SoftSide™

TWO DOLLARS

VOLUME TWO ● NUMBER TWELVE ● SEPTEMBER 1980

DR. LIVINGSTON

IN SEARCH OF . . .

Adventure in the dark continent

FOOSBALL

Fast-action game

STEREO GENERATOR

Harmony for your APPLE

SLEUTH

A mind-boggling word game

ALSO

Part one of the SoftSide Data Base; Computer Graphics and Sound; Programming Hints; and more . . .

ASTEROIDS
in space

TABLE OF CONTENTS

SOFTSIDE ● VOLUME TWO ● NUMBER TWELVE ● SEPTEMBER 1980

ARTICLES

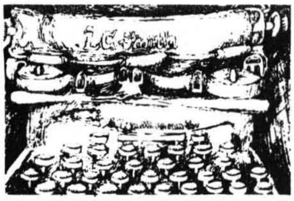
PROGRAMS

HEAVY STUFF — (BEGINNERS BEWARE!)

DEPARTMENTS

Use the following symbols as a guide when reading our ads. They indicate the computer(s) for which the product was designed.

TRS-80 APPLE ATARI



EDITORIAL

by Mark Pelczarski

STAFF

Lester Anderson
Donna Bennett
Brian Berkebile
Subscriptions :
Diana Bishop
Atari submissions:
George Blank
Richard Bouchard
Suzanne Breton
Philip Brown
Clif Campbell
Nancy Chase
Elaine Cheever
Brenda Cookinham
Donna Cookinham
Pam Demmons
Software information :
Mary Edwards
Linda Fedas
Phoebe Fensterman
Karen Fissette
S-80 submissions :
James Garon
Advertising Director :
William F. Gollan
Donna Jean
Customer service :
Bette Keenan
Shipping :
Bea Kimball
Rick Koch
Kathy Maloof
Thomas Marshall
Donna McMahon
Lance Micklus
Doris Miller
Lauri Miller
Glen Ohlund
Mark Ohlund
Apple submissions :
Mark Pelczarski
Advertising :
Mary Reed
David Robitaille
Elizabeth Robitaille
Publisher :
Roger Robitaille Sr.
Eric Skyten
Richard Taylor
Hardware Information:
Jack Torres
Anmar William
Gary Young

SoftSide is published each month by SoftSide Publications, 6 South Street, Milford, New Hampshire 03055. Telephone 603-673-5144. Subscription rates: USA Bulk Rate $18.00 per year. USA First Class, APO, FPO, Canada, Mexico, Overseas surface mail - $25.00 year. Overseas Mail $30.00 year. All remittance must be U.S. funds. Mail subscription inquiries to SoftSide Subscriptions, P.O. Box 68, Milford, New Hampshire, 03055. Entire contents copyright 1980 SoftSide Publications. All rights reserved.

POSTMASTER:
Send address changes to:

SoftSide Publications
6 South Street
Milford, New Hampshire 03055
Controlled Circulation Pending Milford, New Hampshire.

Why a Computer?

I suspect that most people can explain this computer hobby quite easily. They probably use their computer mostly for some kind of business purpose: accounting, inventory, data storage, or maybe word processing. During the off hours it becomes a hobby --learning to program or playing some newfangled game. When we talked about doing an article that would give some reasons justifying the purchase of a computer (see George Blank's in this issue), I found myself wondering if I'd had any really good reasons when I bought one myself.

You see, I'm ordinarily very suspicious of electronic gadgetry. I find things such as electric can openers to be rather silly devices, and, at least philosophically, I think televisions and stereos have done more harm to mankind than good. (But guess what I'm watching as I write this...) Food processors are scary contraptions, and I still haven't figured out why someone would buy an electric hot dog cooker. But back to the subject, why would someone who'd rather be in the High Sierra with a backpack when others would be in a Reno hotel spend over $1000 on a computer?

My admitted reason was that since I was teaching about computer programming, my own real live computer would allow me to stay knowledgeable in that field. That may have fooled others, but not me. Perhaps that was part of the reason, but I wasn't really so dedicated that I'd make that large an investment toward my job out of my own pocket.

What about other reasons? I do find my computer invaluable now as a text editor, but when I bought the computer, before I got involved in editing, my Smith-Corona did just fine. My Christmas card list isn't long enough to merit a data base program, and I don't even last very long with most computer games. Although I can occupy time well with a chess program, some of the adventure programs, or an occasional arcade-type game, I'm basically a pinball dropout.

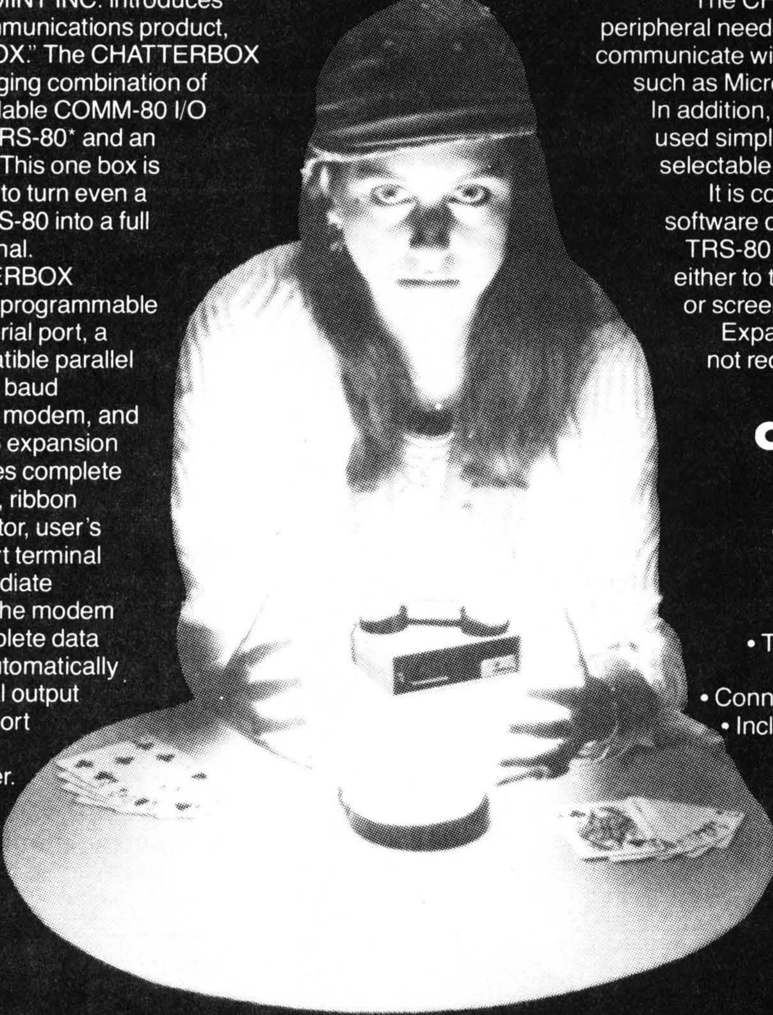
Actually, I CAN'T provide a complete set of reasons why I bought a computer. I suppose the pioneer argument would do -- being like the first ones to buy televisions or automobiles, but that's too profound. Instead, I'm inclined to fall back on something Kurt Vonnegut wrote. It's from a book almost thirty years old, Player Piano.

The story is set in the not so distant future, in which every aspect of production is controlled by machines. A few people supervise acres of automated factories, but most people work little and get bored a lot. Boredom eventually brings on a hatred of the machinery, and at the climax a revolution occurs as people unite to destroy anything that even looks mechanical. After successful destruction, the revolution's leaders decide that they'll try to create a model society of how man can live without machines; but as they confer amidst the battered rubble of an old railway station, they find a group of their revolutionaries gathered around a disabled Orange-O machine. From the back they can see several of them tinkering with it. "Filled the cup almost to the top that time; and she's nice and cold now." "But the light didn't work...fix that too." The very people that professed so strongly against machines had this innate urge to tinker with what was left! Get this to work, and make it do that, etc., etc.

I suppose the idea there comes close to my reasons, but how would others justify the expense? Well, in addition to the data and word processing capabilities mentioned earlier, a lot can be learned from and about computers by having one. The educational side of computers is not yet strongly developed, but there's

continued on page 69

TRS-80 owners Explore new worlds with CHATTERBOX.™



The MICROMINT INC. introduces its latest data communications product, the "CHATTERBOX." The CHATTERBOX is a unique packaging combination of the presently available COMM-80 I/O interface for the TRS-80* and an acoustic modem. This one box is all that is required to turn even a barebones 4K TRS-80 into a full timesharing terminal.

The CHATTERBOX includes a built-in programmable 50-19200 baud serial port, a Centronics compatible parallel printer port, a 300 baud acoustic originate modem, and a spare TRS-BUS expansion connector. It comes complete with power supply, ribbon cable and connector, user's manual, and smart terminal software for immediate operation. When the modem is in use, the complete data conversation is automatically routed to the serial output port and parallel port where it can be logged on a printer.

The CHATTERBOX is the only peripheral needed to allow a TRS-80 to communicate with timesharing systems such as Micronet and the SOURCE. In addition, CHATTERBOX can be used simply to provide an address selectable serial and parallel port. It is completely hardware and software compatible with existing TRS-80 products and connects either to the keyboard connector or screen printer port on the RS Expansion interface. It does not require the RS Expansion interface for operation.

CHATTERBOX.™
$259.95

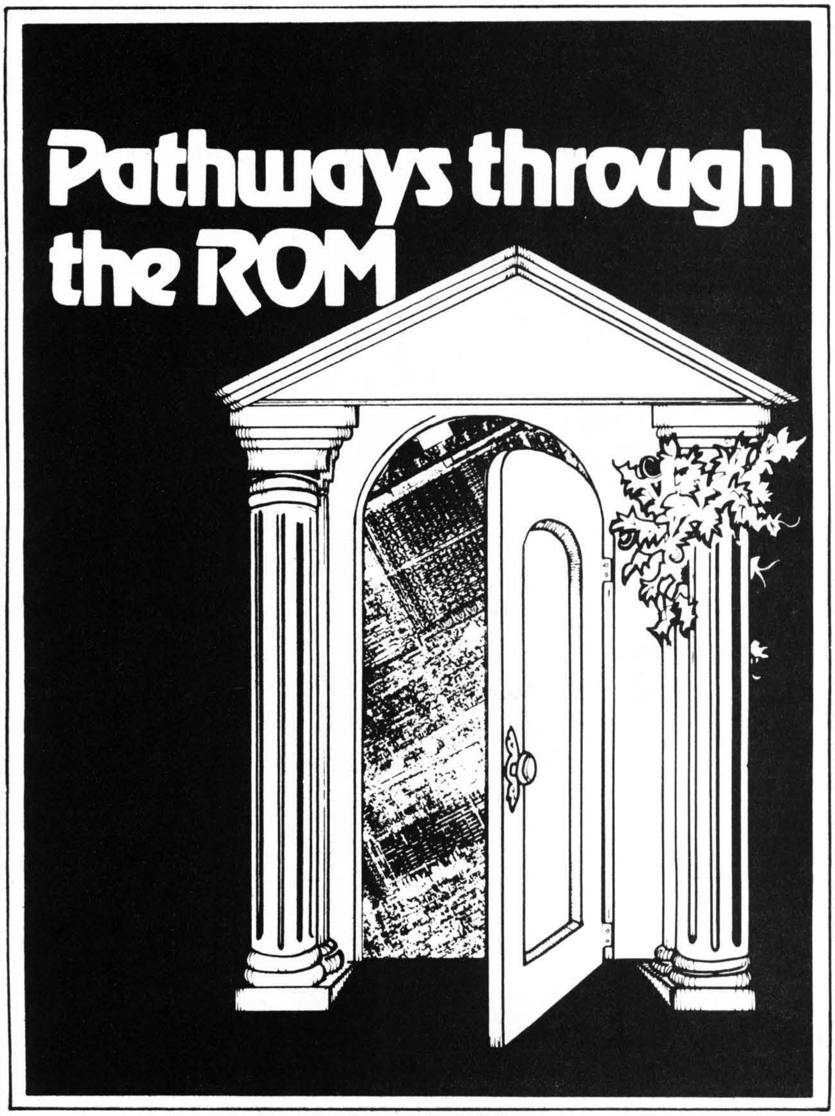
- Full 8-bit parallel port.
- RS-232-C serial port (up to 19,200 baud).
- Acoustic modem.
- TRS-BUS connector for future expansion.
- Connects to Keyboard or E.I.
- Includes terminal software.
- Users manual.
- Power supply.

HARDSIDE™

TO ORDER TOLL FREE:
1-800-258-1790 (in NH call 673-5144)

VISA
MasterCard

- **Unlock the Hidden Power Of Your Computer!**
- **Program fast & easy!**
- **Use ROM routines in your BASIC and Assembly Language programs!** All you need to know is in . . .



INCLUDES:

SUPERMAP
from Fuller Software
previously sold at $18.95

TRS-80 Disassembled Handbook
by Robert Richardson
previously sold at $10.00

Hex Mem
by John Phillipp
monitor written in BASIC.

Z-80 Disassembler
by George Blank

TO ORDER CALL TOLL-FREE
1-800-258-1790
In NH call 673-5144

- **Guide to Level II BASIC and DOS Source Code $19.95** plus $1 shipping

Description of the contents of the Level II BASIC ROM by memory location, by function, and in lesson format. Includes several BASIC and Assembly Language programs in listing format to examine and use ROM routines.

A SoftSide Publication

PROGRAMMING TIPS

Dear SoftSide,

Appletalker by Bob Bishop and Bill Depew, distributed by Softape, is a great way to add speech to your Integer programs. However, Appletalker only saves and loads the "Talkertables" to tape.

The following short Integer program will save "Talkertables" to disk after they have been recorded with the Appletalker. The program also displayes the starting address and length of the table in decimal.

When using the "Talkertables" in a program follow all the instructions furnished with Appletalker, except the voice table loading and saving instructions. Make sure you load the Appletalker subroutine package mentioned in the instructions.

A short routine such as the following may be added to a program to load the Talkertable from disk after it has been saved:

```
10 D$="": REM CTRL D
20 PRINT D$;"BLOAD FILE NAME"
```

Substitute the file name given to the recorded table previously saved for FILE NAME.

Sincerely,
Bart Immings

10-28 Clear screen and print program heading.
29 Has a table been recorded? If yes 32 If no 31
30-31 Instructed to record a table-Exit.
32 Calculates end of table address (assigns variable A).
35 Calculates length of table (assigns variable B).
45 Prints prompt for file name
48 String dimension for file name.
50 Input file name (variable C$).
52 Displays command used for saving table (file name, starting address and length).
55 Assigns CTRL D (variable D$).
60 Saves table to diskette.
100 End.

```
 5 REM  TALKERTABLE SAVER
 6 REM  BY BART IMMINGS 5-18-80
10 CALL -936: VTAB 2
20 PRINT "-----------------------
   ------------"
22 PRINT "<<< PROGRAM FOR SAVING TA
   LKERTABLES >>>"
25 PRINT "-----------------------
   ------------"
26 VTAB 12: PRINT "TABLE MUST BE RE
   CORDED WITH APPLETALKER PROGRAM
   BEFORE TABLESAVER WILL OPERATE!"
28 PRINT : PRINT : PRINT : PRINT
29 IF PEEK (10) OR ( PEEK (11)
   #16) THEN 32: PRINT "A TABLE HAS
    NOT BEEN RECORDED"
30 VTAB 18: PRINT "RECORD TABLE WIT
   H APPLETALKER"
31 END
32 A= PEEK (11)*256+ PEEK (10)
35 B=A-4095
45 VTAB 12: PRINT "WHAT FILE NAME:
   ";
48 DIM C$(40)
50 INPUT C$
52 VTAB 23: PRINT "BSAVE ";C$;
   ",A4096,L";B
55 D$=""
60 PRINT D$;"BSAVE";C$;",A4096,L"
   ;B
100 END
```

SAVING TIME AND FINGERTIPS

SAVING TIME AND FINGERTIPS is primarily for S-80 tape users, but pertains to all computers.

by Shane Causer

Well, you did it! You spent three hours typing "GALAXY COMBAT 2" or "STAR TREK III.4". Boy, all those PEEK and POKE addresses sure gave you a rough time, not to mention all those CHR$'s, but it should be worth it! Slowly, and with immense satisfaction you type RUN, and you get what took you three hours of keyboard time . . . a nice clear screen and the words "MEMORY SIZE?".

If this has ever happened to you, no matter the program, join the club! It happened to me twice with the afore-mentioned programs. But then I learned (from experience) the easy way to save time. This procedure can be described in seven steps as follows:

First, type the command SYSTEM, then / 0. This gives you a completely clean start of memory.

Second, while in command mode, POKE 16396,165. This will disable the BREAK key, but a SHIFTed BREAK works like an ordinary BREAK. If you have ever typed a long line just to accidentally press the BREAK key, you know what happens, it simply erases the line from computer memory !

Third, after answering the "MEMORY SIZE?" question, carefully ENTER the program, being conscious of the SHIFT key. Always remember the eleventh commandment: "Yea verily, thou SHIFTed character be not the same as thou unSHIFTed character", or, translated, SHIFTed characters aren't the same as unSHIFTed characters.

Fourth, "CSAVE" the program before you RUN it. This is important, especially if the program has a POKE, USR, DEFUSR, or some other machine-language-related commands. If you did make a typographical error, and you attempt a RUN, it could result in a "MEMORY SIZE?" question or an unexpected clean sweep of memory.

Fifth, RUN the program. If what happens is supposed to happen, then carefully remove the "Certificate of Honor" that is in this issue between pages 27 and 28. If not, use TRON *and* TROFF to debug the program, but only use this if there are no error messages. If you got an error message, EDIT that line so that it is the same as it was in the source program from which you typed it.

NOTE: Not all errors occur in the lines that the error messages point to. In such cases, EDIT the line preceding the "error message line".

Sixth, keep RUNning the program. Your errors may be in lines that occur later in the program. If your program has several different options choose each of these options so that you know that no errors can occur in these lines.

Seventh, once the program is running properly, CLOAD? it with the program on tape. Unless you made some corrections or changes in the resident program, there should be no discrepancies. If there are (and a BAD message will appear to tell you), CSAVE the program again.

Granted, this will take about 5 - 10 minutes longer than normal, but you'd best decide whether you want to spend 10 minutes longer or three hours longer. The logic behind this set of procedures will take time to become routine, but it is darn well worth the trouble!

How's your love life?

A little dull around the edges? Routine? Predictable? Boring? Maybe all it needs is a little Interlude. Interlude is the most stiumlating computer game ever conceived. It combines a computer interiew, an innovative programming concept, and a one-of-a-kind manual to turn your love life into exciting, adventurous, delicious fun!

With over 100 Interludes, you can satisfy all levels of interest and desire. Each Interlude is fully described in the manual, and the more elaborate ones are detailed with regard to settings, props, and mood enhancing techniques. But we've saved a few super Interludes for that very special time when your interview indicates you're ready! At that time, you will be introduced to one of several Interludes held secret within the computer. (When you learn secret Interlude #99, your love life may never be the same again!) Interlude can give you experiences you'll never forget. Are you ready for it?

For the Apple (16K I)	$16.95 for cassette
or the TRS-80 (Level II, 16K)	$19.95 for diskette

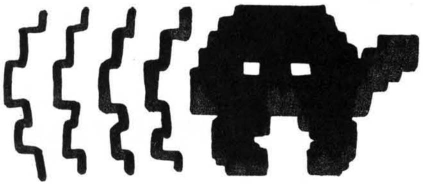
TO ORDER TOLL FREE **1-800-258-1790** (In NH call 673-5144)

Interlude™

The Ultimate Experience.

SUPER INVASION

FROM SENSATIONAL SOFTWARE



- **Super Invasion is the original invasion game, with the original moon creatures and faster action than any other invasion game.**
- **The best Invaders available for the Apple, as good in every way as the famous Invaders game.**
- **High speed action!** ● **Sound effects!**
- **Runs on the Apple II and the Apple II Plus**

Fifty-five aliens advance and shower you with lethal writhing electric worms. As you pick off the aliens, one-by-one, they quicken their descent. They whiz across the screen wearing away your parapets, your only defense, coming closer and closer to your level. Super Invasion is the original invasion game with the original moon creatures and faster action than any other invasion game on the market.

Super Invasion is available for only $19.95 on cassette for a 32K Apple II.

TO ORDER TOLL FREE
1-800-258-1790
(In NH call 673-5144)

INPUTS

A Column of Your Letters

Responses to Software Pirate

Dear SoftSide,

I was amazed that any person would write such an article which displays such selfish attitudes and complete disregard for his fellow man. I am confident he is in the minority.

Believe it or not, software pirate, I am one who does not subscribe to your standard of morality, and I know others who feel the way I do. Instead of attempting to rationalize your actions with the list cited in the article, I suggest you furnish your minister with your letter and the responses you caused and request he prepare a sermon on the subject. Although, you state you can get sermons on Sunday (not that you do) I suggest you arrange to be there. Our country and the world cannot survive with attitudes such as yours.

John J. Koback
Silver Spring, Maryland

Dear SoftSide,

In regards to the letter from the "software pirate" in the July issue -

It appears to me that the person writing the letter has a basic lack of personal integrity and was trying to give reasons or excuses to justify the action (of copying software).

However he (or she) was not totally sure it was the right thing to do as demonstrated by the unsigned letter.

Mike Murnane
Palo Alto, California

Snake (Eggs) in the grass (shack)

Dear Mr. Christopherson

On Saturday, the 12th of July, 1980, I visited a Radio Shack store located at the North Dekalb Shopping Center, North Druid Hills Rd., Decatur, Georgia.

I was looking over the programs available for the TRS-80 (I own a TRS-80 Level II, 16K version). The dealer had on display two cassettes with no documentation, merely labeled "Snake Eggs." The program was contained on Radio Shack's Supertape Gold C45 cassette. I thought it might have

continued on page 72

New Products for Your Apple

STRATEGIC SIMULATIONS

AMBUSH $59.95

Computer controlled mapboard of a typical French village
Step-by-step computer regulated play
Extensive line of sight rules providing for hidden movement
Each soldier individually rated for strength, intelligence, dexterity, and marksmanship
Realisitc weapons, characteristics, and explosives
Sophisticated movement rules permitting running, walking, crawling, dodging, and sneaking
Simultaneous execution of orders
Ability to save a game in progress and restart it at a later date
Playing time 1 to 5 hours

48K Machine Language - Disk

SYNERGISTIC SOFTWARE

MODIFIABLE DATA BASE $79.50

The Modifiable Database is a rapid response, general purpose data management system that can be automatically customized to the user's particular data storage, access, and manipulation requirements. The programs will create, under the user's direction, any number of new special purpose data management programs. One can make his own inventory control, accounts receivable, sales analysis, bibliography, memo programs, etc. The variety of applications is limited only by the user's needs and imagination. Features include:

EASILY CUSTOMIZED - The self prompting set-up program automatically modifies the basic program at the user's direction. The user specifies the number of fields, field lengths, headers, and a new program title. The customized application program is then created and saved to a new disk.

CONVENIENT DATA ENTRY - All required inputs are prompted by the program. Recurring information and default values can be entered with a single keystroke.

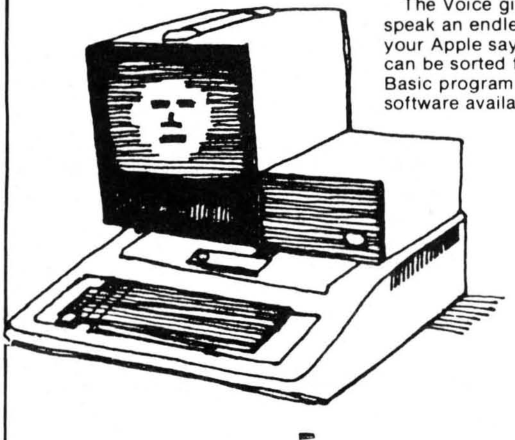
FAST RECORD ACCESS - Single keystroke commands allow any of 230 on-line records to be displayed instantly for editing, printing, or deleting. Up to 2000 records on 2 disks can be searched and/or printed with a single command. Data access and manipulation have never been faster or easier.

MACHINE LANGUAGE SEARCHES AND SORTS - Any record can be found in less than one second by specifying all or part of any 1 or 2 fields. A file of 230 records can be sorted in less than 1 minute. Sorted files can be displayed, printed, or saved. The Modifiable Database is supplied on disk and requires 48K of RAM and Applesoft firmware.

MUSE

THE VOICE $39.95

The Voice gives your Apple the power of speech! Use the standard Voice vocabulary to speak an endless combination of phrases, or easily record your own vocabulary and make your Apple say anything you like. Each data disk stores up to 80 words or phrases which can be sorted for quick reference. What's more, the Voice allows you to speak from any Basic program by using Print Commands. Guaranteed the best and easiest to use speech software available. For the Apple II or Apple II Plus with 48K.



TO ORDER TOLL-FREE
1-800-258-1790
(In NH call 673-5144)

What Do You Do With Your Personal Computer?

by George Blank

The most difficult selling job I can remember was convincing my wife that it was worthwhile to buy a computer. Now, two years later, that computer has paid for itself and is making payments on the house, but many other people are still faced with the question of how to justify the purchase of a computer.

I personally feel that the computer is so versatile that it can be justified in many different ways. It can be a source of extra income, a bridge to a new career, a tool to manage personal information or manage financial investments, an intellectual challenge, a device to broaden the horizon and opportunities of your children, or a relatively inexpensive opportunity for creative leisure.

Programming for small businesses and for the mass market offers many opportunities for part time income related to microcomputers. There are also opportunities in applications, from running a small accounting service bureau to printing biorhythm charts. There are so many potential applications that anyone with a good imagination can probably think of new ones.

People who have the need to keep track of a lot of information will find the computer worthwhile for this purpose. The computer is a natural for investment analysis, and those with stock portfolios or real estate investments will find a microcomputer profitable.

Professional people who bring work home will find the computer helpful. Engineers and scientists may require the computational ability of the computer, while business executives and salesmen will require the ability to organize and display quantities of information. People in many other careers will have different uses for the computer.

Computers are becoming so vital to our modern society that it is worth purchasing and experimenting with one just to understand what they can and cannot do. In a few years, the person who does not understand the applications and limitations of computers will be in the same position in the job market as today's high school dropout.

I have been amazed at the effect of computer literacy on my 6 and 7 year old sons. There are very few complex devices in our society that are available to pre-school children. We tell them that they have to wait until they are sixteen to drive, that they are too young for this and wouldn't understand that, but a computer is an entirely different matter. In a culture where

continued on page 88

COMPUTERS AND EDUCATION

by Mark Pelczarski

The current state of the art of educational usage of computers is often a disappointment to those who see them as having great potential as educational tools. Why? Do people expect too much, or are the solutions really there, but unseen?

The primary educational use of computers is in teaching computer programming: but what beyond that? It appears that with few exceptions, usually where grant money has been made available, not much.

In the schools, if and when funds become available for computer hardware, they go either to administration or to programming classes. The prices of microcomputers may change that: the science, language, home economics, social studies, and industrial arts classes at least have a chance now to have electronic aids, but it's still little more than a chance. Looking past the obvious hindrance of tight school budgets, computer programming classes themselves tend to grow in size and require more and more hardware support. Students are starting to learn about computers at younger and younger ages, hence needs are growing for more levels of programming classes. As computers become more widespread, this demand is likely to increase even more, and more hardware will be drawn away from other possible uses. It's unlikely that widespread innovation will occur in the schools — maybe only in small pockets here and there.

Other sources for educational software development are colleges and commercial software companies. Unfortunately, educational software so far has not been very profitable in the commercial marketplace. Good educational software is extremely difficult to write. Aside from the programming aspect, the instructional goals must be throughly planned; one does not have the spontaneity of personal interaction to help. The most successful educational programs have been simulations, which are seldom sold as "educational". But since good educational programs are so difficult to produce, the overall quality of those on the market is mediocre. Anything labeled educational becomes immediately suspect, and the few good programs become easily overlooked.

continued on page 88

GOAL

GOAL is an Applesoft program requiring at least 16K.

by Fred Pence

We recently received a cassette full of nice little logic games from the author of this program. We were impressed with his implementation of the games and use of the Apple graphics. Expect to see more 'Pence Programs' in the future.

In this game you and the computer alternately move a single counter until it reaches the goal area at the bottom. The player moving into the goal is the winner. You may choose who goes first by typing '0' for the computer to go, or a number 1-6 to make your first move into one of the six positions in the top row. Thereafter, you may move right, left, or down, but never into a previously occupied square. This program was adapted from a game in 'Mathematics in Michigan'.

```
]LIST
0  REM *****************
1  REM **              **
2  REM **     GOAL     **
3  REM **      BY      **
4  REM **  FRED PENCE  **
5  REM **              **
6  REM **  NOV, 1979   **
7  REM **              **
8  REM *****************
20  HOME : GR
30  DIM A(6,7)
40  FOR M1 = 0 TO 6: FOR M2 = 0 TO
     7:A(M1,M2) = 0: NEXT M2: NEXT
     M1
50 R = 1:C = 1
60  HOME : GR : COLOR= 1: FOR X =
     2 TO 38 STEP 6: VLIN 0,30 AT
     X: NEXT X: VLIN 31,39 AT 2: VLIN
     31,39 AT 38
70  FOR Y = 0 TO 30 STEP 6: HLIN
     2,38 AT Y: NEXT Y: HLIN 2,38
      AT 39
80  REM  PRINTING 'GOAL'
90  COLOR= 12
100  HLIN 5,10 AT 33: HLIN 5,10 AT
     37: HLIN 8,10 AT 35: VLIN 33
     ,37 AT 5: VLIN 35,37 AT 10
110  HLIN 14,19 AT 33: HLIN 14,19
      AT 37: VLIN 33,37 AT 13: VLIN
     33,37 AT 19
120  HLIN 22,27 AT 33: HLIN 22,27
      AT 35: VLIN 33,37 AT 22: VLIN
     33,37 AT 27
130  HLIN 30,35 AT 37: VLIN 33,37
      AT 30
140  IF PG >  = 2 THEN  PRINT "O.
     K. YOUR GO ";: GOTO 160
150  GOTO 200
160  PRINT "- A NUMBER FROM 1 TO
     6";: INPUT N
170  IF N < 1 OR N > 6 THEN  PRINT
     : PRINT "NO - YOUR GO ";: GOTO
     160
180  IF N =  INT (N) THEN 280
190  PRINT : PRINT "AN INTEGER ";
     : GOTO 160
200  PRINT "ENTER AN INTEGER FROM
      1 TO 6 IF YOU WANT TO GO FI
     RST."
210  PRINT "  IF YOU WANT ME TO G
     O FIRST ENTER 0";: INPUT N
220  IF  INT (N) = N THEN 260
230  PRINT : PRINT
240  PRINT "   E N T E R   A N
     I N T E G E R"
250  PRINT : PRINT "        (BETWE
     EN 0 AND 6)  ";: INPUT N
260  IF N > 6 OR N < 0 THEN  PRINT
     : PRINT : PRINT : PRINT : GOTO
     240
270  IF N = 0 THEN C =  INT (6 *
      RND (2) + 1):A(R,C) =  - 1:
      GOTO 440
280 C = N:A(R,C) =  - 1:
290  GOTO 490
300  REM  COMPUTER'S GO
310  IF R = 5 THEN CG = CG + 1: GOTO
     760
320  IF (C -  INT (C / 2) * 2) =
     0 THEN 390
330  REM  MOVE RIGHT
340  IF A(R,C + 1) = 0 THEN C = C
      + 1:A(R,C) =  - 1: GOTO 440
350  IF R < 4 THEN J =  INT (3 *
     ( RND (2))): IF J > 1 THEN R
      = R + 1:A(R,C) =  - 1: GOTO
     440
360  IF A(R,C - 1) = 0 AND C > 1 THEN
     C = C - 1:A(R,C) =  - 1: GOTO
     440
370 R = R + 1:A(R,C) =  - 1: GOTO
     440
380  REM  MOVE LEFT
390  IF A(R,C - 1) = 0 THEN C = C
      - 1:A(R,C) =  - 1: GOTO 440

400  IF R < 4 THEN J =  INT (3 *
      RND (2)): IF J > 1 THEN R =
     R + 1:A(R,C) =  - 1: GOTO 44
     0
410  IF A(R,C + 1) = 0 AND C < 6 THEN
     C = C + 1:A(R,C) =  - 1: GOTO
     440
420 R = R + 1:A(R,C) =  - 1
430  REM  COMPUTER'S GO PLOTTED
440  PRINT : PRINT : PRINT : PRINT
     "     T H E    C O M P U T E
     R   I S": PRINT : PRINT "
                 T H I N K I N G";
450  FOR T = 1 TO 1000: NEXT T
460  GOSUB 870: COLOR= 13: PLOT 6
      * C - 1,6 * R - 3
470  FOR T = 1 TO 100: NEXT T: GOTO
     510
480  REM  PLAYER'S GO PLOTTED
490  GOSUB 870: COLOR= 13: PLOT 6
      * C - 1,6 * R - 3: FOR T =
     1 TO 200: NEXT T: GOTO 310
500  REM  PLAYER'S GO
510  PRINT : PRINT : PRINT : PRINT
     "         IT'S YOUR GO":
520  PRINT : PRINT "INPUT AN 'L',
      AN 'R' OR A 'D'   ";
530  INPUT G$
540  IF G$ = "L" AND C > 1 AND A(
     R,C - 1) = 0 THEN C = C - 1:
     A(R,C) =  - 1: GOTO 490
550  IF G$ = "L" AND C = 1 THEN 6
     40
560  IF G$ = "L" AND A(R,C - 1) =
      - 1 THEN 660
```

continued on page 86

Computer Aided Drawing and Design

The examples in this article are written in Applesoft, but the techniques described can also be used with the Atari and S-80 (see below).

by Joan Truckenbrod

This is a continuation of a column that started in May in the Apple edition of SoftSide. Joan teaches about computer-aided design in the Art Department at Northern Illinois University, using a set of Apple computers. For those of you with Atari computers the differences in the programs she uses are minor. For those of you with TRS-80's, see James Garon's article about simulating HPLOT-TO and DRAWTO elsewhere in this issue.

In the sample program, lines 10 to 60 tell the computer that there are 4 points, which are read as X(1), Y(1), X(2), Y(2), etc. The points in the triangle are (100,60), (130,120), and (70,120). The first one repeats at the end to close the figure. The program will simply connect point 1 to point 2, point 2 to point 3, and point 3 to point 4. For the Atari, line 40 has to be changed to:

```
40 READ X,Y:X(J)=X:Y(J) = Y
```

since a READ statement cannot be used with array variables.

Line 100 sets the high resolution mode on the Apple, clears the screen to black, and HCOLOR=7 sets the drawing color to white. Equivalently, the Atari would use:

```
100 GRAPHICS 8:SETCOLOR
1,0,14:SETCOLOR 2,0,0:
COLOR1
```

The only other variations are with the HPLOT command. In line 180 change the word HPLOT to PLOT, and in line 220 change HPLOT TO to DRAWTO. -Ed.

SCALING TECHNIQUES

The size of figures created with the aid of a computer can be changed by various methods to develop dynamic visual effects. Scaling operations play an essential role in the process of creating compositions or animated sequences. Used in combination with translations (see SoftSide: Apple July 1980), rotations, and transformation, scaling operations facilitate the design of complex pictures or figures. There are two significant visual effects that can be created by utilizing scaling procedures. Illusions of depth in three-dimensional space can be created by sequentially changing the size of an object from large to small or small to large, and rapidly drawing each new figure. Each figure can be erased prior to drawing the next size figure, or left on the screen to create a path. As the figure is enlarged from a small form, it appears as if it is emerging from distant space towards the viewer. Also, in an animated sequence, a large figure can be scaled down so that it appears to move back in space and disappear. Scaling techniques can also be used to bring about the effect that an object or figure is being stretched out, pulled out, squeezed, or compressed by proportionally scaling the X and/or Y coordinates.

SCALING PROCEDURES

Scaling operations cause an increase or reduction in the size of a figure. These operations are implemented by multiplying all of the x and/or y coordinates defining a figure by a value called a scaling factor. A figure can be enlarged by multiplying all of these coordinates by a number greater than one. For example, using a value of two for the scaling factor will increase the figure to twice its original size. A scaling factor of four will increase its size to four times the original size. Decimal numbers such as 1.5 or 2.3 can also be used as scaling factors. Reduction of a figure requires a scaling factor between zero and one. A scaling factor of one-half will reduce the image to half its original size, one-third to one third of its original size, and so on. Curved figures will become less smooth the more they are enlarged. The closer the points are in defining a curve, the smoother the curve appears. As the distance between these points is increased, the curve becomes less smooth. Curved figures benefit from being reduced in size as any curved portions of the shape are more refined in smaller scale. In the process of enlarging a form, continually check the x and y values to make sure they do not exceed the graphic limits of the system.

The use of multiplication in the scaling process poses a problem that requires an additional step for successfully changing the size of a figure. In the scaling operation, multiplying the x and y coordinates by the scaling factor also changes the location of these figures on the screen. In order to maintain the original position of the figure on the screen, an expression using a central point of the figure must be included in the procedure. The x coordinate of a center point of the figure can be calculated by averaging the smallest x value in the figure and the largest x value. This is repeated for the y coordinates. The following expressions will be used to calculate the x and y coordinates of the center point of the figure:

XC= (largest x value + smallest x value)/2

YC= (largest y value + smallest y value)/2

These values must be correctly calculated in order to have the scaled figure in the same location as the original figure. The subroutine included in lines 300 to 390 of the program at the end of this article will do this calculation.

ENLARGING OR REDUCING THE SIZE OF A FIGURE

Variables:

S is the scaling factor

XC is the x coordinate of the center point of the figure

YC is the y coordinate of the center point

NP is the number of points used to define the figure

SCALING FORMULA:

NX = (X(J)-XC)*S+XC
NY = (Y(J)-YC)*S+YC

```
150  REM move the cursor to
     the first point in the
     figure
155  REM scale the first pair
     of coordinates
160  NX = (X(1)-XC*S+XC
170  NY = (Y(1)-YC*S+YC
180  HPLOT NX,NY
190  FOR J = 2 TO NP
```

continued on page 83

BUGS, WORMS and Other UNDESIRABLES

by Kay Pasa

In the program DRAW POKER in the April 1980 S-80 SoftSide, there should be no space between TAB and (20) in line 1070 (page 61). A "BS" or "Bad Subscript" error will result from including the space. (Level II stores the left parentheses along with the word; thus "TAB("must not include a space.)
August H. Steiner

LOST DUTCHMAN'S GOLD, in the July 1980 S-80 SoftSide has a couple of errors: on page 63, Line 1345 makes no sense as written. The author probably meant: 1345 IF N(4)=""THEN 1050 (with two quotes.) On page 64, line 1570 ends with a quote. While this does not affect the meaning of the line, it should be removed.
Bryan Elliot

There are reports that the ROM test included in TRSDOS 2.3 does not work with the new 2-chip ROM from Radio Shack. Your ROM is fine, but will fail the test.
Kay

In issue No. 9 (volume 2) 1980 S-80 SoftSide, there is an error in the return POKE address to enable LIST after disable. Instead of "POKE 16836,201", it should read "POKE 16863,201". (You should have seen me trying to recover THAT program!)

Also, in the June issue of PROG-80, the Z-80 disassembler LPRINT routine needed a simple LPRINT statement added to the end of line 62 to keep it from missing data lines as they were being printed out. I have a line printer II, so this may be different for others.
A.J. Wright

continued on page 61

Wandering Along

PATHWAYS THROUGH THE ROM

by Robert F. Nicholas

WANDERING is an S-80 article.

I've always wanted to figure out how my TRS-80 stored a program in memory. Just how does it find everything in RAM (Random Access Memory) when it needs it? Where does a BASIC program start? How does the computer know when you've run out of room? And how can it tell where one line ends and the next line begins?

Well, I found the answers to these questions and a lot of others in "Pathways Through the ROM", recently released by SoftSide Publications (6 South Street, Milford, New Hampshire 03055). The book sells for $19.95 and is well worth it since it combines:

1 - "The TRS-80 Disassembled Handbook" by Robert M. Richardson.
2 - "Supermap" by Roger Fuller.
3 - a HEX-MEM monitor program by John T. Phillipp.
4. the Z-80 Disassembler program by George Blank.
5 - a DOS map of TRSDOS and NEWDOS by John Hartford.
6 - and a WD1771 Controller Specification Sheet (for all you hardware buffs).

The first two are also published as individual books and have been on the market for about a year now. The monitor and disassembler programs appeared in PROG/80 magazine. Purchased separately, everything in this book would cost over $40!

STUMBLING ALONG IN THE DARK:

One thing which is rather entertaining to do is to look through both ROM and RAM and see just what's there. This can be done with a variety of tools including various monitors and debugging programs. If you have a disk system, experiment with DEBUG. Or, you might want to try using my program "PEEKMEM" (see figure 1).

Just type the program into your computer. There is no need to preserve memory size. The program will run in any TRS-80 with Level II BASIC from 16K to 48K, with or without disks. (We'll explain how all of this works in the second program in this article.)

You will be asked to enter the ROM/RAM starting number. Beginning from that location, the program will continue looking through all the memory you have, showing what is located there. If it is 'unprintable' (something other than a single keyboard character), that byte will be shown as a period.

Some of the highlights of Microsoft's ROM are the memory size message (around 250), BASIC's reserved word list (about 5700), assorted small messages (12250). All your BASIC programs will start after 17100 (for Level II BASIC, try 26800 if using a disk system). Experiment with "PEEKMEM" for awhile, and see what you can discover.

Once you reach the section where BASIC has stored "PEEKMEM" itself, you will notice several unusual things. For one, the line numbers don't appear. And words such as REM, FOR, NEXT, PEEK, PRINT, IF, THEN, END, GOTO, INPUT, CHR$ do not show up either. Where did they go? Surely they didn't just disappear? After all, when we list the program, they're still in the same place. So what happened to them?

Nothing! The problem is that those are among the reserved words you saw back at 5700 ROM. BASIC stores those words as one byte. And we can't show the word 'PRINT', which uses five bytes on the screen, in a one-byte position. These were the 'unprintable' bytes referred to earlier. So how do we go about recovering that information, i.e., how do we show 'PRINT' rather than a period?

GETTING STARTED:

One fairly easy method is shown in the program "LIST" (see figure 2). In lines 30230 to 30280 we created a string array, B$(Z),

continued on page 76

NEW PRODUCT

THE DATA FACTORY™

YOU CAN'T WORK HARDER, SO WORK SMARTER

This program is important to you. We at Micro Lab have tested them all. "THE DATA FACTORY" by William Passauer, is the most powerful data base system yet developed. It will provide instant accessibility to your records and files which you can then, rearrange in new combinations to give you information in seconds to make quick and accurate decisions.

OUR GUARANTEE

THIS PROGRAM WILL WORK. Micro Lab chooses to represent a very select group of professional programmers that meet our high standards for quality. Countless hours have been spent in our labs to insure these claims to you, and we back them with a contract to your dealer. Your program can also be updated if any new changes to improve this program are made.

MOST ADVANCED SYSTEM

The latest breakthroughs in a data base system have been incorporated into Bill Passauer's program. The unique new feature that sets it apart from all others is its complete modifiability. You may rearrange your data, removing part of it from the original disk, and form a new data base without reentering the data again. Add, delete, replace, or rearrange and compare fields or data at any time. Do an incredible 20 level search.

The program has no limitation to the amount of fields you can have within the limits of your computer. Your field lengths can have a maximum of 239 characters for each field and you can enter ANY CHARACTER as data (commas, quotation marks, etc.).

EASILY LEARNED

Any one can use it. The program prompts you as it runs. The easy to follow manual leads you through the set up of your data base and all the features. "The Data Factory" is organized in nine program modules. Only the module being used is loaded into memory to manipulate data, rather than the entire program. This saves memory for manipulating data rather than for program storage. There are so many other "common sense" features that set it apart from all others.

A UNIVERSAL SYSTEM

You may use "THE DATA FACTORY" at home or at work. Set up: Inventories, Mailing Lists (a printer is needed for mailing labels); Sales records; Accounts payable or receivable; Budgets; Library, recipe, or phone directories; Appointment calendar; Notices of subscriptions, license or warranty dates; Working or shopping lists, and many other applications that you will discover. All of the above can be accomplished from this one disk oriented program. No need to have separate costly programs for each purpose. With all the data on a disk, you can manipulate the information more easily and efficiently. Find any record using the record number, the data entered or any variation of the data in your data base. The from/to feature selects records by dates or amounts.

REQUIREMENTS

"The Data Factory" is presently being offered in APPLESOFT but will be available in other forms of basic shortly. Check with your dealer for other software varieties currently being handled. You will need 48k and Applesoft in ROM. "The Data Factory" is as powerful with one disk drive as with two. You do not lose any of its capabilities using only one disk drive. A printer is optional.

INSURANCE POLICY

Micro Lab is instituting a revolutionary new policy for the public: Micro Lab Disk Insurance.

You can feel secure. Two identical diskettes of the "The Data Factory" program will be included with the original purchaser's package. If you make a mistake and accidentally blow a disk, there is no time or money lost.

AVAILABLE NOW

"The Data Factory" is being offered nationally for the first time. It has been marketed and tested on a local level and has been received with a most enthusiastic response from both dealers and users.

"The Data Factory" should be at your local Apple dealer now at an introductory price of $100.00. Stop in to see our demo disk.

TO ORDER
TOLL FREE:
1-800-258-1790
(In NH call 673-5144)

micro lab

systems that work

The Stereo Generator

THE STEREO GENERATOR is written for an Apple with Applesoft and requires 16K of memory. It is printed here courtesy of Dandelion Micro Products, who sell a variety of music and graphic software for the Apple as well as the Quad Tracks Interface, an inexpensive way to expand your Apple to 4 voices in stereo. Their address is P.O. Box 615, Mt. Vernon, IL 62864.

The serious computer music lover will probably equip his Apple with a D/A converter, speaker hookup, and procure the necessary software and/or additional hardware to build his music system. The less serious music enthusiast doesn't really care about all that - but he may want his Apple to perform more complex music or sound effects than he can achieve through the single (on-board) Apple speaker. With 'The Stereo Generator' program, the Apple owner simply plugs a speaker into the cassette recorder of his computer. This is all he needs to do to create a mini-music system! 'The Stereo Generator' program will therefore serve the needs of the 'average' Apple owner for playing computer music (and at the same time develop his interests toward the more complex music systems).

'The Stereo Generator' allows the user to program two voice music (two note chords). When the music is played, the computer produces special effects on the programmed music by computing a harmonic wave form. The user, of course, has a choice in selecting the final forms(s) of the programmed music.

Theory - 'The Stereo Generator' uses a simple digital timing loop to compute a sound frequency which is made more complex (especially in the lower frequency range) by using a second timing loop. The second timer is a mathematical function of the first. The process is performed for each speaker. As a result four timing loops are used. The final wave form can be computed and programmed by the user.

Operation - To use the Stereo Generator it is necessary to connect a second speaker through the cassette recorder.

CONNECTING THE SPEAKER:

1. Unplug computer from cassette recorder at monitor jack, but leave computer connected to cassette at the mic/aux jack.

2. Connect any eight ohm speaker to the cassette recorder monitor jack (use RCA phono jack to miniature plug adapter to connect a regular stereo speaker, etc.).

3. To put the speaker into operation, place the cassette recorder in 'record' mode. Now the Stereo Generator is ready to use.

As an alternative to connecting a speaker you may use the earphone supplied with your cassette recorder.

The user may turn off the overtone sounds or either speaker as desired. The following pokes are useful for setting the various speaker modes:

	Apple Speaker	Cassette Speaker
Sound on.......	Poke 801, 48.........	Poke 817, 32
Sound off	Poke 801,0	Poke 817, 0
Overtone on	Poke 809, 48.........	Poke 827, 32
Overtone off....	Poke 809, 0...........	Poke 827, 0

As stated, the Stereo Generator is already programmed to play two note chord music with overtones for both speakers set to 'on'. There is no need to turn anything 'on' or 'off' when first running the program.

PROGRAMMING MUSIC:

The Prefix: To create music, The Stereo Generator program contains lines 0 to 50 which can be used as a prefix to user programs. The lines contain the machine language pokes and note values for the music scale. Before programming music notes the user will wish to include these lines as a prefix to his program.

First load the Stereo Generator program, next delete lines 100-700. After this the user may go ahead and develop his program starting with some line numbered above 50. When the user's program is completed the program is ready to run or save. (Also, the user's program must avoid machine language locations $6-$9, $19-$1F and $300-$35E and the use of the variable names in the prefix to keep the music program intact).

Programming Chords: Music notes are C, D, E, F, G, A, B. Sharps are CS, DS, etc. Octave scales are S1, S2, S3, S4, and S5. (S1 is the lowest scale and S5 is the highest pitched octave).

Notes are programmed first, then '/', followed by the scale. For example: C/S2 would be the C note of the second octave.

Duration of notes and rests (Moderato) are WH (whole note or rest), TH (three quarter), HA (half note), QU (quarter note), EI (eighth note), SI (sixteenth note) etc. (based on 100 beats per minute).

Program music lines using the following format:

1. Select first note of chord equal to N.

2. Select second note of chord equal to NN.

3. Select duration of chord equal to T.

4. Gosub 20 to play chord.

For example:

```
100 N = C/S3: NN = A/S2:
T = HALF: GOSUB 20
```

In line 100 the two notes of the chord are C of the third octave and A of the second octave played for ½ count duration.

```
110 N = FS/S4: NN = GS/S3:
T = WHOLE: GOSUB 20
```

In line 110 the two notes of the chord are F sharp of the fourth octave and G sharp of the third octave played for a whole note duration.

Always end each chord with GOSUB 20 before programming the next notes.

If a single note is desired rather than a chord, make both notes N and NN the same.

120 N= A/S2: NN= A/S2:
T= EIGHTH: GOSUB 20

Line 120 would play the A note of the second octave as an eighth note.

To program rests GOSUB 30 instead of 20. First select the duration then perform the GOSUB 30.

130 T= HALF: GOSUB 30 (would be a half rest), etc.

The T value for duration may also be set to any numerical value between 1 and 250. For example, setting T to 53 produces 64 quarter notes per minute.

Overtones (T1 and T2) are preset as follows: T1= H2 and T2 = H2. These may be changed to H1 or reset throughout the users program as desired such as T1= H1: T2= H1 or T1 = H1, etc. In addition T1 and T2 need not be set to harmonic variables (1 and H2), but may be set to any decimal number between .01 and .99 to achieve special sound effects. In addition, it is possible to move T1 from speaker one to speaker two with a Poke 809, 32, etc. However no resetting of speakers or overtones is necessary when first using the Stereo Generator or user programs as they are preset in the prefix as described above.

In summary, to use the Stereo Generator: load the program, delete lines 100-700, create user program starting on a line number larger than 50, finally run or save program.

The user will wish to use the music demonstration which comes with the program as an indicator for setting the values T1 and T2 within his own program. After a final decision is made, the user may wish to simplify subroutine 20 to increase speed of execution.

```
1  TEXT : HOME
```

Machine locations for music notes generation.

```
5  FOR I = 770 TO 896: READ J: POKE
     I,J:Z = Z + J: NEXT
10  DATA   202,240, 27, 136, 240,
     32, 198, 31,240 , 36,198,30,
     240,42,198, 26,240,3,76,2,3,
     198,25,208,231,76,81,3,234,2
     34
11  DATA   173,48,192,166,6,76,5,
     3,173,48,192,164,7 ,76,8,3,1
     73,32,192,165,8,133,31,76,12
     ,3,173,32,192,165,9,133 ,30,
     76,16,3,162,12,202,208, 253
12  DATA  198,26,208,247,198,25,2
     08,243, 165,29,240,9,133,25,
     169,0,133,29,76,2,3,96
```

Lines 13-15 contain music notes, octave definition, and initial variables for music overtone definition.

```
13 C = 234:CS = 221:D = 209:DS =
     197:E = 186:F = 176:FS = 166
     :G = 156:GS = 148:A = 139:AS
      = 132:B = 124
14 S1 = 1:S2 = 2:S3 = 4:S4 = 8:S5
      = 16:N1 = 6:N2 = 7:N3 = 8:N
     4 = 9:TD = 25
15  DATA  234, 165, 6, 166, 27, 7
     4, 202, 208, 252, 170, 165,
     8, 164, 27, 74, 136, 208, 25
     2, 133, 31, 165, 7, 168, 165
     , 9, 133, 30, 165, 25, 133,
     29, 76, 2, 3
16 REST = 836:H2 = .004:H1 = .02:
   T1 = H2:T2 = H2:TEMPO = 33: GOSUB
   18: GOSUB 19: GOTO 50
18 QU = TEMPO:HA = 2 * QU:WH = 4 *
     QU:EI = QU / 2:SI = QU / 4:P
     LAY = 864: RETURN
```

Set tempo and GOSUB 18 to change music speed in program. Subroutine 20 may be simplified for speed of execution in the user's music program if desired.

```
19 VOICE = 27: POKE VOICE,2: RETURN
     : REM  POKE VOICE, NO.1-8 TO
      CHANGE VOICE FACTOR
20  POKE N1,N: POKE N2,N + (T1 *
     N): POKE N3,NN: POKE N4,NN +
     (T2 * NN): POKE TD,T: CALL P
     LAY
21  RETURN
30  POKE TD,T: CALL REST
31  RETURN
50  REM  PREFIX LINES 0-50
```

Introduction of program.

```
100  INVERSE
105 T = 3:T1 = .01:T2 = H1
110  VTAB 10: HTAB 10: PRINT "THE
       STEREO GENERATOR"
120  PRINT
140  NORMAL : HTAB 11: PRINT "COP
      YRIGHT 1980 BY"
145  PRINT
150  HTAB 8: PRINT "DANDELION MIC
      RO PRODUCTS"
160  GOSUB 190:T1 = 0: GOSUB 190:
      T1 = .02: GOSUB 190:T1 = .03
      : GOSUB 190:T1 = .01: GOSUB
      190:T1 = 0:T = 10:LT = 1: GOSUB
      190
170  FOR I = 1 TO 1000: NEXT I: HOME
      : GOTO 600
```

Scale.

```
190 N = C / S2:NN = A / S1: GOSUB
      20
205 N = D / S2:NN = B / S1: GOSUB
      20
210 N = E / S2:NN = E / S2: GOSUB
      20
215 N = F / S2:NN = D / S2: GOSUB
      20
220 N = G / S2:NN = E / S2: GOSUB
      20
225 N = A / S2:NN = F / S2: GOSUB
      20
230 N = B / S2:NN = G / S2: GOSUB
      20
231  IF LT = 1 THEN T = 25
235 N = C / S3:NN = A / S2: GOSUB
      20
240  RETURN
```

Lines 400-458 contain Barcarolle music demo.

```
400 N = A / S3:NN = FS / S3:T = E
      I: GOSUB 20
402  GOSUB 30
404 N = B / S3:NN = B / S3: GOSUB
      20
406 N = B / S3:NN = FS / S3: GOSUB
      20
408  GOSUB 30
410 N = A / S3:NN = A / S3: GOSUB
      20
412 N = A / S3:NN = FS / S3: GOSUB
      20
414  GOSUB 30
416 N = B / S3:NN = B / S3: GOSUB
      20
```

continued on page 82

The Apple™ Stand

TO ORDER
TOLL FREE (orders only)
1-800-258-1790
(In NH call 673-5144)

KEY
I - Integer BASIC
M - Machine Language (Apple II or Apple II+)
A - Applesoft
ROM - Apple II Plus or Applesoft card only

Your Apple II* software market from The Software Exchange

ADVENTURELAND
Adventure International
Cassette/24K/M $14.95

ADVENTURE SAMPLER
Adventure International
Cassette/24K/M $6.95

AIR TRAFFIC CONTROLLER
Sensational Software
Cassette/16K/M $9.95

AMBUSH
disk 48K/A ROM $59.95

ANDROID NIM
Disk/24K/A ROM $17.95

APPEN I TEXT EDITOR
Muse
Cassette/16K/I $17.95

APPILOT EDU-DISK
Muse
Disk/32K/I $49.95

APPLE 21
Softape
Cassette/24K/I $15.95

APPLELISTNER
Softape
Cassette/16K/I $19.95

APPLETALKER
Softape
Cassette/16K/I $15.95

ASTEROIDS IN SPACE
disk $19.95

BASEBALL
Muse
Cassette/16K/A ROM . $14.95

BEST OF MUSE
Disk/16K/I $39.95

BISMARCK
Disk 48K/A ROM $59.95

BRIDGE CHALLENGER
Personal Software
Cassette/16K/M $14.95

CCA DATA BASE MANAGER
(Works in conjunction with Visicalc)
Personal Software
Disk/32K/A ......$99.50 + $2

COMPU-READ
Edu-ware
Disk/48K/A $24.95

COUNT ADVENTURE
Adventure International
Cassette/24K/M $14.95

DAN'S DISK UTILITIES
Periph. Unl.
Disk/16K/I $39.95

DATA FACTORY
Disk/48K/A $100.00

DATE STONES OF RYN
Tape 32K/A ROM/48K/A
.... $14.95
Disk/48K/A ROM $19.95

DR. MEMORY
Muse
Disk/32K/I $49.95

DUNGEON CAMPAIGN
Synergistic
Cassette/16K/I $14.95
Disk/32K/I $17.50

ELECTRIC CRAYON
Muse
Cassette/16K/M $17.95

ENGINEERING MATH TAPE I
Hayden
Cassette/24K A/16K A ROM $14.95

ESCAPE
Muse
Cassette/16K/I $12.95

FASTGAMMON
Quality Software
Cassette/16K/M $19.95

FORM LETTER
Periph. Unl.
Disk/48K/A ROM $24.95

FORTE
Softape
Cassette/16K/I $19.95

GENERAL MATH TAPE I
Hayden
Cassette/24K A/16K A ROM $14.95

GHOST TOWN ADVENTURE
Adventure International
Tape/24K/M.......... $14.95

GLOBAL WAR
Muse
Tape/32K/A ROM $17.95
Disk/48K/A ROM $24.95

HIGHER GRAPHICS
Synergistic
Disk/32K/I $24.95

HIGHER TEXT
Synergistic Software
Disk/24K/M $35.00

HI-RES MYSTERY HOUSE
On Line Systems
Disk/48K/M $24.95

INTERLUDE
Cassette 16K/I $16.95
Disk/32K/I $19.95

INVASION OF ORION
Tape 32K/A ROM/48KA
.... $19.95
Disk 48K/A ROM $24.95

MAGIC PAINT BRUSH
MP Software
Disk 32K/A ROM $29.95

MAILING LIST
Periph. Unl.
Disk/48K/M $39.95

MAILING LIST DATABASE
Synergistic
Disk/48K/A $34.50

MAZE GAME
Muse
Cassette/16K/I $12.95

MICROSOFT ADVENTURE
Microsoft
Disk 32K/M $29.95

MISSION IMPOSSIBLE ADVENTURE
Adventure International
Cassette/24K/M $14.95

MONTY PLAYS MONOPOLY
Ritan
Cassette/16K/M $24.95
Disk/32K/M $27.95

MORLOC'S TOWER
Tape/32K/A ROM/48K A
.... $14.95
Disk/48K/A ROM $19.95

MUSIC BOX
Muse
Cassette/16K/M $12.95

MYSTERY FUN HOUSE ADVENTURE
Adventure International
Cassette/24K/M $14.95

PIRATES COVE ADVENTURE
Adventure International
Cassette/24K/M $14.95

PRE SCHOOL IQ
Cassette/16K/I $14.95

PROGRAM LINE EDITOR
Synergistic Software
Disk/24K/M $40.00

PYRAMID OF DOOM ADVENTURE
Adventure International
Cassette/24K/M $14.95

RESCUE AT RIGEL
Tape/32K A ROM/48K A $19.95
Disk 48K/A ROM $24.95

SARGON II
Hayden
Cassette/24K/M $29.95
Disk $34.95

STARFLEET OF ORION
Tape 16K/I $19.95
Disk 32K/I $24.95

STEP BY STEP
Cassette/16K/A $14.95

STRANGE ODYSSEY ADVENTURE
Adventure International
Cassette/24K/M $14.95

SUPER INVASION
Sensational Software
Tape 32K/M $19.95

SUPER-TEXT
Muse
Disk/48K/A ROM $99.95

TEMPLE OF APSHAI
Tape 32K/A ROM/48K/A $24.95
Disk 48K/A ROM $29.95

TEXT EDITOR
Periph Unl.
Disk/48K/A ROM $64.95

THREE-D
MP Software
Disk/48K/A ROM $29.95

THREE MILE ISLAND
Muse
Disk/48K/I $39.95

U-DRAW I
Muse
Cassette/16K/M $17.95

U-DRAW II
Muse
Disk/32K/M $39.95

UNCLE SAM'S JIGSAW
Muse
Cassette/32K/A ROM . $12.95

VISICALC
Personal Software
Disk/32K/A .... $150.00 + $2

VOODOO CASTLE ADVENTURE
Adventure International
Cassette/24K/M $14.95

WILDERNESS CAMPAIGN
Synergistic
Cassette/48K/I $17.50
Disk/48K/I $19.95

WILDERNESS & DUNGEON CAMPAIGNS
Disk/48K/I $32.50

WINDFALL
Edu-Ware
Tape/32K/A $14.95
Disk/32K/A $19.95

Z-80 SOFTCARD
.... $349.00

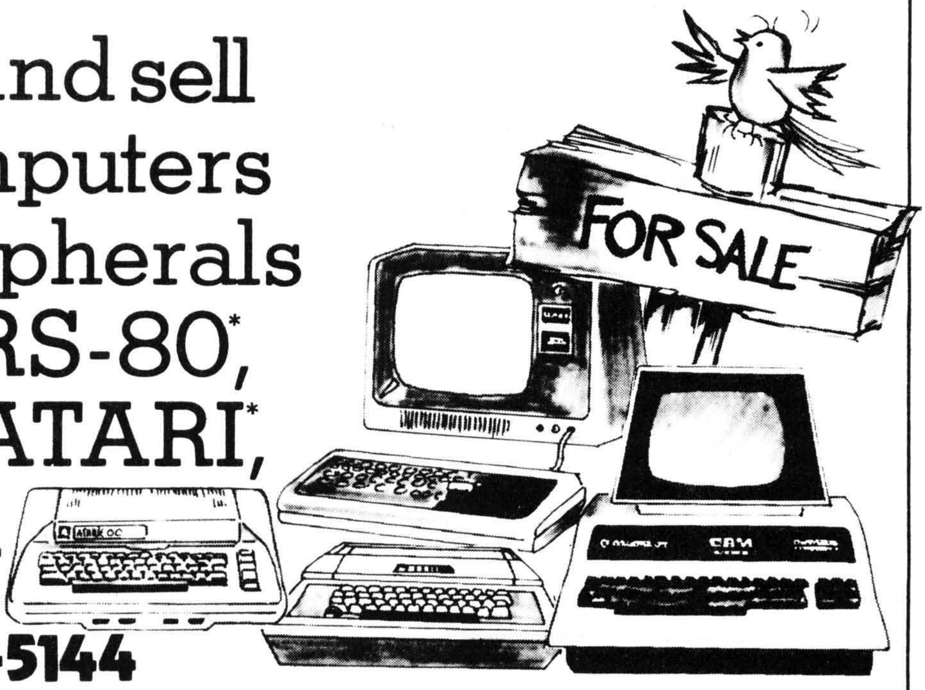
BOOKS

BASIC HANDBOOK
by David Lien ....$14.95 + $1

6502 GOURMET GUIDE & COOK BOOK
.................$12.95 + $1

INTRODUCTION TO LOW RESOLUTION ..$11.95 + $1

LITTLE BOOK OF BASIC STYLE
by John Nevison ..$5.95 + $1



CONCENTRATION II

Last month we presented a nifty version of Concentration for the Apple. Here is a version for the Level II 16K (and up) S-80 computer.

by Randy Hawkins

The game of Concentration has been around for many years -- first as a television game show, and then as a popular board game which has been enjoyed by families long after the TV show has disappeared. Young children play a simple version by laying cards face down and trying to find two that match. There was even a simple version programmed in BASIC appearing in a very early edition of SoftSide magazine by a generally unknown author by the name of Lance Micklus. The version presented here combines all the features of these earlier efforts and adds some new ones -- including an attractive graphics package and elements of skill and strategy not found in others.

The game of Concentration is really very simple. In this version, seventeen pairs of prizes and one "wild card" are randomly placed on a 5 by 7 game board. On each player's turn, two squares are chosen. If the prizes revealed match or one of the squares holds a "wild card," the player receives that prize. In addition, two pieces of a secret puzzle are put in the prize's place and the player is given the chance to identify the puzzle. In this game, the puzzle consists of a five-digit number also chosen at random before the game. Each digit is represented by a tall, thin picture of the numerals zero through nine. Should the player guess the solution to the puzzle, he wins the game and retains the prizes he has collected in that round. If he misses, he gets another turn and tries to select two more matching prizes.

As you will see, the numerals used in the puzzle are squared-off representations of the numbers -- not unlike the numbers as displayed by a calculator. This method was chosen for a specific reason. In this squared-off form, the numbers all look very much alike. In fact, for several numbers even one missing piece can make it impossible to positively identify the numeral. Is that number on the edge an eight or a zero? Is that a one or a seven? Could that be a five or a six? As a result, the game is a little bit more difficult than you might expect. Not only does it take a lot of matches before the puzzle becomes clear, but also the location of the square you are trying to match is very important. That piece in the corner is just the one you need to figure out if it is a one or a seven.

One other possibility might emerge. You have made a lot of matches and are fairly confident that you know the secret number. But you also know that there are a few big prizes waiting on the board. Do you identify the puzzle solution now and take your winnings and run or do you take a chance and try to add a few more prizes to your collection? Remember, one slip-up and your opponent will step in and leave you empty-handed!

A real challenge is to find a willing opponent and settle in for a marathon Concentration match. The program will keep setting up new games as long as you are willing and also keep a running total of your prize winnings. After four or five games in a row, you will begin to have a hard time keeping track of which prize went where. Was the sailboat in that square in this game or was that the game before. . . or the one before that?

The program listing and detailed comments follow and should be very easy to follow and possibly change if you would like to custom tailor it to your needs. A couple of special notes should be added, however. A machine language subroutine is used to paint the entire screen with a graphics character. The character used is randomly chosen so that every time you play, you will get one of sixty-three different board patterns. These vary from polka dots to horizontal and vertical stripes to weaved patterns to solid white. Rather than the USR(0) function to access this function, the NAME command is used. For normal level II and disk systems, this command is not used. It is used in the BASICR mode of disk systems as a renumbering command, but this version of BASIC is not generally used for simple execution of game programs. The instructions for implementing the NAME command are the same for both cassette and disk users making it a very simple one to use. I have also toyed with adding sound effects to the program. By using the NAME command to paint the screen, the USR(0) command is still available for you to access your favorite sound generation routine. As it is stated in some of my favorite textbooks, this is left as an exercise for the reader.

The prize list included in the DATA statements in line 1015 through 1040 is only a start. You may add as many prizes as you wish by following these simple rules. The prize name should be enclosed in quotes and centered in a twelve-character string. For example, 1980 CAR is eight characters long so adding two spaces before and after the string yields " 1980 CAR ". Following the prize string is the value in dollars. The WILD CARD! must be the final prize and it must also be the only zero value prize. Since the prize display uses only whole numbers for the dollar values, no cents are entered. That is also why the CANDY BAR prize is given the minimum value of one dollar. But considering the rate of inflation, by the time you read this article, one dollar for a candy bar may not be such a ridiculous amount after all. In any event, good luck and have fun!

Y$(2) player's name
A(35) contains a number for each board square identifying the prize hidden there.
T$(35) contains information for each board square identifying the picture beneath
PR(2,17) player's prize list
X$ machine language routine storage
S$ secret number string
X1, X2, X3 location of X$ for NAME command
CH ASCII value of graphics character used to create board
B$ graphics strings to draw pieces of puzzle
MT(2) money total for each player
P player number
NP number of prizes
W$ player's square selection
W number of square (1-35) calculated from W$
LO print location for square selected
G$ guess at secret number
P$ name of prize
M monetary value of prize P$

Lines 0-60: Program initialization

This section sets up the variables to be used in the program. Line 5 establishes the two players' names. Line 10 contains the machine language routine to fill the screen. Lines 20 and 30 locate the assembly routine and implement the NAME command. Lines 50 and 60 set up the graphics strings used to create the game board and draw the puzzle pieces.

```
0 REM        *       C O N C E N T R A T I O N        *
             *             BY RANDY HAWKINS,            *
             *            CORPUS CHRISTI, TX            *
5 CLEAR500:CLS:PRINTCHR$(23);" C O N C E N T R A T I O N":PRINT
:INPUT"WHO IS PLAYER #1";Y$(1):INPUT"WHO IS PLAYER #2";Y$(2):DIM
A(35),T$(35),PR(2,17):RANDOM
10 X$=CHR$(229)+CHR$(33)+CHR$(0)+CHR$(60)+CHR$(54)+CHR$(191)+CHR
$(17)+CHR$(1)+CHR$(60)+CHR$(1)+CHR$(255)+CHR$(3)+CHR$(237)+CHR$(
176)+CHR$(225)+CHR$(201)
20 RESTORE:S$="":X1=PEEK(VARPTR(X$)+1):X2=PEEK(VARPTR(X$)+2):X3=
X2*256+X1:X3=X3+65535*(X3>32767)
30 POKE16783,X1:POKE16784,X2:CH=128+RND(63)
50 A$=STRING$(12,CH):PR(1,0)=0:PR(2,0)=0:R$=STRING$(12,24)+CHR$(
26):B$(1)=STRING$(12,32):N$=STRING$(4,CH)+" %% "+STRING$(4,CH)
60 B$(2)="  "+CHR$(191)+STRING$(9,32):B$(3)=STRING$(9,32)+CHR$(1
91)+"  ":B$(4)="  "+CHR$(191)+STRING$(6,32)+CHR$(191)+"  ":B$(5)
="  "+STRING$(8,191)+"  ":POKEX3+5,CH
```

Lines 65-85: Opening of game

This section starts the game in motion. If the total money won equals zero (meaning this is the first game) the players are asked if they would like to see the instructions at subroutine 2000. Lines 75 and 80 prepare the game board. Line 85 writes the name of the game vertically down the right hand side of the screen.

```
65 IFMT(1)+MT(2)>0THEN75ELSEPRINT"
DO YOU NEED TO SEE INSTRUCTIONS?"
70 Z$=INKEY$:IFZ$=""THEN70ELSEIFZ$="Y"THENGOSUB2000
75 CLS:NAME:FORI=49TO53:PRINTUSINGN$;CHR$(I);:NEXTI:PRINT@128,;:
FORI=54TO57:PRINTUSINGN$;CHR$(I);:NEXTI:PRINTUSINGN$;CHR$(65);
80 FORI=2TO6:PRINT@I*128,;:FORJ=1TO5:PRINTUSINGN$;CHR$(55+I*5+J)
;:NEXTJ,I:PRINT@60,CHR$(128)CHR$(156)CHR$(140)CHR$(172);:G1$="CO
NCENTRATION":G2$=CHR$(128)+CHR$(149)+CHR$(33)+CHR$(170)
85 FORI=1TO13:PRINT@60+I*64,;:PRINTUSINGG2$;MID$(G1$,I,1);:NEXTI
:PRINT@956,CHR$(128);STRING$(3,131),"PARDON ME WHILE I PREPARE T
HE GAME";CHR$(30);:PRINT@896,CHR$(140);
```

Lines 90-180: Selection of prizes and secret number

This section chooses which prizes will be available, places them on the board, and selects and sets up the secret number puzzle. Lines 90 and 95 count the number of prizes, NP. Lines 100 through 150 choose a five-digit secret number and set up the information used in drawing these digits in array T$. Lines 160 through 175 select one player to go first, and randomly choose 17 prizes to be used in this game. Line 180 jumbles the location of the prizes among the 35 squares. Throughout this section, a border is drawn along the bottom of the screen to indicate to the players that something is going on inside the program during the approximately 10 second wait.

```
90 RESTORE:NP=0:FORI=1TO10:READZ$:NEXTI
95 NP=NP+1:READP$,M:IFM>0THEN95
100 FORI=1TO5:RESTORE
110 K=RND(10):FORJ=1TOK:READZ$:NEXTJ
120 DATA015444444444445,113333333333333,215433335222225
130 DATA315433335333345,414444445333333,515222225333345
140 DATA615422225444445,715433333333333,815444445444445,91544444
5333345
150 S$=S$+LEFT$(Z$,1):FORJ=0TO6:T$(I+J*5)=MID$(Z$,J*2+2,2):PRINT
CHR$(140);:NEXTJ,I
160 P=RND(2):A(1)=NP:FORI=2TO18
170 J=RND(NP-1):A(I)=J:FORK=1TOI-1:IFA(I)=A(K)THEN170
175 NEXTK:A(I+17)=A(I):PRINTCHR$(140);:NEXTI
180 FORL=1TO50:I=RND(35):J=RND(35):K=A(I):A(I)=A(J):A(J)=K:IFINT
(L/7)*7=LTHENPRINTCHR$(140);:NEXTLELSENEXTL
```

Lines 190-220: First selection

The player chooses a square and is shown the prize hidden beneath. The name of the square chosen is placed in W$ via the INKEY$ command. This is translated into a value from 1 to 35 in line 220. A call is made to subroutine 1000 to print the proper prize description.

```
190 PRINT@965,CHR$(31);Y$(P);", WHAT IS YOUR FIRST CHOICE? ";
200 W$=INKEY$:IFW$=""THEN200
210 IFASC(W$)<49ORASC(W$)>90OR(ASC(W$)>57ANDASC(W$)<65)THEN190
220 PRINTW$;:W=ASC(W$)-55-(W$<"A")*7:IFA(W)=0THEN190ELSEGOSUB100
0:W1=W:L1=LO
```

Lines 230-260: Second selection

This section accepts the player's second choice and functions in the same fashion as the preceding section.

```
230 PRINT@965,CHR$(31);Y$(P);", WHAT IS YOUR SECOND CHOICE? ";
240 W$=INKEY$:IFW$=""THEN240
250 IFASC(W$)<49ORASC(W$)>90OR(ASC(W$)>57ANDASC(W$)<65)THEN230
260 PRINTW$;:W=ASC(W$)-55-(W$<"A")*7:IFA(W)=0ORW=W1THEN230ELSEGO
SUB1000
```

Lines 270-300: Check for match

The two selections are compared for a match and the proper action taken. If the two prize values match or if one of the selections was a wild card, Line 270 jumps to the following section. If they do not match, the squares are replaced and the other player takes over at line 190.

```
270 IF(A(W)=A(W1))OR(A(W)=NP)OR(A(W1)=NP)THEN310
275 FORTI=1TO1000:NEXTTI:W=W+55+(W<10)*7:W1=W1+55+(W1<10)*7
280 PRINT@LO,USINGN$;CHR$(W);:PRINTR$;A$;
290 PRINT@L1,USINGN$;CHR$(W1);:PRINTR$;A$;
295 IFP=1THENP=2ELSEP=1
300 GOTO190
```

continued on page 80

SUPER BARRICADE

SUPER BARRICADE is for an 8K Atari 400 or Atari 800 with two joysticks.

by Paul Johnson

This is one of the simplest of computer games, yet at the same time, it is one of the most popular. It is fast, challenging, and very competitive. It is easy to go through fifty games in a row without wanting to stop.

Each player has a moving wall. The object of the game is to keep your wall from crashing into the other player's wall, your own wall, or the edge of the playing field on the screen, before the other player crashes. There are many subtle strategies for winning. At first, you may simply try to block your opponent so that he or she has no place to go. Another strategy is to block off as large a section of the screen as possible and move around inside it until your opponent runs out of space.

This version differs from versions available for other computers in two ways: First, because it uses joysticks, you have eight possible directions of travel instead of four, allowing you to move horizontally, vertically, or diagonally; and second, the diagonal lines are not solid. If you are skillful or lucky, you can pass right through your own or the opponent's diagonal line. This adds a lot of complexity to the game strategy.

Soldiers, man the barricades!

```
1 REM --- SUPER BARRICADE VERSION 3.1
BY PAUL F. JOHNSON, MANCHESTER, NH,     4
/10/80
2 GRAPHICS 0:PRINT "Get ready to play SU
PER BARRICADE!":? :?
```

3 PI$ equals name of first player
P2$ equals name of second player
Q$ equals input

```
3 DIM P1$(10),P2$(10),Q$(3)
4 PRINT "FIRST PLAYER - Enter your name"
:INPUT P1$
5 PRINT "SECOND PLAYER - Enter your name
":INPUT P2$
7 ? :? "DO YOU NEED INSTRUCTIONS ";:INPU
T Q$
```

8 - This line opens the screen as a file so that the program can read the screen directly. Line 400 does the actual reading.

```
8 OPEN #1,4,0,"S:"
9 IF Q$>"" THEN IF Q$(1,1)="Y" THEN GOSU
B 9000
10 H=69:V=35:DIM X(1),Y(1),L(1)
20 REM -- SET UP PLAY FIELD --
```

30 POKE 82,2 - Sets left border 2 spaces in for 38 character print width so that program will work well even on television sets with overscan problems.

```
30 GRAPHICS 5:SETCOLOR 0,8,8:SETCOLOR 1,
3,10:SETCOLOR 2,10,0:POKE 82,2
40 PRINT "                 SUPER BARRICADE
```

50 POKE 752,1 - Turns off cursor.

```
50 POKE 752,1:PRINT :PRINT P1$,S0,P2$,S1
```

100 - Draw playfield.

```
100 COLOR 3:PLOT 6,1:DRAWTO H+5,1:DRAWTO
 H+5,V:DRAWTO 6,V:DRAWTO 6,1
200 X(0)=7:Y(0)=4:X(1)=H-7:Y(1)=V-5:P=0
210 L(0)=0:L(1)=0
230 FOR I=0 TO 1:COLOR I+1:PLOT X(I)+5,Y
(I)+1:NEXT I
```

290 - Read joystick. P is number of player.

```
290 S=STICK(P)
295 IF S=15 THEN S=L(P)
```

300 - Update position based on joystick reading.

```
300 ON S GOTO 310,310,310,310,335,325,33
0,310,345,355,350,310,340,320
310 IF L(P) THEN S=L(P):GOTO 300
315 GOTO 420
320 Y(P)=Y(P)-1:GOTO 400
325 X(P)=X(P)+1:Y(P)=Y(P)-1:GOTO 400
330 X(P)=X(P)+1:GOTO 400
335 X(P)=X(P)+1:Y(P)=Y(P)+1:GOTO 400
340 Y(P)=Y(P)+1:GOTO 400
345 X(P)=X(P)-1:Y(P)=Y(P)+1:GOTO 400
350 X(P)=X(P)-1:GOTO 400
355 X(P)=X(P)-1:Y(P)=Y(P)-1
```

400 - Check next square to see if it has already been lit up.
Background color is 0 and falls through IF statement.

```
400 POSITION X(P)+5,Y(P)+1:GET #1,A:IF A
 THEN 800
405 COLOR P+1:PLOT X(P)+5,Y(P)+1
410 SOUND 0,(P*20)+60,6,1
415 L(P)=S
420 P=-1*P+1
440 GOTO 290
800 REM HIT A WALL
810 SOUND 0,8,4,8
```

820 - Flashes screen by alternating colors 100 times.

```
820 FOR I=1 TO 100:SETCOLOR 4,3,10
825 SETCOLOR 4,0,0:NEXT I
830 IF P=0 THEN S1=S1+1
840 IF P=1 THEN S0=S0+1
850 SOUND 0,0,0,0:GOTO 20
9000 REM -- DISPLAY THE RULES --
9010 GRAPHICS 0:POKE 82,1
9020 POSITION 0,8
9030 SETCOLOR 1,0,0:SETCOLOR 2,0,0
```

9040 - By alternating between normal and reverse video in these print statements, you can also alternate colors. Note: the Atari printer treats inverse video as normal text, but you can enhance your program by clever use of inverse video.

```
9040 PRINT "  SUPER barricade"
9050 PRINT
9060 PRINT P1$;" IS blue AND"
9070 PRINT "HAS THE LEFT STICK"
9075 PRINT
9080 PRINT P2$;" IS red AND"
9090 PRINT "HAS THE RIGHT STICK"
9100 PRINT
9110 PRINT "  EACH TIME YOU"
9120 PRINT " HIT A WALL YOUR"
9130 PRINT " OPPONENT SCORES!"
```

9180 Graphics 49 - Same as Graphics 1 (20 lines of 20 characters) except that screen is not cleared and split screen is overridden for 24 lines of 20 characters. Adding 16 to graphics modes 1 to 8 gives full screen graphics instead of split screen and adding 32 suppresses the clear screen function, normally part of a graphics definition.

```
9180 GRAPHICS 49:SETCOLOR 0,12,4:SETCOLO
R 1,7,8:SETCOLOR 2,11,11:SETCOLOR 3,3,5:
SETCOLOR 4,0,0
9190 FOR T=1 TO 4000:NEXT T:RETURN
```

TO ORDER TOLL-FREE
1-800-258-1790
(In NH call 673-5144)

MAGiC

from Christopherson

ANDROID NIM

The newest version of TRS-80's first animated graphics game — Android NIM — now with more animation and sound! Level II, 16K **$14.95**

BEE WARY

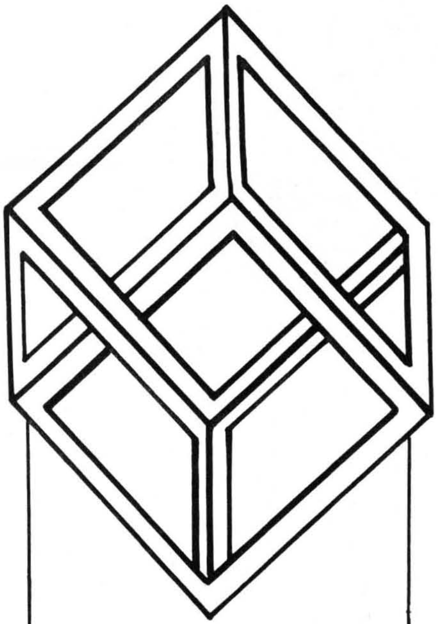
This fast-paced real time action game is a contest between a Bee operated by the player and a Spider operated by the computer. Machine language subroutines,but loads as Level II for easy operation. **$14.95**

SNAKE EGGS

Here is a computerized reptilian version of 21 complete with arrogant snakes and appropriate sound. Level II 16K. **$14.95**

LIFE TWO

Two in one: Game of Life, at an astouding 100 generations a minute, plus Battle of Life with animated creatures and sound. Level II 16K. **$14.95**



THREE D

Create 3-dimensional graphics!

by Mark Pelczarski

3-dimensional figures can be rotated, shifted, scaled, or distorted. Each figure can be saved on disk and later assembled into larger figures, with each part capable of being manipulated. Screen images may be saved and used with other programs.

48K Applesoft ROM
$29.95 on diskette

TO ORDER
TOLL FREE
1-800-258-1790
(In NH call 673-5144)

DR. LIVINGSTON IN SEARCH OF . . .

DR. LIVINGSTON is an S-80 program requiring Level II and 16K.

An adventure game
by Carl & Karen Russell
Ralph & Becky Fullerton

Dr. Livingston is again in darkest Africa, but no message has been received from him for a great while. His rescue is the overall objective of your adventure.

To explore the terrain and locate Dr. Livingston, you need to enter a two-word, verb-noun form of command like: 'get book' or 'go east'. You may also move by direction only: E, W, N, S, U, or D. The computer's vocabulary is somewhat limited, so if a word is not accepted, try a similar word or new approach.

Here are five helpful one-word commands:

HINT - not always available
LOOK - a description of your surroundings
I - (Inventory) - a list of the objects in your possession.
SCORE - points accumulated and number of turns taken
QUIT - ends game and gives final score.

Every object that you 'get' and return to a central location will improve your total score. However, in the long run, the loss of an object may be to your advantage.

Quicken your journey and improve your score. Explore! Be persistent! If you obtain the score of 215, your adventure will automatically end. A bonus may then be added, boosting your total score up to a possible 250.

Dr. Livingston's rescue has consumed your thoughts for many weeks. You have gathered books and maps of the area where he was last seen. Determined to make a rescue attempt, but exhausted from your research, you are preparing for sleep . . .

So begins your African adventure. but it will take some ingenuity to even make it to Africa - alive. Can you survive long enough to rescue Doctor Livingston? Beasts and gems, puzzles and quicksand are all about. The jungle is ever changing. Trails here today are gone tomorrow and yet here again the next.

Explore! Be persistent! Your African adventure is but a dream away . . .

```
0 'COPR.(C)1980 R/B FULLERTON:C/K RUSSELL
50 CLS:PRINT@336,CHR$(23)"DR. LIVINGSTON

":RANDOM:CLEAR420:DEFSTRA-G:DEFINTH-Z:Y=1:XG=1:G6="YOU DON'T HAV
E IT.":G7="NOTHING UNUSUAL.":G8="WITH YOUR PUNY SPEAR?"
100 G3="DON'T BE SILLY.":G2="YOU'RE ON YOUR OWN.":G1="YOU ALREAD
Y HAVE IT.":R$=" 'TIS DONE.":PM=35:VM=27:NM=46:IM=16:DIMV$(VM),N
$(NM),I$(IM),IL(IM),IP(IM),IN(IM),P$(PM),P(PM,5):GOSUB50000:GOTO
1000
200 GOSUB49990:GOTO1000
490 A="":B=A:C=A:GV=A:GN=A:INPUTA
500 Z=LEN(A):IFZ=0THEN490ELSEIFZ-1THEN510ELSEIFA="I"THENV=3:N=0:
RETJRNELSEV=0:C=A:GOTO580
510 GOSUB600:B=D:GV=G:IFZ1=1LETC=B:V=0:GOTO580ELSEFORI=1TOVM:IFB
<>V$(I)NEXTELSEV=I:GOTO550
520 V=0:C=B:GOTO580
550 IF(Z-Z1-1)<1THEN590ELSEA=RIGHT$(A,Z-Z1-1):Z=LEN(A):C=A
560 IF LEFT$(C,1)<>" "THEN570ELSEC=RIGHT$(C,Z-1):Z=LEN(C):IFZ=0T
HEN590ELSE560
570 A=C:GOSUB600:C=D
580 GN=G:FORI=1TONM:IFC<>N$(I)NEXTELSEN=I:RETURN
590 N=0:RETURN
600 FORI=1TOZ:IFMID$(A,I,1)<>" "NEXT
610 D=LEFT$(A,I-1):G=D:D=LEFT$(D,5):Z1=LEN(D):RETURN
1000 CLS:J=FRE(B):GA="
NEITHER ARE SEEN AGAIN.":GB="THE NATIVE TAKES YOUR ":GC="YOU ARE
 LYING IN A WARM SOFT BED.":GD="YOU ARE AT THE CLOSET.":GE="SEVE
RAL NATIVES WAVE HELLO!":G4="THAT'S TOO DANGEROUS.":G5="IT'S NOW
HERE IN SIGHT."
1008 G9="YOU HAD BETTER LEAVE, FAST.":PRINTP$(Y):PRINT:D="YOU CA
N SEE ":Z=12:Z1=Y:IFY=1ANDK=0THENIFIL(1)=1THENPRINTD;I$(1):GOTO1
080ELSEPRINTD;G7:GOTO1080ELSE1020
1010 D="YOU ARE CARRYING ":Z=16:Z1=-1
1020 A=D:J=1
1030 FORI=1TOIM:IFIL(I)<>Z1ANDIL(I)<>-2THEN1070
1035 B=I$(I):IFI=5ANDIL(I)=-2THENIFZ1=YORIL(2)>0THEN1070
1040 IFJ=0THEN1050ELSEJ=0:D=D+B:Z=LEN(D):PRINTD;:GOTO1070
1050 PRINT", ";:IFZ+LEN(B)>60THENPRINT:Z=0
1060 D=B:Z=Z+LEN(D)+2:PRINTD;
1070 NEXTI:IFA<>DTHENPRINT".":GOTO1080ELSEPRINTD;:IFZ1=YTHENPRIN
TG7ELSEPRINT"NOTHING."
1080 IFXB=1PRINTGC
1090 IFV=3ORV=13THEN1250
1200 XD=XD+1:IFY=7THEN5000ELSEIFY=17THEN5100ELSEIFY=34THEN5200EL
SEIFY=22ORY=27THEN5300ELSEIFY=12THEN5400
1210 IFY1=18ORY=18THEN1250ELSEX7=0
1220 XD=0
1250 IFTS=0THENIFT>20IFY>3IFY<>7IFY<>18PRINT"A NATIVE THROWS A S
PEAR AT YOU.
HE MISSES AND RUNS OFF.":IL(6)=Y:TS=1
1260 IFN4=0THENIFIL(13)=-1LETN5=N5+1:A="A CHARMING LITTLE FLUTE"
:IFN5>16PRINT"AFTER MANY HOURS OF WHITTLING, YOUR SUGAR
CANE IS NOW "A".":I$(13)=A:S=S+5:N4=1:IN(13)=15
1270 IFXE=0AND(Y=2ORY=10)PRINT:PRINT"OH NO . . . YOUR FEET ARE B
ARE . . . TOO LATE!":S=S-50:GOTO7000
1300 PRINT:GOSUB490:GOTO1500
1450 IFY=9ORY=10PRINT"A FIERCE BLACK LEOPARD BARS THE WAY.":GOTO
1200
1460 IFY=32THENIFRND(5)<3THENY=34:GOTO1000ELSEY=31:GOTO1000
1470 PRINT"THAT LEADS TO QUICKSAND.":GOTO1200
```

```
1500 T=T+1:IFVTHEN1600ELSEIFNTHEN1520
1510 I=RND(3):IFI=1THENPRINT"WHAT?"ELSEIFI=2THENPRINT"HUH?"ELSEP
RINT"WHAT IN THE WORLD ARE YOU TALKING ABOUT?
1515 GOTO1200
1520 IFN>12THEN1550ELSEIFN<7THEN1525ELSEN=N-6
1525 Z=P(Y,N-1)
1530 IFZ<0THEN1560ELSEIFZ=98THEN1450ELSEIFZ<>0THEN1540ELSEI=RND(
3):IFI=1THENPRINT"THAT DIRECTION IS SEALED OFF.":GOTO1200ELSEIFI
=2THENPRINT"YOU WON'T GET ANYWHERE GOING THAT DIRECTION.":GOTO12
00ELSEPRINT"YOU CAN'T GO THAT WAY.":GOTO1200
1540 Y1=Y:Y=Z:GOTO1000
1550 PRINT"WHAT DO YOU WANT TO DO WITH THE ";GN;"?":GOTO1200
1560 IFY=6THENIFRND(2)=2THENZ=0:GOTO1530ELSEY=-Z:GOTO1000
1570 IFY=12THENIFRND(4)=1THENY=11:GOTO1000ELSE1000
1580 IFY=22THENIFRND(2)=1THENY=21:GOTO1000ELSEY=19:GOTO1000
1585 IFRND(8)<4THENY=-Z:I$(14)="A HUNGRY ALLIGATOR":GOTO1000ELSE
Z=0:GOTO1530
1600 ONVGOTO1610,1800,1010,1900,1900,2100,2200,1000,2900,2500,26
00,2700,1010,2400,3000,3100,3200,3300,3400,3500,2700,3600,3760,3
800,3900,3700,3400
1610 IFN=0PRINT"WHO KNOWS WHAT A "GN" IS?":GOTO1300
1615 IFMI>4PRINT"YOUR LOAD'S TOO HEAVY.":GOTO1300
1617 IFN=16ANDV5=1ANDIL(8)=YTHENI=8:GOTO1690
1620 IFV5ANDN=16THEN1630ELSEGOSUB5900:IFX1>0THEN3710
1625 IFN=11LETN=22:GOTO2740
1627 IFN=34THENIFY=33THEN3720ELSE5960
1630 IFN=24THEN1700ELSEFORI=1TOIM:IFN=IN(I)THEN1635ELSENEXTI:GOT
01510
1635 IFIL(I)=-1OR(IL(2)=-1ANDIL(I)=-2)THENIFI=7THENPRINTLEFT$(G1
,17)+"HIM.":GOTO1200ELSEPRINTG1:GOTO1200
1640 IFIL(I)<>YOR(Y=1ANDK=0ANDI<>1)THEN5960
1642 IFXBIFI<>1PRINT"FROM BED?":GOTO1220
1645 IFN=18THEN3600ELSEIFI=2ANDIL(5)=-2THENMI=MI+1
1650 IFN=37THENIFXFTHEN1690ELSEPRINT"IT'S NOT FREE.":GOTO1200
1660 IFN=38PRINT"DR. LIVINGSTON?
":INPUTA:IFA<>"I PRESUME"GOSUB500:GOTO1500
1690 PRINTR$
1695 IL(I)=-1:MI=MI+1:GOTO1200
1700 IFXEPRINT"YOU'RE WEARING THEM.":GOTO1200
1705 Z=35:IFRND(8)<4THENIFRND(2)=1THENZ=25ELSEZ=27
1710 IFKPRINT"YOUR FEET ARE NOW SNUG IN A PAIR OF HIKING BOOTS."
:S=S+5:XE=1:I$(7)="DR. LIVINGSTON":IL(7)=Z:IN(7)=38:GOTO1300
1720 PRINT"WHAT BOOTS?":GOTO1300
1800 IFN=0THEN1610
1810 IFN=24THENPRINT"THEY'RE STUCK.":GOTO1200ELSEFORI=1TOIM:IFIN
(I)=NTHEN1820ELSENEXTI:GOTO1510
1820 IFIL(I)>-1THEN5950ELSEIFY=1ANDK=0ANDI>1PRINT"THERE'S NO ROO
M.":GOTO1300
1830 IFIL(I)=-2THENIFIL(2)=-1THEN3100ELSE5950
1840 IFN=16THEN3110ELSEIFI=2ANDIL(5)=-2THENMI=MI-1
1890 IL(I)=Y:PRINTR$:GOSUB3480
1895 MI=MI-1:GOTO1200
1900 B="ITS ALREADY ":C=B+"CLOSED.":A=B+"OPEN.":D="FROM BED?":IF
N=0THEN1510ELSEIFV=5THEN2000
1905 IFY-1THEN1930ELSEIFN-22THEN1920
1910 IFXB=1PRINTD:GOTO1200
1915 IFK=0THENK=1:GOTO1000ELSEPRINTA:GOTO1200
1920 IFN=20THENIFIL(1)=-1THENIFXC=0THENPRINT"THE BOOK FALLS OPEN
 TO AN UNDERLINED PASSAGE.":XC=1:GOTO1200ELSEPRINTA:GOTO1200ELSE
5950
1930 IFN=13THENIFIL(2;=-1THENPRINTA:GOTO1200ELSE5950ELSE1510
2000 IFY-1THEN2020ELSEIFN-22THEN2010
2005 IFXB=1PRINTD:GOTO1200
2007 IFK=1THENK=0:GOTO1000ELSEPRINTC:GOTO1200
2010 IFN=20THENIFIL(1)=-1THENIFXC=1THENPRINTR$:XC=0:GOTO1200ELSE
PRINTC:GOTO1200ELSE5950
2020 IFN=13THENIFIL(2)=-1THENIFIL(5)=-2THENPRINT"THE MOUSE WON'T
 ESCAPE!":GOTO1200ELSEPRINT"IT'LL JUST FALL OPEN AGAIN.":GOTO120
0ELSE5950ELSE1510
2100 IFY=2THEN2210ELSEIFIL(1)=-1ANDXB=1THENIL(1)=1:MI=MI-1:Y=2:X
B=0:GOTO1000ELSE1510
2200 IFN>0THEN1510ELSEIFY=1THENY=10:XB=0:IFIL(1)=-1THENMI=MI-1:I
L(1)=1:GOTO1000ELSE1000ELSEIFY<>10THEN1510ELSEIFIL(7)=-1THENIL(7
)=RND(33)+2:PRINT"HELP":MI=MI-1:FORJ=1TO999:NEXT
2210 Y=1:GOTO1000
2400 IFY=29THENIFN=40THENIFIL(4)=-1THENPRINTGB"GROUNDNUTS.":XF=1
:IL(4)=0:MI=MI-1:S=S+10:GOTO1200ELSE5950ELSE2420
2410 IFY=17THENIFN=42THENIFXGIFIL(10)=-1THENPRINTGB;"BEANS, GIVE
S YOU A TRINKET, THEN RUNS OFF.";:I$(10)="A GOLDEN NOSE RING":N$
(26)="RING":XG=0:S=S+15:IN(10)=26:GOTO1220ELSE5950
2420 PRINT"NO DICE.":GOTO1200
2500 IFY=1PRINT"THIS MAY SEEM STRANGE OR EVEN ABSURD
BUT IN THE BOOK IS A MAGICAL WORD!":GOTO1250
2510 IFY=23ORY=24PRINT"YOU WON'T GET ACROSS WITH A 'HOP' OR A 'S
KIP'.":GOTO1250
2520 IFY=25ORY=22ORY=12ORY=27PRINT"BE PERSISTENT.":GOTO1250
2530 IF(Y=34ORY=17)ANDIL(6)=-1PRINT"YOU LOOK HOSTILE.":GOTO1250
2540 IFY=34PRINT"LION TRAPS CIRCLE THE VILLAGE.":GOTO1250
2550 IFY=7PRINT"SOME SNAKES JUST CAN'T BE CHARMED.":GOTO1250
2560 PRINTG2:GOTO1250
2600 A="AFRICA  -  LAND OF THE UNKNOWN     BY  R. U. REDDE":B=".
..PLORER NODDED OFF TO SLEEP, HIS THOUGHTS BEGAN TO DRIFT T...":
IFN=36THENIFIL(11)=-1THENPRINT"SWAMI DIAMOND":GOTO1200ELSE5950
2610 IFN=19THENIFY=29ORY=33THENPRINT"TRADING POST":GOTO1200ELSEI
FY=31THENPRINT"UJIJI  COUNTRY  -  KEEP  OUT
NO HEED 'EM ,   WE EAT 'EM.":GOTO1200ELSE5960
2620 IFN=20ORN=25THENIFIL(1)=-1THENIFXB=1THENIFN=20THENPRINTA:GO
TO1200ELSEIFXC=1THENPRINTB:GOTO1200ELSE5960ELSEPRINT"TRY READING
 IN BED.":GOTO1200ELSE5950ELSE1510
2700 IFN=0THEN1510ELSEIFN<13THEN1520ELSEIFN=17THEN3800ELSEIFN=21
ANDY=3THENY=2:GOTO1000ELSEIFN=46IFY=17THENY=18:GOTO1000
2740 IFN=22ANDY=1PRINTGD:XB=0:GOTO1200ELSEIFN=23THENIFY=1THENPRI
NTGC:XB=1:GOTO1300ELSEPRINT"IN DARKEST AFRICA?":GOTO1200
2760 IFN=41ANDY=34THENY=35:GOTO1000ELSEIFN=29THENIFY=16THENY=17:
GOTO1000ELSEIFY=32THENN=3:GOTO1525
2770 GOTO1510
2900 IFN=35THENIFIL(16)=-1THEN2920ELSE5950
2910 IFN=14THENIFIL(6)=-1THEN2920ELSE5950ELSEPRINTG3:GOTO1200
2920 PRINT"IT SAILS THROUUGH THE AIR...
...AND HITS THE ";:IFY=2PRINT"WATER, QUICKLY SINKING.":J=0:GOTO2
940
2930 J=Y:IFY=1THENPRINT"FLOOR."ELSEPRINT"GROUND."
2935 IFY=7LETJ=6
2940 IFN=14THENIL(6)=JELSEIL(16)=J
2945 IFY=IL(14)ANDRND(2)=2THENFORI=1TO750:NEXT:PRINT"
WAIT!   THAT'S NOT GROUND.  IT'S ALLIGATOR.":I$(14)="A DEAD ALLI
GATOR"
2950 GOTO1895
3000 IF(Y=17ORY=34ORY=29)ANDN=44ANDIL(6)=-1THENPRINTG8:GOTO1200E
LSE3710
3100 IFN=18THENIFIL(5)=-2ANDIL(2)=-1THENIFY=9ORY=10THENPRINT"THE
 MOUSE JUMPS OUT AND STARTLES THE LEOPARD INTO FLIGHT."GA:P(9,4)
=10:P(10,5)=9:IL(5)=0:S=S+15:GOTO1895ELSEPRINT"THE MOUSE JUMPS O
UT AND SCAMPERS OFF.":IL(5)=2:GOTO1895ELSE5950
3110 IFN=16THENIFIL(8)=-1THENIFY=22THENPRINT"THE VIPER WAKES UP,
 ATTACKS THE DOG,
AND BOTH FALL INTO THE QUICKSAND.";GA:IL(8)=0:S=S+15:GOTO3120ELS
EPRINT"THE VIPER SLOWLY SLITHERS AWAY.":IL(8)=7:GOTO1895ELSE5950
ELSE1510
3120 I$(15)="A DEEP-BLUE SAPPHIRE":IN(15)=45:PRINT:PRINT"SOMETHI
NG SPARKLES IN THE GRASS.":GOTO1895
3200 IFN=16THENIFV5=0THENIFY=IL(8)THENIFIL(13)=-1THENI$(8)="A LE
THARGIC VIPER":V5=1:S=S+10:PRINTR$:GOTO1210ELSEPRINT"YOU MAY HAV
E GOOD LOOKS, BUT...":GOTO1200ELSEPRINTG5:GOTO1200ELSEPRINT"AGAI
N?":GOTO1200ELSE3710
3300 IFN=43THEN3310ELSEPRINT"O K  -  HERE GOES
```

continued on next page

continued from previous page

```
J U M P":GOTO1200
3310 IFY<23ORY>24THENPRINT"THERE ISN'T ANY QUICKSAND HERE.":GOTO
1200
3320 PRINT"DO YOU REALLY EXPECT TO JUMP OVER 30 FEET?
":INPUTA:IFA<>"YES"THENGOSUB500:GOTO1500
3330 GC=CHR$(24):GE=GC+GC:G9=GE+GE+GC:GA=CHR$(152):GB=CHR$(191):
G=CHR$(162)+CHR$(183)+GA:GD=CHR$(26):G5=CHR$(176):G4=GD+GE:B=G+G
4+GC+CHR$(139)+GB+CHR$(135)+G4+CHR$(168)+CHR$(145):A=G+CHR$(129)
+G4+GE+GC+GA+CHR$(171)+GB:C=CHR$(184)
3340 G=CHR$(160)+C+CHR$(128):GA=CHR$(140):C=A+GB+G5+G5+C+G4+G9+C
HR$(156)+GA+CHR$(143):A=A+CHR$(159)+G4+GB+GD+GC+CHR$(143)+GA:CLS
:PRINT"O K  -  HERE GOES":PRINT@410,A;:FORI=1TO700:NEXT
3350 FORI=1TO5:G=G+G:NEXT:GA=STRING$(5,128)+GD:D=GA+G9:D=D+D+D+D
+D:PRINT@410,D;:PRINT@473,B;:B="":FORI=1TO300:NEXT:PRINT@473,D;:
PRINT@282,C;:C=""
3360 J=FRE(B):Z=1:FORI=638TO577STEP-1:PRINT@I,LEFT$(G,Z);:Z=Z+1:
NEXTI:Z=14:FORI=1TO84:PRINT@577,MID$(G,Z,62):Z=Z+1:NEXTI:PRINT@2
82,CHR$(31);:PRINT@410,A;:PRINT@353,"TA  DA";:FORI=1TO999:NEXTI:
Q=0:IFY=23THENY=24ELSEY=23
3370 GOTO1000
3400 GOSUB3480
3410 PRINT"YOUR SCORE IS "ST" POINTS OUT OF 250 IN "T"TURNS.":IF
V=27THEN3420ELSE1250
3420 IFY=2ORY=10ORD="QUIT"END
3425 IFRETHENIFRND(5)>2END
3430 PRINT"READY
>_":FORI=1TO5000:NEXT:CLS:PRINT@338,CHR$(23)"A MIRACLE
    YOU HAVE RECOVERED":MI=0:FORI=2TO16:IFIL(I)=-1THENIL(I)=RND
(32)+3:NEXTELSENEXT
3440 FORI=1TO4000:NEXT:Y=2:RE=1:GOTO1000
3480 ST=0:FORI=1TO16:IFIL(I)=1LETST=ST+IP(I)
3485 NEXTI:ST=ST+S:IFST<215RETURN
3492 B="AN '":A="ADVENTURER'":IFT<190LETST=ST+15:B="A 'TALENTED
"
3494 IFT<160LETST=ST+10:B="A 'SKILLED "
3496 IFT<130LETST=ST+10:B="A 'MASTER "
3498 PRINT"
YOU ARE "B;A:D="QUIT":V=27:GOTO3410
3500 IFN=0THEN1510ELSEIFY=1ORY>3THEN3530
3505 IFN=21PRINT"WHERE?":GOTO1200
3520 IFY=2OR(Y=3AND(N=4ORN=10))THEN2700
3530 IFY=23ORY=24PRINT"ON QUICKSAND!":GOTO1200
3540 PRINT"ROW ON LAND?":GOTO1200
3600 IFN=18THENIFY=IL(5)THENIFIL(2)=-1THENPRINT"THE MOUSE IS NOW
 LOST IN THE FOLDS OF YOUR KNAPSACK.":I=0:IL(5)=-2:GOTO1695ELSEP
RINT"YOU DON'T HAVE ANYTHING TO PUT IT IN.":GOTO1200ELSE5960ELSE
3710
3700 IFN=18ORN=34PRINT"IT JUST ATE.":GOTO1200
3710 IFN=0THEN1510ELSEGOSUB5900:IFX1=1THENPRINTG4:GOTO1200ELSEIF
X1=2THEN5960
3720 IFN=18ORN=34THENIFV=17THENPRINT"RODENTS ARE CHARMING ENOUGH
!":GOTO1200ELSEPRINT"IT'S TOO QUICK.":GOTO1200
3730 PRINTG3:GOTO1200
3760 IFY=18THENY=17:GOTO1000ELSE1510
3800 IFY=6ORY=7THENY=7:GOTO1000ELSE1510
3900 A="
UMMMMM  GOOD.":IFN=18ANDIL(5)=-2ANDIL(2)=-1PRINTR$"
Y U K !!":IL(5)=0:GOTO1895
3920 FORI=1TO16:IFIN(I)<>NTHENNEXT:GOTO3960ELSEIFIL(I)<>-1THEN59
50ELSEIFN=42ORN=40ORN=39THENPRINTR$;A:IL(I)=0:GOTO1895
3945 IFN=0THEN1510
3950 GOSUB5900:IFX1=1PRINTG4:GOTO1200
3960 PRINTG3:GOTO1200
5000 IFV5=0THENIFXD=5THEN5140ELSEIFXD=8THEN7000ELSE1250ELSE1210
5100 IFX7=0THENX7=1:IFIL(6)=-1THEN5130ELSE5120
5110 IFIL(6)=-1THENIFXD>4THEN7000ELSE5140ELSE5150
5120 IFXGTHENPRINT"A FRIENDLY NATIVE APPROACHES WITH SOME TRINKE
TS.
IT APPEARS HE WANTS TO MAKE A TRADE.":GOTO1220ELSE1210
5130 PRINT"SUDDENLY, A VOLLEY OF SPEARS FLIES OVER YOUR HEAD;
AN OBVIOUS WARNING!  "
5140 PRINTG9:GOTO1250
5150 PRINTGE:GOTO1220
5200 IFX7THEN5110ELSEX7=1:IFIL(6)=-1THEN5130ELSE5150
5300 IFNOT((Y=IL(15)ANDI$(15)="A WILD DOG")OR(Y=IL(14)ANDI$(14)=
"A HUNGRY ALLIGATOR"))THEN1210ELSEIFXD=8THEN7000ELSEIFXD=3ORXD=5
PRINTG9
5310 GOTO1250
5400 IFXD=3PRINT"A VOICE ECHOES FROM THE MOUTH OF THE CAVE . . .
 S W A M I":GOTO1210ELSE1250
5900 X1=0:IFN=16ORN=27ORN=32ORN=33THEN5910ELSERETURN
5910 IF(N=33ANDY=27)OR(N=32ANDY=22ANDIN(15)=32)OR(N=27AND(Y=9ORY
=10)ANDP(9,4)=98)OR(N=16ANDV5=0ANDY=7)THENX1=1ELSEX1=2
5920 RETURN
5950 IFN=38THENPRINTLEFT$(G6,15)+"HIM."ELSEPRINTG6
5955 GOTO1200
5960 IFN=38THENPRINT"HE"+RIGHT$(G5,20)ELSEPRINTG5
5970 GOTO1200
7000 PRINT:PRINT"ANOTHER EXPLORER IS LOST TO THE WILDS OF AFRICA
.":PRINT:V=27:GOTO3400
50000 FORI=1TOPM:READP$(I):NEXT:DATA"YOU ARE IN A BEDROOM, WEARI
NG PJ'S.  a NIGHTSTAND SITS
BETWEEN THE CLOSET AND THE BED.  THE BEDCOVERS ARE TURNED DOWN."
,"YOU ARE DRIFTING IN A ROWBOAT ON A LAKE.
THERE IS A BEACH ON THE EASTERN SHORELINE."
50020 DATA"YOU ARE ON THE EASTERN SHORELINE.
A BOAT IS TIED TO THE DOCK.","YOU ARE ON A GRASSY PLAIN.","","A
TRAIL BEGINS HERE.  OFF TO ONE SIDE IS A TREE."
50060 DATA"YOU ARE PRECARIOUSLY BALANCED
IN A TALL MOSS-COVERED TREE.","YOU ARE ON A N/S TRAIL.  CLIFFS T
O
THE EAST REACH UP TO A PLATEAU.","YOU ARE AT THE FOOT OF A CLIFF
.
A TRAIL LEADS UP TO AN OPENING."
50110 DATA"YOU ARE AT THE MOUTH OF A CAVE.
A TRAIL LEADS DOWNWARD.","YOU ARE IN A TUNNEL.  A FAINT GLOW EMA
NATES FROM THE WEST.
A SIGN ON THE WALL SAYS:  'CAVERNS AHEAD  -  ENTER AT OWN RISK'"
50130 DATA"YOU ARE IN AN IMMENSE CAVERN.  THE WALLS
ARE COVERED WITH AN IRIDESCENT GLOW.","YOU ARE IN A LONG E/W PAS
SAGE.","","THE TRAIL HERE IS OVERGROWN WITH GRASS.","THE TRAIL W
IDENS OUT TO A NEARBY VILLAGE."
50140 DATA"YOU ARE IN A NATIVE VILLAGE.
THERE ARE SEVERAL CAMPFIRES ABOUT.","YOU ARE IN A GRASS HUT.","Y
OU ARE AT THE EDGE OF A JUNGLE.
A TRAIL LEADS NORTH.","YOU ARE AT THE EDGE OF A JUNGLE.
GRASSLANDS EXTEND TO THE EAST AND SOUTH."
50160 DATA"YOU ARE IN A JUNGLE. TRAILS LEAD OFF IN SEVERAL DIREC
TIONS.","","THERE IS A LARGE BODY OF QUICKSAND
HERE, WITH A TRAIL JUST BEYOND.","","","YOU ARE ON THE GRASS PLA
INS.
THE TERRAIN IS BECOMING SOMEWHAT MARSHY."
50170 DATA"YOU ARE IN THE MARSHES.  SOMETHING IS MOVING THE GRAS
S.","YOU ARE IN A VALLEY.  A SMALL OPENING MARKS THE HILL
BESIDE YOU.  ROLLING HILLS FLOW SOUTHWARD."
50180 DATA"YOU ARE AT A TRADING POST.  AN EAGER NATIVE
SMILES AT YOU AS HE POINTS TO HIS SIGN.","YOU ARE IN A FIELD OF
CORN.
THE GROUND DROPS OFF TO THE NORTH.","YOU ARE ON THE GRASS PLAINS

TWO SHRUNKEN HEADS DECORATE A SIGN."
50190 DATA"YOU ARE ON THE PLAINS.  A VILLAGE LIES TO THE EAST.",
"YOU ARE BEHIND A HUT.  THERE IS A SIGN IN
THE GROUND.  A SQUIRREL JUST RAN BY."
50200 DATA"YOU ARE IN THE UJIJI VILLAGE.  A NATIVE
```

```
STANDS NEAR HOLDING A SPEAR.  HE LOOKS EXCITED.","YOU ARE IN A P
IT.  LIGHT STREAMS IN FROM ABOVE."
50900 P$(5)=P$(4):P$(14)=P$(10):P$(22)=P$(21):P$(25)=P$(21):P$(2
4)=P$(23):PRINTTAB(7)"IN SEARCH OF ...":FORI=1TOPM:FORJ=0TO5:REA
DP(I,J):NEXTJ,I
52000 DATA0,0,0,0,0,0,0,0,3,0,0,0,0,5,4,2,0,0,5,6,4,3,0,0,3,0,6,
4,0,0,-5,8,-4,-5,7,0,0,0,0,0,7,6,6,15,9,0,0,0,0,0,0,8,98,0,0,0,1
1,0,0,98,0,0,12,10,0,13,-11,-11,-11,-11,-11,-11,11,0,14,15,0,0,0
,0,0,13,0,28,0,0,16,8,0,0,15,19,17,19,0,0
52010 DATA16,18,0,0,0,0,17,0,0,0,0,0,16,22,20,21,0,0,0,26,26,19,
0,0,21,23,22,21,21,21,21,22,-1,23,22,22,98,22,98,21,23,23,24,98,
25,98,24,24,-20,25,-20,24,25,25,20,27,30,20,0,0,-26,27,27,27,27,
27,0,30,29,0,14,0,0,0,0,28,0,0
52020 DATA28,26,0,31,0,0,30,32,33,31,0,0,31,0,98,31,0,0,0,0,0,31
,0,0,34,34,34,32,0,35,0,0,0,13,34,0
53000 FORI=1TOVM:READV$(I):NEXT:DATA"GET","DROP","I","OPEN","CLO
SE","DRIFT","SWAMI","LOOK","THROW","HINT","READ","GO","INVEN","T
RADE","KILL","FREE","CHARM","JUMP","SCORE","ROW","ENTER","CATCH"
,"EXIT","CLIMB","EAT","FEED","QUIT"
54000 FORI=1TONM:READN$(I):NEXT:DATA"N","S","E","W","U","D","NOR
TH","SOUTH","EAST","WEST","UP","DOWN","KNAPS","SPEAR","FLUTE","V
IPER","TREE","MOUSE","SIGN","BOOK","BOAT","CLOSE","BED","BOOTS",
"PASSA","","LEOPA"
54010 DATA"DIAMO","VILLA","TRINK","CANE","DOG","ALLIG","SQUIR","
DAGGE","NOTE","CHAIN","LIVIN","CORN","GROUN","PIT","BEANS","QUIC
K","NATIV","SAPPH","HUT"
55000 FORI=1TOIM:READI$(I):NEXT:DATA"A TIMEWORN BOOK","A FADED R
ED KNAPSACK","EARS OF CORN","A BATCH OF GROUNDNUTS","A TINY GREY
 MOUSE","A POISON-TIPPED SPEAR","A PAIR OF HIKING BOOTS","A VICI
OUS VIPER"
55010 DATA"A SPARKLING DIAMOND","A SACK OF COFFEE BEANS","A CRUM
PLED NOTE","A SOLID SILVER CHAIN","SEVERAL STALKS OF SUGAR CANE"
,"A HUNGRY ALLIGATOR","A WILD DOG","A PEARL-HANDLED DAGGER"
56000 FORI=1TOIM:READIL(I),IP(I),IN(I):NEXT:DATA1,0,20,1,2,13,30
,5,39,33,5,40,2,7,18,0,3,14,1,50,24,7,0,16,10,15,28,15,12,42,18,
3,36,29,15,37,24,15,31,27,0,33,22,10,32,12,10,35:RETURN
```

(*Editor's Note: The authors of this program have provided extensive documentation; however, it has been purposely mislaid until our next issue!*)

Illustrations for "Dr. Livingston" by Cheryl Pelczarski

ONE LINERS

Contributions for this department should be sent to:

"ONE LINERS"
c/o SoftSide Magazine
P.O. Box 68
Milford, NH 03055

The following contributions are all for the S-80, although we welcome examples for the Apple and Atari computer also.

PINWHEEL by Harry A. Hopkins, Langley AFB, Virginia:

```
1 DEFSTRA:CLS:A=CHR$(RND(63)+128):P=RND(50)+518:PRINT@P,A;:P=P+1
:PRINT@P,A;:P=P-64:PRINT@P,A;:FORX=2TORND(14):FORI=1TOX:P=P-1:PR
INT@P,A;:NEXT:FORI=1TOX:P=P+64:PRINT@P,A;:NEXT:X=X+1:FORI=1TOX:P
=P+1:PRINT@P,A;:NEXT:FORI=1TOX:P=P-64:PRINT@P,A;:NEXT:NEXTX:RUN
```

Untitled - author unknown

```
1 DEFINTA-Z:C=RND(62)+129:FORA=155TO1STEP-1:FORB=15360TO16383STE
PA:POKEB,C:NEXTB:CLS:NEXTA:RUN
```

BLASTOFF by Richard A. Vossell, Hampton, Virginia

```
1 PRINT:PRINT:R$=CHR$(191):Z$=CHR$(176):Y$=CHR$(188):X=RND(53)+3
:A$=Z$+Y$+R$+Y$+Z$:FORT=1TORND(5)+3:PRINTTAB(X)A$:A$=STRING$(5,R
$):NEXT:PRINTTAB(X-2)Z$;Y$;A$;Y$;Z$:PRINTTAB(X+1)"***":PRINTTAB(
X)"*****":PRINTTAB(X+1)"***":PRINTTAB(X+2)"*":RUN1
```

MIRROR PATTERNS and LINES by Joseph Gaudreau, Malone, NY

```
1 DEFINTA-Z:A=191:B=32:CLS:FORN=0TO15:FORT=64*N+15360TO64*N+1542
3STEPRND(4):IFPEEK(T)=BPOKET,A:POKE31743-T,A:NEXT:NEXT:GOTO1ELSE
POKET,B:POKE31743-T,B:NEXT:NEXT:GOTO1
```

```
1 CLS:DEFINTA-Z:FORN=RND(64)TORND(511)+512STEPRND(75):PRINT@N,ST
RING$(3,191):NEXT:FORN=1TO500:NEXT:GOTO1
```

ALIENS ARRIVING! by Noel Paul Stookey, Maine

```
1 CLS:PRINT@472,"ALIENS ARRIVING!":FORT=1TO500:NEXT:A$=CHR$(188)
:C$=STRING$(9," "):FORX=1TO1005STEP67:S=RND(999):PRINT@S,"*";:PR
INT@X,A$STRING$(7,175)A$:FORT=1TO24:NEXT:PRINT@X,C$:PRINT@X,A$ST
RING$(7,159)A$:FORT=1TO24:NEXT:PRINT@X,C$;:NEXT:GOTO1
```

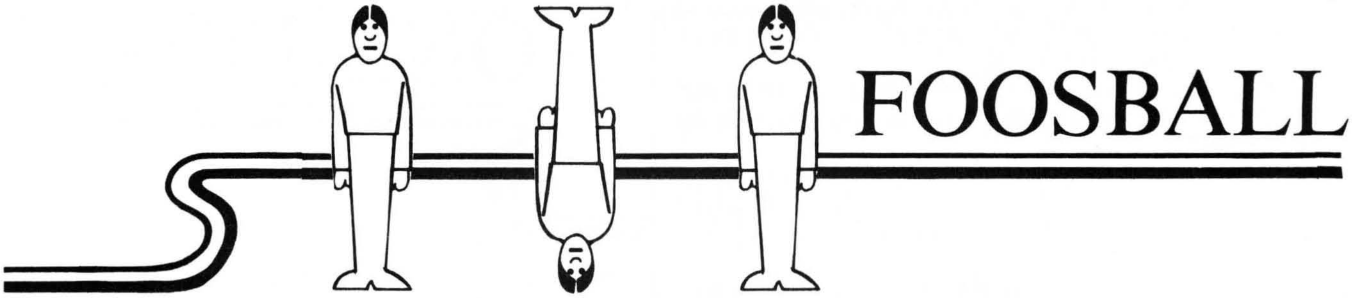
SPACE INVADERS

Mark L. Kayton
Westport, Conneticut

Instructions:

Your mission is to protect the Earth from the invading aliens (a completely original theme). You have just come across a fleet of them while they are sleeping. You must shoot your missiles at them by pressing the space-bar. Your computer will guide the missiles for you. Please note that after you have destroyed most of the aliens, your computer may take longer to decide where to fire the next missile, so be patient. When you have destroyed them all, you will have to press BREAKto end the game:

```
1 CLEAR99:C=RND(62)+33:PRINT@0,STRING$(63,C):FORX=0TO1E+9:IFINKE
Y$=""THENNEXTELSEFORH=0TO1E+9:R=RND(63)-1:IFPEEK(15360+R)<>CTHEN
NEXTELSEFORL=960+RTORSTEP-64:PRINT@L,"↑";:PRINT@L," ";:NEXT:FORL
=0TO9:PRINT@R,CHR$(RND(159)+32);:NEXT:PRINT@R," ";:NEXTXELSENEXT
X
```



FOOSBALL

FOOSBALL is an Apple program requiring Integer BASIC and 16K memory. Documentation of the machine language routine is included, but not necessary for using the program.

by Douglas Johnson

Strategic use of a machine language routine makes this an excellent, fast-paced game. With a choice of 3 paddle sizes, the object is to get the ball past your opponent 15 times. Play starts when both paddle buttons are being pressed.

PROGRAM VARIABLES

Variable Name	Description
X	All purpose FOR-NEXT loop variable
P	In the first part of the program P is the paddle input adjusted for use by the machine subroutine. In the second part P indicates which paddle the ball has hit to adjust the vertical velocity accordingly.
ST	Start flag - when either paddle button is pushed, the initial ball direction is fixed and ST prevents further change.
SL	Left player's score
SR	Right player's score
HV	Horizontal velocity of ball (1 or -1)
VV	Vertical velocity of ball (1, 0, or -1)
HP	Horizontal position of ball
VP	Vertical position of ball
OH	Old horizontal position of ball
OV	Old vertical position of ball
Z	Dummy variable for speaker clicks
A$ and Q9$	Response to input requests
A,B,C	Decimal color values to be poked into memory for use by the machine subroutine
D	Starting addresses for color sequences used by the machine subroutine
I,J,K	Addresses and offsets for position sequences

THE MACHINE LANGUAGE ROUTINE

The program pokes three groups of information into memory. Lines 2000 to 2070 put the machine language subroutine into memory. This routine occupies hexadecimal addresses $1000 to $103D. Lines 2080 through 2180 put the color sequence data into hexadecimal addresses $1200 through $127F. Lines 2190 through 2360 put position sequence data into hexadecimal addresses $1400 through $14FF. These two data sequences are used by the subroutine.

The routine is a loop within a loop. Each time through the loop a color value is retrieved from the color sequence data and placed at an address found in the position sequence data. In the low-res graphics mode this places two color blocks on the screen. The next time through the loop, the next color and address are picked in the sequences. There are two position sequences, one for each set of paddles, in addresses $1400 to $1478 and $1480 to $14F8. The addresses in these sequences draw the paddles from top to bottom and left to right. (Refer to the new reference manual for memory to screen mapping.) Since all three columns of paddles are the same, color data for only one column is stored in memory. As the routine progresses through the position addresses, the color sequence is read to the end of a column and then reset to the starting color value and read again. This is the reason for the inner and outer loops. The apparent position of the paddles on the screen is determined by where in the color sequence the routine begins reading values. There are four color sequences, two for each set of paddles. The reason for two sequences is that each 8 bit byte stores colors for two blocks. Rather than trying to read a half byte, a second sequence which is shifted a half byte is stored. Which paddle to draw and where to start in the color sequence are poked into the routine by the main program. An assembly listing with remarks follows:

```
Address—Operation—Argument ————————Remarks——————
 OFFF———————————————————————— stores starting position which
                              has been poked in by the
                              main program
 1000 ——— LDX ——— #$00 ——— initialize inner loop counter
 1002 ——— LDY ——— #$00 ——— initialize outer loop counter
 1004 ——— LDA ——— $0FFF——— load accumulator with the
                              value in $0FFF
 1007 ——— STA ——— $1017 ——— store value in accumulator in
                              $1017
 100A ——— LDA ——— $14xx ——— load accumulator with the
                              value in $14xx. The lower
                              half of the address xx is
                              poked in by the main
                              program and indicates which
                              set of paddles are to drawn
 100D ——— STA ——— $101B——— store value in accumulator in
                              $101B
 1010 ——— LDA ——— $14xx ——— load the accumulator with the
                              value in $14xx. 'xx' is the se-
                              cond half of the address for
                              paddle position
```

1013	STA	$101A	store value in accumulator in $101A
1016	LDA	$12xx	load accumulator with the value in $12xx. 'xx' was transferred from $OFFF
1019	STA	$xxxx	store value in accumulator in xxxx. 'xxxx' is the address constructed from addresses 14xx and 14xx + 1
101C	INX		increment inner loop counter
101D	CPX	#$15	if the value in register X
101F	BEQ	$1033	is equal to hexadecimal $15 (decimal 21) then branch to $1033
1021	INC	$100B	increment the value in $100B
1024	INC	$100B	twice
1027	INC	$1011	increment the value in $1011
102A	INC	$1011	twice these increment the screen address position pointer
102D	INC	$1017	increment the value in $1017. This increments the color sequence pointer
1030	JMP	$100A	go to $100A
1033	LDX	#$00	re-initialize the inner loop counter
1035	INY		increment the outer loop counter
1036	CPY	#$03	if the value in register Y
1038	BEQ	$103D	equals 3 then branch to $103D
103A	JMP	$1004	go to $1004
103D	RTS		return to the main program

Lines 2-6 change LOMEM to accomodate the machine language routine, call the subroutine that pokes in the routine, and set low resolution graphics mode.

```
  2 POKE 74,5400 MOD 256: POKE
    75,5400/256: POKE 204, PEEK
    (74): POKE 205, PEEK (75)
  4 GOSUB 1000: GOSUB 2000
  6 GR
```

Lines 8-16 create the background.

```
  8 COLOR=2
 10 FOR X=0 TO 39
 12 HLIN 0,39 AT X
 14 NEXT X
 16 GOTO 44
```

Lines 18-42 form the subroutine that draws the paddles. Values are passed to a machine language routine to increase the speed.

```
 18 P=( PDL (0)/10) MOD 12
 20 P=(11-P)/2+32*(P MOD 2)
 22 POKE 4095,P
 24 POKE 4107,0
 26 POKE 4113,1
 28 CALL 4096
 30 P=( PDL (1)/10) MOD 12
 32 P=(11-P)/2+32*(P MOD 2)+64
 34 POKE 4095,P
 36 POKE 4107,128
 38 POKE 4113,129
 40 CALL 4096
 42 RETURN
```

Lines 44-52 are initialization.

```
 44 ST=0
 46 SL=0:SR=0
 48 PRINT " ",SL," ",SR
 50 PRINT
 52 PRINT
```

Lines 54-100 show the flashing ball at the beginning. When both paddle buttons are pushed the loop is broken and play begins.

```
 54 COLOR=15
 56 PLOT 19,19
 58 FOR X=1 TO 15
 60 GOSUB 18
 62 GOSUB 66
 64 GOTO 80
 66 IF ST=1 THEN 78
 68 IF PEEK (-16287)<128 THEN 74

 70 ST=1:HV=1
 72 GOTO 78
 74 IF PEEK (-16286)<128 THEN 78
 76 ST=1:HV=-1
 78 RETURN
 80 NEXT X
 82 COLOR=2
 84 PLOT 19,19
 86 FOR X=1 TO 15
 88 GOSUB 18
 90 GOSUB 66
 92 NEXT X
 94 IF ST=1 THEN 98
 96 GOTO 54
 98 IF PEEK (-16287)>127 AND PEEK
    (-16286)>127 THEN 102
100 GOTO 54
```

Lines 102-120 control the position of the ball.

```
102 ST=0:VV=0:HP=19:VP=19
104 COLOR=2:OH=HP:OV=VP:HP=HP+HV:
    VP=VP+VV: IF HP<0 OR HP>39 THEN
    116: IF VP>-1 AND VP<40 THEN
    106:VV=-VV:VP=VP+2*VV
106 IF SCRN(HP,VP)=2 THEN 114:HV=
    -HV:HP=HP+2*HV:P=1: IF (HP-
    2*HV-4) MOD 12=0 THEN P=0:VP=
    VP-VV
108 Z= PEEK (-16336):Z= PEEK (-
    16336)
110 VV=VV+(((VP-(( PDL (P)/10) MOD
    12)-14+PS*3) MOD 12)/PS-1):
     IF VV>1 THEN VV=1: IF VV<-
    1 THEN VV=-1
112 VP=VP+VV: IF VP<0 THEN VP=0
    : IF VP>39 THEN VP=39
114 PLOT OH,OV: COLOR=15: PLOT
    HP,VP: GOSUB 18: GOTO 104
116 PLOT OH,OV
118 FOR X=1 TO 150:Z= PEEK (-16336
    ): NEXT X
120 IF OH=0 THEN 144
```

continued on next page

continued from previous page

Lines 122-160 control the scoring and determine the winner.

```
 122 SL=SL+1
 124 IF SL<>15 THEN 136
 126 PRINT " ",SL," ",SR
 128 PRINT "*********  ORANGE WINS  *
     ************"
 130 INPUT "ANOTHER GAME (Y/N): "
     ,A$
 132 IF A$(1)="N" THEN 162
 133 INPUT "PADDLE SIZE (1-3) ",
     PS: IF PS<1 OR PS>3 THEN 133
 134 GOSUB 2000: GOTO 44
 136 PRINT " ",SL," ",SR
 138 PRINT
 140 PRINT
 142 GOTO 54
 144 SR=SR+1
 146 IF SR<>15 THEN 154
 148 PRINT " ",SL," ",SR
 150 PRINT "*********  YELLOW WINS  *
     ************"
 152 GOTO 130
 154 PRINT " ",SL," ",SR
 156 PRINT
 158 PRINT
 160 GOTO 54
```

Lines 162-166 end play.

```
 162 TEXT
 164 CALL -936
 166 END
```

Lines 1000-1310 begin the program by asking the desired paddle size.

```
1000 TEXT
1010 CALL -936
1020 VTAB 7
1030 PRINT "            *** GRID BALL
      ***"
1040 PRINT
1042 INPUT "PADDLE SIZE (1-3) ",
     PS
1044 IF PS<1 OR PS>3 THEN 1042
1310 RETURN
```

Lines 2000-2070 poke in the paddle drawing machine language routine.

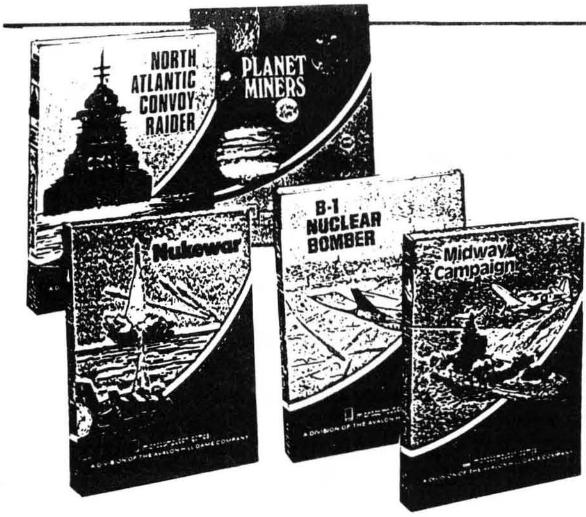
```
2000 POKE 4096,162: POKE 4097,0:
      POKE 4098,160: POKE 4099,0
     : POKE 4100,173: POKE 4101,
     255: POKE 4102,15: POKE 4103
     ,141
2010 POKE 4104,23: POKE 4105,16:
      POKE 4106,173: POKE 4107,0
     : POKE 4108,20: POKE 4109,141
     : POKE 4110,27: POKE 4111,16
2020 POKE 4112,173: POKE 4113,1:
      POKE 4114,20: POKE 4115,141
     : POKE 4116,26: POKE 4117,16
     : POKE 4118,173: POKE 4119,
     121
2030 POKE 4120,18: POKE 4121,141
     : POKE 4122,255: POKE 4123,
     255: POKE 4124,232: POKE 4125
     ,224: POKE 4126,21: POKE 4127
     ,240
2040 POKE 4128,18: POKE 4129,238
     : POKE 4130,11: POKE 4131,16
     : POKE 4132,238: POKE 4133,
     11: POKE 4134,16: POKE 4135
     ,238
2050 POKE 4136,17: POKE 4137,16:
      POKE 4138,238: POKE 4139,17
     : POKE 4140,16: POKE 4141,238
     : POKE 4142,23: POKE 4143,16

2060 POKE 4144,76: POKE 4145,10:
      POKE 4146,16: POKE 4147,162
     : POKE 4148,0: POKE 4149,200
     : POKE 4150,192: POKE 4151,
     3
2070 POKE 4152,240: POKE 4153,3:
      POKE 4154,76: POKE 4155,4:
      POKE 4156,16: POKE 4157,96
```

Lines 2080-2180 poke in the color sequences.

```
2080 A=34:F=153:G=4608: GOTO 2080
     +PS*2
2082 B=34:C=34:D=34:E=146: GOTO
     2088
2084 B=34:C=34:D=153:E=153: GOTO
     2088
2086 B=146:C=153:D=153:E=153
2088 GOSUB 2130
2090 A=41:B=34:G=4640: GOTO 2090
     +PS*2
2092 C=34:D=34:E=34: GOTO 2098
2094 C=34:D=146:E=153: GOTO 2098

2096 C=153:D=153:E=153
2098 GOSUB 2130
2100 A=34:F=221:G=4672: GOTO 2100
     +PS*2
2102 B=34:C=34:D=34:E=210: GOTO
     2108
2104 B=34:C=34:D=221:E=221: GOTO
     2108
2106 B=210:C=221:D=221:E=221
2108 GOSUB 2130
2110 A=45:B=34:G=4704: GOTO 2110
     +PS*2
2112 C=34:D=34:E=34: GOTO 2118
2114 C=34:D=210:E=221: GOTO 2118

2116 C=221:D=221:E=221
2118 GOSUB 2130
2120 GOTO 2190
2130 FOR I=G TO G+19 STEP 6
2140 POKE I,A: POKE I+1,B: POKE
     I+2,C
2150 POKE I+3,D: POKE I+4,E: POKE
     I+5,F
2160 NEXT I
2170 POKE G+24,A
2180 RETURN
```

Lines 2190-2370 poke in the paddle positions.

```
2190 FOR I=5120 TO 5200 STEP 40
2200 FOR J=0 TO 32 STEP 16
2210 FOR K=0 TO 12 STEP 4
2220 POKE I+J+K,4+K/4
2230 POKE I+J+K+2,4+K/4
2240 POKE I+J+K+1,(4+3*(I-5120)/
     10+5*J/2) MOD 256
2250 POKE I+J+K+3,(132+3*(I-5120
     )/10+5*J/2) MOD 256
2260 NEXT K: NEXT J: NEXT I
2270 POKE 5240,0
2280 FOR I=5248 TO 5328 STEP 40
2290 FOR J=0 TO 32 STEP 16
2300 FOR K=0 TO 12 STEP 4
2310 POKE I+J+K,4+K/4
2320 POKE I+J+K+2,4+K/4
2330 POKE I+J+K+1,(10+3*(I-5248)
     /10+5*J/2) MOD 256
2340 POKE I+J+K+3,(138+3*(I-5248
     )/10+5*J/2) MOD 256
2350 NEXT K: NEXT J: NEXT I
2360 POKE 5368,0
2370 RETURN
```



TO ORDER TOLL-FREE
1-800-258-1790
In NH call 673-5144

From Avalon Hill

NORTH ATLANTIC CONVOY RAIDERS ($15)

This game is a computer simulation of the Bismarck convoy raid of 1941. The computer controls the British convoys and British battleships. Will the Bismarck sink the Hood, only to be sunk in turn by the Rodney and King George V, as in history? Or, will the Bismarck cripple or sink the British Home Fleet and go rampaging through the convoy lanes? Your decisions will determine the fate of the Bismarck.

This SOLITAIRE game includes software and instructions for the following computers: TRS-80* Level II, 16K Memory Apple II*, Applesoft, BASIC, 16K Memory beyond BASIC Pet*, 16K Memory

NUKEWAR ($15)

NUKEWAR is a computer simulation of a nuclear confrontation between two hypothetical countries. You must choose the methods to defend your country: either by massive espionage efforts, or by building jet fighter-bombers, missiles, submarines, and anti-ballistic missiles. Meanwhile, your cold and calculating computer will choose its own strategy to defend its country while also trying to destroy you utterly! NUKEWAR is very fast-paced and easy to learn, and can be enjoyed equally by game players of all ages and levels of experience. Best of all, once the nuclear war is over, you can bring the two countries back to life and try it again!

This SOLITAIRE game includes software and instructions for the following computers: TRS-80* Level II, 16K Memory. Apple II*, Applesoft* BASIC, 16K Memory beyond BASIC Pet*, 16K Memory.

PLANET MINERS ($15)

PLANET MINERS gives one to four players the chance to compete with each other and the computer to stake valuable mining claims throughout the solar system in the year 2050. Each player must decide which ships to send to which planets and when to try "dirty tricks" like sabotage and claim-jumping. If there are less than four players, the computer takes the other parts. (It can even play all by itself!) Thus, PLANET MINERS can either be played solitaire or with friends.

This 1-4 player game includes software and instructions for the following computers: TRS-80* Level II, 16K Memory Apple II*, Applesoft* BASIC, 16K Memory beyond BASIC Pet 2001*, 16K Memory.

B-1 NUCLEAR BOMBER ($15)

This game gives you an opportunity to be the pilot of a B-1 bomber on a mission over the Soviet Union. You must fly the plane through the stiff Russian defenses to the target city, bomb it, and return home. Your computer controls the Soviet air defense bases with their almost unlimited numbers of MiG's (fighters) and SAM's (surface-to-air missiles). Your only chance to get through is to rely on the superior technology of your sophisticated ECM (electronic counter measures) and self-defense missiles. When all else fails, you can try violent evasive maneuvers.

This SOLITAIRE game includes software and instructions for the following computers: TRS-80* Level II, 16K Memory Apple II*, Applesoft* BASIC, 16K Memory beyond BASIC Pet*, 16K Memory.

MIDWAY CAMPAIGN ($15)

MIDWAY CAMPAIGN is a computer simulation of the battle for Midway Island. Your microcomputer controls a huge force of Japanese ships whose objective is to invade and capture Midway Island. If the Japanese can win air superiority over Midway, the success of the invasion is virtually guaranteed. If not, they will be forced to turn back to prevent the loss of irreplaceable troops who would be totally vulnerable in their invasion craft. In the actual engagement, the Japanese made several tactical errors which cost them the battle. Your computer probably won't make the same mistakes! You command the badly outnumbered and outranged U.S. Navy forces. Your only advantage is surprise.

This SOLITAIRE game includes software and instructions for the following computers: TRS-80* Level II, 16K Memory Apple II*, Applesoft* BASIC, 16K Memory beyond BASIC Pet*, 16K Memory

LONE STAR CORRAL

LONE STAR CORRAL is an S-80 program requiring Level II and 16K.

by Robert C. Hall III

The sun is starting to set over the far ridge; the only thing keeping you from supper is one last horse to fence-in for the night.

But he's a fast horse; he's so fast that you know where he was just a moment ago, and not now. You do know however, that he must be in one of the spots next to where he was last.

For example, if the "O" shows where the horse was last, then the "X" shows where the horse could be now.

```
X X X
X O X
X X X
```

Note: The horse can't move into a square with a fence already in it!

The object of this game is to completely box the horse in so that he can't make a move. This must be done in the fewest number of moves without placing a section of fence on the horse thus hurting him. Good luck, partner!

VARIABLE LIST

M (11,11) = Memory Board. This array is used to store the occupied spots. Values this array can contain are 0 for an empty spot and 1 for an occupied spot.

HX = The X location of the horse (from 0 to 11)

HY = The Y location of the horse (from 0 to 11)

WX = The X location of the fence (from 0 to 11)

WY = The Y location of the fence (from 0 to 11)

X = Input letter flag (X=0 or X=1)

Y = Input number flag (Y=0 or Y=1)

N1 = ASCII value of the input

Q = Random X move of the horse (from -1 to +1)

W = Random Y move of the horse (from -1 to +1)

A,B,T = Loop values

A$ = INKEY$ variable

```
       SOFTSIDE PRESENTS:
   **************************
   ** THE LONE STAR CORRAL **
   **************************
              BY
       ROBERT C. HALL, III
       BLACKSBURG, VIRGINIA

 Line 1: Executes the instruction subroutine.  Printing
      CHR$(28) changes the double width letters to single
      width letters.  Executes the graphic subroutine to
      occupy time required to dimension arrays and initial-
      ize variables.

1 GOSUB1000:PRINTCHR$(28);:GOSUB800
5 DIM M(11,11)
8 REMARK ** ARRAY M(I,J) = MEMORY BOARD **

 Line 9: Initialize the out-of-bounds spots on the memory board.

9 FORA=0TO11:M(A,0)=1:M(A,11)=1:M(0,A)=1:M(11,A)=1:NEXTA

 Line 10: Start of game loop.  Wipes the playing board clear.

10 FORA=1TO10:FORB=1TO10:M(A,B)=0:NEXTB,A
19 REMARK ** HX AND HY ARE THE LOCATION OF THE HORSE **

 Line 20: Starting location of horse is randomly chosen.

20 HX=RND(10):HY=RND(10)
100 CLS:REMARK ** DRAW THE BOARD **

 Lines 110-230: Draw the board on the screen.

110 PRINTCHR$(23);"  ** LONE STAR CORRAL **"
120 PRINT"  --0-1-2-3-4-5-6-7-8-9--"
130 FORI=65TO74:PRINT"  "CHR$(I)":"TAB(23)":"CHR$(I):NEXT
230 PRINT"  --0-1-2-3-4-5-6-7-8-9--"

 Line 231: Draw an "O" to represent the position of the horse.

231 PRINT@68+(64*HX)+(4*HY),"O";

 Line 233: Playing loop (max. times is 99 (idiot proofing)).
      Executes the subroutine to find the next legal move
      for the horse.

233 FORT=1TO99:GOSUB690
234 PRINT@932,"HORSE WAS";:PRINT@998,"AT (";CHR$(64+HX);RIGHT$(S
TR$(HY-1),1);")";

 Line 235: Set up input prompt.

235 PRINT@896,"ENTER MOVE";:PRINT@962,STRING$(2,176);:PRINT@960,
"?";
237 REMARK ** INPUT "R" TO END GAME **
239 REMARK ** GET PLAYERS MOVE AND CHECK FOR BACKSPACE **
```

Line 240: Complex input routine that scans keyboard for input. Once a key is hit, a check is made to see if it was a backspace or a resignation. If a backspace is detected, input variables are set to null, stars on screen, wait, then go back to the input prompt. If a resignation is detected, go print Lose message.

```
240 A$=INKEY$:IFA$=""THEN240ELSEIFASC(A$)=8THENX=0:Y=0:I$="":PRI
NT@960,"?xx";:FORTT=1TO100:NEXTTT:GOTO235ELSEIFA$="R"THEN300
249 REMARK xx EDIT ALL NON-DESCRIPT KEYS OUT OF INPUT xx
```

Line 250: Save the ASCII value of the input. Then check for a letter input by seeing if a letter has already been entered (X=0 no, X=1 yes). If a letter hasn't been entered and the input IS a letter, then set the letter input flag, calculate the screen position and print the input on the screen.

```
250 N1=ASC(A$):IFX=0ANDN1>64ANDN1<75THENWX=N1-64:X=1:PRINTCHR$(N
1);
```

Line 260: Same as Line 250 except the check is for a number.

```
260 IFY=0ANDN1>47ANDN1<58THENWY=N1-47:Y=1:PRINTCHR$(N1);
269 REMARK xx HAVE BOTH VALUES BEEN ENTERED xx
```

Line 270: If both inputs have been entered then the calue of this seldom used BASIC function will be 3. Thus if the input is incomplete, return to the input routine.

```
270 IFPOS(0)<3THEN240
274 REMARK xx MOVE THE HORSE AND POSITION FENCE xx
```

Lines 275-280: Move the horse and set the fence.

```
275 PRINT@68+(64xHX)+(4xHY)," ";
276 HX=HX+Q:HY=HY+W
277 PRINT@68+(64xHX)+(4xHY),"O";:REM xx HORSE LAST POSITION xx
280 PRINT@66+(64xWX)+(4xWY),CHR$(140)CHR$(191)CHR$(140);:REM xx
FENCE xx
```

Line 287: Clear the input flags. Set the fence on the memory board. The horse cannot go into spot containing a 1. Then check if the fence was placed in the same spot as the horse. If so, print the Lose message.

```
287 X=0:Y=0:M(WX,WY)=1:IF(WX=HX)AND(WY=HY)THEN300
```

Line 290: Continue with the next move.

```
290 NEXTT
```

Lines 300-305: Clear the bottom two lines and print the Lose message. Execute the prompt-wait subroutine, then start a new game.

```
300 PRINT@896,CHR$(30):PRINT@906,"TOO BAD -- YOU LOSE";
305 PRINT@960,CHR$(30);:GOSUB9998:CLS:GOTO10
659 REMARK xx SEE IF HORSE HAS A VALID MOVE xx
```

Lines 690-740: This subroutine finds a new move for the horse.

Line 690: Checks the spots which surround the horse. If a spot can't be found then player wins. This is relayed to the player and the prompt-wait subroutine is executed. Then a new game.

```
690 FORA=-1TO1STEP2:FORB=-1TO1:IFM(HX+A,HY+B)=0THEN700ELSENEXTB,
A:FORA=-1TO1STEP2:IFM(HX,HY+A)=0THEN700ELSENEXTA:PRINT@896,CHR$(
30);:PRINT@904,"YOU WIN IN";T-1;"MOVES";:GOTO305
699 REMARK xx FIND A NEW SPOT FOR HORSE xx
```

Line 700: Pick two random numbers between -1 and 1. Both numbers cannot be zero.

```
700 Q=RND(3)-2:W=RND(3)-2:IFQ=0ANDW=0THEN700
```

Line 710: Prohibits out-of-bounds moves by checking if the new location would have a value less than 1 or greater than 10.

```
710 IFABS((HX+Q)-5.5)>4.5ORABS((HY+W)-5.5)>4.5THEN700
```

Line 720: Check if the new location is already occupied.

```
720 IFM(HX+Q,HY+W)=1THEN700
```

Line 740: If all of the above tests are passed then return.

```
740 RETURN
799 REMARK xx GRAPHICS  xx    NOTE: USE DOWN-ARROW ON LINES
                                    ENDING WITH A QUOTE MARK
```

Lines 800-860: Graphics subroutine. The use of the down arrow on some lines saves having to add a CHR$(13)+STRING$(XX,8)

```
800 PRINT@262,CHR$(188);STRING$(8,131);CHR$(140);CHR$(176);"
 ";CHR$(188);STRING$(3,140);STRING$(7,131);STRING$(4,140);CHR$(1
91);STRING$(2,140);STRING$(2,176);"
";CHR$(143);STRING$(4,140);CHR$(188);STRING$(10,140);CHR$(188);C
HR$(143);
810 PRINTSTRING$(2,140);CHR$(143);"
     ";CHR$(191);"    ";CHR$(188);CHR$(132);"  ";CHR$(131);" ";C
HR$(191);CHR$(144);"
     ";CHR$(191);"    ";CHR$(131);CHR$(129);" ";CHR$(172);STRING
$(2,176);CHR$(191);CHR$(131);"
     ";CHR$(143);CHR$(188);
820 PRINTCHR$(176);"      ";CHR$(176);CHR$(188);CHR$(143);"
      ";CHR$(191);"      ";CHR$(191);"
    ";CHR$(176);CHR$(188);CHR$(143);CHR$(131);"      ";CHR$(131)
;CHR$(143);CHR$(188);CHR$(176);
830 PRINT@311,CHR$(152);CHR$(165);CHR$(160);CHR$(164);CHR$(26);S
TRING$(6,24);CHR$(160);CHR$(154);CHR$(140);CHR$(163);CHR$(135);C
HR$(130);CHR$(156);CHR$(144);CHR$(26);STRING$(10,24);CHR$(160);C
HR$(140);CHR$(129);" ";CHR$(176);"    ";CHR$(130);CHR$(148);
840 PRINTCHR$(26);STRING$(13,24);CHR$(160);CHR$(140);CHR$(129);"
      ";CHR$(160);"  ";CHR$(170);CHR$(26);STRING$(15,24);CHR$(16
0);CHR$(140);CHR$(129);"   ";CHR$(130);CHR$(188);CHR$(156);CHR$(
140);CHR$(134);CHR$(129);"   ";CHR$(149);CHR$(26);
850 PRINTSTRING$(17,24);CHR$(174);CHR$(145);CHR$(140);" ";CHR$(1
76);CHR$(152);CHR$(140);CHR$(163);CHR$(154);"       ";CHR$(170);
CHR$(26);STRING$(16,24);CHR$(130);STRING$(2,131);" ";CHR$(160);C
HR$(140);CHR$(129);"          ";CHR$(173);STRING$(15,24);
860 PRINTCHR$(176);CHR$(140);CHR$(129);:RETURN
999 REMARK xx INSTRUCTIONS xx
```

continued on page 75

ADVENTURER'S GUIDEBOOK

by James Garon

It is contrary to the code of the True Adventurer to give specific and unsolicited hints to fellow adventurers. However, some general hints which can be applied to almost any adventure might perhaps be appreciated. Three topics come to mind: shorthand, maps and mazes.

SHORTHAND

Many people work needlessly hard at their adventures, especially when the keyboard they are using is afflicted with "keybounce". This is the phenomena of typing what you think is SLAY DRAGON, only to find - just after pressing "Enter" or "Return" - that you really typed "SSLLAY DRAGOON". The computer makes doubting remarks about your ancestry, and you must retype the command. Most authors of adventure programs have provided the user with a useful shorthand. For instance, in all of Scott Adams' adventures as well as the "original" Colossal Cave adventure from Microsoft, the computer only looks at the first few letters of each word. The chart shows whether three or four letters are required. This means that the computer will respond to "SLA DRA" just as if you had typed "SLAY DRAGON" (or "SLANDER DRANOFACE," for that matter). Shorthand can save you much time and helps avoid typing mistakes.

In addition to 3- or 4-letter shorthand, you can often use single letter commands for movement. For "GO NORTH", you could, of course, use "GO NOR", but it is much easier to simply type "N" (caution: "GO N" and "NORTH" do NOT work in Scott's programs). Specifically, N, S, E, W, U, D will send you in the four main compass directions plus UP and DOWN.

While Microsoft allows "INVE" as the only abbreviation for "TAKE INVENTORY" (to see what you are currently carrying), Scott permits the use of "I" for this purpose.

In Microsoft's Adventure there are a few other fascinating features. In the case of most two-word sentences, you may enter the words in either order. Thus, "COIN GET" has the same effect as "GET COIN". Perhaps more useful is the fact that you may enter up to 32 characters at a time. In these 32 characters, you may string two-word sentences together as in the following: GO E GET BIRD GO N GO E SCOR. The final command in such a string may be a one-word command. In this example we have used the four-letter abbreviation for SCORE. Of course, you may still enter the commands one-at-a-time, if you wish:

```
GO E
GET BIRD
GO N
GO E
SCOR
```

waiting for the computer's response after each command. Depending on your current situation, you may decide to use both methods at different times in the adventure. For instance, if nasty dwarves have been throwing things at you lately, you would probably wish to move about slowly and cautiously, one command at a time. But if you are starting over (perhaps having recently been killed in the cave), you may wish to quickly skim through the first dozen or so moves while things are still relatively safe. If you've memorized an opening sequence of moves, it is handy to type in several moves at a time. (Notice that Colossal Cave understands either "GO N" or "N", but to make use of the 32 character command line, you must use "GO N").

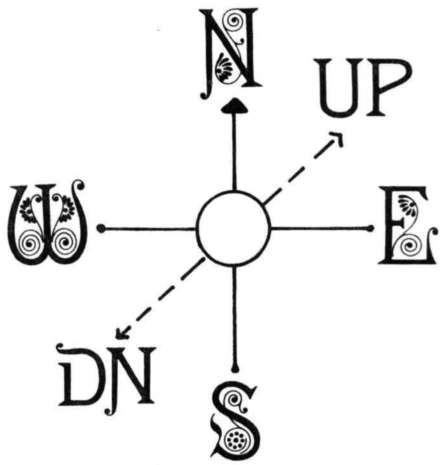
There is a somewhat tricky concept in Colossal Cave: to go in certain directions shorthand is not just a convenience; it is mandatory. Since the cave is quite convoluted in places, you sometimes need to go between two compass directions; for instance, southeast or northwest. But how is this possible when the computer ignores all but the first four characters of each word? The answer is: shorthand! In this case, you use SE, or NW, etc.

MAPS

"You can get lost without a map." Obvious, but true. Yet I have watched beginning adventurers wander around aimlessly, wasting precious turns, because they didn't think to record where they'd been. You may already have a map-making system which works well for you. If not, you may be able to use some tips which have proved helpful to me.

First of all, never start an adventure without a pencil and lots of paper. On your map, you'll want to remind yourself where you found various objects. I place objects (in parentheses) in the room in which I first find them. That way, if I die and have to start over, I can collect them without having to hunt for them. For those adventures which offer 6 primary directions (as do Scott Adams'), I use a special compass:

6-way compass



A sample portion of a fictional adventure map is included at the end of this article.

This map gives a variety of important information. For instance, you go West from the Barbershop (where the flute is located) to get to the Swimming Pool (where the dragon is doing the backstroke). Moving North from the Pool takes you to the Supply Closet, but to get back to the Pool, you must go West. Notice that a trip to the dungeon is a one- way proposition.

I invariably find that I need to redraw my map several times as I go along (that's what the extra sheets of paper were for!). As you discover more rooms, you may find that there isn't enough room to show them on your map. Don't be afraid to start over!.

MAZES

When you've mastered your shorthand and earned your merit badge in cartography, you may think you are prepared for any adventure, but there is always the danger of finding yourself suddenly in a maze. The inexperienced adventurer may tend to panic at this point and begin running off in all directions, getting more and more lost in the process. Many Mazes consist of lots ϙf different rooms, all labeled identically. This makes it nearly impossible to tell if you are moving from room to room or merely standing still.

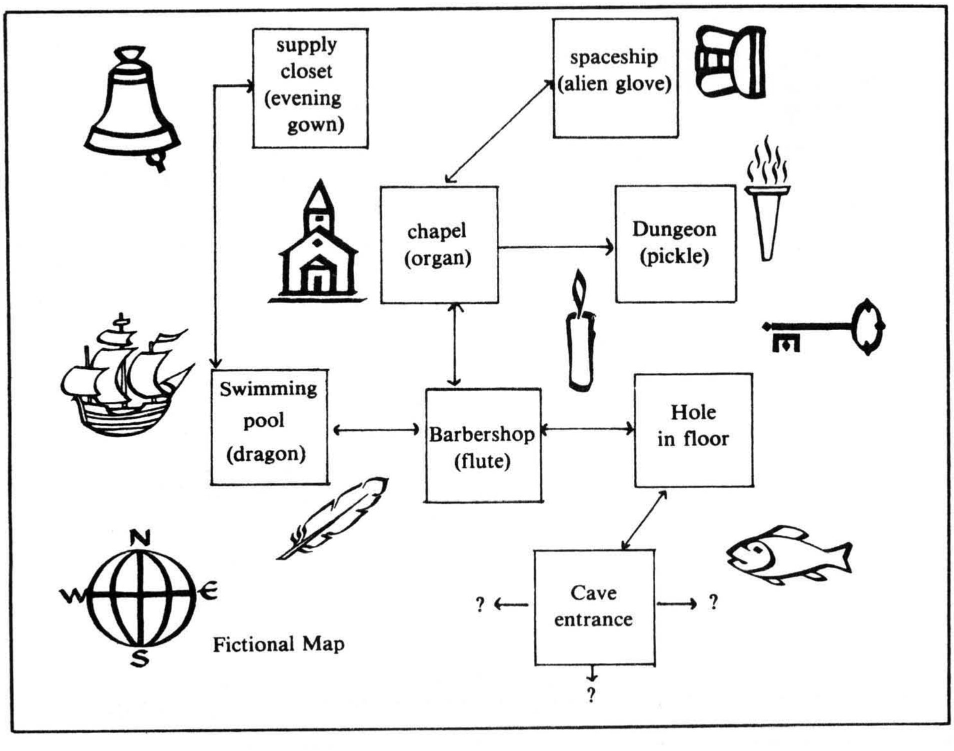
Whenever you locate a maze, the first thing to do is try to retrace your last movement. If that doesn't get you out of the maze, then the best thing to do is to quit and begin the adventure again. Using your map, which is also labeled to show the location of all important objects, make your way back to the spot where you entered the maze. On your way, collect as many objects as you can carry. Carefully enter the maze, and set down one of your objects. Now try to leave the room.

If the object you dropped is still in view, you know that the direction you moved in leads back to the same room. Keep trying different directions until the dropped object disappears. Now drop another. Continue in this way (naturally drawing a map of the maze as you go along), and you will soon get a feeling for how the various rooms of the maze are connected.

From your maze-map, you can then determine what pattern of directions (such as "N", "W", "S", "E") will get you through the maze in the fewest turns. In some of the larger mazes, you may need to make several mapping expeditions. If you can at least find an exit on your initial exploration, you can leave the maze and return with another armload of objects with which to explore further.

Well, that's about it for this month. If you've already played adventures, you know the unique thrill they produce. If you have never played adventure, I really envy you. For you, all the joy of discovery and the unique challenge of adventuring still lie ahead. All of the classic adventures are just waiting for you....

Title	Letters	Object
#1 Adventureland	3	13 treasures
#2 Pirate's Cove	3	2 treasures
#3 Mission Impossible	3	Save the Nuclear Reactor
#4 The Count	3	Kill Count Dracula
#5 Voodoo Castle	3	Remove a Curse
#6 Strange Odyssey	4	Treasures from Alien Universes
#7 Mystery Fun House	4	Recover stolen Spy Plans
#8 Pyramid of Doom	3	13 treasures
#9 Ghost Town	4	13 treasures
Microsoft Adventure	4	15 treasures/Grand Master Puzzle



From Peripherals Unlimited

FORM LETTER

Combines files from the TEXT EDITOR and MAILING LIST programs to enable the user to print out customized letters. A model letter is created using the TEXT EDITOR and special character lines within the body of the letter. When FORM LETTER is run, it replaces these special characters by information contained in the MAILING LIST records. For example, if the model letter contains "Dear $FIRSTNAME," and the MAILING LIST has an entry for 'John Jones' then the finished letter will read "Dear John". Letters can be printed out for any range of records: last names beginning with A through K, or Zip codes 60000 through 70000, or records with a specific code word. 48K Applesoft firmware and disk. $24.95

DAN'S DISK UTILITIES

Allows the user to directly examine and/or modify data on any track and sector of a diskette. Included with DDU is EXPUNGE, which can remove the DOS from a disk freeing up approximately 10K of disk space that could not otherwise be used. SINGLE DISK COPY is a copy program intended for those who do not have two drives, but still need to copy diskettes for back-up-purposes. Requires any size Apple with Integer Basic and a disk. $39.95

TEXT EDITOR

Fast, simple-to-use program designed for the small business or personal user. Text is processed in memory on a line by line basis. Some of its features include margin adjust, search, replace, and line centering. The program will hold about five printed pages in memory at a time. In addition it can be used in conjunction with the Mailing List and Form Letter program to create personalized letters. Requires 48K Applesoft firmware and disk. $64.95

MAILING LIST

Holds up to 500 names and addresses per diskette. You can split files between diskettes, permitting you to store A-K on one disk, L-R on another, and S-Z on a third.

Search and locate any name within 3 seconds. Entries can be sorted alphabetically or by zip code within 2 seconds for a complete diskette. It features user control of printed output recognition of duplicate names, and 7 entry fields.

This program may be used with the FORM LETTER program to print out personalized letters. Requires 48K disk. $39.95

1-800-258-1790

(In NH call 673-5144)

The Software Exchange Library

LEARN MICRO-COMPUTERS
A new multimedia information package. Includes text ("Understanding Microcomputers") plus high-quality cassette. For the beginner just starting in microcomputers. Covers all the basics quickly, easily and enjoyably. Companion tape includes chapter-by-chapter synopsis of the book. Great new idea for self-study. $14.95 plus $1

TAKE MY COMPUTER . . . PLEASE!
An uproariously funny book about the true-life misadventures of author Steve Ciarcia and his computer's inability to cooperate. Hardcover. $5.95 plus $1

MICROCOMPUTER POTPOURRI
A pocket-sized reference for the beginner. Has a really great glossary covering all the jargon. Full digest on understanding microcomputers. $3.95 plus $1

CALCULATING WITH BASIC
Use your computer to calculate home mortgage payments, interest rates, payback periods and more. Complete routines already worked out for problem solving using BASIC language. Also includes mathematics, finance and statistics, mechanical engineering and electronics. $8.95 plus $1

PERSONAL INFORMATION MANAGEMENT SYSTEM
When you're in business you've got a personal stake in how information is managed. That's because information is your key to success. PIMS will allow you to unleash the power of a microcomputer - and you don't have to become a programmer first. Use a computer for accounts receivable . . . accounts payable . . . maintain inventory records . . .run a mailing list . . .keep track of credit charges. These are just a few of the many things you can do with PIMS. It is a ready-to-use data base management program for computers like the Commodore PET and Radio Shack TRS-80. You can define and construct your own data bases. Each record can contain up to ten fields. You can search, list and sort. There's even a command that let's you sum columns of numbers. PIMS provides a valuable introduction to data base management. Complete source listing. $11.95 plus $1

APL - AN INTERACTIVE APPROACH
This revised second edition of APL - An Interactive Approach has been renamed to reflect the fact that several versions of APL are currently being offered. In recognition of APL's growing use in business applications, more examples have been included, and the body of the text itself has undergone a modest shift in orientation toward commercial uses of APL.

Additional functions and features now available in both the IBM and Scientific Time Sharing implementations have been included in this edition, and the chapters on workspace management and function definition have been substantially rewritten providing additional graphic aids to the student. Where appropriate, sections have been included on distinctive features of the IBM 5100 Computer.

For this edition, nearly all the example functions in the text have been placed in a workspace named 1 CLASS. If your APL system lacks this workspace, it may be obtained from Scientific Time Sharing Corporation. $16.95 plus $3.

LITTLE BOOK OF BASIC STYLE
Ideal reference for BASIC programmer, junior high to research scientist. Indexed, illustrated, 151 pages. By John M. Nevison $5.95 plus $1

Z-80 AND 8080 ASSEMBLY LANGUAGE PROGRAMMING
The best introduction to assembly language we sell. You should have experience in BASIC. By Kathe Spracklen $7.95 plus $1

INTRODUCTION TO LOW RESOLUTION GRAPHICS
Covers PET, TRS-80, Apple II. How to draw lines, create shapes, animated figures, charts. $11.95 plus $1

THE BASIC HANDBOOK
A definite reference/idea book. Explains over 50 favorite versions of the BASIC language in detail as used in micros, minis, and mainframes. Everything you need to know about the most important statements, functions, operators, and commands. From the same author as the TRS-80 Users/Learners Manual. $14.95 plus $1

Z80 INSTRUCTION HANDBOOK
Convenient pocket-size manual describes Z80 capabilities in easy-to-understand terms. Designed as a practical reference to mnemonics, machine codings, usage. For programmers of every level - beginner to professional - anyone working in Z80 machine or assembler language. Appendixed. $5.95 plus $1.

SARGON HANDBOOK
Complete documentation covering all algorithms in Sargon can be found in this guide book. Contains complete table of contents, block diagram of the program, four part introduction, Z-80 listing, index to subroutines. Fully annotated. $15.95 plus $1

TRS-80 ASSEMBLY LANGUAGE PROGRAMMING
Covers Z-80, and introduction to Assembly Language, the Radio Shack Editor/Assembler and T-Bug, and debugging methods; also explains how to move data, the use of arithmetic, compare, logic, and bit operators, shifts, strings, tables, input and output, and 12 commonly used subroutines. Well indexed and illustrated. $3.95 plus $1

THE SECRET GUIDE TO COMPUTERS
By Russ Walters A quickie course on computers. $5.95 plus $1

Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK
Over 100 usable subroutines, plus how to use them. $15.95 plus $1

TRS-80 DISK AND OTHER MYSTERIES
The serious programmers' guide to Disk BASIC and the wonders of NEWDOS+. $22.95 plus $1

6502 SOFTWARE GOURMET GUIDE AND COOKBOOK
Includes instruction set, floating point and decimal arithmetic, search and sort routines, and more. $12.95 plus $1

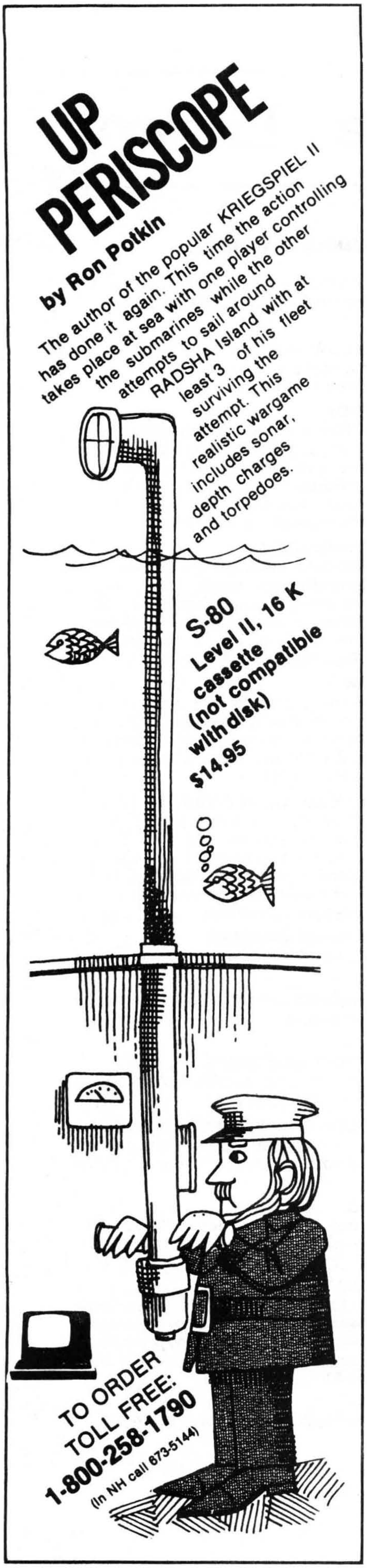
TRS-80 INTERFACING
What you need to know to connect your TRS-80 to the world. Assumes knowledge of some machine language programming. $8.95 plus $1.

PATHWAYS THROUGH THE ROM
The definitive guide to Level II BASIC. Includes Super Map by Fuller Software. The TRS-80 Disassembled Handbook by Robert Richardson, HexMem by J. Philip, Z-80 Disassembler by George Blank and DOS Map by John Hartford. $19.95 plus $1

LEARNING LEVEL II
The User's Guide to Radio Shack Level II BASIC. $15.95 plus $1

To order: Call Toll-Free 1-800-258-1790 (in NH call (603) 673-5144)

The Software Exchange & HardSide (Div. of Robitaille & Sons, Enterprises, Inc.), SoftSide Publications



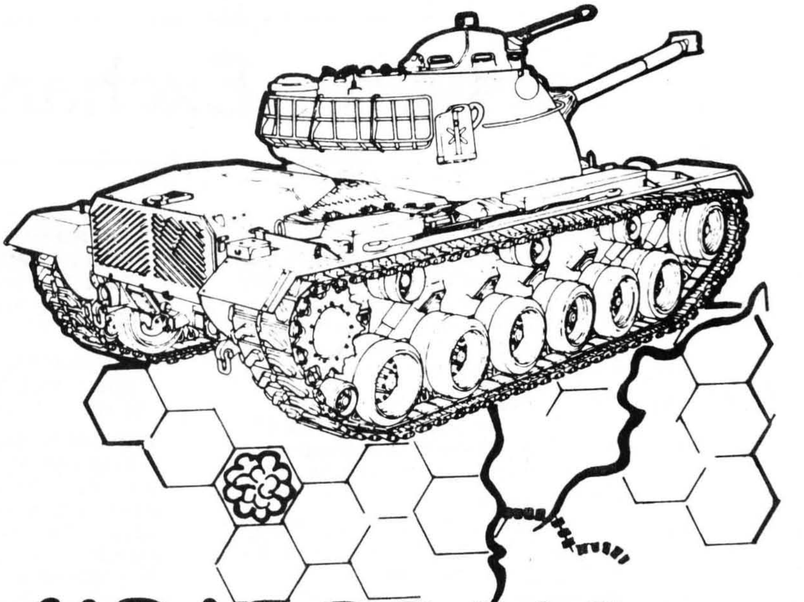
UP PERISCOPE

by Ron Potkin

The author of the popular KRIEGSPIEL II has done it again. This time the action takes place at sea with one player controlling the submarines while the other attempts to sail around RADSHA Island with at least 3 of his fleet surviving the attempt. This realistic wargame includes sonar, depth charges and torpedoes.

S-80
Level II, 16 K
cassette
(not compatible
with disk)
$14.95

TO ORDER
TOLL FREE:
1-800-258-1790
(In NH call 673-5144)



KRIEGSPIEL 2

by Ron Potkin

A Wargamer's delight! Throw away your pencils, charts* and dice. Let the computer do the dirty work while you concentrate on strategy. Kriegspiel II is a much improved, two-player version of the original—with machine language routines for extra speed.

Select one of 9,999 scenarios including towns, tanks, and terrain. Choose the number of mountains (up to 200), then sit back and watch as the computer generates your 31 x 32 game board, sets up the pieces, the towns, the mountains and a river.

Even the weather will become an important factor as play proceeds. To win, you must enter the Capitol City of your opponent (or reduce his fighting strength below half of your own).

KRIEGSPIEL II - Level II 16K$14.95 (cassette version only)

*Combat Resolution Tables, Basic Turn Allowances, Terrain Effects Charts, etc.

To Order
TOLL FREE
1-800-258-1790
(In NH call 673-5144)

SOUND FOR YOUR COMPUTER

by James Hagani

One fantastic feature of most home micro-computers is graphics. Extensive use of graphics can make even a drab game of hi-low more interesting. Include some cute sound effects and the program can become truly captivating. Recently, most programs for the S-80, have made use of sound via several different routines. Apple and Atari programs usually have sound too. If you own any of these machines, you should not be afraid of sound. In fact the use of sound should be as "basic" as a simple FOR-NEXT loop.

Each month, in this column, we will bring you simple subroutines which can be added to actual programs to enhance their appeal. This month we'll start with horses. Most everyone has a horse racing program, but chances are it lacks one important element of realism, sound. Just imagine seeing and hearing your longshot race his way to the finish line.

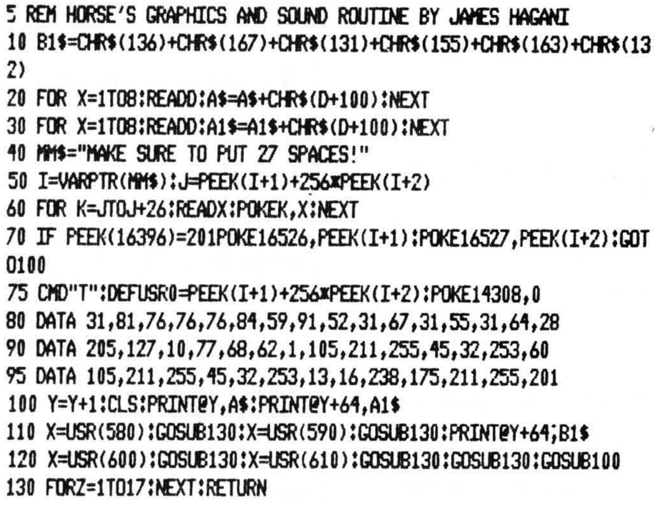
The S-80 subroutine uses a machine language sound generator which is poked into memory from BASIC. By calling different USR numbers, different sounds are created.

The Atari subroutine is much more simple. By using the versatile "SOUND" command, unique sounds are generated.

The Apple subroutine consists of a PEEK statement that produces a short click. When many of them are strung in a series, longer clicks are formed.

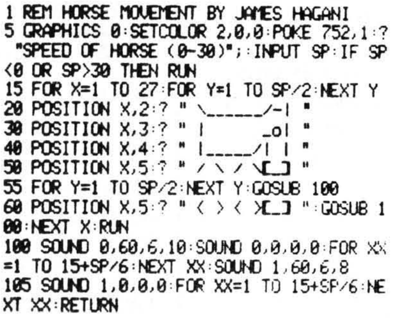
You no longer have to feel powerless when trying to employ this amazing and quite simple feature. With our help, you'll be, from now on, truly Safe In Sound.

S-80



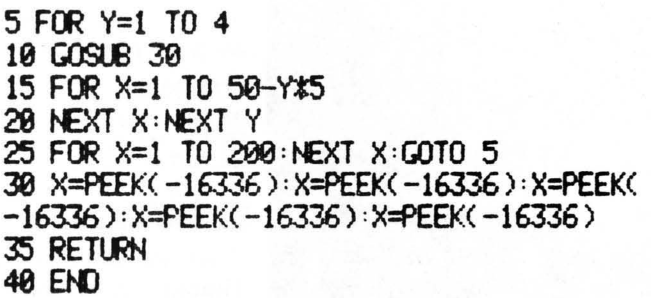
```
5 REM HORSE'S GRAPHICS AND SOUND ROUTINE BY JAMES HAGANI
10 B1$=CHR$(136)+CHR$(167)+CHR$(131)+CHR$(155)+CHR$(163)+CHR$(13
2)
20 FOR X=1TO8:READD:A$=A$+CHR$(D+100):NEXT
30 FOR X=1TO8:READD:A1$=A1$+CHR$(D+100):NEXT
40 MM$="MAKE SURE TO PUT 27 SPACES!"
50 I=VARPTR(MM$):J=PEEK(I+1)+256*PEEK(I+2)
60 FOR K=JTOJ+26:READX:POKEK,X:NEXT
70 IF PEEK(16396)=201POKE16526,PEEK(I+1):POKE16527,PEEK(I+2):GOT
O100
75 CMD"T":DEFUSR0=PEEK(I+1)+256*PEEK(I+2):POKE14308,0
80 DATA 31,81,76,76,76,84,59,91,52,31,67,31,55,31,64,28
90 DATA 205,127,10,77,68,62,1,105,211,255,45,32,253,60
95 DATA 105,211,255,45,32,253,13,16,238,175,211,255,201
100 Y=Y+1:CLS:PRINT@Y,A$:PRINT@Y+64,A1$
110 X=USR(580):GOSUB130:X=USR(590):GOSUB130:PRINT@Y+64;B1$
120 X=USR(600):GOSUB130:X=USR(610):GOSUB130:GOSUB130:GOSUB100
130 FORZ=1TO17:NEXT:RETURN
```

ATARI

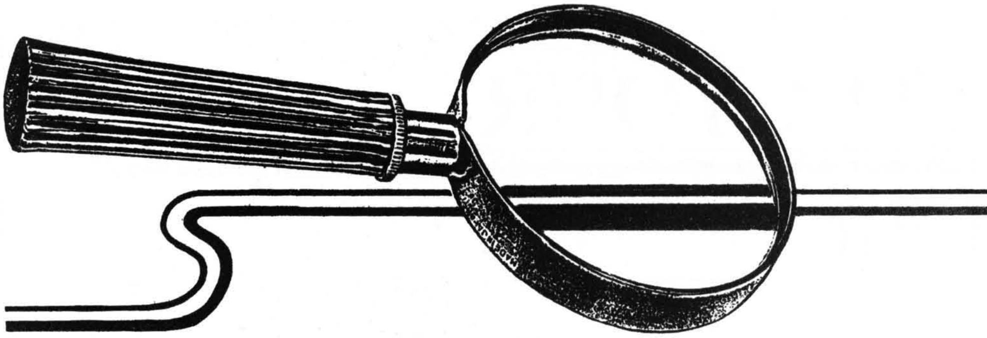


```
1 REM HORSE MOVEMENT BY JAMES HAGANI
5 GRAPHICS 0:SETCOLOR 2,0,0:POKE 752,1:?
 "SPEED OF HORSE (0-30)";:INPUT SP:IF SP
<0 OR SP>30 THEN RUN
15 FOR X=1 TO 27:FOR Y=1 TO SP/2:NEXT Y
20 POSITION X,2:? " \______/-| "
30 POSITION X,3:? " |      _o| "
40 POSITION X,4:? " |_____/| | "
50 POSITION X,5:? " / \ / \[_] "
55 FOR Y=1 TO SP/2:NEXT Y:GOSUB 100
60 POSITION X,5:? " < > < >[_] ":GOSUB 1
00:NEXT X:RUN
100 SOUND 0,60,6,10:SOUND 0,0,0,0:FOR XX
=1 TO 15+SP/6:NEXT XX:SOUND 1,60,6,8
105 SOUND 1,0,0,0:FOR XX=1 TO 15+SP/6:NE
XT XX:RETURN
```

APPLESOFT



```
5 FOR Y=1 TO 4
10 GOSUB 30
15 FOR X=1 TO 50-Y*5
20 NEXT X:NEXT Y
25 FOR X=1 TO 200:NEXT X:GOTO 5
30 X=PEEK(-16336):X=PEEK(-16336):X=PEEK(
-16336):X=PEEK(-16336):X=PEEK(-16336)
35 RETURN
40 END
```



SLEUTH

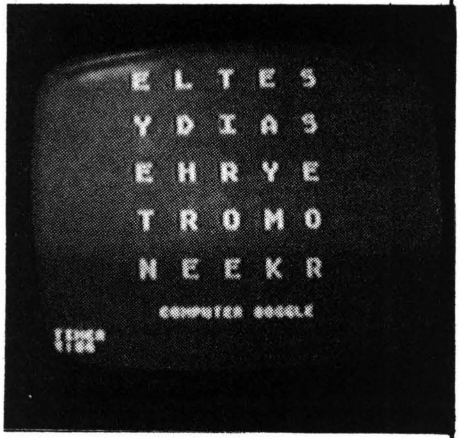
SLEUTH is an Atari program requiring at least an 8K memory.

by Paul Johnson

Gymnastic competition for a meeting of minds! Test your mental baud rate.

Sleuth is a computer adaptation of a popular party game. The computer displays a number of letters on the screen for three minutes. Players form words using adjacent letters. At the end of three minutes, the computer stops the game and the players compare their lists. Any words found by more than one player are eliminated, and players then keep score according to the number of letters in each word left on their list.

Programmers will find this game interesting for its demonstration of the use of the frame counter in the Atari computer as a real time clock.



```
E L T E S
Y D I A S
E H R Y E
T R O M O
N E E K R
  COMPUTER BOGGLE
```

```
10 REM --- SLEUTH  BY PAUL F. JOHNSON
```

} = Shift clear for clear screen

```
20 ? "}COMPUTER SLEUTH"
22 FOR T=1 TO 1500:NEXT T
```

L$ - Holds the letter vocabulary, selected to give a good word distribution. D$ is used to select letters for the display. Q$ is used for user input.

```
30 DIM L$(96),D$(2),Q$(10)
40 L$="AMOBJQFEHIYELERACSUKGLEYWDSNOEFBR
XIOPECADMYLTBIASLTPUEKNTDUOTIVENGROMAHSL
UWIGRTOCAIANEDAZVPHNSEI"
45 GOSUB 1000
```

} = Shift clear for clear screen

```
50 PRINT "}            COMPUTER SLEUTH"
60 ? :? :? :? "Enter 1 for SLEUTH,":? "o
r 2 for BIG SLEUTH ":? :? "Your choice";

70 TRAP 70:INPUT S:TRAP 0
75 IF S=1 THEN S=4:? "}":GOTO 360
80 IF S=2 THEN S=5:? "}":GOTO 360
90 GOTO 50
100 GRAPHICS 0
110 SETCOLOR 1,0,5
```

Starting position for cursor.

```
190 POSITION 2,14
200 FOR R=1 TO S
```

Position cursor for next letter.

```
205 IF S=4 THEN PRINT "    ";
206 IF S=5 THEN PRINT "   ";
210 FOR C=1 TO S
```

Generate Random Sound, use same random seed to select number.

```
220 I=INT(RND(0)*96)+1
225 SOUND 0,I,10,8
```

Select letter.

```
230 D$="  ":D$(1,1)=L$(I,I)
```

Add u if letter is Q.

```
240 IF L$(I,I)="Q" THEN D$(2,2)="U"
```

Print letter.

```
250 PRINT D$;
255 FOR T=1 TO 10:NEXT T
260 NEXT C
270 ?
280 NEXT R
285 SOUND 0,0,0,0
```

Graphics 34 - Same`as Graphics 2, except screen is not cleared. This is a text mode with ten rows of twenty double height, double width characters using a minimum of memory.

```
290 GRAPHICS 34
295 SETCOLOR 0,8,8:SETCOLOR 1,0,13:SETCO
LOR 2,0,0
300 PRINT "            COMPUTER SLEUTH"
310 PRINT :PRINT "TIMER"
```

POKE 755,1 - Sets character mode register to normal mode. Note: If you POKE a 4 into this location, all text on the screen will be printed upside down.

```
320 POKE 755,1
330 GOSUB 400
```

} = Shift clear for clear screen

```
340 GRAPHICS 0:PRINT "}  SLEUTH SCORING
TABLE"
350 GOSUB 800
355 ? :? "       Qu COUNTS AS ONE LETTER.
"
360 ? :? :? :? :? :? "Press RETURN to be
gin a round":INPUT Q$:GOTO 100
400 SECONDS=180
410 GOSUB 500
420 T=TIME
430 PRINT " ":PRINT INT(SECONDS/60);":";
SECONDS-INT(SECONDS/60)*60;
435 IF SECONDS=10 THEN PRINT "}";
440 GOSUB 500
460 IF SECONDS>0 THEN SECONDS=SECONDS-1:
GOTO 410
470 RETURN
```

500 Real time clock - The Atari computers use memory locations 18, 19 and 20 for a one-sixtieth of a second count that is updated by the television frame counter. This is stored as a three-byte binary number, with the most significant bit in location 18 and the least significant bit in location 20. Each count in location 20 represents one-sixtieth of a second. Each count in location 19 represents 4.266 seconds, and each count in location 18 represents 1092.266 seconds, or about 18.2 minutes. If you wish to reset the real time clock, simply poke 0 into each location.

```
500 TIME=INT((PEEK(18)*65536+PEEK(19)*25
6+PEEK(20))/60)
510 IF TIME=T THEN 500
520 RETURN
800 REM --- POINT TABLE ---
810 IF S=5 THEN 850
820 ? :? "NUMBER OF LETTERS: 3 4 5 6 7 8
+"
830 ? "                   ____________"
840 ? "POINTS:            1 1 2 3 5 11":
RETURN
850 ? :? "NUMBER OF LETTERS: 4 5 6 7 8+"
```

continued from previous page

```
860 ? "                         ----------"
870 ? "POINTS:                   1 2 3 5 7"
880 RETURN
1000 REM --- INSTRUCTIONS ---
```

} =Shift clear for clear screen

```
1010 PRINT "} DO YOU NEED INSTRUCTIONS";
:INPUT Q$
1020 IF Q$>"" THEN IF Q$(1,1)="Y" THEN 1
040
1030 RETURN
```

} = Shift clear for clear screen

```
1040 ? "}SLEUTH is a word game for two"
1050 ? "or more players. The computer wi
ll"
1060 ? "randomly pick letters and arrang
e"
1070 ? "them in a square like this:"
1090 ? :? " W T A C"
1100 ? :? " O E F Qu"
1110 ? :? " Y F J P"
1120 ? :? " A S D T"
1130 ? :? "You have 3 minutes to write d
own"
1140 ? "as many words as you can find in
 the"
1150 ? "square. For example, CAT, WET, F
ATE"
1160 ? "and QUAFFS can be found above."
1170 ? "As you can see, words must be ma
de"
1180 ? "of adjacent letters (any combina
tion"
1190 ? "of up,down,left,right,and diagon
al)."
1200 ? "You may not double back and use
the"
1210 ? "same letter twice in one word (S
AYS"
1220 ? "would be illegal).  Press RETURN
  ";:INPUT Q$
```

} = Shift clear for clear screen

```
1230 PRINT "}At the end of 3 minutes, ea
ch player"
1240 ? "reads his list of words. Any wor
d"
1250 ? "that more than one player found
is"
1260 ? "crossed off each list.  Any word
s"
1270 ? "remaining score as follows:"
1280 S=4:GOSUB 800
1290 ? :? "Play continues until one play
er"
1300 ? "reaches 50 points."
1310 ? :? "BIG SLEUTH is played by the s
ame"
1320 ? "rules. The letters are arranged
in"
1340 ? "a 5x5 square and scoring is hard
er:"
1350 S=5:GOSUB 800
1360 ? :? "Beginners should try the smal
ler"
1370 ? "version first.   Press RETURN.
 ";:INPUT Q$
1380 RETURN
```

Unlimited Level II USR Calls

UNLIMITED LEVEL II USR CALLS is an S-80 article for Level II systems.

by Charles C. Edwards

When writing a Level II BASIC. program using several machine language subroutines, I find it annoying to constantly repoke locations 16526-16527 with the addresses of the routines. So I set out to find a way to implement multiple machine language routines. Any method would have to meet the following criteria:

1. It must not require memory size to be set, so that it would be independent of the amount of memory on the machine.
2. It must retain the capability of passing and returning at least one argument.

I finally came up with the following routine:

```
CALL 0A7FH     ; GET THE
                 ARGUMENT
PUSH HL        ; GET IN INTO
                 IX
POP IX
LD L,(IX+1)    ; GET LOW
                 ORDER PART
                 OF STRING
                 ADDRESS
LD H,(IX+2)    ; GET HIGH
                 ORDER PART
                 OF STRING
                 ADDRESS
JP (HL)        ; BRANCH TO
                 SUBROUTINE
```

In order to use this routine, I poke 16526-16527 to point to it and say in basic, "X=USR(VARPTR(S$))" where S$ is a string that contains a machine language subroutine. This routine fetches the address of the string and branches to it. The rest of this article will show an example of its use and how to pass parameters to the called subroutine. I have used the "white-out" and "left-shift" routines from the Level II BASIC Manual as these are well-known and readily-understood subroutines.

First, we clear some string space and get the routine set up:

```
10 CLEAR100:S$="/////////////":I=PEEK(VARPTR(S$)+1)+256*PEEK(VAR
PTR(S$)+2):FORJ=ITOI+12:READX:POKEJ,X:NEXT:POKE16526,IAND255:POK
E16527,INT(I/256):DATA205,127,10,229,221,225,221,110,1,221,102,2
,233
```

Note that the routine is poked into the string rather than using string constructs. This is to ensure that the string will not be moved when string space management takes place. However, this is not necessary for the other routines.

Now, set up the "white-out" routine:

```
20 S0$="":FORI=0TO13:READX:S0$=S0$+CHR$(X):NEXT:DATA33,0,60,54,1
91,17,1,60,1,255,3,237,176,201
```

Now. the "left shift" routine:

```
30 S1$="":FORI=0TO7:READX:S1$=S1$+CHR$(X):NEXT:DATA175,203,21,20
3,20,195,154,10
```

Note that we have omitted the code for "CALL OA7FH". At run time we will insert code to "LD HL,(ARG)" at the beginning of the routine. The argument will then be in the HL register just as if it came from the ROM routine at OA7FH.

Finally, the code to invoke the subroutines:

```
40 X=USR(VARPTR(S0$)):FORI=1TO1000:NEXT:CLS:INPUT"ENTER INTEGER
TO BE LEFT SHIFTED";N
50 S1$=CHR$(33)+CHR$(NAND255)+CHR$(INT(N/256))+S1$:PRINTUSR(VARP
TR(S1$))
```

Line 50 contains the code to "LD HL,(ARG)". Remember that the argument must be in the Z-80 format of least-significant byte first, followed by the most-significant byte. It is a simple matter to expand on this technique to pass several arguments to the subroutine. Simply load them into the appropriate registers as needed. To return an argument, load it into HL and "JP OA9AH" just as you normally do.

“DRAWTO”and “HPLOT...TO...”

. . . for the TRS-80

by James Garon

Both the Atari and the Apple computers have the ability to draw a line connecting any two graphics points. While the TRS-80 cannot do this directly, it is possible to teach it to connect two points. We do this by means of a subroutine. The idea behind the subroutine comes from a subject which many of us remember less than fondly; algebra - specifically, the 2-point form of the equation for a straight line. Those of you who do not wish to be reminded of those dear delightful school days please skip ahead to the actual subroutine. The rest of us will make an effort to understand the silly thing.

Recall that an equation for a straight line is:

Y - Y1 = M * (X-X1)

where X1 and Y1 are the co-ordinates of one of the points (X1,Y1) on the line, and M is the “slope” of the line. (When I was teaching this stuff, I always had a hard time explaining to my class just why mathematicians chose the letter “M” to represent slope: perhaps because it is the first letter of the word “mathematics”.) Recall further that the value of M can be found if we know a second point on the line, (X2,Y2):

M=(Y2-Y1) / (X2-X1)

the so-called “rise-over-run”. Combining these two formulas gives the dreaded “two-point form” of the equation of a line:

Y-Y1 = (Y2-Y1)/(X2-X1)*(X-X1)

or, solving for Y,

Y = Y1 + (Y2-Y1) / (X2-X1) * (X - X1)

Now we would like to adapt this equation for computer use. To increase execution speed, we will use one-character variables A, B, C, D in place of the less friendly X1, Y1, X2, Y2. Our equation for Y will now look like this:

Y = B + (D-B) / (C-A) * (X-A)

The subroutine, beginning at line 1000, expects to know the values of A, B, C and D. A test is made at line 1000 to determine how steep the line is going to be. If it is close to horizontal, then stepping in the X-direction will give our line more points (thickness), whereas a line which is nearly vertical will have more points if we step in the Y-direction. Line 1010 causes the FOR/NEXT loop to STEP by +1 if C is to the right of A, and by -1 if C is to the left of A. (If C and A are equal, the program should have gone to line 1100.) Line 1020 draws the actual line. (To see what purpose is served by adding +.5, try leaving it out.) The last thing the subroutine does (line 1130) is to set A and B equal to C and D, respectively. This is useful for drawing another line which begins where the previous line left off. We'll see an example of this later.

continued on page 78

TONE GENERATION for the STEREO GENERATOR

This article contains a description of the machine language part of the Stereo Generator program elsewhere in this issue.

Dandelion Micro Products

The simple computer driven tone generator (using digital timing) toggles a speaker once per unit of time which produces a given frequency per second, etc. An improved tone generator toggles the speaker twice per unit of time (uptime and downtime). Control of the time width between toggles (either wide or narrow) gives the tone different timbre or voice.

Elaborate music systems use D/A converters, thus eliminating the need for digital time loops to define frequency. This allows the computer more time to perform multi-voice wave forms, amplitude, and envelope control.

The stereo generator uses simple digital timing to compute the frequency for two notes, plus achieves wave form enveloping. This is done as the program uniformly varies the time width between speaker toggles while performing the frequency. The user has limited envelope control in the lower frequency range, and should be able to experiment to find the best cycle that accomodates his piece of music. He always has the option to turn it off altogether and play his piece in the single toggle mode for two note music.

The stereo system is achieved by reading the cassette out port (LDA C020), then using the cassette recorder to amplify the signal to toggle a second speaker. The additional speaker works in similiar manner as the ‘on board’ speaker.

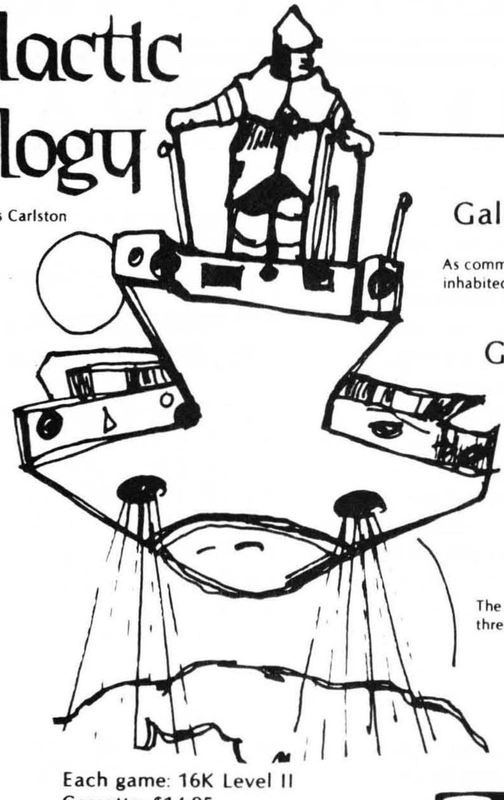
The stereo generator does not work with older Apple II's with revision zero boards. This is because the zero boards toggle the cassette port each time the ‘on board’ speaker is toggled.

STEREO GENERATOR
(Machine Language Routines)

ZERO PAGE LOCATIONS
NOTE-ONE EQU $6
NOTE-ONE ENVELOPE EQU $7
NOTE-TWO EQU $8
NOTE-TWO ENVELOPE EQU $9
DURATION-2 EQU 19
DURATION-1 EQU 1A
VOICE EQU 1B
D DURATION EQU 1D

Frequencies in 6 and 8 are enveloped by 7 and 9. Voice effect is set up by ‘Initialize Routine’

continued on page 74



Unlock the power of the Z-80

STAD

SYMBOLIC TRACE AND DEBUG

A powerful monitor for the TRS-80 with special Debugging. Single Step through the machine language programs or set up to three breakpoints, and look at this display format!

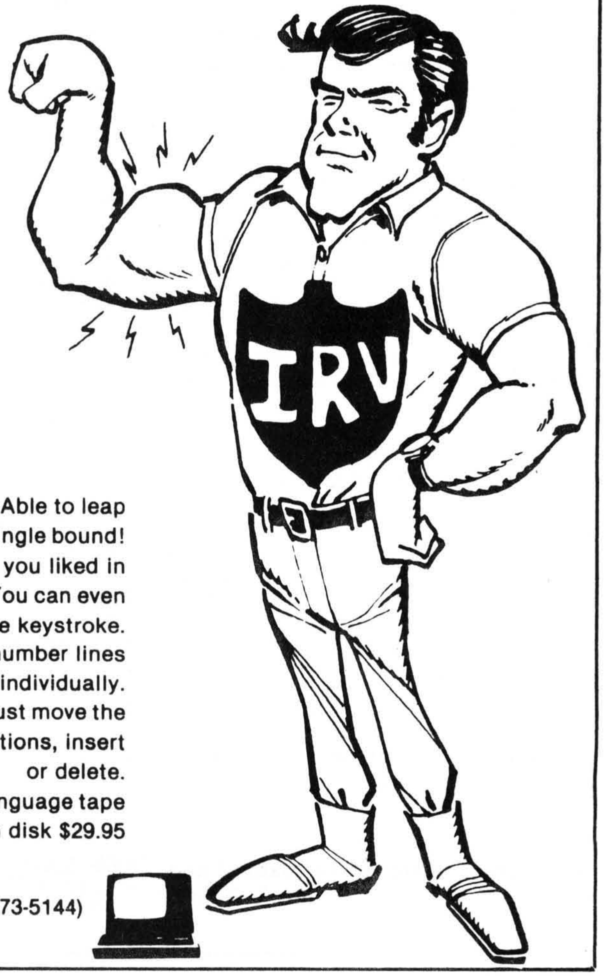
For tape and disk systems, 16, 32, and 48K on one cassette $24.95

A FIRST(0) LAST(FFFF)	ASCII dump
A FIRST 0	formatted ASCII dump
B	start of branch table
B VALA	display in decimal
B VALA VALB(0)	hex arithmetic
C	check system tape
D FIRST(0) LAST(FFFF)	dump hex
E FIRST(0)	edit memory
F FIRST LAST VALUE	find-byte
G BRKPTS (3 max.)	set breakpoints, continue
H FIRST LAST VALUE	find word
I PORT	read port
K	keyboard echo
L	load system tape
L SECTOR MEMORY COUNT(1)	load from disk
M FIRST LAST BLOCK	move memory
N	display symbol table
N 0	symbol table to tape
N VALUE	define value for symbol table
N FIRST 0	define start symbol table
O PORT VALUE	write to port
P	initialize memory blocks
P ENTRY	write memory blocks and start
P FIRST LAST	define a memory block
Q FIRST LAST	calculate checksum
R	display / modify registers
S FIRST LAST OPTION(0)	disassembler
T COUNT OPTION(6)	trace instructions
U FIRST COUNT OPTION(0)	unformatted tape I/O
V FIRST LAST BLOCK	verify memory
W SECTOR MEMORY COUNT(1)	write to disk
X FIRST LAST BLOCK	exchange memory
Z FIRST LAST VALUE(0)	zero memory

```
AF   BC   DE   HL   IX   IY   AF'  BC'  DE'  HL'  SP   PC
0044 0000 C000 B77C 6433 FFFF 0102 0000 4000 3FC0 41FC 4400
4400    LD      A,93
```

TO ORDER TOLL-FREE:
1-800-258-1790
(In NH call 673-5144)

Put IRV on your programming staff!



IRV is faster than Level II BASIC! More powerful than a speed typist! Able to leap ten subroutines at a single bound!

Input shorthand -- one keystroke can enter a whole line! The feature you liked in Level III BASIC and T-Short, now with all keys user programmable. You can even turn your cassette player on or off with a single keystroke.

Relocate a line by simply editing the line number -- renumber lines individually.

Video Editor allows cursor oriented editing of your programs. Just move the flashing cursor to the line to be edited and type the corrections, insert or delete.

IRV comes with manual and auto-repeat on every key. Machine language tape $24.95; on disk $29.95

To Order TOLL FREE **1-800-258-1790** (In NH call 673-5144)

RICOCHET

RICOCHET is an Atari program requiring at least 16K of memory.

by Paul Johnson

One of the classic scenes from Walt Disney's television series was a shooting contest in which the winner set up metal objects all around a tavern, then fired a shot that ricocheted from one object to another and rang the tavern bell.

If you have the same genius for analyzing the ballistic path of a bullet, then this is the game for you. You fire bullets into a closed container with objects inside. If a bullet strikes an object, it is either reflected or absorbed according to fixed rules, explained and demonstrated in the instructions. The purpose of the game is to determine, using deductive reasoning, where the 5 objects are with the least number of shots.

```
10 REM --- RICOCHET ---
20 REM BY PAUL F. JOHNSON, 3/8/80
```

100 Q$ - Is used for player inputs and is limited to 12 characters in length.

Box (9,9) - Is the array that holds the locations of the objects.

TIMER, SHOT, BOARD, GETMOVE, and SOUND - Are variables that contain the line numbers of the subroutines used in the game.

```
100 DIM Q$(12),BOX(9,9):TIMER=300:SHOT=4
00:BOARD=600:LET GETMOVE=800:LET SOUND=1
800
```

Close bracket in printout is actually SHIFT CLEAR to clear screen.

```
110 PRINT "}RICOCHET":? :? "A game of lo
gic and deduction":? :? "Do you need ins
tructions";:INPUT Q$
120 IF Q$>"" THEN IF Q$(1,1)="Y" THEN GO
SUB 5000
150 GOTO 1000
```

300 TIMER subroutine - The length of the delay is controlled by the value assigned to T. See lines 5200 to 5220 for examples.

```
300 FOR TIME=1 TO T:NEXT TIME:RETURN
```

400 SHOT subroutine - X (X1) and Y (Y1) are the horizontal and vertical positions. H and V hold the direction of the shot. If H equals 0 shot is vertical, 1 indicates right (lines 450-458) and -1 indicates left (lines 440-448). If V equals 0 shot is horizontal, 1 indicates down (lines 470-476) and -1 up (lines 460-468). F indicates the result of the shot. T1 and T2 are temporary variables used to test positions in the array to the sides of the present shot location. While balls are only located in positions 1 to 8, the array holds positions 0 and 9, as well, to simplify testing.

```
400 REM --- SHOT ROUTINE ---
410 X=0:Y=0:H=0:V=0:F=0:P=0
420 IF S<9 THEN X=S:V=1:GOTO 430
422 IF S<17 THEN X=9:Y=S-8:H=-1:GOTO 430

424 IF S<25 THEN X=25-S:Y=9:V=-1:GOTO 43
0
426 X=0:Y=33-S:H=1
430 X1=X:Y1=Y
435 IF BOX(X+H,Y+V)>5 THEN F=1:RETURN :R
EM --- ABSORBED! ---
440 IF H<>-1 THEN 450
442 T1=BOX(X-1,Y-1):T2=BOX(X-1,Y+1)
443 IF T1>5 AND T2>5 THEN F=2:RETURN
444 IF T1>5 THEN H=0:V=1
446 IF T2>5 THEN H=0:V=-1
448 GOTO 500
450 IF H<>1 THEN 460
452 T1=BOX(X+1,Y-1):T2=BOX(X+1,Y+1)
453 IF T1>5 AND T2>5 THEN F=2:RETURN
454 IF T1>5 THEN H=0:V=1
456 IF T2>5 THEN H=0:V=-1
458 GOTO 500
460 IF V<>-1 THEN 470
462 T1=BOX(X-1,Y-1):T2=BOX(X+1,Y-1)
463 IF T1>5 AND T2>5 THEN F=2:RETURN
464 IF T1>5 THEN V=0:H=1
466 IF T2>5 THEN V=0:H=-1
468 GOTO 500
470 IF V<>1 THEN STOP
472 T1=BOX(X-1,Y+1):T2=BOX(X+1,Y+1)
473 IF T1>5 AND T2>5 THEN F=2:RETURN
474 IF T1>5 THEN V=0:H=1
476 IF T2>5 THEN V=0:H=-1
500 X=X+H:Y=Y+V
505 P=P+1
510 IF X<1 OR X>8 OR Y<1 OR Y>8 THEN 540

520 GOTO 435
540 FOR T=1 TO 10*(10-P):NEXT T
542 IF P=1 THEN F=2:RETURN
543 IF Y<1 THEN S1=X:RETURN
544 IF X>8 THEN S1=8+Y:RETURN
545 IF Y>8 THEN S1=25-X:RETURN
546 S1=33-Y:RETURN
```

600 BOARD subroutine - Draws display.

See reference manual for graphics chart to put graphics characters in listing.

```
600 GRAPHICS 0:SETCOLOR 1,13,14:SETCOLOR
 2,13,14:SETCOLOR 4,13,2
620 ? "      1 2 3 4 5 6 7 8      | RICOCH
ET  "
625 ? "                           |"
630 ? "     ┌┬┬┬┬┬┬┬┬┐            |◆ = SHO
T"
635 ? "32   | | | | | | | | | |  9 |  REFLE
CTED"
640 ? "     ┌┬┬┬┬┬┬┬┬┐            |"
645 ? "31   | | | | | | | | | | 10|# = SHO
T"
650 ? "     ┌┬┬┬┬┬┬┬┬┐            |  ABSOR
BED"
660 ? "                           |"
665 ? "29   | | | | | | | | | | 12|"
670 ? "     ├┼┼┼┼┼┼┼┼┤            |"
675 ? "28   | | | | | | | | | | 13|"
680 ? "     ├┼┼┼┼┼┼┼┼┤            |"
685 ? "27   | | | | | | | | | | 14|"
690 ? "     ├┼┼┼┼┼┼┼┼┤            |"
695 ? "26   | | | | | | | | | | 15|"
700 ? "     ├┼┼┼┼┼┼┼┼┤            |"
705 ? "25   | | | | | | | | | | 16|"
710 ? "     └┴┴┴┴┴┴┴┴┘            |"
735 ? "                           |"
740 ? "      2 2 2 2 2 1 1 1      |"
750 ? "      4 3 2 1 0 9 8 7      |"
755 SETCOLOR 1,13,2:SETCOLOR 4,13,2
760 RETURN
```

800 GETMOVE subroutine - Accepts player input and processes it. M equals 1 for a shot, 2 to place or remove a ball, and 3 to end game (used in line 1060).

```
800 POKE 752,1
805 POSITION 28,8:PRINT "Type:"
810 POSITION 28,9:PRINT " 1 - 32 TO"
815 POSITION 28,10:PRINT "     SHOOT"
820 POSITION 28,12:PRINT " B TO PLACE"
825 POSITION 28,13:PRINT "      A BALL"
830 POSITION 28,15:PRINT " E TO END"
835 POSITION 28,16:PRINT "   GAME AND"
840 POSITION 28,17:PRINT "   SCORE"
850 POSITION 28,20:PRINT " Your move"
860 POSITION 28,21:? "          ";:POSITIO
N 28,21:INPUT Q$
870 IF Q$="" THEN 860
875 M=0
880 IF Q$(1,1)="B" THEN M=2
890 IF Q$(1,1)="E" THEN M=3
```

895 IF M ... This is a true/false test for the computer.

If M equals 0 , the test is false; otherwise, it is true and the program branches to line 930.

900 TRAP 860 - The VAL function expects a numeric value as the first item in Q$. If there is a letter, an error message is generated. TRAP 860 causes the computer to disregard the error message and go back to line 860 to get another Q$. TRAP 0 in lines 915 and 934 returns the computer to the normal error mode.

```
895 IF M THEN 930
900 TRAP 860
910 IF VAL(Q$)>0 AND VAL(Q$)<33 THEN M=1
:S=VAL(Q$)
915 TRAP 0
920 IF M=0 THEN PRINT CHR$(253);:GOTO 86
0
930 IF M<>2 THEN 980
931 GOSUB 980
932 POSITION 28,13:? "ADD (1) OR";
```

continued from previous page

```
933 POSITION 28,14:? "REMOVE (2)";
934 TRAP 934:POSITION 28,15:? "BALL";:IN
PUT Q:TRAP 0
935 IF Q<0 OR Q>2 THEN 931
936 POSITION 28,17
937 IF Q=1 THEN ? "ADD BALL AT";
938 IF Q=2 THEN ? "TAKE FROM";
940 POSITION 28,19:? "COLUMNK(1-8)";
945 POSITION 28,20:? "              ";:POSITIO
N 28,20:INPUT C
947 IF C<1 OR C>8 THEN ? CHR$(253);:GOTO
 945
950 POSITION 28,21:? "ROW (9-16) ";
955 POSITION 28,22:? "              ";:POSITIO
N 28,22:INPUT R
```

960 - Add or remove ball. C equals column, R equals row.

```
960 IF R<9 OR R>16 THEN ? CHR$(253);:GOT
O 955
970 R=R-8
979 REM --- CLEAR ENTRY WINDOW ---
```

980 - Clear entry display message and return (to line 1060).

```
980 FOR I=8 TO 23:POSITION 28,I:PRINT "
          ";:NEXT I:RETURN
```

1000 Initialization routine - Draw display and set up array.

```
1000 REM --- INITIALIZE ---
1010 GOSUB BOARD
1020 GOSUB 3000
1030 GOSUB 2000
```

1040 - Main program control routine.

```
1040 REM --- MAIN LOOP ---
1050 GOSUB GETMOVE
1060 ON M GOTO 1100,1200,1300
```

1100 Routine if shot was fired - Call SOUND and SHOT, print letters or symbols at proper locations. J equals shot number, CHR$(65plusJ) equals letter to be displayed. K equals current score.

```
1100 GOSUB SOUND:GOSUB SHOT
1110 POSITION 5+2*X1,1+2*Y1
1115 IF F=1 THEN PRINT "#":K=K+1:GOTO 10
40
```

1200 - If Q equals 1, place ball at location guessed. If Q equals 2, remove ball.

```
1120 IF F=2 THEN PRINT "◆":K=K+1:GOTO 10
40
1130 PRINT CHR$(65+J):POSITION 5+2*X,1+2
*Y:PRINT CHR$(65+J):J=J+1:K=K+2:GOTO 104
0
1200 T=BOX(C,R)
1205 IF Q=1 AND T/10=INT(T/10) THEN BOX(
C,R)=T+1:POSITION 5+2*C,1+2*R:? "●";
1210 IF Q=2 AND T/10<>INT(T/10) THEN BOX
(C,R)=T-1:POSITION 5+2*C,1+2*R:? " ";
1299 GOTO 1040
```

1300 End game and compute score - Check each location in the array. Every 1 is a wrong guess; everything over 10 is a correct guess; each 10 is an object that was not guessed. Score is 1 point for each clue displayed around the edge of the board and 5 points for each object not found. The lower the score, the better the rating, from PSYCHIC (line 1390 - 0 to 9 points) to THE PITS (line 1440 - over 26 points).

```
1300 REM --- SCORE ---
1310 FOR I=1 TO 9:FOR J=1 TO 9
1320 T=BOX(I,J):IF T=0 THEN 1360
1330 IF T=1 THEN 1360:REM WRONG GUESS
1340 IF T=10 THEN POSITION 5+2*I,1+2*J:?
 " ":K=K+5
1350 IF T>10 THEN POSITION 5+2*I,1+2*J:?
 " "
1360 NEXT J:NEXT I
1370 GOSUB 980:POSITION 28,10:PRINT "SCO
RE: ";K
1380 POSITION 28,12
1390 IF K<10 THEN PRINT "PSYCHIC !!":GOT
O 1460
1400 IF K<15 THEN PRINT "EXPERT !!":GOTO
 1460
1410 IF K<18 THEN PRINT "PRO !!":GOTO 14
60
1420 IF K<22 THEN PRINT "AVERAGE":GOTO 1
460
1430 IF K<27 THEN PRINT "BEGINNER !":GOT
O 1460
1440 PRINT "THE PITS!!!"
1460 POSITION 28,16:PRINT "  = GUESS":PO
SITION 28,18:PRINT "  = TARGET":POSITION
 28,20:PRINT "  = HIT"
1480 POSITION 21,23:PRINT "PLAY AGAIN";:
INPUT Q$:IF Q$>"" THEN IF Q$(1,1)="N" TH
EN 1490
1485 RUN
1490 GRAPHICS 0:END
```

1800 - Random tone sound routine.

```
1800 SOUND 0,50+150*RND(0),10,8:T=30:GOS
UB TIMER:SOUND 0,0,0,0:RETURN
```

2000 - Place 5 objects at random locations in array.

```
2000 FOR T=1 TO 5
2010 I=INT(8*RND(0)+1):J=INT(8*RND(0)+1)
2020 IF BOX(I,J) THEN 2010
2030 BOX(I,J)=10
2040 NEXT T
2050 I=0:J=0:RETURN
```

3000 - Zero all locations in array for new game.

```
3000 FOR I=0 TO 9:FOR J=0 TO 9:BOX(I,J)=
0:NEXT J:NEXT I:RETURN :REM --- RESET AR
RAY ---
```

5000 - Print instructions.

```
5000 REM --- INSTRUCTIONS ---
```

Close bracket in printout is actually SHIFT CLEAR to clear screen.

```
5010 PRINT "}BLACKBOX":?
5020 PRINT "This puzzle will test your d
eductive":? "abilities.  A box is placed
 before"
5030 ? "you.  Inside are 64 compartments
,":? "arranged as an 8x8 grid.  In 5 of"

5040 ? "those compartments targets have
been":? "placed.  Your task is to deduce
 the"
5050 ? "locations of the targets without
":? "being able to see inside the box.":
?
5055 GOSUB 5900
5060 ? "To obtain clues, you can fire sh
ots":? "horizontally into the sides of t
he"
5070 ? "box.  If the shot doesn't hit an
y-":? "thing it will pass straight throu
gh"
5080 ? "and exit from the opposite side.
":? "If it hits.a target head on, the sh
ot"
5090 ? "will be ABSORBED and will not ex
it.":? "If the shot passes within one co
m-"
5100 ? "partment of a target it will be"
:? "DEFLECTED 90 degrees away from the"
5110 ? "target - in which case, if its n
ew":? "path takes it toward another targ
et"
5120 ? "it may be deflected again or abs
orbed"
5130 ? "by the second target.":?
5140 ? "   If a shot would be deflected"
:? "before entering the box (by a target
"
5150 ? "along the box's edge), it is ":?
 "REFLECTED back.":GOSUB 5900
5160 ? "   In the following example, the
":? "targets and the path of the shots"
5170 ? "will be visible to help you unde
r-":? "stand the rules.":GOSUB 5900
```

5180 - Graphic demonstration of rules.

```
5180 GOSUB BOARD:POKE 752,1
5190 RESTORE 6000:T=2:GOSUB 5800
5200 T=800:GOSUB TIMER
5210 T=6:GOSUB 5800:T=1500:GOSUB TIMER
5215 T=10:GOSUB 5800:T=1500:GOSUB TIMER
5220 T=5:GOSUB 5800:T=5000:GOSUB TIMER
5230 GRAPHICS 0:? "Of course, when you p
lay you won't":? "be able to see the tar
gets or the"
5240 ? "paths of your shots.  Instead, t
he":? "computer will display symbols at
the"
5250 ? "shot's entry and exit points.":?

5260 ? "These are the symbols:":?
5270 ? "  A-Z = entry and exit points.":
?
5280 ? "  #   = shot ABSORBED (no exit).
":?
5290 ? "  `   = shot REFLECTED (entry an
d":? "        exit points identical)."
5300 GOSUB 5900
5310 ? "At any time you may make a guess
 as":? "to the location of a target.  Si
mply"
5320 ? "type 'B' and the computer will":
? "display a ball at the specified point
"
5330 ? "You may remove a ball the same w
ay."
5340 ? :? "To fire a shot, type the numb
er of":? "the entry point (1-32)."
5350 ? :? "To end the game and see the a
ctual":? "locations of the targets, type
 'E'."
5360 ? "You will receive 1 point for eac
h":? "clue you received around the edge
of"
5370 ? "the box, and 5 points for each":
? "target that you didn't find."
5380 ? :? "The lower your score, the bet
ter!":GOSUB 5900
5390 RETURN
```

5800 - Data for graphic demonstration.

```
5800 FOR I=1 TO T:READ X,Y,Q$:POSITION X
,Y:? Q$:NEXT I:RETURN
5900 POSITION 23,23:? "Press RETURN.  ";
:INPUT Q$:? "}":RETURN
6000 DATA 11,3, ,11,9,
6010 DATA 5,7,>,7,7,>,9,7,>,9,5,^,7,5,<,
5,5,<
6020 DATA 13,19,^,13,17,^,13,15,^,13,13,
^,13,11,^,15,11,>,17,11,>,19,11,>,21,11,
>,23,11,>
6030 DATA 11,19,^,11,17,^,11,15,^,11,13,
^,11,11,X
```

1

DEVELOPING DATA BASE

by Mark Pelczarski
S-80 translation by James Garon.
Atari translation by Rich Bouchard.

Handling of information is the most widespread use of computers. There are numerous data base programs available for almost any computer on the market, but it's often difficult to tell whether a particular program will meet one's specific needs. These packages usually cost in the neighborhood of $100, which makes them quite an expensive gamble; one that is often not justified by smaller applications. With packages from reliable software companies, the gamble usually pays off and the user wonders how survival was possible without the system; but wouldn't it be nice if one could have a custom designed data system — for free?

This is the first part in a regular series that SoftSide will be featuring about developing your own data base system. What we will do is put together all the functions of a data base system in a modular approach, explaining each part and the options available as we go along. The first couple of months will consist of creating a simple system that has the basic capabilities necessary for data handling: adding, deleting, changing, and printing information. As time goes on, we'll talk about ways to improve these functions, plus add new functions, such as sorting and selective searching, and explore the different possibilities with each of these. As a result you should have a data system that you can modify to fit your needs, and a little better understanding of how you can get your computer to do what you want without always relying on purchased software.

This month we'll start with the most basic capabilities: putting in information, saving it, and recalling it later. Before we do so, some decisions must be made about how the program will function. To begin, we'll consider the computer systems with which we'll be dealing: the Apple, S-80, and Atari. Included in this series will be listings for each of these machines. Conversions to other systems can be made from the BASIC which is closest to the one you use. In addition we must also consider that some of these systems occur more frequently with cassette storage than with disks. SoftSide's surveys have shown that although over 90% of our Apple owners have disk systems, only 60% of our TRS-80 readers use disk. We don't have figures for the Atari, but it's a safe assumption that there are more cassette users than disk users at this point. Fortunately, due to other factors as well, our data base program will be usable to those with either type of system. Since disk input and output(I/O) is somewhat slow and wearing on disks if in constant use, we can structure our system to read and write to disk only when we want to save or recall an entire file. The information will all be stored in the computer (RAM) in an array, or table, during normal operations. When beginning and ending operations, this information can easily be recalled or stored sequentially with tape or disk. This is called a Sequential Access File. Later we will look at other possibilities open to disk users.

The steps for building a program are to first define your goal, as we have already. The second step is to decide on an internal data structure; how you will store the information in the computer. The next step is to modularize your program, breaking it into logical sections that will be basically independent except for the variables on which they'll operate. The last building step is to actually write the code. There are other steps that follow when writing a good program: debugging, user-proofing, and field testing. Our program will be based on INFO developed by the same author as an interactive data base system for a time shared computer, so we'll assume these last steps are actually somewhat complete before we even start.

The internal data we will wish to keep consists of the array of data that will be stored, plus information about that array. The information necessary will be how many categories each record will have, what those categories are, and how many records there are. A record is considered to be one block of information, such as one person's name, birthday, phone number and address. The number of records would be the number of people for which this information is stored.

In our program, we use a two dimensional string array, I$, for our data. You can think of this as a table that can hold a set of characters (letters and numbers) at each location. (see figure 1) Other variable names we use will be NH, which stands for number of headings, or categories, H$, which will be a one dimensional array, or list, that contains the names of the headings, and NI, the number of items or records. Other miscellaneous variables we used were A$ for user answers to questions we'll ask, MX for the maximum number of items we'll allow, SS, a switch that will be explained later, F$ for filename, (in the case of disk users), and I and J for loop counters.

Due to the nature of the arrays we did a little playing around with NH and NI. Since an array when dimensioned has a zero element, we decided to make use of it. If you said DIM H$(2) you'd actually set aside three locations: H$(0), H$(1), and H$(2). Therefore, to avoid wasting space, NI and NH will always be one LESS than the number of items and number of headings. If there are three headings we'll talk about them as dimensioned above, from 0 to 2.

Because the Atari and a few other computers do not allow string arrays we had to do some other fooling around with that version of the program. Since dimensioning a string on the Atari tells it how many characters the

2

Figure 1

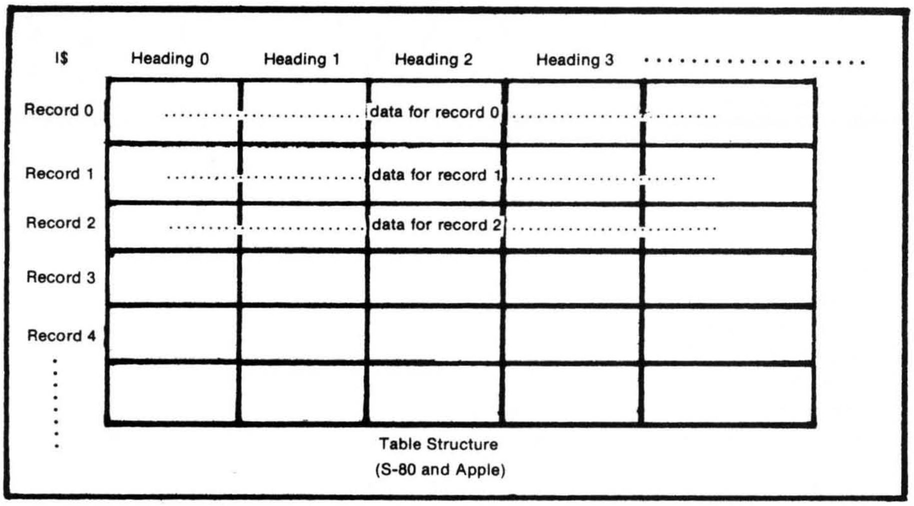


Table Structure
(S-80 and Apple)

string can hold, we can actually put all the data into one string! To do this and be able to tell where in the string the appropriate data lies, we have to specify a maximum length for data items and headings. (see figure 2) An unfortunate result for the time being is that each data item will use the amount of storage that you specify for the largest one. Variables added to the Atari version are IL, for item length, HL, for heading length, and RL, which is the record length computed from NH and IL, and is used to shorten some often used computations. Information will be stored one record after another in the string.

Dividing programs into logical sections before starting to code is often overlooked by novice and even some experienced programmers, but doing this with any program greatly simplifies the entire task. Each section can be treated as a program in itself, and any problems that occur can be isolated much more easily. Our program will consist of a main section, which will offer the user choices and transfer control to the other sections as needed. The other sections will be written as subroutines, using the GOSUB and RETURN statements for the transfer of control. Those subroutines, for now, will be one to read a file from disk or tape (line 1000), one to initialize a new file when there's nothing to read (line 1500), one that will write the file in memory to disk or tape (line 2000), one to print the current data on the screen or a printer (line 3000), and one to accept new data for the file (line 4000).

The main program, in lines 100 to 540, first asks if it is to read an existing file or initialize a new file. It then sends control to the appropriate subroutine at 1000 or 1500. Upon returning, available choices are offered to the user. At this time the choices will be: save the file in memory, print what's in memory, add more data, or quit. We'll add choices here as the program develops, and we'll add new subroutines also.

Notice that in line 500 you see the first occurrence of SS, the switch mentioned earlier. This is a save switch and its purpose is to give you a warning if you try to quit without saving the file in memory. This switch is set to 1 whenever you use the save option and set to 0 if you make any alterations in your file, which for now would mean using the add option. An implied warning is that you should save your file in intermediate stages if you are entering a large amount of data. If something drastic happens, like a power failure or a program bug, you won't have to redo the entire file. You will probably want to keep backup copies of important files also, using the save option for each.

Between lines 1000 and 1400 is the subroutine that reads an existing file. It should contain the appropriate disk or tape read commands for your system. The information that we'll want to know, therefore that which we need to save, is the number of headings, NH, the number of items, NI, and then the list of headings and all the data that was stored, H$(0-NH) and I$(0-NI,0—NH). The arrays are dimensioned here and loops are used to read the headings and data. Note that the Apple uses D$, which is given a value of Control-D, to specify disk operations. The Atari version must read two other pieces of information, NL and IL, but doesn't need loops to read the headings and data since they are all stored in one huge variable. Note also that MX, the maximum number of items, is given a value. Later we'll make more use of this variable and have the computer calculate the largest value it can use.

The subroutine that starts a new file is in lines 1500-1600. It is necessary for initialization of the variables that describe the file.

The subroutine in lines 2000-2200 writes the information in memory to tape or disk. Again, the appropriate commands to write to disk or tape should be given here. This subroutine and the read subroutine at 1000 will be the only two which access disk and tape and will contain the only differences for those two types of users. All the information that the computer will later need to know about the file is saved sequentially here. Note that you may change the file name before saving, meaning you can save a file in two different forms. This may be useful in the future if you want to save a file sorted by two different keys.

Lines 3000-3900 contain the print routine. Line 3050 starts the process of printing to the screen, and line 3400 starts the process of using a printer. Printing starts at record 1 (actually in record 0) and for each record every heading is listed with the corresponding data. This subroutine will undergo

String Structure
(Atari)

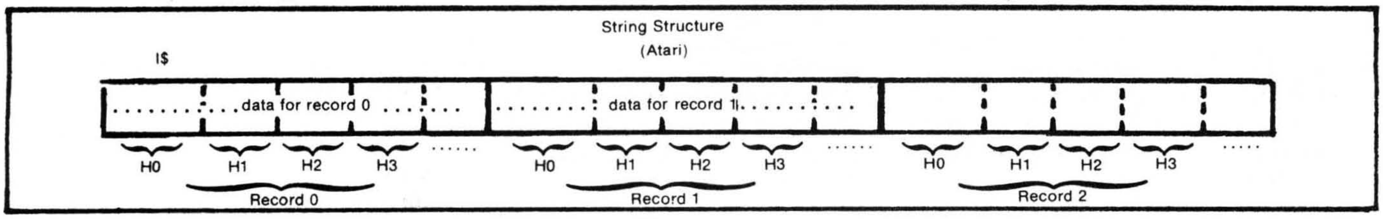
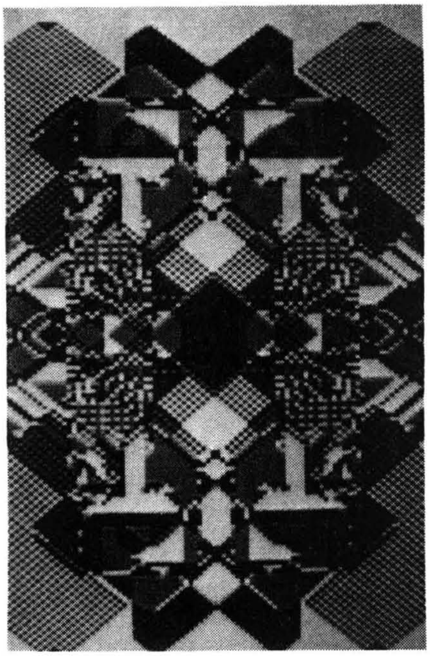


Figure 2

continued on page 66

VIDEO EASEL

VIDEO EASEL for the Atari - a review.
by James Garon



There are two main parts to this fascinating program cartridge: Life and Drawing. Life is a mathematical diversion in which "cells" pass through one generation after another; living, dying, and giving birth according to certain rules. The rules are clearly explained in the instruction manual, which also provides a list of books and magazine articles for further reading. Certain popular patterns are made available in a "menu", such as: Gliders, Factorys, and I-beams. These may be placed anywhere on the screen. You may also specify the position and length of horizontal, vertical, and diagonal lines. If none of these options suits your fancy, you can draw your own starting patterns in normal or "quad" mode. In quad mode, the line you draw with the joystick is reflected so that you actually draw four symmetrical lines at once.

Whatever option or combination of options you use to create an initial "colony", you need only give the START command and the program will take over -calculating and displaying an amazing 95 generations per minute. Since all of this takes place in four colors, it is an incomparable implementation of Life; however, in this reviewer's opinion, the Drawing program is even more engrossing.

To begin with, there is ordinary drawing. The joystick moves the cursor while the fire-button determines whether a line is drawn or not. The excitement starts when you have drawn a simple pattern and then give the command to START. Instantly, the computer begins repeating what you have drawn at such a blistering speed that the screen seems to take on a life of its own. By pulling back on the joystick, you can slow this activity down to the level of human perception. Several things then become apparent:

Your original pattern is being repeated in a specified direction which is determined by the starting and ending cursor-positions in your original drawing. (Thus the same drawing will produce different results when the cursor is moved to different locations before STARTing.) The pattern is repeated in this direction until it encounters the edge of the screen, at which time it will be reflected.

When a line crosses a previously drawn line, it produces a new color at the intersection. A third line crossing the intersection of the first two yields still another color. A little experimentation and some thoughtful hints from the manual will allow you to create magnificent "living" paintings.

When all this is done in four-fold symmetry, the results are truly impressive.

Six built-in paintings are provided, to be run alone or in combination with your own drawings.

Any pattern you produce can be used as the starting generation for the Life portion of the program, and, conversely, at any time you wish, you may halt the Life program and use the current generation as the beginning of a Drawing - either normal or quad.

Perhaps the most enjoyable feature of the cartridge is the capability of changing the three main colors. With a joystick in slot #2, you may cycle through the sixteen possible background colors by pressing forward. If you hold the fire-button while pressing forward, you will change the intensity of the background through the eight possible shades from dark to light. Similar changes take place when the stick is pressed to the left and right, but these changes affect different portions of the drawing.

This reviewer played with quad-drawing for many hours, skipping such delights as Charlie's Angels and Vega$, but perhaps the best recommendation for Video Easel comes from my wife - who is notoriously indifferent to all aspects of computers. Not only did she sit entranced in front of the TV while the patterns and colors changed, but, the next morning, as she saw the review copy of the cartridge being slipped into the old briefcase, she said, in a mournful voice, "Does it have to go back so soon?"

VIDEO EASEL is available from the Software Exchange for $39.95

Make your SoftSide™ Library Complete with . . .

SOFTSIDE: S-80 EDITION

***+ November 1978** End Zone, Level I or Level II Conversion article, Trolls Gold, Shopping List, Octal to Hex.

***+ December 1978** Santa Paravia, Biorhythms, Mortgage Calculation, Spelling Bee, Chess Clock.

January 1979 Round the Horn, Ten Pin, Bowling, Kiddy Slot, Computa Sketch, Writing Game Articles.

February 1979 Income Tax Program, Concentration, Elements Quiz.

March 1979 Tarrot, Metric/English conversion, Jig Saw, Fog Index, Dive Bomb, Personal Finance Part 1.

April 1979 Safari, Personal Finance Part 2, Don't It Make My Brown Eyes Blue, Rabbits and Foxes, Spring Flowers

May 1979 Dog Star Adventure, Awari, Letter Crunch, Math Drill, Super Sub.

June 1979 Atlantic Balloon Crossing, Nim, German Word Quiz, Appliance Record System, Entrapment.

July 1979 All Star Baseball, Yahtzee, Tank, Spelling Bee, Collision.

August 1979 Melt Down, Shooting Gallery, Story Book, Code Tutor, Cryptogram.

September 1979 Engineer, Treasure Dungeon, Hangman, Pyramids.

October 1979 Westward 1847, Battleship, Reaction Time Indicator, What Kind of Word?

December 1979 Oil Baron, Drag Rce, Christmas Show and Tell, Index to Level II Manual, Towers of Hanoi, Slalom.

January 1980 Moving Maze, Dante's Inferno, Bomber, Reflection, Duck Hunter.

February 1980 Deadstick, Backgammon, Parachute, Play It Again Sam, Deep Six.

March 1980 Broadway, System Emulator, Line Four, Add, Sonic Torpedos.

April 1980 Stud Poker, Draw Poker, Compusiot, ICBM, Solitaire.

May 1980 Star Trek III, Encounter in the Near Tholian Sector, Day of The Week.

This issue only:

Magazine	$5.95
with cassette	$14.95
with diskette	$19.95

June 1980 MicroMillionaire, Galaxy Combat, Numbagels, Simple Sound Interface.

July 1980 Adventure Builder, Adventure Interpreter, Lost Dutchman's Gold, Pinball.

PROG/80

PROG/80 The magazine for serious Programmers of the S-80 Computers.

March 1979 Simple Simon, String$ Fever, Level II Variables, Data Statements, INKEY Routines, Super Graphics with Disk

May 1979 Boomer Box, Clock Routines, Rescue, Scattergram/Correlatin, ROM Keyboard Routines, Super Graphics without Disk, Super Graphics Aid, SQUISH/BAS, INIT.

July 1979 Sound Subroutines, Machine Language Video, High Density Data Storage, Histograph and Basic Statistics, Talking Banko, Plot, Cassette Controller, TRS-80 Music.

September 1979 DOS, Kids Stuff, Disk BASIC, Renumber, Squish 2, Data and Time Routine.

December 1979 Pascal Review, Which Language?, Devices, Hex/Decimal Conversion, DIM Statements, Variable Uniqueness, Computer Telephone Dialer, Preference Poll, Reviews of Model II, Macrotronics, M-80, and Exatron Stringy Floppy.

February 1980 The Source, Timesharing, Tool, Program Write Protection, NAME, SuperZap Review, HexMem Monitor, High Speed String Handling, CTR-41 Modifications.

April 1980 Level II Keyboard Redefinition, Writing System Tapes, Voice Synthesizer Lab, BASIC File Utility, Devices (2), First Time on the Source, Forum 80, Repeatable Randomness, Zero Slash Killer, Proofread

June 1980 All Purpose Print Routine, Z-80 Disassembler, Printer Terminal, Random Accessing Techniques, Form Letter, BASIC To Electric Pencil Conversion, New NEWDOS Command-LOC(X), Branching Functions in APL80.

July 1980 Fortran Subroutines, TRS-80 Program Storage and Useful Corrollaries, Splat-An Introduction to Tiny Comp, Program Portability, VARLIST.

* Cassette version not available
+ Disk version not available

SOFTSIDE: APPLE EDITION

*+ **January 1980** Dog Star Adventure, Reverse, Giant Clock, Trolls Gold, Loan Amortization.

*+ **February 1980** Supernim, State Capitols, Elementary Math, Connection, Musical Scales, Sort.

*+ **March 1980** Care and Feeding of Integer HI-RES, Renumber and Merge, Acey-Ducey, Treasure Hunt, Bouncing Ball Catcher, Switch Puzzle.

* **April 1980** Applesoft Ampersand, Shootout, Jigsaw Puzzle, Space War, Melody, Display COntrol Characters.

* **May 1980** Invaders, Small Marquee, Magic Cave, Black Box, Fifteen Game, Hyperboloid.

* **June 1980** Dogfight, HI-RES Drawing, Dodger, ROM the Robot, A Hello Program.

July 1980 Pork Barrel, NO MIS, ROM the Robot, LORES Printout, It's a Small World.

* Cassette version not available
+ Disk version not available

back issues of:

√SoftSide™: Apple

√SoftSide™: S-80

√Prog 80

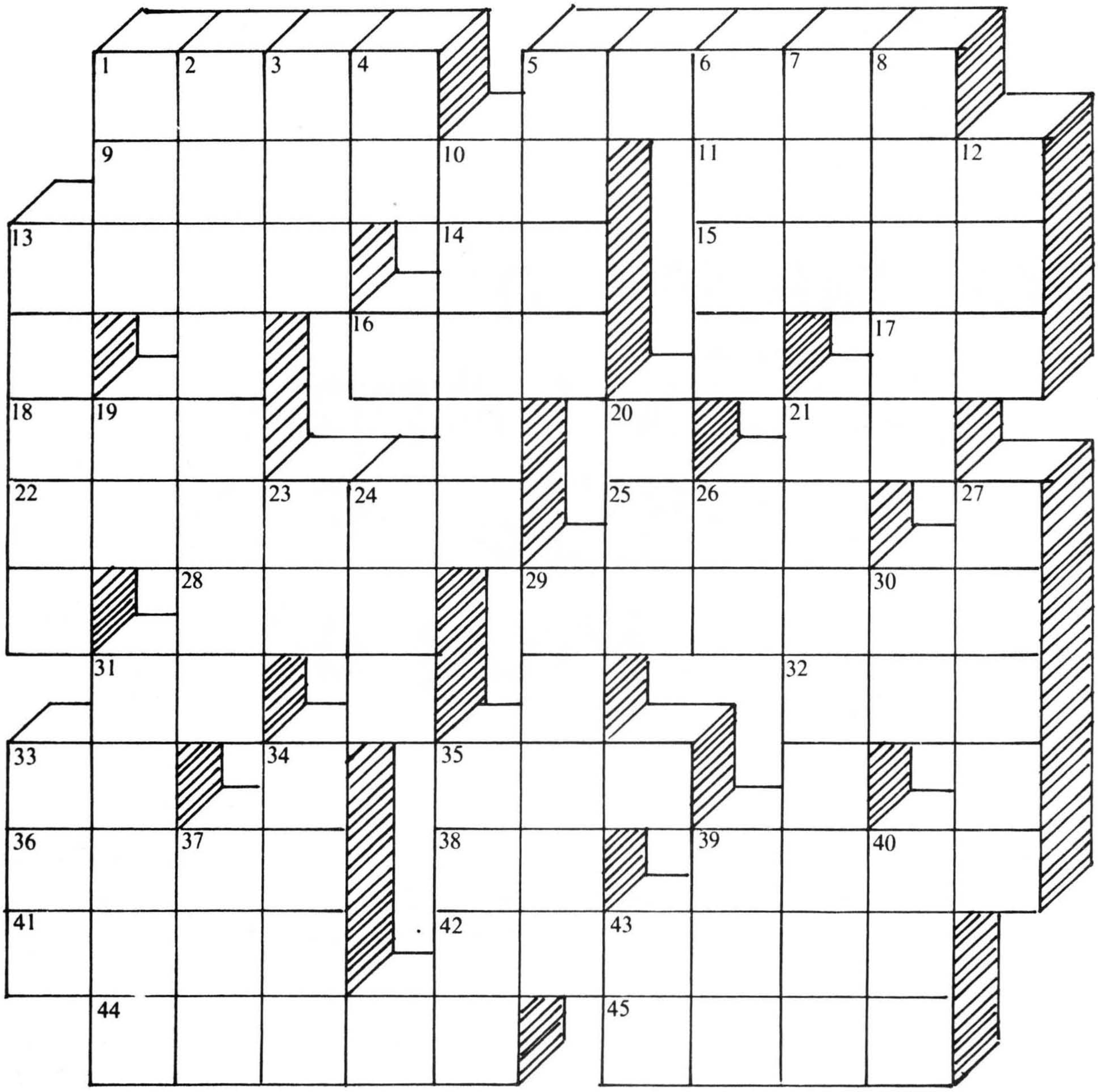
Programs, documentation and lots more! If you have missed out on any past issues, NOW'S THE TIME TO ORDER!!!

SoftSide, S-80 and Apple Back Issues:
Magazine only $2.50
Magazine with Cassette Programs, $9.95
Magazine with Disk Programs, $14.95
Prog/80 Back Issues:
$4.00 (Includes First Class Postage)

MasterCard VISA

To order TOLL FREE
1-800-258-1790
(In NH call 673-5144)

CROSSWORD PUZZLE



ACROSS:

1. Integrated Circuit

5. Beginner's All-purpose Symbolic Instruction Code

9. This comes in two types, even and odd

11. If you eat too much uranium, you might get atomic_ _ _ _

13. A gram of prevention is worth a kilogram of_ _ _ _

14. Two-thirds of a feline

15. Touch

16. President's nickname

17. Science Fiction (abbr.)

18. A bit can take on either of two values. We can think of these as high or ———

21. Operating System (abbr.)

22. An unusual color for a computer - especially an Apple

25. ——— is to S-80 what PROGRAM LINE EDITOR is to the Apple

28. Kanga's offspring, according to A. A. Milne

29. Returns are to Apple and Atari as these are to the S-80

31. Two thirds of the Atomic Energy Commission (abbr.)

32. Giant bird from "1001 Arabian Nights"

33. You and me

35. It is a scientific fact that people who wear this can learn to program as well as anyone else.

36. Shoo!

38. Two thirds of a bit

39. Jai ——— plays a major role in one chapter of the book "Take My Computer . . . Please!"

41. Ireland

42. You should not eat corn on this while programming, as the butter will drip on the keys (2 words)

44. In the interest of reducing errors, we recommend that you do not use this spelling

45. There are at least two kinds - good and bad

DOWN:

1. Central Processing Unit (abbr.)

2. What software specialists blame program malfunctions on

3. Anger

4. Ratio of circle's circumference to its diameter

5. Eight bits; smallest unit of information

6. Lock up important posessions here

7. Cools your drink

8. Popular game available for S-80, Atari, and Apple

10. A helpful way to visualize an array

12. One of several fairy tale creatures who climb into your 1-down while you are asleep and put bugs in your programs

13. Allows multiple statements per line number

19. Companion to AND and NOT

20. Name given to connector on a computer cable

21. Attempting to set an integer variable to 32768 or greater will cause this error

23.. Regarding 18-across, we can also think of "yes" and ——

24. Possibly the only One who can write a bug-free program on the first attempt.

26. Two thirds of a Cathode Ray Tube (abbr.)

27. American Standard Code for Information Interchange. (see 31 down)

29. Planet constantly being "invaded"

30. A companion of DNA and TON (hint: think backwards)

31. Same as 27-down

33. REM statements make your program clearer to others and even to yourself months later - but only if you —— them

34. Often found with FOR and NEXT

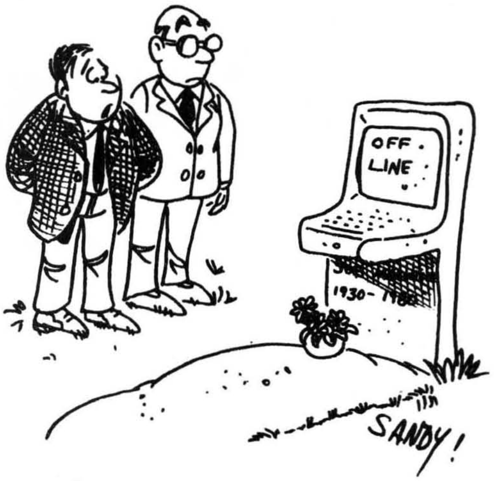
35. There are eightt of these in a bytte (singular)

37. Not the best thing to stick in a disk drive

39. High card

40. Always returns a non-negative number

43. Two thirds of the final program statement



"He was a great programmer."

Mirco Accounting Systems

80
Accounting

With The Best Systems On Today's Market

No other software yet, has provided the simplicity, screen prompting and novice operator error detection. We design totally for you and the TRS-80, with accounting firms and your input.

All of micro-accounting system's flashing cursor programs, edit practically every keyboard entry, to eliminate operator errors. This gives you speed and efficiency of automated bookkeeping, much more simplified with less manual ledger sheets to shuffle through.

M.A.S. GENERAL LEDGER

Our ledger is very simple to use, with up to 200 GL-Accounts, detailed transaction files, trial balance reports, income statement and balance sheet, like a general ledger should. However, out-standing features include: auto loading, batch up to 200 debit & credit entries and auto edit for out-of-balance entries. All transaction programs edit everything, allow 21 character description and 13 for reference, unlimited daily entries. Compare different periods, with comparison reports. Prepair your own depreciation with the depreciation schedual. **$159.00**

M.A.S. ACCOUNTS RECEIVABLES

Our present system will expand up to 1600 customers with auto-billing, mailing labels, and many reports to help improve cash flow. Special features include: Ample room for decriptions (25 char. name, 20 for address, 15 for city and 2 for states). Fast entry of charges and payments, with discounting allowed, complete management report with true ageing, bill-mail code, disc. percent, total charges, pymts, last pymts date and amt., total disc. and remaining balance. Overdue account listings and many more features to assist. Runs by itself or auto posts to our general ledger. Very simple to keep efficient cash flow. Instead of eliminating your receivables, use our programs to help keep them current. **$159.00**

M.A.S. CHECK REGISTER

Complete with reconsilitation form. We have everything needed to keep a efficient checking account. Record checks written by hand, as-well-as writing checks for our accounts payable. Will handle up to 600 checks monthly. Record up to 200 separate checks or total deposits monthly. Print a check register with all checks and deposits, showing ck. #, payee memo, amount, and code. You can correct, checks, deposits, void checks, stop payment and more. Provides information as follows: amt. of total checks, amt. of total deposits, remaining balance, service charges and amt. at beginning of month. Simple to use, with complete error detection. Works stand alone or links to our accounts payable. **$89.00**

M.A.S. ACCOUNTS PAYABLE

Our system will handle up to 300 vendors with checkwriting, mailing labels, and many reports to help improve cash flow Special features include: The same amount of information as accounts receivable, fast entry of charges and payments, with discounting allowed, complete management report with true ageing, overdue account listings and many more features to assist. Runs by-itself or auto posts to our general ledger. Very simple to keep efficient cash flow and receive discounts. **$159.00**

Order The Complete Package For $489.00 And Save $77.00

TOLL-FREE **1-800-258-1790** In NH call 673-5144

```
********** THREE SHORT SAMPLE REPORTS ****** ( BOTTOM RIGHT ) PROGRAM & MAINTENANCE MENU ********
```

```
LIST OVERDUE ACCOUNTS BY ANY NUMBER OF DAYS YOU WISH.  ( THE SAME PROCEDURE FOR INACTIVE ACCOUNTS ).
    =======================================================
    OVERDUE ACCOUNTS AS OF 04/15/80        PAGE 1

    ACCT      NAME                      90 DAYS      AMOUNT
    NO.                                 OVERDUE       OWED
    110  WESTERN LUMBER CO.              856.00     2,240.00
```

MAS 80
SAMPLE
Report

```
( FOR MORE DETAIL, RUN THE ACCOUNTS PAYABLE OR ACCOUNTS RECEIVABLE REPORTS FOR AGED TOTALS AND GRAND TOTAL OF ALL ACCOUNTS. )


    ACCTS PAYABLE  ACTIVITY REPORTS          AS OF  04/15/80       PAGE 1
   *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-* *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
    ID #   NAME              LAST CHRG  LAST PMT  AMT LAST PMT  Y-T-D CHRG  Y-T-D PMT  Y-T-D DISC.  REMAINING
      INV #     INV DATE     < 31 DAYS     31-60 DAYS    61-90 DAYS    > 90 DAYS                  BALANCE
   *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-* *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

    110  WESTERN LUMBER CO.  04/05/80  04/01/80  1000.00    3,240.00    1000.00       00.00     2,240.00

      P.O. 243   04/10/80      384.00      00.00          1000.00     856.00
                               =======     =======        =======     ========
                               384.00      00.00          1000.00     856.00
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

AFTER POSTING ALL TRANSACTION ENTRIES, THEN FINANCIAL REPORTS CAN BE RUN MONTHLY OR MORE OFTEN FOR MANAGEMENT.
```

```
INCOME STATEMENT AS OF 03/31/80                PAGE 1
ACCOUNT NAME           CURRENT BAL  %       Y-T-D BAL  %
==========================================================
                              SALES INCOME
                             **************
INVENTORY SALES         9,170.50  87.3      27,250.35  87.5
ACCESSORIES             1,478.82  13.7       4,786.86  13.5
                        --------  ----      ---------  ----
SALES INCOME           10,659.32  100.0     32,037.21  100.0
                        ========  =====     =========  =====
                              COST OF SALES
                             ***************
INVENTORY PURCHASE      5,170.00  89.1      17,350.25  79.8
ACCESSORIES               989,35  10.9       1,987.23  20.2
                        --------  ----      ---------  ----
TOTAL COST OF SALES     6,159.35  100.0     19,337.48  100.0
                        ========  =====     =========  =====

GROSS PROFIT            4,499.97  28.8      12,699.73  32.3

                             OPERATING EXPENSE
                            *******************
WAGES                   1,400.00  79.9       4,200.00  79.9
RENT                      450.00  20.1       1,350.00  20.1
                        --------  ---        --------  ---
                        1,850.00  100.0      5,550.00  100.0

                          OTHER INCOME & EXPENSES
                         *************************
ENTERTAINMENT            -200.50  100.0      -5,50.00  100.0
                         -------  -----      -------   -----
TOTAL OTHER INCOME       -200.50  100.0      -5,50.00  100.0

NET INCOME ( PROFIT )   2,449.47  22.3       6,599.73  21.5
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%              ACCOUNTS RECEIVABLE MENU
%%          FUNCTION          DESCRIPTION
%%               1..........PROCESS TRANSACTION
%%               2..........POSTING TO GENERAL LEDGER
%%               3..........FILE MAINTENANCE
%%               4..........MANAGEMENT REPORTS
%%               5..........BILLING
%%               6..........MAILLING LIST
%%               7..........OPEN/CLOSED ACCOUNTS
%%               8..........CHANGE SYSTEM DATE
%%               9..........FINISHED
%%
%%                   ENTER FUNCTION- 3
%%
%%
%%
%%
%%              ACCOUNTS RECEIVABLE
%%             MASTER MAINTENANCE MENU
%%           -----------------------------
%%              1.......CHANGE CUSTOMER NAMES
%%              2.......CHANGE CHARGE TRANSACTIONS
%%              3.......CHANGE PAYMENT TRANSACTIONS
%%              4.......LIST CUSTOMER FILE
%%              5.......LIST CHARGE TRANSACTIONS
%%              6.......LIST PAYMENT TRANSACTIONS
%%              7.......COPY DATA DISKS
%%              8.......DIRECTORY OF DATA DISKS
%%              9.......FORMAT DATA DISKS
%%             10.......RETURN TO MAIN MENU
%%
%%                  ENTER FUNCTION- 7
%%
```

DATA BASE REVIEWS

Apple Data Base Programs -What to expect when you buy.

Reviews by Mark Pelczarski and Mike Sullivan

When it comes to selling software, there is no doubt that the real money is in the business market. It stands to reason that almost every major software company would be marketing some kind of data base package, very big sellers in that market with price tags usually in the $100 range. With such a variety of offerings available, however, selecting the one you want can be rather confusing.

The first and best bit of advice we can offer is to narrow your choices down to packages from well known software companies. These companies have a lot at stake with their reputations. If someone buys a bad program from a specific vendor, the tendency is to avoid that vendor in the future. It's a very competitive business with few monopolies; good companies cannot afford bad programs on the market.

Some of the options to look for in a data base are flexibility (can you adapt it to any application, or change your mind after you've begun), print formatting, searching and sorting, and math capabilities. The documentation provided also varies, but most of the major packages provide very complete manuals and are fairly easy to use. In this article we'll take a look at some of the better known packages and capsulize the differences we've found. From that point you should be able to make a decision based on your own needs.

In no particular order, the first package we'll examine is Personal Software's CCA Data Base Manager. Priced at $99.50 it is very typical in its functions - a compliment in that it is very complete for business uses. Included in their manual are examples for inventory control and mail list applications, and it can be used for any application that has data which can be organized in a table structure (almost every data base system uses this structure -some number of headings followed by columns of data). The strongest advantage of the CCA Data Base is its compatibility with Visicalc; files can be exchanged between the two programs. the disadvantages to date have been the inaccuracies and omissions in the documentation provided with the package. Not included is the modification necessary for using the system with a parallel printer -a thoughtless omission that has caused more than a few headaches. Personal Software has promised an update, however, and those people who already own CCA should send in their registration cards and they'll receive the new program disks and manual changes as soon as they are available. The documentation does already include a section of programmer's information that will help the ambitious make their own changes and additions to the package. Overall, despite the shortcomings of the early versions, this is a good choice for the price; especially if you also have Visicalc. CCA Data Base Manager requires at least 32K and Applesoft, and is available from Personal Software or The Software Exchange.

Micro Information System from MUSE at $100 is similar to the CCA Data Base program in options. It does not have the Visicalc compatibility, but it will work with either the Apple II or Apple II plus with 48K. The one feature of this system that is surprising is that the data format is predetermined. You are given six definable headings and five category items. The data items are preset at a maximum length (36 characters) and the category codes may be 2 characters each. Although the package itself is well done, this inflexibility can be a major drawback. MUSE is located at 7112 Darlington Drive, Baltimore, Maryland 21234.

The Information Master, from High Technology, is of the same form, but more flexible like the CCA system. Record length (the combined length of the data items across headings) can be up to 1,980 characters. It is well written and almost impossible to crash -wrong keystrokes, such as RESET, don't harm the program. Printer options are more versatile than with the previously mentioned systems, and the math capabilities include performing calculations as data is entered. Repetitive data may be entered in consecutive records with the right arrow key, a feature that can be quite useful. The disadvantages may be its speed (a 500 record sort takes about 35 minutes), and its price: $150. Its advantages are its ease of operation and options available. Information Master requires 48K, firmware Applesoft, and a disk drive (although 2 are recommended), and is available from High Technology, Inc., 8001 N. Classen Blvd., Oklahoma City, OK 73113.

The Data Factory from Micro Lab has the most impressive packaging I've seen and the programs are again well done. The manual is indexed with tabs for easy reference and it also comes with 2 identical program disks, the first to offer some compromise for a backup master. The packages previously mentioned will replace the master disk at a price if you return the original, but that does not solve the interim problem of an inoperable system. None of the ones yet mentioned have a master that can be copied. The Data Factory offers all the features of a good data base, including math capabilities, versatile print options, and is also the most flexible. You may transfer parts of files to new files, or even add new headings to an existing file. It requires Applesoft and 48K. A drawback is that it is very inefficient with only one disk drive; you'll have to keep swapping disks. Two drives are highly recommended. The Data

continued from previous page

Factory costs $100 and can be purchased from Micro Lab or The Software Exchange.

The rest of the packages reviewed here have the nice option that you can make your own backup copies. It's a shame that pirating is making this feature almost extinct. Hopefully, (I'm a dreamer) consciences of software consumers will allow the manufacturers that still provide that option to continue to do so; otherwise it's going to cost everyone money in the future.

Synergistic Software's Modifiable Data Base, priced at $79.50, is in Consumer Reports' terminology a best buy. The program provided actually creates a program on your own disk that has the data specifications you desire. It has versatile print formatting, allows numeric operations on data, and has fast search and sort routines. Its only drawback may be that it requires a little more technical knowledge for initial set-up. It also has the smallest manual of those listed here; it's complete, but could probably elaborate on some points for the sake of beginners. Still, it is an excellent package for the money. The Modifiable Data Base requires 48K and Applesoft firmware, and is available from Synergistic or TSE.

Perhaps File Cabinet, from Apple, should be the one listed as "best buy"; it's free. Ask your computer dealer for a copy of Contributed Programs Volume 3, and you've got it. You may want to invest in the documentation booklet for Contributed Programs Volumes 3-5 for a mere $2. File Cabinet is fine for small data base applications. All the information resides in RAM; the disk is used only for initial reading of data or for saving files. It does allow a variable number of headings, some print formatting, and even column totals. The sort routine is rather slow, but what can you expect? If your application is small you should try this program before you sink in a lot of money. It will run with Applesoft and any disk system, but the storage capacity is proportional to the memory of your computer.

The most interesting data base program we reviewed is WHATSIT, from Computer Headware. It stands apart from the rest in that it is a free form data base rather than one organized in a table. It is more of a collection area for notes. Each entry consists of a subject, tag, and object, such as Cheryl's Birthday's June 5, Tom's Appointment's Friday, or Chris' Balance's 69.95. You can request all information about any Subject, Tag, or Object by saying something like What's June 5? (WHATSIT would reply with Cheryl's birthday, and anything else it knew about that date.) A wonderfully different approach, it also leaves many things undone. There are no print options; you can dump responses to a printer, but that's all. There is no provision for sorting, no math capability, and some changes of data are awkward. Even with these omissions it can be a very useful program, but the catch is its price: $150. WHATSIT is distributed by Hardhat Software, and requires 48K and Integer BASIC.

BUGS
continued from page 16

I have found an error in DOG STAR ADVENTURE in the May 1979 S-80 SoftSide. This error has not been caught earlier as a result of the complexity of the program and the obscurity of the problem unless the listing has been studied. Two of the messages, variable M7$ and M8$ are not printed at the appropriate time; instead, the note "THAT'S STUPID!" is printed.

This situation may be remedied by deleting line 7300, changing line 7325 to:

```
7325 IFLC 35 AND NO=34 PRINT
M5$GOTO 2125
```

and changing line 7375 to:

```
7375 X=35:GOSUB 21450:IF(Y=1
OR Y  LC) AND NO=35 THEN
PRINT "I DON'T SEE
HER.":GOTO 2125 ELSE IF
NO=35 THEN PRINT M7$: K=1:
FOR I=1 LO: IF OB(I,O)=22
THEN 7525 ELSE NEXT I:
GOTO 2650
```

Otherwise, the program was fantastic.
Thomas Hanlin III

PDI

Preschool I.Q. Builder
Age: Three to six
Skill: Reading readiness
Available for: Apple II Integer (16K)
PET (minimum 8K)
TRS-80, Level II (16K)
Number of programs: Seven to 12, depending on the computer.
Materials: One chrome oxide cassette and Instruction Guide. Decals for key identification are also included.
Description: This series of program teaches vital congnitive skills that children must possess to do well in school. In Part One, *Same and Different,* The child discriminates between two forms — for example, square/circle or A/P.

In Part Two, *Letter Builder,* the child matches a letter on the screen to one on the keyboard. Children can work on their own after a few minutes of instruction.

$14.95

Memory Builder: Concentration
Age: Six and older
Skill: A concentration-like game designed to improve memory and reading skills.
Available for: Apple II Integer (16K)
TRS-80, Level II (16K)
PET (minimum 8K)
Number of programs: Four
Materials: One chrome oxide cassette and Instruction Guide.
Description: *Memory Builder: Concentration* is a game that will improve memory, attention span and mental concentration. The first two games involve matches of single letters. Game three uses 3-letter words and Game four uses 4-letter words. In order to make a match, the student must be able to read the words and remember their locations in the grid.

The player is given three options -- play against the computer; play against himself; or, play against another player. The program is especially good for adults to play with children.

$14.95

Step By Step: How to Program in BASIC
Age: High-school students and older
Skill: Programming in BASIC for the beginner.
Available for: Apple Applesoft (16K)
PET (minimum 8K)
TRS-80, Level II (16K)
Number of Programs: 32
Materials: Three cassettes and 64-page workbook. Materials are packaged in vinyl storage binder.
Description: *Step By Step* is a programming course for beginners. It introduces all the important BASIC commands and programming logic, including simple string logic and one-dimensional arrays. A lesson on graphics is also included. The version for each computer explains the unique features of that computer and how to use them.

The lessons are interactive -- presented in a question and answer format on the computer screen. Each lesson is followed by a quiz and there are two final exam programs at the end of the course. When each lesson is completed, the student practices what he or she has learned and is given sample programs to work with and modify.

$49.95

PDI

Call toll free

1-800-258-1790

(in N.H. call 603-673-5144)

P.O. Box 68 Milford, N.H. 03055

PERSONAL SOFTWARE

CCA DATA MANAGEMENT SYSTEM

- Complete, manual in a three ring binder
- Handles 1 or 2 disk drives
- Written in BASIC, with instructions for writing your own programs to use DMS files, including I/O routines.
- Create, add, update, or delete files easily.
- Search any field for any data.
- Sort by any field.
- Print reports, selecting any fields in any order, and maintain totals and subtotals on any numberic fields.
- Print mailing labels.
- Print records selectively, for example, only customers with New York addresses.
- Complete index in manual
- Scan function to report statistically on file. You might ask how many accounts are overdue and receive a report that started 10 records found in 200 records scanned (5% of file).
- Automatically tests line printer ready state to avoid system lockup.
- Printer alignment pattern for labels.
- Warning issued if you try to compact data without creating a backup.
- On Disk (32K) Applesoft ROM, or TRS-80.
- Works in conjunction with Visicalc. $99.95 plus $2 shipping for the Apple. $74.95 plus $2 shipping for the S-80.

VISICALC

Now you can dispense with pencil, paper and calculator — instead you can have an "electronic sheet" to do your calculations, projections, and planning!

Calculate sales projections, income taxes, personal budget, cost estimates, engineering change — even balance your checkbook. Visicalc is partitioned into a matrix of rows and columns. At each position in this matrix you can define a title, formula, or number. By writing on your "electronic sheet" you can set up individualized charts, tables, and records. For example, you can design your own sales forecasting format to assist in making the important "What if?" business decisions.

What would happen if sales increased by 10%?

What if delivery time were decreased by 2 weeks?

What would be the result if I produced 500 widgets this month instead of 600? What if I produced 700?

Let **Visicalc** assist you in performing these calculations and save you countless hours.

Visicalc requires 32K, Applesoft and disk drive. $150.00

TRS-80 and Atari versions available soon!

TO ORDER TOLL FREE

1-800-258-1790

(In NH call 673-5144)

ATARI®

PROGRAMS FOR YOUR COMPUTER

KINGDOM (CX4102)

This educational program cassette simulates the running of a small agricultural kingdom. Catastrophe may befall your people, your food supply or your crops. You determine the farming strategy and if you are successful you are allowed to reign another term. Teaches and develops mathematical skills. $14.95

BASKETBALL (CXL4404)

You can dribble, shoot, pass, block shots, and even make steals. Five different game variations allow one, two, three or four players to compete. Play against the computer, your family or friends. This game uses up to four Joystick Controllers. Cartridge. $39.95

BLACKJACK (CX4105)

Play one of the most popular casino games at home with your ATARI Personal Computer System. The computer is the dealer and it plays by the rules. You, on the other hand, bet money and decide whether to stand, draw another card or double down in your efforts to hit 21. One cassette. $14.95

COMPUTER CHESS CXL4009)

You can play chess against the computer. Eight levels of play are available. Moves are made using a Joystick Controller. The chess board and pieces are shown on the screen. Both "castling" and "en passant" moves can be made by you or the computer. The board can be set up for a particular chess problem or situation. The computer can play either white or black. Uses one Joystick Controller. Cartridge. $39.95

STAR RAIDERS (CXL4011)

A state of war exists between the Atarian Federation and the Zylon Empire. Your mission: Destroy all Zylon star ships. This strategy and action cartridge features four mission skill levels and rates a player at the end of each session. Dramatic sounds and screen displays make this the most exciting space game available today. Your battleground is the entire galaxy. Meteors fly through space; Zylon fleets surround and destroy friendly star bases, and docking at a starbase enables you to receive a new supply of energy and repair damaged equipment. For one player; uses a Joystick Controller. $59.95

SUPER BREAKOUT (CXL4006)

This cartridge is a sophisticated version of the popular electronic video game BREAKOUT™, in which you smash through layers of bricks and knock out a wall. Four different games are available, up to eight players can play, and a rating is displayed at the end of each game. You can optionally suspend play or request up to five additional serves. This cartridge uses up to four Paddle Controllers. $39.95

HANGMAN (CX4108)

This educational program cassette has 3 levels of play — Beginner, Intermediate, and Expert. You play against the computer by guessing the word the computer has selected. If you do not guess the word after six tries the computer will hang your man. You may use the computer keyboard or an optional Joystick Controller to guess the letter. Suitable for ages 8 to adult. $14.95

3-D TIC-TAC-TOE (CXL4010)

Now you can play Tic-Tac-Toe from a three-dimensional perspective. This game cartridge also contains Bottoms-Up, a variation of 3-D Tic-Tac-Toe. Match wits with the computer or compete against another player in either version. One-player games feature eight difficulty levels. Uses Joystick Controllers. $39.95

VIDEO EASEL (CXL4005)

You can create a design or pattern on the screen using the keyboard or up to four Joystick Controllers, then watch as the computer expands on the original pattern indefinitely. Many variations allow you to create imaginative and colorful graphic designs. Options include Drawing, Quad Drawing, Life (which uses the mathematic principles of John Conway's game of "LIFE"), resetting the color and six preprogrammed Painting designs. Uses Joystick Controllers. Cartridge. $39.95

AN INVITATION TO PROGRAMMING 1, 2, and 3

These lessons are designed to teach the fundamentals of programming in ATARI BASIC. A recorded voice asks you questions, waits for you to respond, and indicates whether you are correct or incorrect. These self-teaching courses will teach you to use BASIC programming statements as well as the color, graphics, sound and music capabilities of the ATARI Personal Computers. Each lesson requires 8K of RAM. $19.95

MUSIC COMPOSER™ (CXL4007)

Use this cartridge and the computer keyboard to compose songs, recreate old tunes, or experiment. When you program musical notes through the keyboard, you hear the melody as it is simultaneously displayed on the screen. You can change any note or the tempo. And you can arrange the music to play different sections in the order you choose. Once you have entered music, you can save it on the ATARI 410 Program Recorder or the ATARI 810 Disk Drive. You can also write BASIC programs that can convert a melody into melody plus harmony. $59.95

BIORHYTHM (CX4107)

Use this program cassette to calculate physical, emotional and intellectual highs, lows, and critical days. By giving the computer your birthdate you can plot a biorhythm chart on your television screen or optional ATARI Printer. $14.95

TO ORDER TOLL FREE

1-800-258-1790

(In NH call 673-5144)

The Microsoft Z-80 SoftCard. New Power for Your Apple II!

Plug the new Microsoft Z-80 SoftCard into your Apple IITM and start using all of the system and application software written for Z-80 based computers. Software that you could never use before on your Apple II.

The SoftCard actually contains a Z-80 processor and lets you switch between the Apple's 6502 and the Z-80 with simple commands, so you can use software written for either processor.

Starting with Two Software Standards. Versatile CP/M, the most widely used microcomputer operating system ever, is included on diskette in the SoftCard package, ready to run on your Apple II.

You get Microsoft's 5.0 BASIC, too, the most powerful version to-date of our famous BASIC interpreter.

PRINT USING, 16-digit precision, CALL, and CHAIN and COMMON are just some of the major BASIC features you'll add. Applesoft's graphics extensions are still included.

More Power Down the Line. You can get even more programming power and versatility by adding Microsoft's FORTRAN, COBOL, BASIC Compiler and Assembly Language Development System. All are available separately to run with the SoftCard system.

And the whole host of CP/M-based business, scientific and educational applications can be easily transferred to your Apple with SoftCard.

The Microsoft Z-80 SoftCard is compatible with most every Apple product from the Apple II to the Apple II Plus, Language Card and peripherals. Independent peripherals for the Apple are supported as well. The SoftCard package requires a system with 48K and a disk drive.

$349 for SoftCard, CP/M, Microsoft BASIC and complete documentation.

To Order TOLL FREE
1-800-258-1790
(In NH call 673-5144)

TM Apple II is a trademark of Apple Computer, Inc.
CP/M is a registered trademark of Digital Research.

MICROSOFT
CONSUMER PRODUCTS

3

DEVELOPING DATA BASE

continued from page 51

revisions in the future that will allow selective printing and formatting.

The last subroutine for this month is the add routine in lines 4000-4100. One is added to the number of items (note that NI started at -1, one less than the number of items as mentioned earlier), then the user is prompted with each heading and asked for the new data. The information is simply put at the end of the file after any already existing data. This means a sort subroutine will be handy in the future.

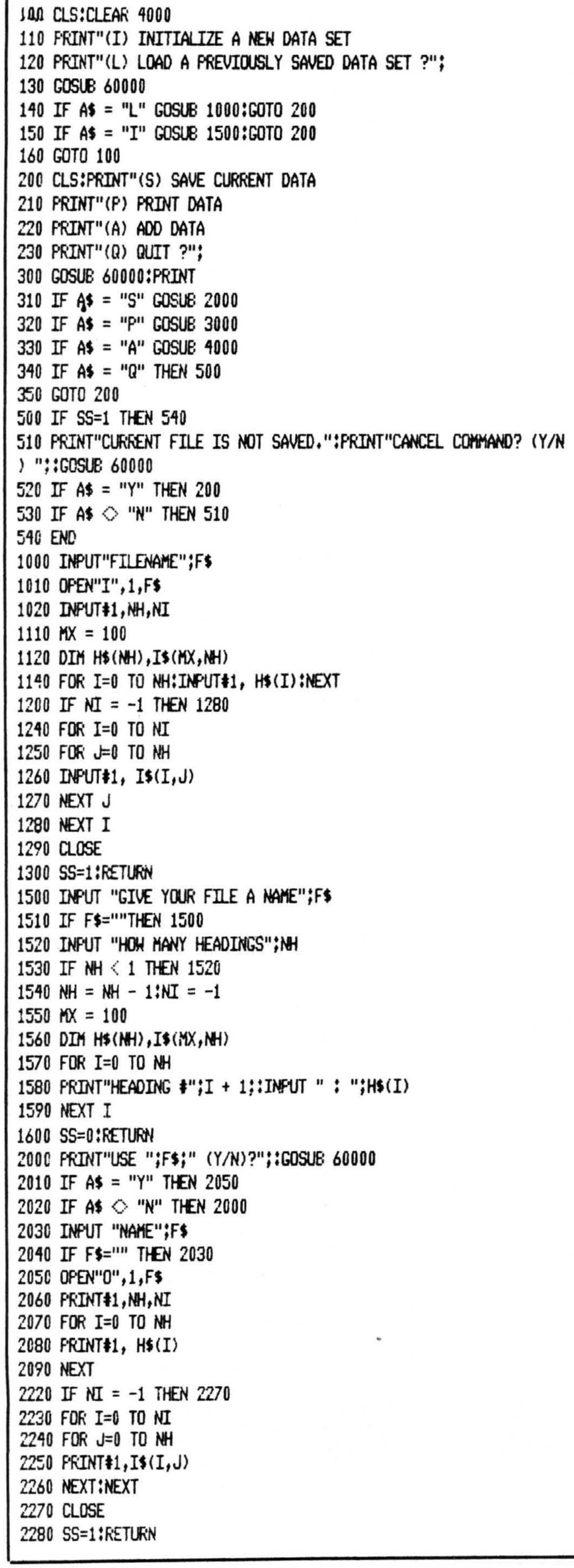
At line 9000 is a short subroutine that gives an error message if you try to print a file with no data (see line 3000). This little routine will also be used in the future in conjuction with other subroutines.

Following are listings of the program for the Apple with disk, for the S-80 with disk, modifications for an S-80 cassette version, and for the Atari with cassette and modifications for a disk version. Each month we'll print add-on or replacement modules that you can use with this basic program. In next month's issue we'll do the delete and alteration subroutines so you can correct any filing mistakes you make this month. This program is also meant to be interactive in more ways than one; we'd like to hear from you if you've got ideas or suggestions concerning it. We've got plenty of ideas of our own, but we'd like to create one of the best packages around and welcome the input.

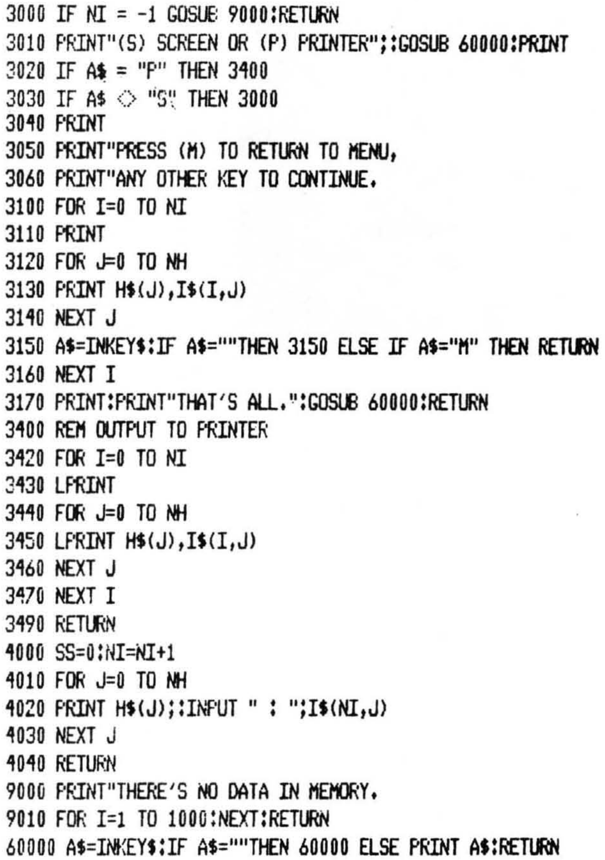
APPLESOFT VERSION

```
10  REM  SOFTSIDE'S DEVELOPING
               DATA BASE
20  REM  MARK PELCZARSKI, 1980
100 D$ = "": REM   CONTROL-D
110  HOME : PRINT "(I) INITIALIZE
      A NEW DATA SET"
120  PRINT "(L) LOAD A PREVIOUSLY
      SAVED DATA SET ?";
130  GET A$: PRINT A$
140  IF A$ = "L" THEN  GOSUB 1000
     : GOTO 200
150  IF A$ = "I" THEN  GOSUB 1500
     : GOTO 200
160  GOTO 130
200  HOME : PRINT "(S) SAVE CURRE
     NT DATA"
220  PRINT "(P) PRINT DATA"
230  PRINT "(A) ADD DATA"
290  PRINT "(Q) QUIT   ?";
300  GET A$: PRINT A$: PRINT
320  IF A$ = "S" THEN  GOSUB 2000

330  IF A$ = "P" THEN  GOSUB 3000

340  IF A$ = "A" THEN  GOSUB 4000

400  IF A$ = "Q" THEN 500
410  GOTO 200
500  IF SS = 1 THEN 540
510  PRINT "CURRENT FILE IS NOT S
     AVED.": PRINT "CANCEL COMMAN
     D? (Y/N) ";: GET A$
520  IF A$ = "Y" THEN 200
530  IF A$ < > "N" THEN 510
540  END
1000  INPUT "FILE NAME? ";F$
1020  PRINT D$;"OPEN";F$
1030  PRINT D$;"READ";F$
1040  INPUT NH: INPUT NI
1110 MX = 100
1130  DIM H$(NH),I$(MX,NH)
1140  FOR I = 0 TO NH: INPUT H$(I
      ): NEXT
1200  IF NI =  - 1 THEN 1280
1240  FOR I = 0°TO NI
1250  FOR J = 0 TO NH
1260  INPUT I$(I,J)
1270  NEXT J: NEXT I
1280  PRINT D$;"CLOSE";F$
1300 SS = 1: RETURN
1500  INPUT "GIVE YOUR FILE A NAM
      E : ";F$
1510  IF F$ = "" THEN 1500
1520  INPUT "HOW MANY HEADINGS? "
      ;NH
1530  IF NH < 1 THEN 1520
1540 NH = NH - 1:NI =  - 1
1550 MX = 100
1560  DIM H$(NH),I$(MX,NH)
1570  FOR I = 0 TO NH
1580  PRINT "HEADING #";I + 1;: INPUT
      " : ";H$(I)
1590  NEXT I
1600 SS = 0: RETURN
2000  PRINT "USE ";F$;" (Y/N)?";:
      GET A$: PRINT A$
2050  IF A$ = "Y" THEN 2100
2060  IF A$ < > "N" THEN 2000
2070  INPUT "NAME? ";F$
2080  IF F$ = "" THEN 2070
2100  PRINT D$;"OPEN";F$
2110  PRINT D$;"WRITE";F$
2120  PRINT NH: PRINT NI
2130  FOR I = 0 TO NH
2140  PRINT H$(I)
2150  NEXT
2220  IF NI =  - 1 THEN 2270
2230  FOR I = 0 TO NI
2240  FOR J = 0 TO NH
2250  PRINT I$(I,J)
2260  NEXT J: NEXT I
2270  PRINT D$;"CLOSE";F$
2280 SS = 1: RETURN
3000  IF NI =  - 1 THEN  GOSUB 90
     00: RETURN
3010  PRINT "(S) SCREEN OR (P) PR
     INTER";: GET A$: PRINT
3020  IF A$ = "P" THEN 3400
3030  IF A$ < > "S" THEN 3000
3040  PRINT
3050  PRINT "PRESS <ESC> TO RETUR
     N TO MENU,"
3060  PRINT "ANY OTHER KEY TO CON
     TINUE."
3100  FOR I = 0 TO NI
3110  PRINT
3120  FOR J = 0 TO NH
3130  PRINT H$(J),I$(I,J)
3140  NEXT J
3150  GET A$: IF A$ =  CHR$ (27) THEN
      RETURN
3160  NEXT I
3170  PRINT : PRINT "THAT'S ALL."
     : GET A$
3180  RETURN
3390  REM   PRINTER IS IN SLOT #1

3400  PRINT D$;"PR#1"
3420  FOR I = 0 TO NI
3430  PRINT
3440  FOR J = 0 TO NH
3450  PRINT H$(J),I$(I,J)
3460  NEXT J
3470  NEXT I
3480  PRINT D$;"PR#0"
3490  RETURN
4000 SS = 0:NI = NI + 1
4010  FOR J = 0 TO NH
4020  PRINT H$(J);: INPUT " : ";I
      $(NI,J)
4030  NEXT J
4040  RETURN
9000  PRINT "THERE'S NO DATA IN M
      EMORY."
9010  FOR I = 1 TO 1000: NEXT : RETURN
```

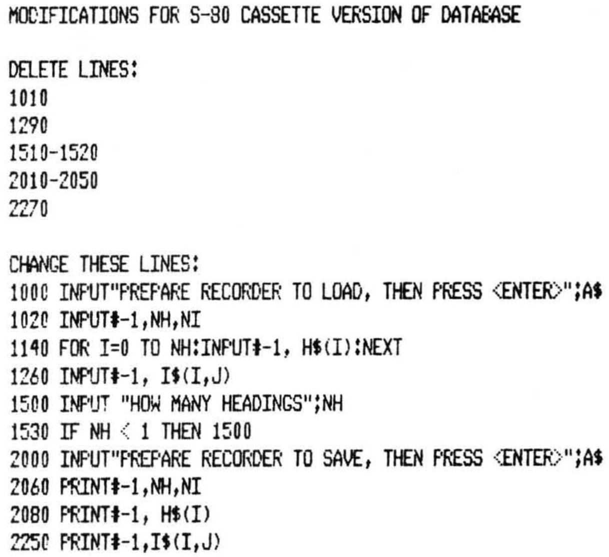
S-80 VERSION



```
100 CLS:CLEAR 4000
110 PRINT"(I) INITIALIZE A NEW DATA SET
120 PRINT"(L) LOAD A PREVIOUSLY SAVED DATA SET ?";
130 GOSUB 60000
140 IF A$ = "L" GOSUB 1000:GOTO 200
150 IF A$ = "I" GOSUB 1500:GOTO 200
160 GOTO 100
200 CLS:PRINT"(S) SAVE CURRENT DATA
210 PRINT"(P) PRINT DATA
220 PRINT"(A) ADD DATA
230 PRINT"(Q) QUIT ?";
300 GOSUB 60000:PRINT
310 IF A$ = "S" GOSUB 2000
320 IF A$ = "P" GOSUB 3000
330 IF A$ = "A" GOSUB 4000
340 IF A$ = "Q" THEN 500
350 GOTO 200
500 IF SS=1 THEN 540
510 PRINT"CURRENT FILE IS NOT SAVED.":PRINT"CANCEL COMMAND? (Y/N
) ";:GOSUB 60000
520 IF A$ = "Y" THEN 200
530 IF A$ <> "N" THEN 510
540 END
1000 INPUT"FILENAME";F$
1010 OPEN"I",1,F$
1020 INPUT#1,NH,NI
1110 MX = 100
1120 DIM H$(NH),I$(MX,NH)
1140 FOR I=0 TO NH:INPUT#1, H$(I):NEXT
1200 IF NI = -1 THEN 1280
1240 FOR I=0 TO NI
1250 FOR J=0 TO NH
1260 INPUT#1, I$(I,J)
1270 NEXT J
1280 NEXT I
1290 CLOSE
1300 SS=1:RETURN
1500 INPUT "GIVE YOUR FILE A NAME";F$
1510 IF F$=""THEN 1500
1520 INPUT "HOW MANY HEADINGS";NH
1530 IF NH < 1 THEN 1520
1540 NH = NH - 1:NI = -1
1550 MX = 100
1560 DIM H$(NH),I$(MX,NH)
1570 FOR I=0 TO NH
1580 PRINT"HEADING #";I + 1;:INPUT " : ";H$(I)
1590 NEXT I
1600 SS=0:RETURN
2000 PRINT"USE ";F$;" (Y/N)?";:GOSUB 60000
2010 IF A$ = "Y" THEN 2050
2020 IF A$ <> "N" THEN 2000
2030 INPUT "NAME";F$
2040 IF F$="" THEN 2030
2050 OPEN"O",1,F$
2060 PRINT#1,NH,NI
2070 FOR I=0 TO NH
2080 PRINT#1, H$(I)
2090 NEXT
2220 IF NI = -1 THEN 2270
2230 FOR I=0 TO NI
2240 FOR J=0 TO NH
2250 PRINT#1,I$(I,J)
2260 NEXT:NEXT
2270 CLOSE
2280 SS=1:RETURN
```

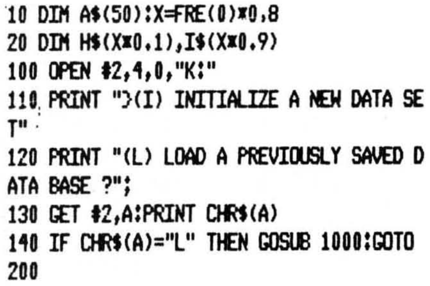


```
3000 IF NI = -1 GOSUB 9000:RETURN
3010 PRINT"(S) SCREEN OR (P) PRINTER";:GOSUB 60000:PRINT
3020 IF A$ = "P" THEN 3400
3030 IF A$ <> "S" THEN 3000
3040 PRINT
3050 PRINT"PRESS (M) TO RETURN TO MENU,
3060 PRINT"ANY OTHER KEY TO CONTINUE.
3100 FOR I=0 TO NI
3110 PRINT
3120 FOR J=0 TO NH
3130 PRINT H$(J),I$(I,J)
3140 NEXT J
3150 A$=INKEY$:IF A$=""THEN 3150 ELSE IF A$="M" THEN RETURN
3160 NEXT I
3170 PRINT:PRINT"THAT'S ALL.":GOSUB 60000:RETURN
3400 REM OUTPUT TO PRINTER
3420 FOR I=0 TO NI
3430 LPRINT
3440 FOR J=0 TO NH
3450 LPRINT H$(J),I$(I,J)
3460 NEXT J
3470 NEXT I
3490 RETURN
4000 SS=0:NI=NI+1
4010 FOR J=0 TO NH
4020 PRINT H$(J);:INPUT " : ";I$(NI,J)
4030 NEXT J
4040 RETURN
9000 PRINT"THERE'S NO DATA IN MEMORY.
9010 FOR I=1 TO 1000:NEXT:RETURN
60000 A$=INKEY$:IF A$=""THEN 60000 ELSE PRINT A$:RETURN
```



```
MODIFICATIONS FOR S-80 CASSETTE VERSION OF DATABASE

DELETE LINES:
1010
1290
1510-1520
2010-2050
2270

CHANGE THESE LINES:
1000 INPUT"PREPARE RECORDER TO LOAD, THEN PRESS <ENTER>";A$
1020 INPUT#-1,NH,NI
1140 FOR I=0 TO NH:INPUT#-1, H$(I):NEXT
1260 INPUT#-1, I$(I,J)
1500 INPUT "HOW MANY HEADINGS";NH
1530 IF NH < 1 THEN 1500
2000 INPUT"PREPARE RECORDER TO SAVE, THEN PRESS <ENTER>";A$
2060 PRINT#-1,NH,NI
2080 PRINT#-1, H$(I)
2250 PRINT#-1,I$(I,J)
```

ATARI VERSION



```
10 DIM A$(50):X=FRE(0)*0,8
20 DIM H$(X*0.1),I$(X*0.9)
100 OPEN #2,4,0,"K:"
110 PRINT "}(I) INITIALIZE A NEW DATA SE
T"
120 PRINT "(L) LOAD A PREVIOUSLY SAVED D
ATA BASE ?";
130 GET #2,A:PRINT CHR$(A)
140 IF CHR$(A)="L" THEN GOSUB 1000:GOTO
200
```

5

continued from previous page

```
150 IF CHR$(A)="I" THEN GOSUB 1500:GOTO
200
160 GOTO 110
200 PRINT "}(S) SAVE CURRENT DATA"
220 PRINT "(P) PRINT DATA"
230 PRINT "(A) ADD DATA"
290 PRINT "(Q) QUIT   ?";
300 GET #2,A:PRINT CHR$(A)
320 IF CHR$(A)="S" THEN GOSUB 2000
330 IF CHR$(A)="P" THEN GOSUB 3000
340 IF CHR$(A)="A" THEN GOSUB 4000
400 IF CHR$(A)="Q" THEN 500
410 GOTO 200
500 IF SS=1 THEN 540
510 PRINT "CURRENT FILE IS NOT SAVED.":P
RINT "CANCEL COMMAND? (Y/N) ";:GET #2,A:
PRINT CHR$(A)
520 IF CHR$(A)="Y" THEN 200
530 IF CHR$(A)<>"N" THEN 510
540 END
1000 PRINT :PRINT "HIT RETURN WHEN READY
 TO READ";:INPUT A$
1020 POKE 764,12:OPEN #1,4,0,"C:"
1040 INPUT #1,NH
1041 INPUT #1,NI
1042 INPUT #1,HL
1043 INPUT #1,IL
1110 MX=1
1140 INPUT #1,H$
1200 IF NI=-1 THEN 1280
1260 INPUT #1,I$
1280 CLOSE #1
1300 SS=1:RETURN
1500 PRINT :PRINT "HOW MANY HEADINGS";:I
NPUT NH
1530 IF NH<1 THEN 1500
1532 PRINT "ENTER MAXIMUM HEADING LENGTH
";:INPUT HL
1534 IF HL<1 THEN 1532
1540 NH=NH-1:NI=-1
1550 MX=100
1560 H$="":I$=""
1570 FOR I=0 TO NH
1580 PRINT "HEADING #";I+1;" :";:INPUT A
$
1581 IF LEN(A$)>HL THEN PRINT "MAXIMUM L
ENGTH IS ";HL;". REENTER":GOTO 1580
1583 IF LEN(A$)<HL THEN A$(LEN(A$)+1)="
":GOTO 1583
1585 H$(LEN(H$)+1)=A$
1590 NEXT I
1600 SS=0:RETURN
2000 PRINT :PRINT "HIT RETURN WHEN READY
 TO SAVE";:INPUT A$
2100 POKE 764,12:OPEN #1,8,0,"C:"
2120 PRINT #1;NH
2121 PRINT #1;NI
2122 PRINT #1;HL
2123 PRINT #1;IL
2130 PRINT #1;H$
2220 IF NI=-1 THEN GOSUB 9000:RETURN
2250 PRINT #1;I$
2270 CLOSE #1
2280 SS=1:RETURN
3000 IF NI=-1 THEN GOSUB 9000:RETURN
3010 PRINT :PRINT "(S) SCREEN OR (P) PRI
NTER ?":PRINT
3015 GET #2,A
3020 IF CHR$(A)="P" THEN 3400
3030 IF CHR$(A)<>"S" THEN 3015
3040 PRINT "PRESS <ESC> TO RETURN TO MEN
U,"
3050 PRINT "ANY OTHER KEY TO CONTINUE."
3060 RL=(NH+1)*IL
3100 FOR I=0 TO NI
3110 PRINT
3120 FOR J=0 TO NH
3130 PRINT H$(J*HL+1,J*HL+HL),I$(I*RL+1+
J*IL,I*RL+J*IL+IL)
3140 NEXT J
3150 GET #2,A
3155 IF A=27 THEN RETURN
3160 NEXT I
3170 PRINT :PRINT "THAT'S ALL."
3175 GET #2,A
3180 RETURN
3400 RL=(NH+1)*IL
3410 RL=(NH+1)*IL
3420 FOR I=0 TO NI
3430 LPRINT " "
3440 FOR J=0 TO NH
3450 LPRINT H$(J*HL+1,J*HL+HL),I$(I*RL+1
+J*IL,I*RL+J*IL+IL)
3460 NEXT J
3470 NEXT I
3490 RETURN
4000 SS=0:NI=NI+1
4005 IF IL=0 THEN PRINT :PRINT "ENTER MA
XIMUM ITEM LENGTH";:INPUT IL
4008 IF IL<1 THEN IL=0:GOTO 4005
4010 PRINT :FOR J=0 TO NH
4020 PRINT H$(J*HL+1,J*HL+HL);" : ";:INP
UT A$
4022 IF LEN(A$)>IL THEN PRINT "TOO LONG.
 MAXIMUM SIZE IS ";IL;". REENTER":GOTO 4
020
4025 IF LEN(A$)<IL THEN A$(LEN(A$)+1)="
":GOTO 4025
4030 I$(LEN(I$)+1)=A$:NEXT J
4040 RETURN
9000 PRINT "THERE'S NO DATA IN MEMORY."
9010 GET #2,A:RETURN
```

DISK CHANGES TO ATARI DATABASE

```
30 DIM F$(20)
1000 PRINT :PRINT "ENTER INPUT FILESPEC"
;:INPUT F$
1020 A$="D:":A$(LEN(A$)+1)=F$:OPEN #1,4,
0,A$
2000 PRINT :PRINT "ENTER OUTPUT FILESPEC
";:INPUT F$
2100 A$="D:":A$(LEN(A$)+1)=F$:OPEN #1,8,
0,A$
```

EDITORIAL
continued from page 4

great potential, and learning about computers is worthwhile in itself. Beyond that, they can also be a lot of fun. With such a variety of software available everyone can find something they like. And the capabilities will keep growing, just like in all the science fiction you see. I used to find it amazing how fiction writers could prophesize so well about the future until I realized that they provide the ideas in the first place; the scientists and technicians create things in the vision of the future with which they've been provided.

TRS-80 PROGRAMMING HINT

While recently making a game program, I became somewhat annoyed at the rather ''slow'' rate at which my graphics filled the video. Referring to my ''Level II BASIC Reference Manual'' (page 8/6) I found a small program which illustrated the speed at which the video could be whited out by POKING the value of 191 into the video RAM. It was apparent that their idea could be applied to my program, but I still found this method too slow.

After experimentation I found a very simple routine illustrated in the ''Level II BASIC Reference Manual''. I have applied my basic concept to several of my programs, and have found it especially useful in repeated graphic displays. It requires some modification to the users specific needs, but regardless of this, the concept remains the same. Below is the routine:

```
10 CLEAR 64

20 FOR X = 1 TO 16: PRINT
STRING$(64,CHR$(191));:NEXT
```

As you can see, the concept is very simple. Yet I cannot recall seeing this method used to date. I hope that you might find this routine of some use, as it was helpful to me.

David W. Morris
Atlanta, Georgia

The Software Exchange
Market Basket

SOFTWARE FOR THE TRS-80

TO ORDER TOLL FREE
1-800-258-1790
(In NH call 673-5144)

HOURS:
MONDAY - FRIDAY
9 AM to 9 PM
SATURDAY
11 AM to 3 PM

*EASTERN TIME

✶ GAMES

TRS-80 Opera Theatre
Magnificent sound! by Richard Taylor
Level II, 16K $9.95

Challenge
Word game with sound by Richard Taylor
Level II, 16K $9.95

Bridge Challenger
by Personal Software
Level II, 16K $14.95

Invasion
by Chris Freund
Level II, 16K $9.95
Disk $14.95

X-Wing Fighter II
by Chris Freund
Level II, 16K $9.95

Taipan
by Art Canfil
Level II, 16K $9.95

Sargon Chess
by Dan and Kathe Spracklen
Level II, 16K $19.95

Sargon II
by Dan and Kathe Spracklen
Level II, 16K $29.95
on disk $34.95

Amazin' Mazes
by Robert Wallace
Level II, 16K $9.95

Snake Eggs
With sound by Leo Christopherson
Level II, 16K $14.95

Life Two
With sound by Leo Christopherson
Level II, 16K $14.95

Android Nim
With sound by Leo Christopherson
Level II, 16K $14.95

Bee Wary
With sound by Leo Christopherson
Level II, 16K $14.95

Tycoon
by David Bohlke
Level II, 16K $7.95

9 Games for Preschool Children
by George Blank
Level II, 16K $9.95

End Zone II
by Roger W. Robitaille
Level II, 16K $9.95

Space Battles
by Level IV
Level II, 16K tape $14.95
32K disk $19.95

Star Trek
by Lance Micklus
Level II, 16K $14.95

Adventures on Tape
by Scott Adams
Level II, 16K $14.95
Choose one:
- Adventureland: Magical beings, perils and puzzles!
- Pirate's Cove: Clue in a bloodsoaked book.
- Mission Impossible: Foil the saboteur, save the reactor!
- The Count: Protect your neck! Who do you think lives here?
- Voodoo Castle: Remove the curse from Count Cristo.
- Strange Odyssey: The ruins of an ancient alien culture?
- Fun House: Takes all your brains to get past the gate!
- Pyramid of Doom: Watch out for the Nomad!
- Ghost Town

Adventures on Disk
by Scott Adams
$39.95 for three-game combinations:
1) Mission Impossible, Pirate's Cove, Adventureland
2) Voodoo Castle, The Count, Strange Odyssey
$24.95 for two-game combinations:
3) Fun House, Pyramid of Doom

Adventure Sampler
by Scott Adams
Mini-version of Adventureland, serves as introduction
Level II, 16K $5.95

Pork Barrel
by George Blank
Level II, 16K $9.95

Barricade
by Small Systems Software
Machine Language $9.95

Pentominoes
by James Garon
Level II, 16K $7.95

Mean Checkers Machine
by Lance Micklus
Level II, 16K tape $9.95
Disk $14.95

Kriegspiel II $14.95

Time Trek
by Joshua Lavinsky
Level II, 16K $14.95

Kamikaze
by Russell Starkey
Level II, 16K $7.95

Fastgammon
from Quality Software
Level II, 16K $19.95

Cribbage
by Roger W. Robitaille Sr.
Level II, 16K $7.95

Galactic Empire
by Doug Carlston
Level II, 16K $14.95

Galactic Trader (a sequel)
by Doug Carlston
Level II, 16K $14.95

Galactic Revolution
by Doug Carlston
Level II, 16K $14.95

Galactic Trader and Empire
on disk 32K $29.95

All 3 on disk $39.95

Airmail Pilot $7.95

Air Traffic Controller $9.95

Air Raid
by Small Systems Software
Level I or II, 4K $9.95

Lost Dutchman's Gold
by Teri Li
Level II, 16K $9.95

Original Adventure
by Microsoft as played on PDP-10
32K disk $29.95

Pigskin (Football)
by Acorn Software
Level II, 16K $9.95

PR Dogfight
by David Bohlke
Level II, 16K $7.95

Journey to the Center of the Earth
by Greg Hassett
Level II, 16K $7.95

See SoftSide on cassettes for additional games

Datestones of Ryn
cassette $14.95
disk $19.95

Starfleet Orion
cassette $19.95
disk $24.95

Invasion of Orion
cassette $19.95
disk $24.95

Morloc's Tower
cassette $14.95
disk $19.95

Rescue at Rigel
cassette $19.95
disk $24.95

Temple of Apshai
cassette $24.95
disk $29.95

Interlude
cassette $14.95
disk $17.95

Bi-Bomber $15.00

Nukewar $15.00

Midway Campaign
.... $15.00

North Atlantic Convoy Raiders $15.00

Planet Miners
.... $15.00

Pinball
cassette $14.95
disk $20.95

Amateur Astronomy Handbook
cassette $14.95

Computer Bismark
cassette $49.95

✶ BUSINESS

Inventory System II.3
by M. Kelleher
Improved version $79.95

Inventory 'S'
by Roger W. Robitaille, Sr.
Level II, 16K tape $24.95
32K disk/without invoicing $39.95
with invoicing $59.95

Payroll
by Stephen Hebbler
Disk 32K $39.95

Mail List II
by SBSG
32K disk $49.95

Small Business Bookkeeping
by R.W. Robitaille, Sr.
Level II, 16K with journal
disk $36.95
tape $31.95
without journal
disk $29.95
tape $24.95

Accounts Receivable(with invoicing)
by Stephen Hebbler
3 disks, 2 manuals; 32K, 2 drives, line printer $69.95

CCA Data Management System
from Personal Software
Ready for transfer to disk, with manual
Tape .... $74.95 + $2

✶ SPECIAL PURPOSE

Histograph/Scattergram
by Gary Breschini
Level II, 16K $9.95

Tiny Comp A BASIC Compiler in BASIC
by David Bohlke
Tape $19.95
Disk $24.95

RPN Calculator
by Russell Starkey
Level II, 16K $9.95

Ham Radio
by Michael Kelleher
Level II, 16K $9.95

Ham Radio
Advanced version for 32K
disk $24.95

Electronic Assistant
by John Adamson
Level II, 16K $9.95

Basic Statistics
by Steve Reisser
Level II, 16K $9.95

Magic Paper Calculator
by Russell Starkey
Level II, 16K $14.95

Keyboard 80 (Improved)
by John Adamson
Level II, 16K ... $9.95

Scripsit ... $95+$2

Dynamic Data Base
by Ken Knecht
Manual and 3 programs on disk for 32K or larger systems with one or more drives ... $39.95

✱ PERSONAL

Typing Tutor
by 80 US
Level II, 16K ... $19.95

I Ching/Biorhythms
together ... $9.95

Secrets of the Tarot
Cards of Fortune
together ... $9.95

Personal Finance
by Lance Micklus
Level II, 16K ... $9.95

Advanced Personal Finance
by Lance Micklus
32K disk ... $24.95

Home Financial Management
by Michael Kelleher
Level II, 16K ... $9.95

Periodical Cross Reference
by Dave Stambaugh
Level II, 16K ... $14.95
32K disk ... $19.95

Roots
by Bill Sholar
32K disk ... $19.95

✱ UTILITIES

T-Short
by Web Associates
Level II, 16K ... $9.95

Z80 Zap Command
disk ... $29.95

IRV
by Robin Mork
Level II, 16,32, and 48K
cassette ... $24.95
disk ... $29.95

APL-80 3.0
by Phelps Gates
Deluxe version—32K disk ... $39.95
Level II, 16K tape—reduced features, no course or book ... $14.95
Book (separately) ... $16.95 + $3

File Manager 80
by Nepenthe
32K disk ... $49.95

Automated Disk Directory
by George Blank
32K disk, requires NEWDOS ... $14.95

STAD
Trace and Debug Monitor for tape and disk systems 16, 32 and 48K on one tape ... $24.95

NEWDOS by Apparat ... $49.95

NEWDOS+ by Apparat ... $99.95

NEWDOS 80 by Apparat ... $149.00

RSM2 by Small Systems Software
Level II, 16K ... $26.95

RSM2D by Small Systems Software
Disk for 16-48K on one tape ... $29.95

KVP by Lance Micklus
Tape # 232 ... $14.95
Disk ... $19.95

ST80 Smart Terminal
by Lance Micklus
Level II, 16K ... $49.95

ST80D Smarter Terminal
by Lance Micklus
For Disk ... $79.95

ST80-UC dedicated to THE SOURCE
by Lance Micklus
Level II, 16K ... $24.95

ST80 III
by Lance Micklus ... $150.00

Micro Text Editor
by Don Coons
Level II, 4K or 16K ... $9.95

Level III BASIC
by Microsoft ... $49.95

Level I in Level II
by Apparat
Level II, 16K ... $15.00

FORTRAN
by Microsoft
32K, 2 disks ... $80.00 + $2.50

Assembler
by Microsoft
32K ... $80.00 + $2.50

FORTRAN Plus Assembler ... $150.00 + $5

Spool
Will print an ASCII file to a parallel line printer at the same time you are using your computer for another program. For 32K disk systems only. Will NOT work with NEWDOS (2.3 or VTOS 3.0 OK).
Disk ... $24.95

Cross Reference Utilities
by Lance Micklus
Level II cassette versions
RENUMX (XREF with RENUMBER and rescue after reset) ... $14.95
RENUMBER separately ... $7.95

RX—Cross Reference for Disk
by Lance Micklus
32K, 1 disk ... $24.95

RENUMBER
by Lance Micklus
The full-featured RENUMBER program. One version now works in 16K, 32K, or 48K ... $7.95

System Copy
by Kalman Bergen
Level II, 16K ... $9.95

Floppy Disk Diagnostic
by Dave Stambaugh
Disk with manual ... $24.95

✱ BOOKS

Sargon Handbook
by Dan and Kathe Spracklen ... $15.95 + $1

The BASIC Handbook
by Dr. David A. Lien ... $14.95 + $1

Z-80 Instruction Handbook
by Scelbi Publications ... $5.95 + $1

The Little Book of BASIC Style
by John Nevison ... $5.95 + $1

Pathways Through the ROM
... $19.95 + $1

TRS-80 Assembly Language Programming
by William Barden, Jr. ... $3.95 + $1

APL — An Interactive Approach
by Gilman and Rose ... $16.95 + $3

Learning Level II
by Dr. David A. Lien ... $15.95 + $1

Z-80 Software Gourmet Guide and Cookbook
from Scelbi Publications ... $15.95 + $1

TRS-80 Disk and Other Mysteries
by Harvard Pennington ... $22.95 + $1

TRS-80 Interfacing
by Jonathan Titus ... $8.95 + $1

Secret Guide to Computers
by Russ Walters ... $5.95 + $1

Information Management System
... $11.95 + $1

Calculating with BASIC ... $8.95 +$1

Understanding Micro Computers
... $9.95 + $1

Take my Computer Please
... $5.95 +$1

Learn Micro Computers
... $14.95 + $1

Micro Computer Pot Pourri
... $3.95 + $1

Personal Information Management System
Scelbi Publications ... $11.95 + $1

TRS-80 Interfacing
by Jonathon Titus ... $8.95 + $1

Z-80 and 8080 Assembly Language Programming
by Kathe Spracklen ... $7.95 + $1

✱ SUPPLIES

Cassettes: Boxes of 10 each
C-10 ... $6.95 + $1
C-20 ... $7.95 + $1

Diskettes:
Dysan (premium quality)
Box of 5 ... $29.95 + $1
BASF
Box of 5 ... $19.95 + $1.50
Box of 10 ... $34.95 + $2
Case of 100 ... $299.00 + $3
Scotch
Box of 10 ... $39.95 + $2

Diskette Storage Box ... $5.00 + $1

Floppy Armor™
Protective envelopes for shipping floppy disks.
5 pack ... $4.95 + $1

SoftSide Vinyl Binders ... $4.95 + $1

Tape Recorder Alignment Kit ... $9.95 + $1

Floppy Disk Savers ... $14.95

Floppy Disk Refills ... $7.95

- **No sales tax.**
- **All C.O.D.'s or special delivery orders are a minimum of $5.00 for special handling.**

For more details descriptions of our software and supplies send for the TSE Catalog - it's FREE! Write or call today for your copy:

TOLL-FREE
1-800-258-1790
(In NH call 673-5144)

SoftSide Publications,
P.O. Box 68, Milford, NH 03055

NOTICE

All prices are subject to change without notice. The Software Exchange is not responsible for typographical errors, including prices.

* TRS-80 is a trademark of Tandy Corporation

? ? MOVING ? ?

TO CORRECT OR CHANGE YOUR ADDRESS ATTACH LABEL FROM YOUR LATEST COPY HERE AND PRINT NEW ADDRESS BELOW:

Name ______________________________

Address ______________________________

City ____________________ State ________ Zip __________

MAIL TO: **SoftSide Publications, P.O. Box 68, Milford, NH 03055**

INPUT

continued from page 10

been the work of a local programmer, and figuring that if the program was no good, at least the cassette would be useful to store programs, it was a real bargain at $2.95! I questioned the store personnel concerning the contents of the tape but the only information they could give me was that they had played it and it was fun.

I have enclosed the tag that was affixed. This tag bears the Radio Shack logo, Tandy Copyright, and lists the contents, quantity, and price. I will send the cassette to you once I have a mail address or to your attorneys if you prefer.

Needless to say, I did find the program enjoyable and while listing it I came across your "all rights reserved" and authorship statements in lines 5 and 10. At that point I began to suspect the copy I had bought was a "pirated" version. I became further suspicious of the copy I had and checked out the software for sale ad in SoftSide S-80 magazine. I found a program listed as "Snake Eggs with sound" by Leo Christopherson for $14.95, which prompted me to write this letter.

I do not buy or use pirated software. I realize the expense and effort involved in writing and debugging programs. I feel that we, the end users, are as much a victim as the authors if such practices are allowed to continue. No one will want to provide quality programs for home computers if he cannot hope to realize a fair return for his efforts.

I thought you would be interested in knowing of this. I do not believe Radio Shack is involved but rather feel one or more of their employees saw an opportunity to make some money; money at the author's expense.

If this practice catches on, that of using a large, respected and established marketing firm as an outlet for "pirated" software, then the future does appear bleak for programmers and legitimate software dealers . . .

I have stored the cassette separately and will send it to any address you give me.

I wish you good luck in stopping this particular pirate. But as you and I know, the real solution is for the end user to refuse to buy or use pirated software.

Respectfully,
Robert N. Perkins
Tucker, Georgia

To: George Blank:

I've turned the information over to our Divisional Vice President for that area. You should hear further from him or me as soon as he has checked it out.

Our company stores are not permitted to sell ANY non-Radio Shack software . . . ESPECIALLY not "pirated". Thanks for sending the information.

Ed Juge
Radio Shack

Return of the Pirate

Dear Roger,

I am DEEPLY moved that you published my letter on page fifty of July SoftSide. I had to read it from a friend who let me Xerox it.

I let my subscription expire last month.

YOUR guilty that issue of doing just what the third paragraph says. I purchased "Lost Dutchman's Gold' and 'Adventure' and what shows up in July? Anything new or helpful? No! The same stuff I already have because I purchased it. So why buy the software AND subscribe to SoftSide for $24.00 only to get the S-A-M-E programs.

You also do a review of Adventure - it's been reviewed in EVERY other magazine 8 months ago and now again in 80-Microcomputing.

You only take issue with the 0=0. You didn't defend or take a position on points 1 - 2 - 3 & 4 so you must think they're OK, but receiving a copy of a program isn't!

1 - Copy machines are in about all libraries by now and books are still being written.

2. - Over one million video recorders were bought last year and 2 new companies are putting them out. People are taping NEW movies off the cable system and still new movies are coming out each week!

3 - Local radio stations are saying ON THE AIR they'll be "playing a new LP tonight WITHOUT interuptions so get your tape-recorders ready" - still new records come out each week!

4 - You deny number 4 and your an ass.

I don't care if Lance Micklus stops writing "games", etc. or any one else. I bought a COMPUTER to write my OWN programs, not a video arcade to play just games. The software market won't dry up. There's two programmers to replace one lost one. Maybe it should dry up a little and let these people who bought COMPUTERS (not game machines) read their manuals and write they're own software. I wish SoftSide everything it deserves.

Sincerely,
Your Friend from page fifty

P.S. Webster's 7th New Collegiate Dictionary: Pirate # 2b, pg. 644 - To lure (a worker) away from another employer by offers of betterment.

You've never done that high!?

We would like to express our appreciation to our software pirate for stimulating such a good discussion. We were able to determine who he is from the information in the letters and the postmarks, but will honor his desire to remain anonymous. G.W.B.

Sound Controversy

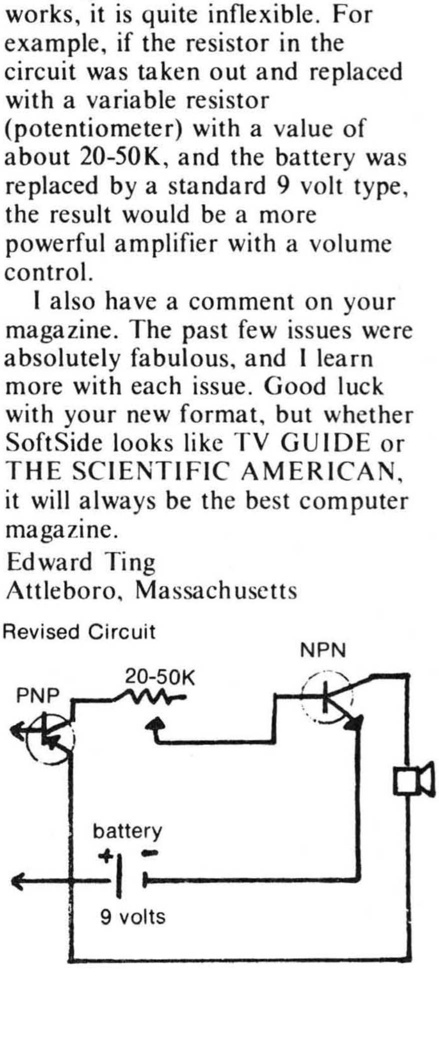
Dear SoftSide:

I am writing in response to Sherry Taylor's response to my sound interface article that appeared in your June issue. Although the method described works, it is quite inflexible. For example, if the resistor in the circuit was taken out and replaced with a variable resistor (potentiometer) with a value of about 20-50K, and the battery was replaced by a standard 9 volt type, the result would be a more powerful amplifier with a volume control.

I also have a comment on your magazine. The past few issues were absolutely fabulous, and I learn more with each issue. Good luck with your new format, but whether SoftSide looks like TV GUIDE or THE SCIENTIFIC AMERICAN, it will always be the best computer magazine.

Edward Ting
Attleboro, Massachusetts

Revised Circuit



Dear SoftSide,

After reading Sherry Taylor's letter on a sound interface, we think we have a better idea. Just disassemble your tape recorder and attach a set of wires from the recorder speaker to the earplug outlet. This allows you to hear sound by pushing the play and record buttons down and also allows hearing your tape as it is being loaded.

Sincerely,
Ron Cellini & John Fin
Philadelphia, Pennsylvania.

ConfUSR

Dear Sirs

If it was your intention to drive budding programmers wild then you have succeeded beyond your wildest dreams. Consider this line from COMBAT GALAXY or is it GALAXY COMBAT, no matter:

```
5001 DEFUSR = 13*256+12
```

I've looked in all of my BASIC books but to no avail. How about a little help.

Thanks
Rich Eidmann
Philadelphia, Pennsylvania

Dear Rich,

Sorry for the confusion. You're not the only one who was surprised by that DEFUSR. It is a disk command to set up USR calls. Level II does this same thing by POKEing the entry address into locations 16526 and 16527 (see page 8/7 in your Level II BASIC reference manual). How can a disk command be in a Level II program, you ask? The answer is that the author did not want his program to become obsolete when you eventually get that disk drive. In Level II, the disk routine does indeed cause an error, but if you look at line 105, you will see that errors send the program to line 10000. Guess what we find there?! A test for the expected "?L3" error and a couple of POKEs into none other than 16526 and 16527.

One person wrote in to say that he stopped typing the program in when he got to the disk stuff. He was quite angry at us for publishing a program which (he thought) would not run in Level II. If he had kept on typing, he would have found that the program works just fine! - J.G.

continued from page 45

```
302   FREQ. 1      DEX
303                BEQ 320 (TO TOGGLE 1)
305   ENV.1        DEY
306                BEQ 328 (TO TOGGLE 2)
308   FREQ. 2      DEC 1F
30A                BEQ 330 (TO TOGGLE 3)
30C   ENV. 2       DEC1E
30E                BEQ 33A (TO TOGGLE 4)
310   DUR 1        DEC 1A
312                BEQ 317 (TO DUR. 2)
314                JMP 302 (TO FREQ. 1)
317   DUR 2        DEC 19
319                BNE 302 (TO FREQ. 1)
31B                JMP 351 (TO D DURATION)
31E                NOP
31F                NOP
320   TOGGLE 1     LDA CO30 (READ APPLE SPEAKER PORT)
323                LDX 6 (RELOAD FREQUENCY 1 COUNTER)
325                JMP 305 (TO ENV. 1)
328   TOGGLE 2     LDA CO30 (READ APPLE SPEAKER PORT)
32B                LDY 7 (RELOAD ENVELOPE 1 COUNTER)
32D                JMP 308 (TO FREQ. 2)
330   TOGGLE 3     LDA CO20 (READ CASSETTE PORT)
333                LDA 8 (RELOAD FREQUENCY 2 COUNTER)
335                STA 1F
337                JMP 30C (TO ENV. 2)
33A   TOGGLE 4     LDA CO20 (READ CASSETTE PORT)
33D                LDA 9 (RELOAD ENVELOPE 2 COUNTER)
33F                STA 1E
341                JMP 310 (TO DUR. 1)
344   REST         LDX #OC
346   DELAY 1      DEX
347                BNE 346 (TO DELAY 1)
349   DELAY 2      DEC 1A
34B                BNE 344 (TO REST)
34D   DELAY 3      DEC 19
34F                BNE 344 (TO REST)
351   D-Duration   LDA 1D
360   INITIALIZE   LDA 6 (FREQUENCY 1)
362                LDX 1B (VOICE)
364   SHIFT 1      LSR (TIME WIDTH IS SET AS ASSIGNED
                           BY VOICE).
365                DEX
366                BNE 364 (TO SHIFT 1)
368                TAX
369                LDA 8 (FREQUENCY 2)
36B                LDY 1B (VOICE)
36D   SHIFT 2      LSR
36E                DEY
36F                BNE 36D (TO SHIFT 2)
371                STA 1F
373                LDA 7 (ENVELOPE 1)
375                TAY
376                LDA 9 (ENVELOPE 2)
378                STA 1E
37A                LDA 19 (DURATION 2)
37C                STA 1D (D DURATION)
37E                JMP 302 (TO FREQ. 1)
353                BEQ 35E
355                STA 19
357                LDA #00
359                STA 1D
35B                JMP 302 (TO FREQ. 1)
35E                RTS
35F                NOP
```

LONE STAR CORRAL

continued from page 35

Lines 1000-1050: Instructions subroutine

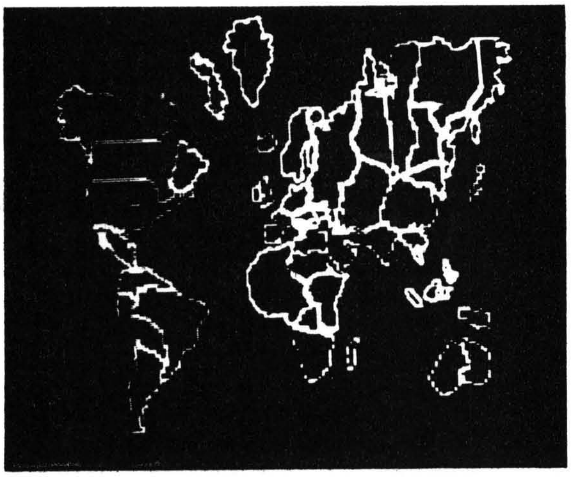
```
1000 CLS:PRINTCHR$(23);TAB(5)"THE LONE STAR CORRAL":PRINTTAB(5);
STRING$(20,45):PRINT"
   THE SUN IS STARTING TO SET
OVER THE FAR RIDGE,  THE ONLY
THING KEEPING YOU FROM SUPPER
IS ONE LAST HORSE TO FENCE-IN
FOR THE NIGHT."
1010 PRINT"   BUT HE'S A FAST HORSE, HE'S
SO FAST THAT YOU KNOW WHERE HE
WAS JUST A MOMENT AGO, AND NOT
NOW.  YOU DO KNOW HOWEVER THAT
HE MUST BE IN ONE OF THE SPOTS
NEXT TO WHERE HE WAS LAST.":GOSUB9998
1020 PRINT"FOR EXAMPLE, IF THE ";CHR$(34);"O";CHR$(34);" SHOWS
WHERE THE HORSE WAS LAST,THEN
THE ";CHR$(34);"X";CHR$(34);" SHOWS WHERE THE HORSE
COULD BE NOW:":PRINT:PRINTTAB(10);"X X X":PRINTTAB(10)"X O X":PR
INTTAB(10)"X X X"
1030 PRINT:PRINT"NOTE:   THE HORSE CAN'T MOVE
-----   INTO A SQUARE WITH A
        FENCE ALREADY IN IT!";:GOSUB9998
1040 PRINT"THE OBJECT OF THIS GAME IS TO
COMPLETLY BOX THE HORSE IN SO
THAT HE CAN'T MAKE A MOVE.

THIS MUST BE DONE IN THE FEWEST
NUMBER OF MOVES WITHOUT PLACING
A SECTION OF FENCE ON THE HORSE
THUS HURTING HIM."
1050 PRINT:PRINTTAB(6)"GOOD LUCK PARTNER!";
```

Line 9998: Print the prompt.

```
9998 PRINT@964,"(PRESS ENTER TO CONTINUE)";
```

Line 9999: Upon hitting any key, the screen clears except for the top two lines and then control returns to the main program.

```
9999 A$=INKEY$:IFA$=""THEN9999ELSEPRINT@192,CHR$(31);:RETURN
```



WANDERING ALONG

continued from page 16

containing 255 elements. We then loaded the appropriate data into each element corresponding to what BASIC interprets it to mean. For instance, B$(49) is the digit '1'. B$(65) is an uppercase letter 'A'. B$(97) is a lowercase letter 'a'. B$(178) is the word 'PRINT'. And B$(247) is equivalent to 'CHR$'. For other BASIC keywords, check out "Pathways," page 42.

Sounds reasonable...now when I PEEK at a memory cell and it returns the number 178, I simply print B$(178) and I will see the word 'PRINT' rather than a period. Since the only numbers that may show up are 0 to 255 (255 is the largest one-byte number), and we've created an array containing all 255 possible characters/keywords, the program will now function perfectly! Right?

Wrong! What about all the line numbers? How do we recover them? And what about line feeds? Where does one line end and the next line begin? And how can we figure out exactly where the program itself begins and ends?

DOWN TO THE NITTY-GRITTY:

Page 62 of "Pathways" informs us that 40A4 hexadecimal contains the pointer to the start of our BASIC program. It actually takes two bytes to do this, so we need 40A4H (least significant byte) and 40A5H (most significant byte) to do the job (16548 and 16549 in decimal). (Note that the program doesn't start here; this simply tells us where the program begins.) Looking at line 30060 in "LIST", we see that S1 has been set equal to

PEEK(16549)*256 plus PEEK (16548) - 1.

S1 is now the starting location of the program. (For our purposes here, we also subtracted one in order to recover the first line number.)

For the meantime, S2, the end address, has been set to 32767 which is the end of your 16K RAM (line 30060). BASIC does something curious...the next memory location after 32767 is not going to be 32768, it will be -32768! From there is proceeds with -32767, -32766, -32765, ... -16385 to complete a 32K system. If you happen to possess 48K, it will continue with -16384, -16382, -16381, ... -1. S2 will now be set to the actual end of your computer's memory.

According to "Pathways", page 64, the top of BASIC memory pointer is contained in 40B1H (16561) and 40B2H (16562). Therefore, line 30200 calculates your memory size as

PEEK(16561) plus 256*PEEK (16562).

Should this number be 32767, then you only had 16K of memory, and the program ends. Otherwise the program continues looking through memory from the new start location (S1 equals 32768) to the new end location (S2).

When the program encounters the command END (byte 128), the run is terminated (see line 30160). Unless this byte shows up, the program will go to the end of your memory size. (Without the END statement in 30320, the program will continue but what you see will be meaningless.)

Now for line feeds and line numbers. Between lines in memory, there are five bytes. The first one is a '0', indicating the end of a line. (This is why we had to subtract one in line 30060 - to find the zero.) When we encounter this, we will print a line feed -- CHR$(10).

The next two bytes contain the RAM address of the next line in the program. We don't need these for our present purposes.

But the last two of these five bytes contain the line number itself! That's why line 30110 calculates the line number as being

PEEK(X + 3) plus 256*PEEK (X + 4).

The rest of the line does a little fancy finagling to convert the number to a string; knock off the blank space in front of it, and add a blank after it so everything will appear exactly as it should on the screen.

AT LAST:

And now we're finished! The program should find its own beginning location; recover the line numbers; reproduce all words, numbers and variables as they appear in the original program; and terminate itself when it encounters the END statement in 30320.

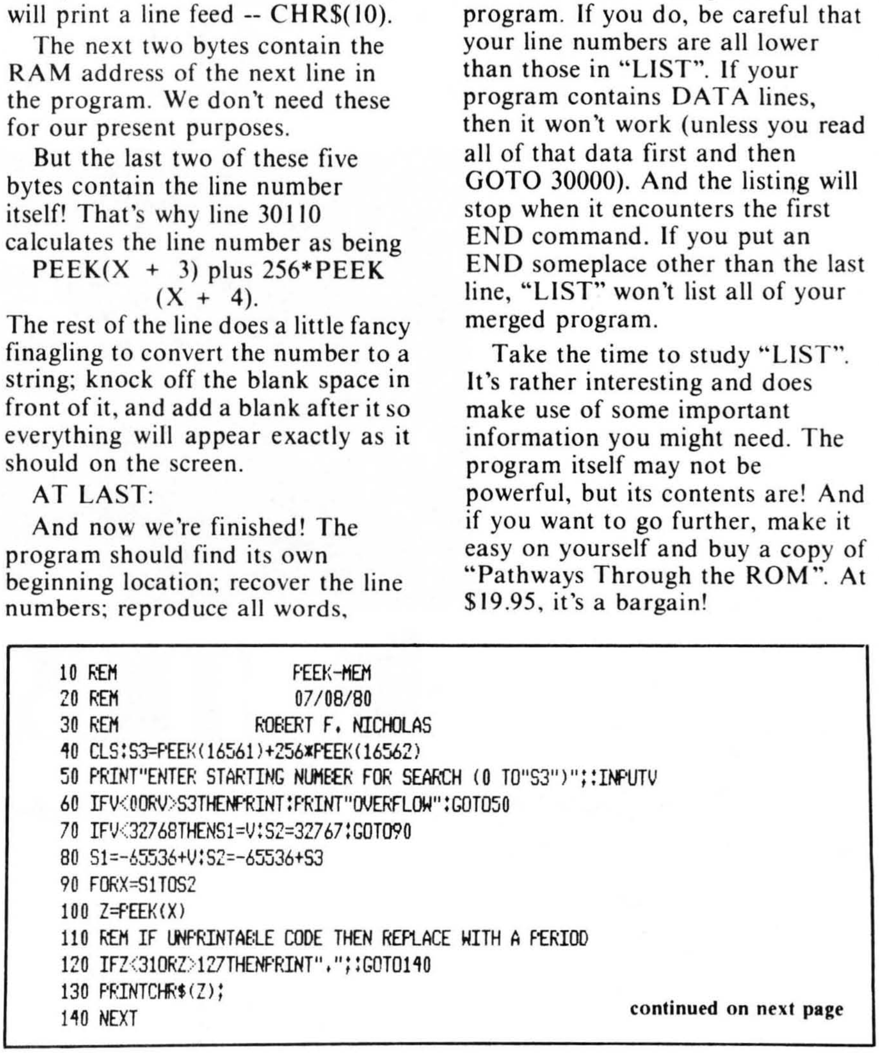
Try listing the program and then run it. What's this? It looks like a slow listing of the program itself! In other words, we've just written a program which lists itself! Listing and running it have the same effect.

Wow! What a waste of time that was. Or was it? Maybe the results of this little program aren't particularly spectacular, but it did provide some insight into how BASIC stores programs in memory. And the information we've gleaned might help in future programming -- certainly if we do any assembly language work (which is what the rest of "Pathways" will help us with).

AND YET A LITTLE MORE:

Incidentally, if you have a disk system and would like to save "LIST" as an ASCII file, you could merge it with another program and then GOTO 30000 and have "LIST" list your first program. If you do, be careful that your line numbers are all lower than those in "LIST". If your program contains DATA lines, then it won't work (unless you read all of that data first and then GOTO 30000). And the listing will stop when it encounters the first END command. If you put an END someplace other than the last line, "LIST" won't list all of your merged program.

Take the time to study "LIST". It's rather interesting and does make use of some important information you might need. The program itself may not be powerful, but its contents are! And if you want to go further, make it easy on yourself and buy a copy of "Pathways Through the ROM". At $19.95, it's a bargain!



```
10 REM                    PEEK-MEM
20 REM                    07/08/80
30 REM                 ROBERT F. NICHOLAS
40 CLS:S3=PEEK(16561)+256*PEEK(16562)
50 PRINT"ENTER STARTING NUMBER FOR SEARCH (0 TO"S3")";:INPUTV
60 IFV<0ORV>S3THENPRINT:PRINT"OVERFLOW":GOTO50
70 IFV<32768THENS1=V:S2=32767:GOTO90
80 S1=-65536+V:S2=-65536+S3
90 FORX=S1TOS2
100 Z=PEEK(X)
110 REM IF UNPRINTABLE CODE THEN REPLACE WITH A PERIOD
120 IFZ<31ORZ>127THENPRINT".";:GOTO140
130 PRINTCHR$(Z);
140 NEXT
```

continued on next page

continued from previous page

```
150 IFS2<0THENEND
160 S2=-65536+S3
170 S1=-32768:GOTO90
30000 REM       SELF-LISTING BASIC PROGRAM
30010 REM                  07/08/80
30020 REM           ROBERT F. NICHOLAS
30030 CLS:CLEAR1000:DIMB$(255)
30040 GOSUB30230
30050 REM START-OF-BASIC-PROGRAM POINTER STORED IN RAM LOCATIONS
: START=256*PEEK(16549)+PEEK(16548).  TO CALCULATE FIRST LINE N
UMBER, THIS PROGRAM SUBTRACTS ONE.
30060 S1=PEEK(16549)*256+PEEK(16548)-1:S2=32767
30070 FORX=S1TOS2
30080 REM PEEK AT PRESENT LOCATION.  IF <>0 THEN NOT AT END OF L
INE YET.
30090 Z=PEEK(X):IFZ<>0THEN30140
30100 REM AFTER FINDING A SINGLE ZERO WE USE FOLLOWING BYTES TO
RECONSTRUCT THE LINE NUMBER ITSELF.  THEN WE DO A LINE FEED, PRI
NT THE LINE NUMBER FOLLOWED BY A SPACE AND SKIP TO THE FIRST BYT
E IN THAT LINE.
30110 Z1=PEEK(X+3)+256*PEEK(X+4):Z1$=STR$(Z1):Z1$=RIGHT$(Z1$,LEN
(Z1$)-1):PRINTCHR$(10)+Z1$+" ";:X=X+4:GOTO30170
30120 REM IN THE LINE ABOVE, CHR$(10) WAS THE LINE FEED, Z1$ WAS
 THE LINE NUMBER AND X=X+4 SET THE RAM COUNTER TO THE FIRST BYTE
 OF INFORMATION IN THE PROGRAM LINE
30130 REM PRINT THE APPROPRIATE CODE FOR THIS BYTE.
30140 PRINTB$(Z);
30150 REM DID WE HIT THE 'END' OF THE PROGRAM YET?
30160 IFZ<>128THEN30170ELSE30320
30170 NEXT
30180 REM IF YOU'VE MERGED THIS WITH ANOTHER BASIC PROGRAM AND H
AVE MORE THAN 16K OF MEMORY, THEN THE NEXT SECTION COMPUTES THE
NEW START AND END ADDRESSES AND RETURNS TO THE FOR:NEXT LOOP IN
30050
30190 IFS2<0THEN30320
30200 S2=PEEK(16561)+256*PEEK(16562):IFS2<=32767THEN30320
30210 S1=-32768:S2=-65536+S2:GOTO30070
30220 REM CREATE ARRAY OF ASCII CHARACTERS AND INTERNAL CODES FO
R BASIC KEYWORDS
30230 FORX=0TO255:READB$(X):NEXT
30240 DATA ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,," ",!,,,#,$,%,&,',(,),*,+,,,-,,,/,0,1,2,3,4,5,6,7,8,9
,":",,,<,=,>,,,@,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X
,Y,Z,,,,,,,,,,
30250 DATA @,a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z
,,,,,,,,,,,END,FOR,RESET,SET,CLS,CMD,RANDOM,NEXT,DATA,INPUT,DIM,
READ,LET,GOTO,RUN,IF,RESTORE,GOSUB,RETURN,REM,STOP,ELSE,TRON,TRO
FF,DEFSTR,DEFINT,DEFSNG,DEFDBL
30260 DATA LINE,EDIT,ERROR,RESUME,OUT,ON,OPEN,FIELD,GET,PUT,CLOS
E,LOAD,MERGE,NAME,KILL,LSET,RSET,SAVE,SYSTEM,LPRINT,DEF,POKE,PRI
NT,CONT,LIST,LLIST,DELETE,AUTO,CLEAR,CLOAD,CSAVE,NEW,TAB,TO,FN,U
SING,VARPTR,USR,ERL,ERR,STRING$
30270 DATA INSTR,POINT,TIME$,MEM,INKEY$,THEN,NOT,STEP,+,-,*,/,,,
AND,OR,>,=,<,SGN,INT,ABS,FRE,INP,POS,SQR,RND,LOG,EXP,COS,SIN,TAN
,ATN,PEEK,CVI,CVS,CVD,EOF,LOC,LOF,MKI$,MKS$,MKD$,CINT,CSNG,CDBL,
FIX,LEN,STR$,VAL,ASC,CHR$,LEFT$,RIGHT$,MID$
30280 DATA ,,,,,,,,,,
30290 REM SPECIAL ASSIGNMENT CODES (ARROWS COMMAS AND QUOTES)
30300 B$(209)=CHR$(91):B$(91)=CHR$(91):B$(123)=CHR$(91):B$(92)=C
HR$(92):B$(124)=CHR$(92):B$(93)=CHR$(93):B$(125)=CHR$(93):B$(94)
=CHR$(94):B$(126)=CHR$(94):B$(95)=CHR$(95):B$(127)=CHR$(95)
30310 B$(63)=CHR$(63):B$(34)=CHR$(34):B$(44)=CHR$(44):B$(59)=CHR
$(59):B$(10)=CHR$(10):B$(13)=CHR$(13):B$(25)=CHR$(25):B$(26)=CHR
$(26):RETURN
30320 END
```

PROGRAMMING HINT

Here's a machine language version of the "Tension Breaker" program; it's much faster than the BASIC program, and it's only 21 bytes long. Just poke in the numbers with the routine provided, and then jump to 32000 with the SYSTEM command. To regain control, press the RESET button.

```
100 FOR X=32000 TO 32020:READ Y: POKE X,Y:NEXT
200 DATA 205,201,1,33,0,56,126,183,40,3,205,51,0,124,254,57,40,2
41,35,24,241
```

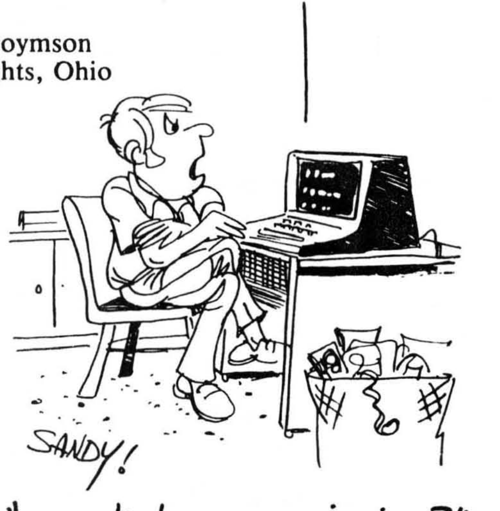
DISASSEMBLED LISTING

Address	Machine Code	Mnemonics	
7D00	CDC901	Call 01C9H	;clear the screen
7D03	21 00 38	LD HL,3800H	;starting address of keyboard memory
7D06	7E	LD A,(HL)	;put character in A register
7D07	B7	OR A	;if no character,
7D08	2803	JR Z,$+05H	;skip video routine
7D0A	CD 33 00	Call 0033H	;print character at cursor
7D0D	7C	LD A,H	;if keyboard scan
7D0E	FE39	CP 39H	;is completed, then
7D10	28 Fl	JR Z,$-0DH	;start again from 7D03H
7012	23	INC HL	;continue scan
7D13	18 Fl	JR $-0DH	;get next character at 7D06H

Interesting Combinations

WERJKL 12345789
WEF ← → WFK (while holding down, press and release OP)
G ← →
ASD (after screen is filled)

Norman Efroymson
Shaker Heights, Ohio



"Oh, Who asked your opinion?"

continued from page 45

Lines 1100 - 1120 take care of the case where the line is drawn in the Y-direction. (Solve the above equations for X if you don't believe me!)

Lines 5 - 60 allow us to test the subroutine. First, you are asked for the co-ordinates of two points. If they do not live on the screen, line 30 will send you back to do your homework. After line 40 has cleared the screen, line 50 will turn on your two specially selected points. After a short but reverent pause, line 60 calls the line-drawing subroutine (note that A, B, C and D are known to the machine at this point). After another pause, the computer will ask for another pair of test points.

```
5 CLS
10 INPUT"1ST POINT (A,B)";A,B
20 INPUT"2ND POINT (C,D)";C,D
30 IFA<0ORA>127ORB<0ORB>47ORC<0ORC>127ORD<0ORD>47THENRUN
40 CLS
50 SET(A,B):SET(C,D):FORI=1TO300:NEXT
60 GOSUB1000:PRINT@0,;:FORI=1TO300:NEXT:GOTO10
1000 IF ABS (B-D) > ABS (A-C) THEN 1100
1010 FOR X=A TO C STEP SGN (C-A)
1020  SET (X,B+(B-D)/(A-C)*(X-A)+.5)
1030 NEXT:GOTO1130
1100 FOR Y=B TO D STEP SGN(D-B)
1110 SET(A+(Y-B)*(C-A)/(D-B)+.5,Y)
1120 NEXT
1130 A=C:B=D:RETURN
```

In the interest of speed and memory conservation, the subroutine may be shortened as follows:

```
1000 IFABS(B-D)>ABS(A-C)THEN1100
1010 E=(B-D)/(A-C):F=B+.5:FORX=ATOCSTEPSGN(C-A):SET(X,F+E*(X-A))
:NEXT:GOTO1110
1100 E=(C-A)/(D-B):F=A+.5:FORY=BTODSTEPSGN(D-B):SET(F+(Y-B)*E,Y)
:NEXT
1110 A=C:B=D:RETURN
```

The subroutine is set up to allow you to easily connect the beginning of a new line to the end of the last line (see line 1130). Thus to create a diamond, centered in the middle of the screen, we could use the following (assuming the line-drawing subroutine was at line 1000):

```
5 CLS
10 A=63:     B=0
20 C=127:    D=23:   GOSUB 1000
30 C=63:     D=47:   GOSUB 1000
40 C=0:      D=23:   GOSUB 1000
50 C=63:     D=0:    GOSUB 1000
60 GOTO 60
```

Notice that after the first line, only the ending co-ordinates need to be specified for each subsequent line.

While this article began as an attempt to simulate the DRAWTO and HPLOT commands of the Atari and Apple, it is apparent that the line-drawing subroutine can have its uses as a stand-alone TRS-80 feature. Enjoy it!

(Note: The second subroutine listed can be used in Joan Truckenbrod's graphics routine if the following changes are also made:

Line 60 in that program should put the figure toward the center of the S-80's 128x48 screen.

Line 100 should be CLS.

Line 180 should be A=NX: B=NY

Line 220 should be C=NX: D=NY: GOSUB 1000)

IMAGE
COMPUTER PRODUCTS

STRATEGY PACK I 6404

Roman Checkers. This ancient game has been a favorite for hundreds of years. It couldn't be easier to play, yet playing the game well takes skill, cunning, and strategy as you try to out-think your opponent.
Frame up. Try to out-manuever your opponent or play against the computer in this game of wits and calculated strategy. You will alternate selecting numbers and controlling your opponents choices. Joysticks are optional.
2 programs on cassette................. $19.95

STRATEGY PACK II 6405

Target Chase. Choose a partner or challenge the computer to this fast action, catch-or-be caught adventure.
Tunnels. In this electronics hide-and-seek, only the computer knows where you will appear next.
Survival. Intercept your opponent before you crash or are devoured.
Snake Hunt. Fast maneuvering and quick response are essential to capture - or escape from - your opponent. Joysticks are required.
4 programs on cassette................. $19.95

SKILL BUILDER I 6406

Bingo Duel. This fast-action skill game for one or two players provides an exciting challenge, because young children and adults can compete equally. The computer adjusts to match your skill and problems are specifically selected to help you gain speed.
Number Hunt. Matching numbers is easy enough for young children, yet this computer game quickly advances in difficulty to challenge the experts. Joysticks are required.
2 programs on cassette................. $19.95

WALL STREET CHALLENGE 6402

This computer simulation of the stock exchange is easy to play and always challenging. Invest in several corporations ranging from Municipal Power and Light, a blue chip stock that usually provides steady growth, to Offshore Industries Limited, a high-flying speculative stock that is certain to change often. Stock charts, and the Dow Jones show you the trends. Both 8K and 16K memory versions are included.
Cassette $14.95

ALL STAR BASEBALL 6401

This computerized version of America's favorite sport is full of fast action fun. Each player takes a turn at bat while the other player is both pitcher and the outfield. Innings, score, batting averages, and even errors are calculated by the computer and displayed on the official scoreboard. Balls, strikes, and all plays are called with absolute precision. Joysticks are optional for both players. Both 8K and 16K memory versions included.
Cassette $14.95

MIND MASTER 6403

This classic strategy game takes on a new dimension as the computer designs the hidden problems and reports the results of each guess. Multiple players may compete against the computer and each player may select the level of difficulty that matches their skill, ability, and patience. This program also contains a formula for solving logic problems. Create the answer and watch the computer use deductive logic to discover the secret code. Both 8K and 16K versions.
Cassette $14.95

TO ORDER TOLL FREE
1-800-258-1790
(In NH call 673-5144)

IMAGE
COMPUTER PRODUCTS

continued from page 23

Lines 310-450: Match section

This section handles the special case of two matching prize selections. Line 310 displays a message and waits a short period before continuing. The two squares are blanked out in line 320. Line 330 prints another message and line 340 adds the prize to the player's prize list. Lines 350 through 370 fill in the pieces of the puzzle and wait for a moment before asking in line 380 for the player's guess at the puzzle. Lines 390 through 410 accept the guess using the INKEY$ command. If the guess is correct the program jumps to the next section; otherwise, the same player continues at line 190. A check is made at line 425 to see if there is only one square left. If there is, there can be no further matches, obviously, and the program jumps to line 600.

```
310 PRINT@965,CHR$(31)," * * *  A  M A T C H !  * * *";:FORTI=1T
O1000:NEXTTI
320 PRINT@L1,B$(1);R$;B$(1);:PRINT@LO,B$(1);R$;B$(1);
330 PRINT@965,CHR$(31);"HERE ARE TWO MORE PIECES OF THE PUZZLE!"
;
340 PR(P,0)=PR(P,0)+1:PR(P,PR(P,0))=A(W):IFA(W)=NPTHENPR(P,PR(P,
0))=A(W1)
350 A(W)=0:A(W1)=0:PRINT@LO,B$(VAL(LEFT$(T$(W),1)));R$;B$(VAL(RI
GHT$(T$(W),1)));
360 PRINT@L1,B$(VAL(LEFT$(T$(W1),1)));R$;B$(VAL(RIGHT$(T$(W1),1)
));
370 FORTI=1TO500:NEXTTI
380 PRINT@965,CHR$(31);Y$(P);", WHAT IS THE SECRET NUMBER? ";
390 G$=INKEY$:G$="":FORI=1TO5
400 H$=INKEY$:IFH$=""THEN400
405 IFASC(H$)<48ORASC(H$)>57THEN400
410 G$=G$+H$:PRINTH$;:NEXTI:FORTI=1TO200:NEXTTI
420 IFG$=S$THEN460
425 J=0:FORI=1TO35:J=J+SGN(A(I)):NEXT:IFJ=1THEN600
430 PRINT@965,CHR$(31);"SORRY!  ";G$;" WAS WRONG, BUT YOU MAY CO
NTINUE";
440 FORTI=1TO1000:NEXTTI
450 GOTO190
```

Lines 460-540: Puzzle has been solved

The solution has been guessed and winnings are displayed. Lines 460 through 490 show the complete puzzle. After a delay, lines 500 through 520 display the winner's prizes. Lines 525 and 526 indicate total winnings for both players to date and lines 530 and 540 wait for player input to begin a new game.

```
460 FORI=1TO35:IFA(I)=0THEN490
470 RO=INT((I-1)/5):CO=I-RO*5:K=RO*128+(CO-1)*12-64
480 FORJ=1TO2:PRINT@K+J*64,B$(VAL(MID$(T$(I),J,1)));:NEXTJ
490 NEXTI:FORTI=1TO1000:NEXTTI
500 CLS:TO=0:PRINT"

YOU GOT IT, ";Y$(P);"!"
510 PRINT"THAT MEANS YOU HAVE WON THE FOLLWING: ":FORJ=1TOPR(P,0
)
515 RESTORE:FORI=1TO10:READZ$:NEXTI
520 FORK=1TOPR(P,J):READP$,M:NEXTK:PRINTUSING"  %              %  WO
RTH $$####";P$,M:TO=TO+M:NEXTJ
525 PRINT"THAT IS A TOTAL OF $";TO:MT(P)=MT(P)+TO:PRINT"
SO FAR THIS SESSION, ";Y$(1);" HAS WON $";MT(1);"
                AND ";Y$(2);" HAS WON $";MT(2)
530 PRINT," TOUCH ANY KEY FOR A NEW GAME";:Z$=INKEY$
540 Z$=INKEY$:IFZ$=""THEN540ELSE10
```

Lines 600-610: Nobody wins

If neither player guesses the solution, a message is printed and a new game is set up if the players wish.

```
600 PRINT@965,CHR$(31);"SORRY!  NOBODY WINS THIS GAME -- THE NUM
BER WAS ";S$;
610 FORTI=1TO2000:NEXTTI:PRINT@960,CHR$(30);:GOTO530
```

Lines 1000-1060: Location and printing of prizes.

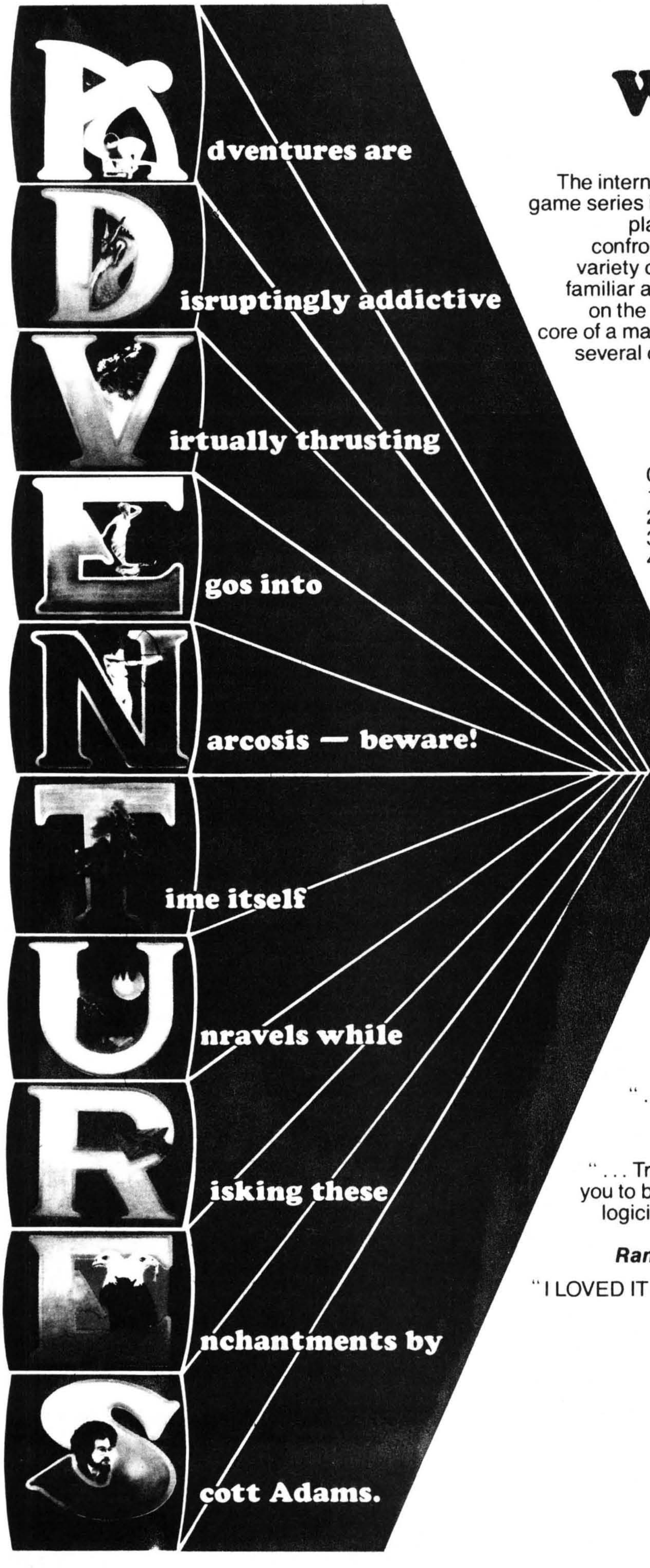
When a square is chosen, this subroutine locates the description of the prize in that square and calculates the print location (LO) to be used in displaying the prize name. The prize is printed in line 1060.

```
1000 RESTORE:FORI=1TO10:READZ$:NEXTI
1010 FORI=1TOA(W):READP$,M:NEXTI
1015 DATA" MOTOR HOME ",12000,"EUROPE TRIP ",3000,"  GOLF CART "
,2000,"    PIANO   ",5000,"  RECLINER  ",600
1020 DATA"  1980 CAR  ",6000,"  FUR COAT  ",5000,"   TRS-80   ",
1000,"  SAILBOAT  ",9000,"   $5000    ",5000
1025 DATA"CLOCK RADIO ",65," MOTOR BIKE ",750,"   CARPET   ",900
,"  WATER BED ",1100,"   CAMERA   ",250
1030 DATA"  COLOR TV  ",1500," DISHWASHER ",500," MICROWAVE  ",6
00,"MEXICO TRIP ",3000,"REFRIGERATOR",800
1035 DATA" POGO STICK ",5,"  PET SKUNK ",50,"  AQUARIUM  ",500,"
 CHINA SET  ",500,"  CANDY BAR ",1,"   STEREO   ",350,"  LUGGAGE
   ",500," TELESCOPE  ",390,"  BICYCLE   ",120
1040 DATA"LINE PRINTER",800," DISK DRIVE ",500,"  2 GOATS   ",25
," HARMONICA  ",5," WILD CARD! ",0
1050 LO=(W-INT((W-1)/5)*5-1)*12+INT((W-1)/5)*128
1060 PRINT@LO,P$;R$;B$(1);:RETURN
```

Lines 2000-2090: Instructions

The instructions for the game are printed for players who request them. In addition, samples of the ten numerals are displayed on randomly chosen backgrounds using the NAME command.

```
2000 CLS:PRINTCHR$(23);"    THIS IS THE GAME OF
       CONCENTRATION":PRINT:PRINT"THE OBJECT OF THE GAME IS TO":
PRINT"DISCOVER THE SECRET NUMBER":PRINT"HIDDEN BEHIND THE GAME B
OARD."
2010 PRINT"THE BOARD IS DIVIDED INTO A":PRINT"5 X 7 CHECKERBOARD
. BEHIND EACH":PRINT"LABELED SQUARE IS ONE OF A":PRINT"PAIR OF P
RIZES. ON YOUR TURN,"
2020 PRINT"YOU CHOOSE TWO SQUARES: IF THE":PRINT"PRIZES MATCH, Y
OU WIN THE PRIZE":PRINT"AND 2 PIECES OF THE PUZZLE":PRINT"
ARE REVEALED.":PRINT" PRESS ANY KEY TO CONTINUE";
2030 I$=INKEY$:IFI$=""THEN2030
2040 CLS:PRINTCHR$(23);"AS LONG AS YOU MAKE A MATCH,":PRINT"YOUR
 TURN CONTINUES. EACH MATCH":PRINT"YOU MAKE ALLOWS YOU TO GUESS"
2050 PRINT"AT THE PUZZLE. THE PUZZLE IS A":PRINT"5-DIGIT NUMBER
CONSISTING OF":PRINT"TALL,THIN VERSIONS OF NUMERALS":PRINT"ZERO
THROUGH NINE. TO SEE THE"
2060 PRINT"NUMERALS AS THEY WILL APPEAR,":PRINT@722,"pRESS ENTER
"
2070 I$=INKEY$:IFI$=""THEN2070
2080 RESTORE:FORI=0TO9:READZ$:CLS:POKEX3+5,128+RND(63):NAME:PRIN
T@984," THE NUMBER ";LEFT$(Z$,1);" ";:PRINT@25,;
2090 FORK=2TO15:PRINTB$(VAL(MID$(Z$,K,1)));R$;:NEXTK:PRINTB$(1);
:FORTI=1TO500:NEXTTI,I:POKEX3+5,CH:RETURN
```



Great Scott! What have you created?

The international reception of Scott Adams' ADVENTURE game series is phenomenal! Each of the ten games takes the player on odyssies of intrigue and danger, through confrontations with various villains, in pursuit of a wide variety of mysteries and treasures, in surroundings both familiar and alien. You may find yourself in a voodoo den, on the edge of a strange and threatening galaxy, in the core of a malfunctioning nuclear reactor, or in any one of the several other settings; ranging from mundane to bizarre.

Challenge your genius! Order an ADVENTURE

Master these ADVENTURES!

0. Special Sampler
1. Adventureland
2. Pirates Adventure
3. Mission Impossible
4. Voodoo Castle
5. The Count
6. Strange Odyssey
7. Mystery Fun House
8. Pyramid of Doom
9. Ghost Town

ADVENTURES are available for Pet™, Sorcerer™, Apple™, TRS-80™ Microcomputer, and other major personal computers.

adventure international
Box 3435 • Longwood, Fl 32750
(305) 862-6917

Look who's raving about ADVENTURE

Out of 50 programs reviewed, Adventure was rated No. 1! "Highly recommended."
80 Software Critique

"Adams' Adventure is exquisite. It is a true tour-de-force . . . "
Recreational Computing

" . . . It is worth the money to buy a computer just to play Adventure."
Software Directory Winter 1980

" . . . Truly absorbing simulations. These games require you to be inquisitive, innovative, a thinker, a risk taker, a logician, a warrior, and a lover of real challenges — in short, an Adventurer."
Ramon Zamora, Recreational Computing Issue 4

"I LOVED IT! There was a real challenge to this program . . . "
Captain 80, 80 Microcomputing Issue 1

Adventures 1-9, $14.95 each

TO ORDER: CALL TOLL FREE
1-800-258-1790
(In NH call 673-5144)

THE STEREO GENERATOR

continued from page 19

```
418 N = B / S3:NN = FS / S3: GOSUB
     20
420  GOSUB 30
422 N = A / S3:NN = A / S3: GOSUB
     20
424 N = FS / S3:NN = D / S2:T = Q
     U: GOSUB 20
426 N = G / S3:NN = FS / S2:T = E
     I: GOSUB 20
428 N = G / S3:NN = D / S3:T = QU
     : GOSUB 20
430 N = FS / S3:NN = FS / S2:T =
     EI: GOSUB 20
432 N = FS / S3:NN = D / S2: GOSUB
     20
434 N = E / S3:NN = D / S2: GOSUB
     20
436 N = G / S3:NN = A / S2: GOSUB
     20
438 N = G / S3:NN = CS / S3:T = Q
     U: GOSUB 20
440 N = FS / S3:NN = A / S2:T = E
     I: GOSUB 20
442 N = FS / S3:NN = D / S2: GOSUB
     20
444 N = E / S3:NN = D / S2: GOSUB
     20
446 N = G / S3:NN = A / S2: GOSUB
     20
448 N = G / S3:NN = CS / S3:T = Q
     U: GOSUB 20
450 N = FS / S3:NN = A / S2:T = E
     I: GOSUB 20
452 N = FS / S3:NN = D / S2:T = Q
     U: GOSUB 20
454 N = FS / S3:NN = A / S2:T = E
     I: GOSUB 20
456 N = FS / S3:NN = D / S3:T = Q
     U: GOSUB 20
458 N = FS / S3:NN = A / S2:T = E
     I: GOSUB 20
 Envelopes T1 and T2
475 T1 =  RND (1) / 25
490 T2 =  RND (1) / 25
505 T1 =  INT (T1 * 100):T1 = T1 /
     100:T2 =  INT (T2 * 100):T2 =
     T2 / 100: VTAB 14: HTAB 4
510  PRINT "WAVE ENVELOPES T1=";T
     1;" AND T2=";T2;"      "

 Tempo selection.

531 X = 28 + ( INT ( RND (1) * 10
     )):TEMPO = X: GOSUB 18
532  VTAB 10: HTAB 11
533  PRINT "TEMPO="; INT (((TEMPO
      - 33) /  - .56) + 100);" BE
     ATS/MIN "
 Voice selection.

535  PRINT :WF =  INT ( RND (1) *
     8 + 1): POKE VOICE,WF
537  HTAB 13: PRINT "VOICE FACTOR
     =";WF;"   "
 Octave selection.

541 X =  INT (3 *  RND (1)) + 1: IF
     X = WW THEN  GOTO 541
542 WW = X
543  ON X GOSUB 545,546,547: VTAB
     16: HTAB 13: PRINT "OCTAVES=
     ";S2;" AND ";S3;"      "
544  GOTO 550
545 S2 = 2:S3 = 2: RETURN
546 S2 = 1:S3 = 2: RETURN
547 S2 = 2:S3 = 1: RETURN
550  GOSUB 650
560  GOTO 400
570  HOME : VTAB 8
580  PRINT "   TO DEVELOP YOUR OW
     N PROGRAM:": PRINT : HTAB 6:
      PRINT "1.DELETE LINES 100-7
     00": HTAB 6: PRINT "2.BEGIN
     HIGHER THAN LINE 50": HTAB 6
     : PRINT "3.REFER TO INSTRUCT
     ION SHEET"
582  VTAB 14
585  PRINT "   FOR BEST RESULTS C
     REATE  PROGRAM  IN OCTAVES S
     1,S2, AND S3 WHEN USING  OVE
     R- TONES."
590  END
600  VTAB 10
620  PRINT "    CONNECT SPEAKER TO
      CASSETTE RECORDER OR COMPUT
     ER TO HIFI AMPLIFIER  (SEE I
     N- STRUCTION SHEET). TURN ON
      CASSETTE RE-  CORDER IN 'RE
     CORD' MODE."
630  GOSUB 650: GOSUB 680: GOTO 6
     30
635  HOME : VTAB 7: HTAB 4
640  PRINT "RANDOM VARIATIONS OF
     'BARCAROLLE'"
645  GOTO 475
650  VTAB 19: INVERSE : HTAB 10: PRINT
     "PRESS ESC KEY TO STOP"
655  NORMAL
660 KEY =  PEEK ( - 16384): POKE
      - 16368,0
661  IF KEY = 155 THEN 570
662  RETURN
680  VTAB 21: INVERSE : HTAB 9: PRINT
     "PRESS RETURN TO CONTINUE"
681  NORMAL
682  IF KEY = 141 THEN 635
683  RETURN
```

continued from page 15

```
195  REM scale the remaining
       coordinates
200  NX = (X(J)-XC) *S+XC
210  NY = (Y(J)-YC) *S+YC
220  HPLOT  TO  NX,NY
230  NEXT  J
```

STRETCHING AND COMPRESSING FIGURES

Proportional Size Changes: Horizontal Scaling

The horizontal scaling procedure stretches or compresses the horizontal dimension of the figure. If the scaling factor is greater than one, scaling produces the effect that the figure is being pulled out from both sides. A scaling factor less than one will create the effect that the sides of the figure have been pushed in or compressed. Horizontal scaling is achieved by changing only the x coordinates with the scaling expression. The vertical coordinates remain constant.

```
160 NX = (X(1)-XC) *S+XC
180  HPLOT  NX,Y(1)
190  FOR J = 2 TO NP
200  NX = (X(J)-XC) *S+XC
220  HPLOT NX,Y(J)
230  NEXT J
```

Vertical Scaling

Vertical scaling with a scaling factor greater than one gives the effect that the top and bottom of the figure have been pulled out. If the scaling factor is less than one, the form will appear to have been pushed in from the top and bottom. This type of scaling is achieved by changing the y coordinates only, using the given scaling expression. The horizontal coordinates remain the same.

```
170  NY =(Y(1)-YC)*S+YC
180  HPLOT X(1), NY
190  FOR J = 2 TO NP
210  NY =(Y(J)-YC) *S+YC
220  HPLOT X(J),NY
230  NEXT  J
```

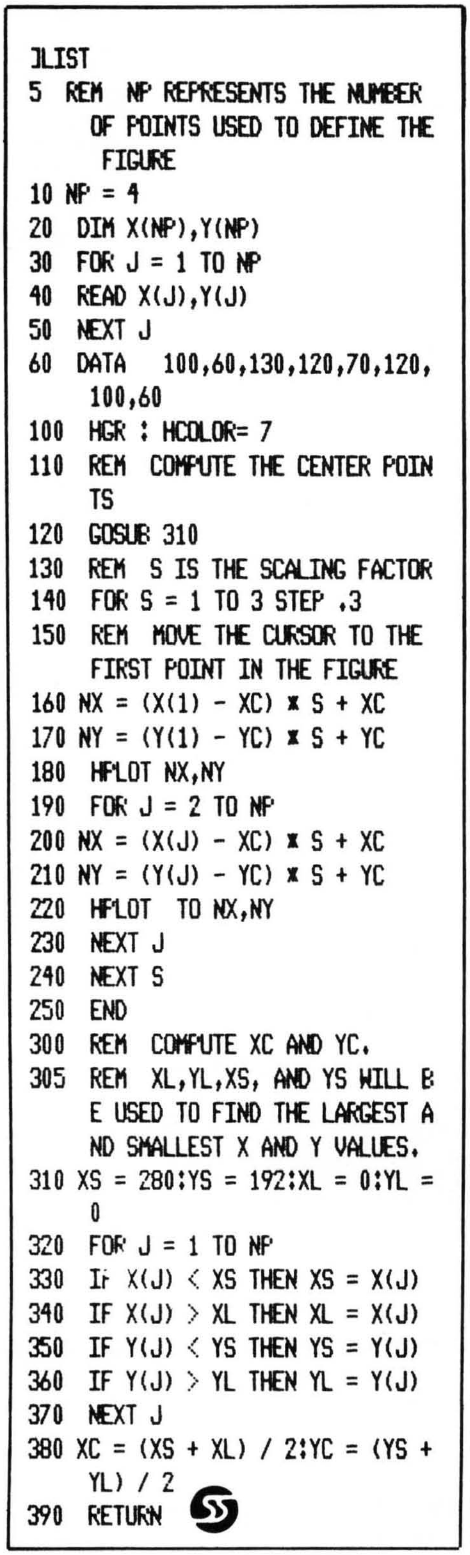
Variable Horizontal and Vertical Scaling Factors

Figures can be changed and distorted in numerous ways by applying different scaling factors to the horizontal and vertical dimensions of the form. In this instance two scale factors will be employed, SX for the x coordinates, and SY for the y coordinates. A broad range of visual effects can be created by changing the horizontal and vertical dimensions of a form differently.

```
200  NX = (X(J)-XC) *SX+XC
210  NY = (Y(J)-YC) *SY +YC
```

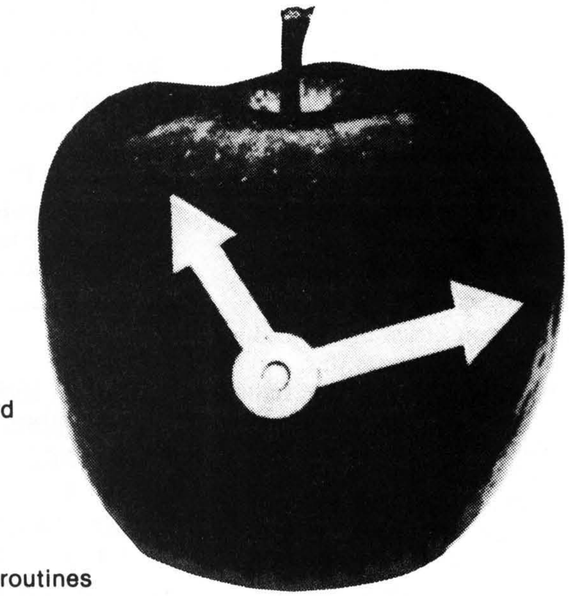
SCALING PROGRAM:

The following program sequentially increases the size of a given triangle:



```
]LIST
5  REM  NP REPRESENTS THE NUMBER
     OF POINTS USED TO DEFINE THE
      FIGURE
10 NP = 4
20  DIM X(NP),Y(NP)
30  FOR J = 1 TO NP
40  READ X(J),Y(J)
50  NEXT J
60  DATA   100,60,130,120,70,120,
     100,60
100  HGR : HCOLOR= 7
110  REM  COMPUTE THE CENTER POIN
     TS
120  GOSUB 310
130  REM  S IS THE SCALING FACTOR
140  FOR S = 1 TO 3 STEP .3
150  REM  MOVE THE CURSOR TO THE
     FIRST POINT IN THE FIGURE
160 NX = (X(1) - XC) * S + XC
170 NY = (Y(1) - YC) * S + YC
180  HPLOT NX,NY
190  FOR J = 2 TO NP
200 NX = (X(J) - XC) * S + XC
210 NY = (Y(J) - YC) * S + YC
220  HPLOT  TO NX,NY
230  NEXT J
240  NEXT S
250  END
300  REM  COMPUTE XC AND YC.
305  REM  XL,YL,XS, AND YS WILL B
     E USED TO FIND THE LARGEST A
     ND SMALLEST X AND Y VALUES.
310 XS = 280:YS = 192:XL = 0:YL =
     0
320  FOR J = 1 TO NP
330  IF X(J) < XS THEN XS = X(J)
340  IF X(J) > XL THEN XL = X(J)
350  IF Y(J) < YS THEN YS = Y(J)
360  IF Y(J) > YL THEN YL = Y(J)
370  NEXT J
380 XC = (XS + XL) / 2:YC = (YS +
     YL) / 2
390  RETURN
```

Mountain Hardware, Inc.



THE APPLE CLOCK.

The Apple Clock from Mountain Hardware keeps time and date in 1mS increments for over one year. On-board battery backup keeps the clock running in the event of power outage. Software controlled interrupts are generated by the clock. That means you can call up schedules, time events, date printouts . . . all in real time on a programmed schedule.

EASY TO USE!
The Apple Clock is easily accessed from BASIC using routines carried in on-board ROM. With it, you can read time and program time-dependent functions for virtually any interval. From milliseconds to days, months or a year.

PLUG IN AND GO!
Plug the Apple Clock into a peripheral slot on your Apple II and you're ready to go.

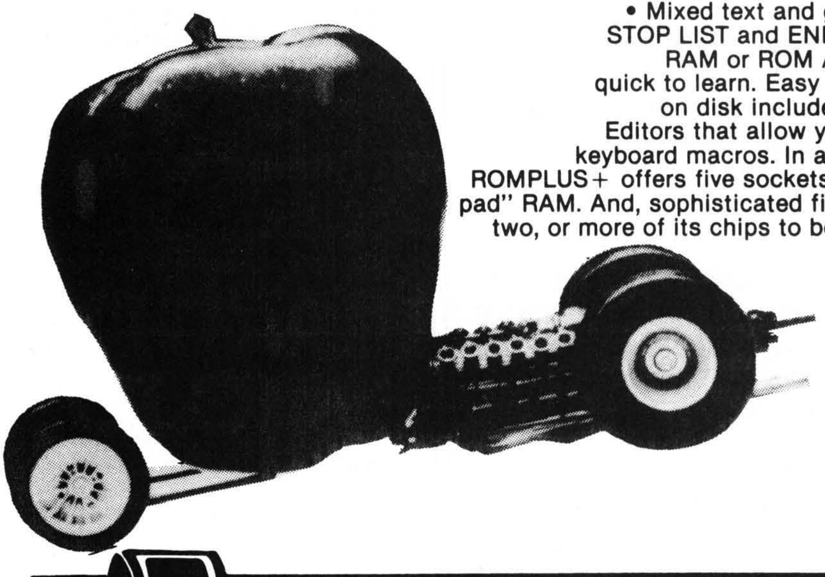
$262.50

ROMPLUS+

NEW EXPANDABILITY.

ROMPLUS+ is a peripheral board whose added features can turn the APPLE* computer into the most powerful personal computer available today. ROMPLUS+ provides six sockets to accept individually addressable 2K ROM's or EPROM's. Keyboard Filter™ a 2K ROM program, comes installed on the ROMPLUS+ board and adds many useful features to your Apple, including: • Upper and lower case letters. • Multiple user-defined character sets. • Colored or inverse-colored letters. • Keyboard macros —two key-stroke, automatic typing of multiple, user-defined words or phrases. Including BASIC and DOS commands. • Mixed text and graphics. • Improved cursor control. STOP LIST and END LIST. • Works with integer BASIC, RAM or ROM Applesoft, and DOS. • And more . . . quick to learn. Easy to use. • Software support provided on disk includes demonstration programs and two Editors that allow you to define your own characters or keyboard macros. In addition to the Keyboard Filter ROM, ROMPLUS+ offers five sockets for ROM or EPROM, plus "scratch-pad" RAM. And, sophisticated firmware on ROMPLUS+ allows one, two, or more of its chips to be used simultaneously for programs longer than 2K.

$187.50



HARDSIDE 6 SOUTH ST , MILFORD, NH 03055

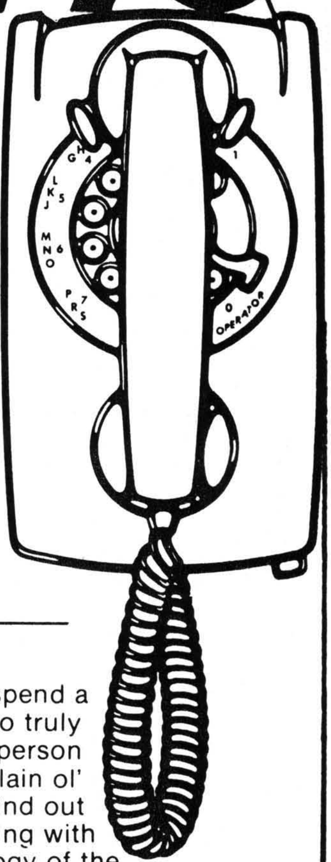
To order call TOLL-FREE 1-800-258-1790 (In NH call 673-5144)

The Software Exchange & HardSide (Div of Robitaille & Sons Enterprises Inc) SoftSide Publications

VISA

Where can you dial one TOLL FREE number and be able to select the finest software from the world's foremost suppliers?

1-800-258-1790



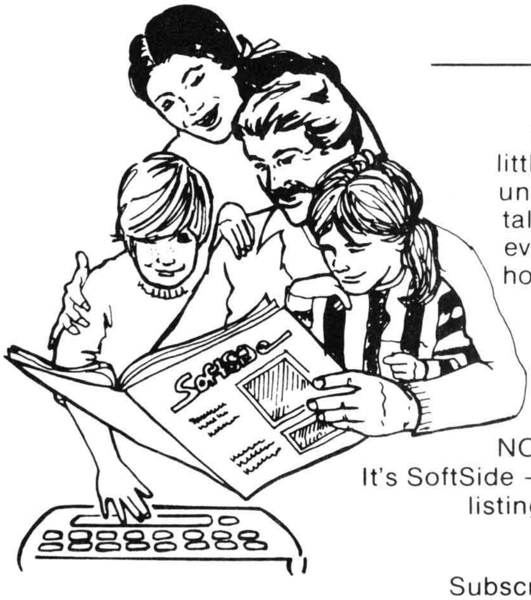
The Software Exchange

Our stockrooms are overflowing with the finest software available in today's marketplace. We carry software from all these companies:

- ●Racet
- ●Adventure International
- ●Personal Software
- ●Web Associates
- ●Small Systems Software
- ●Sensational Software
- ●Synergistic Software
- ●Strategic Simulations
- ●Lance Micklus
- ●Softape
- ●Quality Software
- ●Image Products
- ●Acorn Software
- ●Hayden
- ●Muse
- ●Microsoft
- ●Apparat

We could not possibly list all of the software on this page, however, if you send us $1.00, you'll receive our catalog plus a $2.00 credit toward your next purchase!

TSE



Suppose you and your family could spend a little time each month with someone who truly understands microcomputers. And this person talked to you about your computer in plain ol' everyday English. Suppose you could find out how these people are successfully dealing with the ever-increasing technology of the microcomputer. Suppose too, that each article was geared to the person with a beginning or intermediate interest in personal computing.

NOW you can get all this, and more, at a price you can afford. It's SoftSide — the magazine that provides its readers with tried and true listings, along with structured "walk-thrus" of some of the most stimulating software pieces, month after enjoyable month.

Subscribe today! Until November 1, 1980, the price is only $15.00. After November 1, the price will be $18.00, so why not save $3.00 and start your family on the road to microcomputing?

SS A SoftSide Publication

The Software Exchange 6 SOUTH ST., MILFORD, NH 03055
To order: Call Toll-Free 1-800-258-1790 (in NH call (603) 673-5144)
The Software Exchange & HardSide (Div. of Robitaille & Sons, Enterprises, Inc.), SoftSide Publications

VISA
master charge

MODEL I TO MODEL III PROGRAM CONVERSION

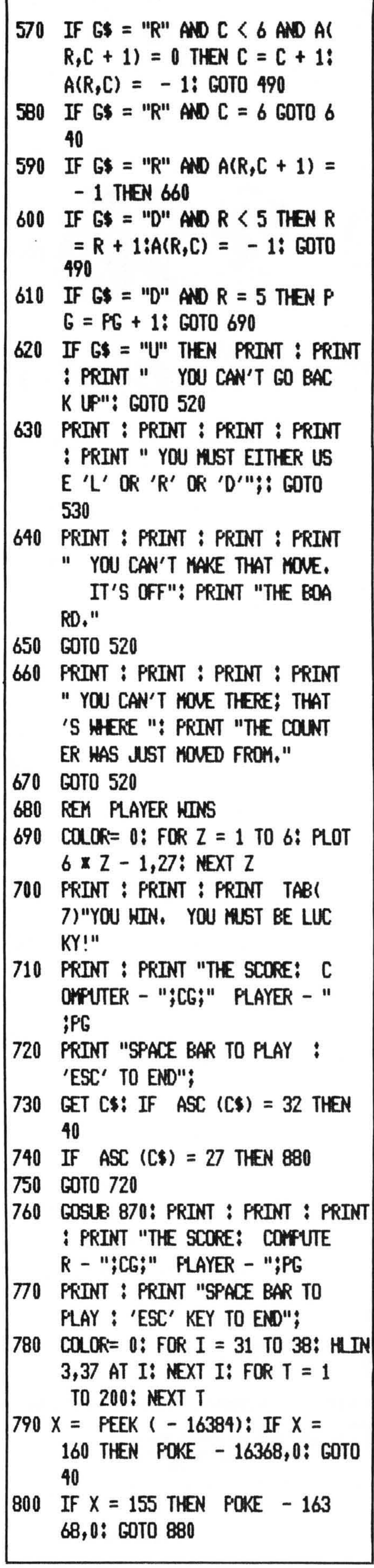
The following information from the TRS-80 Model III reference manual should be of interest, and relief, to all Model I owners.

From a language standpoint, Model III BASIC is fully compatible with Model I Level II BASIC. In fact, the two BASIC'S are identical, except the Model III BASIC includes one more function, TIME$.
However, because of Model III's many special features not available in Model I, there are some internal differences which may require that you modify any Model I Level II BASIC programs you may have.

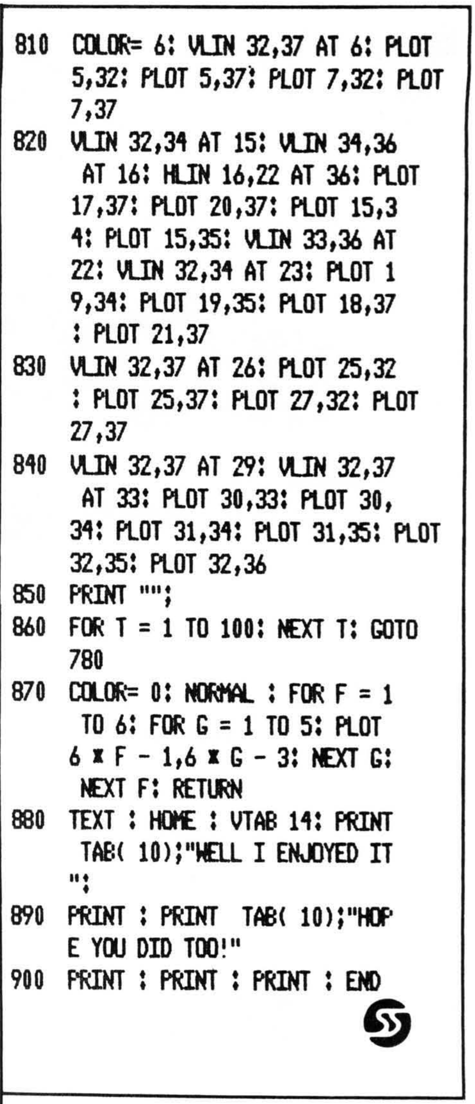
1. For a given TRS-80 (16K, 32K or 48K RAM), the amount of user memory in Model III is 258 bytes less than the amount of Model I.
2. To load a Level II BASIC program, you must select the Low (500 baud) cassette speed on your Model III.
3. When running a Level II BASIC program which requires all-capitals keyboard entries, be sure to select all-caps mode. SHIFT 0 is the on/off toggle for all-caps.
4. Unlike the Model I, Model III lets you interrupt a cassette, line printer, or RS-232-C operation by holding down the BREAK key. Some of your Level II programs may need modification to take this feature into account.
5. The video display character sets are slightly different in Model I and Model III. Model III produces standard ASCII characters for codes 32 through 127; Model I does not. In particular, there is no up arrow, down arrow, left arrow or right arrow in the Model III character set. However, Model III has an additional set of 96 special characters from which you can probably find whatever you need.

Reprinted by permission from the TRS-80 Model III Reference Manual, copyright 1980 Tandy Corporation

GOAL
continued from page 14



```
570  IF G$ = "R" AND C < 6 AND A(
     R,C + 1) = 0 THEN C = C + 1:
     A(R,C) =  - 1: GOTO 490
580  IF G$ = "R" AND C = 6 GOTO 6
     40
590  IF G$ = "R" AND A(R,C + 1) =
      - 1 THEN 660
600  IF G$ = "D" AND R < 5 THEN R
      = R + 1:A(R,C) =  - 1: GOTO
     490
610  IF G$ = "D" AND R = 5 THEN P
     G = PG + 1: GOTO 690
620  IF G$ = "U" THEN  PRINT : PRINT
     : PRINT "   YOU CAN'T GO BAC
     K UP": GOTO 520
630  PRINT : PRINT : PRINT : PRINT
     : PRINT " YOU MUST EITHER US
     E 'L' OR 'R' OR 'D'";: GOTO
     530
640  PRINT : PRINT : PRINT : PRINT
     "  YOU CAN'T MAKE THAT MOVE.
        IT'S OFF": PRINT "THE BOA
     RD."
650  GOTO 520
660  PRINT : PRINT : PRINT : PRINT
     " YOU CAN'T MOVE THERE; THAT
     'S WHERE ": PRINT "THE COUNT
     ER WAS JUST MOVED FROM."
670  GOTO 520
680  REM  PLAYER WINS
690  COLOR= 0: FOR Z = 1 TO 6: PLOT
     6 * Z - 1,27: NEXT Z
700  PRINT : PRINT : PRINT  TAB(
     7)"YOU WIN.  YOU MUST BE LUC
     KY!"
710  PRINT : PRINT "THE SCORE:  C
     OMPUTER - ";CG;"  PLAYER - "
     ;PG
720  PRINT "SPACE BAR TO PLAY  :
     'ESC' TO END";
730  GET C$: IF  ASC (C$) = 32 THEN
     40
740  IF  ASC (C$) = 27 THEN 880
750  GOTO 720
760  GOSUB 870: PRINT : PRINT : PRINT
     : PRINT "THE SCORE:  COMPUTE
     R - ";CG;"  PLAYER - ";PG
770  PRINT : PRINT "SPACE BAR TO
     PLAY : 'ESC' KEY TO END";
780  COLOR= 0: FOR I = 31 TO 38: HLIN
     3,37 AT I: NEXT I: FOR T = 1
      TO 200: NEXT T
790 X =  PEEK ( - 16384): IF X =
     160 THEN  POKE  - 16368,0: GOTO
     40
800  IF X = 155 THEN  POKE  - 163
     68,0: GOTO 880
```



```
810  COLOR= 6: VLIN 32,37 AT 6: PLOT
     5,32: PLOT 5,37: PLOT 7,32: PLOT
     7,37
820  VLIN 32,34 AT 15: VLIN 34,36
      AT 16: HLIN 16,22 AT 36: PLOT
     17,37: PLOT 20,37: PLOT 15,3
     4: PLOT 15,35: VLIN 33,36 AT
     22: VLIN 32,34 AT 23: PLOT 1
     9,34: PLOT 19,35: PLOT 18,37
     : PLOT 21,37
830  VLIN 32,37 AT 26: PLOT 25,32
     : PLOT 25,37: PLOT 27,32: PLOT
     27,37
840  VLIN 32,37 AT 29: VLIN 32,37
      AT 33: PLOT 30,33: PLOT 30,
     34: PLOT 31,34: PLOT 31,35: PLOT
     32,35: PLOT 32,36
850  PRINT "";
860  FOR T = 1 TO 100: NEXT T: GOTO
     780
870  COLOR= 0: NORMAL : FOR F = 1
      TO 6: FOR G = 1 TO 5: PLOT
     6 * F - 1,6 * G - 3: NEXT G:
      NEXT F: RETURN
880  TEXT : HOME : VTAB 14: PRINT
      TAB( 10);"WELL I ENJOYED IT
     ";
890  PRINT : PRINT  TAB( 10);"HOP
     E YOU DID TOO!"
900  PRINT : PRINT : PRINT : END
```

ATARI PROGRAMMING HINT

Normally, when you CSAVE a program, you must press RETURN twice for each copy. If you wish to make more than one copy (which is the prudent thing to do), you must "babysit" the computer, waiting for it to finish each copy, then type CSAVE and press RETURN twice. Here is a short line you can type in when it is time to CSAVE and you would like several copies "just in case":

```
FOR I=1 TO 3: POKE
764,12:CSAVE: NEXT I
```

The computer will now create 3 copies of your program automatically. By POKEing 764 with 12, we have made the computer believe that someone has just pressed RETURN.

James Garon

SYNERGISTIC SOFTWARE

PROGRAM LINE EDITOR

Stronger than Integer! More powerful than Applesoft! Faster than a speeding cursor! Able to leap the longest statements in a single move! The Program Line Editor is the most sophisticated program development tool yet for the Apple II computer. Program Line Editor includes full feature line editors for both Integer and Applesoft programs that render all other techniques obsolete. Program development and modification can now be completed in a tenth the time previously required. The package includes a set of versatile user-definable escape functions which enable you to type commands or sets of commands with only two keystrokes!

The Program Line Editor is available on disk and is compatible with any Apple II or Apple II Plus system with disk drive. **$40.00**

HIGHER TEXT

Higher Text provides the display capabilities you need when the Apple's text output just isn't sufficient. With Higher Text you can print characters in upper case, lower case, upside down, sideways and in any color. If standard characters still aren't adequate, try printing in script, Old English, foreign languages, or special scientific symbology. Define your own specialized or decorative character sets, and easily create professionally customized displays without any hardware modifications.

Higher Text comes on disk and will run on any Apple II or Apple II PLUS system with disk drive. The text generator is in machine language and links easily to the user's application program. **$35.00**

WILDERNESS CAMPAIGN

Free the kingdom of Draconia from the Evil Necromancer! Your party must overcome obstacles, defeat hostile inhabitants, survive natural hazards, explore tombs, temples, castles, and ruins, in search of gold and magical devices. **48K Integer; Cassette $17.50, Disk $19.95**

Both adventures on disk, 48K Integer, $32.50

DUNGEON CAMPAIGN

Direct an expeditionary force into an underground labyrinth. Catacombs filled with treasures and hazards, poisonous vapors, evil sorcerers, stairways, pitfalls and an incredible assortment of MONSTERS. **16K Integer; Cassette $14.95, Disk $17.50**

MAILING LIST DATABASE

Entry and storage of a list of names with associated addresses, cities, states, zip codes, phone numbers, comments, and code designations. List, search, edit, print labels; convert prior mailing lists. Up to 225 records can be loaded on line at any time. **48K AppleSoft, Disk $34.50.**

HIGHER GRAPHICS

A complete set of shape generation, manipulation and utilization tools for the programmer. Quickly and easily, you add sophisticated graphic displays and effects to your programs. Shape Maker, Table Combiner, Screen Creator, 4 shape tables, 3 high-res displays, and the machine language graphics routines required to produce high resolution graphics on the Apple II. **48K, Integer Basic, Disk $24.95.**

To order call TOLL-FREE 1-800-258-1790

(In NH call 673-5144)

WHAT TO DO . . .

continued from page 13

a child is always being told what to do, having a machine that does whatever the child commands is a liberating experience.

The computer builds self confidence, judgement, analytical skills, verbal and aesthetic skills, and even reflexive skills in children. Whether they are flying a space ship with a computer joystick, drawing patterns on the screen, playing a word game, or writing their own programs, they are constantly expanding their abilities and their knowledge. In this respect the computer has the same potential as television, with the advantages that the computer is interactive while the television is passive. In the case of the computer, the owner decides what programming to provide, not the advertising sponsor.

Even if the computer is purchased as a leisure device, it is as easy to justify financially as amateur radio, skiing, boating, and many other hobbies. It does not require good weather, is readily available without notice or travel time, does not require significant operating expenses (at most a few cents an hour for electricity), and offers a wide range of variety.

In a society where information is the most valuable commodity, the computer will soon be indispensable, even in the home. Already, timesharing networks like Compuserve and the Source make available up to the minute information on the stock market, latest news, and developments in many fields. In addition, they offer informational data bases that allow you to search efficiently for information, electronic mail services, and the programming and sorting power of a mainframe computer - all at the call of your home computer.

The intellectual challenge and development of analytical abilities afforded by the personal computer may have a major effect on society. In order to solve a problem with a computer, it is necessary to think through the problem logically. Because people learn to program by starting with small problems and proceed to larger ones, they develop problem solving skills in a systematic way.

The computer has so many potential uses that justifying its purchase is like justifying the purchase of an automobile. You may say that you need a car to get you to work or to get your groceries, but you will end up using it for vacations, going to the theatre, delivering newspapers, or any of a thousand purposes that were not in your original plans. The same thing will happen with a computer. All you need to make the purchase worthwhile is a determination to use the computer long enough to discover what it can do.

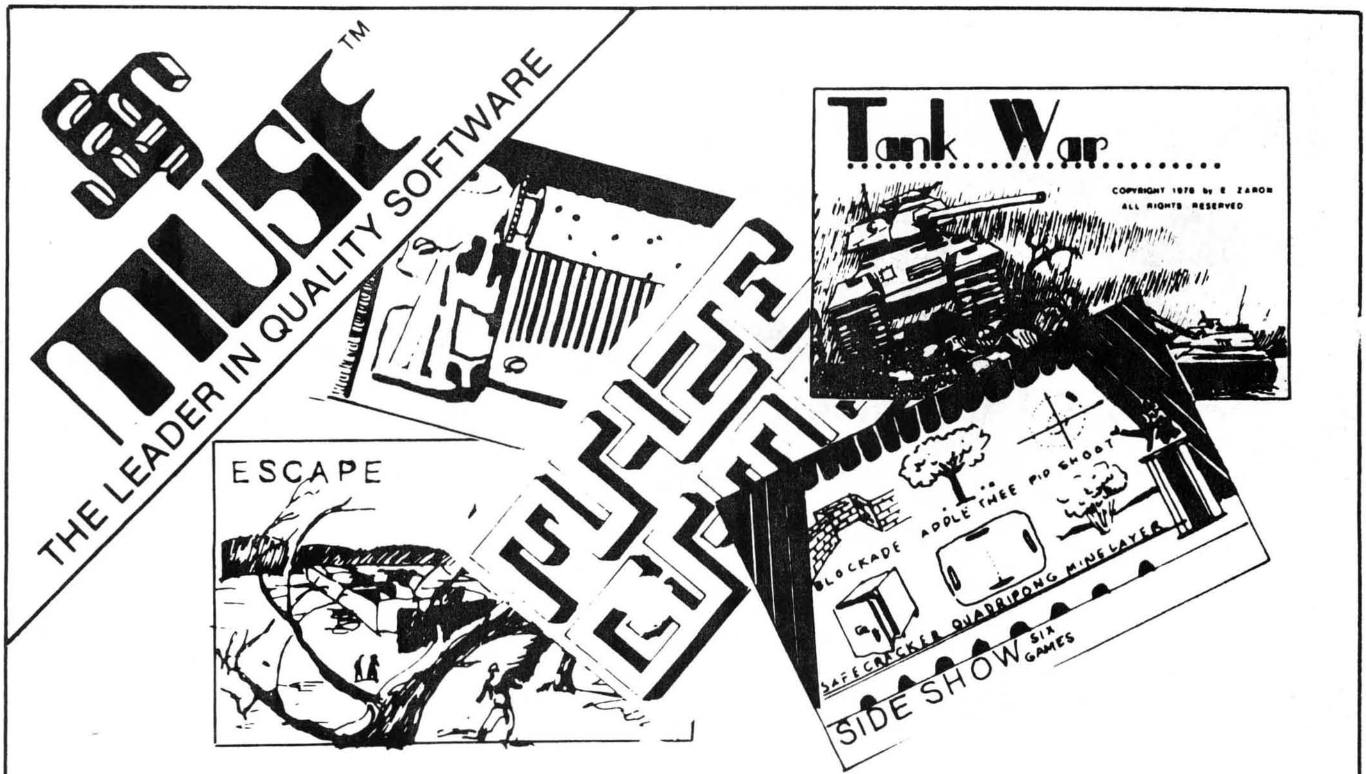
COMPUTERS AND EDUCATION

continued from page 13

Colleges and universities, where occasionally there is a mixture of interest and talent, may offer the best source for the educational market. Equipment is usually more available than in middle and high schools, and marketability is not as important a consideration as with commercial software vendors. Programs that develop at this level don't receive much publicity. They may become known in some circles, but not nationally.

Hopefully some of what's happening at each of these levels will filter its way to us and we can publicize the good programs that are around. But for those who envision sitting down at their computer to take lessons in anything from Spanish to physics to auto maintenance, they'll have to wait a little longer.

"If nothing else, I'm a realist."


MUSE™
THE LEADER IN QUALITY SOFTWARE
Tank War
COPYRIGHT 1978 by E ZARON
ALL RIGHTS RESERVED
ESCAPE
BLOCKADE APPLE TREE PIP SHOOT
SAFECRACKER QUADRIPONG MINELAYER
SIDE SHOW SIX GAMES

THE BEST OF MUSE

A COLLECTION OF OUR FIVE MOST POPULAR CASSETTE PROGRAMS NOW ON DISK

MAZE GAME, the original and still the best maze game for Apple II users. Set the size and difficulty level of your maze and watch as it is randomly generated. Special graphics place you inside a colorful maze, with a blue floor, green and yellow walls, and a pink sky above. Player options include getting a maze map, having a compass available, and leaving retraceable footprints on the maze floor. 16K Integer

ESCAPE is an exciting 3-D adventure with special color graphics that place you inside a maze of hallways. Find your way out by following the clues of people you meet inside. But be careful . . . the liars will not hesitate to give you false maps and compasses. It's hard to quit this game! Rated as the number 1 Apple program by the Mårin Computer Center. 16K Integer.

SIDE SHOWS is a collection of six exciting games selected for their originality, ease of operation and imaginative use of the Apple. Includes the Apple Tree, Safecracker, Pip Shoot, Minelayer, Blockade and Quadripong. In the Apple Tree two players use the paddles to catch randomly falling apples. In Blockade and Minelayer try to surround your opponent with a moving wall or mine field. Pip Shoot can be played by two or more and the object is to shoot down as many "pips" as possible. Quadripong is four wall pong with varying ball speed and sound. Use the paddle carefully to be the first to open the safe in Safecracker. 16K Integer.

TANK WAR is a state-of-the-art shootout for two players with dramatic sound and graphics. Players choose the style and size of their tank. Steer your tank with the game paddles to pursue and outmaneuver your opponent. Fire two types of missiles to disable and destroy the enemy. 16K Machine Language.

THE MUSIC BOX - start the MUSIC BOX and your Apple comes alive with sound and color. Play it as a three-octave instrument or type in a song and the music box will play it for you! A brilliant full color light show accompanies the music. Standard musical notation includes sharps and flats, note time, rests, dotted notes and tempo. Your songs can be saved on disk and recalled later. All notes can be easily retuned from the keyboard to produce special sound effects. Complete with two songs and documentation that explains basic musical notation. 16K Machine Language.

All on one disk. A $64.75 value for only $39.95

TO ORDER
TOLL FREE
1-800-258-1790
(In NH call 673-5144)

6 South St.,
Milford, NH 03055

The above programs are also available on cassette for $12.95 each.

The BONE FONE is a miniaturized AM/FM stereo system that drapes over your shoulders like a scarf. The BONE FONE gives new meaning to the word portable. You don't carry a BONE FONE around with you. You wear it. Ski with it, cycle, jog or just take a leisurely stroll through the woods with it. It's the hands-free stereo that brings beautiful music to every indoor or outdoor activity without disturbing those around you and without covering your ears.

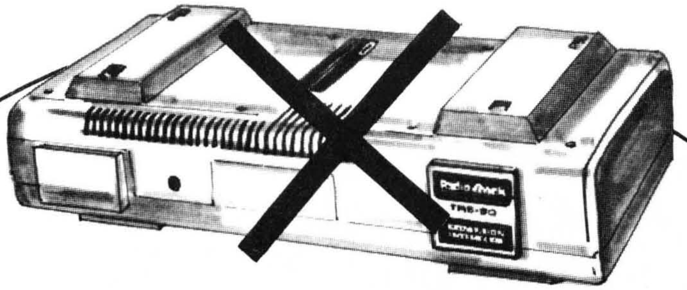
The BONE FONE stereo is covered with a sleeve made of Lycra Spandex - the same material used to make expensive swim suits, so it's easily washable. You simply remove the sleeve, dip it in soapy water, rinse and let dry. The entire system is also protected against damage from moisture and sweat making it ideal for jogging or bicycling.

The BONE FONE has been selling nation-wide for $70. HARDSIDE's special price is ONLY $56.95 plus $2.50 handling (we pay the UPS charges).

9902 . . . . $56.95

Bone Fone

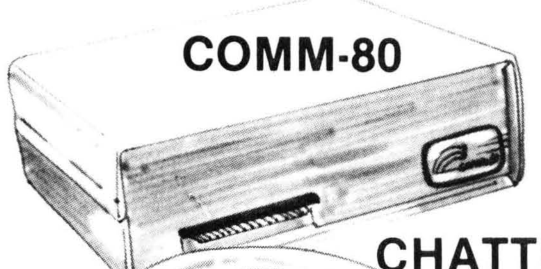
TO ORDER
TOLL FREE
1-800-258-1790
(In NH call 673-5144)



TRS-80* interfacing alternatives...

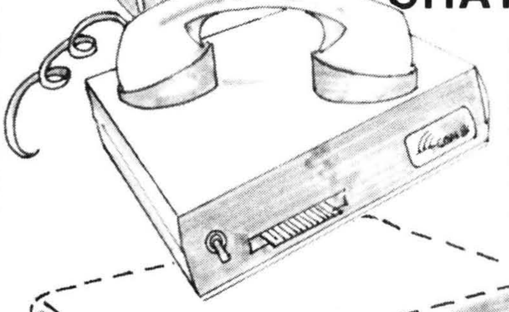
Featuring the Quality Engineering and Manufacturing of Steve Ciarcia and Micromint

COMM-80



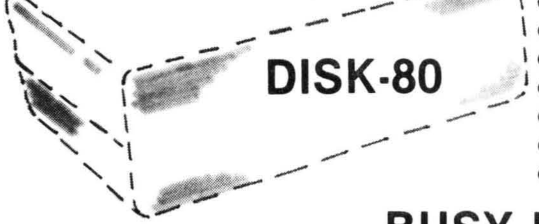
- FULL 8-BIT PARALLEL PORT
- RS-232-C PORT (up to 19,200 baud)
- UP TO 16 UNITS CAN BE CHAINED TOGETHER
- TRS-BUS CONNECTOR FOR FUTURE EXPANSION
- CONNECTS TO KEYBOARD OR E.I.
- ONLY $179.95

CHATTER BOX



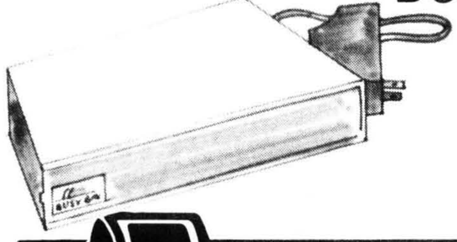
- FULL 8-BIT PARALLEL PORT
- RS-232-C PORT (up to 19,200 baud)
- ACOUSTIC MODEM
- TRS-BUS CONNECTOR FOR FUTURE EXPANSION
- CONNECTS TO KEYBOARD OR E.I.
- INCLUDES TERMINAL SOFTWARE
- ONLY $259.95

DISK-80



- DISK CONTROLLER (up to 4 drives)
- DATA SEPARATOR
- INCLUDES 16K OF RAM
- PROVISION FOR AN ADDITIONAL 16K RAM
- TRS-BUS CONNECTOR FOR FUTURE EXPANSION
- Available September 1980
- ONLY $299.95

BUSY BOX



- 8 1/4" x 6 2/5" PLASTIC CASE
- CABLE AND CONNECTOR FOR TRS-80 KEYBOARD E.I.
- OPERATING MANUAL
- INCLUDES BASIC LISTING FOR CONTROL ROUTINE
- LEVEL II MINUMUM SYSTEM
- ONLY $109.95

*TRS-80 is a trademark of Tandy Corp.

HARDSIDE™ 6 SOUTH ST., MILFORD, NH 03055

To order: Call Toll-Free 1-800-258-1790 (in NH call (603) 673-5144)

VISA

The Software Exchange & HardSide (Div. of Robitaille & Sons, Enterprises, Inc.), SoftSide Publications

"our price vs. list price" compare!

TRS-80

	List Price	Our Price
Level-II, 4K	$649.00	$599.00
Level-II 16K no keypad	$768.00	$669.00
Level-II 16K with keypad	$849.00	$749.00
COMM-80 Interface		$179.95
CHATTER BOX Interface		$259.95
Disk-80 Interface		$299.95
Expansion Interface, no RAM	$299.00	$279.00
Expansion Interface 16K RAM	$418.00	$369.00
Expansion Interface 32K RAM	$537.00	$459.00
RS-232-C Board	$99.00	$89.00
TRS-232 Printer Interface	$59.95	$49.95
16K Memory Kit, TRS-Keyboard	$119 00	$89.00
16K Memory Kit, TRS-Exp. Int	$119.00	$85.00
Percom, TFD-100, 40-track	$399.95	$389.00
Percom, Dual TFD-100 Drives	$795.00	$775.00
Percom, TFD-200, 77-track	$675.00	$599.00
Percom Data Separator		$29.95
Percom Extender Card	$15.95	$15.00
2-Drive Cable	$29.95	$29.00
4-Drive Cable	$39.95	$39.00
Data Dubber	$49.95	$44.95
Percom Electric Crayon w/cbl		$279.95
Busy Box, TRS-80	$109.95	$99.95
BSR X-10 Starter Kit	$124.95	$99.95
Busy Box 4-yr Control Program		$19.95

ATARI

	List Price	Our Price
ATARI 400 Comp. System, 8K	$629.95	$499.00
ATARI 800 Comp. System 16K	$1079.95	$819.00
ATARI 410 Program Recorder	$89.95	$59.00
ATARI 810 Disk Drive	$699.95	$569.00
ATARI 815 Dual Density Disks	$1495.95	$1219.00
ATARI 820 Printer (40 col.)	$599.95	$469.00
ATARI 822 Thermal Printer	$449.95	$369.00
ATARI 825 Printer	$999.95	$819.00
ATARI 830 Acoustic Modem	$199.95	$169.00
ATARI 850 Interface Module	$219.95	$179.00
ATARI Light Pen	$74.95	$59.00
8K RAM Memory Module	$124.95	$99.00
16K RAM Memory Module	$199.95	$159.00
ATARI Joystick Controllers	$19.95	$18.95
ATARI Paddle Controllers	$19.95	$18.95

CARRYING CASES

	List Price	Our Price
TRS-80 Computer Case	$109.00	$99.00
TRS-80 Monitor Case	$84.00	$79.00
APPLE System Case w/1 drive	$109.00	$99.00
APPLE System Case w/2 drive	$119.00	$109.00
APPLE System Case w/mon.	$129.00	$119.00

APPLE

	List Price	Our Price
Mtn. Hard. Apple Clock	$280.00	$262.50
Mtn. Hard. ROMPLUS+ w/Fil	$200.00	$187.50
16K Memory Kit	$119.00	$85.00
Busy Box, Apple	$114.95	$99.95
BSR X-10 Starter Kit	$124.95	$99.95

PET/CBM

	List Price	Our Price
PET 2001 large keyboard, 16K	$995.00	$895.00
PET 2001 large keyboard, 32K	$1295.00	$1195.00
CBM 8032 Business Comp.	$1795.00	$1619.00
CBM Cassette Recorder	$95.00	$84.95
CBM 2040 Dual Disk Drives	$1295.00	$1195.00
CBM 8050 Dual Disk Drives	$1695.00	$1495.00
CBM Modem	$395.00	$349.00
CBM Voice Synthesizer	$395.00	$349.00
CBM to IEEE Cable	$40.00	$37.50
IEEE to IEEE Cable	$40.00	$37.50

PRINTERS

	List Price	Our Price
Centronics 730	$795.00	$699.00
Centronics 737	$995.00	$875.00
Centronics 779	$1559.00	$1095.00
Centronics 779 w/lower case & motor control	$1779.00	$1295.00
NEC 5510 SpinWriter	$3195.00	$2595.00
NEC 5520 SpinWriter	$3395.00	$2895.00
NEC 5530 SpinWriter	$3195.00	$2495.00
NEC Tractor-Feed Option	$225.00	$195.00
LRC 7000+ (64-col.)	$405.00	$349.00
LRC 7000+ (40-col.)	$389.00	$339.00
Okidata Microline-80	$800.00	$709.00
Tractor-Feed Option	$140.00	$129.00
RS-232-C (2K) Option	$299.00	$279.00
LRC to TRS-80 or APPLE		$20.00
LRC to PET, IEEE		$59.00
LRC to RS232C, male/female		$65.00
730 or 737 to TRS-80		$29.00
NEC or 779 to TRS-80		$35.00
RS-232-C/RS-232-C male/male		$24.95

DUST COVERS

TRS-80 (3pc set)	$7.95
APPLE	$7.95
ATARI 800	$7.95
ATARI 400	$7.95
PET/CBM 2001	$11.95

*TRS-80, ATARI, APPLE, and PET/CBM are trademarks of Tandy Corp, Warner Communications, Apple Computer Co., and Commodore respectively.

HARDSIDE 6 SOUTH ST., MILFORD, NH 03055

To order: Call Toll-Free 1-800-258-1790 (in NH call (603) 673-5144)

VISA

The Software Exchange & HardSide (Div. of Robitaille & Sons Enterprises Inc.) SoftSide Publications

The Software Exchange Order Form

DESCRIPTION	MEMORY	TYPE	PRICE

SPECIFY: TRS-80, APPLE, ATARI or PET. Do not use for hardware.

Check

Money Order

master charge OR VISA

ADD handling charges	**$1.00**
Additional charges	
TOTAL	

CHARGE ACCOUNT NUMBER

Signature____________________ Exp. Date__________ Inter. No.__________

Name______________________________

Address______________________________

City____________________ State__________ Zip__________

MAIL TO: The Software Exchange
6 South St.
Milford, NH 03055
(In NH call 673-5144)

or, CALL TOLL FREE
for Orders Only
1-800-258-1790

Special prices in effect 60 days from mailing

Level II software available on disk for a $5.00 (per order) medium charge. This extra fee is for any number of programs transferred to disk from tape when you order. If the order exceeds the capacity of a single disk, we absorb the extra cost.

Be sure to include handling charge and any additional charges when figuring your total. All orders are shipped within 48 hours.

All prices are subject to change without notice. We are not responsible for typographical errors, including incorrect prices.

ALL SOFTWARE GUARANTEED TO LOAD AND RUN. If you experience difficulties, simply return the tape or disk properly protected for free replacement. Send to the attention of Bette Keenan, Customer Service Representative: please enclose a brief note and your name and mailing address with the software.

ALL SOFTWARE SOLD ON AN AS-IS BASIS WITHOUT WARRANTY. TSE assumes no liability for loss or damage caused or alleged to be caused directly or indirectly by equipment or products sold or exchanged by them or their distributors, including but not limited to any interruption in service, loss of business or anticipatory profits or consequential damages resulting from use or operation of such equipment or software.

COMING NEXT MONTH

IMHOTEP - a role playing simulation
WORLD SERIES
MOON LANDING and EARTH PORT II
OCTADRAW
ESCAPE - a maze with 3-dimensional graphics

PLUS:

A new adventure column by Scott Adams, the next installment of SoftSide's Data Base, an in depth look at the inner workings of the "Dr. Livingston . . ." adventure, AND MORE!

RAMWARE™

software you can rely on.

PERSONAL/BUSINESS

SMALL BUSINESS BOOKKEEPING (S-80)
Roger Robitaille Sr.
16K Level II $24.95
Extended version 32K disk
Without Journal $29.95

PERSONAL FINANCE
Lance Micklus
Level II, 16K $9.95

GAMES

X-WING II (S-80)
Chris Freund
Level II, 16K $9.95

INVASION (S-80)
Chris Freund
Level II, 16K $9.95
32K Disk $14.95

SPACE BATTLES (S-80)
Level II, 16K tape $14.95
Level II, 32K disk $19.95

SNAKE EGGS (S-80)
Level II, 16K $14.95

LIFE TWO (S-80)
Level II, 16K $14.95

BEE WARY (S-80)
Level II 16K $14.95

ANDROID NIM (S-80)
Level II, 16K $14.95

AMAZIN' MAZES (S-80)
Robert Wallace
Level II, 16K $9.95

STAR TREK III.5 (S-80)
Lance Micklus
Level II 16K $14.95

UP PERISCOPE (S-80)
Ron Potkin
Level II 16K $14.95

WARPATH (S-80)
Ron Potkin
Level II 16K $14.95

MEAN CHECKERS MACHINE (S-80)
Lance Micklus
Level II, 16K .... $ 9.95
Disk $14.95

KRIEGSPIEL 2 (S-80)
Ron Potkin
Level II, 16K (cassette only) $14.95

EDUCATIONAL GAMES

9 GAMES FOR PRE-SCHOOL CHILDREN
George Blank
Level II, 16K $9.95

UTILITIES

UTILITY (S-80)
Nepenthe Publications
32K and 48K disk systems $29.95

SUPER SIMON (S-80)
George Blank
Level II, 16K (disk or cass) $9.95

STAD (Symbolic Trace and Debug) (S-80)
Paul van der Eijk
16K, 32K and 48K versions
on one cassette $24.95

DYNAMIC DATA BASE (S-80)
Ken Knecht
Disk 32K $39.95

MICRO TEXT EDITOR (S-80)
Don Coons
Level II 4K $9.95

FLOPPY DISK DIAGNOSTIC (S-80)
Dave Stambaugh
16K to 48K, 1 to 4 disk drives $24.95

Z80 ZAP (S-80)
ORG-TEX Industries
16K Disk $29.95

TINY COMP (S-80)
Dave Bohlke
Tape version $19.95
Disk version $24.95

RPN CALCULATOR (S-80)
Russell Starkey
Level II 16K $9.95

KEYBOARD 80 (S-80)
John Adamson
Level II, 16K $9.95

SPOOLER (S-80)
Miles Smither
Disk $24.95

KVP (S-80)
Lance Micklus
16K Level II Tape $14.95

INVENTORY 'S' (S-80)
Roger Robitaille
Level II 16K $24.95
32K Disk $59.95

ST80D (S-80)
Lance Micklus
32K Disk systems $79.95

ST-80 UC (S-80)
4K Level II Cassette $24.95

BASIC CROSS REFERENCE
RX-X-REF-RENUMX
Lance Micklus
RX (Disk, 32K) 1 disk system $24.95
XREF (Level II version
16K on Cassette $19.95
RENUMX (XREF w/renumber and rescue)
Level II on cassette $24.95
RENUMBER (separately) $7.95

LANGUAGES

APL-80 VERSION 3.0 (S-80)
Phelps Gates
32K single disk $39.95
16K Level II cassette $14.95

ADVENTURES

ADVENTURES ON TAPE
Scot Adams
Count, Pirates Cove, Fun House, Pyramid of Doom, Strange Odyssey, Ghost Town, Mission Impossible, Land, VooDoo ... $14.95

PATHWAYS THRU THE ROM
Blank, Fuller, Hartford,
Phillipp and Richardson .... $19.95 + $1

3D (Apple 2)
Mark Pelczarski
A Rom Disk 48K $29.95

MAGIC PAINTBRUSH
Mark Pelczarski
Disk 32K $29.95

ANDROID NIM
Disk/24K $17.95

Join Our SoftSide Celebration!

We're Bigger and Better than Ever

Special Offer to New Subscribers:

SoftSide - $15 for one year (12 issues)

Don't miss a single issue of SoftSide - the finest software source for the Radio Shack TRS-80*, APPLE II*, and ATARI *!

Save hours of typing – buy SoftSide's programs for your computer already on cassette or diskette, with magazine. Try our low introductory price for one year (12 issues).

To order, call TOLL FREE **1-800-258-1790** (in NH call 673-5144) or mail the following information to: SoftSide Publications, Subscriptions, 6 South Street, Milford, NH 03055.

□ SoftSide: $15., one yr.,(12 issues)

□ SoftSide: Apple on cassette $69.

□ SoftSide: Apple on diskette $129.

□ SoftSide: S-80 on cassette $69.

□ SoftSide: S-80 on diskette $129.

SoftSide: Atari on cassette and diskette will be available soon!

I own:
□ Apple
□ TRS-80
□ Atari
□ Pet
□ Other ______

MasterCard VISA

□□□□□□□□□□□□□□□

Exp. Date ______ Interbank #(M/C only) ______

Signature ______

Name ______

Address ______

______ State ______ Zip ______

Subscribe to SoftSide today!

Introductory offer valid until November 1, 1980. Not applicable to overseas subscribers. Write for subscriptions rates.

*TRS-80, Apple and Atari are trademarks of Tandy, Apple Computer Co., and Warner Communications, respectively.