO 19:ML=USR(ADR(POK$),BEG
        IN,SCROLL+I):X=X+1
PE 2370 FOR Y=5 TO 11:POSITION X,Y:? #6;" ":NE
        XT Y:NEXT I
BI 2380 POSITION 3,7:? #6;"PLEASE WAIT"
JO 2390 POSITION 3,9:? #6;"30 SECONDS"
FH 2400 FOR D=1 TO 350:NEXT D
KH 2410 RETURN
JP 2420 END
```

Climb

David A. Miller

For two players with joysticks, "Climb" is a different type of race game—who has faster reflexes?

The object of this two-player game is simply to be the first to reach the flag at the top of the hill. This should be easy to accomplish, but as with most computer games, there is a catch.

After the program has finished initializing, you'll see the playfield which contains two hills and two jumpers, one for each player. At regular intervals an arrow will appear in the middle of the screen near the top. This arrow points in the direction in which both players must push their joysticks. The first player to push the joystick will see his or her jumper leap into the air. The arrow will disappear, only to be quickly followed by another arrow indicating in which direction the joystick must be pushed to complete the jump. After a short delay, the middle arrow will again appear, and both players race to be first to push their joysticks in the proper direction. This continues until one of the jumpers reaches the top.

Now, there's just one more thing I should mention (I told you there would be a catch). Just when you think you're making progress up the hill, you'll find yourself being knocked back down one jump, and ten points will be subtracted from your score. Why does this happen? Well, it seems that every time your opponent makes a successful jump, you get knocked backward.

Climb

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
CC 9 REM *** POKE NEW CHSET INTO MEMORY AND TU
     RN ANTIC OFF ***
JN 10 GOSUB 620:GRAPHICS 18:POKE 756,CHSET/256
     :POKE 559,0:GOSUB 400
PN 20 POKE 709,196
FE 22 REM *** ONE LINE OF GR MODE 1
FF 30 DL=PEEK(560)+256*PEEK(561)+4:POKE DL+12,
     6:POSITION 1,10:? #6;"SCORE":POSITION 14
     ,10:? #6;"score"
ML 40 POKE 559,34:REM *** TURN ANTIC BACK ON
CP 42 REM *** PRINT SCORE AND ARROW
OH 50 SPOS1=169:SPOS2=170:SCORE1=0:SCORE2=0
```

```
OP 60  SOUND Ø,Ø,Ø,Ø:SOUND 1,Ø,Ø,Ø:POSITION 2,1
       1:? #6;"{4 SPACES}":POSITION 15,11:? #6;
       "{4 SPACES}"
AF 7Ø  POSITION 2,11:? #6;SCORE1:POSITION 15,11
       :? #6;SCORE2:GOSUB 34Ø
BJ 8Ø  FOR W=1 TO 2ØØ:NEXT W:POKE SCR+3Ø,RAND:P
       OKE 53279,1
MF 82  REM *** CHECK IF SOMEONE MATCHED
BD 9Ø  FOR W=1 TO 9Ø:S1=STICK(Ø):S2=STICK(1):IF
        S1=ST OR S2=ST THEN 11Ø
CH 1ØØ NEXT W
JD 11Ø POKE SCR+3Ø,Ø
DI 112 REM *** CHECK WHO MATCHED
PO 12Ø IF S1=S2 THEN 7Ø
PA 13Ø IF S1=ST THEN TPOS1=SPOS1:SPOS1=SPOS1-2
       Ø:POKE SCR+SPOS1,199:POKE SCR+TPOS1,Ø:G
       OTO 16Ø
PH 14Ø IF S2=ST THEN TPOS2=SPOS2:SPOS2=SPOS2-2
       Ø:POKE SCR+SPOS2,198:POKE SCR+TPOS2,Ø:G
       OTO 25Ø
DG 15Ø GOTO 7Ø
OO 152 REM *** PLAYER1 MATCHED
EL 16Ø SOUND Ø,1ØØ,1Ø,8:GOSUB 34Ø:POKE SCR+45,
       RAND+128
FI 17Ø FOR W=1 TO 75:S1=STICK(Ø):IF S1=ST THEN
        19Ø:NEXT W:POKE SCR+45,Ø
CP 18Ø NEXT W
NE 19Ø POKE SCR+45,Ø:IF S1<>ST THEN 24Ø
FE 2ØØ TPOS1=SPOS1:SPOS1=SPOS1-1:POKE SCR+SPOS
       1,196:POKE SCR+TPOS1,Ø:IF SPOS1=1 THEN
       WAVE=SPOS1:GOTO 5ØØ
JP 21Ø SCORE1=SCORE1+2Ø:IF SPOS2>=17Ø THEN 6Ø
LK 22Ø TPOS2=SPOS2:SPOS2=SPOS2+19
KF 23Ø POKE SCR+SPOS2,197:POKE SCR+TPOS2,Ø:SCO
       RE2=SCORE2-1Ø:GOTO 6Ø
KG 24Ø POKE SCR+SPOS1,Ø:POKE SCR+TPOS1,196:SPO
       S1=TPOS1:GOTO 6Ø
OP 242 REM *** PLAYER 2 MATCHED
BL 25Ø SOUND Ø,2ØØ,1Ø,8:GOSUB 34Ø:POKE SCR+54,
       RAND+64
FL 26Ø FOR W=1 TO 75:S2=STICK(1):IF S2=ST THEN
        28Ø:NEXT W:POKE SCR+54,Ø
CP 27Ø NEXT W
NF 28Ø POKE SCR+54,Ø:IF S2<>ST THEN 33Ø
KC 29Ø TPOS2=SPOS2:SPOS2=SPOS2+1:POKE SCR+SPOS
       2,197:POKE SCR+TPOS2,Ø:IF SPOS2=18 THEN
        WAVE=SPOS2:GOTO 56Ø
KI 3ØØ SCORE2=SCORE2+2Ø:IF SPOS1>=169 THEN 6Ø
KP 31Ø TPOS1=SPOS1:SPOS1=SPOS1+21
```

```
KA 320 POKE SCR+SPOS1,196:POKE SCR+TPOS1,0:SCO
       RE1=SCORE1-10:GOTO 60
KL 330 POKE SCR+SPOS2,0:POKE SCR+TPOS2,197:SPO
       S2=TPOS2:GOTO 60
DK 332 REM *** RANDOM ARROW DIRECTIONS
MF 340 RAND=INT(RND(0)*4)+8:IF RAND=TRAND THEN
        340
DI 350 IF RAND=8 THEN ST=13
DL 360 IF RAND=9 THEN ST=14
DG 370 IF RAND=10 THEN ST=7
GD 380 IF RAND=11 THEN ST=11
JB 390 TRAND=RAND:RETURN
FF 392 REM *** CREATE PLAYER1'S HILL
JH 400 SCR=PEEK(88)+256*PEEK(89):POKE SCR,129:
       C=1:B=20
FD 410 FOR A=0 TO C:POKE SCR+B+A,3:NEXT A
OG 420 B=B+20:C=C+1:IF B>180 THEN 440
GF 430 GOTO 410
FB 432 REM *** CREATE PLAYER2'S HILL
OL 440 B=39:C=1:POKE SCR+19,65
FJ 450 FOR A=0 TO C:POKE SCR+B-A,3:NEXT A
PI 460 B=B+20:C=C+1:IF B>199 THEN 480
GN 470 GOTO 450
CL 480 POKE SCR+169,196:POKE SCR+170,197
MD 490 FOR A=200 TO 219:POKE SCR+A,3:NEXT A:RE
       TURN
IH 492 REM *** END OF GAME
DE 500 A=PEEK(710):FOR W=1 TO 150:B=PEEK(53770
       ):POKE 710,B:SOUND 0,B,10,6
LJ 510 NEXT W:POKE 710,A:SOUND 0,0,0,0:SOUND 1
       ,0,0,0:GOTO 580
EL 560 A=PEEK(709):FOR W=1 TO 150:B=PEEK(53770
       ):POKE 709,B:SOUND 1,B,10,6
LH 570 NEXT W:POKE 709,A:SOUND 1,0,0,0:SOUND 0
       ,0,0,0
EM 580 POSITION 7,3:? #6;"press":POSITION 8,4:
       ? #6;"start"
JB 590 POKE SCR+WAVE,204:X=COS(5):IF PEEK(5327
       9)=6 THEN 610
JN 600 POKE SCR+WAVE,205:FOR W=1 TO 50:NEXT W:
       GOTO 590
IM 610 RUN
DN 612 REM *** TITLE SCREEN
JM 620 GRAPHICS 17:DL=PEEK(560)+256*PEEK(561)+
       4:POKE DL+10,5:POKE DL+15,4
BM 630 POKE 711,132:POKE 710,136:POKE 709,32:P
       OKE 708,40
HA 640 POSITION 15,9:? #6;"C L I M B":POSITION
        12,15:? #6;"please stand by"
```

```
LJ 642 REM *** REDEFINE CHARACTER SET
CN 650 CHSET=(PEEK(106)-8)*256:IF PEEK(CHSET+1
       *8)=112 THEN RETURN
JM 660 FOR I=0 TO 512:POKE CHSET+I,PEEK(57344+
       I):NEXT I:RESTORE 700
KO 670 READ A:IF A=-1 THEN RETURN
AC 680 FOR J=0 TO 7:READ B:POKE CHSET+A*8+J,B:
       NEXT J
HF 690 GOTO 670
GF 692 REM *** DATA FOR NEW CHSET
IA 700 DATA 1,112,120,124,126,127,96,96,96
CL 710 DATA 3,255,195,129,129,129,129,195,255
AN 720 DATA 4,56,108,60,24,120,24,40,108
HM 730 DATA 5,28,54,60,24,30,24,20,54
IN 740 DATA 6,28,54,60,26,28,56,40,80
LP 750 DATA 7,56,108,60,88,56,28,20,10
OI 760 DATA 8,24,24,24,24,255,126,60,24
OK 770 DATA 9,24,60,126,255,24,24,24,24
LC 780 DATA 10,8,12,14,255,255,14,12,8
IC 790 DATA 11,16,48,112,255,255,112,48,16
HB 791 DATA 12,60,102,60,24,126,153,36,102
HE 792 DATA 13,60,102,61,25,126,152,36,102
BA 800 DATA -1
```

Front Attack

Robert J. Neep

"Front Attack" features vivid scrolling graphics and fast action. You need a joystick, at least 16K RAM, and quick reflexes to play this one.

As you're exploring a newly discovered planet, alien ships begin coming at you. You're not sure if they're friend or foe. But to be safe, you should try to avoid them. Soon, though, you discover that *friendly* is not a word in their vocabulary—it's you or them.

Attack

Once the title screen appears, press the START key to begin the attack. As the aliens attack, you have two options: avoid them by moving your joystick up and down or fire at them by pressing the joystick button. You'll soon discover that your ship is stronger than you had expected. It can withstand three direct hits before the game ends and your score is displayed. The score is based on the number of aliens you manage to down.

Front Attack

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
FQ 7 POKE 623,1
GM 10 GOSUB 1000:LIFE=3
DP 20 DIM P$(1024),D$(50),SHIP$(30),LAZER$(30)
      ,ENE$(30),FM$(30)
FJ 30 VT=PEEK(134)+PEEK(135)*256
DP 40 AT=PEEK(140)+PEEK(141)*256
NC 50 POKE 559,62:RX1=1:RX2=1
BM 60 PM=PEEK(106)-32:POKE 54279,PM
JI 70 POKE 53277,3:POKE 53258,1:POKE 53259,1
JI 80 OFF=PM*256+1024-AT
BK 90 HI=INT(OFF/256)
CO 100 LO=OFF-HI*256:POKE VT+2,LO:POKE VT+3,HI
PG 110 POKE 704,168:POKE 705,248
OH 120 POKE 706,200:POKE 707,250
BF 130 D$="{45 ,}":FOR I=2 TO 1024 STEP 25:P$(
       I)=D$:NEXT I:POKE 53248,80
CN 135 POKE 53257,3
JN 140 X=80:Y=90:X1=3:Y1=347
MA 150 X2=200:Y2=600:X3=200:Y3=855
OI 170 P$(Y)=D$:P$(Y1)=D$:P$(Y2)=D$:P$(Y3)=D$
```

```
BB 180  SHIP$="{5 ,}■■{■}■''■{=}?{=}■''■{■}■■
        {5 ,}"
BL 190  LAZER$="{5 ,}■{13 ,}■{7 ,}"
NP 200  ENE$="{5 ,}{X}{BACK S}<{BACK S}{INSERT}
        fff{INSERT}{BACK S}<{BACK S}{X}{5 ,}"
OA 210  P$(Y2)=ENE$:P$(Y3)=ENE$
LD 310  S=STICK(Ø):SCR=SCR+1:POKE LL,SCR:IF SCR
        >4Ø THEN SCR=1
OF 340  IF S=13 THEN Y=Y+5:IF Y>13Ø THEN Y=13Ø
II 350  IF S=14 THEN Y=Y-5:IF Y<25 THEN Y=2Ø
IK 360  P$(Y)=SHIP$:POKE 53250,X2:POKE 53251,X3
IA 370  IF ACT=Ø THEN IF STRIG(Ø)=Ø THEN X1=8Ø:
        ACT=1:Y1=Y+256:P$(Y1)=LAZER$:SOUND Ø,1Ø
        ,8,15
EB 380  IF ACT=1 THEN X1=X1+15:POKE 53249,X1:IF
         X1>19Ø THEN GOSUB 4ØØØ
MC 390  X2=X2-RX1:X3=X3-RX2:IF X2<4Ø THEN GOSUB
         5ØØ
GD 4ØØ  IF X3<Ø THEN GOSUB 55Ø
PL 41Ø  IF PEEK(53260)>2 THEN 2ØØØ
NA 42Ø  IF PEEK(53261)>1 THEN 6ØØ
GK 49Ø  GOTO 31Ø
BO 5ØØ  X2=255:P$(Y2)=D$:Y2=RND(Ø)*1ØØ+55Ø:POKE
         7Ø6,RND(Ø)*255:SP1=SP1+Ø.5:RX1=RND(Ø)*
        3+SP1:P$(Y2)=ENE$
BC 51Ø  IF SP1>25 THEN SP1=4
HH 52Ø  RETURN
CJ 55Ø  X3=255:P$(Y3)=D$:Y3=RND(Ø)*1ØØ+8ØØ:POKE
         7Ø7,RND(Ø)*254:SP2=SP2+Ø.5:RX2=RND(Ø)*
        3+SP2:P$(Y3)=ENE$
BM 56Ø  IF SP2>36 THEN SP2=5
HM 57Ø  RETURN
IN 6ØØ  SCORE=SCORE+25Ø
PP 61Ø  IF PEEK(53262)>1 THEN 3ØØØ
AC 62Ø  IF PEEK(53263)>1 THEN 31ØØ
GG 63Ø  GOTO 31Ø
DL 8ØØ  GRAPHICS 2+16:POKE 559,62:POKE 53248,Ø:
        POKE 53249,Ø:POKE 53250,Ø:POKE 53251,Ø:
        POKE 53278,Ø
CN 81Ø  POSITION Ø,3:? #6;"■■■■■■■■■■■■■■■■■■■■"
DA 82Ø  POSITION 7,5:? #6;SCORE
LB 83Ø  SOUND Ø,Ø,Ø,Ø:FOR W=1 TO 1ØØØ:NEXT W:RU
        N
EI 1ØØØ  GRAPHICS 2+16:POKE 559,62:? #6
HH 1Ø1Ø  ? #6;"*■*■*■*■*■*■*■*■*■*■*■*■*"
GK 1Ø2Ø  ? #6;"■ compute! books  ■"
EL 1Ø3Ø  ? #6;"*  **■■■■■■**{3 SPACES}*"
DA 1Ø4Ø  ? #6;"■{17 SPACES}■"
IM 1Ø5Ø  ? #6;"* >Fr■■T At■■Ck<  *"
DC 1Ø6Ø  ? #6;"■{17 SPACES}■"
```

```
GO 1070 ? #6;"*{3 SPACES}PRESS START
        {3 SPACES}*"
HP 1090 ? #6;"*■*■*■*■*■*■*■*■*■*■*"
CC 1110 POKE 708,122:POKE 709,236
OF 1120 AA1=0:RXRX=6
HB 1130 FOR I=2 TO 15:SOUND 0,AA1,10,I:SOUND 1
        ,AA1+1,10,I:NEXT I:AA1=AA1+1:IF AA1>RX
        RX THEN 1150
EA 1140 FOR I=15 TO 2 STEP -1:SOUND 0,AA1,10,I
        :SOUND 1,AA1+1,10,I:NEXT I:AA1=AA1+1:G
        OTO 1130
JF 1150 SOUND 1,0,0,0
ED 1160 IF PEEK(53279)=6 THEN 1180
IC 1165 SOUND 0,RND(0)*255,10,2:FOR W=1 TO 10:
        NEXT W
MK 1170 GOTO 1160
MI 1180 GRAPHICS 7+16:DL=PEEK(560)+256*PEEK(56
        1):LL=DL+4:POKE 53256,0
EJ 1190 POKE 708,136:COLOR 1
MM 1200 FOR I=0 TO 50:SOUND 0,RND(0)*255,10,10
BM 1210 PLOT RND(0)*159,RND(0)*95:NEXT I
LO 1220 POSITION 10,10:? #6;"{4 SPACES}AAAAA
        {4 SPACES}"
EC 1230 POSITION 10,11:? #6;"{3 SPACES}AAAAAAA
        {3 SPACES}"
MG 1240 POSITION 10,12:? #6;"  AAAAAAAAA  "
EN 1250 POSITION 10,13:? #6;" AAABAABBAAA "
ND 1260 POSITION 10,14:? #6;"AAABAABBBBAAA"
NH 1270 POSITION 10,15:? #6;"AABAABBBBBBAA"
NI 1280 POSITION 10,16:? #6;"AABAAABBBBBAA"
NJ 1290 POSITION 10,17:? #6;"AAAAABBBBBAAA"
EP 1300 POSITION 10,18:? #6;" AAAABBBBAAA "
MN 1310 POSITION 10,19:? #6;"  AAABBAAAA  "
ED 1320 POSITION 10,20:? #6;"{3 SPACES}AABAAAA
        {3 SPACES}"
MC 1330 POSITION 10,21:? #6;"{4 SPACES}AAAAA
        {4 SPACES}"
AI 1340 POSITION 50,50:? #6;"  BB  "
IM 1350 POSITION 50,51:? #6;" BAAB "
BD 1360 POSITION 50,52:? #6;"BBBABB"
BF 1370 POSITION 50,53:? #6;"BBABBB"
JD 1380 POSITION 50,54:? #6;" BBAB "
BC 1390 POSITION 50,55:? #6;"  BB  "
JK 1400 RESTORE 1500:COLOR 2
AD 1410 FOR I=0 TO 159:SOUND 0,I,8,15
BD 1420 READ MT:PLOT I,MT
FP 1430 DRAWTO I,95:NEXT I:SOUND 0,0,0,0:RETUR
        N
```

```
EK 1500 DATA 90,89,87,86,85,84,83,82,82,83,84,
        85,86,86,85,84,83,82,81,80,79,77,75,73
        ,72,71,70,70,69,69,69,68,68
LK 1510 DATA 68,68,69,69,69
NE 1520 DATA 70,70,71,72,71,70,69,68,67,66,65,
        64,63,62,61,60,61,62,63,63,62,61,62,63
        ,64,65,67,69,72,74,76
CI 1530 DATA 75,74,73,70,68,65,63,60,58,57,58,
        60,61,63,64,65,67,65,64,62,60,60,60
AM 1540 DATA 61,62,63,64,65,66,67,68,69,70,70,
        70,71,74,76,78,79,80,80,80,81
BE 1550 DATA 77,78,79,80,81,82,83,84,85,86,87,
        88,89,90
GI 1560 DATA 90,90,91,92,93,94,94,93,93,92,92,
        91,91,90,90,89,88,89,88,87,86,85,85
EI 1570 DATA 84,83,84,85,86,87,88,89,90,90,90
FE 2000 POKE 712,14:FOR W=1 TO 10:NEXT W:POKE
        712,0:POKE 53256,1:D$="{24 ,}":P$(Y)=D
        $:Q9=3
FG 2001 POKE 53249,1
FN 2005 FB=FB+1:IF FB>9 THEN FB=1
JN 2010 IF FB<5 THEN FM$="{6 ,}{[U]}@{[X]}T[J]J{6 ,}
        ":P$(Y)=FM$
LK 2015 Y=Y+Q9:IF Y>215 THEN Q9=-3
IM 2020 IF FB>4 THEN FM$="{6 ,}R{J}Q({[R]}%{6 ,}
        ":P$(Y)=FM$
BH 2030 ZZ=ZZ+1:POKE LL,ZZ:IF ZZ>40 THEN ZZ=1
CN 2040 X=X+2:POKE 53248,X:SOUND 0,X,8,15:IF X
        >200 THEN X2=200:X3=200:X=80:POKE 5325
        0,X2:POKE 53251,X3:GOTO 2060
MH 2050 GOTO 2005
ME 2060 P$(Y)=D$:Y=90
EF 2065 SOUND 0,0,0,0:P$(Y)=SHIP$:POKE 53248,X
        :POKE 53256,0:POKE 53278,1:LIFE=LIFE-1
        :IF LIFE<1 THEN 800
DJ 2070 GOSUB 500:GOSUB 550:GOTO 310
JC 3000 XXX=X2
NB 3020 FOR I=XXX TO 210 STEP 10:SOUND 0,I,8,1
        5:POKE 706,I:POKE 53250,I:NEXT I:GOSUB
         4000:POKE 53278,0
OC 3030 GOSUB 500:GOTO 310
ED 3100 XXX1=X3:GOSUB 4000
BF 3110 FOR I=XXX1 TO 210 STEP 10:SOUND 0,I-20
        ,8,15:POKE 707,I:POKE 53251,I:NEXT I:P
        OKE 53278,0
OO 3120 SOUND 0,0,0,0:GOSUB 550:GOTO 310
KN 4000 X1=0:ACT=0:SOUND 0,0,0,0:POKE 53249,X1
        :P$(Y1)=D$:RETURN
```

Electronic Football

Buren Earl Wells, Jr.

"Electronic Football" is a game for those who want to do something a little closer to home than shooting aliens or running from goblins. Joystick required.

"Electronic Football" is for two players using one joystick. Each player takes a turn controlling the offense, and the computer controls the defense.

At the beginning of each game, the players have the option of choosing one of five possible skill levels. Each game is divided into two nine-minute, 59-second halves in which the first player starts the first half and the second player starts the second half. The clock stops after each play. Scoring is accomplished by either a field goal (three points) or a touchdown (seven points). The score is displayed when a player scores and at the end of the game. It can also be displayed before any play begins by pressing the trigger button on the joystick.

The offense consists of one man that must elude 11 would-be tacklers in his quest for a score. You can move this man in the upward, downward, or forward direction by using a joystick which is connected in the first joystick port. Four forward steps by the offensive man are equal to a yard. You can't move backward, however, so once a forward step is made, there is no turning back. As in real football the offensive player has four downs in which to move ten yards, or the ball turns over and it's the other player's turn to control the offense.

The Huddle

Before every play the current down, field position, yardage, time remaining, and the player who is controlling the offense are displayed on the border of the playing field. It's at this time that the offensive player must decide what to do. If the player wants to kick, he or she simply presses the START button. If the offensive man is out of range of a field goal (more than 50 yards away), or if the field goal try is unsuccessful, the kick is a punt and the ball turns over. If the offensive player desires to run instead, the play is started by moving the joystick.

Field Position

The color of the sideline markers indicates whose territory the offensive man is in at the beginning of the play. If the sideline is the same color as the offensive man, then the man is inside his own 50-yard line. Otherwise, the offensive man is inside the opponent's 50 and may be nearing a score.

Electronic Football

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
FL 10 GOTO 1000
CO 20 IF STICK(0)<>15 THEN ST=STICK(0):GOSUB 2
      00
FK 30 IF L>LP THEN GOSUB 300
GK 40 L=L+1:GOTO 20
OO 200 H=0:V=H:TR=H:Z=H:IF ST=13 AND PLV<9 THE
      N LOCATE PLH,PLV+1,Z:V=1
JJ 210 IF ST=14 AND PLV>3 THEN LOCATE PLH,PLV-
      1,Z:V=-1
LF 220 IF ST=7 THEN LOCATE PLH+1-20*(PLH=19),P
      LV,Z:H=1:IF PLH>18 THEN H=-19:FIVE=FIVE
      +(Z<>2 AND Z<>130)
OI 230 IF Z=130-128*(PL=2) THEN M(I,1)=PLH:M(I
      ,2)=PLV:GOTO 600
BL 240 IF Z THEN POKE 53279,0:COLOR 32:PLOT PL
      H,PLV:PLH=PLH+H:PLV=PLV+V:COLOR 1+128*(
      PL=2):PLOT PLH,PLV
LD 250 IF PI+INT(FIVE*5+PLH/4)+YDS>99 THEN 800
HI 260 RETURN
IJ 300 I=INT(RND(0)*(11-2*(PS>65 AND PS<89)))*
      (PLH<6)+(10-INT(RND(0)*2)-INT(RND(0)*3)
      *(PLH<9))*(PLH>5)
OF 310 L=0:H=L:V=L:IF RND(0)>0.4 AND M(I,2)<>P
      LV AND I<>CR THEN 350
MK 320 IF PLH<M(I,1) THEN M(I,1)=M(I,1)-1:H=1
PI 330 IF PLH>M(I,1) THEN M(I,1)=M(I,1)+1:H=-1
GK 340 GOTO 370
OM 350 IF PLV<M(I,2) THEN M(I,2)=M(I,2)-1:V=1
BK 360 IF PLV>M(I,2) THEN M(I,2)=M(I,2)+1:V=-1
CP 370 LOCATE M(I,1),M(I,2),Z:IF Z=1+128*(PL=2
      ) THEN 600
KG 380 IF Z=130-128*(PL=2) THEN M(I,1)=M(I,1)+
      H:M(I,2)=M(I,2)+V:RETURN
IF 390 COLOR 32:PLOT M(I,1)+H,M(I,2)+V:COLOR 1
      30-128*(PL=2):PLOT M(I,1),M(I,2):RETURN
EA 400 GRAPHICS 2+16:POKE 756,CHSET/256:T1=PEE
      K(20):T2=PEEK(19):TIME=T1/60+T2*256/60:
      SETCOLOR 1,9,8
```

```
NG 410 Q=128*((PS<51)*(PL=1) OR (PS>50)*(PL=2)
       ):FOR Y=2 TO 10 STEP 8:FOR X=0 TO 19 ST
       EP 4
NF 420 COLOR 131-Q:PLOT X,Y:COLOR 133-Q:PLOT X
       +1,Y:PLOT X+2,Y:COLOR 132-Q:PLOT X+3,Y:
       NEXT X:NEXT Y
AG 430 IF TIME>599 THEN 850
BK 440 POSITION 0,0:? #6;"DOWN POSITION YARDS"
AI 450 TM=INT(TIME/60):TS=(TIME/60-TM)*60
CJ 460 POSITION 2,1:? #6;DOWN:POSITION 8,1:? #
       6;POS:POSITION 16,1:? #6;YARDS
PA 470 POSITION 0,11:? #6;"TIME ";9-TM;".";59-
       INT(TS):IF INT(TS)>49 THEN POSITION 7,1
       1:? #6;"0";59-INT(TS)
NB 480 POSITION 11,11:? #6;"PLAYER ";PL:PLH=0:
       PLV=6:COLOR 1+128*(PL=2):PLOT PLH,PLV
FP 490 T=2:FOR I=0 TO 6:M(I,1)=3:M(I,2)=1+T:T=
       T+1:COLOR 130-128*(PL=2):PLOT 3,M(I,2):
       NEXT I
KN 500 T=3:FOR I=7 TO 8:M(I,1)=6-(PS>89):M(I,2
       )=1+T:T=T+4:PLOT M(I,1),M(I,2):NEXT I
DG 510 T=4:FOR I=9 TO 10:M(I,1)=8-2*(PS>89):M(
       I,2)=1+T:T=T+2:PLOT M(I,1),M(I,2):NEXT
       I
PA 520 IF STRIG(0)=0 THEN GOSUB 900:GOTO 440
CB 530 IF PEEK(53279)<7 THEN POKE 19,T2:POKE 2
       0,T1:GOTO 950
JA 540 IF STICK(0)=15 THEN 520
IE 550 POKE 77,0:LP=(INT(RND(0)*3))*(PS<65)+2*
       (PS<40)+2*(PS<80)+SK:POKE 19,T2:POKE 20
       ,T1:CR=(RND(0)<0.5)*6:GOTO 20
KP 600 SOUND 0,72,10,15:FOR A=1 TO 100:NEXT A:
       SOUND 0,0,0,0:FOR A=1 TO 4:COLOR 32:PLO
       T M(I,1)+H,M(I,2)+V
PM 610 FOR AA=0 TO 60:NEXT AA:COLOR 130-128*(P
       L=2):PLOT M(I,1)+H,M(I,2)+V:FOR AA=0 TO
        60:NEXT AA:NEXT A
IP 620 YDS=INT(FIVE*5+PLH/4)+YDS:YARDS=10-YDS
AH 630 PS=PI+YDS:POS=PS:IF PS>50 THEN POS=100-
       PS
BK 640 IF YDS>9 THEN YARDS=10:DOWN=0:PI=PI+YDS
       :YDS=0
KO 650 FIVE=0:DOWN=DOWN+1:IF DOWN=5 THEN 700
NM 660 IF YARDS>POS AND PS>90 THEN YARDS=POS
GK 670 GOTO 400
JM 700 FOR I=1 TO 10:SOUND 0,72,10,15:FOR A=1
       TO 20:NEXT A:SOUND 0,0,0,0:FOR B=1 TO 5
       :NEXT B:NEXT I
HI 710 IF PL=1 THEN PL=2:GOTO 730
KD 720 PL=1
```

```
DK 730  PS=100-PS:PI=PS:DOWN=1:YARDS=10:YDS=0:P
        OS=PS:IF PS>50 THEN POS=100-PS
GI 740  GOTO 400
CB 800  FOR A=1 TO 4:FOR I=10 TO 70:SOUND 0,I,1
        0,15:NEXT I:SOUND 0,0,0,0:NEXT A
FO 810  SCR(PL)=SCR(PL)+7:PS=80:POS=20:FIVE=0:G
        OSUB 900:GOTO 710
IL 850  IF HH THEN 880
HC 860  POSITION 6,6:? #6;"HALF TIME":GOSUB 900
        :POSITION 0,6:? #6;"player 2 is up NOW"
        :FOR I=1 TO 600:NEXT I
ME 870  HH=1:PL=1:PS=80:POKE 19,0:POKE 20,0:GOT
        O 710
MJ 880  HH=2:POSITION 0,5:? #6;"*** END OF GAME
         ***":? #6:? #6;"PRESS START TO PLAY"
JK 890  GOSUB 900:IF PEEK(53279)<7 THEN RUN
IA 895  GOTO 890
HF 900  POSITION 0,0:? #6;"PLAYER 1{4 SPACES}PL
        AYER 2":POSITION 0,1:? #6;"{19 SPACES}"
NG 910  POSITION 3,1:? #6;SCR(1):POSITION 14,1:
        ? #6;SCR(2):IF HH=2 THEN RETURN
DB 920  FOR I=1 TO 300:NEXT I
AI 930  POSITION 0,0:? #6;"{40 SPACES}":RETURN
LJ 950  IF RND(0)<(PS-40)/44 THEN 970
NE 960  PS=PS+INT(RND(0)*(100-PS)):GOTO 700
CH 970  FOR A=1 TO 2:FOR I=10 TO 70:SOUND 0,I,1
        0,15:NEXT I:NEXT A:SOUND 0,0,0,0
GC 980  SCR(PL)=SCR(PL)+3:PS=80:POS=20:FIVE=0:G
        OSUB 900:GOTO 710
JD 1000  DIM M(11,2),T$(3),SCR(2):PS=20:PI=PS:P
         OS=PS:PL=1:DOWN=PL:YARDS=10:SCR(1)=A:S
         CR(2)=A:OPEN #1,4,0,"K:"
BH 1010  GRAPHICS 2+16:? #6;"ELECTRONIC FOOTBAL
         L":POSITION 2,6:? #6;"levels of skill
         "
FO 1020  POSITION 0,8:? #6;"easy{12 SPACES}hard"
         :? #6;"1{3 SPACES}2{3 SPACES}3
         {3 SPACES}4{3 SPACES}5"
KK 1030  GET #1,SK:IF SK<49 OR SK>53 THEN 1030
IK 1040  CHSET=(PEEK(106)-8)*256:IF PEEK(CHSET+
         12)=16 THEN POKE 19,0:POKE 20,0:SK=54-
         SK:CLOSE #1:GOTO 400
EO 1050  POSITION 4,3:? #6;"please wait"
IC 1060  FOR I=0 TO 512:POKE CHSET+I,PEEK(57344
         +I):POKE 708,PEEK(53770):NEXT I
LO 1070  READ A:IF A=-1 THEN 1040
BI 1080  FOR J=0 TO 7:READ B:POKE CHSET+A*8+J,B
         :POKE 708,PEEK(53770):NEXT J:GOTO 1070
GF 1100  DATA 1,0,60,52,60,16,30,16,40
KC 1110  DATA 2,0,60,36,60,24,126,24,36
```

```
HF 1120 DATA 3,0,128,128,255,255,128,128,0
MP 1130 DATA 4,0,1,1,255,255,1,1,0
FH 1140 DATA 5,0,0,0,255,255,0,0,0,-1
```

The Heavenly Gates

Robert F. Host

Can you keep the demons out and still maintain your composure? "Heavenly Gates" is also a good example of how to mix BASIC with machine language. Requires a joystick.

The demons have broken free of their prison and are storming the gates of heaven. You are the lone defending angel and must keep the attacking demons from penetrating the gates. Your mere touch sends a demon back in the direction it came from, but once a demon hits a border, it can rebound in any direction.

You can repel only one demon at a time, so be sure to break contact with one demon before trying to fight off another. Control the position of your angel with a joystick in port 1. You get points for time the heavenly gates are free of demons, but you lose points at a much faster rate when there is at least one demon inside the gates. A 5-point bonus is awarded every time your angel repels a demon. To advance to the next round, you must at least maintain your score from the previous round. Each round has a time limit of about two minutes. If you make it through all nine rounds, you receive a 5000-point bonus and a salute from the heavenly trumpets.

This game is not as easy as it looks. Your angel can't keep up with the three demons if they're attacking from opposite directions. It's sometimes best to let a demon pass through the gates in order to get all the demons on the same side. In the faster rounds, trying to repel a demon from the side may be ineffective. Try to meet the demons head-on if possible. Learn where the boundaries for point loss are—don't waste time going after demons that aren't costing you points. Remember, the key to scoring points is keeping demons out of the gate area. If there are no demons attacking, keep your angel near the gates—don't chase the demons.

After typing in the program, be sure to *save it before you run it.*

Memory Allocation

The main problem with combining BASIC and machine language is memory allocation. To write the main loop of the program, I had to find an area of memory which BASIC wouldn't touch. Since I used page 6 for a vertical blank, and

since I anticipated that the routine would not be relocatable and thus could not be coded as a string, I had to look elsewhere.

Fortunately, the Atari has a couple of pointers which helped. Memory locations 144 and 145 point to the highest byte of memory used by the BASIC program in memory. By the direct statement PRINT PEEK(144)+256*PEEK(145), I was able to determine the approximate top of memory used by the BASIC program after the bulk of the BASIC was written. I started the machine language routine above this, leaving about 4K between the top of the BASIC and the beginning of the machine language routine.

Location 106 was the other Atari pointer used in the development of this program. It points to the page number of the top of free RAM. However, space for screen memory and player/missile graphics data was needed below the top of free RAM. For graphics 7 screen memory, 32 pages of RAM were reserved, and 8 pages were reserved for double-line-resolution P/M graphics, yielding 40 pages of reserved memory (10K) below the value in location 106. For a 48K system, location 106=160, so the highest memory location available for the machine code was (160−40)*256, or location 30720. However, since the bottom four pages of P/M memory are not used, I used the bottom 27 bytes (30720–30746) for storage of various machine language variables.

Another problem with mixing BASIC and machine language is communication between the two. Such variables as the score, the round number, and the speed of the players must be updated and transferred as the game progresses. You can accomplish this by using additional parameters in the USR function, but this involves complex stack manipulations. An easier way is to POKE the variable value from BASIC into a reserved memory location where it can be picked up by the machine language routine. The PEEK function can be used by BASIC to retrieve a variable updated by the machine language routine and stored in a known location. Examples of these techniques are found in lines 305, 307, 7805, and 16020.

Credit

Lines 8000–9600 were derived from a player/missile movement routine by Chris Nicotra, reprinted courtesy of ANTIC Publishing, Inc. (June 1983).

Program Description

Lines	Description
7–10	Initialize constants and subroutine line numbers
13–18	Initialize demon directions
20–30	BASIC time-delay routine
33–107	Title screen routine
110–226	Initialize player/missile graphics
227–276	Draw playfield
295–307	Enable DLI and screen; initialize timers and round parameters
310–320	Execute main loop (machine language)
7800–7830	End of round handler
7900–7950	Game loss handler
8000–9600	VBLANK data and P/M setup
10000	Data for heavenly gates
12000–12090	Install DLI
14000–15070	Play tunes
16000–16030	Go to next round
17000–18060	Win the game routine
20000–20999	Data for main loop machine language program

Heavenly Gates

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
HK 7 RNDOVR=30745:TUNE=14000:ENTUNE=15000:WIN=
     17000:WINSOUND=18000
DJ 8 POKE 82,0:POKE 83,39:POKE RNDOVR,0
IA 9 SCORE=0:ROUND=50:R=1
PE 10 RANDOM=53770:WAIT=20:PAGE6=8000:PMINIT=9
      000:XDIRECTION=13:YDIRECTION=15:NXTROUND
      =16000:DLI=12000
AC 12 GOTO 33
HP 13 REM ENEMY X DIRECTION
JI 14 DIRX=INT(PEEK(RANDOM)*3/256)-1:IF DIRX<0
       THEN DIRX=255:RETURN
IC 15 REM ENEMY Y DIRECTION
ID 16 DIRY=INT(PEEK(RANDOM)*3/256)-1:IF DIRY<0
       THEN DIRY=255
OC 17 IF DIRX=0 AND DIRY=0 THEN GOTO 16
EJ 18 RETURN
AE 20 REM TIME DELAY
ML 22 FOR DELAY=1 TO JIFFIES
KG 24 TICK=PEEK(20)
MO 26 IF TICK=PEEK(20) THEN 26
BI 28 NEXT DELAY
ED 30 RETURN
```

```
JC 33  REM TITLE ROUTINE
PM 54  GRAPHICS 18:POSITION Ø,Ø:? #6;"*********
       ***********":POSITION Ø,1Ø:? #6;"*******
       ***********"
HB 58  FOR I=1 TO 9:POSITION Ø,I:? #6;"*":POSIT
       ION 19,I:? #6;"*":NEXT I:I=Ø
AI 62  POSITION 4,2:? #6;"ThE tEaUGNlT":POSITIO
       N 7,4:? #6;"FAtES"
LP 77  JIFFIES=6Ø*2:GOSUB WAIT:POSITION 4,7:? #
       6;"BY BOB HOST"
GE 1ØØ GOSUB PAGE6
BH 1Ø1 SETCOLOR 3,1,12
OH 1Ø2 POSITION 4,8:? #6;"press start"
CK 1Ø4 POKE 53279,Ø
NK 1Ø6 IF PEEK(53279)=6 THEN 11Ø
EN 1Ø7 TEMP=PEEK(7Ø8):POKE 7Ø8,PEEK(7Ø9):POKE
       7Ø9,PEEK(71Ø):POKE 71Ø,PEEK(711):POKE 7
       11,TEMP
DC 1Ø8 JIFFIES=3Ø:GOSUB WAIT:GOTO 1Ø6
IO 11Ø GRAPHICS 7:POKE 559,Ø:GOSUB DLI:SETCOLO
       R Ø,15,1Ø:SETCOLOR 4,8,12:SETCOLOR 1,12
       ,1Ø:SETCOLOR 2,1,4:POKE 1562,Ø:POKE RND
       OVR,Ø
BH 111 RESTORE 2ØØØØ:POKE 752,1:I=Ø:? "SCORE:
       {8 SPACES}TIME REMAINING:":? "GOAL:
       {9 SPACES}ROUND:";R;
GL 112 IF PEEK(256ØØ)=1Ø4 THEN GOTO 119
LB 113 READ A:IF A>-1 THEN POKE 256ØØ+I,A:POKE
        712,A:I=I+1:GOTO 113
FG 119 REM PLAYER 0 DATA
CL 13Ø POKE 1571,56:POKE 1572,186
CM 14Ø POKE 1573,254:POKE 1574,56
FJ 15Ø POKE 1575,124:POKE 1576,124
GG 16Ø POKE 1577,254:POKE 1578,254
LC 179 REM SIZE OF PLAYER 0
LO 18Ø POKE 1611,7:POKE 53256,Ø
FG 181 REM PLAYER 1 DATA
GC 182 POKE 1581,129:POKE 1582,129
GE 183 POKE 1583,255:POKE 1584,126
AD 184 POKE 1585,9Ø:POKE 1586,6Ø
AL 185 POKE 1587,66:POKE 1588,6Ø
LE 189 REM SIZE OF PLAYER 1
MB 19Ø POKE 1612,7:POKE 53257,Ø
FI 191 REM PLAYER 2 DATA
GC 192 POKE 1591,126:POKE 1592,219
GK 193 POKE 1593,255:POKE 1594,219
DA 194 POKE 1595,1Ø2:POKE 1596,6Ø
GL 195 POKE 1597,126:POKE 1598,231
LD 196 REM SIZE OF PLAYER 2
MK 197 POKE 1613,7:POKE 53258,Ø
```

```
GA 198 REM PLAYER 3 DATA
FM 199 POKE 1601,129:POKE 1602,255
BM 200 POKE 1603,255:POKE 1604,90
BO 201 POKE 1605,126:POKE 1606,36
ON 202 POKE 1607,60:POKE 1608,24
KJ 203 REM SIZE OF PLAYER 3
MB 204 POKE 1614,7:POKE 53259,0
OL 208 GOSUB PMINIT
ID 209 POKE 704,14:POKE 705,0:POKE 706,30:POKE
       707,40
FO 224 GOSUB XDIRECTION:GOSUB YDIRECTION:POKE
       30731,DIRX:POKE 30735,DIRY
GB 225 GOSUB XDIRECTION:GOSUB YDIRECTION:POKE
       30732,DIRX:POKE 30736,DIRY
GE 226 GOSUB XDIRECTION:GOSUB YDIRECTION:POKE
       30733,DIRX:POKE 30737,DIRY
AD 227 COLOR 3:PLOT 109,79:DRAWTO 81,55:DRAWTO
       76,55:POSITION 49,79:POKE 765,3:XIO 18
       ,#6,0,0,"S:"
FP 228 COLOR 2:PLOT 49,79:DRAWTO 76,55:DRAWTO
       0,55:POSITION 0,79:POKE 765,2:XIO 18,#6
       ,0,0,"S:"
HM 229 PLOT 159,79:DRAWTO 159,55:DRAWTO 81,55:
       POSITION 109,79:POKE 765,2:XIO 18,#6,0,
       0,"S:"
OB 239 REM MOUNTAIN #1
GN 240 COLOR 3:PLOT 56,51:DRAWTO 28,30:DRAWTO
       27,30:POSITION 0,51:POKE 765,3:XIO 18,#
       6,0,0,"S:"
ON 245 PLOT 62,54:DRAWTO 56,51:DRAWTO 0,51:POS
       ITION 0,54:XIO 18,#6,0,0,"S:"
OD 249 REM MOUNTAIN #2
HA 250 PLOT 74,52:DRAWTO 68,47:DRAWTO 67,47:PO
       SITION 59,52:XIO 18,#6,0,0,"S:"
GI 251 PLOT 77,54:DRAWTO 74,52:DRAWTO 59,52:PO
       SITION 62,54:XIO 18,#6,0,0,"S:"
OA 254 REM MOUNTAIN #3
OA 255 PLOT 126,49:DRAWTO 101,21:DRAWTO 100,21
       :POSITION 70,49:XIO 18,#6,0,0,"S:"
MO 256 PLOT 132,54:DRAWTO 126,49:DRAWTO 70,49:
       POSITION 77,54:XIO 18,#6,0,0,"S:"
OG 259 REM MOUNTAIN #4
BH 260 PLOT 152,28:DRAWTO 128,11:DRAWTO 127,11
       :POSITION 108,28:XIO 18,#6,0,0,"S:"
BP 261 PLOT 159,33:DRAWTO 152,28:DRAWTO 108,28
       :POSITION 113,33:XIO 18,#6,0,0,"S:"
CC 262 PLOT 159,54:DRAWTO 159,33:DRAWTO 113,33
       :POSITION 132,54:XIO 18,#6,0,0,"S:"
DH 263 PLOT 113,33:DRAWTO 132,54
HF 269 REM HEAVENLY GATES
```

```
GK 270  COLOR 1:FOR X=0 TO 2:PLOT 49+X,79:DRAWT
        O 49+X,45:NEXT X
NH 271  FOR X=0 TO 2:PLOT 107+X,79:DRAWTO 107+X
        ,45:NEXT X
AL 272  REM DRAW BORDER
AL 273  COLOR 1:PLOT 0,0:DRAWTO 159,0:DRAWTO 15
        9,79:DRAWTO 0,79:DRAWTO 0,0
JF 274  RESTORE 10000:FOR I=0 TO 4:POKE 40807+I
        ,16:NEXT I
JI 275  FOR X=0 TO 48 STEP 4:READ A:PLOT 55+X,7
        9:DRAWTO 55+X,A:NEXT X
GC 276  PLOT 52,53:DRAWTO 60,49:DRAWTO 66,51:DR
        AWTO 72,51:DRAWTO 79,47:DRAWTO 86,51:DR
        AWTO 92,51:DRAWTO 98,49:DRAWTO 106,53
CG 280  POKE 712,140:POKE 54286,192:SCORSAV=0
FO 290  REM INITIAL PLAYER POSITIONS
LG 291  POKE 53248,120:POKE 30722,120
DA 292  POKE 1536,54:POKE 30726,54
CL 293  POKE 53249,55:POKE 30723,55:POKE 1537,3
        0:POKE 30727,30
IH 294  POKE 53250,190:POKE 1538,30:POKE 30724,
        190:POKE 30728,30
HO 295  POKE 53251,120:POKE 1539,30:POKE 30725,
        120:POKE 30729,30
HH 300  REM
BL 305  POKE 30742,ROUND:POKE RNDOVR,0
CK 306  POKE 559,46:JIFFIES=3*60:GOSUB WAIT
OE 307  POKE 18,0:POKE 19,0:POKE 20,0
HH 310  REM MAIN LOOP
GL 320  Q=USR(25600)
BD 7800 REM GO TO NEXT ROUND
AC 7805 SCORE=10000*(PEEK(40807)-16)+1000*(PEE
        K(40808)-16)+100*(PEEK(40809)-16)+10*(
        PEEK(40810)-16)+PEEK(40811)-16
HH 7806 IF SCORE>SCORSAV THEN GOTO 7810
NP 7807 GOTO 7900
JN 7810 ROUND=INT(ROUND*0.87):IF ROUND<15 THEN
         ROUND=15
NN 7820 SCORSAV=SCORE:GOSUB TUNE:GOSUB NXTROUN
        D
JO 7830 GOTO 300
AL 7899 REM END OF GAME
GJ 7900 POKE 1562,1:GRAPHICS 18:FOR X=0 TO 3:P
        OKE 53248+X,0:NEXT X:POSITION 2,3:? #6
        ;"GAME OVER":GOSUB ENTUNE
CG 7905 GOSUB PMINIT
AD 7910 POSITION 1,5:? #6;"FINAL SCORE =";SCOR
        E
DJ 7920 POSITION 2,8:? #6;"PRESS START"
EA 7930 POSITION 2,9:? #6;" TO REPLAY"
```

```
JG 7940 IF PEEK(53279)<>6 THEN 7940
GO 7945 POKE 53278,1
MB 7950 SCORE=0:ROUND=50:POKE RNDOVR,0:R=1:GOT
        O 110
EC 8000 REM ROUTINE TO LOAD THE P/M
NJ 8010 REM HANDLER.
LL 8015 RESTORE 8100
KD 8020 FOR I=0 TO 234
DB 8030 READ J
LE 8040 POKE 1536+I,J
FF 8050 NEXT I
KO 8060 RETURN
ID 8100 DATA 0,0,0,0,0
IE 8110 DATA 0,0,0,0,0
IF 8120 DATA 0,0,0,0,0
IG 8130 DATA 0,0,0,0,0
IH 8140 DATA 0,0,0,0,0
PP 8150 DATA 0,0,35,6,45
BA 8160 DATA 6,55,6,65,6
IK 8170 DATA 0,0,0,0,0
IL 8180 DATA 0,0,0,0,0
IM 8190 DATA 0,0,0,0,0
IE 8200 DATA 0,0,0,0,0
IF 8210 DATA 0,0,0,0,0
IG 8220 DATA 0,0,0,0,0
IH 8230 DATA 0,0,0,0,0
II 8240 DATA 0,0,0,0,0
OO 8250 DATA 0,0,0,0,104
AJ 8260 DATA 162,6,160,90,169
KN 8270 DATA 7,32,92,228,96
KF 8280 DATA 173,26,6,208,6
NC 8290 DATA 32,187,6,32,104
OF 8300 DATA 6,76,98,228,169
CH 8310 DATA 3,133,203,166,203
MN 8320 DATA 189,0,6,133,205
HM 8330 DATA 188,8,6,157,8
KK 8340 DATA 6,189,75,6,133
GH 8350 DATA 204,138,10,170,189
AJ 8360 DATA 16,6,133,206,189
AM 8370 DATA 17,6,133,207,189
AP 8380 DATA 27,6,133,208,189
AN 8390 DATA 28,6,133,209,166
CO 8400 DATA 204,169,0,145,206
DC 8410 DATA 136,192,255,240,3
GA 8420 DATA 202,16,246,164,204
JO 8430 DATA 177,208,164,205,145
JF 8440 DATA 206,136,132,205,192
DJ 8450 DATA 255,240,4,198,204
DN 8460 DATA 16,237,198,203,16
BD 8470 DATA 178,96,162,3,173
```

```
AE 8480 DATA 24,6,133,206,173
AN 8490 DATA 25,6,133,207,188
JO 8500 DATA 12,6,189,231,6
HA 8510 DATA 49,206,145,206,188
HG 8520 DATA 4,6,189,227,6
GE 8530 DATA 17,206,145,206,152
MM 8540 DATA 157,12,6,202,16
KC 8550 DATA 229,96,2,12,32
GO 8560 DATA 192,252,242,207,47
KP 8700 RETURN
MH 9000 REM P/M INITIALIZATION ROUTINE
IF 9005 REM PMBASE {SET GRAPHICS MODE
        {12 SPACES}FIRST}
PA 9020 I=PEEK(106)-32
HL 9030 POKE 54279,I
ML 9045 REM TURN ON PLAYERS & MISSILES
GE 9050 POKE 53277,3
EG 9056 REM SET PRIORITY
PN 9058 POKE 623,1
EK 9060 REM PLAYER & MISSILE ADDRESS
        {12 SPACES}TABLE
MP 9100 PMBASE=I*256
KA 9110 MISSILE=PMBASE+384
HB 9120 J=INT(MISSILE/256)
JJ 9130 POKE 1560,(MISSILE-J*256)
EA 9140 POKE 1561,J
IE 9150 PLAYER0=PMBASE+512
FM 9160 J=INT(PLAYER0/256)
IF 9170 POKE 1552,(PLAYER0-J*256)
EF 9180 POKE 1553,J
IL 9190 PLAYER1=PMBASE+640
FI 9200 J=INT(PLAYER1/256)
ID 9210 POKE 1554,(PLAYER1-J*256)
EC 9220 POKE 1555,J
JC 9230 PLAYER2=PMBASE+768
FN 9240 J=INT(PLAYER2/256)
IK 9250 POKE 1556,(PLAYER2-J*256)
EI 9260 POKE 1557,J
JJ 9270 PLAYER3=PMBASE+896
GC 9280 J=INT(PLAYER3/256)
JB 9290 POKE 1558,(PLAYER3-J*256)
EF 9300 POKE 1559,J
DN 9400 REM CLEAR P/M AREA
EC 9420 FOR J=PMBASE TO PMBASE+1024
KF 9430 POKE J,0
FK 9440 NEXT J
GI 9450 REM START VBLANK ROUTINE
GN 9590 A=USR(1615)
KP 9600 RETURN
```

```
HL 10000 DATA 50,48,48,50,49,48,46,48,49,50,48
         ,48,50
GH 12000 REM INSTALL DLI
NG 12010 DL=PEEK(560)+256*PEEK(561)
AP 12020 POKE DL+59,141
MP 12030 RESTORE 13000:I=30464
FM 12040 READ A:IF A=-1 THEN GOTO 12080
NN 12050 POKE I,A
CE 12060 I=I+1
CK 12070 GOTO 12040
OH 12080 POKE 512,0:POKE 513,119
NM 12090 RETURN
EP 13000 DATA 72,138,72,152,72,169,134,141,10,
         212,141,24,208,104,168,104,170,104,64
IH 13999 DATA -1
MF 14000 REM TUNE SUBROUTINE
MD 14010 SOUND 0,0,0,0
BK 14015 RESTORE 14060
MD 14020 READ A,JIFFIES:IF A=-1 THEN FOR I=0 T
         O 3:SOUND I,0,0,0:NEXT I:RETURN
BP 14030 SOUND 0,A,10,8:SOUND 1,A+1,10,8:SOUND
          2,A+2,10,8:SOUND 3,A+3,10,8
KO 14040 GOSUB WAIT
CK 14050 GOTO 14020
ND 14060 DATA 180,3,145,3,121,3,90,15,121,3,90
         ,25,-1,-1
BN 15000 REM ENDTUNE SUBROUTINE
ME 15010 SOUND 0,0,0,0
BM 15015 RESTORE 15060
ME 15020 READ A,JIFFIES:IF A=-1 THEN FOR I=0 T
         O 3:SOUND I,0,0,0:NEXT I:RETURN
CJ 15025 IF A=0 THEN FOR I=0 TO 3:SOUND I,0,0,
         0:NEXT I:GOTO 15020
CA 15030 SOUND 0,A,10,8:SOUND 1,A+1,10,8:SOUND
          2,A+2,10,8:SOUND 3,A+3,10,8
KP 15040 GOSUB WAIT
CM 15050 GOTO 15020
OO 15060 DATA 200,25,0,1,200,20,0,1,200,3,0,1,
         200,20,0,1,170,15,0,1,180,10,0,1
DL 15070 DATA 180,10,0,1,200,7,0,1,200,10,0,1,
         200,10,0,1,200,50,-1,-1
CK 16000 FOR I=0 TO 4:POKE 40847+I,PEEK(40807+
         I):NEXT I
DF 16010 R=R+1
NE 16015 IF R>9 THEN POP :GOTO WIN
DK 16020 POKE 40860,R+16
NK 16030 RETURN
JA 17000 REM WIN ROUTINE
CC 17010 POKE 1562,1:GRAPHICS 18:FOR X=0 TO 3:
         POKE 53248+X,0:NEXT X
```

```
BP 17020 POSITION 2,1:? #6;"YOU HAVE BEATEN"
CK 17030 POSITION 2,2:? #6;"THE DEMONS BACK!"
PF 17040 POSITION 2,3:? #6;"YOU GET 5000"
BM 17050 POSITION 2,4:? #6;"BONUS POINTS!!!"
CD 17060 SCORE=SCORE+5000
FD 17070 POSITION 2,6:? #6;"FINAL SCORE=";SCOR
         E
LH 17080 POSITION 2,8:? #6;"press start"
PH 17090 POSITION 2,9:? #6;"to play again"
PJ 17091 GOSUB WINSOUND
QA 17100 IF PEEK(53279)<>6 THEN 17100
NJ 17110 ROUND=50:POKE 559,0:SCORE=0:POKE RNDO
         VR,0:R=1:GOSUB PMINIT:GOTO 110
JK 18000 REM WINSOUND ROUTINE
BO 18020 RESTORE 18060
MI 18030 READ A,JIFFIES:IF A=-1 THEN FOR I=0 T
         O 3:SOUND I,0,0,0:NEXT I:RETURN
CI 18035 SOUND 0,A,10,8:SOUND 1,A+1,10,8:SOUND
          2,A+2,10,8:SOUND 3,A+3,10,8
LC 18040 GOSUB WAIT
DD 18050 GOTO 18030
JC 18060 DATA 200,30,180,30,160,30,150,30,133,
         30,118,30,105,35,100,70,-1,-1
AM 19999 REM MACHINE LANGUAGE DATA
FB 20000 DATA 104,169,0,133,77,174,22,120,32,8
         0,103,173,120,2,201,5,240,27,201,6,24
         0,23,201,7,240,19,201,9,240
BC 20010 DATA 24,201,10,240,20,201,11,240,16,1
         69,0,141,10,120,240,14,169,1,141,10,1
         20,169,0,240,5,169,255,141
MK 20020 DATA 10,120,173,120,2,201,5,240,27,20
         1,9,240,23,201,13,240,19,201,6,240,24
         ,201,10,240,20,201,14,240
PI 20030 DATA 16,169,0,141,14,120,240,14,169,1
         ,141,14,120,169,0,240,5,169,255,141,1
         4,120,173,2,120,24,109,10
DD 20040 DATA 120,141,2,120,201,48,208,5,169,4
         9,141,2,120,201,200,208,5,169,199,141
         ,2,120,173,6,120,24,109,14
ND 20050 DATA 120,141,6,120,201,25,208,5,169,2
         6,141,6,120,201,95,208,5,169,94,141,6
         ,120,173,2,120,141,0,208
AE 20060 DATA 173,6,120,141,0,6,162,1,32,52,10
         1,232,224,4,208,248,173,23,120,208,11
         ,173,12,208,240,49,32,192
EE 20070 DATA 101,76,244,100,173,12,208,208,20
         ,169,0,141,23,120,141,0,210,141,1,210
         ,141,2,210,141,3,210,76,244
```

```
CA 20080 DATA 100,169,121,141,0,210,169,168,14
         1,1,210,141,3,210,169,122,141,2,210,1
         69,0,141,21,120,165,20,205
GG 20090 DATA 26,120,240,8,141,26,120,169,1,14
         1,30,208,162,1,32,39,102,173,21,120,2
         08,14,232,224,4,208,243,173
LD 20100 DATA 20,120,24,105,1,141,20,120,32,13
         4,102,169,0,141,20,120,32,89,103,173,
         25,120,208,3,76,1,100,96
PK 20110 DATA 189,2,120,24,125,10,120,157,2,12
         0,189,6,120,24,125,14,120,157,6,120,1
         69,0,141,24,120,189,2,120
CO 20120 DATA 201,49,208,17,169,1,141,24,120,1
         57,10,120,32,72,102,173,19,120,157,14
         ,120,189,2,120,201,200,208
BF 20130 DATA 19,169,1,141,24,120,169,255,157,
         10,120,32,72,102,173,19,120,157,14,12
         0,189,6,120,201,24,208,19
OL 20140 DATA 169,1,157,14,120,173,24,120,208,
         9,32,103,102,173,18,120,157,10,120,18
         9,6,120,201,94,48,19,169
AF 20150 DATA 255,157,14,120,173,24,120,208,9,
         32,103,102,173,18,120,157,10,120,189,
         2,120,157,0,208,189,6,120
DH 20160 DATA 157,0,6,96,173,12,208,201,2,208,
         9,162,1,32,244,101,169,0,240,21,173,1
         2,208,201,4,208,9,162,2,32
PI 20170 DATA 244,101,169,0,240,5,162,3,32,244
         ,101,173,20,120,24,105,4,141,20,120,1
         69,1,141,23,120,96,189,10
AN 20180 DATA 120,201,1,208,9,169,255,157,10,1
         20,169,0,240,9,201,255,208,5,169,1,15
         7,10,120,189,14,120,201,1
AO 20190 DATA 208,9,169,255,157,14,120,169,0,2
         40,9,201,255,208,5,169,1,157,14,120,9
         6,189,2,120,201,89,48,25
PO 20200 DATA 201,159,16,21,189,6,120,201,62,4
         8,14,173,20,120,56,233,3,141,20,120,1
         69,1,141,21,120,96,173,10
MJ 20210 DATA 210,41,3,201,0,240,10,201,1,240,
         12,169,1,141,19,120,96,169,255,141,19
         ,120,96,169,0,141,19,120
GC 20220 DATA 96,173,10,210,41,3,201,0,240,10,
         201,1,240,12,169,1,141,18,120,96,169,
         255,141,18,120,96,169,0,141
IG 20230 DATA 18,120,96,173,20,120,201,0,48,85
         ,24,109,107,159,201,26,16,4,141,107,1
         59,96,56,233,10,141,107,159
```

```
CE 20240 DATA 173,106,159,24,105,1,201,26,16,4
         ,141,106,159,96,169,16,141,106,159,17
         3,105,159,24,105,1,201,26
AE 20250 DATA 16,4,141,105,159,96,169,16,141,1
         05,159,173,104,159,24,105,1,201,26,16
         ,4,141,104,159,96,169,16
JC 20260 DATA 141,104,159,173,103,159,24,105,1
         ,141,103,159,96,24,109,107,159,201,16
         ,48,4,141,107,159,96,24,105
IO 20270 DATA 10,141,107,159,173,106,159,56,23
         3,1,201,16,48,4,141,106,159,96,24,105
         ,10,141,106,159,173,105,159
BN 20280 DATA 56,233,1,201,16,48,4,141,105,159
         ,96,24,105,10,141,105,159,173,104,159
         ,56,233,1,201,16,48,4,141
FO 20290 DATA 104,159,96,24,105,10,141,104,159
         ,173,103,159,56,233,1,201,16,48,4,141
         ,103,159,96,169,16,141,103
DO 20300 DATA 159,141,104,159,141,105,159,141,
         106,159,141,107,159,96,160,0,136,208,
         253,202,208,248,96,169,30
JD 20310 DATA 56,229,19,208,6,169,1,141,25,120
         ,96,201,10,16,12,24,105,16,141,127,15
         9,169,16,141,126,159,96,162
GL 20320 DATA 0,56,232,233,10,201,10,16,248,24
         ,105,16,141,127,159,138,24,105,16,141
         ,126,159,96
IF 20999 DATA -1
```

Hopeful Journey

Greg Furness

Maneuver your spacecraft through increasingly more difficult obstacles as you try to complete your mission. Requires a joystick.

Your mission is to maneuver your spaceship through ten increasingly difficult waves of wandering, deadly defense matter. The defense matter is part of disarrayed defense lines which were attacked by enemy warships. The attack left the lines perforated and broken up into the floating matter. Your ship is radioed, while out on a mission, to tell you of the attack and your federation's dire need of you back on the planet. Hence, your "Hopeful Journey" awaits.

The Journey

This is an easy game to play. Just use your joystick to move down through or around the deadly matter to the gate at the bottom of the screen. The game is over when you either touch some of the defense matter or complete wave 10 and win. In either case you'll receive a final score. Wave 10 is solidly filled with defense matter and is extremely difficult to penetrate. The track at the bottom of each screen is the transporter bar which beams you to the next wave.

Hopeful Journey

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
MC 10  DIM A$(60),B$(8),C$(8),D$(8),E$(1)
GB 20  A$(1,60)="hhhJH{K}■{B}■{H}{Q}■■hJH{
       {K}■{BELL}■■{Q}■{2 H}■hJH{G}■■{M}
       {,}■hJ{G}■■{M}{,}■{2 ,}"
AD 30  B$="Th■ ■Nd■"
DI 40  C$="Yo■ ■In■"
ID 50  GOSUB 590
HM 60  GOSUB 420
CO 70  Q=6:L=1:GOTO 390
HL 80  GOSUB 210
BA 90  J=TIME:POKE 559,34:POKE 559,62:POKE 5327
       8,0:POKE 204,A+4:POKE 203,0:POKE 205,120
AC 95  POKE 53278,0
HM 100 REM **MAIN LOOP**
CM 110 Z=USR(ADR(A$),STICK(0))
NA 120 A=INT(RND(0)*19):B=INT(RND(0)*0):C=INT(
       RND(0)*2)
CO 130 Z=USR(ADR(A$),STICK(0))
```

```
GA 140  IF C=0 THEN E$=CHR$(32)
GD 150  IF C=1 THEN E$=CHR$(42)
CE 160  POSITION A,B:? #6;E$:SOUND 0,A+B,10,15:
        SOUND 0,0,0,0
DC 170  Z=USR(ADR(A$),STICK(0))
AO 180  IF PEEK(53252)<>0 THEN 330
GF 190  GOTO 110
OG 200  REM **DRAW PLAYFIELD**
HG 210  GRAPHICS E:POKE 559,0:POKE 756,MEM/256:
        POKE 704,15:POKE 708,196:POKE 710,66
MA 220  IF L=10 THEN POKE 710,100
KA 230  FOR Y=0 TO V STEP Q:FOR X=0 TO 19:POSIT
        ION X,Y:? #6;"*";:NEXT X:NEXT Y
JG 240  FOR X=0 TO 19:POSITION X,O:? #6;"■":NEX
        T X
KO 250  REM **SET-UP PLAYER**
KA 260  A=PEEK(106)-8:POKE 54279,A:PMBASE=A*256
MO 270  FOR I=PMBASE+1024 TO PMBASE+1280:POKE I
        ,0:NEXT I:POKE 53277,2:POKE 53248,120
LO 280  RESTORE 290:FOR I=PMBASE+1024+20 TO PMB
        ASE+1280:READ H:IF H<>0 THEN POKE I,H:N
        EXT I
PN 290  DATA 28,62,107,62,28,0
AK 300  REM **TIME KEEPER**
EC 310  TIME=(PEEK(20)+PEEK(19)*256+PEEK(18)*25
        6*256)/60:RETURN
CL 320  REM **CHECK COLLISION**
FO 330  POKE 53277,0:IF PEEK(53252)=1 THEN E=1:
        I=255:O=5:D$=B$:GOTO 510
JF 340  REM **NEXT WAVE**
BP 350  FOR T=1 TO 15:FOR R=64 TO 78:SOUND 0,R,
        10,10:POKE 710,R:NEXT R:NEXT T:SOUND 0,
        0,0,0
FF 360  IF Q>2 THEN Q=Q-1
GM 370  L=L+1:IF L=10 THEN Q=1
MG 380  IF L=11 THEN E=255:I=1:O=-1:D$=C$:GOTO
        510
DH 390  GRAPHICS 17:POKE 756,MEM/256
PE 400  POSITION 7,9:? #6;"WAVE ";L:FOR T=1 TO
        5:FOR R=0 TO 255 STEP 5:POKE 708,R:NEXT
         R:NEXT T:GOTO 80
BI 410  REM **SKILL LEVEL**
LK 420  GRAPHICS 0:POKE 710,0:POKE 752,1:POKE 7
        56,MEM/256
PN 430  OPEN #1,12,0,"K:"
KM 440  POSITION 3,10:? "WOULD YOU LIKE TO PLAY
         SKILL LEVEL":POSITION 17,11:? "1 OR 2
        ?"
DK 450  GET #1,A
JL 460  IF A=ASC("1") THEN V=9:O=10:E=18:GOTO 4
        90
```

```
MJ 470 IF A=ASC("2") THEN V=21:O=22:E=17:GOTO
       490
GO 480 GOTO 450
IB 490 CLOSE #1:RETURN
AI 500 REM **FINISH ROUTINE**
KL 510 FOR T=1 TO 3:FOR R=E TO I STEP O:SOUND
       0,R,10,10:NEXT R:NEXT T:SOUND 0,0,0,0
CI 520 GOSUB 310:K=TIME:T=INT(K-J):SCORE=L*100
       0-T*10:IF SCORE>HI THEN HI=SCORE
DD 530 GRAPHICS 17:POKE 756,MEM/256
OB 540 POSITION 4,3:? #6;"HI-SCORE ";HI:POSITI
       ON 5,5:? #6;"SCORE ";SCORE
EM 550 POSITION 6,10:? #6;D$:POSITION 4,17:? #
       6;"PRESS start":POSITION 3,19:? #6;"TO
       PLAY AGAIN"
CE 560 IF PEEK(53279)<>6 THEN 560
DL 570 GOTO 60
DD 580 REM **CHAR SET**
FC 590 POKE 106,PEEK(106)-32:GRAPHICS 0:POKE 5
       59,0
GM 600 MEM=(PEEK(106)+28)*256
ME 610 FOR SET=0 TO 1023:POKE MEM+SET,PEEK(573
       44+SET):NEXT SET
EA 620 POKE 756,MEM/256
MA 630 RESTORE 640:FOR SET=0 TO 7:READ CHAR:PO
       KE 80+MEM+SET,CHAR:NEXT SET
JG 640 DATA 24,36,66,129,129,66,36,24
MG 650 REM **TITLE PAGE**
DK 660 GRAPHICS 17:POKE 559,34:POKE 708,0:POKE
        756,MEM/256
DJ 670 POSITION 3,8:? #6;"HOPEFUL JOURNEY"
AN 680 POSITION 3,13:? #6;"BY GREG FURNESS"
FJ 690 FOR R=2 TO 15:POKE 708,R:FOR T=1 TO 50:
       NEXT T:NEXT R
AD 700 FOR R=15 TO 0 STEP -1:POKE 708,R:FOR T=
       1 TO 50:NEXT T:NEXT R:RETURN
```

Hoppo

Ron Szabo

Can you stay a safe distance from Spike and Grimo while trying to restore power to the islands of Zabonia? Your pursuers are fast and never get tired of the chase. "Hoppo" is a game for one player with a joystick.

It's the twenty-third century and you're in the land of Zabonia. You are Hoppo, the champion of the Zabonians. The evil Grimo has destroyed the world's power plants. Since Zabonia is a world of islands, your mission is to hop from island to island and restore partial power to 54 of them. If you succeed, you will be transported to another part of the world to provide partial power for 54 other islands, but your task is not as easy as it seems. You must avoid the evil Grimo and his treacherous counterpart, Spike. When you reach the third level, you must also avoid the opening vacuum pits. If you survive this level, you'll receive a bonus life.

Island Hoping

"Hoppo" is a game for one player. To play, plug a joystick into port 1. Run the program and wait for about one minute while the program initializes (be sure to save a copy before trying to run it). When the animated title screen appears, you may select the speed and the level at which you wish to begin. To select the speed, move the joystick to the left. To select the level, move the joystick to the right. When you're ready to play, press the fire button.

You begin on the lower-right island. You can move right and left, up and down, but not diagonally.

There are four ways in which you can lose a life. The first way is by contacting Grimo. He'll push you off the island you're sharing. Contacting Spike is just as bad. He causes Hoppo to cry out. The third way is by falling through one of the vacuum pits. They will suck Hoppo into another dimension. Finally, don't let Hoppo fall off the edge of the world. You will fall into the twilight zone.

You receive 10 points for every island you restore power to. If power is restored to 54 islands, you receive a bonus of 100*L, where L is the level completed, and you then advance to the next level. You start out with three lives and gain a bonus life after the third level.

Hoppo

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
ME 10  DIM SPD$(4):GOSUB 1000:GOSUB 4000
MA 20  POKE 77,0:S=PEEK(632):IF S=15 THEN GOTO
       30
GG 22  IF S=11 THEN X=X-20:GOTO 30
DL 24  IF S=7 THEN X=X+20:GOTO 30
HE 26  IF S=14 THEN Y=Y-16:GOTO 30
JN 28  IF S=13 THEN Y=Y+16
EI 30  POKE PLX+1,X:POKE PLY+1,Y:SOUND 0,120,10
       ,8
DG 31  IF Y>164 OR X>172 OR X<72 OR Y<36 THEN G
       OTO 920
NO 35  IF X1=X THEN S=1:GOTO 42
OE 37  IF X1>X THEN S=4:GOTO 42
DD 39  S=7
KJ 42  IF Y1=Y THEN 50
FO 44  IF Y1>Y THEN S=S+1:GOTO 50
KK 46  S=S+2
HN 50  ON S GOTO 900,51,52,53,150,150,54,150,15
       0
CE 51  Y1=Y1-16:GOTO 60
CD 52  Y1=Y1+16:GOTO 60
BP 53  X1=X1-20:GOTO 60
BO 54  X1=X1+20:GOTO 60
PI 60  POKE PLX,X1:POKE PLY,Y1:SOUND 0,240,10,8
AI 61  IF X1=X AND Y1=Y THEN GOTO 900
EN 65  PC=((Y-28)*5)+((X-48)/4)
PJ 67  IF L>2 THEN GOSUB 200
IO 72  IF PEEK(SC+PC)=99 THEN POKE SC+PC,225:SC
       R=SCR+10:POSITION 9,20:? SCR;:LET COM=CO
       M+1:IF COM=54 THEN GOTO 500
FG 74  IF D<1 OR D>5 THEN X2=((INT(RND(0)*6)*20
       )+72):Y2=((INT(RND(0)*9)*16)+36):POKE PL
       X+2,X2:POKE PLY+2,Y2:D=0
GP 76  D=D+1:IF X2=X AND Y2=Y THEN GOTO 940
JK 78  IF PEEK(SC+PC)=101 THEN GOTO 920
CO 80  SOUND 0,0,0,0:FOR I=1 TO SPD:NEXT I:GOTO
        20
LJ 150  I=INT(RND(0)*2):IF (S=5 OR S=6) AND I=1
         THEN GOTO 53
AE 151  IF (S=8 OR S=9) AND I=1 THEN GOTO 54
ME 152  IF (S=5 OR S=8) AND I=0 THEN 51
DJ 153  GOTO 52
GL 200  IF D>0 AND D<6 THEN RETURN
BA 202  LD=L-1:IF LD>9 THEN LD=9
HB 204  FOR I=1 TO LD:TP(I)=((INT(RND(0)*6)*5)+
       6)+((INT(RND(0)*9)*80)+40)
```

```
MJ 206 POKE SC+TP(I),101:POKE SC+TP(I)+1,102:N
       EXT I
FI 208 IF TP(LD+1)=0 AND D=0 THEN GOSUB 220:RE
       TURN
CD 210 FOR I=1 TO LD:POKE SC+TP(LD+I),99:POKE
       SC+TP(LD+I)+1,100:NEXT I
OP 220 FOR I=1 TO LD:TP(LD+I)=TP(I):NEXT I:RET
       URN
EH 500 SOUND 0,0,0,0:BONUS=L*100:POSITION 13,2
       1:? "YOU DID IT !":POSITION 14,22:? "BO
       NUS:";BONUS;:POSITION 9,20
FB 502 SCR=SCR+BONUS:? SCR;:FOR I=1 TO 200:SOU
       ND 0,RND(0)*40+20,10,8:NEXT I
IK 504 SOUND 0,0,0,0:L=L+1:LET COM=0:D=0:IF L=
       4 AND LVL<>3 THEN LIVES=LIVES+1:POSITIO
       N 13,23:? "EXTRA LIFE !";
CB 505 IF L=4 AND LVL<>3 THEN FOR I=1 TO 100:S
       OUND 0,RND(0)*40,10,8:NEXT I:SOUND 0,0,
       0,0
CO 506 IF L>10 THEN L=10
MH 508 GOSUB 780:GOTO 800
IP 700 FOR I=1 TO 50:NEXT I:LIVES=LIVES-1:IF L
       IVES=0 THEN POSITION 14,21:? "GAME OVER
       ";:SOUND 0,0,0,0:GOTO 750
PO 702 POSITION 26,20:? "LIVES=";:GOSUB 780:?
       N$:X=172:Y=164:X1=72:Y1=36:GOSUB 760
MP 703 FOR I=1 TO 20:TP(I)=0:NEXT I:D=0
NI 704 POKE PLX,X1:POKE PLY,Y1:POKE PLX+1,X:PO
       KE PLY+1,Y:GOTO 20
EG 750 POSITION 26,20:? "LIVES={4 SPACES}";:PO
       SITION 8,22:? "TO PLAY AGAIN,PRESS FIRE
       ";
MJ 751 IF PEEK(644)=0 THEN GOSUB 8015:POKE 559
       ,0:SCR=0:L=LVL:LIVES=3:D=0:LET COM=0:N$
       ="b b b":POKE PLX+2,0:GOTO 800
HG 754 GOTO 751
IJ 760 IF TP(LD+1)=0 THEN RETURN
DB 762 FOR I=1 TO LD:POKE SC+TP(LD+I),99
GF 764 POKE SC+TP(LD+I)+1,100:NEXT I
ID 766 RETURN
AC 780 ON LIVES GOTO 781,782,783,784
AP 781 N$="b{6 SPACES}":RETURN
HC 782 N$="b b{4 SPACES}":RETURN
NF 783 N$="b b b   ":RETURN
DI 784 N$="b b b b":RETURN
PN 800 GRAPHICS 0:SETCOLOR 2,0,0:POKE 752,1:PO
       KE PLX,0:POKE PLX+1,0:POKE PLX+2,0:DL=P
       EEK(560)+256*PEEK(561):POKE DL+3,68
MN 801 FOR I=0 TO 18:POKE DL+6+I,4:NEXT I:SETC
       OLOR 1,9,4
```

```
KI 802 POKE 756,CADR/256:X=172:Y=164:X1=72:Y1=
       36:FOR I=1 TO 20:TP(I)=0:NEXT I
PG 804 ? :FOR I=1 TO 9:? "{4 SPACES}cd
       {3 SPACES}cd{3 SPACES}cd{3 SPACES}cd
       {3 SPACES}cd{3 SPACES}cd"
JP 806 ? :NEXT I
LE 808 POSITION 3,20:? "SCORE=";SCR;:POSITION
       26,20:? "LIVES=";N$;
CL 810 SC=PEEK(88)+256*PEEK(89)
ED 812 POKE 559,62:POKE PLX,72:POKE PLY,52:POK
       E PLX+1,172:POKE PLY+1,132:POKE 752,1
OC 814 POKE 559,62:IF SRT=1 THEN SRT=0:RETURN
DK 816 GOTO 20
AC 900 POKE PLY+1,Y+8:FOR I=1 TO 5:FOR OUCH=1
       TO 15:POKE PLY,Y1-3:SOUND 0,255,8,15:FO
       R D=1 TO 2:NEXT D
BB 902 SOUND 1,40-OUCH,10,8:POKE PLY,Y1:FOR D=
       1 TO 2:NEXT D:SOUND 0,0,0,0:POKE 705,64
       +OUCH:NEXT OUCH:NEXT I
PB 904 Y=Y+16:POKE 705,202:GOTO 920
CN 920 FOR I=Y TO 244 STEP 16:POKE PLY+1,I:FOR
        D=0 TO 4:SOUND 0,I-(4*D),10,8:NEXT D:N
       EXT I:SOUND 0,0,0,0:POKE PLX+1,0
GM 922 DUR=6:PIT=20:SOUND 2,75,8,15:ICR=0.79+D
       UR/100:V1=15:V2=15:V3=15
PB 924 SOUND 0,PIT,8,V1:SOUND 1,PIT+20,8,V2:SO
       UND 2,PIT+50,8,V3
CE 926 V1=V1*ICR:V2=V2*(ICR+0.05):V3=V3*(ICR+0
       .08):IF V3>1 THEN 924
JI 928 FOR I=0 TO 2:SOUND I,0,0,0:NEXT I:IF Z=
       1 THEN Z=0:RETURN
GM 930 GOTO 700
MA 940 POKE PLX+2,0:FOR I=1 TO 10:FOR OUCH=0 T
       O 15:POKE 705,64+OUCH:SOUND 0,40-OUCH-I
       ,10,8:NEXT OUCH:NEXT I
DL 942 SOUND 0,0,0,0:POKE 705,202:GOTO 700
IC 999 END
LH 1000 PRINT "{CLEAR}{DOWN}INITIALIZING"
CJ 1001 PRINT "{DOWN}PLEASE WAIT..."
OH 1010 FOR I=1536 TO 1706:READ A:POKE I,A:NEX
        T I
EN 1020 FOR I=1774 TO 1787:POKE I,0:NEXT I
NC 1030 PM=PEEK(106)-16:PMBASE=256*PM
PO 1040 FOR I=PMBASE+1023 TO PMBASE+2047:POKE
        I,0:NEXT I
KA 1050 FOR I=PMBASE+1025 TO PMBASE+1030:READ
        A:POKE I,A:NEXT I
LE 1060 FOR I=PMBASE+1281 TO PMBASE+1288:READ
        A:POKE I,A:NEXT I
```

```
LI 1065 FOR I=PMBASE+1537 TO PMBASE+1544:READ
        A:POKE I,A:NEXT I
LF 1070 POKE 704,40:POKE 705,202:POKE 706,255
GP 1080 PLX=53248:PLY=1780:PLL=1784
GH 1090 POKE 623,1:POKE 1788,PM+4:POKE 53277,3
        :POKE 54279,PM
HI 1100 X=USR(1696)
FP 1110 POKE PLL,6:POKE PLL+1,8:POKE PLL+2,8
CK 1120 POKE PLX,0:POKE PLY,0:POKE PLX+2,0:POK
        E PLY+2,0
LG 1130 POKE PLX+1,0:POKE PLY+1,0
KG 1140 RETURN
PI 2000 REM *  VBLANK ROUTINE DATA  *
GP 2010 DATA 162,3,189,244,6,240,89,56,221,240
        ,6,240,83,141,254,6,106,141
DG 2020 DATA 255,6,142,253,6,24,169,0,109,253,
        6,24,109,252,6,133,204,133
EC 2030 DATA 206,189,240,6,133,203,173,254,6,1
        33,205,189,248,6,170,232,46,255
ED 2040 DATA 6,144,16,168,177,203,145,205,169,
        0,145,203,136,202,208,244,76,87
PE 2050 DATA 6,160,0,177,203,145,205,169,0,145
        ,203,200,202,208,244,174,253,6
KM 2060 DATA 173,254,6,157,240,6,189,236,6,240
        ,48,133,203,24,138,141,253,6
NE 2070 DATA 109,235,6,133,204,24,173,253,6,10
        9,252,6,133,206,189,240,6,133
GL 2080 DATA 205,189,248,6,170,160,0,177,203,1
        45,205,200,202,208,248,174,253,6
CG 2090 DATA 169,0,157,236,6,202,48,3,76,2,6,7
        6,98,228,0,0,104,169
OF 2100 DATA 7,162,6,160,0,32,92,228,96
OC 3000 REM *{3 SPACES}PLAYER DATA{3 SPACES}*
JD 3001 REM * LINE 3010-GRIMO *
JN 3002 REM * LINE 3020-HOPPO *
JF 3003 REM * LINE 3030-SPIKE *
MI 3010 DATA 130,186,214,56,68,56
OP 3020 DATA 108,40,124,84,124,84,68,56
MK 3030 DATA 8,8,8,28,28,62,62,28
KF 4000 POKE 106,(PEEK(106)-5):GRAPHICS 0:POKE
         752,1:SETCOLOR 2,0,0:GOSUB 7000
JN 4010 CADR=256*(PEEK(106)+1)
KC 4020 FOR I=0 TO 1023:POKE CADR+I,PEEK(57344
        +I):NEXT I
KP 4040 FOR I=0 TO 47:READ X:POKE 776+CADR+I,X
        :NEXT I:GOSUB 8000
FK 4060 DATA 15,63,255,255,255,255,63,15,108,4
        0,124,80,124,80,68,56
DD 4070 DATA 10,42,170,170,170,170,42,10,160,1
        68,170,170,170,170,168,160
```

```
IE 4080 DATA 10,40,160,128,128,160,40,10,160,4
        0,10,2,2,10,40,160
GM 5000 SCR=0:LET COM=0:DIM N$(10),TP(20):SRT=
        1:N$="b b b":LIVES=3:GOTO 800
AK 7000 ? :? :? :? :? :? :? "{3 SPACES}H
        {3 SPACES}H  000  PPPP  PPPP
        {3 SPACES}000"
MG 7010 FOR I=1 TO 2:? "{3 SPACES}H{3 SPACES}H
         0{3 SPACES}0 P{3 SPACES}P P
        {3 SPACES}P 0{3 SPACES}0":NEXT I
HA 7020 ? "{3 SPACES}HHHHH 0{3 SPACES}0 PPPP
        PPPP  0{3 SPACES}0"
CI 7030 FOR I=1 TO 2:? "{3 SPACES}H{3 SPACES}H
         0{3 SPACES}0 P{5 SPACES}P{5 SPACES}0
        {3 SPACES}0":NEXT I
FI 7040 ? "{3 SPACES}H{3 SPACES}H  000  P
        {5 SPACES}P{6 SPACES}000"
KM 7050 RETURN
PH 8000 GRAPHICS 0:SETCOLOR 2,0,0:POKE 752,1:D
        L=PEEK(560)+256*PEEK(561):FOR I=20 TO
        22:POKE DL+I,4:NEXT I
JL 8002 GOSUB 7000:POKE 756,CADR/256:SETCOLOR
        1,9,4:? :? :? :? "{4 SPACES}cd
        {3 SPACES}cd{3 SPACES}cd{3 SPACES}cd
        {3 SPACES}cd{3 SPACES}cd"
AA 8004 POKE 559,62:POKE PLY+1,156:POKE PLX+1,
        72:SOUND 0,120,10,8:SC=PEEK(88)+256*PE
        EK(89):POKE SC+646,225
FI 8006 FOR I=1 TO 10:NEXT I:SOUND 0,0,0,0:FOR
         I=0 TO 50:NEXT I:POKE PLX+1,92:SOUND
        0,120,10,8:POKE SC+651,225
JH 8008 FOR I=1 TO 10:NEXT I:POKE PLY,156:POKE
         PLX,72:SOUND 0,240,10,8:FOR I=1 TO 10
        :NEXT I:SOUND 0,0,0,0
HL 8010 FOR I=1 TO 50:NEXT I:POKE PLX+1,112:PO
        KE SC+656,225:SOUND 0,120,10,8:FOR I=1
         TO 10:NEXT I
PK 8012 POKE PLX,92:SOUND 0,240,10,8:FOR I=1 T
        O 10:NEXT I:SOUND 0,0,0,0:FOR I=1 TO 3
        5:NEXT I:POKE SC+661,101
PK 8014 POKE SC+662,102:FOR I=1 TO 15:NEXT I:P
        OKE PLX+1,132:Y=164:Z=1:GOSUB 920
FP 8015 FOR I=20 TO 22:POSITION 0,I:PRINT "
        {34 SPACES}":NEXT I:SPD=1:SPD$="SLOW"
NO 8016 POSITION 4,21:? "SPEED=SLOW";:SPD=1:PO
        SITION 27,21:? "LEVEL=1";:LVL=1:POSITI
        ON 11,22:? "PRESS [RETURN] TO START"
HH 8017 OSS=STICK(0):IF OSS=11 THEN SPD=SPD+1:
        GOSUB 8100
```

```
CG 8018 IF OSS=7 THEN LVL=LVL+1:IF LVL>9 THEN
        LVL=1
KL 8019 FOR I=1 TO 10:NEXT I:IF STRIG(0)=0 THE
        N GOTO 8050
DM 8020 POSITION 10,21:? SPD$;:POSITION 33,21:
        ? LVL:GOTO 8017
FF 8050 ON SPD GOTO 8052,8054,8056
MF 8052 SPD=100:GOTO 8060
JL 8054 SPD=50:GOTO 8060
GI 8056 SPD=0:GOTO 8060
CN 8060 POKE PLX,0:POKE PLX+1,0:L=LVL:RETURN
DE 8100 IF SPD>3 THEN SPD=1
EN 8102 ON SPD GOTO 8104,8106,8108
LI 8104 SPD$="SLOW":RETURN
HJ 8106 SPD$="MED.":RETURN
KF 8108 SPD$="FAST":RETURN
KK 8110 RETURN
AL 10000 GOSUB 5000:GOTO 20
```

Termite

Frank Martone

Using redefined characters, "Termite" is a good example of how graphics and sound can enhance the thrill and excitement of computer games. Requires a joystick.

Tunnel your way through a delicious piece of wood. Munch on knots, spider eggs, and magical goblets. Beware, though, of the deadly spider. This crafty arachnid seeks you out.

Your termite has one powerful weapon against the spider—thought transfer. When in trouble, merely press the joystick button and you will transport yourself to a safe location. Doing this, however, will cost 50 points.

You have only one minute to munch as many tasty objects as you can before advancing to the next screen. When the minute is up, you'll be greeted by a song and a friendly message. You'll start out with three termites; when one is killed, a series of beeps will tell you how many you have left.

Here's the scoring: magical goblets, 300 points, eggs, 100 points, and knots, 10 points. At the end of the game the final score will be given.

"Termite" becomes more difficult as the game progresses.

Termite, the Program

Termite was programmed in graphics 1, using a redefined character set. A redefined character is simply your own customized character. You can change letters and numbers into spaceships or creatures by rearranging the 8 × 8 grid of the character you wish to change, then POKEing into screen memory to display that character:

```
SCR=PEEK(88)+256*PEEK(89)
```

For a more detailed explanation of this technique, look at pages 108–126 of *COMPUTE!'s First Book of Atari Graphics*.

Program explanation:

Lines	Description
4–14	Set up screen display
19–75	Main loop moves termite and spider; uses LOCATE statement to return ATASCII codes to detect collisions
299–316	Termite killed; make explosion, decrement number of termites, reset termite and spider positions
1000–1200	Set up redefined character set

Lines	Description
3000–3008	Title page waits till START is pressed to begin
5000–6000	Game over; print score
7000–7210	When minute is up, play music and print message
10000–10010	Thought transfer: erase previous termite and reposition termite in random spot

Using LOCATE for Collision Detection

The LOCATE statement works differently on graphics modes 0–2 than on modes 3–8. In graphics modes 0–2, the ATASCII code of the character that occupies that space is returned. In modes 3–8, the color of a character is returned. Look at this example:

LOCATE *x,y,what*

where *x* and *y* are the variables of your player's column and row; *what* is the variable in which either the color or the ATASCII code will be stored, depending on the graphics mode you're working in. If *what* equals something other than your character, a collision has been made. Use IF-THEN statements to accomplish this.

The Spider Knows

The termite and the spider both have column and row coordinates. The horizontal coordinate is followed by the vertical. For the termite, they are X and Y; for the spider, they are S1 and S2. Lines 64–67 of the program use IF-THEN statements to compare the positions of the spider and the termite. If the spider's horizontal coordinates are more than the termite's, the horizontal coordinate of the spider must be subtracted. This works the same way on the vertical axis; therefore, the spider's position is updated by your location on the screen.

Termite

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
BD 2 REM TERMITE ATARI 16K
HG 3 GOSUB 3000
CJ 4 GRAPHICS 1+16:SETCOLOR 0,8,10:SETCOLOR 2,
     RND(0)*14,9:TERM=3:SC=0:SL=1:SM=30:KNS=60
     :EG=10:GOB=12
```

```
KF 5  GRAPHICS 1+16:SETCOLOR Ø,8,1Ø:SETCOLOR 2,
      RND(Ø)*14,9:TERM=3:SC=Ø:SL=1:SM=3Ø:KNS=6Ø
      :EG=1Ø:GOB=12:GOSUB 1ØØØ
IK 6  POKE 756,CH/256
BL 7  SCR=PEEK(88)+256*PEEK(89)
IF 8  FOR G=2 TO 22:POSITION Ø,G:? #6;"
      "
EG 9  SOUND Ø,G,1Ø,1Ø:SOUND Ø,Ø,Ø,Ø:NEXT G
DM 11 FOR KNOTS=1 TO KNS:K1=INT(RND(Ø)*19):K2=
      INT(RND(Ø)*17)+3:POSITION K1,K2:? #6;"
      "
      :SOUND Ø,5Ø,1Ø,1Ø
DJ 12 FOR HJ=1 TO 2:NEXT HJ:SOUND Ø,Ø,Ø,Ø:NEXT
       KNOTS
AH 13 FOR EGGS=1 TO EG:E1=INT(RND(Ø)*19):E2=IN
      T(RND(Ø)*17)+3:POSITION E1,E2:? #6;"t":S
      OUND Ø,2ØØ,1Ø,1Ø
AH 14 FOR HJ=1 TO 25:NEXT HJ:SOUND Ø,Ø,Ø,Ø:NEX
      T EGGS
JC 15 FOR C=1 TO GOB:POSITION RND(Ø)*18,RND(Ø)
      *17+3:? #6;"a":SOUND Ø,C,1Ø,1Ø:SOUND Ø,Ø
      ,Ø,Ø:NEXT C
DB 16 X=9:Y=1Ø:S1=19:S2=15:V=1Ø:TIME=Ø
JG 17 POKE SCR+X+2Ø*Y,1Ø:FOR G=1 TO 8Ø:SOUND Ø
      ,G,Ø,1Ø:NEXT G:SOUND Ø,Ø,Ø,Ø
FK 19 REM MAIN LOOP
EG 2Ø SCR=PEEK(88)+256*PEEK(89)
MD 25 TP=SCR+X+2Ø*Y
ID 26 ST=STICK(Ø):TR=STRIG(Ø):SETCOLOR 3,PEEK(
      5377Ø),14
NC 3Ø POKE WS,58:POSITION 3,Ø:? #6;"";SC;"   "
      :POSITION 16,Ø:? #6;"*";TERM
GH 35 SOUND 1,Ø,Ø,Ø:TIME=TIME+1
CM 36 IF TIME>17Ø THEN GOTO 7ØØØ
FJ 37 FOR W=1 TO 1Ø:NEXT W:POKE WS,Ø
FI 38 IF ST=14 THEN POKE TP,Ø:Y=Y-1:V=12
EM 4Ø IF ST=13 THEN POKE TP,Ø:Y=Y+1:V=1Ø
CA 42 IF ST=7 THEN POKE TP,Ø:X=X+1:V=11
KJ 43 IF TR=Ø THEN GOSUB 1ØØØØ
FC 44 IF ST=11 THEN POKE TP,Ø:X=X-1:V=14
OK 47 LOCATE X,Y,BC:IF BC=247 THEN FOR W=15 TO
       Ø STEP -2:SOUND Ø,5Ø,1Ø,W:NEXT W:SC=SC+
      1Ø
EB 48 IF BC=116 THEN POSITION X,Y:? #6;"";:FO
      R R=4Ø TO 1 STEP -1:SOUND Ø,R,2,14:NEXT
      R:SC=SC+1ØØ
AI 49 IF BC=97 THEN POSITION X,Y:? #6;"":FOR
      W=3Ø TO Ø STEP -1:SOUND Ø,W,1Ø,1Ø:NEXT W
      :SC=SC+3ØØ
JB 5Ø POKE SCR+X+2Ø*Y,V
KO 51 IF X>18 THEN POKE SCR+X+2Ø*Y,Ø:X=X-1
```

```
HD 52  IF X<1 THEN POKE SCR+X+20*Y,0:X=X+1
HI 53  IF Y<2 THEN POKE SCR+X+20*Y,0:Y=Y+1
KP 54  IF Y>22 THEN POKE SCR+X+20*Y,0:Y=Y-1
CA 60  WS=SCR+S1+20*S2
DP 61  IF X=S1 AND Y=S2 THEN GOSUB 300
LL 62  POKE WS,185:FOR S=1 TO 10:NEXT S:SOUND 0
       ,0,0,0:IF L=35 THEN SL=2
FI 63  POKE WS,0:L=INT(RND(0)*70):IF L<SM THEN
       GOTO 75
FD 64  IF S1>X THEN S1=S1-SL
FA 65  IF S1<X THEN S1=S1+SL
FJ 66  IF S2>Y THEN S2=S2-SL
FG 67  IF S2<Y THEN S2=S2+SL
BP 69  SOUND 0,90,0,10:SOUND 0,89,0,10
HE 70  SL=1
AH 75  GOTO 20
HH 299 REM TERMITE KILLED
CJ 300 FOR G=14 TO 1 STEP -1:FOR W=1 TO 25:NEX
       T W:SETCOLOR 0,12,G:SOUND 0,G,10,10:SOU
       ND 1,G,10,10:NEXT G
FQ 301 SETCOLOR 0,12,14:SOUND 0,0,0,0:SOUND 1,
       0,0,0
PI 302 FOR EX=4 TO 6:POKE SCR+X+20*Y,EX:FOR W=
       1 TO 70:NEXT W:SOUND 0,100,0,10:NEXT EX
       :POKE SCR+X+20*Y,0:X=9:Y=10
AI 303 FOR S=0 TO 2:SOUND S,0,0,0:NEXT S
DN 304 TERM=TERM-1:IF TERM=0 THEN GOTO 5000
KP 306 FOR L=0 TO 95 STEP 5.9:SOUND 0,L,0,10:S
       OUND 1,L,0,5:SOUND 2,L,8,3:SETCOLOR 4,R
       ND(0)*14,14:NEXT L:SETCOLOR 4,0,0
AD 310 FOR R=0 TO 2:SOUND R,0,0,0:NEXT R
CC 315 S1=15:S2=15
HI 316 FOR FM=1 TO TERM:SOUND 0,50,10,10:FOR L
       =1 TO 50:NEXT L:SOUND 0,0,0,0:FOR L=1 T
       O 50:NEXT L:NEXT FM:RETURN
EJ 1000 REM CH-SET
EL 1010 CH=(PEEK(106)-8)*256:FOR I=0 TO 7:POKE
        CH+I,0:NEXT I
OG 1015 FOR I=0 TO 512:POKE CH+I,PEEK(57344+I)
        :NEXT I
KJ 1020 RESTORE 1100
KG 1030 READ A:IF A<0 THEN RETURN
DN 1040 FOR J=0 TO 7:READ B:POKE CH+A*8+J,B:NE
        XT J
MD 1050 GOTO 1030
DN 1099 REM REDEFINED CHARACTERS
LD 1100 DATA 10,56,124,254,254,84,56,84,146
DK 1110 DATA 11,124,126,251,255,126,124,84,146
LL 1115 DATA 12,56,84,254,254,124,56,84,146
NN 1117 DATA 14,124,126,223,255,127,52,42,73
```

```
GP 1120 DATA 4,0,0,4,0,16,8,64,0
EL 1125 DATA 5,1,66,0,34,8,64,20,128
GL 1130 DATA 6,0,0,4,80,8,20,0,0
MC 1135 DATA 58,68,130,84,56,124,186,68,130
EO 1140 DATA 2,255,253,239,247,255,191,251,255
MB 1143 DATA 58,68,130,84,56,124,186,68,130
CE 1145 DATA 57,0,40,84,186,124,56,68,40
IL 1146 DATA 55,24,36,90,165,165,153,66,60
LH 1148 DATA 52,0,56,124,254,239,218,124,56
PE 1149 DATA 51,7,5,7,56,40,184,128,128
JL 1150 DATA 33,65,65,99,62,28,8,8,28
CH 1155 DATA 34,7,5,7,248,104,248,64,192
DL 1200 DATA -1
HB 3000 GRAPHICS 2+16:POKE 708,14
OI 3001 ? #6:? #6;"{6 SPACES}TERMITE!   "
CA 3002 ? #6:? #6;"{18 SPACES}":? #6:SETCOLOR 4
        ,8,7
HF 3006 ? #6:? #6;"PRESS start TO BEGIN"
OJ 3007 IF PEEK(53279)=6 THEN GOTO 5
PH 3008 POKE 709,RND(0)*14:GOTO 3007
GJ 5000 FOR E=14 TO 0 STEP -1:SETCOLOR 2,4,E:F
        OR R=1 TO 25:NEXT R:SOUND 0,E,0,10:NEX
        T E
PE 5010 GRAPHICS 2+16:? #6:? #6;"{6 SPACES}g■■
        e over "
HG 5050 FOR L=20 TO 70:SOUND 0,L,10,10:SOUND 1
        ,L,10,10:NEXT L
LF 5060 ? #6:? #6;"{3 SPACES}your score is: "
KB 5061 SOUND 0,0,0,0:SOUND 1,0,0,0
FA 5062 ? #6:POSITION 8,5:? #6;SC
ON 5065 FOR W=1 TO 50:NEXT W:POSITION 8,5:? #6
        ;"{12 SPACES}"
FA 5070 FOR W=1 TO 50:NEXT W
KG 5075 ? #6:? #6;" PRESS start TO PLAY":? #6;
        "  OR select TO QUIT"
BN 5080 IF PEEK(53279)=6 THEN GRAPHICS 1+16:TE
        RM=3:GOTO 6
FJ 5085 IF PEEK(53279)=5 THEN END
MM 6000 GOTO 5062
KM 7000 REM MUSIC
JE 7001 SOUND 0,255,10,10:SOUND 1,254,10,10:FO
        R L=1 TO 100:SETCOLOR 2,RND(0)*50,14:N
        EXT L
LF 7002 RESTORE 7100
MC 7005 READ MUSIC:POKE 712,MUSIC:POKE 710,PEE
        K(53770)
OI 7006 IF MUSIC=255 THEN GOTO 7120
EI 7010 SOUND 0,MUSIC,10,10
EC 7020 SOUND 1,MUSIC-1,10,10:FOR G=1 TO 10:NE
        XT G
```

```
NB 7050 GOTO 7005
PN 7100 DATA 121,96,96,121,91,91,81,81,81,91,1
        08,121,144,128,144,121,108,121,72,64,7
        2,91,81,53,47,45,53,33,35,40
GD 7110 DATA 45,47,53,60,60,255
CH 7120 GRAPHICS 2+16:? #6:? #6:? #6:? #6
AN 7125 JJ=INT(RND(1)*6)+1:SOUND 0,0,0,0:SOUND
         1,0,0,0
KL 7126 IF JJ=1 THEN ? #6;"{3 SPACES}TERMITE P
        OWER!":GOTO 7200
FJ 7128 IF JJ=2 THEN ? #6;" CLAW LICK'IN GOOD!
        !  ":GOTO 7200
LP 7130 IF JJ=3 THEN ? #6;"{5 SPACES}SUPERMITE
         !":GOTO 7200
BN 7131 IF JJ=4 THEN ? #6;"{5 SPACES}GO FOR IT
        !{8 SPACES}":GOTO 7200
FG 7132 IF JJ=5 THEN ? #6;" TOTALY AWESOME!!!!
          ":GOTO 7200
HC 7200 FOR W=1 TO 50:SETCOLOR 0,PEEK(53770),P
        EEK(53770):NEXT W
KA 7205 IF KNS<20 THEN KNS=10
IF 7207 EG=EG+5
LD 7210 KNS=KNS-5:SM=SM-10:GRAPHICS 1+16:POKE
        756,CH/256:SETCOLOR 2,RND(1)*14,9:SETC
        OLOR 0,8,10:GOB=GOB+2:GOTO 7
HA 9999 REM THOUGHT-TRANSFER
FH 10000 FOR R=-20 TO 20:SOUND 0,ABS(R),6,10:P
         OKE 712,ABS(R):NEXT R:POKE TP,0:X=INT
         (RND(0)*20):Y=INT(RND(0)*20)
FA 10005 SC=SC-50:POKE 712,0
NC 10010 RETURN
```

Chapter 5

Strategy Games

Teaching Concepts

Vilson J. Leffa

Learning should be an enjoyable, nonthreatening experience. That's the idea behind this game for children and adults.

"Teaching Concepts" was designed to teach and entertain. It can teach language-related skills, from spelling to concept formation, both to young readers and to students who are learning English as a second language. And it can also entertain. The student-players are challenged to use their wits to get to the secret word. Graphics and sound, although economical of space, combine with text to add to the learning—and entertaining—environment.

One, Two, or Three Players

The program, by allowing up to three players working on different options at the same time, is a cross between group work and individualized instruction. Esthetically, the screen may look crowded sometimes, but the advantage is obvious: All the needed information is always displayed. A quick glance can also tell the teacher or parent how much work has been done and how well each player is doing.

The program listed here is ready to run, but should be regarded as a demonstration only. This is, in a way, a utility program which can be used to meet specific needs. Expansion and adaptations can be easily done by entering the language items in the DATA lines (see section on expansion).

Running the Program

From the player's point of view, the program should be self-explanatory, although young children doing it for the first time may need some guidance.

It starts by asking the number of players, their names, and their individual options from the menu. Then it displays the names and, if no changes are required, a horizontal arrow points to the current player.

If the current player, using the hints displayed on the screen, is unable to guess the word the computer is "thinking of," help can be requested by pressing the ESC key. Help may be free or cost some points, depending on certain conditions, such as the player's standing in relation to the others. If help is accepted, the player, using the cursor keys, moves the

vertical arrow below the fence to the desired position and presses RETURN. When the current player is through, the horizontal arrow moves to the next player, and a new problem is displayed. The menu can be individually accessed at any moment through the asterisk (*) key, placed on the other side of the keyboard to avoid confusion with the help key.

Except for the RESET key, all the others that are not needed to input relevant information have been disabled, including the BREAK key. Mistakes can be erased by pressing DELETE. Characters entered by the user are printed on reserved spaces only and are automatically deleted whenever the spaces are filled.

Looking Inside

The program, with the DATA lines listed here, takes up 5220 bytes. REM statements have been omitted and are replaced by the following comments:

Lines 100–210 set up the routine for each game session.

Line 220 starts the main loop (CP = Current Player, NP = Number of Players).

In line 230 a definition is randomly chosen and read within the parameters of the option made by the current player.

Line 240 increases the RESTORE pointer by 1000, reads the number of words (NW), and randomly selects one (notice the flexibility allowed by the RESTORE X command).

The core of the program is in lines 320–480. Input entered by the user is analyzed and processed character by character. This slows down processing when the player is entering correct information, but should be no problem considering the potential users. It appeared to be more effective to reward the player after each correct character is entered than only after the word is finished.

Line 490 is a subroutine that deletes the input line entered by the user and moves the arrow in the help mode.

Line 500 clears the array containing the word typed by the current player.

Lines 510–540 graphically display the area on the screen where input from the user is printed.

Line 550 is the GET subroutine, preceded by the POKEs that disable the BREAK key.

Lines 560–630 guide the players through the game,

instructing them on how to use help (560), printing the number of problems solved (570), pointing to the current player (580), and updating the score (620–630).

The names are printed in line 600. This is done through a two-dimensional array using the ASC codes from the GET loop. The technique slows down printing a bit, but, again, lack of speed is no problem here.

Lines 640–820 contain the help routine. First, it is decided whether or not help is to be free and, if not, how much should be charged (650–710). Then the player is guided on how to obtain it (720–820).

Lines 830–920 display the menu and define the subscripted variables FROM(CP) and ADD(CP) according to the options entered by the current player (CP).

The data is stored in two separate groups: definitions in the 1000's and the words defined in the 2000's, with space in-between to expand the program.

Expanding the Program

This program is meant to be changed, expanded, and refined. Expansion can easily be done, but some basic rules have to be followed if the present structure is kept.

First, select a supercategory such as the three used here (animals, the human body, and the house). Break the supercategory into categories (notice that the examples used in this program could be broken up into many more categories). Then assign a group of words to each category; the only limitation here is that the words have to fit in one DATA line (if another line is needed, the category has to be repeated).

Enter each supercategory in the menu and add the appropriate IF-THEN conditions defining the variables FROM(CP) and ADD(CP). FROM(CP) refers to the DATA line where the reading of definitions (DE) should begin if the supercategory is chosen by the current player, and ADD(CP) to the definition DATA lines that should be added to complete the supercategory. Also increase the C in line 870 to allow input of a higher code.

Define the categories in one or two phrases, up to 19 characters each including spaces. If only one phrase is needed, use a comma (see examples). Enter the definitions in consecutive lines in the 1000's.

Finally, enter the words exactly 1000 lines above the corresponding definitions, preceded by the number of words in the line.

Teaching Concepts

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
AL 100 DIM D1$(20),D2$(20),WO$(20),LE(20),PL(3
       ,6),SC(3),FROM(3),ADD(3):OPEN #1,4,0,"K
       :"
II 110 FOR B=1 TO 3:FOR D=1 TO 6:PL(B,D)=0:NEX
       T D:SC(B)=0:NEXT B:GOSUB 500
AF 120 GRAPHICS 18:POSITION 1,3:? #6;"HOW MANY
        PLAYERS{6 SPACES}(UP TO THREE)":L=1:HL
       M=8:GOSUB 510
JP 130 GOSUB 550:NP=C-48:IF C<49 OR C>51 THEN
       130
JJ 140 FOR CP=1 TO NP:GRAPHICS 18:POSITION 3,2
       :? #6;"[NAME PLAYER ";CP:? #6;" [UP TO 6
       LETTERS]":L=6:HLM=6:GOSUB 510
FB 150 POSITION 7,7:FOR D=1 TO 6:GOSUB 550:IF
       C=155 THEN 180
JE 160 IF C=126 THEN GOSUB 490:GOTO 150
KC 170 ? #6;CHR$(C);:PL(CP,D)=C:NEXT D
EH 180 GOSUB 830:NEXT CP:GRAPHICS 18:CP=1:GOSU
       B 590
IJ 190 POSITION 7,4:? #6;"[WANT TO{8 SPACES}[CHA
       NGE SOMETHING?":? #6;"{3 SPACES}[Y]ES O
       R [N]O"
IK 200 GOSUB 550:IF C=89 THEN RUN
AP 210 IF C<>78 THEN 200
EK 220 LAP=LAP+1:FOR CP=1 TO NP
LC 230 DE=INT(RND(0)*ADD(CP))+FROM(CP):RESTORE
        DE:READ D1$,D2$
DC 240 RESTORE DE+1000:READ NW:WORD=INT(RND(0)
       *NW)+1:FOR A=1 TO WORD:READ WO$:NEXT A
PM 250 L=LEN(WO$):HL=(20-L)/2:HLM=HL-1
IO 260 GRAPHICS 18:POSITION 0,0:? #6;"i have a
        secret word"
DF 270 POSITION (20-LEN(D1$))/2,2:? #6;D1$:POS
       ITION (20-LEN(D2$))/2,3
NC 280 ? #6;D2$:? #6;"{5 SPACES}(";L;" LETTERS
       )"
LB 290 GOSUB 510
KO 300 GOSUB 560
FH 310 CT=0:FOR A=1 TO L
KP 320 GOSUB 550
FN 330 IF C=42 THEN GOSUB 830:GOTO 230
NI 340 IF C=27 THEN 640
```

```
JD 350 IF C=126 THEN GOSUB 490:GOTO 310
GB 360 IF C<65 OR C>90 THEN 320
FE 370 IF C<>ASC(WO$(A,A)) THEN SC(CP)=SC(CP)-
       0.5:GOSUB 620:POSITION HLM+A,7:? #6;CHR
       $(C):GOTO 430
HP 380 IF LE(A)=1 THEN 410
CN 390 LE(A)=1:POSITION HLM+A,6:? #6;CHR$(C)
HD 400 SC(CP)=SC(CP)+1:GOSUB 620
FM 410 POSITION HLM+A,7:? #6;CHR$(C)
LM 420 FOR Z=5 TO 20:SOUND 0,Z,10,10:NEXT Z:SO
       UND 0,0,0,0:CT=CT+1
BH 430 NEXT A
LF 440 GOSUB 490
NE 450 IF CT=L OR CT>L THEN 470
GH 460 GOTO 310
LM 470 GOSUB 500:NEXT CP
GJ 480 GOTO 220
BB 490 POSITION 1,7:? #6;"{18 SPACES}":RETURN
MO 500 FOR A=1 TO 20:LE(A)=0:NEXT A:RETURN
PJ 510 FOR A=1 TO L:POSITION HLM+A,8:? #6;"-"
OM 520 IF LE(A)=1 THEN POSITION HLM+A,6:? #6;W
       O$(A,A):GOTO 540
GL 530 POSITION HLM+A,6:? #6;"#";
DD 540 NEXT A:RETURN
MG 550 POKE 16,64:POKE 53774,64:GET #1,C:RETUR
       N
FE 560 POSITION 11,9:? #6;"ESC=HELP":POSITION
       11,10:? #6;"  M=MENU"
OE 570 POSITION 11,11:? #6;"TAPE";LAP
HH 580 POSITION 6,CP+8:? #6;"<"
HG 590 FOR B=1 TO NP
JC 600 POSITION 0,B+8:FOR D=1 TO 6:? #6;CHR$(P
       L(B,D));:NEXT D:POSITION 7,B+8:? #6;INT
       (SC(B)):NEXT B
HH 610 RETURN
CD 620 IF SC(CP)<0 THEN SC(CP)=0
JN 630 POSITION 7,CP+8:? #6;"{4 SPACES}":POSIT
       ION 7,CP+8:? #6;INT(SC(CP)):RETURN
NK 640 IF SC(CP)<3 THEN 720
AC 650 ME=(SC(1)+SC(2)+SC(3))/NP:IF SC(CP)<ME
       THEN 720
IA 660 MI=INT(SC(CP)/3):IF MI>L/1.5 THEN MI=IN
       T(L/1.5)
MO 670 GRAPHICS 18:? #6;" I CAN HELP YOU FOR
       ":? #6;"{6 SPACES}";MI;" POINTS"
DP 680 ? #6:? #6;"{4 SPACES}<Y> OR <N> ?"
DI 690 GOSUB 550:IF C=78 THEN 260
EG 700 IF C=89 THEN SC(CP)=SC(CP)-MI:GOTO 720
HA 710 GOTO 690
FD 720 GRAPHICS 18
```

```
EP 730 ? #6;"  MOVE ARROW TO THE":? #6;"
       {3 SPACES}LETTER YOU WANT"
FO 740 ? #6;"  THEN PRESS return":GOSUB 510
FO 750 GOSUB 570:POSITION HL,7
EN 760 ? #6;"^":AR=1
FA 770 GOSUB 550:IF C=42 AND AR<L THEN AR=AR+1
       :GOTO 820
OH 780 IF C=43 AND AR>1 THEN AR=AR-1:GOTO 820
BM 790 IF C=155 AND LE(AR)=0 THEN LE(AR)=1:GOT
       O 260
AJ 800 IF C=155 THEN 260
HA 810 GOTO 770
CI 820 GOSUB 490:POSITION HLM+AR,7:? #6;"^":GO
       TO 770
JP 830 GRAPHICS 17:? #6;" HI, ";:FOR D=1 TO 6:
       ? #6;CHR$(PL(CP,D));:NEXT D:POSITION 5,
       1:? #6;"CHOOSE ONE:"
LE 840 ? #6;" a-animals":? #6;" b-the human bo
       dy":? #6;" c-the house"
CL 850 ? #6;" d-all above"
FF 860 POSITION 9,18:? #6;"#"
MK 870 GOSUB 550:IF C<65 OR C>68 THEN 870
LH 880 IF C=65 THEN FROM(CP)=1001:ADD(CP)=3
LN 890 IF C=66 THEN FROM(CP)=1004:ADD(CP)=4
LI 900 IF C=67 THEN FROM(CP)=1008:ADD(CP)=2
LK 910 IF C=68 THEN FROM(CP)=1001:ADD(CP)=9
HL 920 RETURN
GE 1001 DATA IT IS A,WILD ANIMAL
ON 1002 DATA IT IS A,DOMESTIC ANIMAL
MF 1003 DATA ,it is a bird
LO 1004 DATA YOU HAVE IT,IN YOUR FACE
DI 1005 DATA IT IS PART,OF YOUR LEG
DN 1006 DATA it is one,of your senses
CN 1007 DATA you have it,in your hand
NG 1008 DATA YOU CAN SEE IT,IN THE KITCHEN
JH 1009 DATA IT IS,IN YOUR BEDROOM
PK 2001 DATA 5,ELEPHANT,BEAR,TIGER,LION,GIRAFF
       E
LP 2002 DATA 6,CAT,DOG,COW,HORSE,PIG,OX
BO 2003 DATA 5,HAWK,PARROT,SPARROW,CROW,PIGEON
BH 2004 DATA 4,NOSE,MOUTH,EYE,CHEEK,CHIN
IH 2005 DATA 6,ANKLE,HEEL,SHIN,CALF,KNEE,THIGH
EK 2006 DATA 5,TOUCH,SMELL,HEARING,VISION,TAST
       E
DJ 2007 DATA 4,KNUCKLE,PALM,THUMB,FINGER
JK 2008 DATA 6,STOVE,REFRIGERATOR,SINK,OVEN,DI
       SHWASHER,BLENDER
HM 2009 DATA 6,BED,CLOSET,MATTRESS,PILLOW,BLAN
       KET,SHEET
JK 2010 END
```

Deep

Karl F. Kuhn

Learn how refraction affects light, shoot aliens, improve your programming skills, and have fun at the same time. Joystick required.

You are in position on the surface of a quiet lagoon. In the deep, still water below you, a friendly submarine awaits your directions before firing a laser beam at an alien in the air above. You have two responsibilities: You must tell the sub when to fire, and you must give the sub a target on the water surface. The laser beam will pass through your target and continue on—if you have estimated correctly—to hit the alien.

The only hitch is that when a light beam passes through a water-air surface, it bends. Because of this bending (called *refraction),* you cannot simply place the target directly between the sub and the alien. You must estimate the amount of refraction which will occur. Refraction is a phenomenon of nature, however, and occurs with a pattern. Unless you have studied physics, the pattern may seem odd at first, but you will soon get the feel of it, and you will be zapping one alien after another.

Learning the Angles

A side effect of playing the game will be that you'll get the "feel" of how light actually bends when passing through a surface. The angles involved are calculated correctly in the program, and you will even see the light being totally reflected back into the water when you cause the sub's shot to hit the surface at too glancing an angle.

A clock at the upper right of the screen starts with your first shot and shows your remaining time. Your score is shown at the upper left and, after the first game, the highest score achieved so far is displayed at center.

The game requires at least 16K and one joystick. It makes use of a number of Atari features, including player/missile graphics (three players and one missile), a modified display list, and a display list interrupt.

The Program

Let's look at the program in the order in which it executes. This takes us first to the initialization of values at line 6105.

The variable named INDEX is the index of refraction of water. It determines how much the light beam will bend at the surface. After playing the game using this correct index, you might experiment with changing it. Making the index equal to 1 will result in no bending, because 1 is also the index of refraction of air. Diamond has an index of 2.4. Try that for a different game (although I don't know what a submarine would be doing inside a diamond). LIMIT is the time limit of the game in seconds.

The double GRAPHICS command in line 6110 is not a mistake; it is simply a very fast way to clear memory for P/M graphics. Later, we will use the 4K of memory starting 8K below RAMTOP for our player/missile area, and a GRAPHICS 8 command clears the top 8K of memory.

Display List Interrupt

(In the next few paragraphs are some suggestions for places to experiment and change the program. Be sure to save a copy of the program before you begin to experiment.)

Lines 6120–6150 cause a display list interrupt. This simply means that between the sixth and seventh display line on the GRAPHICS 2 screen, the Atari performs the short machine language routine using the DATA in line 6150. The purpose of this interrupt is to change some colors in the bottom half of the screen. The colors in the SETCOLOR registers 2 and 4 are changed to green. I recommend that you play with this even if you don't understand machine language. Change the 10 in line 6120 to change the position on the screen where the interrupt occurs. (We'll use this same interrupt later with the GRAPHICS 7 screen. In that case we will POKE a 141 into the display list about halfway down that screen.) To experiment further, change the 196 on the DATA line in order to cause a different color to appear at the bottom of the screen. The 24 causes the color to be POKEd into color register 2, and the 26 changes register 4. Change either of these numbers to 22 or 23 to change the colors in registers 0 or 1.

Line 6310 turns off Antic with the POKE to 559 and then establishes a GRAPHICS 1 line by POKEing 70 in the LMS command of the display list. The 12 is POKEd into the next display line in order to make a total of 40 bytes of screen memory for the two lines. Then, as promised, the display list

interrupt is set with a POKE to (DL+46), which will be the forty-second display line on the screen.

The Game Logic

The main loop consists of lines 200–290. One unusual aspect here is the use of PTRIG to read joystick positions in lines 210 and 250. It's a happy coincidence that (PTRIG(0) − PTRIG(1)) equals −1, 0, and +1 for joystick positions of left, center, and right respectively. If you use this method to read your joystick, even the diagonal positions are accepted for horizontal motion. (COMPUTE!'s *Mapping the Atari*, by Ian Chadwick, contains this and other neat tricks.)

Notice that before printing the time at lines 231 and 234 (and in the subroutine at 100), we must fool the Atari into thinking that we are in GRAPHICS 2. We do this by POKEing a 2 into location 87. Then, after printing the time, we rePOKE location 87 to the original GRAPHICS 7.

Mathematicians and physicists may want to check my calculation algorithm in lines 310–330, but most people will probably just take my word for it. The problem is that not only must the angle of refraction be calculated (from Snell's Law, physicists), but the destination for the DRAWTO must also be calculated, and it may either be at the top or the side of the screen. Hence, the messy math.

Line 360 checks the collision register for a hit and, IF HIT, the old target is erased, a new target selected, the score updated, and the program returned to the main loop at line 200.

Deep

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
GJ 90 GOTO 6100
GK 100 A=INT((PEEK(20)+PEEK(19)*256)/60):IF A>
      LIMIT THEN 900
BL 110 POKE 87,1:POSITION 16,0:PRINT #6;LIMIT-
      A;"  ":POKE 87,7:RETURN
HG 200 REM
PP 210 FOR X1=ST TO FIN STEP M:POKE H3,X1:A=PT
      RIG(1)-PTRIG(0):IF STRIG(0)=0 THEN GOSU
      B 300
MH 220 CON=CON+0.5:IF  NOT A THEN CON=1
FG 225 SPOT=SPOT-A*CON:IF SPOT<50 THEN CON=1:S
      POT=200
NJ 230 IF SPOT>200 THEN CON=1:SPOT=50
```

```
GI 231 POKE H2,SPOT:POKE 87,1:IF  NOT STARTED
       THEN POSITION 16,0:PRINT #6;"■■":POKE 8
       7,7:GOTO 250
HB 233 A=INT((PEEK(20)+PEEK(19)*256)/60):IF A>
       LIMIT THEN 900
IF 234 POSITION 16,0:? #6;LIMIT-A;" ":POKE 87,
       7
HH 250 SPOT=SPOT+(PTRIG(0)-PTRIG(1))*CON:POKE
       H2,SPOT
FL 270 IF MOVE THEN X=X+WAY:POKE H0,X:POKE HPM
       2,X:IF X=180 OR X=60 THEN WAY=WAY-2*WAY
PF 280 IF  NOT MOVE THEN POKE 707,PEEK(53770)
GJ 290 NEXT X1:GOTO 210
HH 300 REM
KA 305 TRAP 500:SURFX=SPOT-44:SUBX=X-47:POKE 5
       3278,0:IF  NOT STARTED THEN POKE 20,0:P
       OKE 19,0:STARTED=1
PD 310 SHOTS=SHOTS+1:POKE 706,52:XDIS1=ABS(SUB
       X-SURFX):SINAIR=INDEX*(XDIS1)/SQR(1600+
       XDIS1*XDIS1)
BN 315 XDIS2=40*(SINAIR/SQR(1-SINAIR*SINAIR)):
       REM TERM IN () IS TAN OF THE ANGLE WHOS
       E SINE IS SINAIR--(THETAAIR)
CK 320 TRAP 40000:TOPY=2:TOPX=SURFX+XDIS2-2*(S
       UBX>SURFX)*XDIS2
OF 330 IF TOPX>159 OR TOPX<0 THEN TOPY=40*(TOP
       X-(TOPX>159)*159)/(TOPX-SURFX):TOPX=159
       -(TOPX<0)*159
PG 333 POKE 706,14
PG 340 COLOR 2:PLOT SUBX,79:DRAWTO SURFX,40:DR
       AWTO TOPX,TOPY:FOR I=100 TO 220 STEP 20
       :SOUND 0,I,10,15:NEXT I
KG 350 SOUND 0,0,0,0:COLOR 0:PLOT SUBX,79:DRAW
       TO SURFX,40:DRAWTO TOPX,TOPY
MO 357 POKE H0,X:POKE HPM2,X
KI 360 HIT=PEEK(53255):IF HIT THEN 600
HM 390 RETURN
HI 400 REM
JH 410 FOR I=15 TO 1 STEP -1:SOUND 1,20*I,10,I
       :NEXT I:SOUND 1,0,0,0
GO 440 SCORE=SCORE+80-2*SHOTS+10*(SHOTS=1)+20*
       MOVE
KJ 442 POKE 87,1:POSITION 2,0:PRINT #6;SCORE
HL 450 LIMIT=LIMIT+1:GOSUB 100
LC 460 SHOTS=0:RETURN
HJ 500 REM
NA 520 BOTY=79:BOTX=SURFX+XDIS1-2*XDIS1*(SUBX>
       SURFX):IF BOTX>159 THEN BOTY=40+39*(159
       -SURFX)/(BOTX-SURFX)
```

```
LK 522 IF BOTX<Ø THEN BOTY=4Ø+39*SURFX/(SURFX-
       BOTX)
NP 525 BOTX=BOTX-BOTX*(BOTX<Ø)-(BOTX-159)*(BOT
       X>159):POKE 7Ø6,14:GOSUB 1ØØ
IP 529 SHOTS=SHOTS+2:TRAP 55Ø:COLOR 2:PLOT SUB
       X,79:DRAWTO SURFX,41:DRAWTO BOTX,BOTY
ON 53Ø SOUND Ø,15Ø,12,12:SOUND 1,152,12,12:? :
       ? :? "{5 SPACES}TOTAL INTERNAL REFLECT
       ION ":FOR D=1 TO 14Ø+2ØØ*( NOT MOVE)
KF 54Ø NEXT D:COLOR Ø:PLOT SUBX,79:DRAWTO SURF
       X,41:DRAWTO BOTX,BOTY:? CHR$(125)
MG 55Ø GOSUB 1ØØ:SOUND Ø,Ø,Ø,Ø:SOUND 1,Ø,Ø,Ø:G
       OTO 27Ø
HK 6ØØ REM
MP 615 FOR I=MYPM+1792+Y1 TO MYPM+1799+Y1:POKE
        I,Ø:NEXT I:IF HIT=2 THEN GOSUB 4ØØ
OJ 62Ø Y1=6Ø+28*RND(Ø)
AP 63Ø M=3*RND(Ø)-1.5:ST=195*(M<Ø)+5Ø*(M>=Ø):F
       IN=245-ST:IF  NOT MOVE THEN ST=46+15Ø*R
       ND(Ø):M=Ø
MC 635 IF ST<16Ø AND ST>8Ø AND RND(Ø)>Ø.5 THEN
        ST=46+153*(RND(Ø)>Ø.5)
BH 64Ø POKE H3,ST
AJ 65Ø RESTORE 66Ø+RND(Ø)*4:FOR I=MYPM+1792+Y1
        TO MYPM+1799+Y1:READ A:POKE I,A:NEXT I
       :READ A:POKE 7Ø7,A
NK 66Ø DATA 24,6Ø,Ø,255,255,Ø,6Ø,24,2Ø2
CK 661 DATA 129,153,189,255,189,153,129,Ø,234
NO 662 DATA 24,1Ø2,255,153,153,255,1Ø2,24,28
GF 663 DATA Ø,28,8,73,127,73,8,28,46
MJ 664 DATA 85,42,85,42,85,42,85,42,12
GK 69Ø GOTO 2ØØ
LP 9ØØ SOUND Ø,Ø,Ø,Ø:REM ENDING ROUTINE
DP 91Ø POKE 7Ø6,15:? "{DOWN}{14 SPACES}TIME'S
       UP ":IF SCORE>BEST THEN BEST=SCORE:BEST
       $=STR$(BEST):GOTO 97Ø
KA 92Ø SOUND 1,1ØØ,1Ø,1Ø:FOR TIME=1 TO 2ØØ:POK
       E 755,3:POKE 755,2:NEXT TIME:SOUND 1,Ø,
       Ø,Ø
GC 925 BEST$=STR$(BEST):POKE 53279,Ø:SCORE=Ø:S
       TARTED=Ø:? "{CLEAR}{DOWN}{7 SPACES}PRES
       S START FOR SAME GAME."
CB 926 IF MOVE=1 THEN ? "{3 SPACES}PRESS SELEC
       T FOR STILL TARGET GAME.":GOTO 93Ø
PJ 928 ? " PRESS SELECT FOR MOVING TARGET GAME
       ."
NF 93Ø POKE 87,1:POSITION 8,Ø:IF LEN(BEST$)<4
       THEN BEST$(LEN(BEST$)+1)=" ":GOTO 93Ø
NH 94Ø FOR I=1 TO 4:PRINT #6;CHR$(ASC(BEST$(I;
       I))-32);:NEXT I
```

```
PH 950  I=PEEK(53279):IF I=7 THEN 950
PP 955  IF I=5 THEN MOVE=ABS(MOVE-1):IF  NOT MO
        VE THEN X=122:POKE HØ,X:POKE HPM2,X
PA 960  POSITION 2,Ø:PRINT #6;"ØØ  ":POKE 87,7:
        HIT=Ø:POKE 53279,Ø:STARTED=Ø:LIMIT=6Ø:?
         CHR$(125):POKE 77,Ø:GOTO 6ØØ
DH 97Ø  RESTORE 99Ø:HOLD=3
OB 975  READ A,B:IF A=999 THEN SOUND 1,Ø,Ø,Ø:SO
        UND Ø,Ø,Ø,Ø:POKE 755,2:GOTO 925
BE 98Ø  POKE 755,2:SOUND 1,A,1Ø,12:SOUND Ø,A+1,
        1Ø,6:FOR TIME=1 TO B*HOLD:NEXT TIME:POK
        E 755,3:GOTO 975
DE 99Ø  DATA 72,4,Ø,2,72,4,Ø,2,72,4,Ø,2,72,3Ø,6
        4,2Ø,Ø,5,85,1Ø,81,1Ø,72,2Ø,Ø,5,85,1Ø,81
        ,1Ø,72,1Ø
OP 991  DATA Ø,1Ø,85,1Ø,81,1Ø,72,1Ø,Ø,1Ø,64,2Ø,
        57,3Ø,999
GO 61ØØ REM SET UP DISPLAY LIST
AF 6105  INDEX=1.33:LIMIT=6Ø:DIM BEST$(4):POKE
        82,Ø:SPOT=1ØØ:WAY=Ø.5
MI 6110  GRAPHICS 8:GRAPHICS 2:SETCOLOR 4,9,4:D
        L=PEEK(56Ø)+256*PEEK(561)
CF 6120  POKE DL+1Ø,135:RESTORE 6150:POKE 52761
        ,8:FOR I=Ø TO 13:READ A:POKE 1536+I,A:
        NEXT I
IL 6130  POKE 512,Ø:POKE 513,6:POKE 54286,192
IE 6150  DATA 72,169,196,141,1Ø,212,141,24,2Ø8,
        141,26,2Ø8,1Ø4,64
GP 62ØØ REM INITIALIZE VALUES & CHOOSE GAME
DH 6250  COL=49:POSITION 8,1:? #6;"deep":POSITI
        ON 5,7:? #6;"still target":? #6;"
        {5 SPACES}moving target"
NB 6251  POKE 752,1:? :? "{15 SPACES}CHOOSE ONE"
FB 6252  COL=ABS(COL-194):COLOR COL:PLOT 3,7:A=
        ABS(COL-194)+1:COLOR A:PLOT 3,8:FOR T=
        1 TO 5Ø:NEXT T:SOUND Ø,A,1Ø,2
FD 6260  I=PEEK(764):MOVE=(I=3Ø)
NB 6280  IF I<3Ø OR I>31 THEN 6252
GJ 6290  POKE 764,255
HA 63ØØ REM SET UP DISPLAY LIST
KJ 6310  GRAPHICS 7:POKE 559,Ø:DL=PEEK(56Ø)+256
        *PEEK(561):POKE DL+3,7Ø:POKE DL+6,12
OB 6320  POKE DL+46,141
EA 64ØØ REM SET COLORS & DRAW
BM 6405  SETCOLOR 4,8,Ø:SETCOLOR 1,7,1Ø:SETCOLO
        R 2,Ø,14:POKE 752,1
IA 6450  COLOR 2
DM 65ØØ REM SET UP PM GRAPHICS
OP 6503  X=122:POKE 704,Ø:POKE 7Ø6,14
JA 6505  A=PEEK(1Ø6)-32:POKE 54279,A:MYPM=256*A
```

```
PO 651Ø POKE 53277,3:HØ=53248:POKE HØ,X:H2=532
   5Ø:H3=53251:HPM2=53254:POKE HPM2,X:POK
   E 53256,1
GJ 653Ø RESTORE 654Ø:FOR I=MYPM+122Ø TO MYPM+1
   226:READ A:SOUND Ø,A,1Ø,1Ø:POKE I,A:NE
   XT I
IE 6535 POKE MYPM+1656,231:POKE MYPM+1657,231:
   POKE MYPM+966,48
FG 654Ø DATA 32,32,124,255,255,255,Ø
EF 659Ø POKE 559,62:GOTO 62Ø
```

Heroes

Steven Leffler

"Heroes" is an exciting adventure game that combines joystick movement and keyboard commands. Requires at least 16K RAM, a joystick, and more thought than is immediately apparent.

You are a fearless explorer, traveling in search of adventure. You have come a long way and have now arrived at an inn in a strange land. As the innkeeper serves your dinner, he tells you some of the history of this place.

"This country," claims the innkeeper, "which extends for many miles both to the north and east of the inn, was once used by a clan of wizards for their evil experiments. One day the wizards disappeared, leaving their castles to fall into ruin and the creatures which once guarded them to run wild. It is said that, to this day, the creatures the wizards created still roam the wilds of this region and the ruins of the castles that were once their homes. It is also said that, in their haste to depart, the wizards left behind much treasure, and anyone who survives a trip into their demesnes will leave a rich man."

Even before the innkeeper finishes his tale, you have decided to explore this strange country. You rise early the next morning, bid the innkeeper farewell, and depart into the unknown.

The object of "Heroes" is to explore the entire area shown on the screen and get as much treasure as you can without getting killed. Your final score is based on the amount of treasure you have collected during your travels.

You start out with 50 hit points. These show how healthy you are; when you have lost all of them in combat, you die. Fortunately, you can return to the inn to rest and regain hit points (with a small fee for use of the facilities, of course).

As you explore, you will have to face many monsters. Each one guards a pile of treasure which you can take after you slay the beast. The treasure will consist of such things as gold, magical weapons and armor, as well as the very powerful magical scrolls. The names of the scrolls are often cryptic, giving no clue to their purpose, so you will have to be careful when using them.

Playing Heroes

Use a joystick in slot 1 to show your direction of movement and word commands to indicate anything else. Type HELP and press RETURN to get a list of the words you can use. Press RETURN to enter a command. TRIG means push the joystick trigger.

TAKE	Adds items to your treasure
LIST	Prints your treasure list
SPELL	Uses a scroll from the treasure
TRIG	Causes you to attack with your sword if there is a monster present; is the same as LIST at all other times
HELP	Prints a brief summary of these commands
QUIT	Ends the game and displays your total points

Different colors show different terrain:

Brown	Meadow
Green	Forest
Gray	Ruins
Black	Unexplored

You start with 50 hit points (HP). These are a measure of your ability to take damage. Damage reduces your HP total; when it reaches 0, the game ends. If you get low, you may return to the inn, and regain 1 HP per day at a cost of two gold pieces (GP) for food and accommodations.

Finally, a playing tip—victory depends heavily on your ability to make effective use of the scrolls.

Heroes

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
CA 9 GRAPHICS 2:POKE 752,1
MN 10 SETCOLOR 2,0,0:? #6;"{CLEAR}welcome to
      heroes":POSITION 0,2:? #6;" A GAME OF FA
      NTASY"
IN 20 POSITION 0,4:? #6;" AND ADVENTURE IN A":
      POSITION 0,6:? #6;"{3 SPACES}MEDIEVAL ER
      A":? "{2 TAB}{3 SPACES}by Steven Leffler
      "
DO 21 FOR T=1 TO 2000:NEXT T
IM 30 HP=50:EX=1:DIM I$(1),C$(6),B$(12),SC$(80
      ),MT$(24)
JP 35 OPEN #1,4,0,"K:"
MA 160 ? :? "PUSH ANY KEY TO START YOUR ADVENT
       URE":GET #1,T
```

```
LB 170  L1=0:L2=19:S=0:W=1:GRAPHICS 3:SETCOLOR
        0,15,2:SETCOLOR 1,0,4:SETCOLOR 2,12,0
MG 175  REM INPUT SECTION
HD 180  POKE 752,1:COLOR W:PLOT L1,L2
NB 190  C=STICK(0):IF C<>15 THEN 247
KI 195  IF EX>799 THEN 1060
HN 200  IF STRIG(0)=0 AND B1>0 THEN GOTO 1080
KH 205  IF STRIG(0)=0 AND B1<1 THEN GOSUB 480:G
        OTO 180
OC 210  COLOR 0:PLOT L1,L2
JE 220  TRAP 220:C2=PEEK(764):IF C2=12 THEN GOS
        UB 540:GOTO 180
PH 225  IF C2=255 OR C2=60 THEN 180
LM 240  GET #1,C2:IF C2=126 AND LEN(C$)>1 THEN
        C$=C$(1,LEN(C$)-1):? CHR$(C2);:GOTO 180
BA 242  IF C2=126 THEN C$="":? CHR$(C2);:GOTO 1
        80
NB 245  C$(LEN(C$)+1)=CHR$(C2):? CHR$(C2);:GOTO
         180
MD 246  REM MOVEMENT INTERPRETATION
BL 247  C$="":OL1=L1:OL2=L2:IF F=1 THEN ? "THE
        ";B$;" IS BLOCKING THE WAY!":GOTO 180
PH 248  IF B1>0 THEN ? "THE ";B$;" IS CHASING Y
        OU!":IF INT(RND(0)*4+1)>(B1/10) AND SS=
        0 THEN ? "IT'S TOO FAST":F=1:GOTO 180
FF 249  IF B1>0 THEN ? "YOU OUTRAN IT!"
KL 250  SOUND 0,46,0,8:GOSUB 3000:IF C=14 THEN
        ? "N":L2=L2-1:GOTO 330
MA 260  IF C=6 THEN ? "NE":L1=L1+1:L2=L2-1
KD 270  IF C=7 THEN ? "E":L1=L1+1
ME 280  IF C=5 THEN ? "SE":L1=L1+1:L2=L2+1
OC 290  IF C=13 THEN ? "S":L2=L2+1
NF 300  IF C=9 THEN ? "SW":L1=L1-1:L2=L2+1
NN 310  IF C=11 THEN ? "W":L1=L1-1
PM 320  IF C=10 THEN ? "NW":L1=L1-1:L2=L2-1
FL 330  SOUND 0,0,0,0:IF L1<0 OR L2<0 OR L1>39
        OR L2>19 THEN ? "{BELL}YOU CANNOT LEAVE
         THE SCREEN!":L1=OL1:L2=OL2:GOTO 180
AH 335  FOR T20=1 TO 20:NEXT T20:LOCATE L1,L2,W
        :IF W<>0 THEN 440
HL 340  EX=EX+1:W=INT(20*RND(0))+1:IF W<17 THEN
         W=3
LF 350  IF W=18 OR W=17 THEN W=1
PI 360  IF W=20 OR W=19 THEN W=1:IF S+SH>2 THEN
         W=2
DD 370  RESTORE :COLOR W:PLOT L1,L2:ON W GOTO 3
        90,420
LG 380  IF INT(100*RND(0))=0 THEN 400
HC 385  GOTO 180
MJ 390  IF INT(10*RND(0))<>0 THEN 180
```

```
GC 395 REM MINOR MONSTER MAKER
DL 400 FOR T=1 TO (INT(RND(0)*5+1)):READ B$,B1
       ,B2,B3,B4,B5:NEXT T
OF 410 ? "OH,NO THERE'S A ";B$;" COMING!":? "W
       HAT ARE YOU GOING TO DO?":GOSUB 2500:GO
       TO 180
EP 415 REM MAJOR MONSTER MAKER
PN 420 FOR T=1 TO 30:READ B$:NEXT T:FOR T=1 TO
        (INT(RND(0)*6+1)):READ B$,B1,B2
ME 430 READ B3,B4,B5:NEXT T:? "OH,NO THERE'S A
        ";B$;" COMING!":? "WHAT ARE YOU GOING
       TO DO?":GOSUB 2500:GOTO 180
EC 435 REM IF AT INN SEQUENCE
JP 440 IF L1>0 OR L2<19 THEN 180
FO 450 ? "YOU ARE AT THE INN, DO YOU WISH TO
       {4 SPACES}STAY AND REST?":B1=20:GOSUB 2
       500:B1=0
OD 455 GET #1,T:IF T=78 THEN 180
PC 460 T=50-HP:IF GP-2*T<0 THEN T=INT(GP/2)
IK 470 GP=GP-2*T:HP=HP+T:? "HP: ";HP,"GP: ";GP
       :GOTO 180
AL 480 REM TREASURE LIST PRINTER
MC 490 C$="":? "{CLEAR}HP: ";HP,"GP: ";GP,"A +
       ";S;" SWORD",:IF SS=1 THEN ? "ELVEN SLI
       PPERS",
OJ 495 ? "AND +";SH;" ARMOUR":? "PUSH RETURN TO
       CONTINUE"
NM 500 FOR T20=1 TO 50:NEXT T20:IF STRIG(0)=1
       THEN 500
IA 515 ? "{CLEAR}":POKE 752,1:FOR T20=1 TO 20:
       NEXT T20:IF LEN(SC$)=0 THEN ? "{UP}":RE
       TURN
OG 517 POKE 82,0:? "SCROLLS:";
PK 520 FOR T=1 TO (LEN(SC$)/8):? SC$(T*8-6,T*8
       ),:NEXT T:? "PUSH TO SEE":FOR T20=1 TO 20
       :NEXT T20:POKE 82,2
GG 530 IF STRIG(0)=1 THEN 530
DD 535 ? "{CLEAR}":FOR T20=1 TO 50:NEXT T20:RE
       TURN
MI 540 REM MULTIPLE KEYWORD COMMANDS
KA 550 GET #1,T:IF C$="LIST" THEN GOSUB 480:RE
       TURN
LE 560 IF C$="SPELL" THEN C$="":GOTO 800
FM 570 IF C$="TAKE" THEN C$="":GOTO 640
HI 580 IF C$="QUIT" THEN C$="":GOTO 610
EA 590 IF C$<>"HELP" THEN 600
MJ 592 ? "{CLEAR}TAKE:takes treasure{TAB}SPELL
       :casts spell";:? "LIST:shows inventory
       {TAB}QUIT:ends game"
```

```
CH 595 ? "trig:shows inventory/attacks w/ swor
       d"
KH 597 C$="":RETURN
IB 600 ? "{CLEAR}{BELL}WHAT??":C$="":RETURN
EC 605 REM quit
FF 610 ? "{CLEAR}POINTS: ";10*(GP+(LEN(SC$))+S
       +SH):? "ARE YOU SURE YOU WANT TO QUIT?"
       :C$=""
LG 620 GET #1,C:IF C=89 THEN GRAPHICS 0:END
LD 630 ? "{CLEAR}":RETURN
CL 639 REM take
JA 640 ? "{CLEAR}":IF LEN(MT$)=0 THEN 750
KL 650 FOR T=1 TO (LEN(MT$)/8):IF MT$(T*8-7,T*
       8-7)<>"S" THEN 710
LI 660 IF MT$(T*8-7,T*8-3)<>"SWORD" THEN 690
KI 670 C=VAL(MT$(T*8,T*8)):IF C<=S THEN NEXT T
       :GOTO 750
MB 680 S=C:? "+ ";C;" SWORD":NEXT T:GOTO 750
DL 690 IF MT$(T*8-5,T*8)="SSPEED" AND SS=1 THE
       N NEXT T:GOTO 750
MB 700 IF LEN(SC$)=80 THEN ? "YOU CANNOT CARRY
        ANY MORE SCROLLS":NEXT T:GOTO 750
CK 705 SC$(LEN(SC$)+1)=MT$(T*8-7,T*8):? "SCROL
       L:";MT$(T*8-6,T*8):NEXT T:GOTO 750
GJ 710 IF MT$(T*8-7,T*8-7)<>"A" THEN 725
PA 715 C=VAL(MT$(T*8,T*8)):IF C<=SH THEN NEXT
       T:GOTO 750
EL 720 SH=C:? "+ ";C;" ARMOUR":NEXT T:GOTO 750
HE 725 IF MT$(T*8-7,T*8-7)<>"E" THEN 735
BP 730 SSQ=0:SS=1:? "ELVEN SLIPPERS-SUPER SPEE
       D!":NEXT T:GOTO 750
OL 735 ? "POTION OF HEALING... DO YOU WANT TO
       {3 SPACES}USE IT NOW";:INPUT I$:IF I$<>
       "N" THEN HP=HP+5:? "HP:";HP:NEXT T:GOTO
        750
GL 740 ? "IT TURNS TO WATER!":NEXT T
CI 750 GP=GP+MG:? MG;" GOLD PIECES":C$="":MT$=
       "":MG=0:RETURN
CJ 795 REM spell
GC 800 ? "{CLEAR}WHICH SCROLL (TYPE LIST TO SE
       E YOUR{3 SPACES}LIST)";:INPUT C$:IF C$=
       "LIST" THEN GOSUB 515:GOTO 800
IP 810 FOR T=1 TO (LEN(SC$)/8):IF SC$(T*8-5,T*
       8)<>C$ THEN NEXT T:? "YOU DON'T HAVE TH
       AT SCROLL!":C$="":RETURN
KC 820 C=0:IF C$="HEALTH" THEN HP=50:? "HP:50"
       :GOSUB 2600:GOTO 960
KH 830 IF C$="SSPEED" AND SS=0 THEN SS=1:SSQ=1
       :F=0:? "SUPER SPEED ON.":GOSUB 2600:GOT
       O 960
```

```
JH 840 IF C$="GOLUCK" THEN S=S+1:SH=SH+1:GOQ=G
       OQ+1:? "YOU FIGHT BETTER!":GOSUB 2600:G
       OTO 960
HL 850 IF C$<>"STONES" OR B1<1 THEN 860
IP 855 ? "A GUST OF WIND PELTS THE MONSTER WIT
       H STONES!":C=INT(RND(0)*10)+1+S:GOSUB 2
       700:GOTO 960
OB 860 IF C$<>"CONTRA" THEN 900
BK 870 ? "A BEAM FLIES FROM THE PAPER! YOU ARE
         HIT!!":FOR T2=1 TO 200:SETCOLOR 4,4,0
       :SOUND 0,76,10,10
DL 880 SETCOLOR 4,0,0:SOUND 0,102,10,10:NEXT T
       2:EX=1:? #6;"{CLEAR}":COLOR 1:PLOT 0,19
       :W=0:? "YOU ARE TOTALLY LOST!!"
PN 890 SOUND 0,0,0,0:GOSUB 3000:GOTO 960
KG 900 IF C$="FIREBA" THEN ? "A FIREBALL SHOOT
       S FROM THE PAPER!!":C=INT(RND(0)*21)+10
       :GOSUB 2800:GOTO 960
HO 910 IF C$="ENERGY" THEN ? "A LIGHTNING BOLT
        SHOOTS FROM THE{6 SPACES}PAPER!!!":C=I
       NT(RND(0)*11)+20:GOSUB 2900:GOTO 960
OD 920 IF C$<>"BATSHR" THEN 935
KC 925 IF B$="GIANT BAT" THEN ? "THE BAT SHRIN
       KS AND DISAPPEARS!":B1=0:POP :GOSUB 960
       :GOTO 1150
NB 930 ? "THERE IS NO BAT HERE, SO NOTHING
       {6 SPACES}HAPPENS":GOTO 960
OC 935 IF C$<>"SEEING" THEN 955
BG 940 X=INT(RND(0)*38)+1:Y=INT(RND(0)*18)+1:F
       OR T3=X-1 TO X+1:FOR T2=Y-1 TO Y+1:LOCA
       TE T3,T2,T20
AD 945 IF T20<>0 THEN NEXT T2:NEXT T3:C=0:GOSU
       B 2600:GOTO 960
BI 947 EX=EX+1:C=INT(RND(0)*10)+1:IF C>3 THEN
       C=3
IP 950 COLOR C:PLOT T3,T2:NEXT T2:NEXT T3:C=0:
       GOSUB 2600:GOTO 960
HK 955 ? "NOTHING HAPPENS."
BF 960 C$="":IF T*8=LEN(SC$) THEN SC$(T*8-7)="
       ":GOTO 970
CD 965 SC$(T*8-7)=SC$(T*8+1)
BO 970 IF B1<1 THEN RETURN
EM 980 POP :IF C=0 THEN 1110:REM MONSTER ATTAC
       KS
OB 990 ? "YOU DO ";C;" HP DAMAGE!":B1=B1-C:IF
       B1<1 THEN 1150:REM MONSTER DIES
AO 1000 GOTO 1110:REM MONSTER ATTACKS
IG 1059 REM YOU WON!!
GG 1060 GRAPHICS 2+16:POSITION 4,5:? #6;"YOU H
        AVE WON"
```

```
FF 1065 FOR C=1 TO 200:NEXT C
NH 1070 FOR X=0 TO 5:FOR C=0 TO 255:POKE 712,C
        :SOUND 0,C,10,10:NEXT C:NEXT X
NO 1075 SOUND 0,0,0,0:GOTO 1385
GL 1079 REM COMBAT
AG 1080 ? "YOU USE YOUR SWORD.";:FOR T=12 TO 0
         STEP -1:SOUND 0,T,0,8:NEXT T:SOUND 0,
        0,0,0
NI 1085 IF INT(RND(0)*20)+1+S>8 THEN 1095
GG 1090 ? "YOU MISSED!":GOTO 1110
IM 1095 ? "A HIT! ";:FOR T=15 TO 0 STEP -1:SOU
        ND 0,70,4,T:NEXT T
HD 1100 X=INT(RND(0)*10)+1+S:? X;" HP DAMAGE!"
        :F=0:B1=B1-X:IF B1<1 THEN 1150
DA 1110 FOR X=1 TO 100:NEXT X:? "THE ";B$;" AT
        TACKS.";:FOR T=12 TO 0 STEP -1:SOUND 0
        ,T,0,8:NEXT T:SOUND 0,0,0,0
NK 1115 IF INT(RND(0)*20)+1-SH+B2>10 THEN 1130
NG 1120 ? "IT MISSED!":GOTO 180
FE 1130 ? "IT HIT! ";:FOR T=15 TO 0 STEP -1:SO
        UND 0,60,4,T:SETCOLOR 2,2,T:NEXT T:SET
        COLOR 2,12,0
PH 1132 X=INT(RND(0)*10)+1+(B2*2):? X;" HP DAM
        AGE!":HP=HP-X:? "YOU HAVE ";HP;" HP LE
        FT"
FJ 1135 IF HP<1 THEN 1380
JI 1140 GOTO 180
EF 1149 REM MONSTER IS DEAD
NM 1150 ? "THE ";B$;" IS DEAD!":FOR T=50 TO 15
        0:SOUND 0,T,10,10:SOUND 1,T-10,10,10:N
        EXT T
HP 1160 SOUND 0,0,0,0:SOUND 1,0,0,0:FOR T=1 TO
         50:NEXT T:SOUND 0,20,0,10:FOR T=1 TO
        15:NEXT T:SOUND 0,0,0,0
AE 1170 ? "THERE IS A SMALL PILE OF TREASURE!"
        :IF B2=0 THEN 1230
ND 1180 FOR X=1 TO B2:Y=INT(RND(0)*100)+1:IF Y
        <26 THEN MT$(LEN(MT$)+1)="ELF SLIP":NE
        XT X
EG 1190 IF Y>25 AND Y<46 THEN MT$(LEN(MT$)+1)=
        "ARMOUR+3":NEXT X
PL 1200 IF Y>45 AND Y<66 THEN MT$(LEN(MT$)+1)=
        "SWORD +3":NEXT X
CG 1210 IF Y>65 AND Y<81 THEN MT$(LEN(MT$)+1)=
        "S GOLUCK":NEXT X
EE 1220 IF Y>80 THEN MT$(LEN(MT$)+1)="S HEALTH
        ":NEXT X
CK 1230 IF B3=0 THEN 1300
BC 1240 FOR X=1 TO B3:Y=INT(RND(0)*100)+1:IF Y
        <11 THEN MT$(LEN(MT$)+1)="POTHEALI":NE
        XT X
```

```
DK 1250 IF Y>10 AND Y<36 THEN MT$(LEN(MT$)+1)=
        "ARMOUR+1":NEXT X
EE 1260 IF Y>35 AND Y<46 THEN MT$(LEN(MT$)+1)=
        "ARMOUR+2":NEXT X
CM 1270 IF Y>45 AND Y<66 THEN MT$(LEN(MT$)+1)=
        "S SSPEED":NEXT X
AB 1280 IF Y>65 AND Y<91 THEN MT$(LEN(MT$)+1)=
        "SWORD +1":NEXT X
CP 1290 IF Y>90 THEN MT$(LEN(MT$)+1)="SWORD +2
        ":NEXT X
DC 1300 IF B4=0 THEN 1372
MJ 1310 Y=INT(RND(0)*100)+1:IF Y<8 THEN MT$(LE
        N(MT$)+1)="S HEALTH"
DE 1320 IF Y>7 AND Y<30 THEN MT$(LEN(MT$)+1)="
        S STONES"
FL 1330 IF Y>29 AND Y<37 THEN MT$(LEN(MT$)+1)=
        "S CONTRA"
DL 1340 IF Y>36 AND Y<54 THEN MT$(LEN(MT$)+1)=
        "S FIREBA"
FK 1350 IF Y>53 AND Y<61 THEN MT$(LEN(MT$)+1)=
        "S ENERGY"
FD 1360 IF Y>60 AND Y<88 THEN MT$(LEN(MT$)+1)=
        "S SEEING"
IO 1370 IF Y>87 THEN MT$(LEN(MT$)+1)="S BATSHR
        "
GC 1372 MG=INT(RND(0)*B5):B$="":B1=0:B2=0:B3=0
        :B4=0:B5=0:IF SSQ=1 THEN SSQ=0:SS=SSQ
CE 1377 S=S-GOQ:SH=SH-GOQ:GOQ=0:GOTO 180
AL 1379 REM PLAYER DEATH
FD 1380 ? "YOU ARE DEAD!"
HP 1385 T=10*(GP+(LEN(SC$))+S+SH)
EG 1390 ? "YOU HAD ";T;" POINTS":IF HP<1 THEN
        GOSUB 3100
BJ 1395 ? "DO YOU WANT TO PLAY AGAIN";:INPUT I
        $:IF I$="Y" THEN CLR :RUN
BC 1400 ? "ARE YOU SURE";:INPUT I$:IF I$="N" T
        HEN CLR :RUN
FI 1410 GRAPHICS 0:END
MI 2002 REM MINOR MONSTERS
BE 2003 DATA GIANT BAT,20,0,2,0,75,WERE-WOLF,2
        0,0,2,0,50,GIANT SPIDER,10,0,2,0,50,GO
        BLIN,10,0,1,0,25
JF 2005 DATA GHOST,10,0,1,0,10
MB 2007 REM MAJOR MONSTERS
ID 2009 DATA HUGE DRAGON,40,2,0,1,150,DRAGON,3
        0,1,0,1,100,HUGE GIANT,40,2,0,1,100,GI
        ANT,30,1,0,1,75
AC 2011 DATA HUGE SPIDER,40,2,0,1,75,TROLL,30,
        1,0,0,75
PL 2499 REM WARNING SOUND
```

```
PD 2500 FOR T=1 TO 2:FOR T1=15 TO 0 STEP -1:SO
        UND 0,B1*3,10,T1:NEXT T1:NEXT T
GA 2510 FOR T=1 TO 50:NEXT T:RETURN
GG 2599 REM SPELL SOUND
IO 2600 FOR T1=20 TO 5 STEP -1:SOUND 0,INT(RND
        (0)*2)-1+T1,10,10:NEXT T1:SOUND 0,0,0,
        0
MD 2610 FOR T1=1 TO 50:NEXT T1:RETURN
IH 2699 REM "STONES" SOUND
LD 2700 FOR T1=12 TO 0 STEP -1:SOUND 0,T1,8,8:
        NEXT T1:SOUND 0,0,0,0
MI 2710 FOR T1=1 TO 3:FOR T2=15 TO 0 STEP -4:S
        OUND 0,12,0,T2:NEXT T2:NEXT T1:SOUND 0
        ,0,0,0
MF 2720 FOR T1=1 TO 50:NEXT T1:RETURN
ON 2799 REM "FIREBALL" SOUND
LE 2800 FOR T1=12 TO 0 STEP -1:SOUND 0,T1,8,8:
        NEXT T1:SOUND 0,0,0,0
MF 2810 FOR T1=15 TO 0 STEP -1:SOUND 0,50,0,T1
        :FOR T2=1 TO 8:NEXT T2:NEXT T1
MG 2820 FOR T1=1 TO 50:NEXT T1:RETURN
KC 2899 REM "LIGHTNING BOLT" SOUND
FP 2900 FOR T1=0 TO 75:SOUND 0,T1,0,10:NEXT T1
        :SOUND 0,0,0,0:RETURN
KA 2999 REM RESET MONSTER PARAMETERS
AL 3000 B$="":B1=0:B2=0:B3=0:B4=0:B5=0:MT$="":
        MG=0:RETURN
DF 3099 REM DEATH KNELL
KN 3100 RESTORE 3200
BM 3110 FOR T=1 TO 28:READ T2,T3:SOUND 0,T2,10
        ,8*SGN(T2):FOR T2=1 TO T3/2:NEXT T2:SO
        UND 0,0,0,0:FOR T2=1 TO 6:NEXT T2
HD 3120 NEXT T:RETURN
KH 3200 DATA 108,50,108,25,81,150,0,25
HJ 3210 DATA 108,50,81,25,64,150,0,25
GB 3220 DATA 108,50,81,25,64,75
GC 3230 DATA 108,50,81,25,64,75
IN 3240 DATA 108,50,81,25,64,150
EM 3250 DATA 81,50,64,25,53,150,0,25
IC 3260 DATA 64,50,81,38,108,150,0,25
LL 3270 DATA 108,50,108,25,81,200
LF 4998 END
```

Granite Cracker

Thomas Edwards

"Granite Cracker" is one of those easy to play, but hard to master games. Most players will do okay, but few will do really well without a great deal of practice. Requires a joystick.

Zarnog 5 was a great intergalactic space port. Thousands of ships arrived and departed every day. But, suddenly, tragedy struck. Overnight, hundreds of granite cubes, one mile to a side, blocked the spaceways around the planet. Realizing what devastating effects this might have on business, the leaders of Zarnog 5 created a fleet of giant bouncing jackhammers to combat the blocks. However, these machines were very dangerous because they tended to pop out of existence if it took long to destroy the blocks. A crash spelled instant death to the operator. So the leaders gathered together the best pilots for a crack team of jackhammer operators. Now it's your turn to pilot a granite cracker.

Using the Hammer

You can move the jackhammer right and left by the joystick in port 1. Up-and-down movement is provided by bouncing on the granite blocks. If you hit a block by falling from above, it will probably destroy the block. If it doesn't disappear, try hitting it closer to the middle.

Each background color has its own special meaning:

Black	Doing fine.
Red	Watch out! Time is almost up.
Yellow-Green	Oops! You crashed to the ground.
Blue	Oops! Ran out of time.

When you crash or run out of time, the computer will take a few moments to see how many blocks are left. If none is left, you get to play another screen. If some blocks are left on the screen, the computer will tell you your score, and you'll be able to play again or change parameters.

More Bounce

To start the game just press the START key. You'll be given the option of changing the horizontal and vertical speed and the bounce factor. To make the game a bit easier, raise the bounce factor.

"Granite Cracker" is a game that takes some practice, especially if the bounce factor is low.

Granite Cracker

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
GJ Ø DIM PM1$(128),HAM$(16),CLEAR$(128)
HA 1 L1=1:L2=1:L3=8
MI 4 REM SET SPEEDS
HE 5 L1=1:L2=1:L3=8
HJ 1Ø GOSUB 62Ø
IA 2Ø GOSUB 68Ø
CM 29 REM UPDATE LEVEL AND PREPARE SCREEN
JF 3Ø L5=L5+1:GRAPHICS 3+16:COLOR 1:POKE 71Ø,Ø
AL 4Ø FOR I=1 TO 6Ø
KH 41 X=INT(38*RND(Ø)+2):Y=INT(19*RND(Ø)+4):LO
      CATE X,Y,Z:IF Z<>Ø THEN 41
EE 42 PLOT X,Y:NEXT I
DK 5Ø Q=-1:COLOR Ø
BC 59 REM SET TIMERS
LA 6Ø POKE 18,Ø:POKE 19,Ø:POKE 2Ø,Ø
LH 69 REM SET UP JACKHAMMER
JO 7Ø SIZE1=16:SC=Ø:C=Ø
IM 8Ø RESTORE
BI 9Ø FOR ROWS=1 TO SIZE1
OH 1ØØ READ DOTS
GI 11Ø HAM$(ROWS,ROWS)=CHR$(DOTS)
BN 12Ø NEXT ROWS
EO 13Ø DATA Ø,Ø,Ø,Ø,Ø
BI 14Ø DATA 112,32,32,32,32,32
FA 15Ø DATA Ø,Ø,Ø,Ø,Ø
NA 16Ø FOR ROWS=1 TO 128:CLEAR$(ROWS,ROWS)=CHR
      $(Ø):NEXT ROWS
NF 17Ø A=4*(INT(PEEK(742)/4)-1)
EA 18Ø POKE 54279,A
NA 19Ø VSA=256*PEEK(135)+PEEK(134)
KK 2ØØ BOA=256*PEEK(141)+PEEK(14Ø)
DI 21Ø PM=256*A+512
JN 22Ø DISP=PM-BOA
MN 23Ø ADD=2
MI 24Ø PMHIGH=INT(DISP/256)
EE 25Ø PMLOW=DISP-256*PMHIGH
GA 26Ø POKE VSA+ADD,PMLOW
OL 27Ø POKE VSA+ADD+1,PMHIGH
GJ 28Ø DISP=DISP+128:ADD=ADD+8
FF 29Ø PM1$=CLEAR$
GD 3ØØ POKE 559,46:POKE 53277,7:POKE 623,4Ø
BJ 31Ø COLR1=25:COLR2=11:COLR3=74
```

```
BE 320 POKE 704,COLR1:POKE 705,COLR2:POKE 706,
       COLR3
PK 328 REM MAIN GAME LOOP
MK 329 REM POSITION JACKHAMMER
KJ 330 X1=125:Y1=20+L3*L2
IB 340 POKE 53248,X1:IF Y1>0 THEN PM1$(Y1)=HAM
       $:GOTO 360
JA 350 Y1=1
IP 359 REM MOVE JACKHAMMER
OP 360 Y1=Y1+Q:IF PEEK(53252) THEN GOSUB 460
AE 370 IF Q=-1 THEN C=C+1:IF C>L3 THEN Q=1:C=0
EJ 380 X1=X1+L1*(X1<47)-L1*(X1>206):IF PEEK(19
       )>18-L4 THEN SETCOLOR 4,3,2
BK 390 IF PEEK(19)>20-L4 THEN POKE 712,114:GOT
       O 500
CP 400 GOSUB 420:Y1=Y1+L2*Q:SOUND 0,Y1+5,10,10
       :IF Y1>103 THEN POKE 712,20:GOTO 500
GF 410 GOTO 340
HA 419 REM GET STICK INPUT
CF 420 S=STICK(0)
MB 430 DX=(S=5 OR S=6 OR S=7)-(S=9 OR S=10 OR
       S=11)
FF 440 X1=X1+L1*DX
HJ 450 RETURN
GB 459 REM ERASE BLOCK
CP 460 POKE 53278,1
DH 470 PLOT (X1*0.245)-11,(Y1*0.2352)-2:PLOT (
       X1*0.2453)-11,(Y1*0.2352)-1:Q=-1
PN 480 SOUND 0,100,2,10
HN 490 RETURN
OA 499 REM GAME OVER
GC 500 SOUND 0,0,0,0
FI 510 HAM$=CLEAR$
FP 515 POKE 77,0:POKE 53248,0
FM 520 POKE 53277,0:FOR X=4 TO 23:FOR Y=1 TO 3
       9:LOCATE Y,X,Z:SOUND 0,10,10,14*Z:SC=SC
       +Z:SOUND 0,0,0,0:NEXT Y:NEXT X
KM 530 GRAPHICS 1+16
LB 540 SF=SF+(60-SC)*100*L5:IF SC=0 THEN POSIT
       ION 1,1:PRINT #6;"YOU GET AN EXTRA":GOT
       O 860
GH 550 POSITION 2,1:PRINT #6;"score:";SF
HA 560 POSITION 1,5:PRINT #6;"press [OPTION] to"
       :POSITION 1,6:PRINT #6;"change speed."
FE 570 L5=0:POSITION 1,10:PRINT #6;"press [SELE
       CT] to"
GG 580 POSITION 1,11:PRINT #6;"start game."
JH 590 P=PEEK(53279):IF P=3 THEN SF=0:GOSUB 68
       0:GOTO 30
PD 600 IF P=6 THEN SF=0:GOTO 30
```

```
GO 610 GOTO 590
CD 619 REM TITLE PAGE
KN 620 GRAPHICS 2+16
BH 630 POSITION 3,3:PRINT #6;"granite cracker"
       :POSITION 2,6:PRINT #6;"[illegible]
       [illegible]"
MF 640 POSITION 5,10:PRINT #6;"PRESS [illegible]":PO
       SITION 6,11:PRINT #6;"TO BEGIN "
CD 650 FOR I=0 TO 15:SETCOLOR 0,I,I:SETCOLOR 1
       ,15-I,15-I:SOUND 0,I*10,4,10:SOUND 1,J,
       6,10:J=J+1:J=J*(J<>100):NEXT I
CN 660 IF PEEK(53279)=6 THEN RETURN
HB 670 GOTO 650
CE 679 REM CHANGE PARAMETERS
HJ 680 GRAPHICS 1:POSITION 3,1:PRINT #6;"HORIZ
       ONTAL SPEED"
HI 685 SOUND 0,0,0,0:SOUND 1,0,0,0
BD 690 POSITION 4,5:PRINT #6;"VERTICAL SPEED"
LO 700 POSITION 4,8:PRINT #6;"BOUNCE FACTOR"
AB 710 POKE 710,0:PRINT "PRESS OPTION FOR HORI
       ZONTAL SPEED":PRINT "PRESS SELECT FOR V
       ERTICAL SPEED .";
GO 720 PRINT "USE  THE JOYSTICK FOR BOUNCE FAC
       TOR":POKE 752,1
PC 730 PRINT "PRESS START TO START";
KH 740 POSITION 10,2:PRINT #6;L1
KN 750 POSITION 10,6:PRINT #6;L2
DB 760 POSITION 10,9:PRINT #6;L3;" "
GL 770 SOUND 0,0,0,0
AC 780 P=PEEK(53279):IF P=7 THEN 820
ON 790 IF P=3 THEN L1=1+(L1=1):SOUND 0,L*100+5
       0,10,10:FOR I=1 TO 20:NEXT I:GOTO 740
BK 800 IF P=5 THEN L2=1+(L2=1):SOUND 0,L*100+1
       50,10,10:FOR I=1 TO 20:NEXT I:GOTO 740
HJ 810 RETURN
MO 820 IF STICK(0)=15 THEN GOTO 740
AI 830 S=STICK(0):L3=L3-(S=13)+(S=14):L3=L3+13
       *(-(L3=14)+(L3=0))
LA 840 SOUND 0,L3*19,12,10
HB 850 GOTO 740
MO 859 REM EXTRA SCREEN
OC 860 POSITION 1,2:PRINT #6;"screen!"
KP 865 FOR I=1 TO 30:FOR J=1 TO 20*RND(1):SOUN
       D 0,I+10*RND(1),10,8:NEXT J:NEXT I
GP 870 SOUND 0,0,0,0:POKE 19,0:L4=L4+2
HJ 880 SC=0:GOTO 30
```

Pits of Pluto

Michael Dean, Jeff Destefano, Jeff Scammaca, and Timothy Shaw

Using simple graphics and scrolling, "Pits of Pluto" demonstrates some interesting BASIC programming techniques. Requires a joystick.

Your mission is to move to the bottom of the pit without running off the screen and avoid the obstacles that seem to appear as you move lower in the pit. You'll have to press your control (the joystick button) to see if you've reached the bottom.

Selecting a Pit

When the game starts, pick the level of difficulty by pressing SELECT. Levels 1 and 2 are not very difficult, but don't be fooled by the ease of the lower levels. Levels 7, 8, and 9 will make you a bit more humble. Choose random or staggered laser gates by pressing OPTION. Pressing the START key begins your mission.

You have only one chance.

The higher the level, the longer and more difficult the course. Besides red missiles, blue pods are added at level 4, and anti-jeff bombs are included at level 7 and up.

When you reach the core of Plutonian Headquarters, fly to the bottom and use the joystick trigger.

Pits of Pluto

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
PI 10  GRAPHICS 23:ST=PEEK(741)+256*PEEK(742):P
       OKE ST+8,7:POKE ST+9,6
DE 20  DIM A$(256),J(16,2):POKE 559,46:POKE 704
       ,24:POKE 53260,3:I=PEEK(106)-24:POKE 542
       79,I:POKE 53277,3
NB 30  POKE 53256,1:VT=PEEK(134)+256*PEEK(135):
       AT=PEEK(140)+256*PEEK(141):OF=I*256+384-
       AT:T=1:L=1
GA 40  HI=INT(OF/256):LO=OF-HI*256:POKE VT+2,LO
       :POKE VT+3,HI
PA 45  FOR X=1 TO 15:READ A:J(X,1)=A:NEXT X:FOR
        X=1 TO 15:READ A:J(X,2)=A:NEXT X
KO 50  COLOR 2:PLOT 0,3:DRAWTO 0,95:PLOT 159,3:
       DRAWTO 159,95
```

```
CF 65  SETCOLOR Ø,Ø,1Ø:SETCOLOR 1,8,6:SETCOLOR
       2,1,12:COLOR 1:FOR X=1 TO 43:PLOT RND(Ø)
       *159,RND(Ø)*9Ø+5:NEXT X
DJ 66  SOUND Ø,Ø,Ø,Ø
LD 7Ø  POKE 87,Ø:POSITION 4,2:? "[redacted]"
       :POSITION 22,2:? S:POSITION 34,2:? HS:PO
       KE 87,7:SETCOLOR 3,12,8
ED 75  COLOR 2:PLOT Ø,95:DRAWTO 58,95:PLOT 9Ø,9
       5:DRAWTO 159,95:PLOT Ø,3:DRAWTO 159,3
CI 8Ø  IF PEEK(53279)=3 THEN T=-T
HN 84  IF T=1 THEN SETCOLOR 3,12,8
HP 88  IF T=-1 THEN SETCOLOR 3,4,8
NN 9Ø  IF PEEK(53279)=6 OR STRIG(Ø)=Ø THEN GOTO
        99
MH 95  POKE 87,Ø:POSITION 29,2:? "[redacted]";L:IF PEEK(
       53279)=5 THEN L=L+1:IF L=1Ø THEN L=1
BA 96  GOTO 8Ø
CJ 99  X=12Ø:Y=4Ø:N=59:M=Ø:Z=3Ø
KE 1ØØ  IF Y>112 THEN Y=112
BF 1Ø1  A$(Y+128,Y+144)="{2 ,}<fB{BACK S}<f[redacted]
       {2 ,}":POKE 53248,X:SOUND Ø,Y*2+3Ø,8,8
PD 1Ø5  SETCOLOR 2,Ø,Ø:SOUND 1,Ø,Ø,Ø
LL 11Ø  X=X+J(STICK(Ø),1):Y=Y+J(STICK(Ø),2):POK
       E 53278,Ø
CM 115  Z=Z-Ø.Ø4:S=S+1:IF Z<28-L THEN GOTO 2ØØ
KP 12Ø  PLOT Ø,4:POKE 87,Ø:POSITION Ø,4:? "
       {DEL LINE}":POSITION 22,2:? S:POKE 87,7
       :N=N+M:COLOR 1:PLOT Ø,95:DRAWTO N-1,95
PA 122  X=X+J(STICK(Ø),1):Y=Y+J(STICK(Ø),2)
PA 125  PLOT N+Z+1,95:DRAWTO 159,95:COLOR 2:PLO
       T N-1,95:PLOT N+Z+1,95:IF RND(Ø)>Ø.94 T
       HEN M=Ø
HE 13Ø  IF N>1Ø9 OR RND(Ø)>Ø.99 THEN M=-1
MJ 132  IF RND(Ø)>Ø.98 THEN K=1:COLOR 3:PLOT N,
       95:DRAWTO N+Z,95:PLOT N,94:DRAWTO N+Z,9
       4:COLOR 2:PLOT N-5,95
OI 134  IF K=1 THEN DRAWTO N,95:PLOT N+Z,95:DRA
       WTO N+Z+5,95:PLOT N-5,94:DRAWTO N,94:PL
       OT N+Z,94:DRAWTO N+Z+5,94:K=Ø
OF 135  IF N<5 OR RND(Ø)>Ø.99 THEN M=1
NM 14Ø  G=G+1:IF T=1 AND G=INT(Z/3) THEN SETCOL
       OR 2,1,1Ø:SOUND 1,22,14,1Ø:G=Ø
FA 145  IF T=-1 AND INT(RND(Ø)*(Z/3))=INT(Z/3)
       THEN SETCOLOR 2,1,1Ø:SOUND 1,22,14,1Ø
AI 15Ø  C=PEEK(53252):IF C=2 OR C=3 OR C=5 OR C
       =7 THEN GOTO 6ØØ
GC 155  IF C=4 AND PEEK(71Ø)=26 THEN GOTO 6ØØ
BI 17Ø  IF PEEK(53279)=6 THEN GOTO 6ØØ
GN 199  GOTO 1ØØ
OL 2ØØ  H=N+Z:Z=Ø
```

```
KL 210 A$(Y+128,Y+144)="{2 ,}<fB{BACK S}<f
       {2 ,}":POKE 53248,X:X=X+J(STICK(Ø),1):Y
       =Y+J(STICK(Ø),2):SOUND Ø,Y*2+3Ø,8,8
CA 22Ø PLOT Ø,4:POKE 87,Ø:POSITION Ø,4:? "
       {DEL LINE}":POSITION 22,2:? S:POKE 87,7
       :S=S+1
EF 23Ø POKE 53278,Ø:IF N>1Ø THEN N=N-1
JL 235 IF H<15Ø THEN H=H+1
MC 24Ø COLOR 1:PLOT Ø,95:DRAWTO N,95:PLOT H,95
       :DRAWTO 159,95:COLOR 2:PLOT N,95:PLOT H
       ,95
GF 247 Z=Z+1:IF Z>9Ø AND N<12 AND H>148 THEN G
       OTO 3ØØ
DN 25Ø IF PEEK(53252)<>Ø AND PEEK(53252)<>4 TH
       EN GOTO 6ØØ
GI 255 GOTO 21Ø
PD 3ØØ Z=Ø:SETCOLOR 2,4,6
LA 3Ø5 A$(Y+128,Y+144)="{2 ,}<fB{BACK S}<f
       {2 ,}":POKE 53248,X:X=X+J(STICK(Ø),1):Y
       =Y+J(STICK(Ø),2):SOUND Ø,Y*2+3Ø,8,8
CA 31Ø PLOT Ø,4:POKE 87,Ø:POSITION Ø,4:? "
       {DEL LINE}":POSITION 22,2:? S:POKE 87,7
       :S=S+1
IP 32Ø COLOR 1:PLOT Ø,95:DRAWTO N,95:PLOT H,95
       :DRAWTO 159,95:COLOR 2:PLOT N,95:PLOT H
       ,95:POKE 53278,Ø
CJ 325 IF L>3 AND RND(Ø)>Ø.9-L/99 THEN COLOR 2
       :W=RND(Ø)*133+13:PLOT W,94:PLOT W+2,94:
       PLOT W+1,95:PLOT W+1,93
ED 33Ø IF RND(Ø)>Ø.9-(Z/5Ø)/99 THEN W=RND(Ø)*1
       34+12:COLOR 3:PLOT W,95:DRAWTO W,92:PLO
       T W+2,95:DRAWTO W+2,92:K=1
OC 335 IF K=1 THEN PLOT W+1,94:DRAWTO W+1,9Ø:K
       =Ø
KP 337 IF L>6 AND RND(Ø)>Ø.9-L/99/2 THEN W=RND
       (Ø)*12Ø+13:COLOR 1:PLOT W,81:DRAWTO W,8
       1+L:PLOT W-L/2,81+L/2:K=1
HE 338 IF K=1 THEN DRAWTO W+L/2,81+L/2:K=Ø
CA 34Ø Z=Z+1:IF Z>L*5Ø THEN GOTO 35Ø
EI 343 IF PEEK(53252)<>Ø THEN GOTO 6ØØ
GN 345 GOTO 3Ø5
CI 35Ø FOR Q=1 TO 9Ø:PLOT Ø,4:POKE 87,Ø:POSITI
       ON Ø,4:? "{DEL LINE}":POKE 87,7:POKE 53
       278,Ø
KN 352 IF Y>112 THEN Y=112
LF 355 A$(Y+128,Y+144)="{2 ,}<fB{BACK S}<f
       {2 ,}":POKE 53248,X:X=X+J(STICK(Ø),1):Y
       =Y+J(STICK(Ø),2):SOUND Ø,Y*2+3Ø,8,8
EN 357 IF PEEK(53252)<>Ø THEN GOTO 6ØØ
```

```
PN 360  COLOR 1:PLOT 0,95:DRAWTO N,95:PLOT H,95
        :DRAWTO 159,95:COLOR 2:PLOT N,95:PLOT H
        ,95:NEXT Q:Q=79:W=80
EB 400  COLOR 3:SETCOLOR 2,12,8:FOR V=10 TO -10
         STEP -1:K=SGN(V)
KP 408  IF Y>112 THEN Y=112
ID 410  A$(Y+128,Y+144)="{2 ,}<fB{BACK S}<f■
        {2 ,}":POKE 53248,X:X=X+J(STICK(0),1):Y
        =Y+J(STICK(0),2)
GM 420  PLOT 0,4:POKE 87,0:POSITION 0,4:? "
        {DEL LINE}":POKE 87,7
Bh 425  IF PEEK(53252)<>0 THEN POKE 53278,0:GOT
        O 600
KP 430  COLOR 1:PLOT 0,95:DRAWTO N,95:PLOT H,95
        :DRAWTO 159,95:COLOR 2:PLOT N,95:PLOT H
        ,95:COLOR 3
LN 435  Q=Q-K:W=W+K:PLOT Q,95:DRAWTO W,95:NEXT
        V
CC 450  FOR V=1 TO 25:N=N+1:H=H-1
HC 453  PLOT 0,4:POKE 87,0:POSITION 0,4:? "
        {DEL LINE}":POKE 87,7
MK 455  COLOR 1:PLOT 0,95:DRAWTO N,95:PLOT H,95
        :DRAWTO 159,95:COLOR 2:PLOT N,95:PLOT H
        ,95
BM 457  IF PEEK(53252)<>0 THEN POKE 53278,0:GOT
        O 600
LE 458  IF Y>112 THEN Y=112
LC 460  A$(Y+128,Y+144)="{2 ,}<fB{BACK S}<f■
        {2 ,}":POKE 53248,X:X=X+J(STICK(0),1):Y
        =Y+J(STICK(0),2):SOUND 0,Y*2+30,8,8
EG 465  NEXT V:COLOR 2:PLOT N,95:DRAWTO H,95:CO
        LOR 1:FOR V=1 TO 25
HH 467  PLOT 0,4:POKE 87,0:POSITION 0,4:? "
        {DEL LINE}":POKE 87,7
LF 468  IF Y>112 THEN Y=112
JB 469  A$(Y+128,Y+144)="{2 ,}<fB{BACK S}<f■
        {2 ,}":POKE 53248,X:X=X+J(STICK(0),1):Y
        =Y+J(STICK(0),2)
LH 470  PLOT 0,95:DRAWTO 159,95:NEXT V
KI 500  IF Y>112 THEN Y=112
KO 510  A$(Y+128,Y+144)="{2 ,}<fB{BACK S}<f■
        {2 ,}":POKE 53248,X:X=X+J(STICK(0),1):Y
        =Y+J(STICK(0),2):SOUND 0,Y*2+30,8,8
EI 550  IF PEEK(53252)<>0 THEN GOTO 600
JK 560  IF X<120 AND STRIG(0)=0 THEN GOTO 700
HF 599  GOTO 500
HK 600  REM
GI 610  POKE 87,0:IF S>HS THEN HS=S:POSITION 34
        ,2:? HS
```

```
ML 620  SOUND 0,0,0,0:Z=40:S=0:POKE 704,72:FOR
        V=1 TO 9:A$(V+128+Y)=CHR$(RND(0)*255):S
        OUND 1,V*5,6,10:NEXT V
KC 630  SOUND 1,0,0,0:POSITION 0,0
OG 640  IF PEEK(53279)=6 OR STRIG(0)=0 THEN POK
        E 704,10:? "{CLEAR}":POKE 87,7:A$(128+Y
        ,144+Y)="{11 ,}":GOTO 65
HL 699  GOTO 640
HL 700  REM
LO 710  POKE 87,0:POSITION 4,2:? "YOU WIN PLUTO
        ":H=0
LE 720  HS=L*500+500:POSITION 34,2:? HS:POSITIO
        N 22,2:? L*500+500:POSITION 0,0
JB 730  H=H+1:IF H=255 THEN H=0
MF 740  Y=112
PP 750  IF PEEK(53279)=6 THEN POKE 704,10:? "
        {CLEAR}":POKE 87,7:A$(128+Y,144+Y)="
        {12 ,}":Z=40:S=0:GOTO 65
DE 760  SOUND 0,H,14,8:POKE 710,H
HB 770  GOTO 730
HL 999  DATA 0,0,0,0,1,1,1,0,-1,-1,-1,0,0,0,0,0
        ,0,0,0,1,-1,0,0,1,-1,0,0,1,-1,0
```

The Catacombs of Your Mind

Doug Wheeler

This graphics-based adventure game has all the features of an adventure plus graphics that produce a visual image of the games maze. Requires 32K without a disk drive (48K if you use a disk) and a joystick.

You are about to enter a world that exists only within your mind and on the video screen before you. It takes the form of a large, complex maze and contains many monsters. The methods you need to destroy these monsters are also found within the catacombs.

The opaque green door leading to the outside is composed of an almost indestructible material. It can be destroyed only by magical means.

All items needed to complete your journey must be found within the boundaries of the maze. No items will be brought into the maze and none will leave.

You must perform a very specific set of steps before you can pass through the door which leads out of this world and back to reality.

The Adventure

This is a graphics-based adventure game for Atari computers with at least 32K bytes of memory (48K if you are using a disk).

In calling this a *graphics-based* adventure game, I'm saying that it has all the features of a text adventure, plus graphics. The graphics consist of the view from your position within a mazelike labyrinth where you can see four locations ahead of you. A location is any position at which you may reside and where hallways may lead off in other directions (other than the direction you are currently facing).

You move about this labyrinth by using a joystick plugged into port 1. Pushing the joystick forward will move you one location in the direction you are facing. Moving the joystick to the left or right turns you in the corresponding direction, while you remain in your current location. Pulling the joystick toward you causes you to turn around and face the opposite direction. If you should fall into one of the many

pits located throughout the catacombs, you will fall into another of the 360 locations in the catacombs, and you may even change direction.

Text Input

If, at any time, you wish to do something other than move around with the joystick, type in a command on the keyboard of the computer and press RETURN. This command must consist of two words: a verb and a noun, in that order. The objects required to complete your journey are placed at certain locations in the catacombs. The objects are inside yellow boxes. By moving to the same location as a box, you can open it and examine its contents. If you type OPEN BOX, the computer will respond with a description of the object contained in that box. You can then get the object by typing GET followed by the name of the object. After playing the game several times, you may get to know where certain objects are, in which case you can move to the location of the object and get it without having to open the box.

Many other verbs and nouns exist, but part of the fun of playing any adventure is discovering what you can and cannot do in any given situation.

Two other commands which you should know are INVENTORY (or INV) and QUIT. INV gives you an inventory of all the items that you possess, including the items you are wearing. You have a limit of ten items which can be held at once. QUIT allows you to end your current game. After typing QUIT, you're asked to verify your command, type in your answer, and press RETURN. If you type YES, the program will exit to BASIC. If you type NO, you will be returned to your present location within the labyrinth.

Monster Encounters

Monsters are located throughout the catacombs. Their only purpose is to try to destroy you. When you meet one of these monsters, there is usually only one or two things you can do to destroy it. You may not leave that location until either you or the monster is destroyed. Any attempt to leave is interpreted as an offensive action and will only speed your death.

Another problem is that you have only three turns in which to destroy the monster—or it'll get you.

Typing Note

Lines 10000–10419 contain the DATA for the rooms. Each DATA line has just four numbers. Don't change the line numbers or try to consolidate the DATA into fewer lines. The RESTORE statement is used to READ the proper DATA for the current room position.

The Catacombs of Your Mind

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
OI Ø  DIM A$(15):GOSUB 32676
ML 1  GOTO 1Ø
MH 4  IF P$="" THEN RETURN
FF 5  FOR P=N1 TO LEN(P$):J=ASC(P$(P,P)):? CHR$
      (ABS(J+(187-N2*J)*(J>64 AND J<123)));:NEX
      T P:? :RETURN
DE 6  V$="":IF VØ$="" THEN RETURN
CM 7  FOR P=N1 TO LEN(VØ$):J=ASC(VØ$(P,P)):V$(L
      EN(V$)+N1)=CHR$(ABS(J+(187-N2*J)*(J>64 AN
      D J<123))):NEXT P:RETURN
OM 1Ø  NØ=Ø:N1=1:N2=2:N3=3:N4=4:CR=192:CD=N1:NO
       =NØ:POKE 764,255:N2Ø5=2Ø5:MN=5:KC=Ø
OD 15  DIM V$(2Ø),VØ$(2Ø),P$(3ØØ),D(N4,N4),P(4)
       ,P1(4),PD(4)
AE 2Ø  GOSUB 2ØØØØ:REM {C}{E}INITIALIZE OBJECT
       AND MONSTER TABLES{Z}{C}
CG 1ØØ  GRAPHICS 7:GOSUB 32676:COLOR N3:POKE 75
        2,1:FOR A=Ø TO 4:FOR B=Ø TO 4:D(A,B)=Ø:
        NEXT B:NEXT A
GK 1Ø5  FOR A=Ø TO 4:IF P(A)=CR AND O(11)<>-2 T
        HEN CR=P1(A):CD=PD(A):GOTO 11Ø
BI 1Ø7  NEXT A
CP 11Ø  RESTORE 1ØØØØ+CR:READ A,B,C,D:D(NØ,N1)=
        A:D(NØ,N2)=B:D(NØ,N3)=C:D(NØ,N4)=D
GM 115  IF D(NØ,CD)=NØ THEN 15Ø
DH 12Ø  FOR E=N1 TO N4:RESTORE 1ØØØØ+D(E-N1,CD)
        :READ A,B,C,D:D(E,N1)=A:D(E,N2)=B:D(E,N
        3)=C:D(E,N4)=D
DE 125  IF D(E,CD)=NØ THEN 15Ø
BI 13Ø  NEXT E
CJ 15Ø  IF CD=N1 THEN 2ØØ
GB 153  TRAP 1ØØ
MC 155  FOR B=1 TO CD-1:FOR A=Ø TO 4:C=D(A,N1):
        D(A,N1)=D(A,N2):D(A,N2)=D(A,N3):D(A,N3)
        =D(A,N4):D(A,N4)=C:NEXT A:NEXT B
MH 156  TRAP 4ØØØØ
IK 2ØØ  GOSUB 31ØØØ:GOSUB 91ØØ
```

```
JL 205 ST=STICK(Ø):IF ST<>15 AND MF<>Ø THEN GO
       SUB 9ØØØ:KC=KC-1:GOTO N2Ø5
NL 21Ø IF ST=11 THEN CD=CD+1:IF CD>4 THEN CD=C
       D-4
LA 215 IF ST=7 THEN CD=CD-1:IF CD<1 THEN CD=CD
       +4
NP 22Ø IF ST=13 THEN CD=CD+2:IF CD>4 THEN CD=C
       D-4
GH 23Ø IF ST=14 AND D(NØ,N1)<>Ø THEN CR=D(NØ,N
       1)
AM 24Ø GOSUB 15ØØØ
AG 25Ø ON (ST=15)+1 GOTO 1ØØ,N2Ø5
FC 3ØØ IF LEN(VØ$)<3 THEN VØ$(LEN(VØ$)+1)=" ":
       GOTO 3ØØ
AB 3Ø2 ? "{CLEAR}";:GOSUB 6:VØ$=V$(LEN(V$)-2):
       V$=V$(1,3)
GP 3Ø5 IF V$<>"1kv" THEN 4ØØ
JK 31Ø IF VØ$<>"y1c" THEN 35Ø
LH 315 FOR A=1 TO 19:IF O(A)<>CR THEN NEXT A:G
       OTO 349
HP 32Ø ? "The box contains a: ";:P$=O$((A-1)*1
       5+1,(A-1)*15+15)
NJ 321 IF P$(1,1)=" " THEN P$=P$(2):GOTO 321
KG 325 GOSUB 4:GOTO N2Ø5
IB 347 ? "{CLEAR}{DOWN}You can't do that!":FOR
        A=1 TO 2ØØ:NEXT A:GOTO N2Ø5
NH 348 ? "{CLEAR}{DOWN}You don't have that!":F
       OR A=1 TO 2ØØ:NEXT A:? "{CLEAR}";:GOTO
       N2Ø5
CB 349 ? "{CLEAR}{DOWN}I don't see that here!"
       :FOR A=1 TO 2ØØ:NEXT A:? "{CLEAR}";:GOT
       O N2Ø5
LH 35Ø IF (VØ$<>"1xp" AND VØ$<>"11i") OR CR<>4
       3 OR CD<>N1 THEN 349
AP 355 IF O(17)<>Ø THEN P$="bLF WLM'G SZEV GSV
        PVB!":GOSUB 4:FOR A=1 TO 2ØØ:NEXT A:?
       "{CLEAR}";:GOTO N2Ø5
BA 36Ø CR=359:GOTO 1ØØ
HE 4ØØ IF V$<>"tvg" THEN 45Ø
BH 4Ø5 FOR A=NØ TO 18:IF VØ$=O$(A*15+13,A*15+1
       5) AND O(A+N1)=CR THEN B=A:GOTO 425
EO 41Ø NEXT A:IF VØ$="y1c" THEN ? "It is secur
       ely bolted to the floor.":GOTO N2Ø5
GP 42Ø GOTO 349
LL 425 GOSUB 9ØØØ:IF MF<>Ø THEN GOSUB 91ØØ:GOT
       O N2Ø5
LJ 43Ø IF NO=1Ø THEN ? "You can carry no more.
       ":GOTO N2Ø5
BL 435 ? "OK.":O(B+N1)=NØ:NO=NO+N1:GOTO N2Ø5 .
HO 45Ø IF V$<>"ivz" THEN 51Ø
```

```
KG 451 IF VØ$<>"ovg" THEN 46Ø
MH 452 IF O(7)<>NØ THEN 348
GP 455 IF O(7)=NØ THEN P$="gL TVG DSVIV GSRMV
       IVUOVXGRLM ORVH,{3 SPACES}XZOO GL GSV T
       IVZG xGSFOSF GSV DRHV.":GOSUB 4:GOTO N2
       Ø5
JO 456 IF VØ$="ovg" THEN O(7)=-1:GOTO N2Ø5
PM 46Ø IF VØ$<>"o 1" THEN 48Ø
JC 463 IF O(3)<>Ø OR CR<>273 OR CD<>1 THEN 47Ø
HL 465 SF1=SF1-1:IF SF1=Ø THEN P$="{CLEAR}
       {DOWN}gSV HXILOO WRHZKKVZIH!!":O(3)=-1
KC 467 GOTO 8ØØØ
HE 47Ø IF O(3)<>Ø THEN 347
FH 475 P$="{CLEAR}gSRH YVRMT GSV WRHRMGVTIZGRL
       M HKVOO{3 SPACES}LU GSV DRAZIW aLILZHGV
       I, RG HSLFOW YV FHVW DRHVOB!":GOSUB 4
LP 476 GOTO N2Ø5
CC 48Ø IF VØ$<>"o 2" OR O(6)<>Ø THEN 347
FG 485 IF O(6)<>Ø OR MF<>4 THEN 5ØØ
EP 49Ø O(6)=-1:GOTO 95ØØ
HB 5ØØ IF O(6)<>Ø THEN 347
KG 5Ø5 P$="gSV RMHXIRKGRLM DRGSRM HSZOO KILGVX
       G  GSV YVZIVI UILN GSV FMWVZW FKLM YVRM
       T IVZW.":GOSUB 4
LJ 5Ø6 GOTO N2Ø5
GK 51Ø IF V$<>"rme" THEN 55Ø
AB 512 GRAPHICS NØ:GOSUB 32676:POKE 752,1:? "
       {CLEAR}{2 DOWN}INVENTORY":? "{9 M}"
OL 515 FOR A=1 TO 19:IF O(A)<>Ø AND O(A)<>-2 T
       HEN 535
CF 52Ø P$=O$((A-1)*15+1,(A-1)*15+15)
OF 525 IF P$(1,1)=" " THEN P$=P$(2):GOTO 525
EM 53Ø GOSUB 4
MI 535 NEXT A:? :? "{3 SPACES}PRESS RETURN TO
       CONTINUE"
HN 54Ø CLOSE #1:OPEN #1,4,Ø,"K":GET #1,A:CLOSE
        #1:GOTO 1ØØ
LM 55Ø IF V$<>"kfg" AND V$<>"wil" THEN 58Ø
JP 555 FOR A=NØ TO 18:IF VØ$=O$(A*15+13,A*15+1
       5) AND (O(A+N1)=Ø OR O(A+N1)=-2) THEN 5
       7Ø
LL 556 IF VØ$="lhv" AND O(19)=Ø AND MF=9 THEN
       P$="gSV NLMTLLHV ZGGZXPH GSV XLYIZ.":GO
       SUB 4:O(19)=-N1:GOTO 95ØØ
KC 557 IF VØ$="orx" AND O(1Ø)=NØ AND MF=11 THE
       N O(1Ø)=-N1:GOTO 95ØØ
FG 558 IF VØ$="nli" AND O(16)=NØ AND MF=8 THEN
        P$="gSV IFHG NLMHGVI LEVI-HGFUUH SRNHV
       OU.":GOSUB 4:O(16)=-N1:GOTO 95ØØ
```

```
JC 559 IF VØ$=O$(A*15+13,A*15+15) AND O(A+N1)=
       -1 THEN 348
CN 560 NEXT A:GOTO 348
BL 570 FOR B=N1 TO 19:IF O(B)=CR THEN ? "The h
       all is too crowded.":GOTO N205
PD 575 NEXT B:O(A+N1)=CR:NO=NO-N1:? "OK.":GOTO
        N205
GO 580 IF V$<>"gsi" THEN 620
HL 581 IF VØ$="ixs" AND TL=1 AND MF=56 THEN O(
       5)=-1:GOTO 9500
AM 582 IF VØ$="vin" AND LL=N1 AND MF=56 THEN O
       (8)=-N1:GOTO 9500
OJ 583 GOSUB 9000
JM 585 IF VØ$<>"ili" THEN 600
ME 590 IF O(N1)<>Ø THEN 348
MC 595 ? "gSV NRIILI HSZGGVIH RMGL NROORLMH LU
         GRMB KRVXVH!":O(N1)=-N1:NO=NO-N1:GOSU
       B 4:GOTO N205
KH 600 IF VØ$<>"orm" THEN 616
PF 605 IF O(14)<>NØ THEN 348
IK 610 IF M(1Ø)=CR THEN NO=NO-N1:GOTO 9500
IG 615 P$="gSV QZEVORM UORVH WLDM GSV SZOO ZMW
       {3 SPACES}HSZGGVIH ZTZRMHG GSV UZI DZOO
       !":O(14)=-N1:NO=NO-N1:GOSUB 4:GOTO N205
LN 616 IF VØ$<>"mxv" THEN 555
MM 617 IF O(9)<>NØ THEN 348
BB 618 IF M(2)=CR THEN NO=NO-1:O(9)=-N1:GOTO 9
       500
KA 619 NO=NO-N1:? "OK.":GOTO N205
CL 620 IF V$="fmo" THEN 350
GO 630 IF V$<>"jfr" THEN 670
JE 645 GRAPHICS NØ:POSITION 4,12:? "Are you su
       re you want to quit";:INPUT A$:IF A$(1,
       1)="N" THEN 100
FD 650 IF A$(1,1)<>"Y" THEN 645
LJ 655 GRAPHICS NØ:POSITION 4,12:? "Would you
       like to play again";:INPUT A$:IF A$(1,1
       )="Y" THEN RUN
HM 660 GRAPHICS NØ:END
IE 670 IF V$<>"dvz" THEN 715
EC 675 GOSUB 9000:IF VØ$<>"rmt" THEN 695
EG 680 IF O(11)=Ø THEN ? "OK.":O(11)=-2:GOTO N
       205
CJ 685 IF O(11)=-2 THEN ? "You are already wea
       ring it.":GOTO N205
HH 690 GOTO 348
KL 695 IF VØ$<>"nli" THEN 347
EJ 700 IF O(16)=Ø THEN ? "OK.":O(16)=-2:GOTO N
       205
```

```
CH 7Ø5 IF O(16)=-2 THEN ? "You are already wea
       ring it.":GOTO N2Ø5
HA 71Ø GOTO 348
HP 715 IF V$<>"ivn" THEN 735
JC 72Ø IF VØ$="rmt" AND O(11)=-2 THEN ? "OK.":
       O(11)=Ø:GOTO N2Ø5
JB 725 IF VØ$="nli" AND O(16)=-2 THEN ? "OK.":
       O(16)=Ø:GOTO N2Ø5
HC 73Ø GOTO 348
DI 735 IF V$<>"ort" AND V$<>"rmt" AND V$<>"yfi
       " THEN 8Ø5
MD 74Ø IF VØ$<>"xsv" THEN 755
NC 745 IF O(12)=Ø THEN P$="gSV NZGXS UOZIVH FK
        ZMW TLVH LFG.":GOSUB 4:MN=MN-1:O(12)=-
       1*(MN=Ø):GOTO N2Ø5
HD 75Ø GOTO 248
LO 755 IF VØ$<>"ixs" THEN 775
DB 76Ø IF O(12)=Ø AND O(5)=Ø THEN ? "OK.":MN=M
       N-1:O(12)=-1*(MN=Ø):TL=1:GOTO N2Ø5
EI 765 IF O(12)<>Ø THEN P$="bLF WLM'G SZEV NZG
       XSVH.":GOSUB 4:GOTO N2Ø5
HG 77Ø GOTO 348
MH 775 IF VØ$="o 1" AND O(3)=Ø AND O(12)=Ø THE
       N P$="rG YFIMH DRGS Z YIROORZMG UOZNV."
       :GOSUB 4:O(3)=-1:GOTO N2Ø5
MK 78Ø IF VØ$="o 2" AND O(6)=Ø AND O(12)=Ø THE
       N P$="rG YFIMH DRGS Z YIROORZMG UOZNV."
       :GOSUB 4:O(6)=-1:GOTO N2Ø5
HM 785 IF VØ$="ovg" AND O(7)=Ø AND O(12)=Ø THE
       N P$="rG YFIMH DRGS Z YIROORZMG UOZNV."
       :GOSUB 4:O(7)=-1:GOTO N2Ø5
KH 79Ø IF (VØ$="gvi" OR VØ$="vzi") AND M(2)=CR
        AND TL=1 OR (MN>Ø AND O(12)=Ø) THEN 95
       ØØ
BH 795 IF VØ$="vim" AND MN>Ø AND O(8)=Ø AND O(
       12)=Ø THEN P$="gSV OZMGVIM XLMGZRMH ML
       UFVO.":GOTO N2Ø5
PC 8ØØ GOSUB 9ØØØ:GOTO 347
IK 8Ø5 IF V$<>"vzg" THEN 835
OC 81Ø GOSUB 9ØØØ
DG 815 IF VØ$="11w" AND O(4)=Ø THEN ? "OK.":O(
       4)=-1:GOTO N2Ø5
JG 82Ø IF VØ$="orx" AND O(1Ø)=Ø THEN ? "YYEEEC
       CCHHH!!!":O(1Ø)=-1:GOTO N2Ø5
KP 825 IF VØ$<>"1hv" THEN 835
CP 83Ø IF O(19)=Ø THEN P$="rG TVG'H HGFXP RM B
       LFI NLFGS ZMW BLF  IVNLEV RG... 1MV DVG
        NLMTLLHV.":GOSUB 4:GOTO N2Ø5
```

```
EB 835  IF V$="xgs" AND V0$="osf" AND O(7)<1 TH
        EN CR=2:CD=3:GOTO 100
PN 840  IF V$="h1 " AND M(6)=0 THEN CR=58:CD=N1
        :GOTO 100
FI 845  IF V$="zgg" OR V$="pro" THEN 9000
HF 850  IF V$<>"dze" THEN 870
NB 855  IF V0$="zmw" AND O(15)=0 AND MF=1 THEN
        9500
OH 860  GOSUB 9000
HK 865  GOTO 347
IG 870  IF V$="slo" AND V0$="ili" AND MF=3 THEN
         9500
HL 872  IF V$="slo" AND V0$<>"ili" THEN GOSUB 9
        000:GOTO 347
PG 875  IF V$<>"hdr" THEN GOTO N205
HL 880  IF V0$="liw" AND MF=5 THEN 9500
II 885  IF V0$="tvi" AND MF=6 THEN 9500
IG 890  IF V0$="ixs" AND MF=7 THEN 9500
OC 900  GOSUB 9000
LM 905  GOTO N205
PE 8000  GRAPHICS 0:? "{3 DOWN}{5 SPACES}You ha
         ve won!":? "{2 DOWN}You are now standi
         ng just on the edge of the trap door y
         ou fell through."
AP 8010  ? "{DOWN}Do you want to jump back into
          the pit":INPUT V$:IF V$(1,1)="Y" THEN
          RUN
BH 8020  ? "{2 DOWN}Congratulations!!":END
IG 9000  FOR A=N0 TO N10:IF V0$=M$(A*15+13,A*15
         +15) AND M(A+N1)=CR THEN 9020
PG 9005  IF V0$="gvi" AND M(A+N1)=CR THEN 9020
PK 9010  NEXT A:KC=KC+1:IF KC=3 THEN 9020
IF 9012  IF MF=0 THEN KC=0
KP 9015  RETURN
II 9020  POP :GRAPHICS 0:? "{3 DOWN}{7 SPACES}Y
         ou are dead!!":? "{3 DOWN}Would you li
         ke to play again"
PK 9025  INPUT V$:IF V$(1,1)="Y" THEN RUN
KD 9030  END
AF 9100  FOR A=N0 TO 10:IF M(A+N1)<>CR THEN NEX
         T A:GOTO 9125
BJ 9105  POKE 657,0:? "You're being attacked by
          a ";:P$=M$(A*15+1,A*15+15)
OJ 9110  IF P$(1,1)=" " THEN P$=P$(2,LEN(P$)):I
         F LEN(P$)>=2 THEN 9110
IE 9115  GOSUB 4
AD 9120  ? "defend yourself.":MF=A+1
NF 9125  IF CR<>303 OR CD<>1 THEN 9140
AK 9130  COLOR 2:PLOT 102,64:DRAWTO 102,17:DRAW
         TO 55,17:POSITION 55,64
```

```
KO 9135 POKE 765,2:XIO 18,#6,Ø,Ø,"S":RETURN
OK 9140 IF CR<>273 OR CD<>1 THEN RETURN
LO 9145 COLOR 2:PLOT 126,79:DRAWTO 126,1:DRAWT
        O 39,1:POSITION 39,79
KL 9150 POKE 765,2:XIO 18,#6,Ø,Ø,"S":RETURN
BH 95ØØ ? "The ";:P$=M$((MF-1)*15+1,(MF-1)*15+
        15)
PJ 95Ø5 IF P$(1,1)=" " THEN P$=P$(2,LEN(P$)):I
        F LEN(P$)>=2 THEN 95Ø5
ML 951Ø GOSUB 4:? "is dead!"
EM 9515 IF MF=6 THEN P$="yVULIV SV WRVH SV NFG
        GVIH 'rU BLF DZMGGL OVZEV GSV XOLHVW I
        LLN, QFHG HZB HL.":GOSUB 4
NC 9585 M(MF)=NØ:MF=NØ:KC=NØ
PF 959Ø GOTO N2Ø5
JD 9999 REM * room data *
EP 1ØØØØ DATA Ø,Ø,Ø,Ø
IJ 1ØØØ2 DATA Ø,Ø,32,3
IK 1ØØØ3 DATA Ø,2,33,Ø
LJ 1ØØ11 DATA Ø,Ø,41,12
IE 1ØØ12 DATA Ø,11,Ø,Ø
IK 1ØØ13 DATA Ø,Ø,43,Ø
MC 1ØØ14 DATA Ø,Ø,44,15
MB 1ØØ15 DATA Ø,14,Ø,16
MG 1ØØ16 DATA Ø,15,46,Ø
ML 1ØØ17 DATA Ø,Ø,47,18
MK 1ØØ18 DATA Ø,17,Ø,19
MP 1ØØ19 DATA Ø,18,49,Ø
LJ 1ØØ2Ø DATA Ø,Ø,5Ø,21
LI 1ØØ21 DATA Ø,2Ø,Ø,22
LL 1ØØ22 DATA Ø,21,Ø,23
MA 1ØØ23 DATA Ø,22,53,Ø
ML 1ØØ26 DATA Ø,Ø,56,27
AG 1ØØ27 DATA Ø,26,57,28
MP 1ØØ28 DATA Ø,27,58,Ø
IQ 1ØØ32 DATA 2,Ø,62,Ø
IP 1ØØ33 DATA 3,Ø,Ø,34
MG 1ØØ34 DATA Ø,33,64,Ø
NB 1ØØ37 DATA Ø,Ø,67,38
NA 1ØØ38 DATA Ø,37,Ø,39
MK 1ØØ39 DATA Ø,38,Ø,4Ø
PL 1ØØ4Ø DATA Ø,39,7Ø,41
CI 1ØØ41 DATA 11,4Ø,71,42
IK 1ØØ42 DATA Ø,41,Ø,Ø
JA 1ØØ43 DATA Ø,Ø,73,Ø
MH 1ØØ44 DATA 14,Ø,74,Ø
MO 1ØØ45 DATA Ø,Ø,75,46
MJ 1ØØ46 DATA 16,45,Ø,Ø
AM 1ØØ47 DATA 17,Ø,77,48
BC 1ØØ48 DATA Ø,47,78,49
```

```
EH 10049 DATA 19,48,79,50
DB 10050 DATA 20,49,80,51
MB 10051 DATA 0,50,0,52
MG 10052 DATA 0,51,82,0
MH 10053 DATA 23,0,83,0
MO 10054 DATA 0,0,84,55
MN 10055 DATA 0,54,0,56
MM 10056 DATA 26,55,0,0
JE 10057 DATA 27,0,0,0
JG 10058 DATA 28,0,0,0
MF 10062 DATA 32,0,0,63
MK 10063 DATA 0,62,0,64
EB 10064 DATA 34,63,94,65
NC 10065 DATA 0,64,95,0
NE 10067 DATA 37,0,0,68
NJ 10068 DATA 0,67,0,69
NO 10069 DATA 0,68,99,0
IK 10070 DATA 40,0,0,0
IM 10071 DATA 41,0,0,0
PF 10072 DATA 0,0,102,73
AE 10073 DATA 43,72,0,74
MM 10074 DATA 44,73,0,0
DH 10075 DATA 45,0,105,76
NG 10076 DATA 0,75,0,77
NF 10077 DATA 47,76,0,0
AD 10078 DATA 48,0,108,0
AG 10079 DATA 49,0,109,0
MF 10080 DATA 50,0,0,81
PD 10081 DATA 0,80,111,0
PE 10082 DATA 52,0,112,0
PH 10083 DATA 53,0,113,0
PK 10084 DATA 54,0,114,0
AB 10085 DATA 0,0,115,86
NJ 10086 DATA 0,85,0,87
NM 10087 DATA 0,86,0,88
AI 10088 DATA 0,87,118,0
PN 10094 DATA 64,0,124,0
NH 10095 DATA 65,0,0,96
AF 10096 DATA 0,95,126,0
AK 10097 DATA 0,0,127,98
AL 10098 DATA 0,97,128,0
GN 10099 DATA 69,0,129,100
JC 10100 DATA 0,99,0,0
BJ 10101 DATA 0,0,131,102
FB 10102 DATA 72,101,0,103
IC 10103 DATA 0,102,133,104
LI 10104 DATA 0,103,0,0
PK 10105 DATA 75,0,135,0
LO 10106 DATA 0,0,0,107
CJ 10107 DATA 0,106,137,0
```

```
AD 10100 DATA 78,0,138,0
JJ 10109 DATA 79,0,0,0
LG 10110 DATA 0,0,140,0
FF 10111 DATA 81,0,141,112
FF 10112 DATA 82,111,0,113
FL 10113 DATA 83,112,143,0
PK 10114 DATA 84,0,144,0
PN 10115 DATA 85,0,145,0
CL 10116 DATA 0,0,146,117
CM 10117 DATA 0,116,147,0
AG 10118 DATA 88,0,148,0
PL 10124 DATA 94,0,0,125
CH 10125 DATA 0,124,0,126
PP 10126 DATA 96,125,0,0
JJ 10127 DATA 97,0,0,0
HF 10128 DATA 98,0,158,129
NL 10129 DATA 99,128,159,130
IL 10130 DATA 0,129,160,131
BK 10131 DATA 101,130,0,0
CF 10132 DATA 0,0,162,133
II 10133 DATA 103,132,0,134
JC 10134 DATA 0,133,164,135
PM 10135 DATA 105,134,165,136
CP 10136 DATA 0,135,166,0
DA 10137 DATA 107,0,167,0
KA 10138 DATA 108,0,168,139
JN 10139 DATA 0,138,169,140
CD 10140 DATA 110,139,0,0
BP 10141 DATA 111,0,0,142
IO 10142 DATA 0,141,172,143
CD 10143 DATA 113,142,0,0
CK 10144 DATA 114,0,174,0
CN 10145 DATA 115,0,175,0
DA 10146 DATA 116,0,176,0
ME 10147 DATA 117,0,0,0
DG 10148 DATA 118,0,178,0
CI 10151 DATA 0,0,181,152
CJ 10152 DATA 0,151,182,0
CO 10153 DATA 0,0,183,154
CP 10154 DATA 0,153,184,0
MK 10157 DATA 0,0,0,158
DF 10158 DATA 128,157,0,0
DM 10159 DATA 129,0,189,0
CC 10160 DATA 130,0,0,161
JB 10161 DATA 0,160,190,162
PM 10162 DATA 132,161,192,163
CN 10163 DATA 0,162,0,164
JK 10164 DATA 134,163,194,0
ME 10165 DATA 135,0,0,0
DG 10166 DATA 136,0,196,0
```

```
KI 10167 DATA 137,0,197,168
KK 10168 DATA 138,167,198,0
DE 10169 DATA 139,0,0,170
JC 10170 DATA 0,169,200,171
CK 10171 DATA 0,170,0,172
IN 10172 DATA 142,171,202,0
CL 10173 DATA 0,0,203,174
CP 10174 DATA 144,173,0,0
CN 10175 DATA 145,0,205,0
DH 10176 DATA 146,0,0,177
DM 10177 DATA 0,176,0,178
DL 10178 DATA 148,177,0,0
CE 10181 DATA 151,0,211,0
JD 10182 DATA 152,0,212,183
JF 10183 DATA 153,182,213,0
JL 10184 DATA 154,0,214,185
DJ 10185 DATA 0,184,0,186
DF 10186 DATA 0,185,216,0
DP 10187 DATA 0,186,0,188
DL 10188 DATA 0,187,218,0
KG 10189 DATA 159,0,219,190
CP 10190 DATA 0,189,220,0
CH 10191 DATA 161,0,221,0
ME 10192 DATA 162,0,0,0
MK 10193 DATA 0,0,0,194
AM 10194 DATA 164,193,224,195
KF 10195 DATA 0,194,225,196
DL 10196 DATA 166,195,0,0
DJ 10197 DATA 167,0,227,0
ED 10198 DATA 168,0,0,199
DG 10199 DATA 0,198,0,200
IP 10200 DATA 170,199,0,201
BK 10201 DATA 0,200,231,0
CE 10202 DATA 172,0,232,0
CF 10203 DATA 173,0,0,204
CD 10204 DATA 0,203,234,0
CN 10205 DATA 175,0,235,0
CL 10206 DATA 0,0,236,207
CK 10207 DATA 0,206,0,208
CP 10208 DATA 0,207,238,0
FK 10209 DATA 0,0,0,0
BM 10210 DATA 0,0,240,211
IF 10211 DATA 181,210,0,212
IL 10212 DATA 182,211,242,0
CI 10213 DATA 183,0,0,214
JB 10214 DATA 184,213,0,215
JC 10215 DATA 0,214,245,216
CP 10216 DATA 186,215,0,0
MG 10217 DATA 0,0,247,0
KF 10218 DATA 188,0,248,219
```

```
DI 10219 DATA 189,218,0,0
CC 10220 DATA 190,0,0,221
PB 10221 DATA 191,220,251,222
CB 10222 DATA 0,221,0,223
CG 10223 DATA 0,222,253,0
DA 10224 DATA 194,0,254,0
DB 10225 DATA 195,0,0,226
JK 10226 DATA 0,225,256,227
AP 10227 DATA 197,226,257,228
DD 10228 DATA 0,227,0,229
CN 10229 DATA 0,228,0,230
CJ 10230 DATA 0,229,260,0
II 10231 DATA 201,0,261,232
II 10232 DATA 202,231,0,233
JC 10233 DATA 0,232,263,234
CG 10234 DATA 204,233,0,0
CN 10235 DATA 205,0,265,0
CO 10236 DATA 206,0,0,237
MG 10237 DATA 0,236,0,0
DG 10238 DATA 208,0,268,0
FN 10239 DATA 0,0,0,0
BP 10240 DATA 210,0,0,241
IO 10241 DATA 0,240,271,242
CD 10242 DATA 212,241,0,0
FI 10243 DATA 0,0,0,0
DB 10244 DATA 0,0,274,245
JI 10245 DATA 215,244,0,246
DF 10246 DATA 0,245,276,0
DG 10247 DATA 217,0,277,0
DH 10248 DATA 218,0,0,249
KF 10249 DATA 0,248,279,250
JH 10250 DATA 0,249,280,251
IO 10251 DATA 221,250,281,0
CO 10252 DATA 0,0,282,253
JE 10253 DATA 223,252,0,254
JI 10254 DATA 224,253,0,255
DD 10255 DATA 0,254,0,256
DC 10256 DATA 226,255,0,0
DJ 10257 DATA 227,0,287,0
MO 10258 DATA 0,0,0,259
DG 10259 DATA 0,258,0,260
CM 10260 DATA 230,259,0,0
JE 10261 DATA 231,0,291,262
CP 10262 DATA 0,261,292,0
DA 10263 DATA 233,0,293,0
DH 10264 DATA 0,0,294,265
KA 10265 DATA 235,264,0,266
KK 10266 DATA 0,265,296,267
KO 10267 DATA 0,266,297,268
KO 10268 DATA 238,267,298,0
```

```
GA 10269 DATA 0,0,0,0
CF 10270 DATA 0,0,300,271
JE 10271 DATA 241,270,0,272
CJ 10272 DATA 0,271,302,0
MB 10273 DATA 0,0,303,0
JL 10274 DATA 244,0,304,275
KB 10275 DATA 0,274,305,276
DI 10276 DATA 246,275,0,0
DN 10277 DATA 247,0,0,278
KN 10278 DATA 0,277,308,279
EB 10279 DATA 249,278,0,0
CE 10280 DATA 250,0,310,0
JD 10281 DATA 251,0,311,282
JM 10282 DATA 252,281,0,283
MI 10283 DATA 0,282,0,0
DE 10284 DATA 0,0,314,285
DM 10285 DATA 0,284,0,286
DP 10286 DATA 0,285,0,287
LA 10287 DATA 257,286,0,288
EF 10288 DATA 0,287,0,289
DP 10289 DATA 0,288,0,290
DC 10290 DATA 0,289,320,0
ME 10291 DATA 261,0,0,0
DE 10292 DATA 262,0,0,293
JN 10293 DATA 263,292,323,0
KD 10294 DATA 264,0,324,295
DP 10295 DATA 0,294,0,296
KJ 10296 DATA 266,295,326,0
DM 10297 DATA 267,0,327,0
NC 10298 DATA 268,0,0,0
BP 10300 DATA 270,0,0,301
BL 10301 DATA 0,300,0,302
IL 10302 DATA 272,301,332,0
JB 10303 DATA 273,0,333,304
JD 10304 DATA 274,303,334,0
DA 10305 DATA 275,0,335,0
MG 10307 DATA 0,0,337,0
DH 10308 DATA 278,0,0,309
JJ 10309 DATA 0,308,339,310
JA 10310 DATA 280,309,340,0
CF 10311 DATA 281,0,0,312
CD 10312 DATA 0,311,342,0
CI 10313 DATA 0,0,343,314
AA 10314 DATA 284,313,344,315
CK 10315 DATA 0,314,0,316
JK 10316 DATA 0,315,346,317
JO 10317 DATA 0,316,347,318
KC 10318 DATA 0,317,348,319
CN 10319 DATA 0,318,0,320
PK 10320 DATA 290,319,350,321
```

```
IK 10321 DATA 0,320,351,322
CG 10322 DATA 0,321,352,0
JJ 10323 DATA 293,0,353,324
JJ 10324 DATA 294,323,0,325
CN 10325 DATA 0,324,0,326
KD 10326 DATA 296,325,356,0
DM 10327 DATA 297,0,357,0
MM 10328 DATA 0,0,358,0
CF 10332 DATA 302,0,0,333
CG 10333 DATA 303,332,0,0
CL 10334 DATA 304,0,0,335
CM 10335 DATA 305,334,0,0
DG 10337 DATA 307,0,367,0
DN 10338 DATA 0,0,368,339
DI 10339 DATA 309,338,0,0
IM 10340 DATA 310,0,370,341
CJ 10341 DATA 0,340,371,0
CK 10342 DATA 312,0,372,0
CN 10343 DATA 313,0,373,0
JM 10344 DATA 314,0,374,345
DD 10345 DATA 0,344,0,346
KC 10346 DATA 316,345,376,0
DJ 10347 DATA 317,0,377,0
KM 10348 DATA 318,0,378,349
DG 10349 DATA 0,348,0,350
JH 10350 DATA 320,349,380,0
CK 10351 DATA 321,0,381,0
CN 10352 DATA 322,0,382,0
DA 10353 DATA 323,0,383,0
MK 10354 DATA 0,0,384,0
DK 10355 DATA 0,0,385,356
KE 10356 DATA 326,355,0,357
KK 10357 DATA 327,356,387,0
DP 10358 DATA 328,0,388,0
MD 10360 DATA 13,0,43,0
DP 10367 DATA 337,0,397,0
EA 10368 DATA 338,0,0,369
LB 10369 DATA 0,368,399,370
DC 10370 DATA 340,369,0,0
JD 10371 DATA 341,0,401,372
CP 10372 DATA 342,371,0,0
CN 10373 DATA 343,0,403,0
JP 10374 DATA 344,0,404,375
DM 10375 DATA 0,374,0,376
KM 10376 DATA 346,375,0,377
LA 10377 DATA 347,376,0,378
CA 10378 DATA 348,377,408,379
KM 10379 DATA 0,378,409,380
JK 10380 DATA 350,379,410,0
CK 10381 DATA 351,0,411,0
```

```
DE 10382 DATA 352,0,0,383
JN 10383 DATA 353,382,413,0
DD 10384 DATA 354,0,414,0
DN 10385 DATA 355,0,0,386
DL 10386 DATA 0,385,416,0
DM 10387 DATA 357,0,417,0
DP 10388 DATA 358,0,418,0
DD 10389 DATA 389,389,389,389
EG 10397 DATA 367,0,0,398
EL 10398 DATA 0,397,0,399
EK 10399 DATA 369,398,0,0
LI 10400 DATA 0,0,0,401
IJ 10401 DATA 371,400,0,402
CF 10402 DATA 0,401,416,0
MD 10403 DATA 373,0,0,0
MF 10404 DATA 374,0,0,0
MC 10405 DATA 0,0,0,406
CN 10406 DATA 0,405,0,407
DA 10407 DATA 0,406,0,408
DI 10408 DATA 378,407,0,0
MP 10409 DATA 379,0,0,0
LP 10410 DATA 380,0,0,0
CI 10411 DATA 381,0,0,412
CE 10412 DATA 0,411,0,413
CM 10413 DATA 383,412,0,0
DB 10414 DATA 384,0,0,415
CN 10415 DATA 0,414,0,416
AH 10416 DATA 386,415,402,417
KF 10417 DATA 387,416,0,418
DL 10418 DATA 388,417,0,0
CN 10419 DATA 389,389,389,389
IE 15000 IF PEEK(764)=255 THEN RETURN
IG 15005 V0$="":C=1:? "{CLEAR}";
DH 15010 INPUT V0$:GOTO 300
NB 20000 DIM B$(15),SP$(15),O$(300),M$(165),O(
         20),M(11):SP$="{15 SPACES}":RESTORE 25
         000
NF 20005 FOR A=N1 TO 19:READ B:O(A)=B:READ A$
MD 20010 GOSUB 21000:O$((A-N1)*15+N1,(A-N1)*15
         +15)=A$:NEXT A
ML 20050 FOR A=N1 TO 11:READ B:M(A)=B:READ A$
MG 20060 GOSUB 21000:M$((A-N1)*15+N1,(A-N1)*15
         +15)=A$:NEXT A
NJ 20100 FOR A=0 TO 4:READ B,C,D:P(A)=B:P1(A)=
         C:PD(A)=D:NEXT A
NE 20200 RETURN
OP 21000 IF LEN(A$)<15 THEN B$=SP$(1,15-LEN(A$
         )):B$(LEN(B$)+1)=A$:A$=B$
NI 21005 RETURN
```

```
BD 25000 DATA 126,nriili,71,nztrx hdliw,13,hxi
         loo 1,258,yzhpvg lu ullw,255,glixs,35
         4,hxiloo 2
OL 25010 DATA 106,hnzoo ovzuovg,193,ozmgvim,32
         8,hroevi ozmxv,147,yfw lu tziorx,237,
         nztrx irmt,307,ylc lu nzgxsvh
FC 25020 DATA 57,nztrx wzttvi,109,uoznrmt qzev
         orm,403,dzmw,405,hfrg lu zinli,304,tl
         ow pvb,110,yzhpvg lu ullw
FA 25050 DATA 295,nlmtllhv,73,wqrmmr,236,dvivy
         vzi,340,yzhrorhp,382,hkvxgiv,116,ivw
         wiztlm,79,hszwld,367,lxsivqvoob
NF 25060 DATA 69,ifhg nlmhgvi,378,xlyiz,286,tr
         zmg sbwiz,181,eznkriv
IK 25100 DATA 241,332,1,343,206,3,384,409,1,23
         8,419,1,298,354,3
FO 31000 IF D(N0,N1)<>N0 THEN 31020
JM 31005 PLOT 38,N0:DRAWTO 127,N0
ED 31010 IF D(N0,N4)<>N0 THEN PLOT 127,N0:DRAW
         TO 159,N0
PC 31011 IF D(N0,N2)<>N0 THEN PLOT 38,N0:DRAWT
         O N0,N0
PG 31013 IF D(N0,N4)=N0 THEN PLOT 127,N0:DRAWT
         O 127,79
JH 31014 IF D(N0,N2)=N0 THEN PLOT 38,N0:DRAWTO
          38,79
DC 31019 GOTO 32015
GF 31020 IF D(N1,N2)<>N0 THEN 31050
EN 31025 PLOT 38,N0:DRAWTO 54,16:PLOT 40,79:DR
         AWTO 54,65:GOTO 31075
EF 31050 PLOT 30,N0:DRAWTO 30,79:PLOT 30,16:DR
         AWTO 54,16:IF D(N1,N1)<>N0 THEN DRAWT
         O 54,65
EC 31055 PLOT 30,65:DRAWTO 54,65
HE 31075 IF D(N1,N4)<>N0 THEN 31125
PM 31100 PLOT 119,N0:DRAWTO 103,16:PLOT 117,79
         :DRAWTO 103,65:GOTO 31150
PF 31125 PLOT 127,0:DRAWTO 127,79:PLOT 127,16:
         DRAWTO 103,16:IF D(N1,N1)<>N0 THEN DR
         AWTO 103,65
JO 31130 PLOT 127,65:DRAWTO 103,65
GF 31150 IF D(N1,N1)<>N0 THEN 31200
NP 31175 PLOT 54,16:DRAWTO 103,16:IF D(N1,N4)=
         N0 THEN DRAWTO 103,65
LC 31180 PLOT 103,65:DRAWTO 54,65:IF D(N1,N2)=
         N0 THEN DRAWTO 54,16
DI 31185 GOTO 31900
IP 31200 REM [redacted]
HG 31225 IF D(N2,N2)<>N0 THEN 31275
```

```
DB 31250 PLOT 54,16:DRAWTO 66,28:PLOT 54,65:DR
         AWTO 66,53:GOTO 31300
EO 31275 PLOT 54,16:DRAWTO 54,65:PLOT 54,28:DR
         AWTO 66,28:IF D(N2,N1)<>NØ THEN DRAWT
         O 66,53
EF 31280 PLOT 54,53:DRAWTO 66,53
GM 31300 IF D(N2,N4)<>NØ THEN 31350
JC 31325 PLOT 103,16:DRAWTO 91,28:PLOT 103,65:
         DRAWTO 91,53:GOTO 31375
MF 31350 PLOT 103,16:DRAWTO 103,65:PLOT 103,28
         :DRAWTO 91,28:IF D(N2,N1)<>NØ THEN DR
         AWTO 91,53
HB 31355 PLOT 103,53:DRAWTO 91,53
HI 31375 IF D(N2,N1)<>NØ THEN 31425
IJ 31400 PLOT 66,28:DRAWTO 91,28:IF D(N2,N4)=N
         Ø THEN DRAWTO 91,53
IM 31405 PLOT 91,53:DRAWTO 66,53:IF D(N2,N2)=N
         Ø THEN DRAWTO 66,28
CP 31410 GOTO 31900
JN 31425 REM * 3rd room *
GO 31450 IF D(N3,N2)<>NØ THEN 31500
EH 31475 PLOT 66,28:DRAWTO 74,36:PLOT 66,53:DR
         AWTO 74,45:GOTO 31525
EM 31500 PLOT 66,28:DRAWTO 66,53:PLOT 66,36:DR
         AWTO 74,36:IF D(N3,N1)<>NØ THEN DRAWT
         O 74,45
EJ 31505 PLOT 66,45:DRAWTO 74,45
HP 31525 IF D(N3,N4)<>NØ THEN 31575
DH 31550 PLOT 91,28:DRAWTO 83,36:PLOT 91,53:DR
         AWTO 83,45:GOTO 31600
FC 31575 PLOT 91,28:DRAWTO 91,53:PLOT 91,36:DR
         AWTO 83,36:IF D(N3,N1)<>NØ THEN DRAWT
         O 83,45
EK 31580 PLOT 91,45:DRAWTO 83,45
HA 31600 IF D(N3,N1)<>NØ THEN 31650
JD 31625 PLOT 74,36:DRAWTO 83,36:IF D(N3,N4)=N
         Ø THEN DRAWTO 83,45
IN 31630 PLOT 83,45:DRAWTO 74,45:IF D(N3,N2)=N
         Ø THEN DRAWTO 74,36
DI 31635 GOTO 31900
KE 31650 REM * 4th room *
IB 31675 IF D(N4,N2)<>NØ THEN 31725
EC 31700 PLOT 74,36:DRAWTO 77,39:PLOT 74,45:DR
         AWTO 77,42:GOTO 31750
FM 31725 PLOT 74,36:DRAWTO 74,45:PLOT 74,39:DR
         AWTO 77,39:IF D(N4,N1)<>NØ THEN DRAWT
         O 77,42
EF 31730 PLOT 74,42:DRAWTO 77,42
HH 31750 IF D(N4,N4)<>NØ THEN 31800
```

```
EF 31775 PLOT 83,36:DRAWTO 80,39:PLOT 83,45:DR
         AWTO 80,42:GOTO 31825
EK 31800 PLOT 83,36:DRAWTO 83,45:PLOT 83,39:DR
         AWTO 80,39:IF D(N4,N1)<>NØ THEN DRAWT
         O 80,42
EC 31805 PLOT 83,42:DRAWTO 80,42
ID 31825 IF D(N4,N1)<>NØ THEN 31875
JE 31850 PLOT 77,39:DRAWTO 80,39:IF D(N4,N4)=N
         Ø THEN DRAWTO 80,42
JH 31855 PLOT 80,42:DRAWTO 77,42:IF D(N4,N2)=N
         Ø THEN DRAWTO 77,39
DI 31860 GOTO 31900
LC 31875 PLOT 78,40:PLOT 79,40:PLOT 78,41:PLOT
          79,41
NL 31900 FOR A=NØ TO N4:IF P(A)=D(NØ,N1) THEN
         31913
JN 31901 IF P(A)=D(N1,N1) THEN 31915
KA 31902 IF P(A)=D(N2,N1) THEN 31925
KD 31903 IF P(A)=D(N3,N1) THEN 31935
PJ 31904 NEXT A:GOTO 31980
CJ 31913 PLOT 40,79:DRAWTO 54,65:DRAWTO 103,65
         :DRAWTO 117,79:PLOT 54,65:DRAWTO 54,7
         9:PLOT 103,65:DRAWTO 103,79
DM 31914 GOTO 31904
JH 31915 PLOT 54,65:DRAWTO 66,53:DRAWTO 91,53:
         DRAWTO 103,65:DRAWTO 54,65
NF 31917 PLOT 66,53:DRAWTO 66,65:PLOT 91,53:DR
         AWTO 91,65
DJ 31920 GOTO 31904
GN 31925 PLOT 66,53:DRAWTO 74,45:DRAWTO 83,45:
         DRAWTO 91,53:DRAWTO 66,53
NC 31927 PLOT 74,45:DRAWTO 74,53:PLOT 83,45:DR
         AWTO 83,53
DK 31930 GOTO 31904
GK 31935 PLOT 74,45:DRAWTO 77,42:DRAWTO 80,42:
         DRAWTO 83,45:DRAWTO 74,45
MP 31937 PLOT 77,42:DRAWTO 77,45:PLOT 80,42:DR
         AWTO 80,45
EC 31938 GOTO 31904
JD 31980 FOR A=N1 TO 19:IF O(A)=NØ THEN 31984
KK 31981 IF O(A)=D(NØ,N1) THEN 31985
KN 31982 IF O(A)=D(N1,N1) THEN 31995
JO 31983 IF O(A)=D(N2,N1) THEN 32005
PC 31984 NEXT A:GOTO 32010
PD 31985 COLOR N1:PLOT 98,65:DRAWTO 98,77:DRAW
         TO 59,77:DRAWTO 59,65:DRAWTO 98,65:DR
         AWTO 91,58:DRAWTO 66,58
MO 31987 DRAWTO 59,65:COLOR 3
EH 31990 GOTO 31983
```

```
HE 31995 COLOR 1:PLOT 86,53:DRAWTO 86,57:DRAWT
         O 71,57:DRAWTO 71,53:DRAWTO 86,53:DRA
         WTO 83,50:DRAWTO 74,50
MG 31997 DRAWTO 71,53:COLOR 3
DG 32000 GOTO 31983
MM 32005 COLOR 1:PLOT 81,45:DRAWTO 81,47:DRAWT
         O 76,47:DRAWTO 76,45:DRAWTO 81,45:PLO
         T 77,44
LF 32007 DRAWTO 80,44:COLOR 3
DP 32008 GOTO 31984
NG 32010 RETURN
DA 32015 IF CR<>43 OR CD<>N1 THEN RETURN
BM 32020 COLOR 2:PLOT 109,36:DRAWTO 111,36:DRA
         WTO 113,38:DRAWTO 113,41:DRAWTO 111,4
         3
KC 32025 DRAWTO 109,43:DRAWTO 107,41:DRAWTO 10
         7,38:DRAWTO 109,36
EH 32030 PLOT 109,38:DRAWTO 111,38:PLOT 109,39
         :DRAWTO 111,39:PLOT 110,40:DRAWTO 110
         ,41
KP 32035 PLOT 117,79:DRAWTO 117,16:DRAWTO 74,1
         6:DRAWTO 74,79:COLOR 3
NJ 32040 RETURN
PI 32676 POKE 764,12:POKE 16,64:POKE 53774,64:
         INPUT A$:POKE 752,1:? "{CLEAR}";:RETU
         RN
```

Double Brickout

Grant Albrecht

You'll need paddles and at least 40K RAM to play this fast-action game.

"Double Brickout" has 11 different skill levels, so even a novice (level 0) will be able to start right in and practice. If you think you're an expert, you might want to try level 10. As each screen of bricks is cleared, the computer increases the skill level by one.

To begin play, just follow the screen prompts, press START, and use your paddle to prevent the balls from passing you by.

Double Brickout

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
FK 9 IF PEEK(106)>120 THEN POKE 106,120
MA 10 BASE=38912
FP 13 POKE 559,0:GOSUB 550:GOSUB 790:DIM SCORE
      (1),B(1),L(1),R4$(40),R5$(40),R6$(40),R7
      $(40),C(1)
KD 20 POKE 1780,180:SCORE(0)=0:SCORE(1)=0:L=20
      :GOTO 390
MG 25 BASE=38912
GP 30 POKE 106,120:GRAPHICS 1+16:POKE 559,62:P
      OKE 756,140:POKE 1724,64
NL 40 POSITION 2,1:? #6;"%$$$$$$$$$$$$$$$$$&":F
      OR I=2 TO 21:POSITION 2,I:? #6;"'
      {16 SPACES}(":NEXT I
HP 50 X=USR(ADR("h%X{E}P%Y{E}@hM{D}■S{Q}PHEcP
      JHI{T}{X}eP{E}P){,}e@{E}@h8ij@JPE"))
OG 51 REM above= "h % X INV-CTRL-E P % Y INV-C
      TRL-E @ ) h M CTRL-D INV-SPACE S INV-CTR
      L-Q P H @ c P U H ) CTRL-T
MB 52 REM CTRL-X e P INV-CTRL-E P ) CTRL-, e @
       INV-CTRL-E @ h 8 i @ J P 5 CTRL-."
LN 70 COLOR 0:POKE 1722,0:POSITION 7,14:? #6;"
      PLAYER■";P+1:POSITION 3,23:? #6;"
      {5 SPACES}ready!{5 SPACES}"
PE 80 POKE 1725,12:POKE 53256,1:POKE 53249,0:P
      OKE 53250,0:FOR I=1 TO 100:X=255-PADDLE(
      P):POKE 53248,X:NEXT I
KE 90 POKE 1780,180:POKE 1781,110:POKE 1782,11
      0:FOR I=1772 TO 1774:POKE I,1:NEXT I:POS
      ITION 6,23:? #6;"{10 SPACES}"
```

```
JF 100 POSITION 7,14:? #6;"{8 SPACES}":RESTORE
        110:FOR I=1712 TO 1719:READ X:POKE I,X
       :NEXT I:POKE 706,0:POKE 705,15
BE 110 DATA 24,2,76,68,0,0,1,0
OD 120 T=RND(0)*3+1:POKE 1716,T:T=RND(0)*3+1
IG 130 POKE 1727,0:POKE 1726,P:FOR I=1 TO 80:X
       =USR(BASE):SOUND 0,0,0,0:NEXT I
NL 140 POKE 1727,1:POKE 1717,T:POKE 1719,1:POK
       E 706,15
IJ 150 X=USR(BASE):POKE 77,0:IF PEEK(53279)<>7
        THEN 390
NM 160 SCORE(P)=SCORE(P)+PEEK(1722):POKE 1722,
       0:IF PEEK(1724) THEN 220
IK 170 J=40+250*RND(0):FOR I=1 TO J:X=USR(BASE
       ):X=USR(ADR("h■A{F}■2{H}■P{E}JPW`")):NE
       XT I:POKE 1727,0
HH 171 REM above="h ■ A CTRL-F INV-SPACE 2 INV
       -CTRL-H ■ E INV-{=} J P W CTRL-."
LD 180 X=USR(ADR("h%X{E}P%Y{E}QJh"{D}■S{C}PHEc
       PYHJ{T}{X}eP{E}PJ{,}eQ{E}Qh8■@JPE`"))
OO 181 REM above="h % X INV-CTRL-E E % Y INV-C
       TRL-E C J F ■ CTRL-D ■ S INV-CTRL-Q E H
        E c E F H J CTRL-T CTRL-X
DL 182 REM e E INV-CTRL-E E J CTRL-, e C INV-C
       TRL-E Q h 8 ■ @ J E E CTRL- "
AI 200 COLOR 0:POKE 1724,64:X=USR(BASE):IF PEE
       K(53279)<>7 THEN 390
DC 210 L(P)=L(P)-1:POKE 1729,L(P):POKE 1727,1:
       X=USR(BASE):GOTO 150
MC 220 B(P)=B(P)-1:POKE 53248,0:POSITION 8,18:
       ? #6;"PLYR1 PLYR2":POSITION 3,19:? #6;"
       ball{4 SPACES}";5-B(0);"{5 SPACES}";5-B
       (1)
AL 230 POSITION 3,20:? #6;"speed{3 SPACES}":PO
       SITION 3,21:? #6;"score"
EI 235 POSITION 11-(L(0)<11),20:? #6;20-L(0);"
        ":POSITION 17-(L(1)<11),20:? #6;20-L(1
       );" "
DL 240 POSITION 9+(SCORE(0)<10)+(SCORE(0)<100)
       ,21:? #6;SCORE(0):POSITION 15+(SCORE(1)
       <10)+(SCORE(1)<100),21
AH 250 ? #6;SCORE(1):C(P)=PEEK(1724)
FK 260 J=P*19+1:FOR I=3 TO 18:LOCATE I,4,X:R4$
       (I+J)=CHR$(X):LOCATE I,5,X:R5$(I+J)=CHR
       $(X):LOCATE I,6,X
GK 270 R6$(I+J)=CHR$(X):LOCATE I,7,X:R7$(I+J)=
       CHR$(X):NEXT I:R4$(40)="{,}":R5$(40)="
       {,}":R6$(40)="{,}":R7$(40)="{,}"
PH 290 IF N THEN P=(P=0):POKE 1726,P
```

```
OM 300 IF  NOT B(P) THEN POSITION 6,14:? #6;"G
       AME  OVER":FOR I=1 TO 800:NEXT I:GOTO 3
       90
LJ 310 POSITION 6,23:? #6;"press fire":IF PEEK
       (53279)<>7 THEN 390
OM 320 IF (PTRIG(0) AND P=0) OR (PTRIG(1) AND
       P) THEN 310
FI 330 POSITION 6,23:? #6;"{10 SPACES}":J=P*19:
       IF FLAG AND N THEN FLAG=0:GOTO 30
JE 340 I=P*19+4:POSITION 3,4:? #6;R4$(I,I+15):
       POSITION 3,5:? #6;R5$(I,I+15)
JK 350 POSITION 3,6:? #6;R6$(I,I+15):POSITION
       3,7:? #6;R7$(I,I+15)
GF 360 POSITION 8,18:? #6;"{11 SPACES}":POSITIO
       N 3,19:? #6;"{15 SPACES}":POSITION 3,20
HD 370 ? #6;"{15 SPACES}":POSITION 3,21:? #6;"
       {16 SPACES}"
NH 380 POKE 1724,C(P):POKE 1729,L(P):GOTO 70
PL 390 POKE 1729,L:GRAPHICS 17:FOR I=53248 TO
       53250:POKE I,0:NEXT I:? #6:? #6:? #6;"
       {3 SPACES}double  brickout"
NL 400 ? #6;"{3 SPACES}DOUBLE  BRICKOUT":? #6
GB 410 POSITION 4,6:? #6;N+1;"  PLAYER GAME":?
        #6:? #6
GB 420 ? #6;"{5 SPACES}PRESS OPTION":? #6;"
       {4 SPACES}FOR DIFFICULTY":? #6;"
       {7 SPACES}SPEED=";20-PEEK(1729);"  ":?
       #6
OE 430 ? #6:? #6;"{5 SPACES}PRESS SELECT":? #6
       ;"{5 SPACES}FOR ";1+(N=0);" PLAYER":? #
       6:? #6:? #6;"{5 SPACES}PRESS START"
KO 440 ? #6;"{7 SPACES}TO PLAY":? #6:? #6:? #6
       ;"  PLYR 1{5 SPACES}PLYR 2"
IO 450 POSITION 4,23:? #6;SCORE(0):POSITION 15
       ,23:? #6;SCORE(1)
CD 460 IF PEEK(53279)<>7 THEN 460
OK 470 IF PEEK(53279)=5 THEN N=(N=0):GOTO 410
PC 480 IF PEEK(53279)=3 THEN L=PEEK(1729):L=(L
       -5)*(L>10)+20*(L=10):POKE 1729,L:GOTO 4
       10
AA 490 IF PEEK(53279)<>6 AND PTRIG(0) THEN 470
GG 500 B(0)=5:B(1)=5:P=0:SCORE(0)=0:SCORE(1)=0
       :L(0)=L:L(1)=L:POKE 77,0:FLAG=1:GOTO 30
GJ 550 REM  LOAD M.L. ROUTINE IN MEMORY
EN 560 RESTORE 590:I=0
OO 570 READ X:IF X<>-1 THEN POKE BASE+I,X:I=I+
       1:GOTO 570
HN 580 RETURN
```

```
FM 590 DATA 104,172,190,6,169,255,56,249,112,2
       ,201,58,176,2,169,58,201,195,144,2,169,
       195,141,0,208
GB 595 DATA 141,184,6,169,0,141,1,210,162,1,18
       9,180,6,24,125,176,6,157,176,6,74,24,10
       5,69,157
HI 600 DATA 178,6,157,1,208,189,182,6,24,125,2
       45,6,157,245,6,169,1,157,237,6,189,178,
       6,201,69
AB 605 DATA 176,12,32,48,154,169,0,56,253,180,
       6,157,180,6,189,178,6,201,194,144,12,32
       ,48,154,169
CL 610 DATA 0,56,253,180,6,157,180,6,189,245,6
       ,201,43,176,20,32,48,154,169,4,141,189,
       6,169,0
CN 615 DATA 141,8,208,56,253,182,6,157,182,6,1
       89,245,6,201,205,144,21,169,250,157,245
       ,6,169,0,157
ND 620 DATA 193,2,157,182,6,157,180,6,169,100,
       157,178,6,189,245,6,201,42,176,5,169,43
       ,157,245,6
HA 625 DATA 189,178,6,201,68,176,5,169,69,157,
       178,6,189,178,6,201,194,144,5,169,195,1
       57,178,6,189
FC 630 DATA 182,6,48,63,240,61,189,13,208,201,
       1,240,42,189,178,6,24,105,3,205,184,6,1
       44,43,56
ID 635 DATA 233,2,56,237,189,6,205,184,6,176,3
       1,189,245,6,201,178,144,24,201,204,176,
       20,201,183,144
IN 640 DATA 4,201,199,144,12,32,48,154,169,0,5
       6,253,182,6,157,182,6,202,208,6,76,35,1
       52,32,176
DH 645 DATA 154,162,1,189,178,6,201,71,144,39,
       201,193,176,35,189,5,208,240,30,141,185
       ,6,189,178,6
JN 650 DATA 56,233,4,74,74,74,56,233,5,141,192
       ,6,172,185,6,185,193,154,141,187,6,32,5
       9,154,202
CM 655 DATA 208,3,76,22,153,169,0,141,30,208,1
       41,188,6,162,4,165,88,133,208,165,89,13
       3,209,160,83
BE 660 DATA 177,208,240,3,238,188,6,200,192,99
       ,208,244,169,20,24,101,208,133,208,169,
       0,101,209,133,209
EL 665 DATA 202,208,226,162,2,189,244,6,240,89
       ,56,221,240,6,240,83,141,254,6,106,141,
       255,6,142,253
```

```
AM 670  DATA 6,24,169,0,109,253,6,24,109,252,6,
        133,204,133,206,189,240,6,133,203,173,2
        54,6,133,205
CK 675  DATA 189,248,6,170,232,46,255,6,144,16,
        168,177,203,145,205,169,0,145,203,136,2
        02,208,244,76,209
NJ 680  DATA 153,160,0,177,203,145,205,169,0,14
        5,203,200,202,208,244,174,253,6,173,254
        ,6,157,240,6,189
EE 685  DATA 236,6,240,48,133,203,24,138,141,25
        3,6,109,235,6,133,204,24,173,253,6,109,
        252,6,133,206
OG 690  DATA 189,240,6,133,205,189,248,6,170,16
        0,0,177,203,145,205,200,202,208,248,174
        ,253,6,169,0,157
LP 695  DATA 236,6,202,48,3,76,124,153,173,191,
        6,240,15,173,188,6,240,10,173,180,6,208
        ,6,173,181
HB 700  DATA 6,208,1,96,32,176,154,76,1,152,169
        ,170,141,1,210,169,100,141,0,210,96,173
        ,187,6,10
DG 705  DATA 10,141,188,6,10,10,24,109,188,6,24
        ,109,192,6,168,177,88,208,1,96,169,0,14
        5,88,32
ID 710  DATA 48,154,173,185,6,24,109,186,6,141,
        186,6,169,8,56,237,187,6,141,185,6,189,
        180,6,48
IA 715  DATA 9,173,185,6,157,180,6,76,132,154,1
        69,0,56,237,185,6,157,180,6,189,182,6,4
        8,11,169
FP 720  DATA 0,56,233,1,157,182,6,76,153,154,16
        9,1,157,182,6,189,245,6,24,125,182,6,24
        ,125,182
CM 725  DATA 6,24,125,182,6,24,125,182,6,157,24
        5,6,96,174,193,6,160,220,136,208,253,20
        2,208,248,96
DA 730  DATA 0,0,0,0,0,0,7,6,0,5,0,0,0,4,-1
EJ 790  REM {7 SPACES}SUBR, LOAD NEW CHAR *
        {10 SPACES}DEF N AT ADDR+(CHAR*8)
FB 800  CHBAS=140:ADDR=CHBAS*256:PAGE=8
BD 810  A=USR(ADR("hh{■}■h{■}Nhh{■}■{,}{■}■
        {■}{■}MW{A}■{,}■■{Q}NHPUTFMFONdTPE‘{,}")
        ,ADDR,PAGE)
BD 811  REM above= "h h INV-CTRL-E ■ h INV-CTRL
        -E ■ h h INV-CTRL-E ■ ■ CTRL-, INV-CTRL
        -E ■ ■
JP 812  REM INV-CTRL-. INV-CTRL-E ■ ■ CTRL-A IN
        V-SPACE CTRL-, ■ ■ INV-CTRL-Q ■ ■ ■ ■ ■
         ■ ■ ■ ■ ■ ■ ■ ■ CTRL-. CTRL-,
```

```
OD 820 RESTORE 850:FOR I=35904+256 TO 35904+26
       3:READ X:POKE I,X:NEXT I
PF 830 RESTORE 850:FOR I=35864 TO 35864+55:REA
       D X:POKE I,X:NEXT I
JC 840 FOR I=35929 TO 35936:POKE I,0:NEXT I
IG 850 DATA 0,254,254,254,254,254,254,254,0,0,
       0,255,0,0,0,0,0,0,0,3,2,2,2,2,0,0,0,128
       ,128,128,128,128
LE 860 DATA 2,2,2,2,2,2,2,2,128,128,128,128,12
       8,128,128,128
LO 870 REM VBLANK PM
MC 890 FOR I=1774 TO 1787:POKE I,0:NEXT I:POKE
        53257,0:POKE 53258,0
KB 900 PM=128:PMBASE=PM*256
CB 910 FOR I=PMBASE+1023 TO PMBASE+2047:POKE I
       ,0:NEXT I:DRWBAS=PMBASE+1
JD 930 RESTORE 1090:FOR I=DRWBAS TO DRWBAS+26:
       READ X:POKE I,X:NEXT I
CC 940 FOR I=DRWBAS+256 TO DRWBAS+260:READ X:P
       OKE I,X:POKE I+256,X:NEXT I:POKE 704,10
       :POKE 705,15:POKE 706,15
PP 950 POKE 1784,27:POKE 1785,5:POKE 1786,5:PO
       KE 53256,1:FOR I=1772 TO 1774:POKE I,1:
       NEXT I
FI 960 POKE 623,1:POKE 1788,PM+4:POKE 53277,3:
       POKE 54279,PM:POKE 1771,PM:RETURN
DN 1080 REM PLAYER IMAGE DATA FOLLOWS
MM 1090 DATA 0,255,255,255,255,0,0,0,0,0,0,0,0
        ,0,0,0,0,0,0,0,0,0,255,255,255,255,0
IE 1100 DATA 0,0,8,0,0
```

Saguaro

Walter, Carol, Jennifer, and Andrew Bulawa

You'll have to be fast and skillful with your joystick to grab up Saguaro eggs while avoiding the vulture.

In the desert regions of the great American Southwest grows the Saguaro cactus. Until recently the method of propagation of this plant has been a mystery to the scientific community. However, during a recent vacation to that part of the country, we discovered what could be the solution to this puzzle. Hidden under a large rock, which bore several unprintable Anasazi petroglyphs (graffiti), was a hermetically sealed mayonnaise jar in which was placed an ancient computer printout that describes the legend of the Saguaro cactus. Although we cannot, at this time, bring out any of the details of this legend, we can and must reveal that, according to this legend, the Saguaro cactus begins as an egg dropped onto the hot desert sand by the not-often-seen desert vulture. To give this legend the attention that it deserves, this program was written.

Playing Saguaro

You're a rancher over whose ranch flies the desert vulture (that's right, the same one described in the legend). As the vulture flies, it will periodically deposit eggs on the hot desert sand. You must rope these eggs before they turn into cacti and take them back to your cabin to get points. You will receive one point for each egg placed in the cabin. To rope the eggs, point the joystick in the direction you wish (right, left, or up only) and press the trigger. If an egg is at that location on the screen, it will disappear and you will be credited with having it in your possession; the egg counter will increment to record the number of eggs in hand. Don't worry, the eggs will not hatch while you're holding them. To deposit the eggs, extend your rope, as described above, to the front door of the cabin.

While you're running around your ranch chasing down eggs, you cannot, of course, walk through existing cacti (ouch!) nor the prairie-dog mounds. As the game proceeds, your movement will grow increasingly more difficult as more and more cacti fill the screen, having hatched from the eggs that you did not retrieve in time. Always try to maintain a clear path to your door. In that you cannot rope eggs immediately

below you, don't allow yourself to get trapped by having your downward path blocked.

The Heat of the Desert

Because the desert is so hot, you cannot stay out in the sun indefinitely and so a timer has been provided to show you the number of seconds that you have remaining before you must return to the cabin. If you do not make it back to the cabin and deposit at least one egg within the allotted 50-second time period, you will be carried off by the desert vulture.

Oh, another thing—while you are out gathering eggs, do not let the desert vulture grab you; for, if he does, he will carry you off and the game will be over. Good luck.

Saguaro

For error-free program entry, read "The Automatic Proofreader," Appendix C, before typing in this program.

```
FH 10 REM SAGUARO
KK 40 GOSUB 6000
KM 50 GOSUB 7000
KO 60 GOSUB 8000
GC 90 IB=1
HE 100 REM MAIN LOOP
HI 102 TIME=50-INT((PEEK(19)*256+PEEK(20))/60)
       :POSITION CMIN+6,RMAX+2:? #6;TIME;" ":I
       F TIME<=0 THEN 800
MB 110 REM ** BIRD MOTION
LO 115 READ X:IF X=-1 THEN RESTORE 590:GOTO 12
       0
MG 116 SOUND 2,X,14,2
EK 120 SOUND 0,10,10,6:MOVE=INT(9*RND(0))+1:IF
        MOVE=5 THEN SOUND 0,0,0,0:GOTO 210
EH 160 LOCATE BCOL+CDIFF(MOVE),BROW+RDIFF(MOVE
       ),X:SOUND 0,0,0,0:IF X=217 THEN 120
PC 180 POSITION BCOL,BROW:IF RND(0)<0.3 THEN I
       F BPREV=32 THEN BPREV=234
JD 188 ? #6;CHR$(BPREV):BPREV=X
CP 190 BCOL=BCOL+CDIFF(MOVE):BROW=BROW+RDIFF(M
       OVE)
MH 200 IB=-IB:POSITION BCOL,BROW:? #6;CHR$(74+
       IB)
DO 205 REM ** BIRD BITES PERSON
BL 206 IF BPREV=86 OR BPREV=87 OR BPREV=88 THE
       N TRAP 1000:GOTO 700
BC 210 REM ** SWATTING MOTION
PM 260 S=STICK(0):T=STRIG(0)
```

```
HN 280  YDIFF=-1*(S=14)+(S=13):XDIFF=-1*(S=11)+
        (S=7)
KK 300  LOCATE PCOL+XDIFF,PROW+YDIFF,X
OE 301  IF T<>0 THEN 400
PH 302  REM ** TRIGGER PRESSED
MF 303  IF YDIFF>0 THEN 100
JC 304  IF X=110 THEN GOSUB 600
NJ 306  SOUND 1,20,8,6:R=PROW+YDIFF:C=PCOL+XDIF
        F
DF 307  POSITION C,R:? #6;CHR$(70+2*XDIFF):FOR
        I=1 TO 20:NEXT I:SOUND 1,0,0,0
PH 330  IF X=234 THEN X=32:EGGS=EGGS+1:POSITION
         CMIN+17,RMAX+1:? #6;EGGS
IF 332  POSITION C,R:? #6;CHR$(X)
GF 340  GOTO 500
HJ 400  REM ** PERSON MOTION
CF 410  IF X<>32 THEN 425
EH 412  SOUND 1,15,4,6:POSITION PCOL,PROW:? #6;
        " "
GP 420  PCOL=PCOL+XDIFF:PROW=PROW+YDIFF
CE 425  POSITION PCOL,PROW:? #6;CHR$(87+XDIFF):
        SOUND 1,0,0,0
KM 500  REM ** CHANGE EGG TO CACTUS
PJ 510  C=INT(18*RND(0))+1:R=INT(21*RND(0))+1
AD 520  LOCATE C,R,X:IF X=234 THEN POSITION C,R
        :? #6;"E"
GC 530  GOTO 100
OI 590  DATA 81,81,0,91,0,96,96,108,96,96,0,0,1
        62,0,162,121,121,0,121,0,121,0,121,121,
        128,0,121,0,108,108,0,0
JP 592  DATA 162,0,162,121,0,108,0,96,96,0,0
KJ 594  DATA 121,128,144,144,91,0,91,0,91,91,0,
        0,91,0,91,96,96,0,108,121,0,128,128,121
        ,0,108,0,121,121,0,0,0,0,-1
AM 600  REM ** DEPOSIT EGGS
JE 605  POKE 77,0
MI 610  IF EGGS=0 THEN RETURN
FM 630  FOR N=50 TO 0 STEP -2:SOUND 1,40,10,N/4
        :NEXT N:SOUND 1,0,0,0:SC=SC+EGGS:EGGS=0
KF 640  POSITION CMIN+7,RMAX+1:? #6;SC:POSITION
         CMIN+17,RMAX+1:? #6;"0   "
LP 649  POKE 19,0:POKE 20,0
HL 650  RETURN
HD 700  REM ** BIRD FLYS AWAY WITH PERSON
GL 705  SOUND 2,0,0,0
EG 710  BPREV=32:LOCATE BCOL,BROW+1,PPREV
JI 712  IF IB=1 THEN FOR I=20 TO 40:SOUND 2,I,1
        0,(40-I)/2:NEXT I
MJ 714  LOCATE BCOL+1,BROW,X:POSITION BCOL,BROW
        :? #6;CHR$(BPREV);CHR$(74+IB):BPREV=X
```

```
KC 715 LOCATE BCOL+1,BROW+1,X:POSITION BCOL,BR
       OW+1:? #6;CHR$(PPREV);CHR$(87+IB):PPREV
       =X
HH 716 BCOL=BCOL+1
AL 720 FOR N=1 TO 50:NEXT N
CK 760 IB=-IB:GOTO 712
PM 800 REM ** BIRD ATTACKS
GP 810 SOUND 0,0,0,0:SOUND 2,0,0,0
JJ 820 XDIFF=INT((2.6*RND(0)-0.6)*SGN(PCOL-BCO
       L))
NA 822 YDIFF=INT((2.6*RND(0)-0.6)*SGN(PROW-BRO
       W))
NC 828 IF ((YDIFF=0)*(XDIFF=0)) THEN 800
JG 830 LOCATE BCOL+XDIFF,BROW+YDIFF,X
CC 832 IF X=217 THEN 800
ME 850 POSITION BCOL,BROW:? #6;CHR$(BPREV):BPR
       EV=X
DP 860 BCOL=BCOL+XDIFF:BROW=BROW+YDIFF
NI 865 IB=-IB:POSITION BCOL,BROW:? #6;CHR$(74+
       IB)
AG 880 FOR I=1 TO 30:NEXT I
IA 885 IF ((BPREV=86)+(BPREV=87)+(BPREV=88)) T
       HEN TRAP 1000:GOTO 700
HC 890 GOTO 800
OG 1000 REM **END DISPLAY
FG 1010 ? #6;"{CLEAR}"
LD 1020 SETCOLOR 4.3,2:SETCOLOR 0,1,6
PK 1060 FOR I=1 TO 10:R=INT(5*RND(0))+18:C=INT
        (19*RND(0)):POSITION C,R:? #6;"E":NEXT
         I
HO 1070 FOR I=1 TO 5:R=INT(5*RND(0))+18:C=INT(
        19*RND(0)):POSITION C,R:? #6;"q":NEXT
        I
HM 1071 IF SC>TS THEN TS=SC
KK 1080 POSITION 3,1:? #6;"TOP SCORE:";TS
DK 1090 POSITION 3,5:? #6;"YOUR SCORE:";SC
LJ 1094 BCOL=0:BROW=12:TRAP 1095:GOTO 700
AG 1095 SOUND 2,0,0,0:POSITION BCOL,BROW:? #6;
        " ":POSITION BCOL,BROW+1:? #6;" "
BE 1096 POSITION 5,11:? #6;"PRESS START"
HA 1099 POKE 53279,8
PM 1100 RESTORE 1115:REM RED RIVER VALLEY
PC 1105 READ X:IF X=-1 THEN FOR J=1 TO 200:NEX
        T J:GOTO 1100
BN 1110 IF PEEK(53279)=6 THEN SOUND 3,0,0,0:GO
        TO 60
NI 1114 SOUND 3,X,10,8:FOR J=1 TO 25:NEXT J:GO
        TO 1105
GO 1115 DATA 108,81,0,64,0,64,72,0,81,0,72,81,
        0,96,0,81,81,81,0
```

```
CO 1117 DATA 108,81,0,64,0,81,64,0,53,0,60,64,
        0,72,72,72,0,0,0
EK 1118 DATA 108,60,0,64,0,64,72,0,81,0,72,64,
        0,53,0,60,60,0,96,96,0,108,0,85,81,0,7
        2,0,64,72,0,81,81,0,0,0,-1
EJ 6000 REM TITLE PAGE
NN 6020 GRAPHICS 2+16
DI 6030 POSITION 7,4:? #6;"SaGUaRO"
GN 6040 POSITION 5,10:? #6;"PRESS START"
MG 6050 COLOR 121:PLOT 1,0:DRAWTO 1,8:DRAWTO 1
        9,8:DRAWTO 19,0:DRAWTO 1,0
GE 6100 POKE 53279,8
KO 6200 RESTORE 1100
PO 6250 READ X:IF X=-1 THEN FOR J=1 TO 200:NEX
        T J:GOTO 6200
CA 6300 IF PEEK(53279)=6 THEN SOUND 2,0,0,0:GR
        APHICS 2+16:POSITION 4,5:? #6;"please 
        wait...":RETURN
NL 6400 SOUND 2,X,10,8:FOR J=1 TO 20:NEXT J:GO
        TO 6250
CE 7000 REM INITIALIZATION 1
PD 7020 DIM CHNEW$(15)
LF 7100 REM ** DEFINE CHAR SET
GN 7120 CHNEW$="IKVWXDFHBJZLMNQ"
JI 7130 MEMTOP=PEEK(106)*256:CHBASE=MEMTOP-204
        8
AF 7140 FOR I=0 TO 511:POKE CHBASE+I,PEEK(5734
        4+I):NEXT I
MC 7150 RESTORE 7190
CG 7160 FOR I=1 TO 15:CHADD=CHBASE+(ASC(CHNEW$
        (I,I))-32)*8
BM 7170 FOR J=0 TO 7:READ N:POKE CHADD+J,N:NEX
        T J:NEXT I
LA 7190 DATA 0,129,90,60,24,0,0,0
LB 7191 DATA 0,0,0,24,60,90,129,0
PG 7192 DATA 16,184,144,112,16,56,72,72
GH 7194 DATA 72,92,72,127,8,28,20,54
NE 7196 DATA 8,29,9,15,8,30,18,18
DK 7198 DATA 0,17,42,68,68,68,40,16
CA 7200 DATA 8,20,34,34,34,20,68,64
BN 7210 DATA 0,136,84,34,34,34,20,8
KB 7212 DATA 9,9,9,79,72,120,8,8
EH 7214 DATA 0,0,56,124,124,124,56,0
CF 7216 DATA 1,3,7,15,63,255,82,115
DC 7218 DATA 82,127,64,127,65,127,65,127
IF 7220 DATA 128,192,224,240,248,255,2,254
EN 7222 DATA 2,254,2,254,18,30,18,30
OL 7224 DATA 0,0,8,28,62,127,127,0
KL 7300 DIM RDIFF(9),CDIFF(9)
LM 7310 RESTORE 7330
```

```
PN 7320 FOR I=1 TO 9:READ X:CDIFF(I)=X:READ X:
        RDIFF(I)=X:NEXT I
EN 7330 DATA -1,-1,0,-1,1,-1,-1,0,0,0,1,0,-1,1
        ,0,1,1,1
MB 7998 RETURN
DI 8000 REM DRAW BORDER
NO 8002 GRAPHICS 1+16
DK 8003 POKE 756,CHBASE/256
NK 8004 SETCOLOR 0,0,0:SETCOLOR 1,1,6:SETCOLOR
         2,12,4:SETCOLOR 3,3,12:SETCOLOR 4,1,8
FF 8010 RMIN=0:RMAX=20:CMIN=0:CMAX=19
DN 8020 COLOR 217:PLOT CMIN,RMIN:DRAWTO CMAX,R
        MIN:DRAWTO CMAX,RMAX:DRAWTO CMIN,RMAX:
        DRAWTO CMIN,RMIN
PH 8580 N=CMAX-CMIN-1:J=RMAX-RMIN-3
HB 8600 FOR I=1 TO 20
BG 8610 C=INT(N*RND(0))+CMIN+1
DB 8620 R=INT(J*RND(0))+RMIN+1
BK 8630 POSITION C,R:? #6;"E"
FK 8640 NEXT I
HH 8650 FOR I=1 TO 12
LG 8660 C=INT(N*RND(0))+CMIN+1:R=INT(J*RND(0))
        +RMIN+1
IP 8670 POSITION C,R:? #6;"q":NEXT I
OK 8800 BCOL=10:BROW=10:LOCATE BCOL,BROW,X:BPR
        EV=X
BG 8810 PCOL=CMIN+3:PROW=RMAX-1
BI 8849 REM ** MAKE EGG DEPOSITORY
MI 8850 POSITION CMIN+1,RMAX-2:? #6;"zm"
LM 8852 POSITION CMIN+1,RMAX-1:? #6;"ln"
FM 8900 SC=0:EGGS=0:POSITION CMIN,RMAX+1:? #6;
        "SCORE: ";SC;:? #6;"{3 SPACES}EGGS: ";
        EGGS
FC 8902 ? #6;"SECS:"
JI 8950 RESTORE 590
NA 8960 IB=1
BI 8998 POKE 19,0:POKE 20,0:RETURN
```

Appendices

Beginner's Guide to Typing In Programs

A computer cannot perform any task by itself. Like a car without gas, a computer has *potential*, but without a program, it isn't going anywhere. All the programs in this book are written in a computer language called BASIC. Atari BASIC is easy to learn.

BASIC Programs

Computers can be picky. Unlike the English language, which is full of ambiguities, BASIC usually has only one right way of stating something. Every letter, character, and number is significant. Common mistakes are substituting a letter such as *O* for the numeral *0*, a lowercase *l* for the numeral *1*, or an uppercase *B* for the numeral *8*. Also, you must enter all punctuation marks, such as colons and commas, just as they appear in the book. Spacing can be important. To be safe, type in the listings *exactly* as they appear.

Braces and Special Characters

The exception to this typing rule is when you see the braces, for example, {DOWN}. Anything within a set of braces is a special character or characters that cannot easily be listed on a printer. When you come across such a special statement, refer to "How to Type In Programs," Appendix B.

About DATA Statements

Some programs contain a section or sections of DATA statements. These lines provide information needed by the program. Some DATA statements contain actual programs (in machine language), while others may contain graphics codes. These lines are especially sensitive to errors.

If a single number in any one DATA statement is mistyped, your machine could lock up, or crash. The keyboard, BREAK key, and RESET key may seem dead, and the screen may go blank. But don't panic. No damage has been done. To regain control, turn off your computer and then turn it back on. This will erase whatever program was in memory, *so always save a copy of your program before you run it.* If your computer crashes, you can load the program and look for your mistake.

Sometimes a mistyped DATA statement will cause an error message when the program is run. The error message may refer to the program line that READs the data. *However, the error is still in the DATA statements.*

Get to Know Your Machine

You should familiarize yourself with your computer before attempting to type in a program. Learn the statements you use to store and retrieve programs from tape or disk. You'll want to save a copy of your program so that you won't have to type it in every time you want to use it. Learn to use your machine's editing functions. How do you change a line if you make a mistake? You can always retype the line, but you should at least know how to backspace. Do you know how to enter inverse-video, lowercase, and control characters? It's all explained in your manual.

To insure accurate entry of each program line, we have included a checksum program. Please read "The Automatic Proofreader," Appendix C, before typing in any of the programs in this book.

A Quick Review

1. Type in the program a line at a time in order. Press RETURN at the end of each line. Use BACKSPACE to correct mistakes.
2. Check the line you've typed against the line in the book. You can check the entire program again if you get an error when you run the program.
3. Make sure you've entered statements in braces as the appropriate control key.
4. Be sure to save the program on tape or disk *before* running the program.

How to Type In Programs

In order to make special characters, inverse-video, and cursor characters easy to type in, we use the following listing conventions for all the programs in this book. Please refer to the table and explanations if you come across an unusual symbol in a program listing.

Conventions

Characters in inverse video will appear like this:

INVERSE VIDEO

Enter these characters with the Atari key:

When you see	Type	See
{CLEAR}	ESC SHIFT <	Clear Screen
{UP}	ESC CTRL -	Cursor Up
{DOWN}	ESC CTRL =	Cursor Down
{LEFT}	ESC CTRL +	Cursor Left
{RIGHT}	ESC CTRL *	Cursor Right
{BACK S}	ESC DELETE	Backspace
{DELETE}	ESC CTRL DELETE	Delete Character
{INSERT}	ESC CTRL INSERT	Insert Character
{DEL LINE}	ESC SHIFT DELETE	Delete Line
{INS LINE}	ESC SHIFT INSERT	Insert Line
{TAB}	ESC TAB	TAB key
{CLR TAB}	ESC CTRL TAB	Clear TAB
{SET TAB}	ESC SHIFT TAB	Set TAB stop
{BELL}	ESC CTRL 2	Ring Buzzer
{ESC}	ESC ESC	ESCape key

Graphics characters such as CTRL-T, the ball character, will appear as the normal letter enclosed in braces, {T}.

A series of identical control characters, such as 10 spaces, 3 cursor lefts, or 20 CTRL-R's, will appear as {10 SPACES}, {3 LEFT}, {20 R}, and so on. If the character in braces is in inverse video, that character or characters should be entered with the Atari key.

Program entry can be mistake-proof if you use "The Automatic Proofreader" by Charles Brannon; see Appendix C.

The Automatic Proofreader

Charles Brannon

At last there's a way for your computer to help you check your typing. "The Automatic Proofreader" will make entering programs faster, easier, and more accurate.

The strong point of computers is that they excel at tedious, exacting tasks. So why not get your computer to check your typing for you?

"The Automatic Proofreader" will help you type in program listings without typing mistakes. It's a short error-checking program that hides itself in memory. When activated, it lets you know immediately after you type a line from a program listing if you have made a mistake. Please read these instructions carefully before typing any programs in this book.

Preparing the Proofreader

1. Using the listing below, type in the Proofreader. Be very careful when entering the DATA statements—don't type the letter *l* instead of the number *1*, the letter *O* instead of the number *0*, extra commas, and so forth.
2. Save the Proofreader on tape or disk at least twice *before running it for the first time.*
3. After the Proofreader is saved, type RUN. It will check itself for typing errors in the DATA statements and warn you if there's a mistake. Correct any errors and save the corrected version. Keep a copy in a safe place—you'll need it again and again, every time you enter a program from this book or from *COMPUTE!* magazine.
4. When a correct version of the Proofreader is run, it activates itself. You are now ready to enter a program listing. If you press SYSTEM RESET, the Proofreader is disabled. To reactivate it, just type PRINT USR(1536) and press RETURN.

Using the Proofreader

All listings in this book have a *checksum* found immediately to the left of each line number. *Don't enter the checksum when typing in a program.* It is just for your information.

When you type in a line from a program listing and press

RETURN, the Proofreader displays the checksum letters at the top of your screen. *These checksum letters must match the checksum letters in the printed listing.* If they don't match, it means you typed the line differently from the way it is listed. Recheck your typing immediately. You can correct any mistakes you find.

The Proofreader is not picky with spaces. It will not notice extra spaces or missing ones. This is for your convenience, since spacing is generally not important. But occasionally proper spacing *is* important, so be extra careful with spaces. The Proofreader will catch practically everything else that can go wrong.

Due to the nature of a checksum, the Proofreader will not catch all errors. The Proofreader will not catch errors of transposition. In fact, you could type in a line in any order, and the Proofreader wouldn't notice.

There's another thing to watch out for: If you enter a line by using abbreviations for commands, the checksum will not match up. But there is a way to make the Proofreader check it. After entering the line, LIST it. This eliminates the abbreviations. Then move the cursor up to the line and press RETURN. It should now match the checksum. You can check whole groups of lines this way. The only abbreviation that cannot be handled this way is when a question mark (?) is used instead of PRINT; they are not the same to the Proofreader.

The Automatic Proofreader

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:READ A:POKE I,A:CK=C
    K+A:NEXT I
120 IF CK<>19072 THEN ? "ERROR IN DATA STAT
    EMENTS.  CHECK TYPING.":END
130 A=USR(1536)
140 ? :? "AUTOMATIC PROOFREADER NOW ACTIVAT
    ED."
150 END
1536 DATA 104,160,0,185,26,3
1542 DATA 201,69,240,7,200,200
1548 DATA 192,34,208,243,96,200
1554 DATA 169,74,153,26,3,200
1560 DATA 169,6,153,26,3,162
1566 DATA 0,189,0,228,157,74
1572 DATA 6,232,224,16,208,245
```

```
1578 DATA 169,93,141,78,6,169
1584 DATA 6,141,79,6,24,173
1590 DATA 4,228,105,1,141,95
1596 DATA 6,173,5,228,105,0
1602 DATA 141,96,6,169,0,133
1608 DATA 203,96,247,238,125,241
1614 DATA 93,6,244,241,115,241
1620 DATA 124,241,76,205,238,0
1626 DATA 0,0,0,0,32,62
1632 DATA 246,8,201,155,240,13
1638 DATA 201,32,240,7,72,24
1644 DATA 101,203,133,203,104,40
1650 DATA 96,72,152,72,138,72
1656 DATA 160,0,169,128,145,88
1662 DATA 200,192,40,208,249,165
1668 DATA 203,74,74,74,74,24
1674 DATA 105,161,160,3,145,88
1680 DATA 165,203,41,15,24,105
1686 DATA 161,200,145,88,169,0
1692 DATA 133,203,104,170,104,168
1698 DATA 104,40,96
```

Index

To order your *Atari Collection, Vol. 2 Disk* package, call our toll-free US order line: 1-800-334-0868 (in NC call 919-275-9809) or send your prepaid order to:

Atari Collection, Vol. 2 Disk
COMPUTE! Publications
P.O. Box 5058
Greensboro, NC 27403

Send _____ packages of the *Atari Collection Vol. 2 Disk* at $19.95 per package. Each disk package contains two disks.

All orders must be prepaid (check, charge, or money order). NC residents add 4.5% sales tax.

Subtotal $_____

Shipping & Handling: $2.00/package $_____

NC residents add 4.5% sales tax $_____

Total payment enclosed $_____

All payments must be in U.S. funds.

□ Payment enclosed
□ Charge □ Visa □ MasterCard □ American Express

Acct. No. _____ Exp. Date _____
(Required)

Name _____

Address _____

City _____ State _____ Zip _____

Please allow 4-5 weeks for delivery.

45N0293

COMPUTE! Books

P.O. Box 5058 Greensboro, NC 27403

Ask your retailer for these **COMPUTE! Books**. If he or she has sold out, order directly from **COMPUTE!**.

For Fastest Service
Call Our **TOLL FREE US Order Line**
800-334-0868
In NC call 919-275-9809

Or write **COMPUTE! Books**,
P.O. Box 5058, Greensboro, NC 27403

Quantity	Title	Price	Total
_____	COMPUTE!'s First Book of Atari (00-0)	**$12.95**	_____
_____	COMPUTE!'s Second Book of Atari (06-X)	**$12.95**	_____
_____	COMPUTE!'s Third Book of Atari (18-3)	**$12.95**	_____
_____	COMPUTE!'s First Book of Atari Graphics (08-6)	**$12.95**	_____
_____	COMPUTE!'s Second Book of Atari Graphics (28-0)	**$12.95**	_____
_____	COMPUTE!'s First Book of Atari Games (14-0)	**$12.95**	_____
_____	The Atari BASIC Source Book (15-9)	**$12.95**	_____
_____	Inside Atari DOS (02-7)	**$19.95**	_____
_____	COMPUTE!'s Atari Collection, Volume 1 (79-5)	**$12.95**	_____
_____	COMPUTE!'s Atari Collection, Volume 2 (029-7)	**$14.95**	_____
_____	Machine Language for Beginners (11-6)	**$14.95**	_____
_____	Second Book of Machine Language (53-1)	**$14.95**	_____
_____	Computing Together: A Parent and Teacher's Guide to Using Computers with Young Children (51-5)	**$12.95**	_____
_____	*SpeedScript:* The Word Processor for the Atari (003)	**$ 9.95**	_____
_____	Mapping The Atari, Revised (004)	**$16.95**	_____

Add $2.00 per book shipping and handling. Outside US add $5.00 air mail or $2.00 surface mail.

NC residents add 4.5% sales tax _____

Shipping & handling _____

Total payment _____

All orders must be prepaid (money order, check, or charge). All payments must be in US funds.

☐ Payment enclosed Please charge my: ☐ Visa ☐ MasterCard ☐ American Express

Acct. No. _____ Exp. Date _____

Name _____

Address _____

City _____ State _____ Zip _____

Country _____

*Allow 4–5 weeks for delivery.
Prices and availability subject to change without notice.

45N0293

All New Atari

With nearly 30 never-before-published articles and programs, including more than a dozen games, *COMPUTE!'s Atari Collection, Volume 2* brings you high-quality software for the Atari personal computer at an affordable price.

Like its forerunner, *COMPUTE!'s Atari Collection, Volume 1*, this book offers outstanding programs that run on the Atari 400, 800, XL, and XE computers. All the programs have been fully tested and are ready to type in using the error-checking utility, "The Automatic Proofreader."

Here's a sample of what's inside:

- A complete database management program that's easy to adapt to your individual needs;
- "Double Brickout," a fast-action paddle game;
- A short routine you can use in your programs which adds turtle-graphics-style commands to BASIC;
- A graphic adventure that combines the traditional use of keyboard commands with joystick movement and a graphics screen;
- "Atari Bibliography," an easy-to-use database to index all your computer magazine articles for quick retrieval;
- Utilities that change variable names and delete blocks of lines;
- Two graphics routines for creating displays you can save and use in your own programs.

As with all COMPUTE! books, the programs are ready to type in and use. Or, if you prefer, you can purchase a disk package which includes all the programs on disk, ready to load and run.

Atari owners know what to expect from COMPUTE! books—clear instructions and top-notch programs. *COMPUTE!'s Atari Collection, Volume 2* meets those expectations, and more.

ISBN 0-87455-029-7