

JUNE 1981

# SoftSide™

TWO DOLLARS  
AND FIFTY CENTS

Your BASIC Software Magazine • VOLUME IV • NUMBER NINE

OLD GLORY

Catacombs  
of the Phantoms

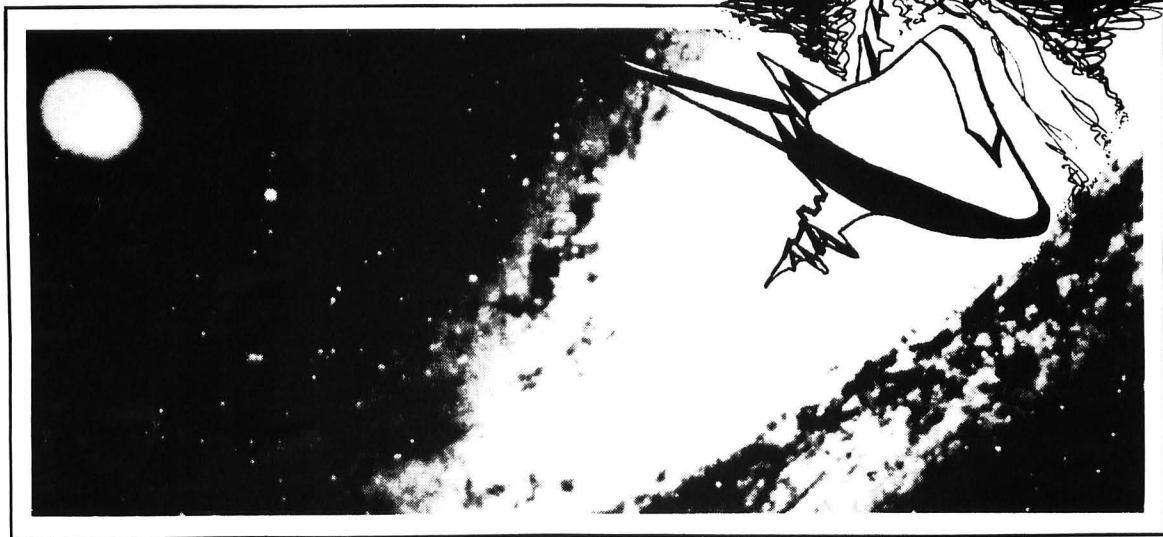
Anallist

Math Decathlon

Hex Puzzle

and much much  
more...

**Hot pursuit  
through space  
and the  
vortices  
of time!**



RAINYWARE PRESENTS...

# Time Lord

The fallen Time Lord, who presumptuously calls himself The Master, is at large. The elders of Waldrom have supplied you with the hyperspace-worthy vessel TARDIS, and commissioned you to eliminate the evil "Master". Your resources include clones who will fight for you, the formidable CRASER weapons of the TARDIS, and magic weapons such as Fusion Grenades and Borelian Matrix Crystals.

Travelling through hyperspace in search of the evil one, you will encounter Time Eaters, Neutron Storms, and other alien creatures and phenomena. Entering real space to search planets, you will encounter still other dangers. You will enter native settlements to buy food and supplies — or to fight for survival.

And once you find The Master can you destroy him?

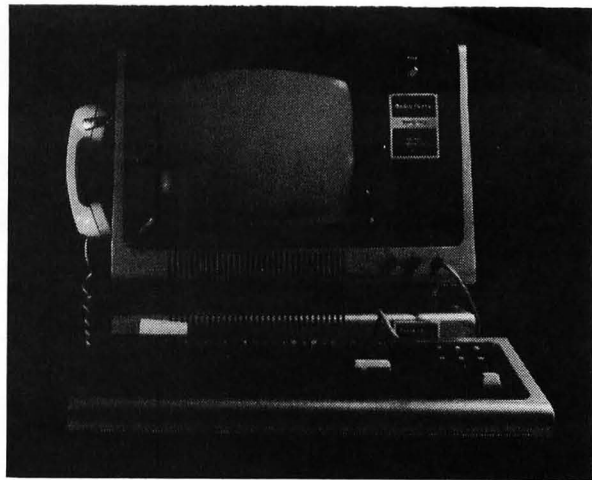
  
**TSE-HARDSIDE**  
6 South St., Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790

Based on Dr. Who of PBS fame.  
Apple Integer Basic,  
Disk, 48K ... \$29.95

#47-191001D



# EXPAND YOUR COMPUTING CAPABILITIES!



## THE CHATTERBOX

All of the-features of the Comm-80 Interface plus a built-in acoustic modem for 0-300 baud operation. Includes Full/Half Duplex terminal software. Unit is covered by a 90-day guarantee from MICROMINT. #4-81 (5 lbs.) .....\$239.00

## ST80\* SMART TERMINAL PACKAGE

All four programs include the ability to use an unmodified TRS-80™ keyboard to produce RUB, ESC, and other control characters for time sharing, software control of the RS-232C board, repeat key, bell, software support for the three most common upper/lower case hardware conversion, and line printer output.

### ST80\* UC

Preset parity, word length, and baud rate (regardless of switch settings on the RS-232C board) for THE SOURCE, MICRONET, and FORUM 80, automatic testing of the RS-232C board, and even spooling of prepared messages on tape directly into FORUM 80 using a BASIC program supplied as a line listing. 4K Mod I/Mod III cassette ..... #26-ST8S .. \$24.95

### ST80\*

Reprogram your RS-232-C board from the keyboard, and run at different baud rates. Note: does not have auto testing of the RS-232-C or tape spooling. 4K Level II Cassette .... #26-ST80 ..... \$49.95

### ST80\* D

Connection time clock, option of user-created translation tables for keyboard, gathering and pre-formatting data to be sent directly from disk to host computer, spooling of received files to disk or printer, editing of received files, and auto logon. Use it with VTOS 3.1, and get device-driven I/O, job logging, and chaining. 32K disk program. .... #26-ST8D ..... \$79.95

### ST80\* III

Represents the "state of the art" in communications processors and was designed for complex communications processing. Included in this package is a set of programs which allows your Model I or Model III to talk to a time-sharing computer, transfer files to and from the central computer, and customize your ST80-III program to suit your specific time-sharing environment. Altogether you receive seven programs with this system. 32K disk program ..... #26-ST83 ..... \$150.00

## THE SOURCE

Would you like to send mail across the country overnight, write one letter and send it to a hundred people, read the news right off the wire service, program in COBOL, FORTRAN, or RPG II, advertise on a nationwide electronic bulletin board, join a real time national computer club, leave messages when you are out of town, make travel arrangements through your computer, use a discount shopping service, reach a whole library of specialized information on business, real estate, science and engineering, and personal applications, find out what the weather is around the country, and use dozens of high-powered applications programs?

Connect your microcomputer to THE SOURCE! THE SOURCE is a bank of mainframe computers in McLean, Virginia connected by TYMNET and TELEMET data transmission services to hundreds of cities around the country. Most urban areas can access THE SOURCE by local phone call. Costs:

One time subscription fee (includes manuals) .....	\$100.00
Connect time after midnight (per hour) .....	\$2.75
Connect time between 6 p.m. and midnight (per hour) .....	\$4.25
Connect time 8 a.m. to 6 p.m. (per hour) .....	\$15.00

**TSE-HARDSIDE**  
 6 South St. Milford, NH 03055 (603) 673-5144  
 TOLL FREE OUT-OF-STATE 1-800-258-1790

\*ST80 is a trademark of Lance Micklus, Inc. TRS-80 is a trademark of Tandy Corp.

# Settle for More from Your TRS-80

**BASIC Compiler.** With TRS-80 BASIC Compiler, your Level II BASIC programs will run at record speeds! Compiled programs execute an average of 3-10 times faster than programs run under Level II. Make extensive use of integer operations, and get speeds 20-30 times faster than the interpreter.

Best of all, BASIC Compiler does it with BASIC, the language you already know. By compiling the same source code that your current BASIC interprets, BASIC Compiler adds speed with a minimum of effort.

And you get more BASIC features to program with, since features of Microsoft's Version 5.0 BASIC Interpreter are included in the package. Features like the WHILE . . . WEND statement, long variable names, variable length records, and the CALL statement make programming easier. An exclusive BASIC Compiler feature lets you call FORTRAN and machine language subroutines much more easily than in Level II.

Simply type in and debug your program as usual, using the BASIC interpreter. Then enter a command line telling the computer what to compile and what options to use.

Voila! Highly optimized, Z-80 machine code that your computer executes in a flash! Run it now or save it for later. Your compiled program can be saved on disk for direct execution every time.

Want to market your programs? Compiled versions are ideal for distribution.\* You distribute only the object code, not the source, so your genius stays fully protected.

BASIC Compiler runs on your TRS-80 Model I with 48K and disk drive. The package includes BASIC Compiler, linking loader and BASIC library with complete documentation. \$195.00.

\*Microsoft royalty information for the sale of programs compiled with BASIC Compiler is available from Microsoft.

## muMATH Symbolic Math System

expands your TRS-80 beyond the limits of numerical evaluation to a much higher level of math sophistication.

Symbolic mathematics is muMATH's power. For the first time, algebra, trigonometry, calculus, integration, differentiation and more can be performed on a system smaller than an IBM 370. And in a fraction of the time you could do them manually.

Yet for all its power, muMATH is simple to use.

To perform a differentiation you could enter: ?DIF (A \* X ↑ 3 + SIN(X ↑ 2), X);

In almost no time, the computer would reply with: @2 \* X \* COS(X ↑ 2) + 3 \* A \* X ↑ 2.

Or to add fractions: ?1/3 + 5/6 + 2/5 + 3/7;

The instantaneous answer: 419/210.

Or to perform a more difficult trigonometric expansion you enter: SIN(2 \* Y) \* (4 \* COS(X) ↑ 3 - COS(3 \* X) + SIN(Y) \* (COS(X + Y + #PI) - COS(X - Y)));

Just a few seconds later, the computer replies: @4 \* SIN(Y) \* COS(X) \* COS(Y).

muMATH has virtually infinite precision with full accuracy up to 611 digits.

If you use math, you'll find countless ways to save time and effort with muMATH. It's a professional tool for engineers and scientists. A learning tool for students at any level from algebra to calculus.

And if you want to expand your capabilities even beyond the standard muMATH, the option is open. muSIMP, the programming language in which muMATH is written, is included in the muMATH package. A superset of the language LISP, muSIMP is designed especially for interactive symbolic mathematics and other artificial intelligence applications.

muMATH and muSIMP were written by The Soft Warehouse, Honolulu, Hawaii. Priced at \$74.95, the package includes muMATH, muSIMP and a complete manual. It requires a Model I TRS-80 with 32K and single disk. muMATH for the Apple II Computer will be available later this year.



FROM

**MICROSOFT**  
CONSUMER PRODUCTS



**TSE-HARDSIDE**

6 South St., Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790



# TABLE OF CONTENTS

## ARTICLES

- 11 THE RACE OF THE SORTS    A Showdown of sorts ..... S-80. .... Rik Karlsson
- 30 COMPUTER GRAPHICS    More Pattern Generation ..... Apple. .... Joan Truckenbrod
- 47 CUSTOM TAPES    Waste not, want not ..... Ed Ting
- 52 LEMONADE OR CHAMPAGNE    Will Hagenbuch's book in serial form ..... Will Hagenbuch
- 58 ALL RAMS ARE NOT CREATED EQUAL    Tribulations in Chip City ..... Joesph V. Cesaitis
- 78 REVIEWS    Lords of Karma; Poker Tournament; Thrilogy of Games; Micro-Painter ..... Dave Albert, Jon Voskuil

## S-80, APPLE, AND ATARI PROGRAMS

- 48 OLD GLORY    For all you flag lovers ..... William Morris & John Cope
- 92 WORD-SEARCH PUZZLE GENERATOR    Befuddle your friends ..... David Durkee, Jon Voskuil

## APPLE PROGRAMS

- 20 MATH DECATHLON    The final installment ..... Jon Voskuil
- 26 HEXADECIMAL PUZZLE    A new twist to an old puzzle ..... George W. Ziegler
- 28 MAZE SEARCH    Translated from the Atari ..... Carl Mueller, Dave Bohke
- 32 MUSI-KEYS INPUT ROUTINE    Let your keys talk back to you ..... Jon Voskuil

## ATARI PROGRAMS

- 34 VOLCANO    Mt. St. Helens erupts yet another time ..... Art Cestaro, Victor T. Albino
- 38 CATACOMBS OF THE PHANTOMS    Explore the tunnels beneath Petiteville ..... Tom Plassman

## S-80 PROGRAMS

- 62 BATS    The Electronic Vampire Nightmare Game ..... John Baker, D.V.M.
- 66 DIVIDE AND CONQUER    Are they aliens or amoebae? ..... Phillip C. Soine
- 74 KRAZY TALK    Are you creatively articulate? ..... David Gash
- 90 ANALLIST    A listing utility ..... Rod Fitzgerald

## DEPARTMENTS

- 4 EDITORIAL ..... Jon Voskuil
- 5 ABOUT THIS ISSUE ..... Munchkins Galore
- 6 INPUT ..... From our readers
- 7 OUTGOING MAIL ..... Dave Albert
- 25 CALENDAR ..... Editors
- 42 SAY YOHO ..... Scott Adams
- 46 WORD-SEARCH PUZZLE #2 SOLUTION ..... Editors
- 57 WHAT'S NEW ..... Ed Umlor
- 77 BUGS, WORMS & OTHER UNDESIRABLES ..... Kay Pasa
- 81 HARDWARE CORNER ..... Ed Umlor
- 82 THE SENSUOUS PROGRAMMER ..... "J"
- 96 WORD-SEARCH PUZZLE #3 ..... Editors

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, and Pet are registered trademarks of Tandy Corporation, Apple Computer Company, Warner Communications, and Commodore Business Machines.

# STAFF

## PUBLISHER:

Roger Robitaille Sr.

## EDITORIAL DEPARTMENT

Scott Adams  
Dave Albert  
Rich Bouchard  
Sandy Dean  
Steve Justus  
Mary Locke  
Lance Micklus  
Mark Pelczarski  
Joan Truckenbrod  
Jon Voskuil  
Joan Witham

## PRODUCTION DEPARTMENT

Donna Bennett  
Cindy Boucher  
Lynda Fedas  
Damian Henriques

## STAFF

Patricia Acampora  
Lester Anderson  
Ruth Anderson  
Judie Aveni  
Brian Berkebile  
Diana Bishop, **Subscriptions**  
Kathleen Boucher  
Suzanne Breton  
Philip Brown  
Jeff Carroll  
Brenda Cookingham  
Donna Cookingham  
Joanne Cummings  
Pam Demmons  
Mary Edwards, **Software Information**  
Mark Eric, **Atari Submissions**  
Anthony Fraser  
Mary George  
William F. Gollan, **Advertising**  
Mylene Grigas  
Kathleen Hannon  
Pamela Horne  
Robert Hunsacker  
Dave Hutchings  
Donna Jean  
Bette Keenan, **Customer Service**  
Bea Kimball, **Software Shipments**  
Randal Kottwitz  
Karen Lawrence  
Kathy Maloof  
Jean Matthews  
Dick Mehlhorn, **Hardware Information/Shipments**  
Doris Miller  
Clem Morey, **Apple Submissions**  
Robin Moss  
Glen Ohlund  
Mary Reed, **Dealer Information**  
Carol Roane  
David Robitaille  
Elizabeth Robitaille, **Personnel Administrator**  
Cindy Schalk  
Ken Sicard, **S-80 Submissions**  
Otto Snow  
Christine Spade  
Alan Thulander  
Joanne Tracy  
Anmar William  
Nancy Wood  
Ed Umlor  
Gary Young  
Cynthia Zawacki

SoftSide is published each month by SoftSide Publication, 6 South Street, Milford, New Hampshire 03055. Telephone 603-673-5144. Controlled circulation postage paid, Milford, New Hampshire 03055 and additional entries. ISSN: 0274-8630. Application to mail at controlled circulation postage rate is pending at Concord, NH 03301. Subscription rates: USA \$24.00 per year. USA First Class, APO, FPO, Canada, Mexico, Overseas surface mail -\$32.00 per year. Overseas air mail \$48.00 per year. All remittance must in U.S. funds. Mail subscription inquiries to SoftSide Publications, P.O. Box 68, Milford, New Hampshire, 03055. Entire contents copyright 1981. SoftSide Publications. All rights reserved.

## POSTMASTER:

Send address changes to:  
SoftSide Publications  
6 South Street  
Milford, New Hampshire 03055



# EDITORIAL

by Jon Voskuil

Whenever someone asks to "see my computer" for the first time, I'm always somewhat at a loss for what to do. Oh, it's easy to show it to kids, because they can so easily get caught up in any number of games. But it's much harder to "show" your computer to most adults. They're generally too self-conscious to throw themselves into computer games, afraid of making fools of themselves by getting clobbered in front of their friends. Compared to younger folk, they're also more sophisticated, more pragmatic, and more suspicious of machines. Especially machines of the type that send them overdue bill notices and make mistakes on their charge accounts.

Maybe the most difficult thing about showing one's computer to most post-adolescents is that you not only have to "show" it but also to justify its existence (and the small fortune that you obviously must have spend on it). You can show it by exposing its innards and connecting its cables and booting its disks and running all its neat programs — but how do you justify it to a skeptic, who sees it as a whimsical, materialistic indulgence of the first order, with no redeeming social value?

The things about your computer that you can show someone in half an hour really don't justify its existence. Unless, for example, you're a writer and use it as a text processor — that's something "practical"! But I don't think that all the games in **SoftSide's** (or your) library justify owning a home computer. What does justify one, even for a person like me (and probably you) who has no practical use for one, is that the home computer is able to function as a mind-extension tool. And that's something that's very nearly impossible to "show" to anyone else, especially a skeptic.

This isn't to say that everyone should extend their mind via a computer, or that a maze of silicon, PC boards, and plastic is the only way (or even the best way) to expand the horizons of your thought processes. But it's one way, and a good one at that. I occasionally have anti-rational and anti-technological fits that leave me disillusioned with mind and

machine, but most of the time I think that rational intellect is good and that technology (while not inherently good) has a lot of beneficial uses — when we're able to control it, rather than vice-versa.

My wife loves to tell the story of a discussion we had a few years ago with two other couples, our very closest friends. In it we were all sharing some of our gut-level dreams and visions for our lives. And amidst the noble and altruistic aspirations of the other five people, I confessed in a paroxysm of honesty that one of my lifelong dreams was to live in a house with every conceivable electronic gadget built into it.

Now, admittedly, that's a pretty materialistic goal for one's life! And I knew then, as I know now, that people are of a whole different order of importance and value than things, and that all the electronic marvels in the world can't add up to real happiness. (Usually all they add up to is a lot of repair bills.) So much for the homily. But there is a certain kind of (very valuable) intellectual exercise that complex machines like computers can provide. They foist upon you a relentless rationality which you cannot escape (short of pulling the plug), and to which you must respond in kind if you want to accomplish anything.

That kind of restriction simply doesn't apply in our everyday contacts with people, where nonrational and even nonverbal factors are often more significant than the purely rational and intellectual. There's something rather comfortable and also stimulating about working from time to time within that verbal/rational boundary which computers prescribe. As long as it doesn't become a substitute for, or escape from, the full depth of human contact, it's not only fun but beneficial. The hardware and firmware and software that make up computers were all designed by people as expressions of their rational intellect, and the impersonal machine between that designer and me allows our minds to meet in a unique way. I can't show you that meeting of minds in half an hour, but that's why I own that computer.





# ABOUT THIS ISSUE

Howdy, howdy folks, it's us again, Munchkin City ready to bend your eyeballs for a page or so. First a word from Uncle Fred...

Boy are you guys sharp-eyed. Next Easter we're going to scramble the fershlugginer egg and scatter it throughout the magazine! It was too easy this year, why we got scads of letters, all of them correctly noting the egg under the question mark after the word "board". No more Mr. Nice Munchkin, we're going to get tough! In the meantime, here's the list of those astute enough to find the egg:

Matt Friedenberg — Morristown, NJ  
Greg Perry — Tulsa, OK  
Shawn Morrison — Satellite Beach, NJ  
Dennis L. Wasson — Council Bluffs, IA  
V.S. Gavande — Austin, TX  
David Gantenbern — Sterling Hts., MI  
Cherryl Chamblee — Raleigh, NC  
S. Louise Rankin — Nashville, TN

Sharon Burton & Keith Shirley — St. Albans, WV  
Patricia M. Finkenbine — Albuquerque, NM  
James Marcolesco — Cilla Park, CA  
Keith Durbin — Paducah, KY  
Alaa El Ghatit — Brookfield, WI  
Alan J. Wallace — Southfield, MI  
Dan Diegmuller — Cincinnati, OH  
Bob Cross — Cheshire, CT  
Leanne Phillips — Montpelier, VT  
Thomas O'Hara — Anson, ME  
John Eng — Plymouth, MN  
Thanks Fred.

This month our erstwhile bosses have decided to swamp you with software of all sorts, ranging from utilities to games, with a little patriotic fervor thrown in for good measure. Our feature article is a bit late for Flag Day and a tad early for the Fourth of July...but we figured (as we are wont to do on rare occasions) that you'd rather have this gem before our nation's birthday: We proudly present, for all of the systems we support, "Old Glory" by (who else?) William Morris and John

Cope. Those fellows don't ever stop cranking out that good stuff, do they?

And for you animal lovers, we've included a program by a veterinarian, John Baker, entitled "Bats", the Electronic Vampire Nightmare Game. What we munchkins would like to know is how does an electronic vampire differ from a flesh and blood vampire? They probably only put the bite on transistors and television sets...

Our utility this month is "Anallist" by Rod Fitzgerald. The program will take packed listings and list them out in a single statement, structured format, making it substantially easier to understand how a particular program works. Unless, of course, you're blessed with Munchkin perspicacity, in which case you wouldn't understand a one liner.

For you S-80 space jockeys we have "Divide and Conquer," by Philip C. Soine. This program has monsters from Krypton and a variety

continued on page 37

## COLUMN CALCULATOR 4.1

by David T. Gray

COLUMN CALCULATOR is a "word processor for numbers," a number processor designed to be used like a calculator. But it can handle large blocks of information as if handling one number at a time. The work space can be thought of as a large matrix with rows and columns much like an accountant's spreadsheet. Data can be easily entered into columns; and the columns can then be moved around. Columns can be overlaid from an existing data file on disk. One column can be added, subtracted, multiplied, divided, or raised to a power of another and the results put in another column. Columns can be compared to one another. Columns can be totalled, or set with a constant, and any column can be sorted, carrying the rest of the columns with it. A predefined function can be defined, thereby preprogramming the worksheet.

The statistical section provides analysis of the data. The analysis includes simple statistics, linear regression, simple correlation, histogram and the T-test.

The information can be printed out on the line-printer in a compressed format at any stage in the development of a data base. Thus, it can be used as a finished report or as

##	DESC	2	RENT	3	FOOD	4	CLOTH	5	MISC	6	TOTL
1	FRY CR										
2	TOG FOOD			25.45							
3	RENT	266.75									
4	SPICES			15.45							
5	CAR FMT					145.75					
6	MEAT			8.95							
7	MILK			3.89							
8	COFFEE			2.67							
9	BEER			3.15							
10	CHIPS							.89			
11	EGGS							1.42			
12	LIP							.75			
13	SALUCE							.98			
COMPUTE COL # 2 (+) COL # 3 = COL # 8											

a copy of the worksheet to permit the filling in of additional data for later entry into the data base. The data base can be saved on disk and recalled at a later date for modification or for generating a report. Any column in a file on disk can be referenced and added to the current worksheet.

All user communication with COLUMN CALCULATOR uses FLASH, the line input/editor routine. This enables the user to not only key in instructions, but to edit errors or data as well.

S-80, 32K disk . . . #26-COLC . . . . . \$39.95

**TSE-HARDSIDE**  
6 South St. Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790



Dear **SoftSide**,

I have recently obtained an Apple II Plus, and discovered your magazine shortly thereafter. I am generally satisfied with **SoftSide**, but there are two things that bother me enough to cause me to write. 1) Please, please do not squash the programs when they are listed. It is very difficult to read the listings and separate the commands when everything is run together. 2) Let's see some more creative programs for the Apple in Applesoft. I am especially interested in Adventures, and, since October, 1980, all Adventure programs have been for the S-80. It is very frustrating to see a highly interesting program, just to discover that it is not in a language I can use. How about more translations, or at least more instruction on how to translate into Applesoft.

This is not to give the impression that I am entirely dissatisfied. I have enjoyed a number of the programs so far, although I still can't get "Moon Landing" to run. Keep up the good work, but do take my suggestions into consideration.

D. Scott Harper  
Skokie, IL

**Editor's Reply: See "Outgoing Mail" this issue for a discussion of the Apple adventure matter.**

Dear **SoftSide**,

Being a beginner to computers I think as far as introducing computers, you have a great magazine. But I think it would be nice if you put a few S-80 Level I programs in along with some of the Atari ones. But what I really wanted to say was that I very much enjoyed your "Big Apple Adventure" (March issue).

Keep up the good work!!

Marc R. Ellingstad  
Burlington, WI

## SoftSide™ INPUT POLICY

**SoftSide** Magazine welcomes your comments and thoughts on both the magazine and the field of microcomputing. We try to publish as many of our readers' letters each issue as we can.

For the sake of clarity and legibility, all letters should be typewritten and double-spaced. Send your letters to:

SoftSide Publications,  
Input  
6 South St.,  
Milford, N.H. 03055

We reserve the right to edit any letters prior to publication.

Dear **SoftSide**,

I would like to congratulate you on publishing an excellent magazine.

Now to the nuts and bolts of this letter. Why don't you publish more games for the poor unfortunates who only have Level I S-80 computers with only 4K of RAM? I know it is extremely difficult to have an exciting game with only 4K of RAM to work with, but surely there are some around.

While I have your attention, do you plan to publish games for the S-80 Color Computer in the future?

Jerry Muller  
Kissimmee, FL

**Editor's Reply: Sorry about Level I programs, but due to the limitations of that BASIC, we stick to Level II and Disk BASIC. As for Color Computer programs, we are eagerly awaiting submissions for that system.**

Dear **SoftSide**,

Re: J. Dineen's letter (3/81 issue) regarding "Protour 80" in the 8/80 issue.

I just love his distance gauge for putting. It has improved my score by at least five strokes.

To return the favor, maybe he would like to make the following changes to the same program which, by the way, is excellent.

Rather than put the woods, irons, and wedge distances in PRINT AT statements, I made the following additions to lines 9705, 9714, and 9716.

With these changes, you have the distances on the screen as you are selecting your club.

9705 — Eight spaces after wedge(s) I added wood = 200-260 yards. Then three spaces and I added iron = 90-200 yards. Then three more spaces and add sand wedge = 30-70 yards.

9714 — Three spaces after (1-9) I added 1 = 200 yards. Then three spaces and add 9 = 90 yards.

9716 — Three spaces after (1-4) I added 1 = 260 yards. Then three more spaces and add 4 = 200 yards.

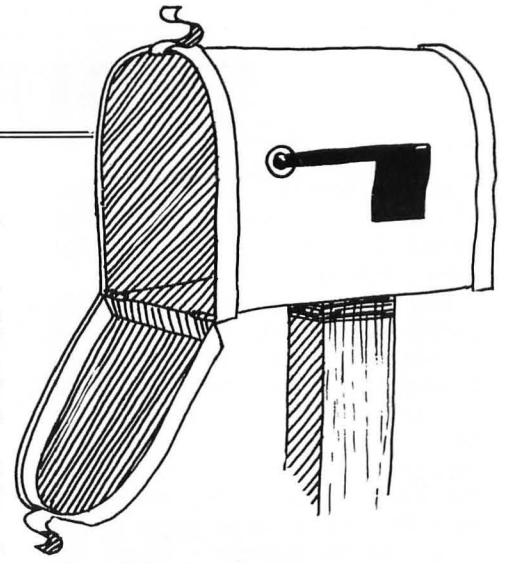
As previously stated, there is no referring back necessary as the distances and club selections are right in front of you.

I hope this minor improvement will help Joseph to improve his game. Perhaps it may even benefit a few of your other subscribers.

Suggestion — perhaps a column devoted strictly to improvements on your programs by readers would be a nice addition to your magazine.

Finally, may I say that **SoftSide** is the best medicine that any doctor could have prescribed for my S-80 and I am quite certain that numerous Atari and Apple owners are of the same opinion.

Raymond Nehilla  
Ambridge, PA



Dear **SoftSide**,

I enjoy your magazine for its game programs and other information within **SoftSide**'s covers. However, it would be nice to have covers on the issues I receive. The February, 1981, issue arrived in a protective envelope (I do not know why a protective envelope instead of an ordinary manila envelope) and when it was opened, the magazine inside was in sad shape. The cover was off and almost torn along the fold between front and rear covers. Pages inside the magazine were folded over and a few were torn. My little but expanding library of books and magazines is for the most part, in very good shape. This issue is not up to standard. I ordered a large number of rod binders to be able to keep the magazines in good shape and it would be difficult to keep this issue in a binder without totally destroying it in the process.

Other than that, I enjoy the magazine and eagerly await its arrival so that I can spend untold hours punching away at the keyboard of my Apple, putting in the programs and correcting the mistakes I make. I originally ordered the Apple version of **SoftSide**, but the first issue to arrive was the first combined issue. Contrary to what a few people said in the following issues, I have no problem with the combined format. I would like to know about the possibility of getting the back issues of **SoftSide: Apple edition**, and maybe some of the TRaSh-80 issues. May I also suggest a series of articles on translating one version of BASIC to another. I have learned a few tricks due to the side-by-side publishing of programs, but I would like to know more. Thank you, and keep up the good work.

Reginald Wagner  
Edmonton, Alberta, Canada

**Editor's Reply: 1) We are exploring solutions to the problem of the covers getting damaged in the mail. 2) Back issues are available. See page 45.**

Dear **SoftSide**,

I am attempting to compile a list of S-80 Model I programs that will (or will not) run on the Model III. I would appreciate any input from your readers.



# OUTGOING MAIL

by Dave Albert

To those who kindly respond, please mention if the program was on disk or tape; BASIC, SYSTEM, or disk CMD file; whether it runs under Level II or DOS; and, if DOS, which one. If any changes needed to be made to the program, what were they?

All those who send me information will be sent the compiled list (after a reasonable length of time to get all input) if a SASE is included with your information.

Thank you for your help.

Ken Knecht  
1340 W. 3rd ST. #130  
Yuma, AZ 85364

Dear **SoftSide**,

Most Atari owners know that if a key is not pressed in approximately nine minutes, the computer will go into a random color switching mode, to prevent damage to phosphor on the cathode ray tube of the monitor.

To understand this feature of the Atari, we can view the counting system at memory location 77.

Type in the following short program and view the results.

```
10 POKE 752, 1
20 PRINT PEEK(77)
30 GOTO 10
```

As you can see by observing the run, RAM location 77 starts the count at 0 and steps by a 1 count each 4 to 4.5 seconds. While the program is running, press any key except BREAK or CTRL keys. Memory location 77 will return to 0 and the countdown will begin again.

If you allow the program to run until the count reaches 128, it will set to 254 and the random changes begin.

The lower seven bits of memory location 77 are used to count from 0 to 127. At count 127 the lower seven bits are binary 1s. Adding one more bit changes the eighth bit from 0 to 1 and triggers the switching circuit. (Any number from 128 to 255 POKED into location 77 will cause an immediate color rotation.) Example:

```
5 POKE 77,200
```

If you wish to save time and avoid the nine-minute wait, add the following line to the above program:

```
15 POKE 77,120
```

This addition will start the count at 120 and the color rotation will begin in about 30 seconds.

To defeat this timer (Which can be annoying during a program using joysticks or paddles) insert a POKE 77,0 into the program where it will be executed frequently.

Kenneth Parsons  
Linn Creek, MO

Dear **SoftSide**,

I have been a subscriber for over two years. I must agree with some others. I will not be renewing my subscription. The old format for the S-80 was more to my liking.

I looked forward to each edition. That feeling has passed.

J.J. O'Malley, Jr.  
Wilkes-Barre, PA

Dear **SoftSide**,

I appreciate the effort that Jon Voskuil has made to include information about the PEEK and POKE statements in his "Math Decathlon" program. I am using an S-80 Model II which doesn't have PEEK and POKE, but I can do most of the things with other methods if I know what the PEEK and POKE statements in a program are doing. I hope that more authors will give me this kind of help. Maybe someone who knows can write an article for you that will give more help in solving the PEEK/POKE problem for those of us who are using S-80 Model IIs.

I would also like some help on using the system calls on my Model II. I may have the necessary information in the manual, but so far I have not been able to figure how to use them.

Donald M. Dealy  
Cumberland, RI

Dear **SoftSide**,

This letter is in response to a "human being" who has stopped looking for other human beings. This person obviously does not understand a number of things.

Before I get to them however, let me speak out for those of us who resent the idea that we are not considered HUMAN. **SoftSide** has to be one of the most human magazines around, along with being the single best for home or hobby computers.

The first thing this reader should understand is that in the free enterprise system the guy with the best product makes the most money automatically. It is an insult to the intelligence of the consumer to imply anything else. Certainly he considers himself intelligent enough to be a cause of that. In other words, I bet he buys what he considers best.

Second, if this reader understood even a little about the programs he reads in **SoftSide**, he would know that whatever computer a program is written for, it has an algorithm (a certain way) to solve a given problem. Which is the same for S-80s, Apples and IBM 370:165s. This fact increased the value of **SoftSide** fourfold.

Third, Tandy should hope to reach half the income from computers that IBM has. Working in the computer industry, I know that, although phenomenal for three short years as a computer marketing company, Tandy has but a crumb of the microcomputer cake.

Fourth, I think this reader should compare Tandy's software with the volumes of software written by others. He should be complaining about all the software written for CP/M systems not the S-80.

Finally, I think you guys do a great job and us "humanoids" are behind you 101%.

Larry Eiss  
Baldwinsville, NY

Dear **SoftSide**,

Just a short note to let you know that I

continued on next page

Hello, hello. It's that time again, I guess. Another look into the confused workings of **SoftSide**, courtesy of your friendly editors. The more I have to write this column, the more I realize the fractured nature of what needs to be said. It seems that the first order of business each month is to try to respond to some of the mail we receive, so let's get on with it.

One of the more common complaints that we receive is that we don't publish Adventures for the Apple and the Atari. It's a point well taken, but not easily resolved. As I've said here before, and undoubtedly will say again and again, **SoftSide** lives off of what our readers submit. And they don't submit Apple and Atari Adventures. We can't print what we don't get. So if all of you 6502 owners out there would write some Adventures, the rest of you will see them in **SoftSide** soon enough.

Another complaint we frequently hear is that we seem to favor the S-80 over the Apple and Atari. (Boy, you Apple/Atari owners sure are a vociferous bunch!) On the face of it, there's no arguing with that — it's true. We publish more S-80 programs than either of the other two. Yet recently we have published an article on how to make alterations to your Atari memory, and a series on programming ("Math Decathlon") written in Applesoft by our resident Apple wizard, Jon Voskuil. Furthermore, in this issue ("Hardware Corner") there's a tip on how to fix a Macrotronics cable so that you can use an Epson MX-80 to print text with an Atari. These are articles and programs that we create ourselves in order to help out the Apple/Atari folk. But the bottom line is that there are just a whole lot more people out there that own S-80s than Apples or Ataris, and they are certainly more prolific writers than Apple/Atari owners. And once again, **SoftSide** lives on what is submitted. If it doesn't come in the mail, it never sees print. Perhaps the "primitive" S-80 inspires people to write more, or maybe it's something else; but if we sort our mail into categories delimited by the type of computer it applies to, the vast majority of the S-80 mail is in the form of program

continued on next page



Wargamer's delight

# Three from Potkin

## Warpath

The Indians are on the warpath! The Chief, along with 24 braves, is out to take the garrison at the fort, or at least to stop reinforcements from entering the stockade. The General, with his 14 troopers, is trying to relieve the garrison before the flag is captured. The player determines the scenario through placement of boulders that provide both shelter and obstacles. Favorite scenarios may be replayed.

**S-80 Level II, 16K cassette \$14.95**  
#26-KRGS2

## Up Periscope

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 three of his fleet surviving the attempt. This realistic wargame includes sonar, depth charges, and torpedos.

**S-80 Level II, 16K cassette \$14.95**  
#26-UPER

## Kriegspiel II

A much improved two-player version of the original. Kriegspiel II is a wargamer's delight. Choose the number of mountains (up to 200) and pick a scenario from the 9,999 possible, and then watch the computer set up the pieces, towns, mountains and a river. To win, you must enter the capital city of your opponent or reduce his fighting strength to below half of your own

**S-80 Level II, 16K cassette \$14.95**  
#26-WARP



  
**TSE: HARDSIDE**  
6 South St., Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790

INPUT continued from previous page

am most pleased with the TEAC drive that I purchased from you at the N.Y. show. I realize that most people only write to complain but I am writing to praise you. The drive works beautifully and is much greater than my Shugart 400. The 40 tracks are nice too. No read data errors occurred so far (about 30 days). I was looking for a Percom or Pertec drive but this one I bought on your reputation entirely. I have not been disappointed.

My only suggestion to you would be to publicize the brand (TEAC) as a producer of fine audio equipment and let the S-80 world in on the quality/price of the their product. They are obviously newcomers to the hobby drive market and I would not have purchased it without your backing. I am a long-time subscriber to **SoftSide** and **Prog 80** as well.

Keep up the fine work, I look forward to doing business with you in the future.

Al Abrahamson  
Norwalk, CT

**Editor's Note: The drive referred to is marketed as "Hardside" drive.**

Dear **SoftSide**,

Thank you for the sample copy of the new **SoftSide**. It seems to be of high quality and must have taken much time and expense to produce.

I already get **Creative**, **Kilabaud** and **Byte** and would consider adding a "pure Apple" magazine to that list. I find the mixed (Apple, S-80, Atari, PET) format of **SoftSide** somewhat annoying since I must sift through for the Apple articles.

I'm only submitting this as a constructive personal criticism and may not necessarily reflect the consensus.

Good luck with **SoftSide**. I suspect you know this business much better than I do.

Richard Steck  
Lake Forest, IL



### OUTGOING MAIL

continued from previous page

or article submissions, while the Apple and Atari mail seems to be predominantly demands for programs rather than submissions. We still publish more BASIC software for those machines than any other magazine around, but that may end if you people don't get on the stick and write!

Well, that's all for now. In the next issue or so you may notice more changes in the way **SoftSide** looks and reads. Please write and let us know how you feel about the magazine as it grows.



• TRS-80 • APPLE 2 • ATARI • SORCERER •

SORCERER • TRS-80 • APPLE 2 • ATARI • SORCERER • PET

PET • TRS-80 • APPLE 2 • ATARI • SORCERER • PET • TRS-80

# FREE CATALOG

over 150 items

ORDER TOLL FREE

## (800) 327-7172

IN FLORIDA (305) 862-6917

'nuf said

TRS-80 • APPLE 2 • ATARI • SORCERER



(305) 862-6917

# Adventure

## INTERNATIONAL

A DIVISION OF SCOTT ADAMS, INC

BOX 3435 • LONGWOOD, FLA. 32750

# PASCAL-80™ on your S-80™



Phelps Gates, the author of "APL-80", brings you "Pascal-80" for your S-80. Now you can add another dimension to your programming skills by using this fast version of the compiled language Pascal.

"Pascal-80" is a powerful, structured and well-defined language for the S-80 microcomputer. This easy-to-use language makes writing well-structured, and therefore easily understandable programs simple. "Pascal-80" supports most of the features of UCSD Pascal, including RECORD, SET (to 256 members), FILE (text and record oriented), n-dimensional ARRAY (and ARRAY of ARRAY, etc.), global GOTO, ELSE in CASE statements, and BCD arithmetic accurate to a full 14 places (including log and trig functions), 6-digit optional. "Pascal-80" features a 23600 byte workspace in 48K, a 1000 line per minute compiler, an easy-to-use text editor, and plain English error messages, all the features you would expect in a Pascal costing hundreds more.

Variable Types:	Boolean, integer, char, real, real6, and text.
Constants:	Maxint, minint, true, false, and pi.
Files:	Input, output, and lp.
Procedures:	Read, readln, write, writeln, reset, rewrite, close, seek, cls, and poke.
Functions:	Abs, arctan, call, chr, cos, eof, eoln, exp, inkey, ln, mem, odd, ord, peek, pred, round, sin, signif, sqr, sqrt, succ, and trunc.

"Pascal-80" does not implement variant records, pointer and window variables, or functions and procedures used as parameters.

S-80 32K Disk #26-141001D ..... \$99.95



**TSE-HARDSIDE**  
 6 South St., Milford, NH 03055 (603) 673-5144  
 TOLL FREE OUT-OF-STATE 1-800-258-1790



# THE RACE OF SORTS

by Rik Karlsson

Sorting is a primary in the programmer's art. George Blank's article "Out of Sorts?" presented three sort subroutines that could easily be added to a BASIC programmer's programs. But, what about the relative efficiency of the sorts presented? Here is some illuminating information about the specific efficiencies of the three sorts that could help you in choosing one or another.

George's generalized sort subroutine program was the jumping off point for my attempt to determine what, if any, differences existed between the three sorts measured in the time needed to get the job done. The three sorts — Bubble, Ripple, and Shell-Walters all do the same thing. They take an array of data and alphabetize it A to Z or, in the case of numeric data, one to maximum number in the array.

First I modified George's generalized program to do the timing for me. Figure 1 is the modified program. I added code to direct the output to a printer as well as to the screen. The test would take nearly five hours and I didn't relish taking down the information developed by hand! Second, I added code (line 210) that initializes the real-time clock each time a sort sequence begins. After the sort routine finishes, a subroutine (lines 640-670) computes in total seconds, the length of time the sort has taken. The titles and data outputs to the screen were altered for my purposes because while George's excellent demonstrator programs depended upon screen output and formatting, thus limiting them to ten items, I wanted to sort arrays going up to 250 items. Last, I added a loop to the program that STEPPed ten items per loop and caused the sorting of the same data by each of the sort subroutines. Now I was set to RUN my creation and check out those sorting efficiencies.

Running the program causes a loop to be entered that starts with ten items in the A(F) array that are also duplicated in the B(F) "holding" array. The A(F) array is sorted by the Bubble sort first. The A(F) array is then reset to its original presort condition and the Ripple sort is performed, and so on through the Shell-

continued on next page

```
100 REM * THREE SORTS * GEORGE BLANK *12/2/80
110 REM * FROM FEBRUARY 1981 SOFTSIDE
120 REM * LOOPED COMPARE AND LPRINT MODS BY
130 REM * RIK KARLSSON 02/15/81
140 CLEAR200:LPRINT:CLS:DIMA(250),B(250)
150 FORST=10T0250STEP10:CLS:N=ST
160 FORF=1TON
170 A(F)=RND(99):B(F)=A(F)
180 NEXTF
190 I=1
200 IF I<>1THENFORF=1TON:A(F)=B(F):NEXT
210 FORF=1T06:POKE&H4040+F,0:NEXTF
220 A%=TIME%
230 ON I GOSUB270,380,500
240 I=I+1:IF I>3THENLPRINT:NEXTST:END
250 GOTO200
260 REM * BUBBLE SORT *
270 PRINT@64,"BUBBLE SORT DEMONSTRATOR";"(";N;");
280 LPRINT"BUBBLE SORT";"(";N;");
290 PRINT@128,RIGHT$(A$,8);:LPRINTRIGHT$(A$,8);
300 FOR A=1TON-1
310 FORB=A+1TON
320 IF A(A) <=A(B)THEN 340
330 T=A(A):A(A)=A(B):A(B)=T
340 NEXT B
350 NEXT A
360 A%=TIME%:T2%=A%:PRINT@192,RIGHT$(A$,8);:LPRINT " ";
RIGHT$(A$,8);:GOSUB640:RETURN
370 REM * RIPPLE SORT *
380 PRINT@320,"RIPPLE SORT DEMONSTRATOR";"(";N;");
390 LPRINT"RIPPLE SORT";"(";N;");
400 PRINT@384,RIGHT$(A$,8);:LPRINTRIGHT$(A$,8);
410 FOR C=1TON-1:FLAG=0
420 FOR B=1TON-C:A=B+1
430 IFA(B)<=A(A)THEN 460
440 FLAG =1
450 T=A(A):A(A)=A(B):A(B)=T
460 NEXT B
470 IF FLAG =1THENNEXTC
480 A%=TIME%:T2%=A%:PRINT@448,RIGHT$(A$,8);:LPRINT " ";
RIGHT$(A$,8);:GOSUB640:RETURN
490 REM * SHELL WALTERS *
500 PRINT@576,"SHELL WALTERS SORT DEMONSTRATOR";"(";N;");
510 LPRINT "SHELLW SORT";"(";N;");
520 PRINT@640,RIGHT$(A$,8);:LPRINTRIGHT$(A$,8);
530 C=N
540 C=INT(C/3)+1
550 FORA=1TON-C
560 IFA(A)<=A(A+C)THEN610
570 T=A(A+C):E=A
580 A(B+C)=A(B):B=B-C
590 IF B>0 THEN IF T<A(B) THEN 580
600 A(B+C)=T
610 NEXT A
620 IF C>1THEN 540
630 A%=TIME%:T2%=A%:PRINT@704,RIGHT$(A$,8);:LPRINT " ";
RIGHT$(A$,8);:GOSUB640:RETURN
640 REM * LENGTH OF TIME COMPUTER SUBROUTINE *
650 T2=VAL(MID$(T2$,13,2))*60:T2=T2+VAL(MID$(T2$,16,2))
660 LPRINT" TOTAL TIME: ";T2
670 RETURN
```

Figure 1

continued from previous page

Walters. After the three sorts have been accomplished and their data printed out, the outside loop increments by an additional ten items and the process is started all over again. This sequence occurs through a grand sort of 250 items.

Okay, what did I learn? Figures 2 and 3 show the results of the run. The total time column is total seconds the sort took plus, of course, processing time for a few lines of code. Since each sort subroutine is coded identically, that overhead processing time can be considered a constant between the three sorting routines. Thus, the relative magnitudes between the sorts can be directly compared with the assurance that the sort taking the least time would still take the least

time if the timing and printing overhead processing code was removed.

Results: The Shell-Walters sort beat the socks off the other two. The Ripple sort is a real time waster, showing up so at even the 20-30 item level. Beyond that, you could make a whole career out of waiting for it to complete a sort of a data base of any size. The Bubble sort is not much better, but would be the clear Sort-of-Choice were the Shell-Walters not around.

What about the processing time per item? Figure 4 shows my computation of processing time per item. Notice that the Shell-Walters almost levels off at about .29 seconds per item at 130. From there up to 250 it "maxed out" at .32 seconds per item.

The Bubble just kept on climbing each time reaching over 2.5 seconds per item at 250 items in the array. Old Mister Ripple outdid that performance on the negative side, screaming for altitude with a whopping 4.34 seconds per item at 250. Wow! I guess Ripple can get you higher faster after all!

The Shell-Walters sort is the hands-down winner in this contest. And that's the bottom line. BASIC programmers looking for a quicker sort should seriously consider dumping the Bubble sort for the Shell-Walters. Modifying present BASIC programs should be relatively easy using George Blank's February **Soft-Side** article...and after reading this, you've got plenty of reason to make the effort!

BUBBLE SORT ( 10 )	00:00:00	00:00:02	TOTAL TIME: 2	BUBBLE SORT ( 140 )	00:00:00	00:03:40	TOTAL TIME: 220
RIPPLE SORT ( 10 )	00:00:00	00:00:02	TOTAL TIME: 2	RIPPLE SORT ( 140 )	00:00:00	00:05:46	TOTAL TIME: 346
SHELLW SORT ( 10 )	00:00:00	00:00:01	TOTAL TIME: 1	SHELLW SORT ( 140 )	00:00:00	00:00:39	TOTAL TIME: 39
BUBBLE SORT ( 20 )	00:00:00	00:00:05	TOTAL TIME: 5	BUBBLE SORT ( 150 )	00:00:00	00:04:03	TOTAL TIME: 243
RIPPLE SORT ( 20 )	00:00:00	00:00:07	TOTAL TIME: 7	RIPPLE SORT ( 150 )	00:00:00	00:06:35	TOTAL TIME: 395
SHELLW SORT ( 20 )	00:00:00	00:00:03	TOTAL TIME: 3	SHELLW SORT ( 150 )	00:00:00	00:00:46	TOTAL TIME: 46
BUBBLE SORT ( 30 )	00:00:00	00:00:11	TOTAL TIME: 11	BUBBLE SORT ( 160 )	00:00:00	00:04:35	TOTAL TIME: 275
RIPPLE SORT ( 30 )	00:00:00	00:00:16	TOTAL TIME: 16	RIPPLE SORT ( 160 )	00:00:00	00:07:19	TOTAL TIME: 439
SHELLW SORT ( 30 )	00:00:00	00:00:05	TOTAL TIME: 5	SHELLW SORT ( 160 )	00:00:00	00:00:47	TOTAL TIME: 47
BUBBLE SORT ( 40 )	00:00:00	00:00:20	TOTAL TIME: 20	BUBBLE SORT ( 170 )	00:00:00	00:05:02	TOTAL TIME: 302
RIPPLE SORT ( 40 )	00:00:00	00:00:27	TOTAL TIME: 27	RIPPLE SORT ( 170 )	00:00:00	00:08:20	TOTAL TIME: 500
SHELLW SORT ( 40 )	00:00:00	00:00:08	TOTAL TIME: 8	SHELLW SORT ( 170 )	00:00:00	00:00:53	TOTAL TIME: 53
BUBBLE SORT ( 50 )	00:00:00	00:00:29	TOTAL TIME: 29	BUBBLE SORT ( 180 )	00:00:00	00:05:40	TOTAL TIME: 340
RIPPLE SORT ( 50 )	00:00:00	00:00:37	TOTAL TIME: 37	RIPPLE SORT ( 180 )	00:00:00	00:09:16	TOTAL TIME: 556
SHELLW SORT ( 50 )	00:00:00	00:00:10	TOTAL TIME: 10	SHELLW SORT ( 180 )	00:00:00	00:00:50	TOTAL TIME: 50
BUBBLE SORT ( 60 )	00:00:00	00:00:42	TOTAL TIME: 42	BUBBLE SORT ( 190 )	00:00:00	00:06:13	TOTAL TIME: 373
RIPPLE SORT ( 60 )	00:00:00	00:00:57	TOTAL TIME: 57	RIPPLE SORT ( 190 )	00:00:00	00:10:14	TOTAL TIME: 614
SHELLW SORT ( 60 )	00:00:00	00:00:14	TOTAL TIME: 14	SHELLW SORT ( 190 )	00:00:01	00:00:57	TOTAL TIME: 57
BUBBLE SORT ( 70 )	00:00:00	00:00:56	TOTAL TIME: 56	BUBBLE SORT ( 200 )	00:00:00	00:06:56	TOTAL TIME: 416
RIPPLE SORT ( 70 )	00:00:01	00:01:20	TOTAL TIME: 80	RIPPLE SORT ( 200 )	00:00:00	00:11:03	TOTAL TIME: 663
SHELLW SORT ( 70 )	00:00:00	00:00:16	TOTAL TIME: 16	SHELLW SORT ( 200 )	00:00:00	00:00:58	TOTAL TIME: 58
BUBBLE SORT ( 80 )	00:00:00	00:01:09	TOTAL TIME: 69	BUBBLE SORT ( 210 )	00:00:00	00:07:44	TOTAL TIME: 464
RIPPLE SORT ( 80 )	00:00:00	00:01:49	TOTAL TIME: 109	RIPPLE SORT ( 210 )	00:00:00	00:12:27	TOTAL TIME: 747
SHELLW SORT ( 80 )	00:00:00	00:00:21	TOTAL TIME: 21	SHELLW SORT ( 210 )	00:00:00	00:01:05	TOTAL TIME: 65
BUBBLE SORT ( 90 )	00:00:00	00:01:31	TOTAL TIME: 91	BUBBLE SORT ( 220 )	00:00:00	00:08:22	TOTAL TIME: 502
RIPPLE SORT ( 90 )	00:00:00	00:02:11	TOTAL TIME: 131	RIPPLE SORT ( 220 )	00:00:00	00:13:41	TOTAL TIME: 821
SHELLW SORT ( 90 )	00:00:00	00:00:21	TOTAL TIME: 21	SHELLW SORT ( 220 )	00:00:00	00:01:04	TOTAL TIME: 64
BUBBLE SORT ( 100 )	00:00:00	00:01:52	TOTAL TIME: 112	BUBBLE SORT ( 230 )	00:00:00	00:09:07	TOTAL TIME: 547
RIPPLE SORT ( 100 )	00:00:00	00:02:54	TOTAL TIME: 174	RIPPLE SORT ( 230 )	00:00:00	00:15:06	TOTAL TIME: 906
SHELLW SORT ( 100 )	00:00:00	00:00:25	TOTAL TIME: 25	SHELLW SORT ( 230 )	00:00:00	00:01:15	TOTAL TIME: 75
BUBBLE SORT ( 110 )	00:00:00	00:02:22	TOTAL TIME: 142	BUBBLE SORT ( 240 )	00:00:00	00:09:49	TOTAL TIME: 509
RIPPLE SORT ( 110 )	00:00:00	00:03:33	TOTAL TIME: 213	RIPPLE SORT ( 240 )	00:00:00	00:16:28	TOTAL TIME: 988
SHELLW SORT ( 110 )	00:00:00	00:00:30	TOTAL TIME: 30	SHELLW SORT ( 240 )	00:00:00	00:01:17	TOTAL TIME: 77
BUBBLE SORT ( 120 )	00:00:00	00:02:42	TOTAL TIME: 162	BUBBLE SORT ( 250 )	00:00:00	00:10:34	TOTAL TIME: 634
RIPPLE SORT ( 120 )	00:00:00	00:04:10	TOTAL TIME: 250	RIPPLE SORT ( 250 )	00:00:00	00:18:04	TOTAL TIME: 1084
SHELLW SORT ( 120 )	00:00:00	00:00:30	TOTAL TIME: 30	SHELLW SORT ( 250 )	00:00:00	00:01:20	TOTAL TIME: 80
BUBBLE SORT ( 130 )	00:00:00	00:03:11	TOTAL TIME: 191				
RIPPLE SORT ( 130 )	00:00:00	00:05:05	TOTAL TIME: 305				
SHELLW SORT ( 130 )	00:00:00	00:00:38	TOTAL TIME: 38				

Figure 2

Figure 3

continued on next page

continued from previous page

SECONDS PER ITEM

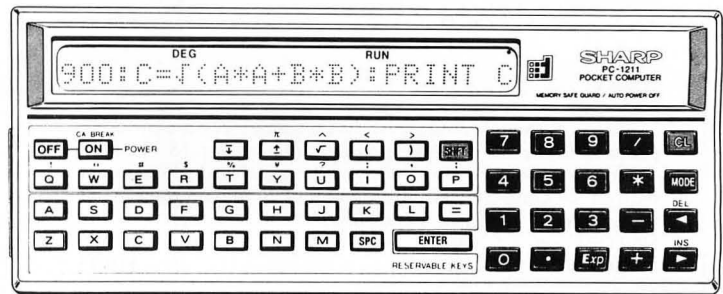
#Items	Bubble	Ripple	Shell-Walters
10	.20	.20	.10
20	.25	.35	.15
30	.37	.53	.17
40	.50	.68	.20
50	.58	.74	.20
60	.70	.95	.23
70	.80	1.14	.23
80	.86	1.36	.26
90	1.01	1.46	.23
100	1.12	1.74	.25
110	1.29	1.94	.27
120	1.35	2.08	.25
130	1.47	2.35	.29
140	1.57	2.47	.28
150	1.62	2.63	.31
160	1.72	2.74	.29
170	1.78	2.94	.31
180	1.89	3.09	.28
190	1.96	3.23	.30
200	2.08	3.32	.29
210	2.21	3.56	.31
220	2.28	3.73	.29
230	2.38	3.94	.33
240	2.45	4.12	.32
250	2.54	4.34	.32

Figure 4



# PC-1211

## Pocket Computer



### For Engineers, Scientists, Businessmen

USE IT WHEREVER YOU GO — IN THE FIELD, CLASS, COMMUTING.

Handy pocket size: 6 oz., 2 3/4 x 6 7/8 x 11/16"

BASIC Language: Most of the features of the Level I TRS-80 microcomputer. Program capacity of 1424 steps.

Editing Functions: Cursor shifting, insertion, deletion, line up and down.

Prerecorded Programs: A growing library of programs is available on cassette tape. Memory retains data and programs even with the power off.

Includes: Hard case, long-life batteries, Instruction and Application manuals, BASIC textbook, two templates, and 90-day Sharp warranty.

Special Introductory Price: **\$229.50** (includes shipping)

### BRAND NEW! CE-122 Printer/Cassette Interface

Use to load prerecorded software, store and save programs. Use to print out programs and calculation records.

Size: 11 3/32 x 3 3/4 x 1 3/8" (same dimensions with PC-1211 inserted)

Printer: 16-digit, 1 line/second, standard paper

Compatibility: completely compatible with TRS-80 Pocket Computer.

Includes: Ni-Cd rechargeable battery, AC adapter, carrying case, and cassette cable.

Price: **\$139.50** (includes shipping)

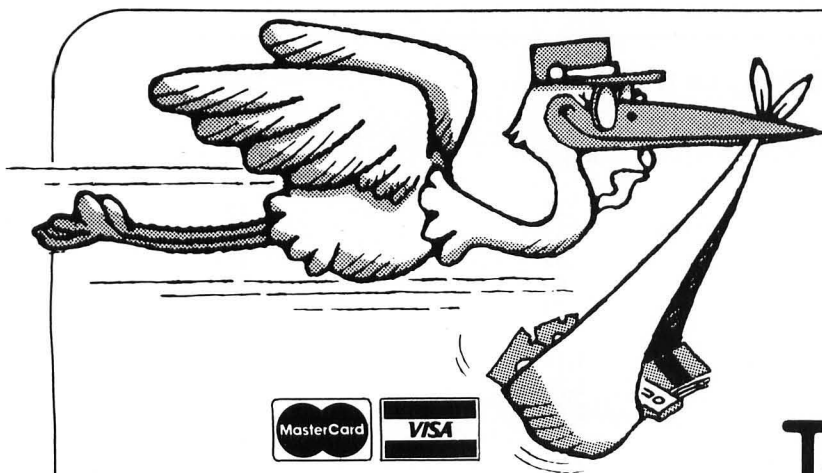
Combination Price: (PC-1211 plus CE-122) **\$359.50** (includes shipping)

To order  
or  
for additional information:

Payment by Personal check, Master Charge, or VISA #. Massachusetts residents include 5% Sales Tax.

**ATLANTIC NORTHEAST MARKETING, INC.**

P.O. Box 921 Dept. SS61  
Marblehead, MA 01945  
617-839-0285



# NEW ARRIVALS AT:

## TSE-HARDSIDE

6 South St., Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790

### Datsoft Inc.

#### MICRO-PAINTER

You will find your computer screen filled with specially designed patterns which you can fill with 21 vibrant colors in virtually unlimited combinations. "Micro-Painter" challenges your imagination as you explore this newest dimension in computerized art.

Apple II Disk #47-279001D ..... \$34.95

#### ATARI MAILING LIST

A versatile disk-based system, with the efficiency of a micro format and capability equal to that of many larger computers. The "Atari Mailing List" will become an important facet of an improved business system for you.

Atari Disk #36-279002D ..... \$24.95

#### IAGO

"Iago", Othello's antagonist, has returned. We are proud to present the classic Shakespearean challenger to Othello. "Iago" incorporates one of the most powerful strategies ever written for computers. If you think you've played the best at Othello, wait until you meet "Iago"!

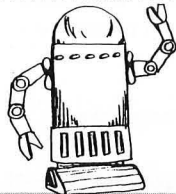
S-80 Disk #26-279003D ..... \$24.95

S-80 Cassette #26-279003C ..... \$19.95

## PROGRAMMA

#### FLEET

"Fleet" is an exciting battleship game. The object is to destroy all the ships in the sea in as few moves as possible. Atari 16K Cassette, #36-277018C ..... \$6.95



#### CHASE

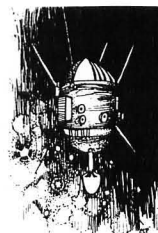
The object of this game is to escape. You are pursued by robots and you must make them run into mines to destroy them so you can get away.

Atari 8K Cassette, #36-277017C ..... \$6.95

#### HUNT/HUNTWRITER

This is a first-of-its-kind program in which you design and define the game you would like to play. The only limitation is your imagination. This meta-game is used to write adventure games to your specifications.

S-80 16K Cassette, #26-277016C ..... \$19.95



#### SPACE WARS

This is a super-fast action Hi-Res game. Two players use paddles to maneuver their ships into position in order to fire upon one another.

Apple II 16K Cassette, #47-277015C ..... \$9.95

#### FORTH

A vocabulary-based threaded language that allows the user to tailor the system to meet a specific application. System includes 200 predefined words, incremental assembler, text editor, block I/O buffers and much more. Comes with 50-page user manual.

Apple II 32K Disk #47-277002D ..... \$49.95

#### LISA ASSEMBLY Version 2.0

"Lisa" is a totally interactive symbolic assembler. With all its features it is a powerful assembler for the Apple II. Sweet 16 mnemonics, more commands, more psuedo opcodes and extended mnemonics are just some of its special features. Files may be saved as either binary type or text files. A 1000 line program only takes three seconds to assemble and all error messages are displayed in plain English, not just error codes. The program comes with a 144 page manual.

Apple II 48K Disk #47-277001D ..... \$49.95

#### SUPER STARWARS

This is an excellent Hi-Res game which starts you from hyper-space and moves you into normal space, where you try to destroy 32 Three-D Tie Fighters.

Apple II 32K Cassette, #47-277014C ..... \$15.95





## 6502 APPLICATIONS BOOK

This second book in the 6502 series presents practical, real-life application techniques for the 6502 microprocessor. Readers will learn how to interface the 6502 to the outside world in useful applications. Programs and circuits presented include a complete home alarm system (fire and intrusion), an electronic piano, a motor speed regulator, a time of day clock, and an appliance controller, as well as industrial applications and analog-digital conversion. This book teaches all the basic skills needed to use the 6502 in practical applications. This is the I/O book for the 6502. Elementary knowledge of programming is assumed.

#65-261008B ..... \$12.95



## 6502 GAMES

Games are fun and provide one of the best ways to learn advanced programming techniques. This book explains how to design and program all kinds of computer games ranging from the passive (music) to the strategic (tic-tac-toe). It presents algorithms and detailed programming techniques for ten types of computer games with the goal of sharpening skills and developing competence in Assembly Language programming so readers can design programs for original games and other applications. Each game section includes rules, instructions for playing a typical game, algorithm(s) and a program (data structures, programming techniques, and detailed description). This the enjoyable way to learn Assembly Language programming. All games can be played with a SYM board and a single accessory games-board as described in the book. Elementary Assembly-level programming required.

#65-261007B ..... \$12.95

## 50 PASCAL PROGRAMS

This book contains a useful library of important programs, each fully documented and described, including the algorithm and specific programming techniques used to solve the problem. Programs are written in UCSD Pascal and cover a broad range of applications providing a learning tool for anyone interested in developing skills in Pascal programming. Assumes a basic knowledge of Pascal.

#65-261004B ..... \$13.95

## FIFTY BASIC EXERCISES

This book is designed to teach BASIC through actual practice. It presents graduated exercises in mathematics, business, operations research, games, and statistics. Each exercise contains a statement and analysis of the problem, a solution with flowchart and comments, and a program implementing the solution, accompanied by sample runs. Besides allowing readers to check their understanding and progress while learning, this method teaches problem solving in a "top-down" manner: sub-problems are identified and solved separately, then combined into a modular program that's easy to read and modify. The book stresses programming style and the reasons behind each design decision. All programs are written in Microsoft BASIC and will run on an S-80, PET/CBM, Apple or any other computer equipped with Microsoft BASIC.

#65-261003B ..... \$12.95



## INSIDE BASIC GAMES

This book teaches the reader how to design error-free interactive BASIC programs including games and other real-time situations. Eight different kinds of computer games (a total of 14 games) are described in detail, then completely explained and analyzed in order to illustrate how the games were designed and developed in BASIC. All facets of game program design, including program structuring, cursor positioning, randomization and other concepts are discussed. Programs for games such as Hangman, Ten-Key Flicker, and Taxman are coded in Microsoft BASIC and versions are provided for PET/CBM, Apple II, and S-80. Some knowledge of BASIC programming is assumed.

#65-261002B ..... \$13.95

## DON'T! (Or How to Care for Your Computer)

This is exclusively dedicated to the care, preservation and correct operation of a small computer system: the computer itself, the CRT terminal, the printer, the magnetic disks and tapes — even the computer room. Most "computer failures" are caused by operator ignorance or negligence. In everyday language this book gives all the do's and don'ts of successful operation for each piece of computer hardware and software, including correct procedures for safety and security. It also tells what to do when something doesn't work.

#65-261001B ..... \$8.95

## PROGRAMMING THE 6502

This first book in the 6502 series is an educational text designed to teach Assembly Language programming for the 6502 microprocessor. From elementary concepts through advanced data structures and program development, all essential aspects of programming are explained in a logical format using everyday language. This systematic and clear presentation, with exercises of increasing difficulty, builds programming skills to the point where the reader is able to understand and write 6502 programs of considerable complexity. The book is carefully designed for easy reading yet it is thorough and complete. All important aspects of programming are presented individually including trade-offs between hardware and software, a detailed explanation of each 6502 instruction, and 6502s internal register and bus operation.

#65-261006B ..... \$12.95



## PROGRAMMING THE Z80

This book covers the Z80, all the way from basic concepts to advanced programming techniques and will enable every reader to write complete application programs for a Z80 based computer system. This progressive, highly organized presentation includes exercises to measure progress and comprehension at each step, from hardware organization through data structures. **Programming the Z80** treats all aspects of Z80 programming in a comprehensive yet simple way, starting by explaining the effect of each instruction and systematically working up to the development of all common type programs, from arithmetic utilities to parallel or serial input/output packages.

#65-261005B ..... \$14.95

# Strategic Simulations



## TORPEDO FIRE

Using 3-D graphics, the submarine commander gets a realistic view of the battlefield through the periscope, which must be used to sight and demolish the enemy convoy. The escort player must safely guide freighters through the waiting net of submarines.

The thirty different WWII ships and subs are rated in strict adherence to historical accuracy.  
 Apple II+ 48K Disk, #47-219006D ..... \$59.95



Hayden Book Company, Inc.

## BASIC BASIC: AN INTRODUCTION TO COMPUTER PROGRAMMING IN BASIC LANGUAGE, Second Edition

Over 100 sample programs present the essential statements of BASIC. Each new language statement or capability is clearly explained at the time that it is first used in a sample program. Every section is followed by practice problems. Includes: writing a program; loops and lists; elementary data processing; strings and files; the quadratic function; and much more.

#65-210009B ..... \$10.50

## TRS-80 BEGINNER'S PROGRAMS WITH EXPLANATIONS

Here's a valuable book of practical and interesting programs for home use that can be understood and used immediately by the beginner in personal computer programming. You'll learn step-by-step how 21 sample TRS-80 programs work. Program techniques are described line-by-line within the programs, and a unique Matri-Dex matrix index will enable you to locate other programs using the same BASIC commands and statements. Each program includes a detailed description, a complete listing, an explanation of what the program does, and instructions for modification. To help you understand the programs, photographs show how they will appear on the screen.

#65-210004B ..... \$7.95

## BASIC COMPUTER PROGRAMS FOR BUSINESS: Vols. 1 and 2

These two volumes provide a wealth of practical business applications. Each program is documented with a description of its functions and operation, a listing in BASIC, a symbol table, sample data, and one or more samples.

Volume 1 contains over 35 programs covering: budgets, depreciation, cash flow, property comparisons, accounts payable, order entry, etc.

Volume 2 contains over 30 programs covering: marketing and sales planning and analysis for customer lists; customer call-back scheduling; salesman scheduling; personnel planning and analysis for personnel lists, employee time card recording; and more.

Volume 1, #65-210005B ..... \$9.95  
 Volume 2, #65-210006B ..... \$10.95



Prentice-Hall, Inc.

## TRS-80 ASSEMBLY LANGUAGE

Now for both the first-time user as well as experienced users of the S-80 microcomputer, here is a book that explains Assembly Language programming in a thorough, yet easy-to-understand style. **TRS-80 Assembly Language** contains all of the information you need in order to develop Machine Language programs.

In this book you will find clear presentations of all introductory concepts in the use of the S-80, completely tested practical programs and subroutines, details of ROM, RAM, and disk operating systems, comprehensive tables, charts, appendices and much more!

#65-280001B ..... \$9.95



## ADDRESS BOOK

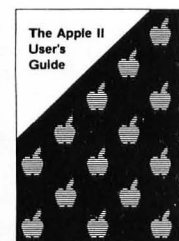
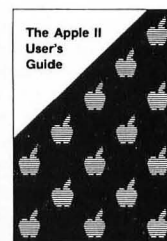
A great mailing list program for the Apple! Store 700 addresses per disk. Select by record contents for viewing, label printing, or automatic phone dialing. Quickly sort the file in any order.

Apple II 48K Disk #47-214004D ..... \$49.95

## FORM LETTER

Major capabilities include automatic, repetitive printing of letters from mailing list records using the "Address Book" software; direct entry mode allowing individual printing of letters without permanent mailing list storage; insertion of mailing list information anywhere in a letter; on-screen prompts query operator for information to be inserted in individual letters; and special "commands" allow changing content of individual letters depending on the recipient.

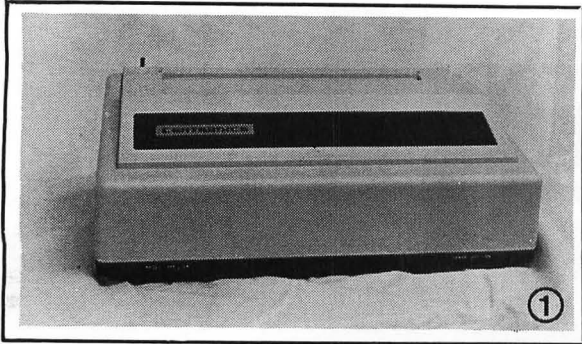
Apple II or II+ 48K Disk #47-214003D ..... \$100.00



## THE APPLE II USER'S GUIDE

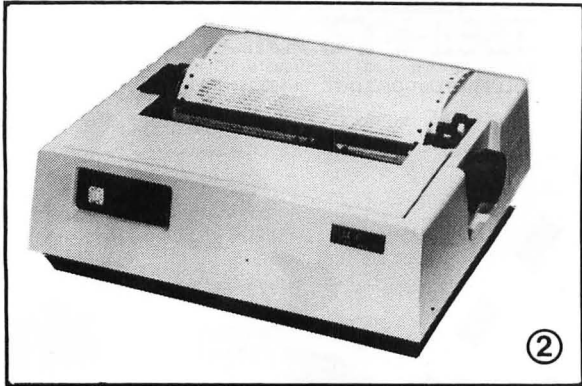
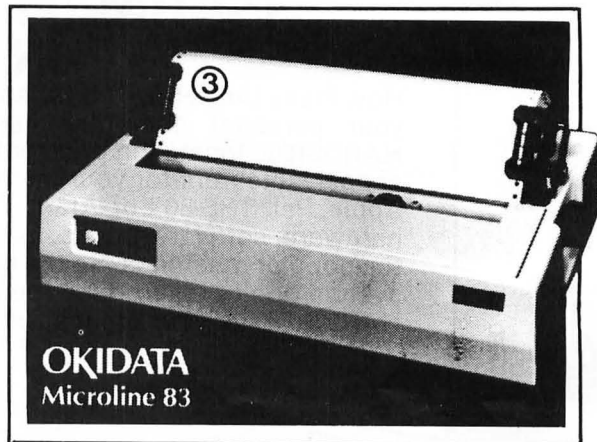
This guide is the key to unlocking the full power of the Apple II Plus computer. Topics include: Applesoft and Integer BASIC programming — especially how to make the best use of Apple's sound, color, and graphics capabilities, and a thorough description of every BASIC statement, command, and function; Machine level programming — although not a Machine Language programming guide, this book covers the Machine Language Monitor in detail; Hardware features — the disk drive and printer are covered in separate chapters; Advanced programming — special sections describe high resolution graphics techniques and other advanced applications.

#65-255001B ..... \$15.00



# PICK YOUR PRINTER FROM TSE•HARDSIDE TODAY!

- TYPRINTER 221 ..... (9-221) \$2495.00
- NEC 5510 SpinWriter ..... (9-5510) \$2695.00
- NEC 5530 SpinWriter ..... (9-5530) \$2595.00
- ① CENTRONICS 730 ..... (9-730) \$619.00
- CENTRONICS 737 ..... (9-737) \$729.00
- OKIDATA Microline-80 ..... (9-80) \$459.00
- ② OKIDATA Microline-82 ..... (9-82) \$649.00
- ③ OKIDATA Microline-83 ..... (9-83) \$949.00
- OKIDATA SL300 (300 LPM) ..... (9-SL300) \$3795.00
- EPSON MX-70 ..... (9-MX70) \$419.00
- EPSON MX-80 ..... (9-MX80) \$519.00
- EPSON MX-80FT ..... (9-MX80 FT) \$619.00



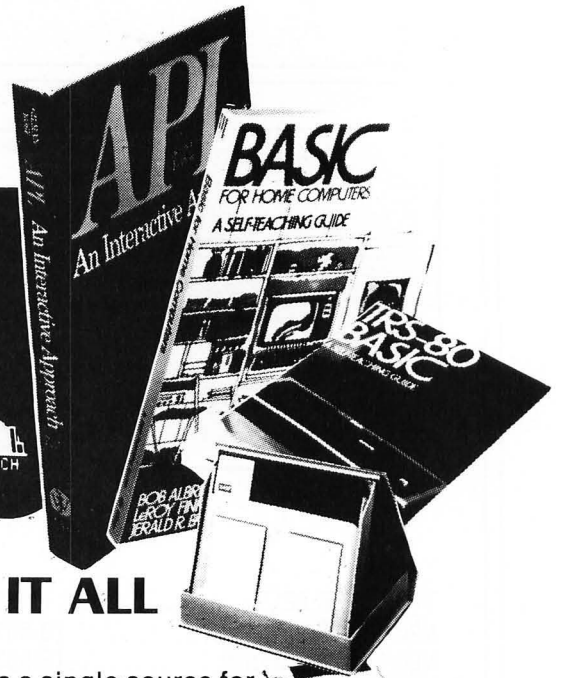
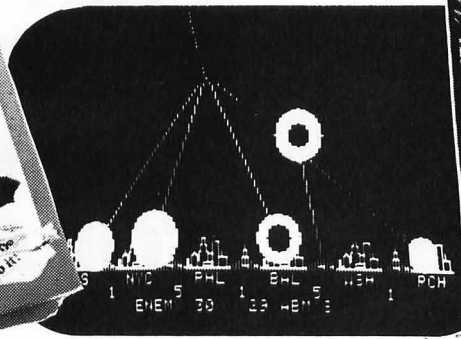
- RS MOD-I Printer (36-pin) Int. Cable (26-1411) \$59.00
- RS MOD-I Printer (40-pin) Int. Cable (26-1416) \$59.00
- RS MOD-I & III Printer (36-pin) Cable (26-1401) \$29.00
- RS MOD-I & III Printer (40-pin) Cable (26-1415) \$29.00
- RS MOD-I & III LRC Printer Cable ..... (9-10) \$29.00
- APPLE Parallel Int. & Cable (36-pin) . (47-936) \$100.00
- APPLE Parallel Int. & Cable (40-pin) . (47-940) \$100.00
- APPLE Asynchronous RS-232C Int (47-7710A) \$159.00
- ATARI-Macrotronics Print. (36-pin) Int(36-936) \$69.95
- ATARI-Macrotronics Print. (40-pin) Int(36-940) \$69.95

- NEC Tractor-Feed Option ..... (9-5000) \$229.00
- BDT Sheet-Feeder (NEC only) Option(9-5005) \$1849.00
- Microline-80 or 82 Tractor-feed Option(9-80-T) \$59.00
- Centronics Zip-Pack Ribbons (3-pack) . (21-01) \$15.95
- SpinWriter Multi-Strike Ribbons (3) ... (21-02) \$19.95
- Microline Printer Ribbon (3-pack) ..... (21-04) \$15.95
- EPSON Printer Ribbons (2-pack) ..... (21-05) \$29.95

**TSE•HARDSIDE**  
6 South St., Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790

TERMS: Prices and specifications are subject to change. HARSDIDE accepts VISA & MASTERCARD. Certified checks and Money Orders; Personal checks accepted (takes 3 weeks to clear). HARSDIDE pays all shipping charges (within 48 states) on all PREPAID orders over \$100.00. On all orders under \$100.00 a \$2.50 handling charge must be added. COD orders accepted (orders over \$250 require 25% deposit) there is a \$5.00 handling charge. UPS Blue Label, Air Freight available at extra cost.





## TSE-HARDSIDE HAS IT ALL IN ONE!

How many times have you wished that there was a single source for your personal computer needs? Well look no further, TSE-HARDSIDE, located in pleasant New Hampshire, has virtually every conceivable item for your micro. Whether you're shopping for your Apple, Pet, TRS-80™ or Atari, TSE-HARDSIDE has it all. We stock hardware, software, books, magazines and specialty items for all of the popular machines. So the next time you're out shopping for your system don't be surprised, be satisfied. Remember the name TSE-HARDSIDE as your choice for quality, service and reliability.



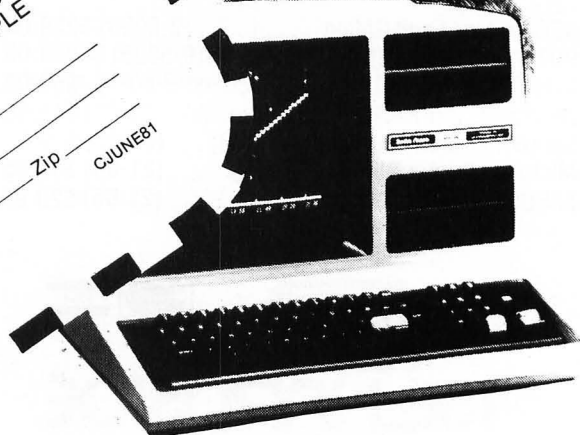
**TSE-HARDSIDE**  
6 South St., Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790



# Save \$2.00

If you or a friend would like to receive our TSE/HARDSIDE Complete Computer Catalog, simply send \$1.00 and receive a \$2.00 certificate good toward your first purchase of software, hardware, books or specialty items from TSE/HARDSIDE. I have a (check one)  TRS-80,  APPLE  ATARI  PET. A photocopy of this coupon will be accepted.

Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_  
MAIL TO: TSE/HARDSIDE  
6 South Street  
Milford, NH 03055  
CJUNE81





# APL

by Phelps Gates

Now a high-level, scientific programming language for the home computer that doesn't cost \$200 or \$300. The power of this language is in its strong mathematical operations, especially with regard to matrices and vectors. Programs requiring matrix multiplication or other matrix problem solving that would require hours of programming time in BASIC are solved quickly and with minimal effort in APL.

To aid in learning APL, lessons are included on the disk. Starting from the basics, you are brought step by step through the various programming techniques involved with APL. These lessons act as a tutor which will have you "talking APL" in no time. Also available is the book, "APL: An Interactive Approach," which reinforces many of the examples given in the lessons and provides additional insight into APL programming.

## FEATURES

APL-80 on disk contains the following features: )SAVE and )LOAD workspace on disk; )COPY other workspaces into current ones; Return to DOS for directory or commands without losing your workspace; Send output to lineprinter; Five workspaces of lessons included; Sequential and random files; 15 digit precision; Monadic and dyadic transposition; Easy editing within FUNCTION lines; Latent expressions (FUNCTION can "come up running" when loaded); Tracing of function execution; Real-time clock; User-control of random link; Workspace is 25587 bytes (in 48K machine); Arrays may have up to 63 dimensions.

## COMMANDS APL-80

APL-80 supports the following commands; Absolute value, add, and assign, branch, catenate, ceiling, chr\$/asc, circular, combinational, comment, compress, deal, decode, divide, drop, encode, equal, expand, exponential, factorial, floor, format, grade down, grade up, greater, greater/equal, index generator, indexing, index of, inter product, label, less, less/equal, logarithm, maximum, member, minimum, multiple, nand, negate, nor, not, not equal, or, outer product, peek, poke, quad, quote quad, random, ravel, reciprocal, reduction, reshape, residue, reverse, rotate, scan, shape, sign, system, subtract, take, transposition.

## SPECIFICATIONS

Minimum system requirements: 32K disk system (&48K recommended) includes APL-80, Five workshapes of lessons, instruction manual. . . #26-APLD . . . \$39.95 on disk

Reduced feature: 16K Level II tape version, no lessons.

Transpositions, format, and inner product not implemented. Reduced domain for some functions, 6 digit accuracy.

. . . #26-APLT . . . \$14.95 on cassette

## LIMITATIONS

Due to the absence of the special APL character set on the TRS-80 , APL-80 uses shifted letters to represent the various APL characters. In addition to the keyboard limitations, lamination, domlno, and matrix inverse are not implemented but can be derived with user-defined functions. Multiple specifications must be split into two statements unless the left-hand assignment is to a quad. This also applies to implied multiple specifications. Reduction and reshape (p) are not permitted for empty arguments; the argument of add/drop may not be scalar; empty indices are not permitted. A quad (q) can't be typed in response to a quad (nor can the name of a function which itself gets input from a quad). Quote-quad (m) is permitted. No more than 32 user functions can be defined in a single workspace and a function may not contain more than 255 lines.

A comment (c) must occupy a separate line: a comment can't follow a function statement on the same line.

In the tape version, arrays are limited to five (5) dimensions.

**TSE-HARDSIDE**  
6 South St., Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790



# MATH DECATHLON

## Part 4

by Jon Voskuil

By now all you mathletes should be developing strong, healthy crania from all the mental gymnastics of the past three months. Not to mention strong, healthy phalanges from all the typing of the past three months. After entering this final installment, you will have a complete, 16,595-byte program to call your own. The last three of the ten events, plus the routine which plays the complete national song of the winner, will finish you. . . er, will finish off the program. Again, let me mention that back issues are available if you have missed any of the first three installments. (Cassette and disk subscribers have the complete program with this issue.)

### EVENT #8: MENTAL MATH

After loading your existing program from tape or disk, and adding to lines 310 and 330 as indicated, you're ready for the eighth event. Here the computer starts you off with a number, and instructs you to do a series of calculations. Each instruction is erased by the next one, so you have to keep track of the results in your head. Players at skill level 0 must do four operations, and two more are added for each skill level increase.

Including the starting number, then, there are up to nine numbers involved in the chain of calculations, which are stored in variables N1 through N9. These are chosen at random (within defined limits), and then simply incorporated into PRINT statements, in lines 8070-8180. The correct result is computed in 8200, and compared to the player's input. Note that N6 through N9 are initialized in 8070, so that if no other values are given to them (as happens at lower skill levels) they will not influence the computed answer.

The length of time that each instruction is displayed is controlled by M1. Each successive timing loop is shorter than the previous one, decreasing more than two-thirds by the last instruction for level 2. If the overall timing seems too slow or too fast for you, simply adjust the value of M1 in line 8010.

### EVENT #9: PENNY BATTLE

This is the only event in which the computer participates as something more than a scorekeeper. This is a variation of a very ancient type of game, where two people take turns picking up objects from a pile until only one is left. In this case one of the people is the Apple, and the pile is of pennies. The size of the pile, and the maximum number of pennies which may be removed at a time, are chosen by the computer in lines 9070-9100. These are always the same at level 0, but vary at the higher skill levels.

The nature of this game is that the person who begins can always win, if he or she makes no mistakes. (Line 9100 assures this.) The challenge of it is to discover a winning strategy, so I'm not too inclined to explain the computer's calculations in lines 9100 and 9140-9160. I will say that line 9160 introduces a random chance of error on the computer's part, giving the human player a chance even if the human doesn't play perfectly. Without this, one mistake would be fatal. If you want to increase the computer's errors, you can increase the quantity .4 in that line to a larger number. For example, increasing it to 1 would make the computer play a completely random game at level 0, a half-random game at level 1, and a one-third-random game at level 2. Decreasing it to 0 would eliminate random errors altogether.

### EVENT #10: BINARY NUMBERS

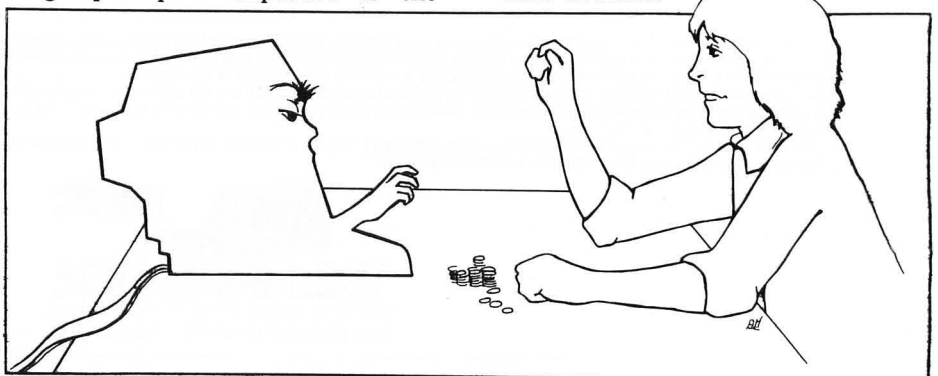
The final event checks out the players on their understanding of binary numbers. Nothing terribly complicated, just binary-decimal conversion and (for more skilled players) binary addition. These exercises may seem to you to be anything from trivial to unfathomable, depending upon prior exposure to the

binary number system. (You can always hit 'ESC' when the instructions are displayed if you get too intimidated.)

The heart of the event is the decimal-to-binary conversion routine in lines 10100-10140. This accepts as input the non-negative decimal integer DEC, and gives as output the string BIN\$, which contains the binary equivalent of DEC. The variable II is the number of binary digits which will be necessary to represent the decimal number. It starts as 1 in line 10100, and is increased to the required number in line 10110. The loop in 10120-10140 builds the string BIN\$ bit by bit using 0s and 1s, and then returns control to the main program.

The rest of the coding consists of three sections which display, respectively, 1) a binary-to-decimal conversion problem, 2) a decimal-to-binary conversion problem, and 3) a binary addition problem. Players at skill level 0 get two passes at the first type and one at the second; other players get one problem of each type. Each section simply chooses a random number (or two), calls the conversion subroutine, and then presents the appropriate type of problem on the screen.

You may wonder why, in scoring, the VALues of the binary strings are compared, rather than just comparing the strings directly. (Tell the truth, now — DID you wonder??) The reason is that this eliminates the significance of any spaces or leading zeroes that a player may have put into an answer. The strings "1010", "01010", and " 1010 " are all different, but they all represent the same binary number; and the VALue function will return the same quantity for each of them.



## THE THRILL OF VICTORY AND THE AGONY OF DEFEAT

The only task remaining is to find out who won and to play the song of the winning player. After a short, reverent pause in line 10800, lines 10810-10840 determine the high score for the ten combined events. Lines 10850-10870 then check each player to see if the score equals that high score, and if so plays the song. (This takes first-place ties right in stride.) The subroutine which actually defines the tunes to be played starts at 10900, and operates in the same way as the one in 11160-11190 which you typed in two months ago.

(You may recall my promise from

previous months to include notes along the way to facilitate translation for other computers. I haven't forgotten that for these last two installments; it's just that there has been very little in the coding which is unique to Applesoft. And the keywords that are unique, such as HOME, VTAB, and HTAB, have already been mentioned in prior months.)

That's about if for "Math Decathlon". I was considering a Language Decathlon a while back, but never have gotten around to it. Maybe you'd want to try your hand at it?

### VARIABLES (Part 4)

B1\$, B2\$, B3\$, BIN\$: Strings con-

taining binary numbers.

D: Used in binary-decimal conversion.

D1, D2, D3, DEC: Decimal numbers for binary conversion.

HP: High points for the game.

I, II: General loop variables.

M, M1; Timing loop counters.

MAX: Maximum number of pennies which may be removed from stack.

N1\$: Input string.

N1-N9: Numbers used to generate Speed Math problems.

NUM: Number of pennies in the pile.

R: Temporary variable.

TB: Tab position for printing.

ZZ: Time delay loop variable.

New lines to replace previous ones.

```
310 E$(1) = "MISSING #";E$(2) = "
    GUESS A #";E$(3) = "SPEED MA
    TH";E$(4) = "STARS";E$(5) =
    "L.C.D.";E$(6) = "MAGIC SOR"
    ;E$(7) = "CLOCK MATH";E$(8) =
    "MENTAL MATH";E$(9) = "PENNI
    ES";E$(10) = "BINARY #'S"
330 EE$(6) = " 6. MAGIC SQUARES";
    EE$(7) = " 7. CLOCK MATH";EE
    $(8) = " 8. MENTAL MATH";EE$
    (9) = " 9. PENNY BATTLE";EE$
    (10) = "10. BINARY NUMBERS"
```

Event #8:

Mental Math

Print instructions.

```
8000 E = 8:NXT = 0
```

```
8010 M1 = 2000
```

```
8020 FOR P = 1 TO NP: GOSUB 10
```

```
8030 PRINT "IN THIS EVENT I WILL
    START BY GIVING YOUA NUMBER
    , AND THEN GIVE YOU A SERIES
    OF MATHEMATICAL OPERATIONS
    TO DO, ONE AFTERTHE OTHER.
    THEN, I'LL ASK YOU FOR THE
    ANSWER."
```

```
8040 PRINT : PRINT "BEWARE!! I'
    LL START OUT SLOWLY, BUT
    THEN I'LL START SPEEDING UP,
    SO BE SURE TO PAY ATTENTION
    !"
```

```
8050 GOSUB 90
```

```
8060 IF NXT THEN 9000
```

Present a series of mathematical operations, one step at a time.

```
8070 FOR PROB = 1 TO 3:N6 = 1:N7
    = 0:N8 = 0:N9 = 1:N1 = INT
    ( RND (1) * 7) + 3: HOME : VTAB
    10: PRINT TAB( 9)"START WIT
    H THE NUMBER ";N1
```

```
8080 FOR M = 1 TO M1: NEXT M:N2 =
    INT ( RND (1) * 3 + 2) * 2:
    HOME : VTAB 10: PRINT TAB(
```

```
14)"MULTIPLY BY ";N2
```

```
8090 FOR M = 1 TO M1 * .9: NEXT
    M:N3 = INT ( RND (1) * 4 +
    2) * 2: HOME : VTAB 10: PRINT
    TAB( 15)"SUBTRACT ";N3
```

```
8100 FOR M = 1 TO M1 * .83: NEXT
    M:N4 = 2: HOME : VTAB 10: PRINT
    TAB( 15)"DIVIDE BY ";N4
```

```
8110 FOR M = 1 TO M1 * .75: NEXT
    M:N5 = INT ( RND (1) * 10) +
    5: HOME : VTAB 10: PRINT TAB(
    17)"ADD ";N5
```

```
8120 FOR M = 1 TO M1 * .67: NEXT
    M: IF S(P) = 0 THEN 8190
```

Continue if skill level is 1 or 2.

```
8130 N6 = INT ( RND (1) * 2) + 2
    : HOME : VTAB 10: PRINT TAB(
    14)"MULTIPLY BY ";N6
```

```
8140 FOR M = 1 TO M1 * .59: NEXT
    M:N7 = INT ( RND (1) * 5) +
    5: HOME : VTAB 10: PRINT TAB(
    17)"ADD ";N7
```

```
8150 FOR M = 1 TO M1 * .50: NEXT
    M: IF S(P) = 1 THEN 8190
```

Continue if skill level is 2.

```
8160 N8 = INT ( RND (1) * 10) +
    3: HOME : VTAB 10: PRINT TAB(
    15)"SUBTRACT ";N8
```

```
8170 FOR M = 1 TO M1 * .4: NEXT
    M:N9 = 2: HOME : VTAB 10: PRINT
    TAB( 14)"MULTIPLY BY ";N9
```

```
8180 FOR M = 1 TO M1 * .3: NEXT
    M
```

Input and evaluate answer.

```
8190 HOME : VTAB 10: PRINT TAB(
    11)"THE ANSWER IS: "; INPUT
    AN$
```

```
8200 A2 = (((N1 * N2 - N3) / N4 +
    N5) * N6 + N7 - N8) * N9:PX =
    10: GOSUB 30
```

```
8210 GOSUB 100: NEXT PROB,P: GOSUB
    11000
```

Event #9:

Penny Battle

Print instructions.

```
9000 E = 9:NXT = 0
```

```
9010 FOR P = 1 TO NP: GOSUB 10
9020 PRINT "IN THIS EVENT YOU AN
    D I WILL TAKE TURNS REMOVING
    PENNIES FROM A STACK, AND T
    HE ONE WHO PICKS UP THE LAS
    T PENNY LOSES."
```

```
9030 PRINT : PRINT "YOU MUST PIC
    K UP AT LEAST ONE PENNY ON
    EACH TURN, BUT NO MORE THAN
    THE MAXIMUM WHICH I'LL SPECI
    FY."
```

```
9040 PRINT : PRINT "I'LL GIVE YO
    U THE ADVANTAGE BY LETTING
    YOU PICK UP FIRST."
```

```
9050 GOSUB 90
```

```
9060 IF NXT THEN 10000
```

Choose size of pile and maximum number which may be removed.

```
9070 FOR PROB = 1 TO 3:NUM = 23:
    MAX = 3
```

```
9080 IF S(P) = 1 THEN NUM = INT
    ( RND (1) * 10 + 20)
```

```
9090 IF S(P) = 2 THEN NUM = INT
    ( RND (1) * 10 + 20):MAX = INT
    ( RND (1) * 4 + 3)
```

```
9100 IF NUM - INT (NUM / (MAX +
    1)) * (MAX + 1) = 1 THEN 908
    0
```

```
9110 HOME : VTAB 3: PRINT "WE'LL
    BEGIN WITH A STACK OF ";NUM
    ;" PENNIES.": PRINT : PRINT
    "YOU MAY PICK UP FROM 1 TO "
    ;MAX;" OF THEM.": PRINT
```

Take turns removing pennies until only one is left.

```
9120 INPUT "YOU TAKE HOW MANY? "
    ;N1$:N1 = VAL (N1$): IF N1 <
    1 OR N1 > MAX THEN 9120
```

continued on next page

continued from previous page

```
9130 NUM = NUM - N1: IF NUM = 1 THEN
  PRINT : PRINT "YOU WIN, ";P
  L$(P);"!";PZ(P,E) = PZ(P,E) +
  10: GOTO 9190
9140 R = NUM - INT (NUM / (MAX +
  1)) * (MAX + 1);N2 = R - 1
9150 IF N2 = - 1 THEN N2 = MAX
9160 IF ( RND (1) * (S(P) + 1) <
  .4 AND NUM > = MAX * 2) OR
  N2 = 0 THEN N2 = INT ( RND
  (1) * MAX + 1)
9170 NUM = NUM - N2: PRINT "I TAK
  E ";N2; ", LEAVING ";NUM: IF
  NUM = 1 THEN PRINT : PRINT
  "I WIN!": GOTO 9190
```



```
9180 PRINT : GOTO 9120
9190 GOSUB 100: NEXT PROB,P: GOSUB
  11000
```

### Event #10: Binary Numbers

Print instructions.

```
10000 E = 10:NXT = 0
10010 FOR P = 1 TO NP: GOSUB 10
10020 PRINT "BINARY NUMBERS ARE
  WRITTEN USING ONLY THE DIG
  ITS 0 AND 1, INSTEAD OF THE
  TEN DIGITS THAT ARE USED IN
  DECIMAL NUMBERS."
10030 PRINT "INSTEAD OF HAVING P
  LACE VALUES OF 1'S, 10'S, 1
  00'S, ETC., BINARY NUMBERS H
  AVE PLACE VALUES OF 1'S, 2'
  S, 4'S, 8'S, ETC."
10040 PRINT "THUS, THE BINARY NU
  MBER 1010 EQUALS TEN --ZERO
  1'S, PLUS ONE 2, PLUS ZERO 4
  'S, PLUS ONE 8."
10050 PRINT : PRINT "IN THIS EVE
  NT YOU WILL HAVE THREE PROB-
  LEMS DEALING WITH BINARY NU
  MBERS."
10060 GOSUB 90
10070 IF NXT THEN 10410
10080 GOTO 10200
```

Subroutine to convert a decimal  
number into a binary number.

```
10100 II = 1:BIN$ = "":D = DEC
10110 IF D > 2 ^ II - 1 THEN II =
  II + 1: GOTO 10110
10120 FOR I = II - 1 TO 0 STEP -
  1: IF D > = 2 ^ I THEN BIN$
  = BIN$ + "1":D = D - 2 ^ I:
  GOTO 10140
10130 BIN$ = BIN$ + "0"
10140 NEXT I: RETURN
```

Display binary-to-decimal  
conversion problem.

```
10200 FOR PROB = 1 TO 2: IF S(P)
  > 0 THEN PROB = 2
10210 HOME : VTAB 8:DEC = INT (
  RND (1) * 12 * 2 ^ S(P)) +
  2 ^ (S(P) + 2): GOSUB 10100
```



```
10220 PRINT "CONVERT THE FOLLOWI
  NG BINARY NUMBER INTO A REGUL
  AR DECIMAL NUMBER!": PRINT :
  PRINT : PRINT TAB( 18)BIN$
```

```
10230 PRINT : PRINT : HTAB 14: INPUT
  "YOUR ANSWER: ";AN$: PRINT :
  PRINT
10240 AZ = DEC:PX = 10: GOSUB 30
10250 GOSUB 100: NEXT PROB
```

Display decimal-to-binary  
conversion problem.

```
10260 HOME : VTAB 8:DEC = INT (
  RND (1) * 12 * 2 ^ S(P)) +
  2 ^ (S(P) + 2): GOSUB 10100
10270 PRINT "CONVERT THE FOLLOWI
  NG DECIMAL NUMBER INTO A
  BINARY NUMBER!": PRINT : PRINT
  : PRINT TAB( 19)DEC
10280 PRINT : PRINT : HTAB 12: INPUT
  "YOUR ANSWER: ";AN$: PRINT :
  PRINT
```

```
10290 AZ = VAL (BIN$):PX = 10: GOSUB
  30
```

```
10300 GOSUB 100
```

Display binary addition problem.

```
10310 IF S(P) = 0 THEN 10400
10320 HOME : VTAB 4:DEC = INT (
  RND (1) * 12 * S(P)) + 4 *
  S(P):D1 = DEC: GOSUB 10100:B
  1$ = BIN$
10330 DEC = INT ( RND (1) * 12 *
  S(P)) + 4 * S(P):D2 = DEC: GOSUB
  10100:B2$ = BIN$:DEC = D1 +
  D2: GOSUB 10100:B3$ = BIN$
10340 PRINT "ADD THE FOLLOWING T
  WO BINARY NUMBERS, GIVING
  THE ANSWER AS ANOTHER BINARY
  NUMBER!": PRINT : PRINT
```

```
10350 TB = LEN (B1$) - LEN (B2$
  )
```

```
10360 PRINT TAB( 18)B1$: PRINT
  : PRINT TAB( 15)"+";: PRINT
  TAB( 18 + TB)B2$: PRINT TAB(
  15)"-----"
```

```
10370 PRINT : PRINT : PRINT : INPUT
  " YOUR ANSWER: ";AN$: PRINT
  : PRINT
```

```
10380 AZ = VAL (B3$):PX = 10: GOSUB
  30
```

```
10390 GOSUB 100
10400 NEXT P
10410 GOSUB 11000
```

End of game. Determine high score  
and play song of winning player(s).

```
10800 FOR ZZ = 1 TO 500: NEXT ZZ
```

```
10810 HF = 0
10820 FOR P = 1 TO NP
10830 IF PZ(P,0) > HP THEN HP =
  PZ(P,0)
10840 NEXT P
10850 FOR II = 1 TO NP
10860 IF PZ(II,0) = HP THEN GOSUB
  10900
10870 NEXT II
10880 END
```

Subroutine to play complete  
national song(s).

```
10900 IF C(II) = 1 THEN SKIP = 0
  :PLAY = 43
10910 IF C(II) = 2 THEN SKIP = 4
  3:PLAY = 102
10920 IF C(II) = 3 THEN SKIP = 1
  45:PLAY = 116
10930 IF C(II) = 4 THEN SKIP = 2
  61:PLAY = 53
10940 GOSUB 13000: RETURN
```





# THE NATIONAL COMPUTER SHOWS HAVE WE GOT A PROGRAM FOR YOU IN '81

Attend the biggest public computer shows in the country. Each show has 100,000 square feet of display space featuring over 50 Million Dollars worth of software and hardware for business, industry, government, education, home and personal use.

You'll see computers costing \$150 to \$250,000 including mini and micro computers, software, graphics, data and word processing equipment, telecommunications, office machines, electronic typewriters, peripheral equipment, supplies and computer services.

All the major names are there including; IBM, Wang, DEC, Xerox, Burroughs, Data General, Qantel, Nixdorf, NEC, Radio Shack, Heathkit, Apple, RCA, Vector Graphic, and Commodore Pet. Plus, computerized video games, robots, computer art, electronic gadgetry, and computer music to entertain, enthrall and educate kids, spouses and people who don't know a program from a memory disk.

Don't miss the Coming Of The New Computers—Show Up For The Show that mixes business with pleasure. Admission is \$5 for adults and \$2 for children under 12 when accompanied by an adult.

## Ticket Information

Send \$5 per person with the name of the show you will attend to National Computer Shows, 824 Boylston Street, Chestnut Hill, Mass. 02167. Tel. 617 739 2000. Tickets can also be purchased at the show.

## THE SOUTHWEST COMPUTER SHOW

**DALLAS**  
Dallas Market Hall  
2200 STEMMONS FRWY  
AT INDUSTRIAL BLVD

**THURS-SUN**  
**APRIL 9-12**  
**10 AM TO 7 PM**

## THE MID-WEST COMPUTER SHOW

**CHICAGO**  
McCormick Place  
SCHOESSLING HALL  
23RD & THE LAKE

**THURS-SUN**  
**SEPTEMBER 10-13**  
**10 AM TO 7 PM**

## THE MID-ATLANTIC COMPUTER SHOW

**WASHINGTON, DC**  
DC Armory/Starplex  
2001 E. CAPITOL ST. SE  
(E CAP ST EXIT OFF I295  
— KENILWORTH FRWY)  
ACROSS FROM RFK STADIUM

**THURS-SUN**  
**SEPTEMBER 24-27**  
**10 AM TO 7 PM**

## THE NORTHEAST COMPUTER SHOW

**BOSTON**  
Hynes Auditorium  
PRUDENTIAL CENTER

**THURS-SUN**  
**OCTOBER 15-18**  
**10 AM TO 7 PM**

## THE SOUTHEAST COMPUTER SHOW

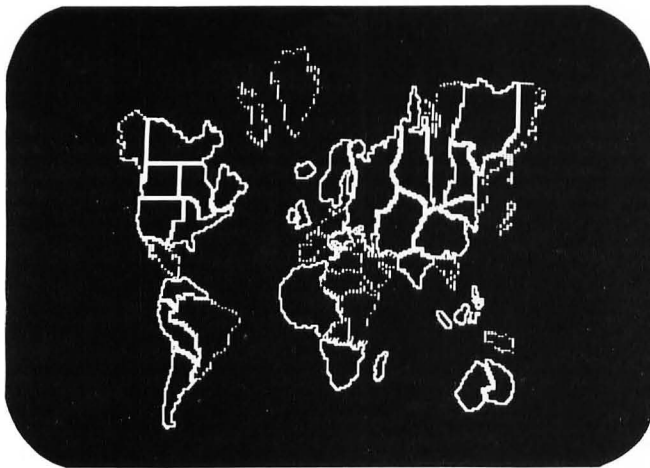
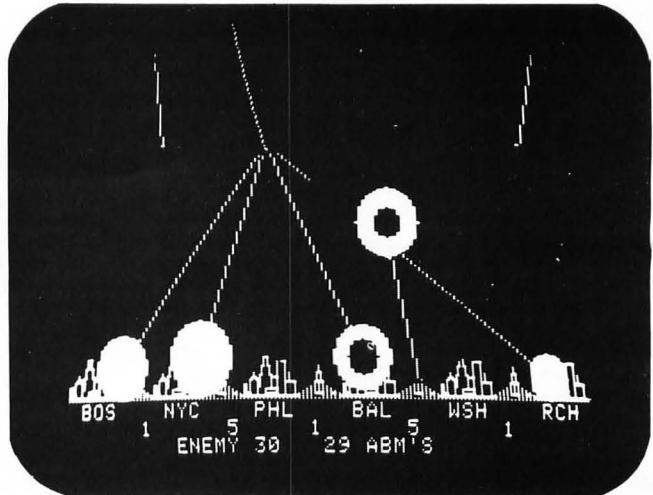
**ATLANTA**  
Atlanta Civic Center  
395 PIEDMONT AVE NE AT  
RALPH MCGILL BLVD

**THURS-SUN**  
**OCTOBER 29-NOVEMBER 1**  
**10 AM TO 7 PM**

# TWO EXCITING GAMES FOR THE APPLE II® OR II plus®

## ABM by silas warner

Invader and Asteroids move over... ABM has arrived! Command your launch sites to fire 1 and 5 kiloton anti-ballistic missiles (ABMs). Save the East Coast from increasingly fierce Enemy nuclear attack. Position your target crosshairs to blast the green streamers before they fire—ball your cities—or worse—split into multiple warhead MIRVs turning the entire coast into a thundering specter of destruction. Hi-res color graphics, sound, high score to date memory, paddle or joystick control. On disk, requires Applesoft ROM. (\$24.95)



## GLOBAL WAR by alan boyd

A challenging strategy game for 2 to 9 players. Your 'War Room' features a detailed map of the earth plus territorial occupation status and invasion strategy displays. Players are assigned armies and territories and the war begins. The computer battle simulator calculates the results of each invasion. Occupy all countries and you are the Winner! Games may be saved on disk to be continued later. On disk, requires 48K and Applesoft ROM. (\$24.95) cassette (\$17.95)

**available now at your local computer store**

**MUSE** SOFTWARE™

Apple II is a trademark of Apple Computer Corp.

**TSE-HARDSIDE**



6 South St., Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790

# CALENDAR

June 6 and 7

APPLEFEST '81

The Plaza Castle, Boston, MA

Largest event in the world for Apple users. Exhibits applications, and seminar program.

Contact: The Boston Computer Society's Apple User Group, 3 Center Plaza, Boston, MA 02108 (617-367-8080).

June 6-9

The Atlantic Small Computer Show

The Atlanta Hilton, Atlanta, GA

Exhibits and programs. Producers of small computers, peripherals, supplies, and services will be exhibiting.

Contact: Professional Exposition Development Corp., 4060 Janice Drive, Suite-C1, East Point, GA 30344 (404-767-9798).

June 9-11

Understanding and Using Computer Graphics

Chicago, IL.

Seminar covering the latest in graphics-system technology — hardware and software applications.

Contact: Bob Sanzo, Frost & Sullivan Inc., 106 Fulton St., New York, NY 10038, (212-233-1080).

June 14-18

The Second National Conference of the National Computer Graphics Assoc.

Demonstrations, exhibits, and workshops.

Contact: National Computer Graphics Assoc., Inc., 2033 M. Street NW, Suite 330, Washington, DC 20036 (202-466-5895).

June 16-18

NEPCON East '81

New York Coliseum, New York, NY

Aimed at engineers, prototype developers, production specialists and testing personnel. Technical programs will be presented.

Contact: Conference Management Inc., 222 W. Adams St. Chicago, IL 60606 (312-263-4866).

June 17-19

National Educational Computing Conference

North Texas State University, Denton, TX

Forum for individuals and institutions interested in educational computing. Computer literacy, teacher-computer education, computers in education, etc. are topics covered.

Contact: Dr. Jim Poirot, NECC-81 General Chairman, Computer Sciences Dept., North Texas State University, Denton, TX 76203

June 20-22

The Fifth Annual Computerfest

Franklin University, Columbus, OH

Talks on robots, calculators, microcomputers, and small business systems are presented.

Contact: Computerfest '81, Paul Pittenger, 215 Delhi Ave., Apt. J, Columbus, OH 43202 (614-224-6237).

June 24-26

Computer Industry Trade Expo

Atlantic City Convention Center, Atlantic City, N.J.

Exhibits and conference program.

Contact: C.W. Conference Management Group, 375 Cochituate Rd., Rt. 30, Framingham, MA 01701 (800-225-4698).

June 29-July 1

The Nineteenth Annual Meeting of the Association for

Computational Linguistics

Stanford University, Stanford, CA

Computational semantics; discourse analysis and speech acts;

speech analysis and synthesis, machine and machine aided

translation; and mathematical foundations of computational

linguistics are some of the topics to be discussed.

Contact: Don Walker, Artificial Intelligence Center, SRI International, Menlo Park, CA 94025 (415-326-6200)



written by  
David Bolke

# Sports fans

## MASTERS GOLF

Tired of golf as a sport you can only play during the warmer half of the year? Now you can play it year 'round with MASTERS GOLF! For the Atari and the S-80, MASTERS GOLF promises hours and hours of delight.

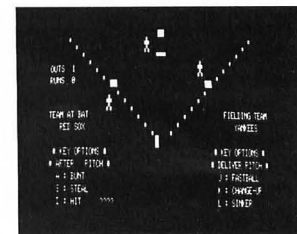
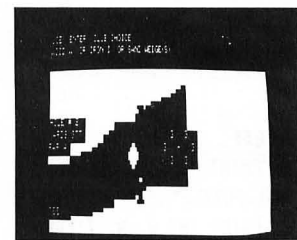
(S-80 version originally published as "Protour Golf")

S-80 Cassette #26-104004C ..... \$9.95  
Atari Cassette #36-104000C ..... \$9.95

## WORLD SERIES

Ah yes, it's springtime and a young man's thoughts turn to...baseball? How would you like a baseball season with no threats of player strikes or free agent negotiations that leave the shattered remnants of once-mighty teams strewn about the playing fields? We offer you your own league. Batter up!

S-80 Cassette #26-104005C ..... \$9.95  
Atari Cassette #36-104005C ..... \$9.95  
Apple Cassette #47-104005C ..... \$9.95  
Apple Disk #47-104005D ..... \$14.95



**TSE-HARDSIDE**  
6 South St. Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790



Enjoy two of your favorites  
without leaving home!



# HEXADECIMAL PUZZLE

by George W. Ziegler, Jr.

**"Hexadecimal Puzzle" is an Apple program requiring Applesoft and 16K RAM.**

It wasn't many years ago, that only the VERY elite few had ever heard the word "hexadecimal", let alone understood what a base-16 number system was all about. Today it is becoming almost second nature for many who are involved in personal computing to represent numbers in hexadecimal form rather than in decimal. What could be more natural, then, than a puzzle based on hexadecimal digits?

For those who are relatively new to this whole thing and may not yet be among the elite, a brief word of explanation is in order. "Hexadecimal", which is often shortened to "hex", comes from two words that mean "six" and "ten". A logical name for a base-16 number system, no? Instead of 10 digits (0-9), the hexadecimal number system has 16: 0-9, plus the "digits" A-F which represent the numbers 10 through 15.

When you add a second digit, for representing numbers greater than 15 (as you must do in the regular

decimal system to represent numbers greater than nine), the higher-order digit then represents the "16's place" (rather than the 10's place as in decimal). So, for example, the hex number 2C represents two 16s plus 12 (C) 1s, or 44. Additional digits have place values of 256, 4096, and successively higher multiples of 16.

You don't really have to understand hexadecimal place values, though, to enjoy this puzzle game — just the basic digits 0 through F. At the beginning of the game you are confronted with a four-by-four square color graphics display of these digits, in random order, with a blank space in place of the F. The object is to move the digits adjacent to the blank space into that space, in such a manner as to rearrange all the digits into their proper order: 0-3 across the first row, 4-7 across the second, and so on, leaving the blank space at the bottom right. When you accomplish this, you will be rewarded with the appearance of the elusive sixteenth digit in its proper place, and an account of the number of moves it took you to solve the puzzle.

The current arrangement of the digits is stored in the computer's

memory in array H(R,S). These positions are translated into X, Y screen coordinates for use by the drawing subroutines in lines 698-715. If you try to move a digit that is not adjacent to the open square, or try to enter a character that is not a hexadecimal digit, you'll get a beep and an "illegal entry" message. During play, if you find yourself getting desperate with frustration, you can (a) kick the dog, (b) kick the computer, (c) hysterically press RESET, or (d) calmly press ESC.

## VARIABLES

- D\$: Current hexadecimal digit.
- D: Decimal equivalent of current hexadecimal digit.
- H (\*,\*) : Array representing arrangement of digits.
- I, J: Loop counters.
- N: Counter for number of moves.
- M\$: General-purpose string input variable.
- R, S: Subscripts representing horizontal and vertical locations in the H array.
- X, Y: Horizontal and vertical screen locations.

<pre> HEXADECIMAL PUZZLE  Program Listing  Title and instructions.  11 HOME : TEXT 12 VTAB 6: PRINT "      HEX   ADECIMAL PUZZLE" 13 PRINT : PRINT "BY G. W. ZIEGL   ER, JR.      APRIL 1981" 14 PRINT : PRINT : PRINT : PRINT   "THE OBJECT OF THE GAME IS T   O MOVE EACH" 15 PRINT "OF THE HEXADECIMAL DIG   ITS (0 TO E) INTO" 16 PRINT "THE BLANK SQUARE IN SU   CH A MANNER AS TO"  17 PRINT "REARRANGE THE DIGITS I   NTO ASCENDING" 18 PRINT "ORDER. IF YOU ACCOMPL   ISH THIS THE ELU-": PRINT "S   IVE SIXTEENTH DIGIT (F) WILL   APPEAR.": PRINT "TO END GAM   E AT ANY TIME HIT (ESC)."</pre>	<pre> 19 VTAB 24: PRINT "      (PRESS   ANY KEY TO BEGIN)"; GET M\$  Draw the playing board.  20 CLEAR : HOME : TEXT 25 LET N = 0 30 GR : COLOR= 4 40 FOR I = 0 TO 4: HLIN 7,31 AT   I * 9: NEXT I 50 FOR I = 0 TO 4: VLIN 0,36 AT   7 + (6 * I): NEXT I  Initialize the array.  60 DIM H(5,5),HS(16) 70 FOR J = 1 TO 4: FOR I = 1 TO   4 80 LET H(I,J) = 15 90 NEXT I: NEXT J  Generate a random puzzle.</pre>	<pre> 100 VTAB 22: PRINT "I'M CREATING   YOUR RANDOM PUZZLE NOW. . .   " 110 FOR D = 0 TO 14 120 LET R = INT ( RND (1) * 4 +   .5): LET S = INT ( RND (1) *   4 + .5) 130 IF H(R,S) &lt; &gt; 15 THEN 120 140 LET H(R,S) = D 170 NEXT D  Check puzzle to see if it can be solved.  172 LET S = 0 174 FOR J = 1 TO 4: FOR I = 1 TO   4 176 LET HS((J - 1) * 4 + I) = H(   I,J) 178 NEXT I: NEXT J 180 FOR I = 1 TO 15: FOR J = I +   1 TO 16 182 IF HS(I) &gt; HS(J) THEN LET S   = S + 1 184 NEXT J: NEXT I</pre>
---	---	---

```

186 IF HS(2) = 15 OR HS(4) = 15 OR
HS(5) = 15 OR HS(7) = 15 OR
HS(10) = 15 OR HS(12) = 15 OR
HS(13) = 15 OR HS(15) = 15 THEN
LET S = S + 1
188 LET HS(0) = INT (S / 2): IF
HS(0) * 2 < > S THEN GOTO
20

```

Display puzzle on the screen.

```

190 HOME
200 FOR S = 1 TO 4: FOR R = 1 TO
4: LET D = H(R,S): IF D = 16
THEN 206
202 LET X = 8 + ((R - 1) * 6): LET
Y = 1 + ((S - 1) * 9)
204 IF D < 15 THEN GOSUB 699: ON
D + 1 GOSUB 700,701,702,703,
704,705,706,707,708,709,710,
711,712,713,714,715
206 NEXT R: NEXT S

```

Input a character and check for legality.

```

210 VTAB 21: PRINT " ENTER H
EX DIGIT FROM 0 TO E : ";
212 GET D$: LET D = ASC (D$): IF
D = 27 THEN GOTO 500
215 IF D > 64 AND D < 70 THEN LET
D = D - 48 - 7: GOTO 250
220 IF D > 47 AND D < 58 THEN LET
D = D - 48: GOTO 250
230 PRINT CHR$ (7): GOTO 210

```

Find the character in the array.

```

250 PRINT : FOR S = 1 TO 4: FOR
R = 1 TO 4
260 IF H(R,S) = D THEN GOTO 310
270 NEXT R: NEXT S

```

Check for empty adjacent square.

```

310 IF H(R - 1,S) = 15 THEN X =
R - 1:Y = S: GOTO 360
320 IF H(R + 1,S) = 15 THEN X =
R + 1:Y = S: GOTO 360
325 IF H(R,S - 1) = 15 THEN X =
R:Y = S - 1: GOTO 360
330 IF H(R,S + 1) = 15 THEN X =
R:Y = S + 1: GOTO 360
340 HOME : VTAB 21: PRINT "
ILLEGAL MOVE - REENTER: ";
CHR$ (7): GOTO 212

```

Move character to new location and clear old location.

```

360 LET H(X,Y) = D: LET H(R,S) =
15

```

```

370 LET X = 8 + ((X - 1) * 6): LET
Y = 1 + ((Y - 1) * 9): GOSUB
699: ON D + 1 GOSUB 700,701,
702,703,704,705,706,707,708,
709,710,711,712,713,714,715
380 LET X = 8 + ((R - 1) * 6): LET
Y = 1 + ((S - 1) * 9): GOSUB
698
390 LET N = N + 1

```

Check for solution.

```

410 FOR J = 1 TO 4: FOR I = 1 TO
4
420 IF H(I,J) < > ((4 * (J - 1)
) + I - 1) THEN GOTO 210
430 NEXT I: NEXT J
440 LET X = 26: LET Y = 28: GOSUB
699: GOSUB 715
450 VTAB 21: PRINT CHR$ (7);"
Y O U W I N !
"
452 PRINT " IT TOOK YOU
";N;" MOVES."
455 PRINT " ANOTHER GAME
(Y OR N)? "; GET M$: IF M$
= "Y" THEN GOTO 20
460 IF M$ < > "N" THEN VTAB 24
: HTAB 1: GOTO 455
500 TEXT : HOME : VTAB 14: PRINT
" GAME OVER": VTAB
24: END

```

Subroutine to clear box.

```

698 COLOR= 0: FOR J = 0 + Y TO 7
+ Y: HLINE 0 + X,4 + X AT J:
NEXT J: RETURN

```

Subroutine to color-in box.

```

699 COLOR= 9: FOR J = 0 + Y TO 7
+ Y: HLINE 0 + X,4 + X AT J:
NEXT J: RETURN

```

Subroutines to print digits 0 through E.

```

700 COLOR= 0: PLOT 2 + X,1 + Y: VLINE
2 + Y,5 + Y AT 1 + X: VLINE 2
+ Y,5 + Y AT 3 + X: PLOT 2 +
X,6 + Y: RETURN
701 COLOR= 0: PLOT 1 + X,2 + Y: VLINE
1 + Y,5 + Y AT 2 + X: HLINE 1
+ X,3 + X AT 6 + Y: RETURN
702 COLOR= 0: PLOT 2 + X,1 + Y: PLOT
1 + X,2 + Y: PLOT 3 + X,2 +
Y: PLOT 3 + X,3 + Y: PLOT 2 +
X,4 + Y: PLOT 1 + X,5 + Y: HLINE
1 + X,3 + X AT 6 + Y: RETURN

```

```

703 COLOR= 0: HLINE 1 + X,3 + X AT
1 + Y: PLOT 3 + X,2 + Y: PLOT
2 + X,3 + Y: VLINE 4 + Y,6 +
Y AT X + 3: HLINE 1 + X,2 + X
AT 6 + Y: RETURN
704 COLOR= 0: VLINE 1 + Y,3 + Y AT
1 + X: PLOT 2 + X,3 + Y: VLINE
1 + Y,6 + Y AT 3 + X: RETURN

```

```

705 COLOR= 0: HLINE 1 + X,3 + X AT
1 + Y: VLINE 2 + Y,3 + Y AT 1
+ X: PLOT 2 + X,3 + Y: VLINE
4 + Y,5 + Y AT 3 + X: HLINE 1
+ X,2 + X AT 6 + Y: RETURN

```

```

706 COLOR= 0: PLOT 2 + X,1 + Y: VLINE
1 + Y,6 + Y AT 1 + X: HLINE 2
+ X,3 + X AT 6 + Y: VLINE 3 +
Y,5 + Y AT 3 + X: PLOT 2 + X
,3 + Y: RETURN

```

```

707 COLOR= 0: HLINE 1 + X,3 + X AT
1 + Y: VLINE 2 + Y,3 + Y AT 3
+ X: PLOT 2 + X,4 + Y: VLINE
5 + Y,6 + Y AT 1 + X: RETURN

```

```

708 COLOR= 0: PLOT 2 + X,1 + Y: PLOT
1 + X,2 + Y: PLOT 3 + X,2 +
Y: PLOT 2 + X,3 + Y: VLINE 4 +
Y,5 + Y AT 1 + X: VLINE 4 + Y
,5 + Y AT 3 + X: PLOT 2 + X,
6 + Y: RETURN

```

```

709 COLOR= 0: VLINE 1 + Y,3 + Y AT
1 + X: VLINE 1 + Y,6 + Y AT 3
+ X: PLOT 2 + X,1 + Y: PLOT
2 + X,3 + Y: PLOT 2 + X,6 +
Y: RETURN

```

```

710 COLOR= 0: PLOT 2 + X,1 + Y: VLINE
2 + Y,6 + Y AT 1 + X: VLINE 2
+ Y,6 + Y AT 3 + X: PLOT 2 +
X,4 + Y: RETURN

```

```

711 COLOR= 0: VLINE 1 + Y,6 + Y AT
1 + X: PLOT 2 + X,1 + Y: PLOT
3 + X,2 + Y: PLOT 2 + X,3 +
Y: VLINE 4 + Y,5 + Y AT 3 + X
: PLOT 2 + X,6 + Y: RETURN

```

```

712 COLOR= 0: HLINE 1 + X,3 + X AT
1 + Y: VLINE 2 + Y,5 + Y AT 1
+ X: HLINE 1 + X,3 + X AT 6 +
Y: RETURN

```

```

713 COLOR= 0: VLINE 1 + Y,6 + Y AT
1 + X: PLOT 2 + X,1 + Y: VLINE
2 + Y,5 + Y AT 3 + X: PLOT 2
+ X,6 + Y: RETURN

```

```

714 COLOR= 0: HLINE 1 + X,3 + X AT
1 + Y: VLINE 2 + Y,5 + Y AT 1
+ X: HLINE 1 + X,3 + X AT 6 +
Y: PLOT 2 + X,3 + Y: RETURN

```

```

715 COLOR= 0: HLINE 1 + X,3 + X AT
1 + Y: VLINE 2 + Y,6 + Y AT 1
+ X: PLOT 2 + X,3 + Y: RETURN

```



# MAZE SEARCH

Translated by Carl Mueller from David Bohlke's Atari original.

**"Maze Search" requires Apple Integer BASIC with Programmer's Aid ROM Chip and 16K RAM.**

Those of us who don't own Ataris have been missing out on a good game ever since "Maze Search" appeared last August in **SoftSide**. Now Apple owners can also be challenged by this original game.

The game is different from other maze games in that your object is not to begin at one place in the maze and work your way through to another place. Rather, you accumulate points by capturing blocks which are placed at random locations within the corridors. The computer first generates a random maze on the full low-resolution graphics screen, and then places 48 blocks within it. You are placed in the middle of the maze, and must travel through the passageways capturing blocks. Each time a block is captured (run over) it is added to your pile at the left side of the screen.

Meanwhile, on the right side of the screen, a growing line marks the time elapsed in the game. You'll find it VERY difficult to capture all 48 blocks in the allotted time, especially since each section of the maze has only one opening onto the outer corridor. You've really got to keep moving to get a good score.

You make your moves by using the I, J, K, and M keys for up, left, right, and down. (These are the same keys used for screen editing in Applesoft.) You can also use the REPT key along with one of these four, to cover distances quickly in any given direction. After the time is up (or, wonder of wonders, you capture all 48 blocks), pressing RETURN will start another game. RETURN will also interrupt the drawing of a maze, or a game in progress.

Incidentally, the program as written makes use of the built-in sound routines in the Programmer's Aid #1 ROM chip. You probably have this chip, unless you have an early Apple. But if you don't, your program will crash at the first "CALL-10473", in line 290. In this event, you must

either eliminate those CALLs and do without sound, or add a short Machine Language sound routine such as the one used in "Math Decathlon" or other programs. Then, instead of POKEing the pitch and length of the note into addresses 767 and 766, and CALLing -10473, you would use the addresses appropriate to your added routine.

## VARIABLES

C: Counter used in generating maze.  
C(\*): Array of color numbers.  
CT: Elapsed time counter.  
DX, DY: X and Y directions (1, 0, or -1).  
H: Horizontal coordinate.  
H(\*): Array used in generating maze.  
HT: Number of blocks captured.  
I: Loop variable.  
S: ASCII value of keyboard input character.  
V: Vertical coordinate.  
V(\*): Array used in generating maze.  
X, Y: Position coordinates.  
Z: Color of screen at a particular point.

### Initialization.

```
0 POKE 765,32
1 POKE 204, PEEK (74): POKE 205
  , PEEK (75): GOTO 100
```

### Subroutines to change location coordinates.

```
10 Y=Y-1: RETURN
11 X=X-1: RETURN
12 X=X+1: RETURN
13 RETURN
14 Y=Y+1: RETURN
```

### Initialize full-screen graphics, draw border, dimension variables.

```
100 POKE -16302,0: POKE -16304,
  0: CALL -1998
110 POKE 766,2
130 COLOR=12: HLIN 4,36 AT 0: VLIN
  0,46 AT 36: HLIN 4,36 AT 46
  : VLIN 0,46 AT 4
140 DIM H(600),V(600)
150 C=0: X=20: Y=24
160 PLOT X,Y
```

### Loop to construct maze.

```
170 IF SCRN(X+2,Y)=0 THEN 220
180 IF SCRN(X-2,Y)=0 THEN 220
```

```
190 IF SCRN(X,Y+2)=0 THEN 220
200 IF SCRN(X,Y-2)=0 THEN 220
210 GOTO 320
```

```
220 GOTO 230+ RND (4)*10
230 DX=-1: DY=0: GOTO 270
240 DX=0: DY=1: GOTO 270
```

```
250 DX=1: DY=0: GOTO 270
260 DX=0: DY=-1
270 IF SCRN(X+DX*2,Y+DY*2) THEN
  220
```

```
280 PLOT X+DX,Y+DY: PLOT X+DX*2
  ,Y+DY*2
290 POKE 767,X/2+Y/2+1: CALL -10473
```

```
300 IF PEEK (-16384)<128 THEN 310
  : POKE -16368,0: IF PEEK (-
  16384)=13 THEN 1
310 X=X+DX*2: Y=Y+DY*2: C=C+1: H(C)
  =X: V(C)=Y: GOTO 170
```

```
320 X=H(C): Y=V(C): C=C-1
330 IF C=0 THEN 350
340 GOTO 170
```

### Plot random blocks within maze.

```
350 POKE 204, PEEK (74): POKE 205
  , PEEK (75)
360 H=19: V=23
```

```
370 DIM C(4): C(0)=13: C(4)=11
380 C(3)=1
```

```
390 C(2)=6: COLOR=C(2): PLOT H,
  V
400 C(1)=12: FOR I=1 TO 48: GOSUB
  590: NEXT I: POKE -16368,0
```

Check for keypress. If RETURN, start over; if I, J, M, or K then move in appropriate direction.

```
410 S= PEEK (-16384): IF S<128 THEN
  530
420 POKE -16368,0: IF S=141 THEN
  1
```

```

430 IF S<201 OR S>205 THEN 530
440 X=H:Y=V: GOSUB S-191
450 COLOR=C(0): PLOT H,V:Z= SCRN(
  X,Y)

```

If can't move in that direction,  
skip move/score section.

```
460 IF Z=C(1) THEN 520
```

If block captured, make sound and  
plot scoring block.

```

470 IF Z#C(3) THEN 480:HT=HT+1:
  POKE 766,3: FOR I=1 TO 20:
  POKE 767,I+30: CALL -10473
  : NEXT I: IF HT=48 THEN 550

```

```

480 IF HT=0 THEN 510
490 COLOR=C(3): IF HT MOD 2=0 THEN
  PLOT 1,47-HT

```

```

500 IF HT MOD 2 THEN PLOT 2,47-
  HT
510 H=X:V=Y

```

Plot current position.

```
520 COLOR=C(2): PLOT H,V
```

Add to timer.

```

530 CT=CT+2: COLOR=C(2): HLIN 38
  ,39 AT 46-CT/100: IF CT<4600
  THEN 540: HLIN 38,39 AT 0:
  GOTO 550

```

```
540 POKE 766,1: POKE 767,H/2+V/
  2+1: CALL -10473: GOTO 410
```

Time is up, or all blocks captured.

```
550 POKE -16368,0
560 IF PEEK (-16384)#141 THEN 580
```

```

570 POKE -16368,0: GOTO 1
580 POKE 766,2: POKE 767, RND (
  48)+1: CALL -10473: GOTO 560

```

Subroutine to plot a random block.

```

590 COLOR=C(3):X= RND (16)*2+5:
  Y= RND (23)*2+1
600 IF SCRN(X,Y) THEN 590
610 PLOT X,Y: RETURN

```



## APPLE ONE LINERS

Applesoft

```

1 S$ = "2021380322442430440001920
  16246044016192096": FOR I =
  0 TO 13: POKE 800 + I, VAL (
  MID$(S$,3 * I + 1,3)): NEXT
  : HGR : HOME : VTAB 22: HTAB
  10: PRINT "HIT ANY KEY TO QU
  IT": CALL 800: TEXT : HOME

```

Jerry Dubnoff  
Newton, MA

Applesoft

```

1 HGR : POKE - 16302,0: FOR I =
  0 TO 3E6: HCOLOR= 7:X = INT
  ( RND (1) * 242) + 15:Y = INT
  ( RND (1) * 162) + 15:R = 6 +
  RND (1) * 8: FOR J = 0 TO 6
  ,28318 STEP ,2: H$PLOT SIN (
  J) * R + X, COS (J) * R + Y:
  NEXT J,I

```

Peter Olszowka  
& Hank Neeman

Applesoft

```

1 GR : FOR T = 1 TO 10000:A = 0:
  B = 39: FOR I = 0 TO 39: COLOR=
  RND (1) * 16: HLIN A,B AT I
  : VLIN A,B AT I: HLIN A,B AT
  39 - I: VLIN A,B AT 39 - I:A
  = A + 1:B = B - 1: NEXT : NEXT

```

Kenneth Baker  
Meridian, ID

# Dragonquest!



From

## The Programmer's Guild

In a desperate race against the sun you search for SMAEGOR Monarch of Dragonfolk, who has kidnapped the Princess of the Realm and holds her in a distant and unknown place. In a quest for honor and glory, you must search the land, seeking out the tools needed for the ultimate confrontation. On the river delta, in the abandoned Temple of Baathteski, Goddess of the Blade, everywhere, clues abound. But WHERE is the Princess?

Now, as never before, the genius of CHARLES FORSYTHE shines in this new Machine Language ADVENTURE. DRAGONQUEST! Can YOU save M'lady from the iron clutches of SMAEGOR?

S-80 Level II 16K Cassette #26-221001T . . . \$15.95  
S-80 32K Disk #26-221001D . . . . . \$21.95

  
**TSE-HARDSIDE**  
6 South St. Milford, NH 03055 (603)673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790



# COMPUTER GRAPHICS

## Variation in Pattern Generation Part I

by Joan R. Truckenbrod

Regular patterns are created by repeating a figure at equal intervals, in a series of parallel horizontal or vertical rows, as discussed and illustrated in the May issue of **SoftSide**. These regular patterns were created by placing a figure in each box of a square grid, similar to graph paper. The use of the computer as an aid in creating patterns allows the designer or artist to introduce different types of variation into the pattern program to create different patterns. In creating patterns in this manner, two basic elements can be varied simultaneously or independently to change the character of a pattern. These two elements are the ordering system of the pattern or the underlying grid like the graph paper, and the design of the original figure or pattern element. This article and my next one will illustrate various ways to change the underlying grid or ordering structure in order to create new patterns. Variation in the character of the pattern element will be discussed in future issues.

The grid underlying a pattern can be varied in many ways as the squares in the grid can be juxtaposed in numerous positions. A checkerboard pattern is one example in which every other square is deleted. Alternate squares on the grid can be shifted up, down, left, right or moved on a diagonal. These modified grids are used to create varied patterns as a figure is drawn in each invisible square in the grid. Various visual effects can be created by using modified grids for generating patterns. The two grids shown here are examples of variations that can be developed in the underlying grids. The first grid shifts alternate squares a half unit vertically or horizontally to create clusters of figures around small empty squares. The second grid has overlapping squares. Each square overlaps the corners of two other squares. This second grid creates strong diagonal sets of figures in the pattern. Use of different figures will create dynamically different patterns as is illustrated by the range of patterns shown here, each created with its respective grid.

The graphics programs used to

create these two sets of patterns are very similar to the original pattern-generation program listed in the May issue. Changes are made in this original program to accommodate the different starting points for the figures in different rows, and the difference in incremental values throughout the pattern. In each of these examples, the pattern is constructed in sets of four rows. The beginning point for each row in this set is different so the starting value for each row is kept in the S4 array. An independent counter L is used to step through this array as the main loops in the program are used to construct and repeat each horizontal row. The S4 array is dimensioned in

line 11 with the values assigned in READ and DATA statements in lines 12 and 13. The counter L is set to 1 in line 85, and incremented and tested in lines 192 through 194. In addition, the H PLOT statements in lines 140 and 170 are changed in each pattern variation program because the X and Y coordinate values defining the figure are incremented differently according to the specific arrangement of the squares in the underlying grid. These incremental values can be figured out by drawing the desired grid on graph paper and calculating the increments necessary to create the pattern. Additional pattern variations, with the programs, will be illustrated in the next issue.

### PATTERN GENERATION PROGRAM

#### Option 1

```
5 REM PATTERN GENERATION PROGRAM
6 REM OPTION 1
7 REM BY JOAN R. TRUCKENBROD
10 REM S4 ARRAY STORES THE STARTING POINT FOR EACH LINE
11 DIM S4(5)
12 FOR I = 1 TO 5: READ S4(I): NEXT I
13 DATA 20,0,30,10,40
20 DIM X(50),Y(50)
30 REM NP = NUMBER OF POINTS IN THE FIGURE
40 NP = 5
50 FOR I = 1 TO NP
60 READ X(I),Y(I)
70 NEXT I
80 DATA 0,0,20,0,20,0,20,0,0
84 REM L IS THE COUNTER FOR THE S4 ARRAY
85 L = 1
86 HGR2 : HCOLOR= 3
90 REM THE R LOOP KEEPS COUNT OF THE VERTICAL ROW NUMBER
100 FOR R = 0 TO 16
110 REM THE B LOOP KEEPS COUNT OF THE HORIZONTAL COLUMN NUMBER
120 FOR B = 0 TO 4
130 REM MOVE THE PEN TO THE FIRST POINT IN THE FIGURE
140 H PLOT X(1) + S4(L) + B * 50,Y(1) + R * 10
150 REM THE FOLLOWING LOOP CONNECTS THE REMAINING POINTS IN THE FIGURE.
160 FOR P = 2 TO NP
170 H PLOT TO X(P) + S4(L) + B * 50,Y(P) + R * 10
180 NEXT P
190 NEXT B
191 REM L IS THE COUNTER FOR THE S4 ARRAY
192 L = L + 1
193 IF L < 5 THEN GOTO 200
194 L = 1
200 NEXT R
210 END
```

#### Option 2

CHANGE THE FOLLOWING LINES IN THE ORIGINAL PROGRAM

```
11 DIM S4(5)
12 FOR I = 1 TO 4: READ S4(I): NEXT I
13 DATA 20,10,0,30

130 REM MOVE THE PEN TO THE FIRST POINT IN THE FIGURE
140 H PLOT X(1) + S4(L) + B * 40,Y(1) + R * 10
150 REM THE FOLLOWING LOOP CONNECTS THE REMAINING POINTS IN THE FIGURE
160 FOR P = 2 TO NP
170 H PLOT TO X(P) + S4(L) + B * 40,Y(P) + R * 10

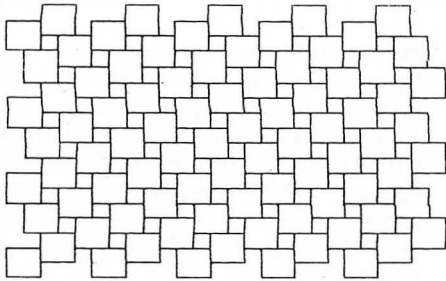
192 L = L + 1
193 IF L < 5 THEN GOTO 200
194 L = 1
```

continued on next page

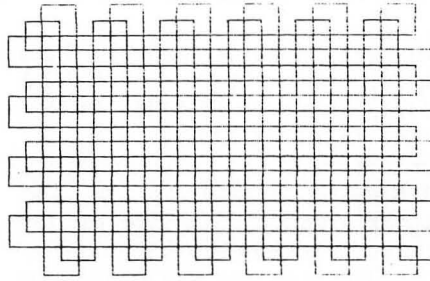


continued from previous page

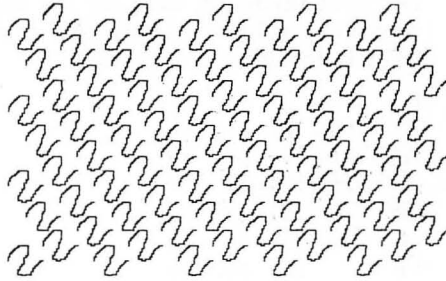
Pattern Option 1



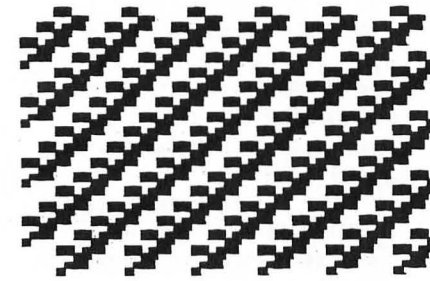
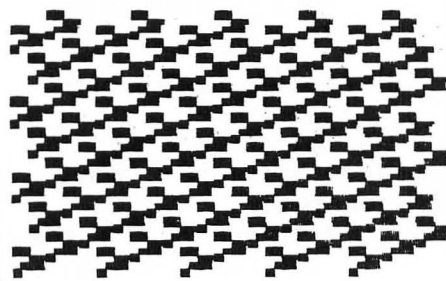
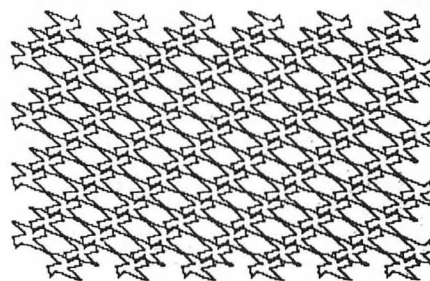
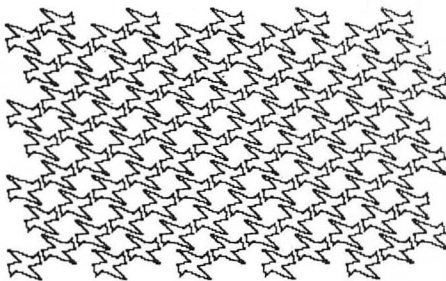
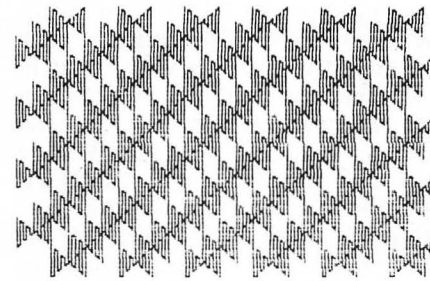
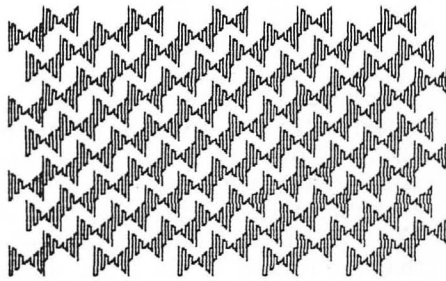
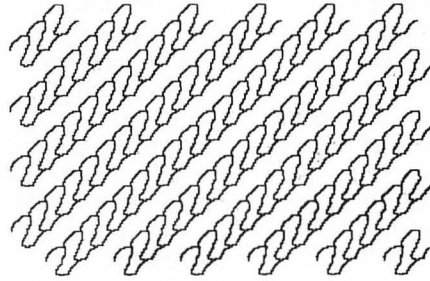
Pattern Option 2



Grid



Grid



### Atari One Liners

```

1 DEG :GRAPHICS 23:POKE 708,RND(1)*25
5:FOR L=0 TO 360:COLOR 1+RND(1)*3:PLO
T 80,48:DRAWTO 80+(47*SIN(L)),48+(46*
-COS(L)):NEXT L:RUN

```

David Simmons  
Redondo Beach, CA

```

0 GRAPHICS 18:POSITION 3,4: ? #6;*NSI
FUTURE BAND*:FOR X=10000 TO 0 STEP
-1.5:SOUND 0,0,8,X:POKE 708,INT(RND(
0)*222):NEXT X

```

John Niem and Zvl Arifin  
Hong Kong

```

1 GRAPHICS 23:C=INT(RND(0)*8):C=C+(C
=0):COLOR C:FOR Y=0 TO 95 STEP C:PLO
T 0,Y:DRAWTO 39,95-Y:DRAWTO 79,Y:DRA
WTO 119,95-Y:DRAWTO 159,Y:NEXT Y:RUN

```

Lynn Wallace  
Rapid City, SD

```

1 CLR :GRAPHICS 8+16:SETCOLOR 2,0,0:
FOR I=1 TO 48:PLOT X,Y:DRAWTO X,191-
Y:DRAWTO 319-X,191-Y:DRAWTO 319-X,Y:
DRAWTO X,Y:X=X+5:Y=Y+4:NEXT I:GOTO 1

```

Pery Pearson  
Modesto, CA



### duplicating service

307 West Main Street  
Maple Shade, NJ 08052

(609) 667-1667

- ① ● AMP "Data-settle" blank cassettes for digital use
- ② ● Cassette Storage Boxes
- ③ ● Cassette Labels Custom printing & blank
- ④ ● Custom Record Album production from your tapes
- ⑤ ● Stereo and Spoken Word cassette duplication

Call or write to:

*Jerry*

for more information.

All cassette work at  
AMP R. & D. is custom work  
to fit your needs.



# MUSI-KEYS INPUT ROUTINE

by Jon Voskuil

Here is a short routine that you can use in any Applesoft program to add a little interest when the user needs to enter information from the keyboard. Its effect is simply to play a short musical note whenever a key is pressed — a different pitch for each key. So if the user is asked to type in his name, he creates music while he does so (rather like Touch-Tone phone dialing).

Not only does this give the advantage of pleasant audio feedback, but inherent in the routine is the ability to input ALL punctuation and most control characters. If you use this routine in place of a normal INPUT statement, you will never again get an EXTRA IGNORED message from your Apple when someone enters his name as "Anderson, Bob". Commas, colons, and quotation marks will be treated as part of the input string, not as separators. And if you feel a need to enter control characters, most of them also will be accepted as part of the input string. A CTRL-G, for example, will place a beep in the string. (Did you know that INPUT will also accept most control characters?) However, note that CTRL-H or the left arrow will still cause a backspace, CTRL-C will still stop execution (immediately, not waiting for RETURN), and CTRL-M will still function just like RETURN, terminating the input.

I chose to use the underline character for the cursor, rather than the standard blinking square, just for the sake of variety. If you're a real consistency freak and simply MUST have the blinking square, you can

make the following change in lines 20, 23, and 24: In place of "CHR\$(95);", insert ": FLASH: PRINT CHR\$(32);: NORMAL: PRINT". For that matter, you can choose any NORMAL, FLASHing, or INVERSE character you like for the cursor, by making similar modifications. This is an easy way to give your programs a unique, individual touch.

Using the routine in a program is simplicity itself. Just copy lines 10 through 25 into the beginning of your program, and then start the main program itself at line 100. Of course, you can change the line numbers if you like (along with the various GOTOs), just as long as the code in lines 10-11 gets executed before calling the subroutine starting in line 20. It is advisable, for the sake of speed, to keep this subroutine near the beginning of your program rather than moving it to the end.

In your main program, whenever you would use an INPUT statement, use a GOSUB 20 instead. The subroutine will return your input in the string variable A\$, which can then be manipulated in the usual ways — including taking its VALUE for use as numerical input. "GOSUB 20", then, is the exact substitute for "INPUT A\$". If A\$ already has another purpose in your program, you can feel free to substitute a different variable name in the appropriate places in the subroutine. Likewise, if the variables CHR, PITCH, TIME, or MUSIC (i.e. any real variables starting with CH, PI, TI, or MU) have special uses already, those may be changed. The variables POK\$ and P are used only once, in

poking in the sound routine, and may be reused in any other part of the program for other purposes.

An extra goodie that you get along with the input routine is the sound routine. This can be used from anywhere in your program by POKEing values into PITCH and TIME (0 to 255) and then CALLing MUSIC. It's a standard tone routine that you find in all sorts of Apple programs. For your reference, here are the appropriate values to POKE into PITCH for different notes:

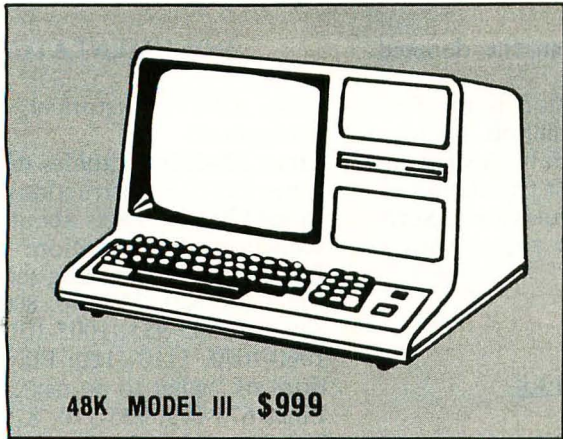
C	192	96	48	24
B	204	102	51	25
A#	216	108	54	27
A	229	114	57	29
G#	242	121	61	30
G	225	128	64	32
F#		136	68	34
F		144	72	36
E		153	76	38
D#		162	81	40
D		171	86	43
C#		182	91	45
C		192	96	48

You can use lower numbers for higher notes, following the same pattern; but pitch errors get more pronounced as you are forced to use, for example, either 12 or 13 instead of 12.5 for the next higher B. Notice that the routine is not POKEd into memory using DATA statements, so as not to interfere with other use of DATA statements in the main program.

The notes inserted in the listing below should explain the operation of the subroutine. Have fun using (and modifying) it to brighten up your programs.

<pre> Poke in machine-language sound-generating routine. 10 POK\$ = "173,048,192,136,208,00    4,198,001,240,008,202,208,24    6,166,000,076,000,003,096" 11 FOR P = 0 TO 18: POKE 768 + P    , VAL ( MID\$ ( POK\$, P * 4 + 1    , 3 ) ) : NEXT P : PITCH = 0 : TIME =    1 : MUSIC = 768 : GOTO 100  Clear keyboard, print prompt and cursor. (CHR\$(8) backspaces without erasing the character displayed on the screen.) 20 POKE - 16368,0: PRINT "? "; CHR\$(    95); CHR\$(8);A\$ = ""  Check for keypress. </pre>	<pre> 21 CHR = PEEK ( - 16384 ) : IF CHR    &lt; 127 THEN 21 Play tone, with pitch related to ASCII value of character. If character is RETURN, then return to main program. 22 POKE - 16368,0:CHR = CHR - 1    28: POKE TIME,15: POKE PITCH    ,CHR * 2: CALL MUSIC: IF CHR    = 13 THEN PRINT " ": RETURN Print character over cursor, and advance cursor to next space. Add character to A\$. 23 IF CHR &lt; &gt; 8 THEN PRINT CHR\$(    CHR); CHR\$(95); CHR\$(8);:    A\$ = A\$ + CHR\$(CHR): GOTO    21 </pre>	<pre> If character is backspace, erase last character, reposition cursor, and delete rightmost character from A\$. 24 PRINT " "; CHR\$(8); CHR\$(8)    ; CHR\$(95); CHR\$(8);: IF LEN    (A\$) &gt; 1 THEN A\$ = LEFT\$(A    \$, LEN(A\$) - 1): GOTO 21 25 A\$ = "": GOTO 21 Begin main program at line 100. For example: 100 HOME 110 PRINT "TYPE IN YOUR NAME:"; 120 GOSUB 20 130 PRINT : FOR I = 1 TO 15: PRINT    TAB(I)"HELLO, ";A\$: NEXT I 140 END </pre>
--	--	--

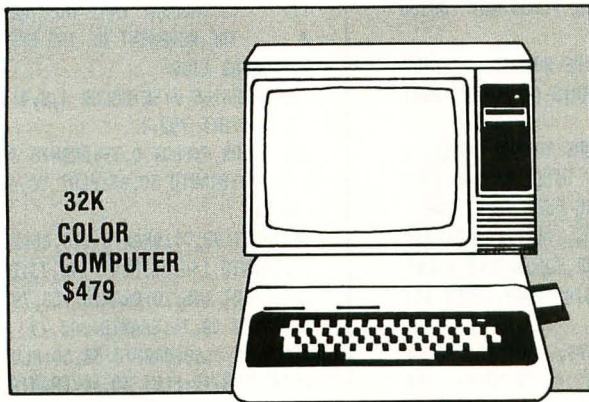




**48K MODEL III \$999**

### COMPUTERS

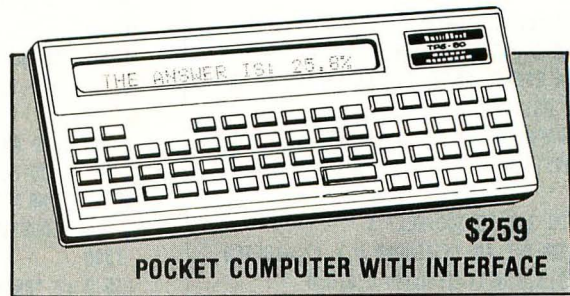
- TRS-80 Model III, 16K RAM (#26-1062) . . . . \$919.00
- TRS-80 Model II, 48K RAM (#26-1062+) . . . \$999.00
- TRS-80 Mod. III 48K RS232 2-dr. (#26-1063) \$2299.00
- TRS-80 Pocket Comp. w/Interface (#26-3501+) . \$259
- TRS-80 Video Tex (#26-5000) . . . . . \$299.00
- TRS-80 Color Computer, 4K RAM (#26-3001) . \$359.00
- TRS-80 Color Comp. 16K RAM (#26-3001+) . \$399.00
- TRS-80 Color Comp. 32K (#26-3001++) . \$479.00
- TRS-80 Color Comp. Ext. BASIC (#26-3002) . \$529.00
- TRS-80 Color Comp. Ext. BASIC 32K (#26-3002+) . . . . . \$599.00



**32K  
COLOR  
COMPUTER  
\$479**

### MODEL I PERIPHERALS

- COMM-80 Interface (#4-80) . . . . . \$159.00
- CHATTERBOX Interface (#4-81) . . . . . \$239.00
- DISK-80 Interface, 32K RAM (#4-82) . . . . \$339.00
- DISK-80 + Interface, 32K RAM (#4-83) . . . \$369.00
- BUSY BOX Interface (#4-01) . . . . . \$89.95
- LYNX Communications Interface (#19-80) . . \$229.00
- RS Exp. Interface, 32K RAM (#26-1140-32) . \$389.00
- 16K Memory Kit, TRS-keyboard (#5-1102-1) . \$49.00
- 16K Memory Kit, TRS-Interface (#5-1102) . . \$49.00
- ORCHESTRA-80 (#15-03) . . . . . \$79.95
- Upper/Lower Modification Kit (#15-02) . . . \$29.95
- CPU Speed-up Modification kit (#15-04) . . . \$45.00
- Video Reverse Modification kit (#15-05) . . . \$23.95
- 2-port TRS-BUS Extender (#15-12) . . . . . \$29.95
- 3-port TRS-BUS Extender (#15-13) . . . . . \$34.95
- TRS-80 Model I Dust Cover Set (#16-01) . . . . \$7.95
- TRS-80 Model I Carrying Case (#7-201) . . . . \$109.00
- TRS-80 Monitor Carrying Case (#17-202) . . . . \$84.00
- Dual Joysticks for Color Computer (#26-3008) . \$24.95
- CTR-80A Cassette Recorder & Cable (#26-1206) \$59.95
- TRS-80 Model III Dust Cover (#16-05) . . . . . \$7.95
- TRS-80 Color Dust Cover (#16-06) . . . . . \$7.95
- DIGI-TALKER MOD-I (#4-DT1) . . . . . \$189.00
- DIGI-TALKER MOD-III (#4-DT3) . . . . . \$189.00
- 80-Graphic Board (#26-80G) . . . . . \$139.00
- LYNX Auto-Answer/Auto Dial (#19-82) . . . . \$279.00



**\$259  
POCKET COMPUTER WITH INTERFACE**

### MODEL I DISK DRIVES

- HARDSIDE 40-track Drive (#7-40) . . . . . \$319.00
- PERCOM TFD-40 Drive (#7-99) . . . . . \$379.00
- PERCOM TFD-100 Drive (#7-100-1) . . . . . \$399.00
- HARDSIDE 80-track Drive (#7-80) . . . . . \$449.00
- PERCOM Dual TFD-100 Drives (#7-100-2) . . . \$799.00
- PERCOM Data Separator (#7-03) . . . . . \$29.95
- PERCOM Doubler (#7-07) . . . . . \$219.95
- HARDSIDE Extender Cable (#7-02) . . . . . \$15.95
- HARDSIDE 2-Drive Cable (#7-04) . . . . . \$29.00
- HARDSIDE 4-Drive Cable (#7-05) . . . . . \$39.00

TSE-HARDSIDE

6 South St., Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790

TERMS: Prices and specifications are subject to change. HARDSIDE accepts VISA & MASTERCARD. Certified checks and Money Orders: Personals checks accepted (takes 3 weeks to clear). HARDSIDE pay all shipping charges (within the 48 states) on all PREPAID orders OVER \$100.00. On all orders under \$100 a \$2.50 handling charge must be added. COD orders accepted (orders over \$250 require 25% deposit), there is a \$5.00 handling charge. UPS Blue Label, and Air Freight available at extra cost. TRS-80 is a trademark of Tandy Corp.



# VOLCANO

by Victor T. Albino (Atari Translation by Art Cestaro)

**"Volcano" is an Atari program which requires 24K of RAM.**

Here is the Atari version of "Volcano", a combination of game and educational software. The program explains current scientific theory about volcanos, complete with details about different types of lava, mud flows, and the force of nature's most powerful explosive. After you finish studying about volcanos, you are then presented with a quiz or survival test. You are placed on Mount St. Helens at the moment of eruption and given a number of options. Should you choose the correct course of action, you will be presented with further decisions until you either

perish or escape from the doomed mountain.

This Atari translation is the first winner in our translation contest. Each month we will publish what we consider to be the best translation of a feature program in an issue of **SoftSide**, and reward the author with a software certificate.

## VARIABLES

E: Contains the variable of your action.

E\$: Contains the string "Enter the number of your action".

P\$: Contains the string "Press fire".

Q: Variable used in mountain rumbling noise.

## DOCUMENTATION

Lines 100-171: Historical background.

Lines 180-230: Graphics display.

Lines 235-515: Instructions.

Lines 520-605: Facts about the consequences of an eruption.

Lines 609-645: Opening scene.

Lines 650-660, 730-740, 800-810, 885-895, 945-955, 1005-1015,

1060-1070, 1110-1120: Player's options of "what to do next."

Lines 675-695, 755-770, 820-850, 910-930, 970-980, 1085-1095: Perish routines, program branches here when you have made a fatal decision.

Lines 710-715, 780-785, 865-870, 985-990, 1030-1045, 1140-1175:

Player's current location.

Lines 1140-1205: Win routine.

```

1 REM ** VOLCANO **
2 REM Translated by
3 REM Art Cestaro III
5 DIM E$(45),A$(3):E$="ENTER THE NO. OF
  THE ACTION "
10 GRAPHICS 0:POKE 752,1:SETCOLOR 2,8,
3:POSITION 8,5:? "      V O L C A N O
  "
15 FOR X=1 TO 300:NEXT X
20 FOR X=1 TO 255:SOUND 0,X,12,6:SETCO
LOR 2,3,X:NEXT X:SOUND 0,0,0
30 SETCOLOR 2,8,3:SETCOLOR 4,8,3:? "}"
100 ? :? :? "ON MAY 18, 1980, AT 8:32
AM,
      MOUNT ST. HELENS ERUPTED"
105 ? "WITH A FORCE 500 TIMES GREATER
      THAN THAT OF THE BOMB THAT FELL
  "
110 ? "ON HIROSHIMA!! THE TOP 1300 FE
ET
      OF THE MOUNTAIN WAS IMMEDIATELY
  "
112 ? "PULVERIZED AND THROWN 12 MILES
INTO THE SKY. WITHIN MOMENTS"
114 ? "EVERYTHING WITHIN 150 SQUARE
MILES WAS ENGULFED IN THE FURY"
115 ? "OF THE VOLCANO!!!":GOSUB 1300
116 ? :? :? "THICK, BLACK SMOKE ACCENT
ED WITH"
117 ? "PINK AND PURPLE LIGHTNING, FLYI
NG
      BOULDERS AND CHUNKS OF ICE BLAS
TED"
118 ? "A SWATH OF DESTRUCTION 20 MILES
TO THE NORTH. HOT ASH, ROCKS A
ND"
119 ? "GAS ROARED DOWN THE MOUNTAIN IN
PYROCLASTIC FLOWS AT OVER 100 M
PH."

```

```

120 ? "THE 800 DEGREE FLOWS COLLIDED I
NTO SPIRIT LAKE AND BECAME A TORREN
T"
121 ? "OF SUPERHEATED MUD, CARRYING OF
F
      100-TON LOGGING TRUCKS, BRIDGES
  "
122 ? "AND ANYTHING ELSE IN ITS PATH.
THE
      MUD FLOWED ON INTO THE TOUTLE,"
123 ? "COWLITZ AND COLUMBIA RIVERS
CAUSING SEVERE FLOODING.":GOSUB
      1300
135 ? :? "DRAWN BY THE HOPE OF SEEING
      SOME VOLCANIC ACTIVITY, MANY
  "
136 ? "PEOPLE HAD COME TO THE MOUNTAIN
THAT WEEKEND, SINCE ST. HELENS"
137 ? "BEGAN EMITTING PUFFS OF STEAM
BACK IN MARCH. ALTHOUGH THERE"
138 ? "WERE SIGNS AND ROADBLOCKS WARNI
NG
      EVERYONE TO STAY AWAY, THEY STI
LL"
139 ? "CAME: TOURISTS, CAMPERS, SCIEN
TISTS, PHOTOGRAPHERS, ADVENTURERS, AND
  "
140 ? "JUST PLAIN CURIOUS.":GOSUB 1300
145 ? :? :? "BUT THEN NO ONE EXPECTED
THAT
      THE MOUNTAIN WAS WAITING
TO"
146 ? "EXPLODE LIKE SOME KIND OF GIGAN
TIC
      NUCLEAR TIME BOMB.":GOSUB 1300
155 ? :? :? "ABOUT SEVENTY OF THE VISI
TORS TO
      THE MOUNTAIN THAT WEEKEND
DID"
156 ? "NOT LEAVE ALIVE. EXACTLY HOW M
ANY
      PEOPLE PERISHED WILL LIKELY NEV
ER"
157 ? "BE KNOWN."

```

```

160 ? :? "WHAT IS KNOWN IS THAT AIR FO
RCE
      AND ARMY NATIONAL GUARD HELI
COPTERS"
161 ? "RESCUED 197 PEOPLE, SNATCHING T
HEM
      FROM AROUND THE BOILING MOUNTAI
N.":GOSUB 1300
170 ? :? :? "SCIENTISTS ARE NOT EXACTL
Y SURE
      WHAT CAUSES A VOLCANO, BU
T IT"
171 ? "IS THOUGHT THAT THEY RESULT FRO
M
      THE MOVEMENT OF THE EARTHS CRUS
T.":GOSUB 1300
180 GRAPHICS 7:SETCOLOR 4,8,4:SETCOLOR
2,1,1:POKE 752,1
181 COLOR 3:PLOT 0,79:DRAWTO 30,68:PLO
T 30,68:DRAWTO 50,63:PLOT 50,63:DRAWTO
98,71
182 PLOT 98,71:DRAWTO 110,65:PLOT 110,
65:DRAWTO 140,73:PLOT 140,73:DRAWTO 14
5,70:PLOT 145,70:DRAWTO 158,79
183 PLOT 98,71:DRAWTO 103,79
190 PLOT 0,60:DRAWTO 38,50:PLOT 38,50:
DRAWTO 58,40:PLOT 58,40:DRAWTO 69,28
191 PLOT 69,28:DRAWTO 80,20:PLOT 80,20
:DRAWTO 88,24:PLOT 88,24:DRAWTO 95,21
192 PLOT 95,21:DRAWTO 97,23:PLOT 97,23
:DRAWTO 108,36:PLOT 108,36:DRAWTO 130,
50:PLOT 130,50:DRAWTO 158,73
193 PLOT 80,20:DRAWTO 87,17:PLOT 87,17
:DRAWTO 95,21
200 ? "GREAT SLABS OF EARTH'S CRUST RU
B
      AGAINST EACH OTHER GENERATING"
201 ? "TREMENDOUS HEAT AND PRESSURE AN
D"
202 FOR X=1 TO 900:NEXT X:? "}"
205 ? "FORMING MAGMA OR MOLTEN ROCK.
THE"

```

206 ? "HOT MAGMA RISES AND EVENTUALLY WORKS ITS WAY TO THE SURFACE."  
 208 FOR X=1 TO 900:NEXT X:?"})"  
 209 ? "WHEN IT ERUPTS, A VOLCANO IS BORN!!"  
 350 FOR H=1 TO 300:NEXT H  
 400 FOR G=0 TO 190:SOUND 0,6,12,14:SET COLOR 2,6,6:NEXT G:SETCOLOR 2,1,1  
 500 GOSUB 1400:FOR U=230 TO 80 STEP -1 :SOUND INT(RND(1)\*4),RND(1)\*U,8,8  
 505 COLOR RND(1)\*8:PLOT 87,21:DRAWTO INT(RND(0)\*90)+40,INT(RND(1)\*23)+4:POKE 712,U  
 506 IF U<160 THEN COLOR 1:PLOT 87,21:DRAWTO RND(0)\*90+45,RND(0)\*20+35  
 509 NEXT U  
 510 GRAPHICS 0:SETCOLOR 2,8,3:SETCOLOR 4,8,3:FOR Z=0 TO 3:SOUND Z,0,0,0:NEXT Z  
 511 ? :? :? "THIS PROGRAM CREATES A SCENARIO "  
 512 ? "SIMILAR TO THOSE REPORTED BY SOME OF THE APPROXIMATELY 300 PEOPLE "  
 513 ? "WHO WERE CAMPED AROUND MT. ST. HELENS ON THE MORNING OF "  
 514 ? "MAY 18,1980."  
 515 ? :? "YOUR GOAL IS SIMPLE: SURVIVE!!"  
 516 ? :? "THE HAZARDS ARE MANY...":GOSUB 1300  
 517 ? :? " LAVA: UNLIKE THE MORE LIQUID LAVA OF THE HAWAIIAN VOLCANOES, THE "  
 518 ? "LAVA OF ST. HELENS IS OF THE ANDESITE VARIETY--A STICKY, GUMMY "  
 519 ? "MATERIAL. IT MOVES SLOWLY AND DOES NOT TRAVEL FAR."  
 520 ? :? " ASH: MOST OF THE DEATHS AT ST. HELENS WERE DUE TO SUFFOCATION "  
 521 ? "CAUSED BY HOT ASH FORCED INTO THE LUNGS."  
 522 ? :? " PYROCLASTIC FLOWS: MASSES OF HOT, DRY ROCK THAT MOVE LIKE A FLUID "  
 523 ? "BECAUSE THEY ARE MIXED WITH HOT AIR AND OTHER GASES. THESE FLOWS "  
 524 ? "TRAVEL AT OVER 100 MPH AND AFFECT AREAS FAR FROM THE VOLCANO.":GOSUB 1300  
 525 ? :? " MUD FLOWS: THESE LOOK LIKE A WAVE OF HOT, FLOWING CONCRETE AND CAN "  
 526 ? "MOVE AT SPEEDS UP TO 50 MPH. THEY CARRY ALONG BOULDERS, TREES, AND "  
 527 ? "DEBRIS IN THEIR WAKE."  
 528 ? :? " GASES: HOT GASES SUCH AS CARBON DIOXIDE, CARBON MONOXIDE, CHLORINE, "  
 529 ? "AND SULFUROUS FUMES ARE SPEWED OUT TOGETHER WITH MOLTEN OR SOLID ROCK."

530 ? :? " DEBRIS: LARGE CHUNKS OF EARTH, ROCK, AND ICE CAN BE THROWN OUT FROM "  
 531 ? "ANY OF THE MOUNTAIN'S FLANKS AS WELL AS ITS SUMMIT.":GOSUB 1300  
 532 ? :? " LIGHTNING: FLASHES OF LIGHTNING GENERATED BY HIGHLY CHARGED DUST "  
 533 ? "PARTICLES, STREAK AROUND THE MOUNTAIN."  
 534 ? :? " FLOODS: MUDFLOWS JOIN WITH RIVERS SWELLING THEM MANY TIMES THEIR "  
 535 ? "NORMAL SIZE CAUSING WIDESPREAD FLOODING AND KILLING WILDLIFE."  
 536 ? :? " EARTHQUAKES: THESE RESULT FROM THE MOVEMENT OF THE MOLTEN ROCK "  
 537 ? "WITHIN THE MOUNTAIN."  
 538 ? :? " FIRES: ABOUT A HUNDRED FIRES WERE STARTED DURING THE ST.HELENS "  
 539 ? "ERUPTION BY HOT GASES,PYROCLASTIC FLOWS AND LIGHTNING.":GOSUB 1300  
 610 POKE 752,1:?" THE SCENE "  
 611 ? :? :? "YOU ARE CAMPED ON THE WEST SIDE OF MT.ST.HELENS."  
 612 ? "YOU HIKE UP FROM THE END OF A LOGGING ROAD WHERE YOU LEFT YOUR "  
 613 ? "CAR LAST NIGHT. YOU HAVE A SMALL FIRE GOING TO MAKE SOME COFFEE."  
 616 ? :? "THERE IS AN EERIE QUIET. THERE ARE NO BIRDS SINGING.":GOSUB 1300  
 617 ? :? :? "SUDDENLY YOU ARE SHAKEN BY AN EARTHQUAKE!!!!"  
 618 ? "THE GROUND UNDULATES AROUND YOU. TALL FIR TREES SWAY AWAY."  
 619 ? :? "THEN...A GIANT ROAR,AND THE TOP OF THE MOUNTAIN EXPLODES THROUGH "  
 620 ? "OF FEET INTO THE AIR SUSPENDED ON A THICK COLUMN OF BLACK SMOKE."  
 621 ? :? "THE DENSE CLOUD EXPANDS AND BEGINS TO MOVE IN YOUR DIRECTION.":GOSUB 1300  
 650 ? :? :? "WHAT SHOULD YOU DO ?"  
 651 ? :? :? "1) USE YOUR CAMPING SHOVEL TO DIG IN UNTIL IT'S SAFE TO MOVE."  
 652 ? "2) TAKE PHOTOGRAPHS.THEY SHOULD BE WORTH PLENTY."  
 653 ? "3) START BACK DOWN THE MOUNTAIN TO YOUR CAR."  
 654 ? "4) GET TO HIGH GROUND.  
 5) TRY TO FIND SOME SHELTER."  
 655 ? :? E\$:INPUT D  
 660 ON D GOTO 675,680,685,690,695  
 675 ? "5) :? :? "YOU JUST DUG YOUR OWN GRAVE.HOT ASH WILL BURY YOU.":GOTO 700  
 680 ? "6) :? :? "CONGRATULATIONS. YOU "

HAVE TAKEN SOME REALLY SPECTACULAR PICTURES"  
 681 ? "THAT WILL BE DUPLICATED IN 100 NEWSPAPERS. YOU WILL RECEIVE"  
 682 ? "MANY AWARDS--ALL POSTHUMOUSLY.":GOTO 700  
 685 ? "7) :? :? :? "NOW YOU ARE USING YOUR HEAD.GET OUT OF THERE FAST.":GOTO 710  
 690 ? "8) :? :? :? "FORGET IT! HOT SULFUR DIOXIDE GAS COMING OUT OF THE SIDE OF THE "  
 691 ? "MOUNTAIN GETS YOU AS YOU CLIMB HIGHER.":GOTO 700  
 695 ? "9) :? :? :? "YOU ARE TOO CLOSE TO THE HOT ASH FOR ANY SHELTER TO BE EFFECTIVE."  
 696 ? "THOSE WHO TRIED THIS WERE BURIED ALIVE.":GOTO 700  
 700 POKE 752,1:FOR T=0 TO 1000:NEXT T:?"10) :? :? :? " YOU HAVE PERISHED "  
 701 FOR G=1 TO 400 STEP 2:SOUND 0,6,12,13:SETCOLOR 2,6,6:NEXT G  
 702 SOUND 0,0,0,0:SETCOLOR 2,8,4:?" :? "WANT TO PLAY AGAIN (YES OR NO)";  
 703 INPUT A\$:IF A\$="YES" THEN 610  
 704 IF A\$="NO" THEN 707  
 705 IF A\$<>"YES" OR A\$<>"NO" THEN ? :? "ANSWER WITH YES OR NO!":GOTO 702  
 707 ? "11) :? :? :? "EITHER TAKE A SURVIVAL COURSE OR STAY AWAY FROM SMOKING MOUNTAINS..."  
 708 ? "PREFERABLY BOTH!":END  
 710 ? :? :? "AS YOU APPROACH THE HIKING TRAIL THAT LEADS DOWN TO THE LOGGING "  
 711 ? "ROAD, YOU FIND THAT MANY TREES HAVE FALLEN ACROSS THE TRAIL MAKING "  
 712 ? "PASSAGE DIFFICULT."  
 713 ? :? "YOU THINK YOU REMEMBER A SHORT CUT DOWN THE MOUNTAIN THAT ALSO LEADS "  
 714 ? "TO THE END OF THE LOGGING ROAD.":GOSUB 1300  
 730 ? :? "WHAT SHOULD YOU DO?"  
 740 ? :? "1) ATTEMPT TO NAVIGATE THE TRAIL."  
 741 ? "2) TAKE THE SHORT CUT."  
 745 ? :? E\$:INPUT D  
 750 ON D GOTO 765,755  
 755 ? "3) :? :? "IN AN EMERGENCY WE OFTEN FORGET THOSE THINGS THAT WE KNOW WELL--SUCH "  
 756 ? "AS OUR OWN PHONE NUMBER. IN YOUR EXCITEMENT YOU QUICKLY FIND YOURSELF "  
 757 ? "LOST.YOUR TIME RUNS OUT AND YOU ARE OVERRUN BY A PYROCLASTIC FLOW.":FOR T=1 TO 1000:NEXT T:GOTO 700  
 765 ? "4) :? :? :? "UNDER THE CIRCUMSTANCES THIS IS A BETTER CHOICE.AT LEAST YOU KNOW "  
 766 ? "THAT THE TRAIL EVENTUALLY LEADS BACK TO THE ROAD. YOU CLIMB OVER "  
 ER"

continued on next page

continued from previous page

767 ? "AND UNDER HUGE FALLEN TREES AS YOU MAKE YOUR WAY BACK DOWN"  
768 ? "THE TRAIL.":GOSUB 1300  
780 ? :? :? :? "WHEN YOU ARRIVE AT YOUR CAR, YOU FIND THE ROAD IS BLOCKED BY"  
781 ? "LARGE ROCKS WHICH ROLLED ON IT DURING THE QUAKE. YOU KNOW THAT THE"  
782 ? "END OF THESE LOGGING ROADS IS ONE OF THE FIRST PLACES THAT RESCUERS"  
783 ? "WILL LOOK."  
784 ? :? "ON THE OTHER HAND THERE IS A CHANCE YOU COULD MOVE THOSE ROCKS JUST ENOUGH TO GET THROUGH."  
788 GOSUB 1300  
795 ? :? "WHAT SHOULD YOU DO?"  
800 ? :? "1) GET IN THE CAR FOR PROTECTION AND WAIT TO BE RESCUED."  
802 ? "2) FORGET THE CAR AND GO DOWN THE ROAD ON FOOT."  
803 ? "3) BUILD A FIRE TO ATTRACT RESCUERS."  
804 ? "4) TRY TO GET THE ROCKS OUT OF THE WAY."  
810 ? :? E\$;:INPUT D:?" :?":ON D GOTO 820,825,830,835  
820 ? :? "SOME PEOPLE DID EXACTLY THAT UNFORTUNATELY, THEIR CARS BECAME"  
821 ? "THEIR TOMBS.":GOTO 700  
825 ? :? "IT IS HARD TO OVERTAKE A VOLCANO. YOU ARE OVERTAKEN BY A MUDSLIKE.":GOTO 700  
830 ? :? "YOU WON'T HAVE TO BUILD A FIRE HERE. IT WILL BE PLENTY HOT IN A FEW MINUTES.":GOTO 700  
835 ? :? "IF THERE IS A CHANCE YOU CAN USE YOUR CAR TO GET AWAY, TAKE IT!!"  
836 ? "BY USING THE LIMBS AS A LEVER AND PUSHING OTHER ROCKS OUT OF THE WAY"  
837 ? "WITH THE CAR, YOU MANAGE TO GET THROUGH. AS SOON AS YOU'RE CLEAR,"  
838 ? "YOU FLOOR THE ACCELERATOR. YOU'RE GOING 80 MPH DOWN A DIRT ROAD. YOU"  
840 ? "ARE APPROACHING A BRIDGE THAT CROSSES THE TOUTLE RIVER WHEN"  
841 ? "SUDDENLY A MUDFLOW HITS. THE BRIDGE EXPLODES BEFORE YOUR EYES."  
845 ? "YOU SLAM ON YOUR BRAKES AND STOP JUST SHORT OF THE STEAMING"  
846 ? "CHOCOLATE OOZE. YOUR WAY BLOCKED, YOU GET OUT OF YOUR CAR AND BEGIN"  
847 ? "RUNNING. AFTER A WHILE YOU TIRE AND SLOW DOWN, BUT LOOKING UP YOU START"  
848 ? "TO RUN AGAIN."  
850 ? "THE ASH CLOUD IS COMING!!":GOSUB 1300

865 ? :? :? "AS THE BLACK CLOUD DESCENDS, IT IS AS IF SOMEONE HAS THROWN A BLACK"  
866 ? "VELVET CURTAIN OVER YOUR HEAD. ALL LIGHT VANISHES. YOU CANNOT SEE YOUR"  
867 ? "HAND IN FRONT OF YOUR FACE. IT IS HOT, BURNING."  
870 ? "YOU TRY TO CATCH YOUR BREATH, BUT YOUR THROAT FEELS LIKE IT IS"  
871 ? "STUFFED WITH WARM COTTON FLUFF. YOUR LUNGS BURN AND YOUR EYES"  
872 ? "STING. YOU STUMBLE AND FALL IN THE DARKNESS.":GOSUB 1300  
885 ? :? "WHAT SHOULD YOU DO?"  
890 ? :? "1) LIE DOWN WITH YOUR FACE ON THE GROUND."  
891 ? "2) FIND SOME SHELTER."  
893 ? "3) RUB THE ASH FROM YOUR EYES."  
894 ? "4) KEEP MOVING IN THE DARK."  
900 ? :? E\$;:INPUT D:?" :?":ON D GOTO 910,915,920,925  
910 ? :? "THE RIGHT DECISION, IF YOU WANT TO MAKE THIS YOUR ETERNAL RESTING PLACE!!":GOTO 700  
915 ? :? "I HOPE YOU LIKE THE SPOT YOU FIND, BECAUSE YOU'RE GOING TO BE THERE FOREVER!!":GOTO 700  
920 ? :? "VOLCANIC ASH IS VERY ABRASIVE. YOUR EYES SWELL UP. YOU WANDER AROUND"  
921 ? "AND ARE OVERCOME.":GOTO 700  
925 ? :? :? "THIS IS THE ONLY REAL CHANCE YOU HAVE. TO STAY WHERE YOU ARE"  
926 ? "MEANS CERTAIN DEATH."  
930 ? :? :? "YOU GET UP AND BEGIN WALKING WITH YOUR HANDS OUT BEFORE YOU TO"  
931 ? "FEEL THE WAY. AFTER A WHILE YOU NOTICE THAT YOUR FEET FEEL WET."  
932 ? "YOU THINK YOU ARE WALKING IN A CREEK.":GOSUB 1300  
940 ? :? "WHAT SHOULD YOU DO?"  
950 ? :? "1) GET OUT OF THE WATER AND KEEP MOVING."  
951 ? "2) WASH OUT YOUR EYES WITH THE WATER."  
952 ? "3) WALK IN CREEK."  
955 ? :? E\$;:INPUT D:?" :?":ON D GOTO 970,975,980  
970 ? :? "SINCE YOU HAVE NO WAY TO KNOW WHERE YOU'RE GOING, YOU WANDER"  
971 ? "IN A CIRCLE UNTIL EXHAUSTION TAKES OVER.":GOTO 700  
975 ? :? "THE ASHFALL IS TOO HEAVY. THE WATER MIXES WITH IT TO FORM A GRITTY PASTE.":GOTO 700  
980 ? :? "SINCE YOU HAVE NO IDEA OF WHICH DIRECTION YOU'RE GOING, THIS OFFERS"  
981 ? "THE BEST ALTERNATIVE. YOU STOP DOWN AND FEEL THE WAY THE WATER"

982 ? "IS FLOWING AND FOLLOW THE CURRENT DOWNHILL."  
985 ? :? :? "YOU TRAVEL DOWN THE CREEK FOR SOME TIME. THE, WITHOUT WARNING, YOU"  
986 ? "PLUNGE INTO HOT WATER UP TO YOUR NECK. INSTINCTIVELY, YOU REACH OUT"  
987 ? "IN THE DARKNESS AND GRAB HOLD OF SOMETHING..A FALLEN TREE."  
988 ? "IT PULLS YOU ALONG DOWNSTREAM.":GOSUB 1300  
1005 ? :? "WHAT SHOULD YOU DO?"  
1015 ? :? "1) LET GO OF THE LOG AND SWIM FOR THE BANK."  
1016 ? "2) SWIM UNDERWATER."  
1017 ? "3) HANG ON."  
1020 ? :? E\$;:INPUT D:?" :?":ON D GOTO 1030,1030,1035  
1030 ? :? "THE RIVER IS FILLED WITH DEBRIS. YOU ARE CRUSHED IN A LOG JAM!":GOTO 700  
1035 ? :? "THE LOG CARRIES YOU ALONG UNTIL IT STOPS AT A LOG JAM. THE END"  
1037 ? "YOU ARE HOLDING ON TO SWINGS AROUND AND RESTS ON A SHALLOW"  
1038 ? "AREA BESIDE THE BANK. YOU SCRAMBLE OUT OF THE WATER."  
1040 ? :? "FINALLY, YOU CAN SEE SOMETHING.. A DIM GLOW IN THE DISTANCE. IT"  
1042 ? "LOOKS LIKE A SEARCH LIGHT. THEN YOU REALIZE THAT IT IS THE SUN"  
1043 ? "SHINING THROUGH THE ASH CLOUD. AS YOU GO ON THE ASH GETS THINNER."  
1045 ? "FINALLY, THE SUN BREAKS THROUGH AND YOU CAN SEE THE SKY AGAIN."  
1047 ? "UP AHEAD YOU SEE A RIDGE, BELOW YOU IS A PLATEAU OF INDESCRIBABLE"  
1048 ? "DEVASTATION...A GRAY ASH COVERED MOONSCAPE.":GOSUB 1300  
1060 ? :? "WHAT SHOULD YOU DO?"  
1065 ? :? "1) YOU ARE BLEEDING A LITTLE FROM YOUR HEAD AND ARMS. TEND TO YOUR WOUNDS."  
1067 ? "2) GO UP TO THE RIDGE."  
1068 ? "3) START DOWN TOWARDS THE PLATEAU."  
1069 ? "4) REST YOURSELF."  
1070 ? :? E\$;:INPUT D:?" :?":ON D GOTO 1085,1095,1090,1085  
1085 ? :? "YOU HAVE WASTED VALUABLE TIME. YOU FALL ASLEEP AND NEVER WAKE UP!":GOTO 700  
1090 ? :? "WHY ON EARTH WOULD YOU WANT TO GO DOWN THERE!! POISON GAS GETS YOU.":GOTO 700  
1095 ? :? "UP ON THE RIDGE YOU CAN BE SEEN BETTER FROM THE AIR. ONCE ON"

1096 ? "TOP YOU GET A BETTER APPRECIATION FOR THE AWESOME MAGNITUDE OF THE HE"

1097 ? "VOLCANO. EVERYTHING BELOW YOU IS COVERED WITH THICK, IMPENETRABLE"

1099 ? "SMOKE.":GOSUB 1300

1100 ? :? "WHAT SHOULD YOU DO?"

1120 ? :? "1) CONTINUE DOWN THE MOUNTAIN."

1121 ? "2) TRY TO ATTRACT RESCUERS."

1125 ? :? E\$;:INPUT D:?"":ON D GOTO 1135,1140

1135 ? :? "THERE IS NO PLACE TO GO BUT BACK INTO THE DARKNESS."

1136 ? "YOU SUCCUMB TO SMOKE INHALATION.":GOTO 700

1140 ? :? :? "SINCE YOU ARE IN AN AREA EASILY VISIBLE TO RESCUE HELICOPTERS,AND"

1141 ? "EVERYTHING BELOW YOU IS COVERED WITH TOXIC SMOKE, YOU WISELY"

1144 ? "DECIDE TO BUILD A SIGNAL THAT CAN BE SEEN FROM THE AIR."

1145 ? "YOU ARRANGE ROCKS SO THEY SPELL OUT 'S O S 'AND POINT A LONG"

1146 ? "ARROW TO A SMALL FIRE.FINISHED YOU FALL TO THE GROUND EXHAUSTED,"

1147 ? "BUT FIGHTING THE URGE TO SLEEP YOU MUST REMAIN ALERT FOR THE"

1148 ? "POSSIBILITY OF RESCUE."

1149 ? :? "AGAINST THE RUMBLING OF THE

VOLCANO ,YOU HEAR A NEW SOUND..."

1150 ? "ALSO LOUD,BUT CHOPPY.":GOSUB 1300

1165 ? :? :? "IT'S A HELICOPTER !!!!!"

1168 ? :? "YOU TAKE OFF YOUR TORN SHIRT AND WAVE IT WILDLY. ASH FALLS"

1170 ? "OFF YOU IN A SMALL CLOUD AS YOU JUMP UP AND DOWN YELLING AS LOUD"

1171 ? "AS YOU CAN. THEY SEE YOU AS A PLASTER GRAY FIGURE RUNNING"

1172 ? "AROUND IN CIRCLES. THEY LOWER A WICKER BASKET, AND YOU JUMP IN.":GOSUB 1300

1190 ? :? :? " \* \* YOU MADE IT \* \*"

1195 ? :? :? "YOU BEAT THE MOUNTAIN!!"

1196 ? :? :? :? " \* \* \* CONGRATULATIONS \* \* \*"

1199 FOR G=240 TO 0 STEP -1:SOUND 0,G,10,14:NEXT G:END

1300 ? :? :? "PRESS ANY KEY TO CONT."

1305 IF PEEK(764)<>255 THEN POKE 764,255:?"":RETURN

1310 GOTO 1305

1400 SOUND 0,99,8,8:SOUND 1,75,8,8:SOUND 2,60,8,8:SOUND 3,20,8,8

1402 FOR L=1 TO 16:POKE 712,15\*(RND(1)\*17):NEXT L:RETURN

1404 REM ART V. CESTARO III, 4/10/81

LOG# AT153 TRANSLATION



**ABOUT THIS ISSUE** continued from page 5

of ways to play, ranging from the merely difficult to the sublime. Good shootin'.

David Durkee, along with some translating aid from resident CPU wizard Jon Voskuil, has graced our pages with a "Word-Search Puzzle Generator" that lets you create those word-search puzzles that we've featured in the magazine these past few months. Again, this program will work for any of the systems we support.

For you Atari owners that complain about a lack of Adventures, we don't have any. But we do have "Catacombs of the Phantoms", by Tom Plessman, a fine dungeon program that's full of surprises. And we also have our first translation contest winner: Art Cestaro, who has come up with an Atari version of

"Volcano", complete with on-screen eruption.

David Gash has provided us with a unique S-80 game: "Krazy Talk", which tests both verbal acuity and crossword skill, but not in any way that we've seen before. Plus we have an Apple "Hexadecimal Puzzle" from George Zeigler — a computer rendition of those little plastic number puzzles that used to drive you crazy when you were a kid. And the aforementioned Mr. Voskuil wraps up "Math Decathlon" this issue.

Finally, we have a "Maze Search" program for the Apple from Carl Mueller. And of course a whole slew of fine writings and ramblings from our assorted crazies, from that Adams fellow down south, to the mysterious "J". Enjoy.



**S-80 ONE LINER**

1 CLS:A=15360:B=16383:FORN=0T01023:POKEA+N,RND(63)+128:POKEB-N,RND(63)+128:NEXTN:FORT=1T01200:NEXTT:RUN  
 Stephen Roy Hugli  
 Alexander, ME



**STAR TREK III.5**

**ALL NEW VERSION!**

by Lance Micklus

Now with Sound Capability and Increased Speed of Execution.

You are in command of the starship Enterprise and her complement of 371 officers and crew. You must enter and explore the Omega VI region of the galaxy with its 192 quadrants containing star systems and planets (a few of which are habitable).

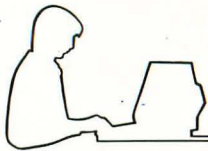
Astronomical hazards such as pulsars, Class 0 stars, and black holes are known to be present in the region. Klingon battle cruisers are also present, so the utmost care is needed.

Star Trek III.5 includes: playboard 8 by 3 by 3 quadrants; weapons system of Phasers and Photon Torpedos; Warp and Impulse power systems; Science and Ship's computers; Long and Short Range sensors; Damage Control and Status reports; and 20 Klingon battle cruisers, and 100 stars, planets, black holes, and pulsars.

Available on Digital Cassette for Level II, 16K ~~128-5183~~ \$14.95



**TSE-HARDSIDE**  
 6 South St. Milford, NH 03055 (603) 673-5144  
 TOLL FREE OUT-OF-STATE 1-800-258-1790



# CATACOMBS OF THE PHANTOMS

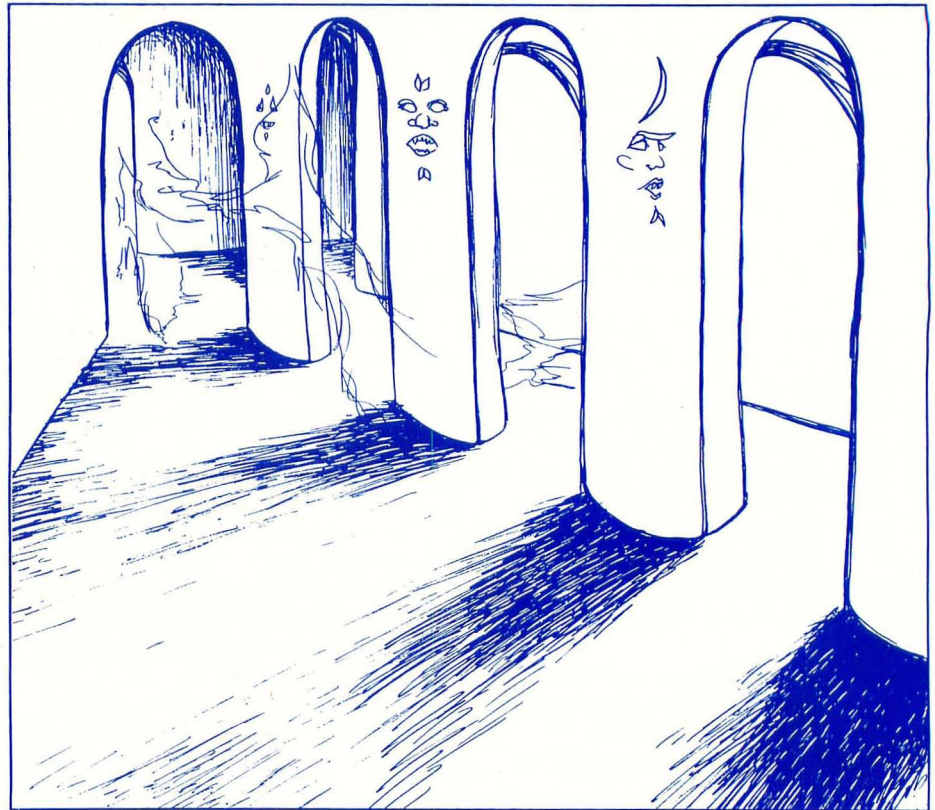
by Tom Plassman

**"Catacombs of the Phantoms" is written for an Atari with 16K RAM.**

The opportunities in Petiteville were certainly not great, so it was perfectly understandable when you went to consult the town's doddering old wizard about opportunities for adventure. He wasn't particularly lucid, but in the course of his ramblings he did mention the "Catacombs of the Phantoms". When you pressed him for more information, his mind seemed to clear a bit and he told you about the Golden Goddess of Power that the Sorcerer Agalinta was reported to have hidden in the caves, and about the fearsome collection of monsters that were held there by its power. He even mentioned the phantoms which gave the catacombs their name and told of their curious reputation for both aiding and destroying would-be adventurers.

The old wizard dozed off about then, raising more questions in your mind than he'd answered; but what he had said piqued your curiosity and set you searching for other legends of the catacombs among the town's people. The town chemist told you about the fountains of blood that were said to increase a man's strength tenfold. The town cleric warned that the phantoms were the agents of Beebulbub, the Evilest One, and that the catacombs were the first stop on the road to damnation. The innkeeper at the Sniveling Hen spoke only of the immense treasure that the catacombs were said to contain. But the most important piece of information came from Twap, the town's oracle and most prominent drunk. He said the old wizard had a potion that could send a person to the catacombs in the blink of an eye.

So back you went to the wizard with a bag of food, a change of clothes, and your father's battered but trusty old sword. The wizard was oddly alert and seemed disquietingly eager to send you on your way. As he fiddled with his amulets and rummaged through his potions, he said, "Remember, there are only three ways to leave the catacombs, boy. You can search out nature's entrance, you can conquer the guardians of the Golden Goddess, or you can die. And the last is by far the simplest!"



He handed you a small leather bag and told you it contained three magical worms. "Use these wisely," he said, "They have the power to bore through stone and always search out a lower catacomb."

Then, reciting an incantation, he slipped a piece of paper into your hand, sprinkled a powder over your head, and FLASH!!!!

Here you are, lost and alone in the Catacombs of the Phantoms...

Looking at the paper you read the following:

## INSTRUCTION CODES

Enter:

The room number — to move to another room (0-59).

Any negative number — to use a stairway.

77 — to fight a monster.

88 — to search a chest for gold.

99 — to drink from a fountain.

102 — to use a worm.

200 — to leave from the natural exit in Room 0.

It doesn't make much sense to you but you're determined to do your best to survive, so you resolve to faithfully shout "77" every time you attack a vile monster.

## VARIABLES

A(x,0): Holds monster seed number.

A(x,1): Holds treasure seed number.

A(x,2): Holds trap seed number.

A(x,3): Holds phantom seed number.

A(x,4): Holds seed number for stairway matching.

A(x, 5-9): Holds seed numbers for tunnel matching.

P(59): Used in searching for tunnels.

STAI(7): Seed in stairway searching.

A\$: Input for yes/no question.

B\$: Holds player name.

B: Chest opening flag.

DD: Agility penalty if caught by trap.

DRINK: Number of drinks allowed from a fountain.

E: Used to place secret room and then room colors.

EX: Your experience.

F: Counter for stairway searches.

G: Fountain present flag.

GA: Giant agility bonus.

GD: Giant present flag.

GN: Giant counter.

GO: Your gold.

GOT: Amount of gold you get when you open chest.

H: Dummy variable for GET.



HEAL: Number of uses of fountain finding slate.

I/J: For/Next loop variables.

JUMP: Agility value needed to avoid trap.

LEVEL: Current level.

LR: Defines level for room/tunnel searches.

MH: Monster's base hit on you.

MIR: Mirror possession flag.

MST/MAG: Monster's strength and agility.

Q: Current room.

R: Command input variable.

ST: Seed number for computing monster and trap values.

STONE: Gargoyle stoner possession flag.

T: Number of giants guarding the goddess.

TR: Amount of gold in chest.

TYPE: Directs program to type of monster or trap.

W: Mirror use flag.

WORM: Number of worms remaining.

X: Flag to stop agility addition by fountain.

YH: Your base hit on the monster.

YST/YAG: Your strength and agility.

## DOCUMENTATION

Lines 5-70: Set up the array which holds the values for the room contents and tunnels of the catacombs.

Line 100: Initializes various variables.

Lines 110-270: Represent the main body of the program. They tell you which room you are in and what is to be found in that room. Lines 170-210 search to see which rooms are connected by tunnels to the one you're in. Lines 215-230 print out those rooms. Lines 240 and 250 lead to a stairway search subroutines and Line 260 is an error trap for inputs at line 270.

Lines 275-325: Are the command

sorter, directing the program to other routines dependent upon your command.

Lines 330-361: Represent the movement routine. In Line 330 the program checks to see if the attempted movement was legal. Lines 340-360 check to see if the monster hit you if you tried to run.

Lines 370-415: Are the monster fighting routine. Line 402 saves the monster's new value, if you should decide to leave if half dead. This is set to 0 when the monster is killed. Note the effects of agility in Lines 380-395. Lines 420-440: Determine how much gold you get out of the chest. Line 420 prevents you from opening the chest more than once while in the room. Line 422 directs the program to check for traps.

Lines 460-490: Check to see if the mirror can locate the Golden Goddess. It too can be used only once per room occupation (Line 462).

Lines 510-520: Are the worm digging routine. Note that the use of worms is not without sometimes unexpected and unpleasant side effects. If you are digging into a room that you have previously explored which had other stairways, it is almost certain that the worm will wreck havoc with the other stairs, closing some or directing them to other rooms.

Lines 530-540: Are where the gargoyle stoner is employed.

Lines 550-570: Are the fountain search routine. Note that the program will only tell you about the first fountain that it finds on each level. The others you will have to find the hard way.

Lines 580-600: Make sure that there is a stairway to the room that you want to go to and then take care of the movement.

Lines 1000-1097: Are the monster setup routine. Note that the type of monster and its strength are derived

from a single two-digit number.

Lines 1100-1110: Determine how much gold is actually in a room's treasure chest.

Lines 1200-1275: Determine if the treasure chest is trapped. Again the type of trap and its severity are stored in a single number.

Lines 1340-1410: Are the phantom routines. Lines 1340-1350 tell you the phantom is there and then 1355 determines which phantom you are confronting. Note that the earth tremor (Lines 1360, 1365) will probably close more tunnels than it opens. If you prefer a more generous tremor, reduce the 75. Also note that the phantom in any room can only be talked to once.

Lines 1500-1540: Are the fountain routine. Note the experience-trading routine in lines 1515-1530. On the first drink in a room you will gain some agility, but on subsequent drinks everything will go toward strength. Also note that you can drink a fountain dry, and if you do it disappears forever. The maximum number of drinks per time in a room is three, but it may be only one.

Lines 1600-1640: Check to see if, after winning or dying, you want to try again.

Lines 2000-2075: Print out the special information associated with the Golden Goddess's room.

Lines 2500-2520: Check to see if there are stairs leading up from a room and then print them out.

Lines 2600-2620: Check for stairs leading down.

Lines 3000-3015: Just stop the program to let you digest the most recent events.

Lines 4000-4120: Are the introductory subroutine. The pokes in Line 4010 set the color registers. They are used in place of setcolor statements to save memory.

```
5 ? "SETTING UP."  
10 DIM A(59,9),P(59),A$(3),STAI(7),B$(9)  
20 FOR I=0 TO 59:A(I,0)=INT(RND(0)*90)  
:A(I,1)=INT(RND(0)*46)+1:A(I,2)=INT(RND(0)*50):A(I,3)=INT(RND(0)*18)-8  
30 A(I,4)=INT(RND(0)*60)  
40 FOR J=5 TO 9:A(I,J)=INT(RND(0)*50)+1  
50 NEXT J:NEXT I  
60 E=INT(RND(0)*30)+30:A(E,0)=98:T=INT(RND(0)*3)+1  
70 GOSUB 4000  
100 Q=7:LEVEL=1:YST=100:YAG=12:EX=0:GD=0:MIR=0:WORM=3:GN=0  
110 E=INT(RND(0)*16)  
111 POKE 752,1:GA=0:GD=0
```

```
112 ? "J"? B$;" YOU ARE IN ROOM ";Q;"  
, LEVEL ";LEVEL:F=0:FOR I=0 TO 7:STAI(I)=0:NEXT I  
120 ? "YOUR STRENGTH IS ";YST:" YOUR  
ABILITY IS ";YAG  
125 ? "EX";EX;" GOLD=";GO;" WORMS=";  
WORM  
127 IF A(Q,0)>89 OR A(Q,0)=-5 THEN 200  
0  
130 IF A(Q,0)<10 THEN ? "NO MONSTER":G  
OTD 140  
135 GOSUB 1000  
140 IF A(Q,1)<=30 THEN GOSUB 1100  
150 IF A(Q,3)=9 THEN GOSUB 1500  
160 LR=(LEVEL*15)-1  
170 FOR I=LR-14 TO LR:P(I)=0:IF Q=I TH  
EN 210
```

```
180 FOR J=5 TO 9:IF A(Q,J)=A(I,J) THEN  
P(I)=1  
200 NEXT J  
210 NEXT I  
215 ? "TUNNELS FROM ROOM ";Q;" TO ";  
220 FOR I=LR-14 TO LR:IF P(I)=1 THEN ?  
I;" ";  
230 NEXT I:  
240 IF LEVEL<>1 THEN GOSUB 2500  
250 IF LEVEL<>4 THEN GOSUB 2600  
260 TRAP 270  
270 ? ? "WHAT DO YOU WANT TO DO";:IMP  
UT R  
275 IF R=103 THEN 530  
280 IF R>77 AND A(Q,0)>=10 THEN ? "CAN  
'T DO THAT YET.":GOTO 270  
290 IF R=77 AND A(Q,0)=10 THEN 370  
continued on next page
```

continued from previous page

```
297 IF A(Q,0)<10 AND RND(0)<0.45 AND A(Q,3)>0 AND A(Q,3)<9 THEN 1340
300 IF R=88 THEN 420
302 IF R=99 AND G=1 THEN 1510
305 IF R=101 THEN 460
310 IF R=102 THEN 510
315 IF Q=0 AND R=200 THEN ? "YOU'RE OUT":GOTO 2075
320 IF R=104 THEN 550
325 IF R<0 THEN 580
330 IF P(R)<>1 THEN ? "CAN'T DO THAT!":GOTO 270
340 IF A(Q,0)>10 AND RND(0)<0.5 THEN ? "MONSTER HIT YOU":YST=YST-INT(RND(0)* (MST/3)):FOR I=1 TO 50:NEXT I:GOTO 360
350 IF A(Q,0)>10 THEN ? "ESCAPED"
360 IF YST<1 THEN 396
361 G=0:W=0:X=0:B=0:Q=R:LEVEL=INT(Q/15)+1:DRINK=INT(RND(0)*3)+1:SETCOLOR 2,E,4:GOTO 110
370 YH=INT(RND(0)*YST)+INT(YAG/3):MH=INT(RND(0)*MST)+INT(MAG/2)
380 IF MAG>YAG THEN 395
390 MST=MST-YH:YST=YST-INT(RND(0)*MH):GOTO 396
395 ? "MONSTER ATTACKS!":YST=YST-MH:MS T=MST-INT(RND(0)*YH)
396 IF YST<1 THEN ? "YOU'RE DEAD!!!":G
```

```
430 GOT=INT(RND(0)*TR)+YAG:IF GOT>TR THEN GOT=TR
440 ? "YOU GOT ";GOT;" PIECES OF GOLD.":TR=TR-GOT:A(Q,1)=INT((TR/2)/LEVEL):G O=60+GOT:B=1:GOTO 3000
460 IF MIR<>1 THEN 270
462 IF W=1 THEN ? "WON'T WORK NOW":GOT 0 270
465 ? "I AM SEARCHING FOR THE GOLDEN G ODDESS.":W=1
470 IF RND(0)<0.55 THEN ? "I CAN'T FIN D IT.":GOTO 270
480 FOR I=0 TO 59:IF A(I,0)=98 THEN ? "IT IS IN ROOM ";I;", LEVEL ";INT(I/15)+1
490 NEXT I:GOTO 270
510 IF WORM=0 THEN ? "YOU USED ALL YOU R WORMS.":GOTO 270
512 ? "THE WORM DIGS A BIG HOLE TO ROO M ";Q+15
520 A(Q+15,4)=A(Q,4):WORM=WORM-1:GOTO 3000
530 IF STONE<>1 THEN 270
535 IF TYPE<>7 THEN ? "SILLY!":GOTO 27 0
540 ? "GARGOYLE IS DEAD!":A(Q,0)=0:STO NE=0:GOTO 270
```

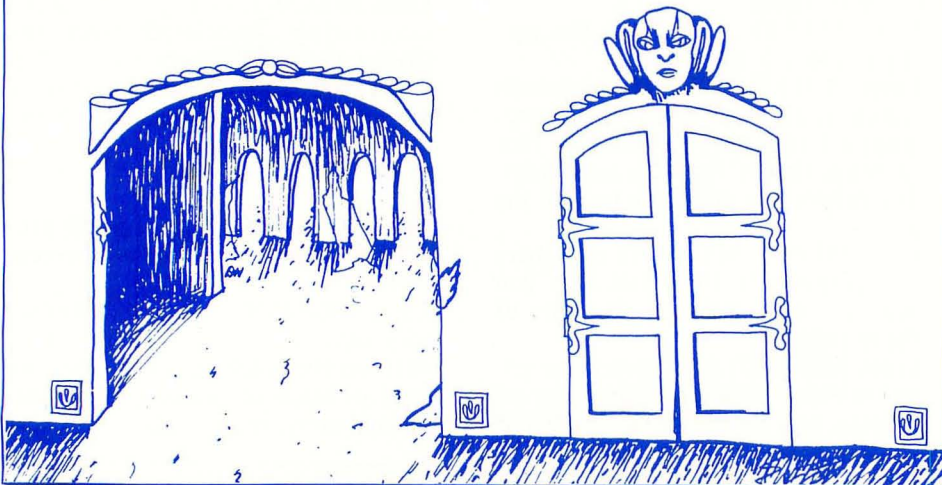
```
1010 MST=ST*TYPE*LEVEL+TYPE:MAG=ABS(ST -TYPE)*LEVEL+6A
1020 ? ? "THERE IS A ";
1030 ON TYPE GOSUB 1060,1065,1070,1075 ,1080,1085,1090,1095,1097
1040 ? " IN THE ROOM.":? "MONSTER'S ST REMGTH=";MST:? "MONSTER'S ABILITY=";MA G:RETURN
1060 ? "GOBLIN":RETURN
1065 ? "WOLF":RETURN
1070 ? "ORC":RETURN
1075 ? "BEAR":RETURN
1080 ? "OGRE":RETURN
1085 ? "TROLL":RETURN
1090 ? "GARGOYLE":RETURN
1095 ? "DEMON":RETURN
1097 ? "GIANT":RETURN
1100 TR=A(Q,1)*LEVEL*2
1110 ? "THERE IS A TREASURE CHEST IN T HE":? "ROOM.":RETURN
1200 TYPE=INT(A(Q,2)/10):ST=A(Q,2)-TYP E*10
1210 IF ST>5 THEN RETURN
1220 ? "THE CHEST WAS TRAPPED WITH ";
1230 ON TYPE GOSUB 1260,1265,1270,1275
1240 IF RND(0)<JUMP THEN ? "YOU AVOID E D THE TRAP!":RETURN
1245 ? "IT GOT YOU!!! YOU SUFFER ";ST* LEVEL+TYPE;" DAMAGE."
```

```
1250 YST=YST-(ST*LEVEL+TYPE):YAG=YAG-D D:RETURN
1260 ? "A POISONED":? "BLADE.":JUMP=0. 6:DD=0:RETURN
1265 ? "FIRE.":JUMP=0.4:DD=1:RETURN
1270 ? "A BOULDER":? "FROM THE CEILING .":JUMP=0.45:DD=2:RETURN
1275 ? "POISONED":? "GAS.":JUMP=0.25:D D=3:RETURN
1340 ? "THERE IS A Phantom IN THE CORN ER.":? "IT'S INVISIBLE EXCEPT FOR A TR ENCH"
1341 ? "COAT, SLOUCH HAT, GARTERS AND ":? "BLACK STOCKINGS."
1345 ? "DO YOU WISH TO TALK TO HER/IT ":INPUT A#
1350 IF A*(1,1)="N" THEN 300
1355 ON A(Q,3) GOTO 1360,1380,1385,138 5,1395,1400,1400,1385
```

```
1360 ? "SHE SCREAMS AND CAUSES AN EART H":? "TREMOR."
1365 FOR I=0 TO 59:FOR J=5 TO 6:A(I,J) =INT(RND(0)*75):NEXT J:NEXT I:A(Q,3)=0 :GOTO 3000
1380 ? "SHE KISSES YOU, SLIPS A SMALL ":? "MIRROR INTO YOUR HAND AND MUTTERS 'A":? "GUIDE' AS SHE FADES."
1382 ? "THE MIRROR BEARS THE INSCRIPTI ON":? "CODE 101":A(Q,3)=0:MIR=1:GOTO 3 000
1385 ? "SHE KISSES YOU AND SLIPS A KNI FE":? "INTO YOUR BELLY":YST=YST-INT(R ND(0)* (YST/2)):GO=GO-INT(GO/4)
```

```
OTO 1600
400 IF MST>0 THEN ? "YST=";YST;" YAG=" ;YAG:" MST=";MST;" MAG=";MAG:GOTO 402
401 GOTO 410
402 A(Q,0)=INT(((MST-TYPE)/TYPE)/LEVEL +TYPE*10):GOTO 270
410 ? "MONSTER'S DEAD!":A(Q,0)=0:EX=IN T(TX+TYPE*5+ST):IF GD=1 THEN A(Q,0)=-5 :GN=GN+1:GOTO 2002
415 GOTO 3000
420 IF B=1 THEN ? "THAT'S A NO-NO!":GO TO 270
422 IF A(Q,2)>=10 THEN GOSUB 1200
425 A(Q,2)=0:? "THE CHEST SNAPS SHUT I N 5 SEC."
```

```
550 IF HEAL<1 THEN ? "NO GOOD.":GOTO 2 70
555 HEAL=HEAL-1
560 FOR I=LR-14 TO LR:IF A(I,3)=9 THEN ? "HEALING IN ROOM ";I:GOTO 270
570 NEXT I:? "NO HELP ON THIS LEVEL.": GOTO 270
580 ? ? "USE STAIRS TO WHICH ROOM";:I NPUT R:FOR I=0 TO 7
590 IF STAI(I)=R THEN Q=R:LEVEL=INT(Q/ 15)+1:W=0:G=0:B=0:X=0:DRINK=INT(RND(0)*3)+1:SETCOLOR 2,E,4:GOTO 110
600 NEXT I:GOTO 270
1000 TYPE=INT(A(Q,0)/10):ST=A(Q,0)-TYP E*10
```



```

1387 YAG=YAG-2:? "SHE STEALS SOME GOLD
TOO.":A(Q,3)=0:GOTO 3000
1395 ? "SHE BRUSHES AGAINST YOU, PURRI
NG,":? "AND SLIPS A VIAL INTO YOUR HAN
D."
1397 ? "ITS NOTE READS GARGOYLE STONER
.":? "CODE 103.":A(Q,3)=0:STONE=1:GOTO
3000
1400 ? "SHE GIVES A HAUGHTY LAUGH AND
DROPS":? "A SLATE. IT BEARS THE WORDS"
:? "I FIND HEALING.CODE 104."
1410 A(Q,3)=0:HEAL=2:GOTO 3000
1500 ? :? "THERE IS A FOUNTAIN IN THE
ROOM":? "WITH CLEAR RED WATER.":G=1:RE
TURN
1510 IF EX<1 THEN ? "NO DICE.":GOTO 27
0
1512 IF DRINK=0 THEN A(Q,3)=0:? "YOU U
SED IT UP.":GOTO 3000
1515 IF X=0 AND EX>5 THEN EX=EX-5:YAG=
YAG+5
1520 IF EX>0 AND EX<=50 THEN YST=YST+E
X:EX=0:GOTO 1540
1530 YST=YST+50*(1+LEVEL*0.3):EX=EX-50
1540 ? "YOU TAKE A DRINK.":DRINK=DRINK
-1:X=1:GOTO 3000
1600 ? "DO YOU WANT TO TRY AGAIN":;IMP
UT A$
1610 IF A$(1,1)="N" THEN END
1620 ? "THE SAME TUNNELS":;INPUT A$
1630 IF A$(1,1)="N" THEN RUN

```

```

1640 GOTO 100
2000 IF A(Q,0)=-5 THEN ? "THE GIANT IS
DEAD AND ROTTING.":? "IT STINKS!":GOT
0 2050
2002 IF GN>0 AND GN<>T THEN A(Q,0)=99:
? "THERE IS ANOTHER GIANT!":GOTO 2005
2004 IF GD=1 THEN 2040
2005 ? :? "YOU ARE IN THE SECRET ROOM!
":? :? "YOU MUST DEFEAT THE GIANT TO W
IN THE":? "GOLDEN GODDESS OF POWER."
2010 GA=21:GD=1:GOTO 135
2040 ? :? "YOU HAVE KILLED THE GIANT!"
:? "THE GODDESS OF POWER IS YOURS."
2050 ? "A WISH IN THIS ROOM WILL SEND"
:? "YOU HOME."
2060 ? "DO YOU WISH IT":;INPUT A$
2065 IF A$(1,1)="N" THEN 140
2070 ? "CONGRATULATIONS! YOU HAVE CON-
":? "QUERED THE CATACOMBS OF THE PHANT
OMS!"
2075 ? "YOU HAVE ";GD;" PIECES OF GOLD
.":? :? "GOOD LUCK!":GOTO 1600
2500 FOR I=LR-29 TO LR-15
2510 IF A(Q,4)=A(I,4) THEN ? "STAIRS U
P TO ";I;";";STAI(F)=I:F=F+1
2520 NEXT I:?:RETURN
2600 FOR I=LR+1 TO LR+15
2610 IF A(Q,4)=A(I,4) THEN ? "STAIRS D
OWN TO ";I;";";STAI(F)=I:F=F+1
2620 NEXT I:?:RETURN
3000 OPEN #1,4,0,"K:"

```

```

3010 ? "PRESS A KEY TO CONTINUE.":GET
#1,H
3015 CLOSE #1:GOTO 110
4000 GRAPHICS 2:POKE 752,1
4010 POKE 708,166:POKE 711,122:POKE 71
2,228:POKE 710,228
4020 POSITION 6,1:?: #6;"welcome":? #6;
"
to the"
4030 ? #6;"\\\/\\\/\\\/\\\/\\\/\\\/"
4040 POSITION 0,5:?: #6;" catacombs
":? #6;" OF THE"
4050 ? #6;" phantoms"
4055 POSITION 0,9:?: #6;"\\\/\\\/\\\/\\\/
\\\/\\\/"
4059 ? " by TOM PLASSMAN"
4060 FOR I=1 TO 5:FOR J=0 TO 14
4070 POSITION J,J:?: #6;"danger"
4072 SOUND 0,J*2+50,10,8:SOUND 1,150-J
*2,14,8
4075 POSITION J+2,9:?: #6;"run"
4080 POSITION 0,3:?: #6;"\\\/\\\/\\\/\\\/
\\\/\\\/"
4085 POSITION 0,9:?: #6;"\\\/\\\/\\\/\\\/
\\\/\\\/"
4090 NEXT J:NEXT I
4100 SOUND 0,0,0,0:SOUND 1,0,0,0
4105 SOUND 0,210,10,10:FOR I=1 TO 175:
NEXT I:SOUND 0,250,10,12
4106 FOR I=1 TO 170:NEXT I:SOUND 0,0,0
,0
4110 ? "Enter your name in":? " The B
OOK Of The DEAD":;INPUT B$
4120 GRAPHICS 0:RETURN

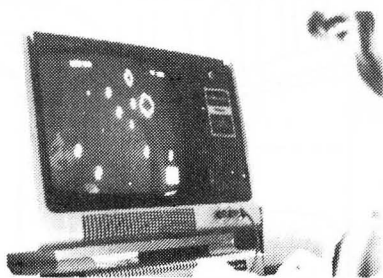
```



Games from **BIG FIVE** will  
turn your computer into a

# TRS-80™ HOME ARCADE

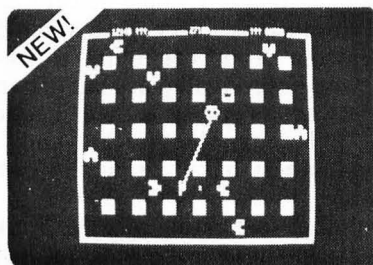
## SUPER NOVA®



"Huge ASTEROIDS have invaded the galaxy! Your mission is to destroy them and the alien saucers before they destroy you!" *Our #1 top selling game!*

\$15.95 16K Mod I or III

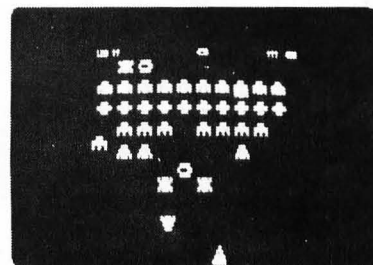
## ATTACK FORCE®



"Eight alien ramships are warping down toward your destroyer ship. You must shoot them down quickly before they crush you!" *With sound!*

\$15.95 16K Mod I or III

## GALAXY INVASION®



"The newest and most exciting Invaders-type game yet! Smooth sound effects, sharp graphics, and the Flagship alien from Super Nova combine to make this our finest TRS-80 game!"

\$15.95 16K Mod I or III



6 South St. Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790



# SAY YOHO

by Scott Adams

The 6th Annual San Francisco Computer Faire and The First Annual Dallas Computer Show are now history, but oh what fun they were!

San Francisco is a fantastic city, a hodgepodge of hills, parks, skyscrapers, and great restaurants. This I'm sure helps to make The Computer Faire what it is every year. But more importantly it's the people. Both exhibitors and attendees were the real reason this Faire is the fantastic success it always is.

This year I had the chance to meet some talented folks who help to make our industry what it is today. Bill Hogue of Big 5 Software, stopped in at our booth and said hello; it gives you a good feeling to meet Bill, as he quite likeable. It sounds like Bill, who has set some of the industry standards for arcade games on the S-80, has some big surprises coming up. Watch for them, they're bound to be great.

I also had a chance to meet Ken and Roberta Williams of On-Line Systems. These are two of the most vivacious people to grace of our industry. I'm sure you'll be hearing even more from them in the upcoming years.

I didn't catch the final figures but I know the attendance was well over 30,000 people. Amazing! We premiered our new six-booth Adventure Island Trading Post and it got a fantastic reception. It felt like every one of those 30,000 came through our booth! The booth is built on the order of a large thatched hut, complete inside with quadrasonic sound system with the sound of waves lapping on the shore and Polynesian music. If you get a chance to attend one of the 30 shows we are exhibiting at this year be sure and stop in and say hi!

The First Annual Dallas Computer Show was a very good show considering it was only its first year. Only one quarter the size of the San Francisco Faire, the attendance was still quite high, with some very interesting exhibits. One such exhibit touches our industry only briefly: It was a computer-controlled four-seater electric car built on the lines of a Maserati with a 100-150 mile range, complete with stereo, air condition-

ing and many other luxury features. Detroit watch out!


While in San Francisco we stopped at Verbatim and learned how they make disks.

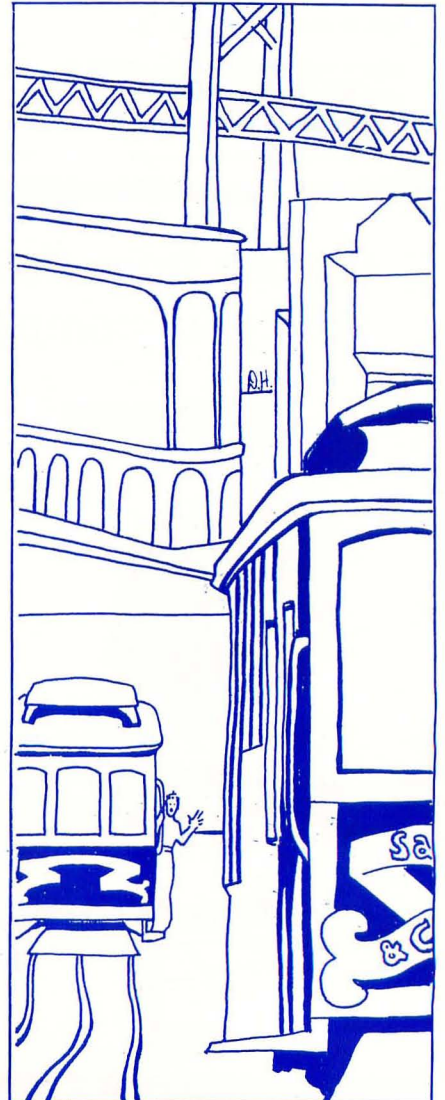
The first step is to take large rolls of Dupont Mylar which are run through what looks like a mud bath. This mud bath contains the magnetic media and the adhesive that will hold it to the Mylar. Depending on the ultimate medium being made (disk or tape) the magnetic poles are aligned for tape or randomized for disk. The next step is the sheet goes through an oven to bake the materials on. Next, the roll is flipped over and the backside is done. The completed roll, called a web, is then taken to where it will be sliced up into many smaller rolls for use as computer tape, or to the disk punching room. Here the disks are punched out of the web by automated machines. The scraps are discarded. The flat disk media are then taken and one by one, have the microscopic hills and valleys smoothed out by hand. From this point on the work is very labor-intensive, passing from hand to hand. Next the disks are put in their sleeves and then sent to certification.

Depending on the disks, DataLife, regular, double density, etc., the disks are hand-loaded into the certification machines and tested. If a disk fails on a side, it is then downgraded: i.e., double-sided disks become rated single-sided or double-density is marked single-density etc. Verbatim keeps a very close watch on the media from beginning to end. The web has a small piece cut off and filed away, each disk is then encoded with both a manufacture date and the web number that it originally came from.

Another interesting piece of information is that besides the problem with the self cleaning sleeve dirtying the disk when you flip it around in a floppy drive, is the fact there is actually a slight bow in all disks too. Verbatim strongly suggests you do not use the backside of a disk in a floppy drive for any important data!

One thing that surprised me was Verbatim's disk drive library. Here they have disk drives from every manufacturer in every model number so that they verify their disks will indeed work on all disk drives! Now

that makes sense! For those who haven't tried them yet, Verbatim's new DataLife disks, are absolutely top quality and I personally recommend them for all uses! While in Dallas I also had the chance to visit MOSTEC and see how they manufacture memory chips. We only had time to see two out of the 14 buildings, but it was an experience to remember. From start to finish the average chip goes through an amazing number of quality-control steps. These folks really want to put out only quality products! We also stopped at one of the many engineering terminals and looked into their games section. Yup, sure enough we found Adventure™, Zork™, Star Trek™ and all the other familiar games of the industry! Happy Adventuring until next month. Yoho and away! 



## Programming Hints

To prevent lock-up due to accidentally typing LLIST or LPRINT, simply POKE 16422, PEEK(16414): POKE 16423, PEEK(16415).

After you do these POKes, the computer ignores the LLIST and LPRINT as commands, and does nothing.

Shane Causer  
Brunswick, GA

To have sound for the S-80, without going to the trouble of Machine Language POKEing and string packing techniques, simply hook the AUX. plug into an audio amplifier and enter the following line: PRINT #1,“”;END. You can put this into a program and use it as an audio prompt for input, or for alarms in a space game. This is extremely useful as a subroutine; sound is achieved by a simple GOSUB.

Shane Causer  
Brunswick, GA

## APPLE ONE LINERS

Integer

```
1 POKE 766,9: POKE 765,32: GR
: FOR I=1 TO 99:C=RND(16)
: COLOR=C: POKE 767,C*3:X=RND
(40):Y=RND(40): PLOT X,Y:
CALL -10473: PLOT 39-X,39-
Y: CALL -10473: NEXT I: GOTO
1
```

Jon Voskuil  
Milford, NH

Applesoft

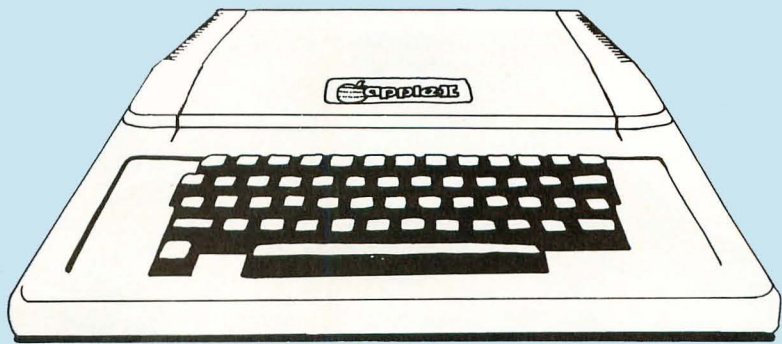
```
1 HGR2 : FOR C = 1 TO 7: HCOLOR=
C: FOR X = 0 TO 270 STEP 10:
X1 = 279 - X: Y = X * (191 /
279): Y1 = 191 - Y: HFPLOT X,Y
TO X1,Y1 TO X,Y1 TO X,Y / 5
TO Y / 2,Y TO 279,Y TO X,Y1
TO Y1,Y1 TO X1,191 TO X1,Y TO
Y,Y1: NEXT X,C: GOTO 1
```

Kris Livingston  
Mason City, IA

Applesoft

```
1 HGR : HCOLOR= RND(3) * 3 + 1
: ZX = RND(3) * 5 + 3: FOR
I = 0 TO 80 STEP ZX: HFPLOT 1
40,0 TO 0,I TO 140,80 TO 0,I
+ 80 TO 140,159: NEXT I: FOR
I = 140 TO 279 STEP 6: HFPLOT
140,159 TO I,0 TO 279,159: NEXT
: GOTO 1
```

Patrick Homer  
Pampa, TX



## COMPUTERS

APPLE II PLUS Computer, 48K RAM (#47-203).....\$1299.00

## PERIPHERALS

SUP-R-MOD RF Modulator (#47-100) .....	\$34.95
APPLE II Disk Controller w/Drive (#47-004) .....	\$619.00
APPLE II Disk Drive (#47-005) .....	\$499.00
LOBO Disk Drive (#47-3101) .....	\$379.00
MICROSOFT Z-80 SoftCard (#47-80) .....	\$269.00
MICROSOFT RAMCard (#47-81) .....	\$159.00
Integer BASIC Language Card (#47-1C) .....	\$195.00
PASCAL Language Card (#47-PAS) .....	\$459.00
MOUNTAIN COMPUTER Apple Clock (#47-MH003) .....	\$269.00
MOUNTAIN COMPUTER ROMWriter (#47-MH015) .....	\$169.00
MOUNTAIN COMPUTER ROMPlus w/Filter (#47-MH007) .....	\$189.00
MOUNTAIN COMPUTER Music System (#47-MH022) .....	\$519.00
MOUNTAIN COMPUTER A/D + D/A (#47-MH023) .....	\$329.00
MOUNTAIN COMPUTER Expansion Chassis (#47-MH024) .....	\$609.00
HARDSIDE Memory Upgrade Kit (#5-1102) .....	\$49.00
APPLE Silentype Printer w/Interface (#47-000) .....	\$569.00
Parallel Printer Card and Cable (#47-9) .....	\$100.00
CCS Arithmetic Processor (#47-7811C) .....	\$399.95
CCS Asynchronous Serial Interface (#47-7710A) .....	\$159.95
CCS Synchronous Serial Interface (#47-7712A) .....	\$159.95
CCS GPIB (IEEE-488) Interface (#47-7490A) .....	\$300.00
CCS Calendar/Clock Module (#47-7424A) .....	\$125.00
LYNX Communications Interface (#19-85) .....	\$229.00
KURTA APPLE Graphics Tablet (#47-1000) .....	\$659.00
VERSAWRITER (#47-1100) .....	\$239.00
ALF AM-II Music Synthesizer (#47-1200) .....	\$189.00
LEEDEX 12" B/W Monitor (#5-100) .....	\$159.00
NEC 12" Monochrome Monitor (#5-200) .....	\$239.00
Color Video Receiver (#26-3010) .....	\$389.00



TERMS: Prices and specifications are subject to change. HARDSIDE accepts VISA & MASTERCARD. Certified checks and Money Orders: Personal checks accepted (takes 3 weeks to clear). HARDSIDE pays all shipping charges (within the 48 states) on all PREPAID orders over \$100.00. On all orders under \$100 a \$2.50 handling charge must be added. COD orders accepted (orders over \$250 require 25% deposit), there is a \$5.00 handling charge. UPS Blue Label, and Air Freight available at extra cost. TRS-80 is a trademark of Tandy Corp.

# AN ENTIRE STAR FLEET FOR \$39.95. YOU'LL LOVE IT!



Like some future starship admiral hurtling through the vast void of hyperspace at speeds beyond comprehension, you are challenged to a battle for cosmic supremacy.

**THE WARP FACTOR**, the latest computer strategy game from SSI, is what every space war fan has been waiting for - the ability to command a star fleet in realistic battle simulation against alien vessels.

It is light years ahead of all

other "space" games because it doesn't just fill your screen with pretty pictures and little substance. **THE WARP FACTOR** is a high-powered tactical simulation that places you squarely in the Captain's role, dealing with the critical parameters of interstellar battle such as sensor and scanner readings; energy allocation for weapons (phasers, disruptor bolts, photon and plasma torpedoes), shields, and warp engines; and battle damage.

**THE STARSHIPS.** With twelve different starship designs - ranging from dreadnoughts and fighters to star bases and base stations - representing five Galactic Empires, you can set up an astronomical variety of confrontations against another player or the computer.

Each class of vessels is awarded a point value to reflect its relative strength so you can assemble fleets of comparable power for a balanced game. Of course, you're free to play the intrepid hero against seemingly hopeless odds - perhaps mere fighters against a star base!

Employing up to ten ships, both sides can give individual or fleet orders, the latter allowing all your ships to execute your commands in unison.

**THE COMPUTER**, aside from being the game's perfect administrator and referee, also serves as your ever-ready, ever-capable nemesis in the multiple solitaire scenarios provided: The Reman Chase (replete with the Cloaking Device, Plasma Torpedoes, and Neutral Zone); Attack on Star Base; Attack on Base Station; and Dogfight.

**THE TWO-PLAYER VERSION** is essentially free-form. With each player choosing starships from a different Empire, you can create scenarios ranging from space skirmishes to a full-scale, all-out star war!

**FOR \$39.95**, **THE WARP FACTOR** is undeniably the most complete and detailed simulation of tactical starship combat yet designed. It comes with the 5¼" program disc; a Starship Operating Manual; 3 Starship Data Cards; and a Game Selection Card - all of which will convert your computer into the gateway to galactic adventure.

## THE WARP FACTOR™. The Universe Awaits Your Command.

While you're at it, you can also get our other games:

FOR YOUR APPLE®:

- Computer Bismarck:** \$59.95.
- Computer Ambush** (a tactical simulation of man-to-man combat in WWII): \$59.95.
- Computer Napoleonic**, the Battle of Waterloo: \$59.95.

- Computer Quarterback** (a real-time strategy football game): \$39.95.
- Computer Conflict** (two modern-day tactical warfare simulations featuring *REBEL FORCE* and *RED ATTACK!*): \$39.95
- Computer Air Combat** (a simulation of air combat in WWII): \$59.95.

FOR YOUR TRS-80®:

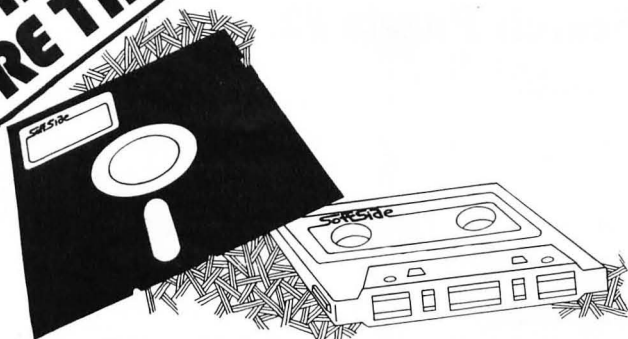
- Computer Bismarck**, 48K Disc: \$59.95. 32K Cassette: \$49.95.

  
**TSE-HARDSIDE**  
6 South St., Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790

Apple is a registered trademark of Apple Computer Inc.

TRS-80 is a registered trademark of Tandy Corporation.

**NOW GET  
MORE THAN BEFORE**



# THE WHOLE IS GREATER THAN THE SUM OF ITS PARTS!

## More Good Reasons to Receive SoftSide With Cassette or Disk

You could always count on **SoftSide** to make it as simple as possible to get great software for your S-80, Apple or Atari computer. Now, when you subscribe to **SoftSide** on cassette or disk (let's just call it "**SoftSide** on Media" from here on), you'll get more than ever. Over 2,000 people have taken advantage of the convenience of having their typing done for them by subscribing to **SoftSide** on Media. We'd like you to sign up too. Here's how we've improved on it.

### Reason #1

#### More programs for you and your computer, already typed in.

(Programs that you won't find in **SoftSide** Magazine)

**SoftSide** Magazine by itself will continue to bring you valuable programming hints; reviews of software and hardware; and code that's ready to type right into your computer, but you'll find many programs on the media that you won't find in the magazine or anywhere else.

We've got lots of programs that we'd love to make available to you through the magazine, but, for one reason or another, they are just too cumbersome to be printed within our pages.

Programs, for instance, that are written in machine code, or a hybrid code (our term for a program that prints fine but lists gibberish), or where the program is data intensive or data continual.

You'll also find programs which were originally published for one particular computer but have since been converted to run on your computer (we'll put them on the media but not in the magazine).

### Reason #2

#### We've Brought Down the Price of Subscribing to SoftSide on Disk.

Imagine that. More product for less money. In order to get you to take full advantage of **SoftSide** in the environment for which it is intended, we've drastically reduced the cost of a disk subscription:

Magazine & Disk,	6 months:	was \$64, now \$55
	12 months:	was \$125, now \$99.95
Magazine & Cassette	6 months:	\$39
	12 months:	\$75

### Reason #3.

The ease of a media subscription frees up your time so that you can do more with your computer (like putting our programming hints and how-to articles in the magazine to work, or maybe even writing for **SoftSide**!).

No more hours of typing. No hunting for typing errors. The programs are tested and ready to go the day you receive them.

We believe that **SoftSide** can have its fullest meaning, and be of maximum value to you when the magazine complements the media and vice versa. Each has a unique character and when blended with the other becomes something more than just a magazine and a disk — **SoftSide** on media may be the only software magazine you'll ever need.

If you are already a **SoftSide** subscriber, you will receive credit for the remainder of your subscription when you send in your order. If you don't currently subscribe, take advantage of our low media rates while they last!

Send check, money order, or bank charge number (don't forget to indicate whether you've got an S-80, Apple, or Atari) to:

**SoftSide Publications**  
6 South Street  
Milford, NH 03055

## Solution to Word-Search Puzzle #2

```

- - - - - T N I - U S R - - T X E N - -
- - - S - - - - - - - - - - - - -
- - - T G O S U B - - K E E P E N - - -
- - - O N E R R - M - - - - - E S - - -
- - N P - - - - - E - - - N - H - - L - -
- - T - - - - R - - - - G T - - - - E -
- - A - - - - - - L D P S R L D - - T -
- - - - - - - - - - - R - U A - N - E -
- - S - - - - - - - - - N - N V - - A L -
- - O - - - - N O T - - D - - - D I M - -
- - C - - - - - E - - - - - - - - - -
C S A - - - - R G - N - - - - - - - -
- - - - - - - I - A - - - - - R - - -
T - - - - - - G - - T - - C - - - E - -
T N - - - - - H - - - - - A - - - S - -
U - I - - - - T - R - - N L - - - T - -
F R - R - - - $ - E - I - L G O T O - -
N - E - P - - - - T S - P - - - B R - -
I D - S - - - - U - E - - A - E - - -
- - A - U C - - - R - - T E T - - - $
- - - E H M - - - N - T S R - - - - D
- - - R R - E $ - - N - F - - - - I
- - $ E O - - - T - O - - - - - $ - M
- - - V - - - F C - N S - - - - R - -
- - - A - - - E - E - L - - - - T - -
- - - S - - - L L - - C - - - - S - -
- - - - - - - D - - - N - E - - S - -
- - - - - - - E - F O - S M - - Q E -
- - - - - - - L - - - I - D O - R - K -
- A T A D - - - E - - - N H P - - O -
- O - - - - - T - - - - E - - - P -
- D - T - - - - E - - - - F P X E - C -
- A P O P - - - - - W - - - O - - L -
- O - - - - - - - E - - - R - I E -
- L - N - L - - - - N - - - N - A -
- - - F - - O T - - - - - K - - R -
- - - F - - G S - - - - - E - - - -
- - - E - - S B A I - - - Y - - - -
- - - D - - - - - L - $ - - - - -
- - - - - - - - - - - - - - - -

```

### WORD LIST

ABS	NEW
AND	NEXT
ASC	NOT
ATN	ON
CALL	ONERR
CHR\$	OR
CLEAR	PDL
CLS	PEEK
CONT	POKE
COS	POP
DATA	POS
DEFFN	PRINT
DELETE	READ
DIM	REM
ELSE	RESTORE
END	RESUME
EXP	RETURN
FOR	RIGHT\$
FRE	RND
GET	RUN
GOSUB	SAVE
GOTO	SGN
HOME	SIN
IF	SQR
INKEY\$	STEP
INPUT	POTS
INT	STR\$
LEFT\$	TAB
LEN	TAN
LET	THEN
LIST	TO
LOAD	USR
LOG	VAL
MID\$	

This month's puzzle is on page 96.







# CUSTOM TAPES

by Ed Ting

Gee, that's a great program. I've just got to save it for my permanent tape collection. Now to get a tape. Hmm... The only thing I have is that brand new 60-minute tape I bought yesterday. Can't waste that. I know, I'll just put it on this other tape with ten other programs on it. No, that won't work either; I'd never be able to find it. Guess I'll have to use up my new tape. But wait! I'm using the new tape for my program submission! This is going to mean another trip down to the record store...

I'll bet something like that happens to you every time you want to save a new program. Most of us end up bunching programs together on one tape or using one program per tape. The first method is excruciatingly inefficient and confusing, the second too expensive and wasteful. Wouldn't it be nice to have short lengths of tape for individual programs, just like the kind you get from The Software Exchange?

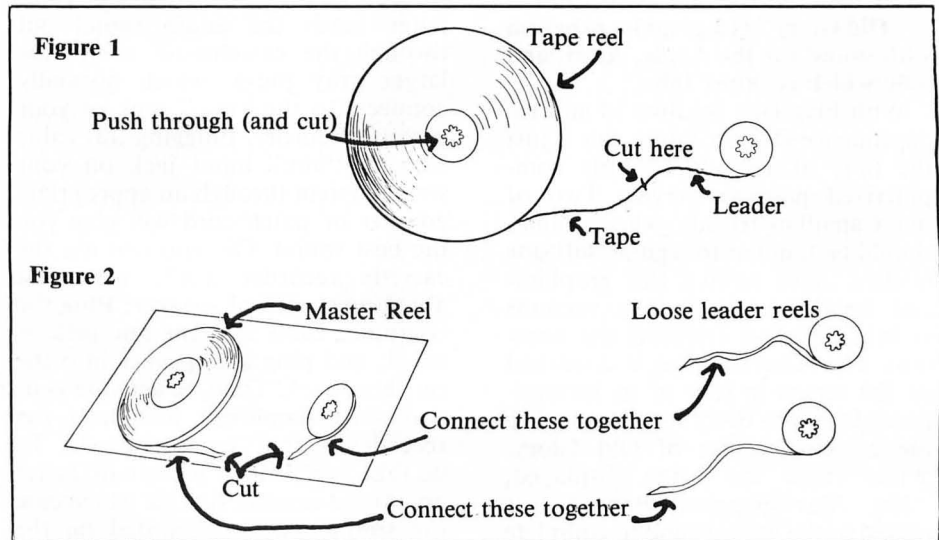
Well, now there's a way. With just a little bit of patience and practice, splicing your own tape can be both economical and enjoyable. The only items you'll need are:

- 1) a tape splicer (\$1.99 from Radio Shack, cat. no. 44-216)
- 2) splicing tape (69¢ from Radio Shack, cat. nos. 44-1125 — 44-1127)
- 3) a handful of old cassette cases

These cases, even if you don't have any, should be easy to obtain. There are probably some old audio cassettes lying around the house that you'll never use. You can also buy them at cheapo department stores. They often sell three for a buck, so you can stock up on them quite easily. Just make sure that:

- 1) they have plastic windows on the side, not flimsy slip-sheets which can damage tape, and
- 2) they are the screw-together type.

Here's how it's done. You'll need a "master reel" from which you'll splice out short lengths of tape to be fitted into your individual cassette housings. I have had enormous success with TDK's "D" and "AD" series. Get the 30 or 60-minute lengths. Longer lengths are too thin, and they stretch and stick too easily. First, unscrew the case of your master reel, save the screws, and tape



it firmly back together again. This case will be opened often in the future, and peeling off some masking tape is much less annoying than removing screws.

Next, take your individual (cheapie) cassette housings. Unscrew them, save the screws, and open one of them up. Take a pair of scissors and snip off the free empty reel where the leader and tape meet. Then, push out the little plastic wheel through the center of the full reel and snip it in the same place. (See Figure 1.)

Throw away the remaining tape. (Of course, it's always fun to play with it a while before throwing it away.) You now have (or should have) two little plastic wheels with the leaders running off them. Take these two wheels, along with the case, and set them aside for a moment.

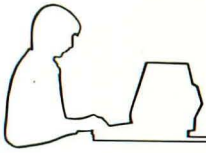
Determine the approximate length of the program on your recorder's digital counter. Place the master reel in the recorder, clear the counter, and fast-forward it until it reads about 15 counts past the length of the program. (For example, if your program is about 40 counts long, then fast-forward the tape until it reads about 55). Take out the master reel and bring it over to where your disassembled case is. Carefully open the master reel and snip the tape in the most convenient location, taking care to touch the least amount of tape possible with your fingers. Connect this end (if this is your first time, that would be the end with the LESSER amount of tape) and attach it to the end of one of the leader reels via your

tape splicer. (The splicer has very thorough instructions on it, so I won't go into details here. I will say this, though. The splicer is supplied with a razor blade for cutting the splicing tape. My advice to you about this razor blade is this — throw it away. Use scissors, they're much easier and they spare you the cut fingers.) (See Figure 2.)

Now (carefully!) transport the spliced system into its new home. Tape up the case, bring it to your computer, and save your program on both sides. (The reason I said "tape", not "screw" the case back together is that something could have gone wrong despite your efforts, and you want to make sure that both copies of the program are working before you screw it back together again.)

While this is happening, you'll want to "close up" your master reel. Connect the remaining plastic leader reel to the master tape reel and tape the whole thing up again. Store it for later use. This set of steps may seem mind-boggling to you at first, but after a few times, it will all be automatic. Your hands will work swiftly without your mind even thinking about it.

Voila! You now have one program on one tape, with no waste, just like the kind you buy! I have built up quite a respectable software collection using this method, and I hope you will, too. People are always asking me how I get tapes in such short lengths. I sit back and smile. Now you know the secret, too.



# OLD GLORY

by William Morris and John Cope

**"Old Glory" is a graphics program with sound for the Apple, Atari, and S-80 which requires 16K.**

With Flag Day on June 14 and Independence Day on July 4, this is just the time of year for a little computerized patriotic fervor. Two of our Canadian friends, whose names should be familiar to regular **SoftSide** readers, have written this graphics-and-sound program in three versions to help all of us celebrate the occasion. The American flag is displayed on the screen in four of its incarnations, from the Betsy Ross original to the current version of Old Glory. While these are being displayed, "The Star-Spangled Banner" is played in the background to complete the effect.

The S-80 version of the program requires an external speaker in order to play the music. This is very easy to hook up, using either a cassette

recorder or a stereo system. The computer sends the audio signal out through the cassette-out cable (the larger gray plug), which normally connects to the "mic" jack of your cassette recorder. Plugging this cable into an "aux" input jack on your stereo system through an appropriate adapter or patch cord will give you the best sound. Or, you can use the cassette recorder itself, with just about any external speaker: Plug the computer cable into the mic jack as usual, and plug the speaker into the earphone jack. Then, to activate your makeshift amplifier, just start the recorder in the "record" mode. To do this, you'll either have to (a) insert an unused cassette first (in which case the sound will be recorded on the cassette as well as played through the speaker), or (b) just reach into the cassette compartment and push back on the little metal or plastic finger in the back left-hand corner, which will

then allow you to press the "record" key.

The Apple and Atari versions are, of course, in full color, and use the normal sound facilities of the machines. The Apple plays through its built-in speaker, and the Atari through the television speaker. There's no question that Atari owners get the best deal with this program, since they get the whole show not only in color but with four-part harmony as well!

## VARIABLES

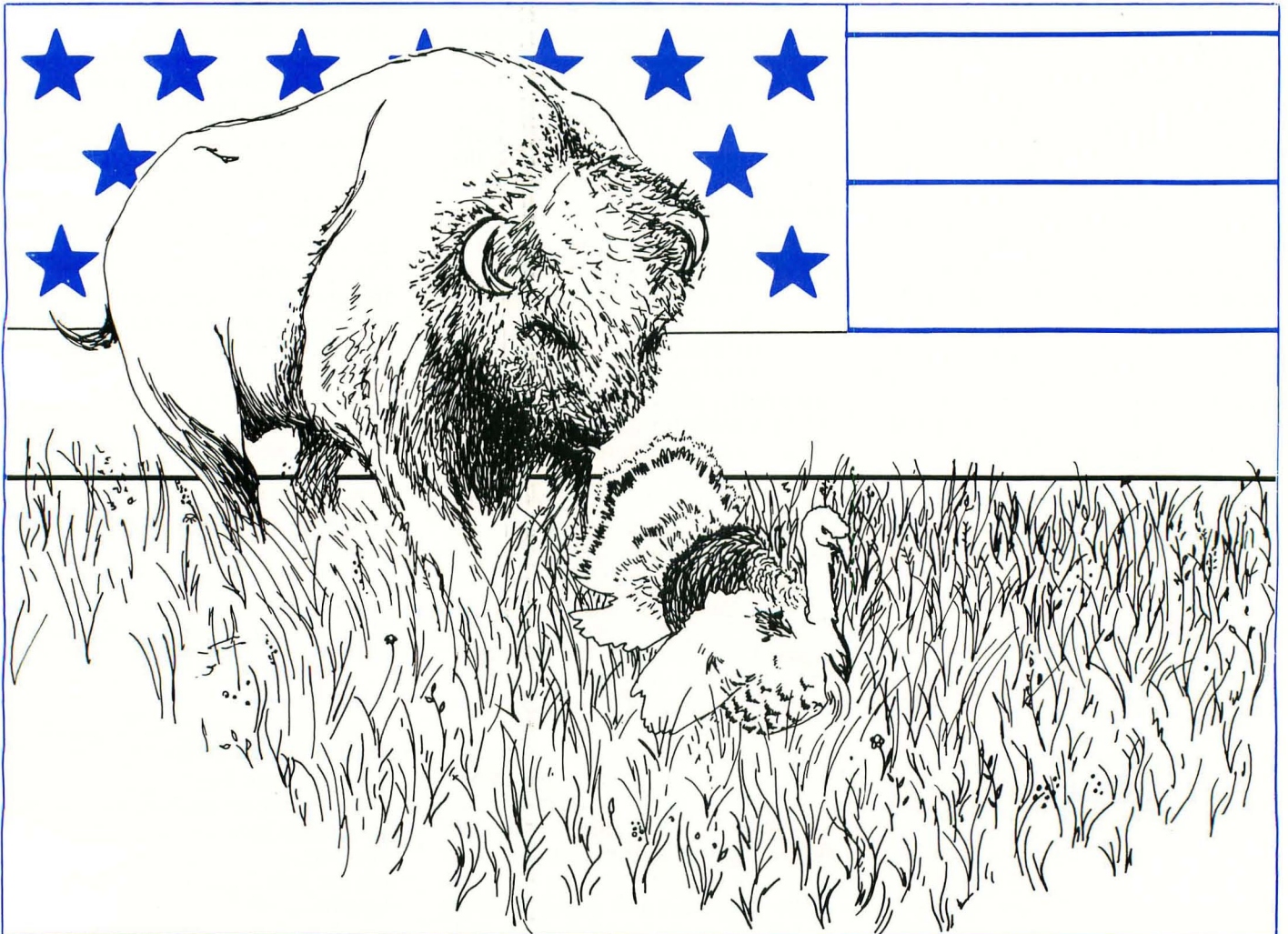
J, K, L, M: Tone variables (Atari version).

JE, JF, JG, JH: Graphics strings (S-80 version).

UL, UN: Length and pitch of notes.

V: Counter which keeps track of data pairs read.

W, X, Y, Z: Used for plotting positions and loop counters.



### S-80 Version

Set error-handling routine and print title.

```
10 CLS: CLEAR1000: DEFSTR A-J: DEFINT V-Z: ON ERROR GOTO 19999
20 PRINT CHR$(23): PRINT @B6, "OLD GLORY": PRINT @456, "THE STAR SPANGL
ED BANNER": PRINT @B98, "(C) WM. MORRIS & J. COPE 1981": FOR Z=1 TO 200
0: NEXT
```

Set up graphics strings and draw the basic outline of the flag.

```
100 CLS: JA=CHR$(170)+STRING$(62,129)+CHR$(149): JB=CHR$(170)+STR
ING$(24,32)+STRING$(38,188)+CHR$(149): JC=CHR$(170)+STRING$(24,32)
+STRING$(38,131)+CHR$(149): JD=CHR$(170)+STRING$(62,188)+CHR$(149
)
110 JE=CHR$(170)+STRING$(62,131)+CHR$(149): JF=CHR$(138)+STRING$(
62,132): JG=CHR$(170)+STRING$(24,32): JH=CHR$(170)+STRING$(24,188)
120 PRINT JA;: FOR Z=1 TO 3: PRINT JB; JC;: NEXT: FOR Z=1 TO 3: PRINT JD; JE;: N
EXT: PRINT JF;: POKE 16255, 133: GOSUB 30000
```

Draw the stars for the Betsy Ross flag.

```
160 FOR Z=1 TO 13: READ X, Y: PRINT @X, CHR$(Y);: NEXT
170 PRINT @978, "THE BETSY ROSS FLAG 1777";
```

Read in data for notes to be played. Call subroutine to play notes and subroutines to draw different versions of the flag.

```
200 READ UN, UL: GOSUB 30100: V=V+1: IF V=256 GOSUB 300
210 IF V=506 GOSUB 400
220 IF V=736 GOTO 500
230 GOTO 200
```

Read data for notes and for concurrent plotting points for the 1981 flag.

```
240 READ UN, UL, W, X, Y, Z: GOSUB 30100: SET(W, X): SET(W+1, X): SET(Y, Z): SE
T(Y+1, Z): GOTO 240
```

Draw the Star-Spangled Banner, adding stripes and extending the field.

```
300 GOSUB 600: PRINT @448, JG;: PRINT @512, JG;: FOR Z=4 TO 36 STEP 16: FORY=4
TO 24 STEP 10: SET(Z, Y): SET(Z+1, Y): NEXT
310 FOR Z=12 TO 44 STEP 16: FORY=9 TO 19 STEP 10: SET(Z, Y): SET(Z+1, Y): NEXT
NEXT
320 PRINT @768, JE; JD; JE; JF;: POKE 16383, 133: PRINT @977, "THE STAR SPA
NGLED BANNER 1795 ";: RETURN
```

Draw the revised flag of 1818.

```
400 GOSUB 600: PRINT @512, JE; JD;: FOR Z=4 TO 44 STEP 10: FORY=2 TO 18 STEP 5: S
ET(Z, Y): SET(Z+1, Y): NEXT: NEXT
410 PRINT @768, JE; JF; CHR$(133); CHR$(255): POKE 16319, 32: PRINT @980, "
THE REVISED FLAG 1818";: RETURN
```

Draw the current version of Old Glory.

```
500 GOSUB 600: PRINT @980, " OLD GLORY 1981 ";: GOTO 240
Subroutine to erase stars.
```

```
600 PRINT @0, JA;: FOR Z=64 TO 384 STEP 64: PRINT @Z, JG;: NEXT: PRINT @448, JH
;: RETURN
```

Data for notes and plotting points.

```
1000 DATA 17, 121, 45, 14, 255, 33, 1, 1, 45, 122, 237, 97, 67, 16, 254, 237, 10
5, 67, 16, 254, 61, 32, 243, 21, 32, 239, 201
1010 DATA 12, 141, 17, 177, 84, 176, 213, 131, 276, 140, 338, 140, 398, 131, 39
4, 131, 327, 140, 260, 140, 195, 131, 68, 176, 7, 177
1020 DATA 162, 19, 193, 10, 243, 18, 193, 20, 162, 22, 121, 36
1030 DATA 96, 25, 108, 13, 121, 25, 193, 20, 173, 21, 162, 31
1040 DATA 162, 19, 162, 11, 96, 35, 108, 19, 121, 26, 128, 35
1050 DATA 144, 20, 128, 13, 121, 26, 121, 26, 162, 22, 193, 20, 243, 22
1060 DATA 162, 19, 193, 10, 243, 18, 193, 20, 162, 22, 121, 36
1070 DATA 96, 25, 108, 13, 121, 25, 193, 20, 173, 21, 162, 31
1080 DATA 162, 19, 162, 11, 96, 35, 108, 19, 121, 26, 128, 35
1090 DATA 144, 20, 128, 13, 121, 26, 121, 26, 162, 22, 193, 20, 243, 22
1100 DATA 96, 25, 96, 14, 96, 29, 91, 30, 81, 31, 81, 39
1110 DATA 91, 21, 96, 29, 108, 27, 96, 29, 91, 30, 91, 42
1120 DATA 91, 30, 96, 35, 108, 19, 121, 26, 128, 35
1130 DATA 144, 20, 128, 13, 121, 26, 193, 20, 173, 22, 162, 31
1140 DATA 162, 22, 4, 2, 12, 2, 121, 26, 20, 2, 28, 2, 121, 26, 36, 2, 44, 2, 121, 1
8, 8, 4, 16, 4, 128, 18, 24, 4, 32, 4, 144, 24, 40, 4, 40, 4, 144, 24, 4, 6, 12, 6, 144
, 24, 20, 6, 28, 6
1150 DATA 108, 27, 36, 6, 44, 6, 91, 21, 8, 8, 16, 8, 96, 20, 24, 8, 32, 8, 108, 19,
40, 8, 40, 8, 121, 18, 4, 10, 12, 10, 121, 26, 20, 10, 28, 10, 128, 35, 36, 10, 44, 1
0
1160 DATA 162, 16, 8, 12, 16, 12, 162, 16, 24, 12, 32, 12, 121, 31, 40, 12, 40, 12
, 108, 19, 4, 14, 12, 14, 96, 20, 20, 14, 28, 14, 91, 21, 36, 14, 44, 14, 81, 44, 8, 1
6, 16, 16
1170 DATA 121, 18, 24, 16, 32, 16, 108, 19, 40, 16, 40, 16, 96, 35, 4, 18, 12, 18,
91, 21, 20, 18, 28, 18, 108, 27, 36, 18, 44, 18, 121, 45, 36, 18, 44, 18
```

Error-handling routine: When data are all read, this line freezes the display briefly and then RUNS the program again.

```
19999 FOR Z=1 TO 3000: NEXT: RUN
```

Subroutine to poke in Machine Language sound-generating routine.

```
30000 JM="": FOR Z=1 TO 27: READY: JM=JM+CHR$(Y): NEXT: IF PEEK(16396)=20
1 THEN 30030
30010 CMD "T": U=VARPTR(JM): U=PEEK(U+2)*256+PEEK(U+1): IF U>32767 THEN
NU=U-65536
30020 DEFUSR0=U: RETURN
```

Subroutine to generate proper note.

```
30030 U=VARPTR(JM): POKE 16526, PEEK(U+1): POKE 16527, PEEK(U+2): U=PEE
K(U+2)*256+PEEK(U+1): RETURN
30100 POKEU+1, UN: POKEU+2, UL: US=USR(0): RETURN
```

### Apple Version

Set error-handling routine and print title.

```
10 TEXT : HOME : GOSUB 3000: ONERR
GOTO 2000
```

```
20 VTAB 3: HTAB 16: NORMAL : PRINT
"OLD GLORY": VTAB 12: HTAB 8
: FLASH : PRINT "THE STAR SP
ANGLED BANNER": VTAB 22: HTAB
6: NORMAL : PRINT "(C) WM MO
RRIS & J. COPE 1981": FOR Z =
1 TO 2000: NEXT
```

Plot the basic outline of the flag.

```
100 GR : COLOR= 2: FOR Z = 0 TO
17: HLINE 0,17 AT Z: NEXT : COLOR=
1: FOR Z = 0 TO 17 STEP 6: FOR
Y = 0 TO 2: HLINE 18,39 AT (Z
+ Y): NEXT : NEXT
```

```
110 FOR Z = 18 TO 36 STEP 6: FOR
Y = 0 TO 2: HLINE 0,39 AT (Z +
Y): NEXT : NEXT
```

continued on next page

continued from previous page

```
120 COLOR= 15: FOR Z = 3 TO 17 STEP
    6: FOR Y = 0 TO 2: HLIN 18,3
    9 AT (Z + Y): NEXT : NEXT
130 FOR Z = 21 TO 36 STEP 6: FOR
    Y = 0 TO 2: HLIN 0,39 AT (Z +
    Y): NEXT : NEXT
```

Plot the stars for the Betsy Ross flag.

```
160 HOME : VTAB 22: HTAB 10: PRINT
    "BETSY ROSS FLAG "; INVERSE
    : PRINT "1777": NORMAL
170 PLOT 8,1: PLOT 10,2: PLOT 11
    ,4: PLOT 12,7: PLOT 12,10: PLOT
    11,13: PLOT 9,15: PLOT 7,15:
    PLOT 5,13: PLOT 4,10: PLOT
    4,7: PLOT 5,4: PLOT 6,2
```

Read in data for notes to be played; pause between versions of the flag (when datum read into UN is zero).

```
200 READ UN,UL: IF UN = 0 THEN FOR
    Z = 1 TO 650: NEXT : GOTO 20
    0
```

Call subroutine to play notes and subroutines to draw different versions of the flag.

```
202 GOSUB 4000:V = V + 1: IF V =
    23 THEN HOME : VTAB 22: HTAB
    7: PRINT "STAR SPANGLED BANN
    ER "; INVERSE : PRINT "1812
    ": NORMAL
205 IF V = 25 THEN GOSUB 300
210 IF V = 50 THEN GOSUB 400
220 IF V = 73 THEN 500
230 GOTO 200
```

Read data for notes and for concurrent plotting points for the 1981 flag.

```
240 READ UN,UL,W,X,Y,Z: GOSUB 40
    00: PLOT W,X: PLOT Y,Z: GOTO
    240
```

Draw the Star-Spangled Banner, adding stripes at the bottom and extending the field.

```
300 GOSUB 600: FOR Z = 18 TO 23:
    HLIN 0,17 AT Z: NEXT : COLOR=
    15: FOR Z = 4 TO 21 STEP 8: FOR
    Y = 1 TO 13 STEP 6: PLOT Y,Z
    : NEXT : NEXT
310 FOR Z = 8 TO 16 STEP 8: FOR
    Y = 4 TO 17 STEP 6: PLOT Y,Z
    : NEXT : NEXT
320 HOME : POKE - 16302,0
330 COLOR= 15: FOR Z = 39 TO 41:
    HLIN 0,39 AT Z: NEXT : COLOR=
    1: FOR Z = 42 TO 44: HLIN 0,
    39 AT Z: NEXT : COLOR= 0: FOR
    Z = 45 TO 47: HLIN 0,39 AT Z
    : NEXT : FOR Z = 1 TO 300: NEXT
    : RETURN
```

Draw the Revised flag of 1818.

```
400 POKE - 16301,0: HOME : COLOR=
    0: HLIN 0,39 AT 39: COLOR= 1
    : FOR Z = 18 TO 20: HLIN 0,1
    7 AT Z: NEXT : COLOR= 15: FOR
    Z = 21 TO 23: HLIN 0,17 AT Z
    : NEXT
410 GOSUB 600: COLOR= 15: FOR Z =
    2 TO 14 STEP 4: PLOT 2,Z: PLOT
    5,Z: PLOT 8,Z: PLOT 11,Z: PLOT
    14,Z: NEXT
430 HOME : VTAB 22: HTAB 10: PRINT
    "THE REVISED FLAG "; INVERSE
    : PRINT "1818": NORMAL : FOR
    Z = 1 TO 300: NEXT : RETURN
```

Draw the current version of Old Glory.

```
500 HOME : GOSUB 600: VTAB 22: HTAB
    16: PRINT "OLD GLORY": PRINT
    : HTAB 18: INVERSE : PRINT "
    1981": NORMAL : COLOR= 15: FOR
    Z = 1 TO 300: NEXT : GOTO 24
    0
598 STOP
```

Subroutine to erase stars.

```
600 COLOR= 2: FOR Z = 0 TO 17: HLIN
    0,17 AT Z: NEXT : RETURN
```

Data for notes and plotting points.

```
1000 DATA 173,48,192,136,208,5,
    206,1,3,240,9,202,208,245,17
    4,0,3,76,2,3,96
1010 DATA 162,192,193,64,243
    ,255,193,255,162,255,121,255
    ,0,0
```

```
1020 DATA 96,192,108,64,121,2
    55, 193,255,173,255,162,255,
    0,0
1030 DATA 162,128,162,128,96,2
    55,108,255,121,255,128,255,0
    ,0
1035 DATA 144,128,128,128,121
    ,255,121,255,162,255,193,255
    ,243,255
1040 DATA 162,192,193, 64,243,
    255,193,255,162,255,121,255,
    0,0
1050 DATA 96,192,108,64,121,255
    ,193,255,173,255,162,255,0,0
1060 DATA 162,128,162,128,96,25
    5,108,255,121,255,128,255,0,
    0
1080 DATA 144,128,128,128,121
    ,255,121,255,162,255,193,255
    ,243,255
1090 DATA 96,192,96,64 ,96,255
    ,91,255,81,255,81,255,0,0
1095 DATA 91,128,96,128,108,255
    ,96,255,91,255,91,255,0,0
1100 DATA 91,255,96,255,108,255,
    121,255,128,255,0,0
1110 DATA 144,128,128,128,121,
    255,193,255,173,255,162,255
1130 DATA 162,255,1,1,4,1,121,2
    55,7,1,10,1,121,255,13,1,16,
    1,121,128,2,3,5,3,128,128,8,
    3,11,3,144,255,14,3,14,3,144
    ,255,1,5,4,5,144,255,7,5,10,
    5
1140 DATA 108,255,13,5,16,5,91,
    128,2,7,5,7,96,128,8,7,11,7,
    108,128,14,7,14,7,121,128,1,
    9,4,9,121,255,7,9,10,9,128,2
    55,13,9,16,9
1150 DATA 162,128,2,11,5,11,162
    ,128,8,11,11,11,121,255,14,1
    1,14,11,108,128,1,13,4,13,96
    ,128,7,13,10,13,91,128,13,13
    ,16,13,81,255,2,15,5,15
1160 DATA 121,255,8,15,11,15,108
    ,255,14,15,14,15,96,255,1,17
    ,4,17,91,255,7,17,10,17,108,
    255,13,17,16,17,121,255,13,1
    7,16,17
```

Error-handling routine: When data are all read, this line freezes the display briefly and then RUNs the program again.

```
2000 FOR Z = 1 TO 5000: NEXT : RUN
```

Subroutine to poke in Machine Language sound-generating routine.

```
3000 FOR X = 770 TO 790: READ Z:
      POKE X,Z: NEXT
3010 HOME : RETURN
```

Subroutine to poke in note and length, and call tone routine.

```
4000 POKE 768,UN: POKE 769,UL: CALL
      770: RETURN
```

### Atari Version

Underlined characters should be typed in reverse video using the Atari logo key.

```
10 TRAP 999:GOSUB 30200:POKE 87,2:POSIT
  ION 6,1:? #6;"old glory"
20 POKE 87,1:POSITION 2,6:? #6;"the st
ar spangled":POSITION 7,7:? #6;"banner
  ":POKE 752,1
30 POKE 87,0:POSITION 5,14:? "(c) Wm.
  Morris & J. Cope 1981"
40 FOR Z=1 TO 3000:NEXT Z:FOR Z=0 TO 3
  :SOUND Z,240,6,8:NEXT Z
100 GRAPHICS 7:SETCOLOR 0,3,0:SETCOLOR
  1,0,8:SETCOLOR 2,7,0:SETCOLOR 4,7,4
110 COLOR 3:PLOT 60,34:DRAWTO 60,0:DRA
  WTO 5,0:POSITION 5,34:POKE 765,3:XIO 1
  8,#6,0,0,"S:"
120 Y=0:FOR Z=0 TO 12:X=Z#5:Y=Y+1:IF Y
  =3 THEN Y=1
130 W=5:IF X<35 THEN W=61
140 COLOR Y:PLOT 154,X+4:DRAWTO 154,X:
  DRAWTO W,X:POSITION W,X+4:POKE 765,Y:X
  IO 18,#6,0,0,"S":NEXT Z
150 FOR Z=0 TO 3:SOUND Z,250,6,40:SOUN
  D Z,0,0,0:NEXT Z:FOR Z=1 TO 500:NEXT Z
160 POKE 752,1:? CHR$(125);CHR$(29);"
  1777 THE BETSY ROSS FLAG":COLOR 2
170 PLOT 32,3:PLOT 39,5:PLOT 46,9:PLOT
  48,16:PLOT 47,23:PLOT 42,29:PLOT 36,3
  2
180 PLOT 24,5:PLOT 19,9:PLOT 16,16:PLD
  T 17,23:PLOT 22,29:PLOT 28,32
200 READ J,K,L,M:SOUND 0,J,10,6:SOUND
  1,K,10,4:SOUND 2,L,10,2:SOUND 3,M,10,2
210 V=V+1:IF V=48 THEN GOTO 300
220 IF V=96 THEN GOTO 400
230 IF V=145 THEN GOTO 500
240 IF V>145 THEN READ X,Y:COLOR 2:PLD
  T X,Y
250 FOR Z=1 TO 100:NEXT Z:GOTO 200
300 ? CHR$(125);CHR$(29);" 1795 THE
  STAR SPANGLED BANNER":Y=1:FOR Z=13 TO
  14:X=Z#5:Y=Y+1:IF Y=3 THEN Y=1
310 COLOR Y:PLOT 154,X+4:DRAWTO 154,X:
  DRAWTO 5,X:POSITION 5,X+4:POKE 765,Y:X
  IO 18,#6,0,0,"S":NEXT Z
315 COLOR 3:FOR X=35 TO 44:PLOT 5,X:DR
  AWTO 60,X:NEXT X:COLOR 2
320 PLOT 13,6:PLOT 29,6:PLOT 45,6:PLOT
  21,14:PLOT 37,14:PLOT 52,14:PLOT 13,2
```

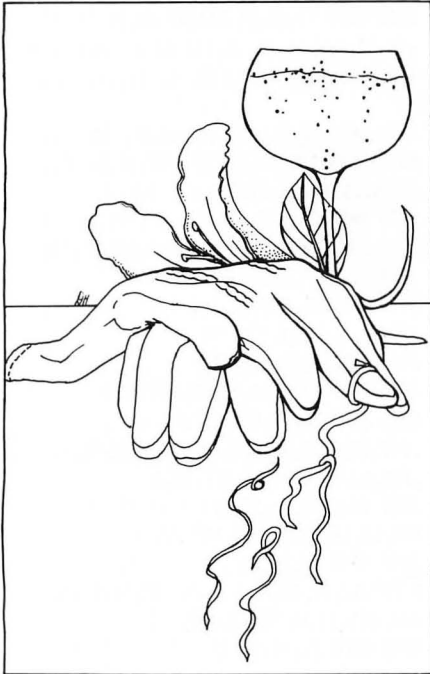
```
2:PLOT 29,22:PLOT 45,22
330 PLOT 21,30:PLOT 37,30:PLOT 52,30:C
  OLDR 2:PLOT 13,38:PLOT 29,38:PLOT 45,3
  8:COLOR 3
350 COLOR 3:GOTO 170
400 ? CHR$(125);CHR$(29);" 1818
  THE REVISED FLAG":COLOR 4:FOR Z=74 TO
  65 STEP -1:PLOT 5,Z:DRAWTO 154,Z:NEXT
  Z
405 COLOR 1:FOR X=44 TO 40 STEP -1:PLD
  T 5,X:DRAWTO 60,X:NEXT X:COLOR 2:FOR X
  =39 TO 35 STEP -1:PLOT 5,X:DRAWTO 60,X
  406 NEXT X
410 FOR Z=7 TO 28 STEP 7:FOR Y=14 TO 5
  2 STEP 9:PLOT Y,Z:NEXT Y:NEXT Z:COLOR
  3:GOTO 320
500 ? CHR$(125);CHR$(29);"
  OLD GLORY "
505 ? " 1981"
510 COLOR 3:GOTO 410
999 FOR Z=0 TO 3:SOUND Z,0,0,0:NEXT Z:
  FOR Z=1 TO 2000:NEXT Z:RUN
1000 DATA 81,108,128,162
1005 DATA 81,108,128,162,96,108,128,16
  2,121,162,193,243,121,162,193,243,96,1
  21,162,193,96,121,162,193
1010 DATA 81,108,128,162,81,108,128,16
  2,60,96,121,144,60,96,121,144,60,96,12
  1,144,60,96,121,144
1020 DATA 47,64,96,153,53,64,96,153,60
  ,72,96,243,60,72,96,243,96,121,193,243
  ,96,121,193,243,85,108,121,217
1030 DATA 85,108,121,217,81,108,128,16
  2,81,108,128,162,81,108,128,182,81,108
  ,128,182
1040 DATA 81,108,81,193,81,108,128,217
  ,47,60,81,243,47,60,81,243,47,60,81,24
  3,53,64,91,217,60,72,96,193,60,72,96
1050 DATA 193,64,81,108,162,64,81,108,
  162,64,81,108,162,72,81,121,162,64,81,
  108,162,64,81,108,162
1060 DATA 60,81,121,193,60,81,121,193,
  60,81,60,193,60,81,60,193,81,108,128,1
  62,81,108,128,162
1070 DATA 96,128,162,182,96,128,162,18
  2,121,162,193,243,121,162,193,243
2000 DATA 81,108,128,162,96,108,128,16
  2,121,162,193,243,121,162,193,243,96,1
  21,162,193,96,121,162,193
2010 DATA 81,108,128,162,81,108,128,16
  2,60,96,121,144,60,96,121,144,60,96,12
  1,144,60,96,121,144
2020 DATA 47,64,96,153,53,64,96,153,60
  ,72,96,243,60,72,96,243,96,121,193,243
  ,96,121,193,243,85,108,121,217
2030 DATA 85,108,121,217,81,108,128,16
  2,81,108,128,162,81,108,128,182,81,108
  ,128,182
2040 DATA 81,108,81,193,81,108,128,217
  ,47,60,81,243,47,60,81,243,47,60,81,24
  3,53,64,91,217,60,72,96,193,60,72,96
2050 DATA 193,64,81,108,162,64,81,108,
  162,64,81,108,162,72,81,121,162,64,81,
  108,162,64,81,108,162
2060 DATA 60,81,121,193,60,81,121,193,
  60,81,60,193,60,81,60,193,81,108,128,1
  62,81,108,128,162
```

```
2070 DATA 96,128,162,182,96,128,162,18
  2,121,162,193,243,121,162,193,243
3000 DATA 47,81,121,243,47,81,47,243,4
  7,81,121,243,47,81,121,243,45,60,81,21
  7,45,60,81,217
3010 DATA 40,60,40,193,40,60,40,193,40
  ,60,81,193,40,60,81,193,40,60,81,193,4
  5,60,81,217,47,60,81,243,47,60,81,243
3020 DATA 53,64,81,162,53,64,81,162,47
  ,60,81,162,47,60,81,162,45,53,81,162,4
  5,53,81,162,45,64,81,162
3030 DATA 45,64,81,162,45,64,81,162,45
  ,60,81,144,45,53,81,128,45,53,81,128,4
  7,60,81,243
3040 DATA 47,60,81,243,47,60,81,243,53
  ,64,91,217,60,81,96,193,60,81,96,173,6
  4,81,108,162,64,81,108,162,64,81,108,1
  62
3050 DATA 72,81,121,162,64,81,108,182,
  64,81,108,182,60,81,121,193,60,81,121,
  193,96,121,144,243,96,121,144,243
3060 DATA 85,108,121,217,85,108,121,21
  7,81,108,128,162,81,108,128,162,81,108
  ,128,162,81,108,128,162
3070 DATA 81,108,81,162,7,5,81,108,81,
  182,17,5,60,81,121,193,27,5,60,81,121,
  193,37,5,60,96,60,243
3075 DATA 47,5,60,96,60,243,57,5
3080 DATA 60,81,121,193,12,8,64,81,121
  ,193,22,8,72,91,121,182,32,8,72,91,121
  ,182,42,8,72,91,72,217,52,8
3085 DATA 72,91,72,217,7,11,72,81,114,
  193,17,11,72,81,114,193,27,11
3090 DATA 53,91,108,217,37,11,53,91,10
  8,217,47,11,45,72,108,217,57,11,47,72,
  114,193,12,14,53,72,108,182
3095 DATA 22,14,60,72,102,173,32,14
3100 DATA 60,81,96,162,42,14,60,81,96,
  162,52,14,64,81,108,162,7,17,64,81,108
  ,162,17,17,64,81,108,162,27,17
3105 DATA 64,81,108,162,37,17
3110 DATA 81,91,128,162,47,17,81,91,81,
  182,57,17,60,81,121,193,12,20,60,81,1
  21,193,22,20,60,81,121,193,32,20
3120 DATA 53,81,128,217,42,20,47,81,12
  1,243,52,20,45,81,121,217,7,23,40,81,1
  21,193,17,23,40,81,121,193
3125 DATA 27,23,40,81,121,193,37,23
3130 DATA 60,96,121,193,47,23,53,91,12
  8,162,57,23,53,91,128,162,12,26
3140 DATA 47,81,121,162,22,26,47,81,12
  1,162,32,26,47,81,121,162,42,26,45,72,
  108,144,52,26,53,91,128,162
3145 DATA 7,29,53,91,128,162,17,29
3150 DATA 60,96,121,243,27,29,60,96,12
  1,243,37,29,60,96,121,243,47,29,60,96,
  121,243,57,29,60,96,121,243,57,29
3160 DATA 60,96,121,243,57,29,60,96,12
  1,243,57,29,60,96,121,243,57,29,60,96,
  121,243,57,29,60,96,121,243,57,29
30200 GRAPHICS 0:SETCOLOR 2,3,1:SETCOL
  OR 4,3,1:UB=PEEK(560)+PEEK(561)*256+4:
  POKE UB-1,70:POKE UB+2,7:POKE UB+3,7
30210 FOR UZ=4 TO 8:POKE UB+UZ,6:NEXT
  UZ:POKE UB+22,65:POKE UB+23,PEEK(560):
  POKE UB+24,PEEK(561):SETCOLOR 3,8,6
30220 RETURN
```



# LEMONADE OR CHAMPAGNE

Will Hagenbuch has been in the software business for some time now, writing business and utility programs such as "File Manager 80", "Accounts Receivable", and "Utility". SoftSide is proud to serialize his book *Lemonade or Champagne*, a guide to the creation of business software.



by Will Hagenbuch

Note: "Lemonade" is an adjective often coupled with the operation of a microcomputer business. Its connotation may sometimes be construed to indicate that the "Lemonade Entrepreneur" is something less than "professional. I take my TRS-80 seriously, and I would hope that, by employment of some of the techniques described in the following pages, we might dispel some of that thinking.

If you are over 25 years of age, then you were born B. C. (Before Computers)! In case you have never pondered this point, let's very briefly explore the history of computers and automated business data processing.

The first computers became available for business use during the middle to late 1950's. Only the "rich" were able to afford these early monsters. However, during the 1960's, miniaturization technology brought us "time-sharing" and the mini-computer. Computers, or at least computer power, became available to the "not-so-rich". But,

during the 1970's, a real phenomenon occurred; the Personal Home Computer placed the heretofore inaccessible or unavailable at the fingertips of anyone with the price of a used automobile — and from the looks of our highways that must mean almost everyone.

Back in the 1960's, I heard the profound statement that "more engineers are living today than have lived during the entire course of history." I'll venture that the same statement will be true of programmers — EVERY YEAR OF THE NEXT DECADE! The Personal Home Computer (which, from here on out, we will refer to simply as the microcomputer) has made, or will make, this possible.

True, the definition of "programmer" has changed somewhat. Today, it is almost anyone who has altered a BASIC statement or "SUPER-ZAPed" memory, but in those early days we were all paid professionals (I'll take poetic license by using the term "professional") because we worked on the "rich guy's" equipment. During this era, we had a big thing about professionalism and how to attain professional status in the eyes of the world, I often wonder if this were really the case, or if we were just jealously guarding a good thing. After all, on the QT (and usually when drinking), one programmer could quite often be heard admitting to another, "Don't tell any outsiders how easy this is, or they will all be doing it!" Well, the phenomena of the 1970's has let the proverbial cat out of the bag and many "outsiders" have found our secret. To those of you who have, I bid you welcome to the "Club".

Personally, I am happy that things have changed. It is very gratifying to visit our local computer shops and see the interest of the many young people and the interest that the computer store personnel take in helping these young folks. If, indeed, the 1970's were phenomenal, what might we expect from the 1980's as these young people matriculate in the field of computers and automation?

This booklet is dedicated to all of you former "outsiders" as well as those of you who have already "paid your dues" in this business. In it, we will attempt to treat some of the old

truisms of System Development in light of the modern-day microcomputer; offer some suggestions that may make your efforts easier and more productive; and, just possibly excite you enough to start you thinking about how you might turn a "Lemonade" computer business into a "Champagne" computer business by adding the touch of professionalism that we have been searching for lo' these last two and one-half decades.

The bottom line of the dedication must, however, go to Jane, my wonderful wife, who maintains the environment in which a booklet such as this, and its supporting software, could be created.

This booklet is about the automation of business data processing. It presents accepted concepts of Systems Development in terms of today's microcomputers. It assumes that you know, or have made up your mind to learn, the fundamentals of BASIC programming. It is, by far, less than a complete analysis of all of the things you would need to know to survive in a competitive world of data processing consulting — however, if that is your aim, it can give you a good start in that direction. If you simply want to use the microcomputer to develop your own business applications, then this booklet will provide some insight as to how the professionals do it — or fail to do it!

This booklet is divided into four sections. Section I provides a discussion of the Data Processing System Development Cycle. In other words, those things to consider if you want to turn your "Lemonade" computer business into a "Champagne" computer business — or things you should know if you just want to design and install your own microcomputer system. It will suggest a systematic approach to System Development whether you are developing for yourself or for others. You will find that the pitfalls of Systems Development are discussed quite candidly and this is what we mean by a "hard look". If you wear the shoes of the "User" or "Client", please don't take offense as to how you have been profiled. In reality, the great majority of Clients are quite naive and must be led through a first-

time installation with great care. However, mark these words, this situation will change significantly during the next decade and Users are going to become increasingly more knowledgeable.

Section II provides five examples of forms which the System Analyst/Programmer may want to use during the System Development effort. The author will be happy to supply full-sized versions of these forms to any reader who desires copies of them. Just send your request to Nepenthe Programs, 3014 Biggs Ct., National City, CA 92050. and, if at all possible, include a brief description of "what you are up to" in the microcomputer field. We like to keep touch with "Lemonade Entrepreneurs".

Section III will focus attention on some of the finer points of Program Development and Document Preparation. It will concentrate mainly on the creation and use of randomly accessed data files. Very little has been written thusfar about the real "nuts & bolts" of using random access files in the "real world"; however, this facet of Program Development is considered to be of paramount importance to the business application programmer.

Section IV will describe several software solutions to ease your programming efforts. These "Software Solutions" are programs for TRS-80, Model I, Disk Systems available from the author. Each of these programs was written by the author and are in every-day use in the "Systems-for-Sale" environment of a Professional Microcomputer Consultant.

## SECTION ONE

The term "Systems", as used throughout this booklet, will refer to a program, or a set of related programs, designed to employ the microprocessor to process data into a usable product. In fact, that is what data processing is all about — rendering raw data into useful information. We treat the development of microcomputer systems with the same seriousness as the development of systems for large-scale computers. The size or cost of the equipment in no way minimizes the planning considerations for a system; The information it produces is only as good as the raw data it is given and the program which processes the data. After all, should we consider the information we get from a \$2500 microcomputer of any less importance than

that which someone else may get from his \$250,000.00 computer?

The development of a computer system is viewed as a systematic procedure. It could be likened to the construction of a building in that the plans must first be committed to paper, the foundation established, and the remainder of the structure erected brick-by-brick. If you have been in this profession for any length of time at all, you will have already experienced what happens when a system, or even a single program, is begun in the middle without considering the consequences of the beginning or the end — about the same thing that happens when a building is constructed beginning with the second story!

The systematic approach to the construction of a computer system is what we will call the "development cycle". A flowchart of the system development cycle is included as Figure 1-1. It consists of all the steps

necessary to insure that a computer system is well-conceived, well-executed, and can be maintained (and even enhanced) throughout its useful life. This development cycle consists of seven distinct, but interrelated, areas of effort: problem definition, system (problem) analysis, system design, programming, unit/system testing, system documentation, and system implementation & maintenance. If this seems like a high-priced mouthful of buzzwords, let me hasten to add that you have already done each of these things for every program you have created — or you have not yet created your first working application program!

## PROBLEM DEFINITION

Problem definition is probably the most important aspect of the development cycle since, if you do not know what problem is to be solved, it is very difficult to come up with a solution. I sometimes think that much of the data processing world is made up of frustrated programmers running around with a solution looking for a problem. Give them one and they will hide in their corner until they have applied one or more of their readymade solutions. The problem here is that if they did not fully understand the problem when they went into the corner, the solution will not be what was expected.

If your computer application is for yourself, then you probably know pretty well what it is that you want to do. Still, if you write it down, you will surely come up with some embellishments that you probably overlooked. It will sure save you a big chunk of time if you think of those embellishments at the beginning of your development effort.

If the development effort is for someone else, then your problem definition efforts increase by some quantum jump. Communication between individuals, and the problems attendant thereto, fill many books. These communication problems are compounded when the "computer expert" and the "subject matter expert", each with his own language, try to communicate. The solution — write it down so that you both can see a clear picture of what is required to solve the problem.

There is a formula that can be used in the problem definition phase of the development cycle. It is 90% listening and 10% talking on your part. You are there to get a comprehension of the prospective client's problem; not

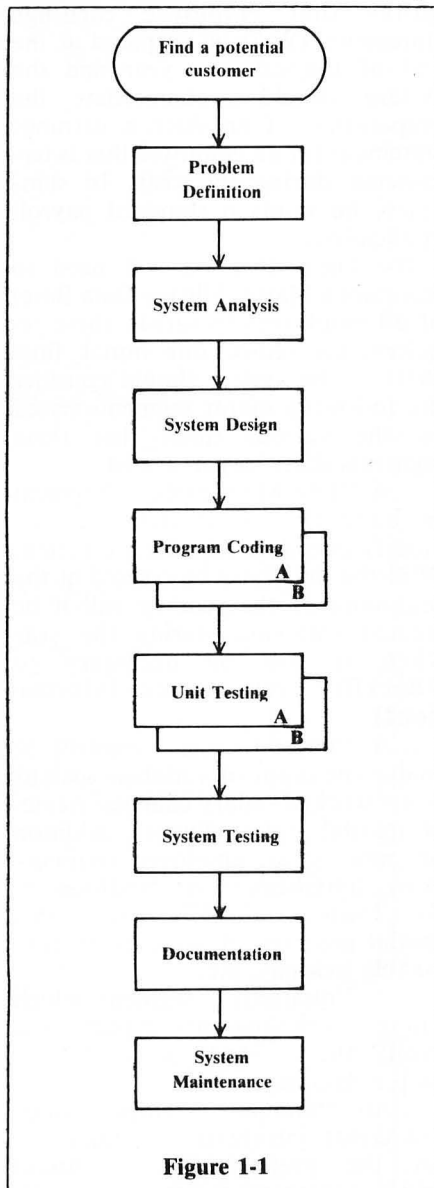


Figure 1-1

continued on next page

continued from previous page

to tell him what he should be doing. Remember that his business has probably been successful thus far without you and the only reason you are invited in is to hear some problem that the businessman has and to see if you can solve it. Don't try to solve the problem on the spot unless you do not want a contract to provide services. Limit your talking to the following.

...Ask questions relative to anything the prospective client has said that you do not fully understand.

...Answer questions asked and when you do not know the answer, write down the question, admit that you do not feel confident to answer at this time, and tell him you will get back with the answer.

**AND DO IT!**

One of the major efforts in problem definition is the collection of the various forms that are required in the data processing effort. During the initial problem definition interview with a prospective client, you will want to collect copies of all forms currently being used; both standard forms used in the particular type industry or application, and forms used in current operations for the application being considered for automation. Not only should you secure one copy of a blank for each of these forms, but you should attempt to secure at least one completed sample form. These forms will subsequently be used for an analysis of the data elements contained thereon as well as formulation of the system test plan.

## SYSTEM ANALYSIS

Some suggested computer applications are best left to manual operation! The System Analysis (or Problem Analysis) phase of the development cycle is often referred to as the feasibility study and should answer the question of automated data processing versus manual data processing. In effect, if tangible benefits of automating cannot be ascertained at this point, there is little reason to proceed with the development cycle because the remaining effort will be for naught. Be candid with the prospective client. If you have reason to doubt the wisdom of automating some facet of the application, tell him your reasons. Of course, if he insists that you proceed with it, you have very few alternatives.

The effort during the Systems Analysis phase is largely concerned with "thinking the system through".

It is most important that you first sketch out a rough flow chart of the proposed system which shows the interaction between the major segments of total system operation. It would be most prudent that you actually do two of these, one showing the way data flows and is processed under today's manual operation and one which shows how you would propose to do it with an automated system. A comparison of these two flow charts should point out the feasibility of automation; if it is indeed feasible.

As an example of the System Analysis process, let's assume that we want to automate a payroll system. We might have been told by the prospective client that he desires a payroll system that will provide weekly payroll registers for both salaried and hourly-rate employees. He wants to accumulate this payroll data and produce the quarterly reports required by both state and local regulatory agencies. He also requires that employee earnings statements (W-2) be prepared at the end of the calendar year and the system should accommodate the preparation of an interim earnings statement for an employee that is terminated during the year. In summary, he wants a standard payroll application.

We know that we will need to maintain a Master File (or Data Base) of all employees to satisfy these requirements. Also, our initial flow chart of the system should consider the following major segments based on the various times that these segments must be performed.

...A "File Maintenance" segment to build the initial data base or modify that data base after creation. (Will the data base be created at the beginning of the year or will it be created sometime during the year when it will be necessary to "Backfit" year-to-date information?)

...A "payroll cycle" segment to enable the input of variables such as hours worked, salary changes, name- or marital status changes, addition of new hires, employee terminations, initialization or modification of "non-standard" deductions, special payment of bonuses or non-taxable sick pay, etc.

...A "quarterly" segment which will produce quarterly reports and "roll" the quarterly accumulations on the data base.

...An "annual" segment which will create information summarized for the preparation of annual employee earnings statements and the

"rolling" of payroll accumulations for the new year.

...An "as required" segment which will enable the preparation of interim employee earnings statements upon demand. This segment should also contain the flexibility to accommodate some of those extra features that may have been overlooked during initial discussions with the client, such as an employee to employee number cross-reference listing or an employee mailing list.

This "overview" flow chart should be in sufficient detail to show the interactions between the various segments and the interfaces with the data base. It should show the reports that are produced by the segments and the input that is required for each of the segments by the operator of the system. Please observe that we are not yet at the "data element" level in our flow charting since we want to jump a very large hurdle before spending time on the "nitty-gritty" detail.

This hurdle is the final discussion with the prospective client. The purpose of this discussion is threefold.

...First, to show the prospective client the potential savings to be realized in automating under your proposed system;

...Second, to make sure that the problem solution you are proposing is what the prospective client has in mind; and,

...Third, to change his status from "Prospective Client" to "Client".

In other words, this should be the final meeting before you begin work on the computer system in earnest and you should walk away from it with a firm understanding of just what is to be done, a positive feeling that you are capable of doing the job in the time frame you have agreed upon, and a firm commitment on the part of the Client that he is going to accept the system when you have finished the work. If you have not "sold" your services to the point where you are able to collect at least twenty percent of the total fees, don't bother working any further for this Client! Either he was just looking for free information to pass on to his programmer-brother-in-law (who, incidentally, has offered to do it on a 4K tape system), or you are not exhibiting the required professionalism. If you lose the case, the reason will, most probably, be hard to ascertain because most of the time you will be



given some fictitious excuse why he is not putting his money on the line. Whatever the reason, if it happens (and it surely will once and a while), you want to do a little additional analysis to insure it was not a question of your professionalism and, if it was, to correct the problem in the future.

The System Analysis phase of the development cycle is where the novice becomes the professional. For example, the system we used for illustration was a payroll application, one of the more prevalent automated data processing applications. When you begin as a novice you either must find existing systems of this type and learn them thoroughly, or you must apply your own inventive genius and do it the hard way. Each time you fall on your face when offering your Super-Duper Payroll System to a prospective Client, you must reappraise what you are offering and update your software so that you will have more chance for success the next time out. Soon you will have a viable package that will survive the test. Subsequent sales of this software (be it Payroll or some other application) will be on the first interview. This is what I mean by "paying your dues".

## SYSTEM DESIGN

System Design is the art of integrating the various segments of the application as conceived in the Systems Analysis phase into a cohesive whole. I call it an "art" because where much of the effort which follows is pure trial and error labor, this phase provides you all of the freedom and innovation that your experience with the hardware and software will allow.

In effect, as in the game of Monopoly, you are at "GO". You have made a professional commitment to a Client, taken his money in good faith, and are ready to find out whether or not you can do it! Of course you can — you put your professional competence on the line, didn't you? But, the point is that an ill-conceived system design, at this point, will cost you time that you did not bargain for in your initial optimism. It is what I call the "Infernal Loop" and it will cause you to pass "GO" more than once without collecting your \$200.00.

Figure 1-2 reflects graphically the seriousness of the "Infernal Loop". The value of "N" is, of course, dependent upon the personal relationship you have with the Client.

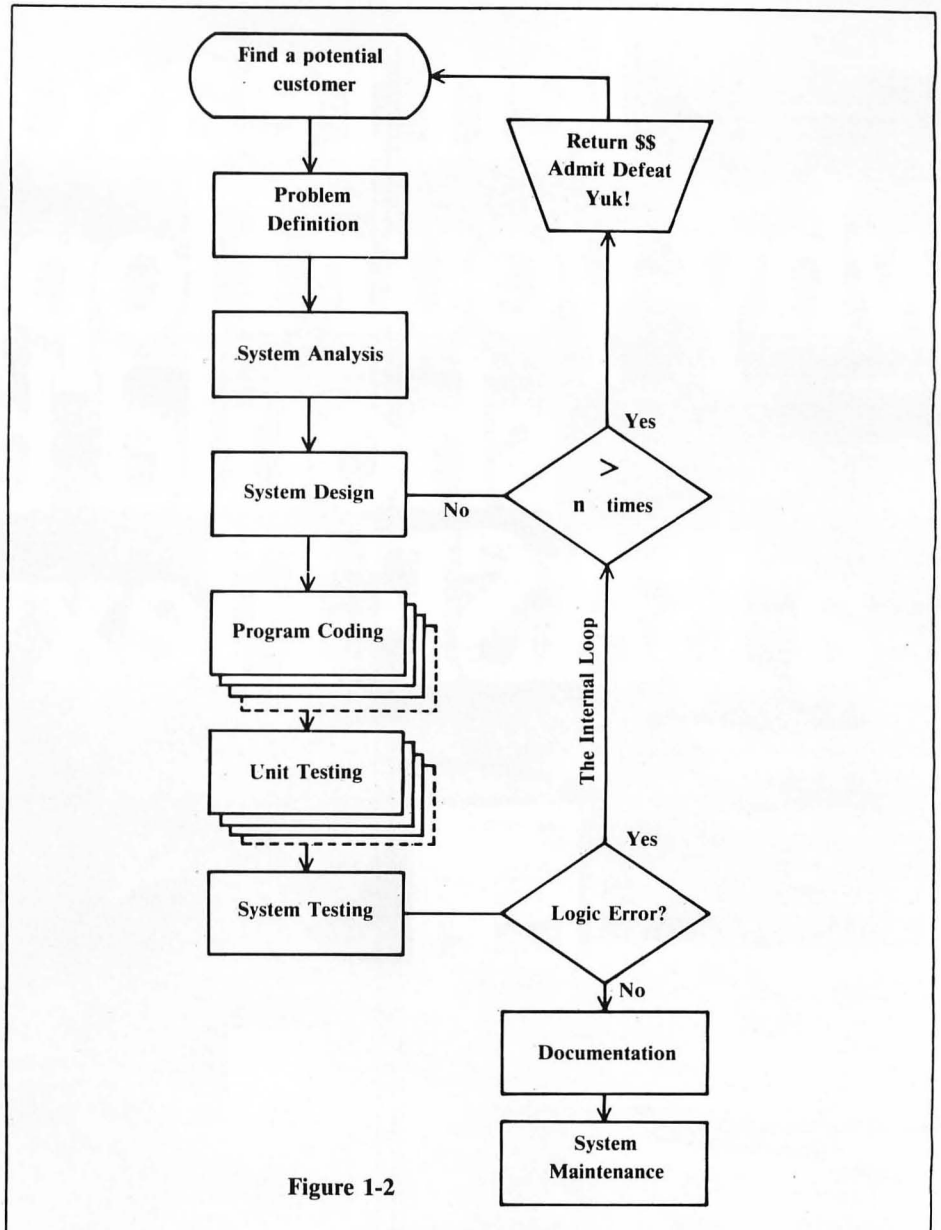


Figure 1-2

Obviously, if you can still complete the project on time (or your Client is a bosom buddy whom you can con into waiting), the effect will not be too serious — just frustrating. However, if you are to be professional, and compete with all those other tigers out there, a faulty system design could prove sickening or even fatal. Unfortunately, the only antidote is experience (or "paying your dues").

Just what are the elements of Systems Design. Well, for an opener, let's call it the nitty-gritty of "how" your system is going to accept data and process it into information. As a starting point we will need to isolate all of the data elements required by the system. We will define a data element as any separate and distinct piece of information required by the system, such as "Employee Name" or "Social Security Account Number" or "Gross Pay".

Remember when we told you to collect documents during your initial meeting with the Client? Examination of these documents will give you a good start in ascertaining the data elements that will be required on both "input" and "output" sides of your systems. In addition, these documents should be studied so that we can ascertain how data is currently being processed. The conversion from a manual to an automated system is a traumatic experience for your Client's employees and, if we can minimize the use of strange new documents, we can lessen this trauma significantly.

Once we have ascertained the data elements that will be required (or at least taken our first cut at doing so), and we have noted them for future reference, we can start to plan our data base(s). A data base is simply a collection of stored information

continued on page 84

# Ten reasons why your floppy disk should be a BASF FlexyDisk<sup>®</sup>.

More than four decades of experience in magnetic media – BASF invented magnetic recording tape, the forerunner of today's wide range of magnetic media, back in 1934, and was the first independent manufacturer of IBM-compatible floppy disks.

Tough Tyvek sleeve – no paper dust, no static electricity.

Special self-cleaning jacket and liner help eliminate data errors and media wear and tear.

Packaging to suit your requirements – standard flip-top box, Kasette 10<sup>®</sup> storage case, or bulk pack.

Center hole diameter punched to more accurate standards than industry specifications, for top performance.

100% certification – every single disk is tested at thresholds 2-3 times higher than system requirements, to be 100% error-free.

Bi-axially oriented polyester substrate – for uniform and reliable performance year after year.

Cross-linked oxide coating – for low head wear and long trouble-free media life.

Total capability – one of two manufacturers in the world that makes both 8" and 5.25" models, has tape and disk experience, and manufactures floppy disk drives.

Double lubrication – lubricants both in the formula and on the disk surface, to minimize media wear due to head friction.

For the name of your nearest supplier, write BASF Systems, Crosby Drive, Bedford, MA 01730, or call 617-271-4030.



# BASF

Floppy Disks Mag Cards Cassettes Computer Tapes Disk Packs Computer Peripherals

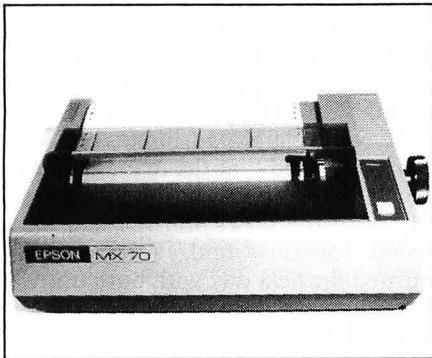


# WHAT'S NEW

by Edward E. Umlor

## HARDWARE

The hardware item for this month is the EPSON MX-70 printer. Do not look for this printer to have all the goodies that the MX-80 has. This is a low-cost printer designed to do all the basic print functions. The price tag is some \$100 cheaper and that money had to come from somewhere.



**MX-70 PRINTER**

Printer operation is in two print sizes only — 80 characters per line and 40 per line. There isn't a condensed mode. You get only single strike and no emphasized mode. The lower case does not have descenders. The graphics mode is not block, but what is commonly referred to as dot addressable graphics. This is the one area the MX-70 outshines the MX-80. You can do pictures with finer resolution and many more shades of gray. This printer is the one for all you graphics people. However, if you are into word processing, the MX-80 would be the better choice.

There is one big problem with the MX-70 when using a cable other than the EPSON cable. Pin 35 in the 36 pin connector end **MUST, REPEAT: MUST** be removed for the printer to operate with the S-80. If you have a 26-1401 cable, then pin 35 has to be removed. When this is done, the operation is normal and you should be very happy.

## SOFTWARE

I have not received any software for review this month. This means

that all I can do is ramble around a bit and see if my BBs still rattle. I have been doing some programming lately using NEWDOS80. I really like this as a programmer's DOS. Let me run down just a few of the goodies when using the modified Disk BASIC:

1. To list a line — L (xxx) xxx: line number.
2. To edit a line — E (xxx).
3. Down arrow — displays next sequential line.
4. Up arrow — displays previous sequential line.
5. Shift up arrow — displays first line of program.
6. DI aaa, bbb — delete line aaa and insert it at bbb.
7. DU aaa, bbb — duplicate line aaa at bbb.
8. RENUM aaa, bbb, ccc, ddd — renumber aaa (first line), bbb (in increments of bbb), ccc (starting at ccc), ddd (last line of block to be renumbered). If nothing is specified, the whole program is renumbered starting at 10, by 10s.
9. REFxnn — will list all the places a variable nn is used in the program. The x code is for screen or printer as the display device.

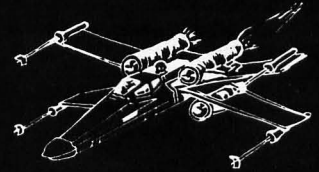
If you happen to get zapped or lock up your keyboard, reboot DOS and type BASIC\* to get back into your program without losing it. There are a lot of niceties for the programmer that have been built into NEWDOS80. Being able to duplicate lines without retyping the whole thing (DU), or being able to move out of place lines by a simple DI command is sure a lot faster and easier. It has gotten to the point that I do all my BASIC programming in the NEWDOS80 BASIC.

Well, I guess that's all for now. I still haven't heard from anyone out there. I thought that this was a fast-moving industry that needed new products to put in front of the user public. Shows you how muddled my thinking can get at times. However, I am still requesting NEW PRODUCT RELEASES and submissions from guys and gals that are using some of these new widgets.

# X-WING

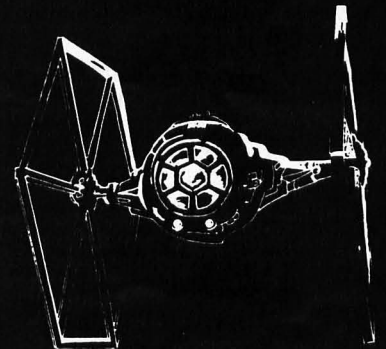


by  
Chris Freund



For the thousands who have enjoyed X-Wing Fighter, X-Wing II presents a totally new element in the game!

You are the pilot of an X-Wing fighter . . . Your Mission, Destroy the Death Star!



Where X-Wing I left Death Star looming on the screen, X-Wing II lets you guide your fighter into the trench, find the exhaust port, aim and fire — all the while avoiding enemy fighters. Excellent graphics, 12 levels of play, and extensive INKEY\$ commands make this one of our most exciting "real time" games.

S-80 16K Cassette . . . \$9.95  
1763XVII

  
**TSE-HARDSIDE**  
6 South St. Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790



# ALL RAMS ARE NOT CREATED EQUAL

by Joseph V. Cesaitis

If you are the owner of a Radio Shack Model I S-80 disk system and have been observing any or all of the following symptoms you may be headed down the road of sleepless nights and a depleted checkbook.

1. Random rebooting of your system while in BASIC;
2. Bad loads from disk, particularly where large files are involved;
3. Keyboard lockup after entering BASIC from DOS; or
4. Unexplained out of memory or syntax error messages in Disk BASIC.

In my case I experienced all of these problems. I had read that the expansion interface (E/I) with the buffered cable (such as mine) had some design problems which could account for the strange behavior of my machine. I took the advice of a number of magazine articles in an effort toward solving these problems. My method was to begin with those solutions which had little or no associated cost and to work up from there.

First, I ran TEST1 and TEST2 as described in the TRSDOS manual. Neither test turned up any problems except a random reboot during one of ten executions of TEST2. I thought nothing of this reboot since TEST2 runs under Disk BASIC and this was the very problem I was trying to solve.

Second, I cleaned all printed circuit board edge connections with a pink eraser to make sure that I had no oxidation problems at my cable connections. This procedure did nothing to alleviate my problems.

Third, I interchanged each 16K RAM set between the CPU and the E/I, to see if a certain combination would work. After shuffling various RAM sets between the CPU and the E/I, I finally found a combination which didn't cause a rebooting or keyboard problem. My rejoicing was premature, however, because in less than two days the problem reappeared. This put me on the trail of suspecting the RAM chips themselves.

Fourth, I ran a lengthy RAM test (eight to ten hours) on each RAM set in hope of ferreting out the defective chip(s). Each RAM set passed this test with flying colors. The RAM trail

exhausted, I decided to call the Tandy hot line.

I spoke with a hardware specialist who offered the following possible causes for my problem:

1. A bad copy of TRSDOS;
2. Static electricity;
3. Power surges on AC line; or
4. Noise on adjacent traces in buffered cable.

One by one I went down the service technician's list of possible causes. I used Percom's Double DOS disk as my alternate copy of TRSDOS but observed no improvement. The Tandy service technician explained that I could be experiencing a static electricity problem which was particularly acute in the winter months due to a lack of humidity. He indicated that the S-80 operates best in an environment of at least 50% humidity.

Armed with my children's vaporizer and an hygrometer I raised the level of humidity in the room to 70%. There was still no improvement, so I decided that perhaps a \$50 investment in Radio Shack's Power Line Filter would do the trick. Unfortunately this addition failed as well, but as an aside, this \$50 purchase was not totally in vain because I can now turn on my entire system with the flip of a single switch. This was quite an improvement over plugging and unplugging eight plugs and turning on six different printer interface and cassette player have no power switches).

I decided to delay action on possible cause #4 until I tried replacing the two sets of 16K RAM chips in the E/I with entirely new chip sets. I visited my local computer store and with the agreement that I could return the chips if they didn't solve my problem, I installed two sets of Fujitsu 200 nanosecond access time, 16K RAM chips. Instead of improving, the problem got worse. Question marks, commas, and other extraneous characters were displayed on the screen when I entered Disk BASIC. I promptly returned the Fujitsu chips and moved on to the service technician's fourth suggestion.

I stopped by a Radio Shack store near where I work and asked about the modification for installing a twisted pair. The manager informed me that Radio Shack had a "campaign" on for the last eight months

to get buffered cable E/I owners to have this mod performed. I am writing this in late February, 1981, and I never saw any Radio Shack promotional material for this mod outside of the technician's suggestion — and only then after I brought it up. Nonetheless, I asked the manager if I should bring in both the CPU and the E/I and he stressed that only the E/I was necessary.

The next day I dropped off the E/I for the necessary work. The manager was not there and the sales personnel who were there acted as though I had dreamed up the twisted pair mod. After assuring them that this was a Radio Shack approved mod, one salesman called the local repair center to verify my claim. During his call we both discovered that the manager had been in error about needing just the E/I. Both the CPU and the E/I are needed for this mod. Annoyed, I returned the next day with both units. The following day I received a call from the Radio Shack salesman indicating that the twisted cable had been installed but did not fix the problem. He did indicate that installation of new RAM had fixed the problem. Hesitant at first to spend \$119 per 16K RAM set having just performed a similar replacement days earlier, I nonetheless decided to take the plunge.

I took both units home and observed the modification. The twisted pair is actually a gray cable of five wires interconnected between the CPU and the E/I with a DIN-type plug. After connecting all the cables I pushed the reset button on the back of the CPU to boot up. Nothing happened. I was able to get the machine to operate as a 48K Level II computer but I could not get the reset button to function with or without the E/I connected. Disgusted, I opened the CPU unit and performed a continuity check on the reset switch. The switch checked out fine but during my inspection I noticed a blob of solder across two foil patterns. I removed the solder and retried the reset button. Eureka! It worked. I quickly reconnected the E/I and the disk and tried the reset button again. The disk booted up with no problem. I next tried calling Disk BASIC and left the machine in this system for 45 minutes. I observed no random rebooting such as I had before. I have


had the machine back for one week now and it is working perfectly except for one minor flaw which I will mention later.

I opened up the E/I to see what type of RAM chips Radio Shack installed. I found 16 Motorola chips number 8041016. There was a second number on each chip which was not the same for each chip. The chips in sockets Z1 through Z8 had 7940 printed on them and the chips in sockets Z9 through Z16 either had a 8015 or 8016 stamped on the case. I suspect that these four-digit numbers are batch control numbers or something like that. I had read in one of the computer magazines that Motorola and NEC chips were the most reliable for proper E/I operation. I found this hard to believe until now. The chips which I had been using were from two different sources but each set had a one year guarantee, something I recommend you look for when purchasing any kind of hardware. One set was marked Singapore 4116-2DC, F8040 which I purchased from Godbout Electronics. I have been in contact with this company and already received my \$39 back plus the \$2 shipping charge. The other set of

RAM was Hitachi HM4716A-4N, Japan 9M1 which I purchased from California Digital. I haven't talked to them but they advertise an unconditional guarantee. I'll see. Both of these chip sets worked fine in the CPU except for the problem I mentioned earlier. With the Hitachi chips in the CPU the display would jitter when the machine was first turned on. This problem disappeared after two or three minutes of operation.

I hope that by sharing my experience with other S-80 owners, perhaps I have been able to shed some light on a particularly annoying problem. Also, I hope I have saved others some time and/or expense in resolving this type of problem.

**EDITOR'S NOTE:**

Mr. Cesaitis' experience is not an exceptional one. Memory chips are not created equal, the quality varies with the manufacturer. Some chips, due to impurities in the materials, cause "soft errors" or sporadic changes of memory location. 

## Computer Games!

### How can we tell you about 400 computer games in one advertisement?

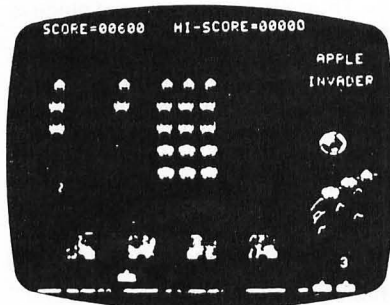
We've got the world's largest line of computer games. Over 400 in all. They're on cassette and disk for eight popular personal computers: Atari, Apple, TI 99/4, PET, TRS-80, Sorcerer, Sol and CP/M.

From A to Z, Action Games to Z-Chess II, we've got loads of best-sellers including "Super Invader" for the Apple, a complete line of six Adventure games, Backgammon, Milestones and Cycle Jump.

Not only that, we publish the best-selling books, *Basic Computer Games* and *More Basic Computer Games* with over 500,000 copies in print.

We've also got a nifty board game, *Computer Rage*, sets of three binary dice, acrobatic toy robots, T-shirts and lots of other goodies.

You'll find comprehensive descriptions of all of our software, books, games and peripherals in our huge 48-page catalog. It's unique in the small computer field. For your free copy, write or call us today or circle our number on the reader service card.



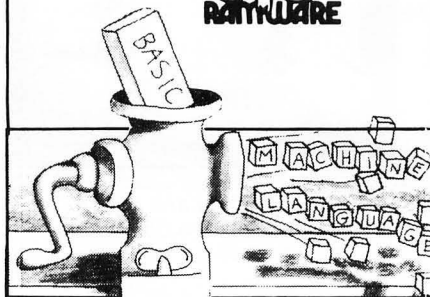
Super Invader features superb high-resolution graphics, nail-biting tension and hilarious antics by the moon creatures.

## creative computing

39 East Hanover Ave.  
Morris Plains, NJ 07950  
Toll-free 800-631-8112  
In NJ 201-540-0445

## The Lazy Man's Shortcut to Machine Language

from



# Tiny Comp

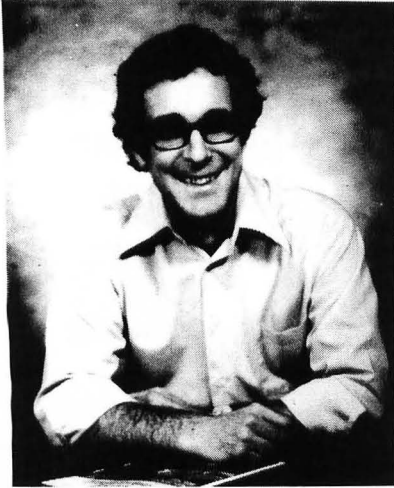
A BASIC Compiler in BASIC!  
Run your source program in BASIC, compile it into FAST Z-80 Code and execute the compiled version — all without reloading. 26 integer variables, GOTO, GOSUB, END, REM, RND, LET, +, \*, /, IF, THEN, =, INKEY\$, CLS, PRINT@, CHR\$, PEEK, POKE. Compiled programs may be saved via TAPEDISK.

Supplied with game program, "3-D TIC TAC TOE", which uses all of the TINY COMP statement set and is ready to compile.

Manual includes several sample programs as well as thorough documentation of the Compiler for those who like to know "how things work" and for those who might even wish to EXPAND on TINY COMP's capabilities.

16K, S-80 Cassette #26-TYTT . . . . \$19.95  
32K, S-80, Disk #26-TYND . . . . . \$24.95

  
**TSE-HARDSIDE**  
6 South St. Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790



David Ahl, Founder and  
Publisher of *Creative Computing*

# Creative Computing

***"The beat covered by Creative Computing is one of the most important, explosive and fast-changing."—Alvin Toffler***

You might think the term "creative computing" is a contradiction. How can something as precise and logical as electronic computing possibly be creative? We think it can be. Consider the way computers are being used to create special effects in movies—image generation, coloring and computer-driven cameras and props. Or an electronic "sketchpad" for your home computer that adds animation, coloring and shading at your direction. How about a computer simulation of an invasion of killer bees with you trying to find a way of keeping them under control?

## Beyond Our Dreams

Computers are not creative per se. But the way in which they are used can be highly creative and imaginative. Five years ago when *Creative Computing* magazine first billed itself as "The number 1 magazine of computer applications and software," we had no idea how far that idea would take us. Today, these applications are becoming so broad, so all-encompassing that the computer field will soon include virtually everything!

In light of this generality, we take "application" to mean whatever can be done with computers, *ought* to be done with computers or *might* be done with computers. That is the meat of *Creative Computing*.

Alvin Toffler, author of *Future Shock* and *The Third Wave* says, "I read *Creative Computing* not only for information about how to make the most of my own equipment but to keep an eye on how the whole field is emerging.

*Creative Computing*, the company as well as the magazine, is uniquely light-hearted but also seriously interested in all aspects of computing. Ours is the magazine of software, graphics, games and simulations for beginners and relaxing professionals. We try to present the new and important ideas of the field in a way that a 14-year old or a Cobol programmer can under-

stand them. Things like text editing, social simulations, control of household devices, animation and graphics, and communications networks.

## Understandable Yet Challenging

As the premier magazine for beginners, it is our solemn responsibility to make what we publish comprehensible to the newcomer. That does not mean easy; our readers like to be challenged. It means providing the reader who has no preparation with every possible means to seize the subject matter and make it his own.

However, we don't want the experts in our audience to be bored. So we try to publish articles of interest to beginners and experts at the same time. Ideally, we would like every piece to have instructional or informative content—and some depth—even when communicated humorously or playfully. Thus, our favorite kind of piece is accessible to the beginner, theoretically non-trivial, interesting on more than one level, and perhaps even humorous.

David Gerrold of *Star Trek* fame says, "*Creative Computing* with its unpretentious, down-to-earth lucidity encourages the computer user to have fun. *Creative Computing* makes it possible for me to learn basic programming skills and use the computer better than any other source.

## Hard-hitting Evaluations

At *Creative Computing* we obtain new computer systems, peripherals, and software as soon as they are announced. We put them through their paces in our Software Development Center and also in the environment for which they are intended—home, business, laboratory, or school.

Our evaluations are unbiased and accurate. We compared word processing printers and found two losers among highly promoted makes. Conversely, we found one computer had far more than its advertised capability. Of 16 educational packages,

only seven offered solid learning value.

When we say unbiased reviews we mean it. More than once, our honesty has cost us an advertiser—temporarily. But we feel that our first obligation is to our readers and that editorial excellence and integrity are our highest goals.

Karl Zinn at the University of Michigan feels we are meeting these goals when he writes, "*Creative Computing* consistently provides value in articles, product reviews and systems comparisons... in a magazine that is fun to read."

## Order Today

To order your subscription to *Creative Computing*, send \$20 for one year (12 issues), \$37 for two years (24 issues) or \$53 for three years (36 issues). If you prefer, call our toll-free number, **800-631-8112** (in NJ 201-540-0445) to put your subscription on your MasterCard, Visa or American Express card. Canadian and other foreign surface subscriptions are \$29 per year, and must be prepaid. We guarantee that you will be completely satisfied or we will refund the entire amount of your subscription.

Join over 80,000 subscribers like Ann Lewin, Director of the Capital Children's Museum who says, "I am very much impressed with *Creative Computing*. It is helping to demystify the computer. Its articles are helpful, humorous and humane. The world needs *Creative Computing*."

# Creative Computing

Attn: Elyse  
P.O. Box 789-M  
Morristown, NJ 07960  
Toll-free **800-631-8112**  
(In NJ 201-540-0445)

The story behind the two best selling computer games books in the world.

# Computer Games

by David H. Ahl

Everybody likes games. Children like tic tac toe. Gamblers like blackjack. Trekkies like Star Trek. Almost everyone has a favorite game or two.

## It Started in 1971

Ten years ago when I was at Digital Equipment Corp. (DEC), we wanted a painless way to show reluctant educators that computers weren't scary or difficult to use. Games and simulations seemed like a good method.

So I put out a call to all our customers to send us their best computer games. The response was overwhelming. I got 21 versions of blackjack, 15 of nim and 12 of battleship.

From this enormous outpouring I selected the 90 best games and added 11 that I had written myself for a total of 101. I edited these into a book called 101 Basic Computer Games which was published by DEC. It still is.

When I left DEC in 1974 I asked for the rights to print the book independently. They agreed as long as the name was changed.

Contents of *Basic Computer Games* (right) and *More Basic Computer Games* (below).

Artillery-3	Life Expectancy
Baccarat	Lissajous
Bible Quiz	Magic Square
Big 6	Man-Eating Rabbit
Binary	Maneuvers
Blackbox	Mastermind
Bobstones	Masterbagels
Bocce	Matpuzzle
Boga II	Maze
Bumbrun	Millionaire
Bridge-It	Minotaur
Camel	Motorcycle Jump
Chase	Nomad
Chuck-A-Luck	Not One
Close Encounters	Obstacle
Column	Octrix
Concentration	Pasart
Condot	Pasart 2
Convoy	Pinball
Corral	Rabbit Chase
Countdown	Roadrace
Cup	Rotate
Dealer's Choice	Safe
Deepspace	Scales
Defuse	Schmoo
Dodgem	Seabattle
Doors	Seawar
Drag	Shoot
Dr. Z	Smash
Eliza	Strike 9
Father	Tennis
Flip	Tickertape
Four In A Row	TV Plot
Geowar	Twonky
Grand Prix	Two-to-Ten
Guess-It	UFO
ICBM	Under & Over
Inkblot	Van Gam
Joust	Warfish
Jumping Balls	Word Search Puzzle
Keno	Wumpus 1
L Game	Wumpus 2

Introduction	Hi-Lo
The Basic Language	High I-Q
Conversion to Other Basics	Hockey
Acey Ducey	Horseace
Amazing	Hurkle
Animal	Kinema
Awari	King
Bagels	Letter
Banner	Life
Basketball	Life For Two
Batnum	Literature Quiz
Battle	Love
Blackjack	Lunar LEM Rocket
Bombardment	Master Mind
Bombs Away	Math Dice
Bounce	Mugwump
Bowling	Name
Boxing	Nicomachus
Bug	Nim
Bullfight	Number
Bullseye	One Check
Bunny	Orbit
Buzzword	Pizza
Calendar	Poetry
Change	Poker
Checkers	Queen
Chemist	Reverse
Chief	Rock, Scissors, Paper
Chomp	Roulette
Civil War	Russian Roulette
Combat	Salvo
Craps	Sine Wave
Cube	Slalom
Depth Charge	Slots
Diamond	Splat
Dice	Stars
Digits	Stock Market
Even Wins	Super Star Trek
Flip Flop	Synonym
Football	Target
Fur Trader	3-D Plot
Golf	3-D Tic-Tac-Toe
Gomoko	Tic Tac toe
Guess	Tower
Gunner	Train
Hammurabi	Trap
Hangman	23 Matches
Hello	War
Hexapawn	Weekday
	Word

## Converted to Microsoft Basic

The games in the original book were in many different dialects of Basic. So Steve North and I converted all the games to standard Microsoft Basic, expanded the descriptions and published the book under the new name Basic Computer Games.

Over the next three years, people sent in improved versions of many of the games along with scores of new ones. So in 1979, we totally revised and corrected Basic Computer Games and published a completely new companion volume of 84 additional games called More Basic Computer Games. This edition is available in both Microsoft Basic and TRS-80 Basic for owners of the TRS-80 computer.

Today Basic Computer Games is in its fifth printing and More Basic Computer Games is in its second. Combined sales are over one half million copies making them the best selling pair of books in recreational computing by a wide margin. There are many imitators, but all offer a fraction of the number of games and cost far more.

The games in these books include classic board games like checkers. They include challenging simulation games like Camel (get across the desert on your camel) and Super Star Trek. There are number games like Guess My Number, Stars and Battle of Numbers. You'll find gambling games like blackjack, keno, and poker. All told there are 185 different games in these two books.

Whether you're just getting started with computers or a proficient programmer, you'll find something of interest. You'll find 15-line games and 400-line games and everything in between.

The value offered by these books is outstanding. Every other publisher has raised the price of their books yet these sell for the same price as they did in 1974.

## Moneyback Guarantee

Examine one or both of these books and key some games into your computer. If you're not completely satisfied we'll refund the full purchase price plus your return postage.

Basic Computer Games costs only \$7.50 and More Basic Computer Games just \$7.95 for either the Microsoft or TRS-80 edition (please specify your choice on your order). Both books together are \$15. Send payment plus \$2.00 shipping and handling to Creative Computing Press, Morris Plains, NJ 07950. Visa, MasterCard and American Express orders should include card number and expiration date. Charge card orders may also be called in toll-free to 800-631-8112 (in NJ 201-540-0445).

Order today to turn your computer into the best game player on the block.

# creative computing

Morris Plains, NJ 07950  
Toll-free 800-631-8112  
(In NJ 201-540-0445)



# BATS

by John Baker, D.V.M.

**"Bats" is an S-80 program requiring at least 16K of RAM.**

## THE ELECTRIC VAMPIRE NIGHTMARE GAME

This elaborate and challenging game has been kicking around some mainframe business computer systems for several years. The version presented here allows us S-80 micro users to get in on some of the fun.

The object of the game is survival. You are placed into a field surrounded by a maze of electrified fences. In the field with you are some not-so-friendly bats who will mindlessly pursue you until they succeed in biting you or forcing you to stumble into a fence. Your only hope is to outwit them and lure all of them into getting zapped by the fences first.

You move around by using this ar-

ray of ten keys which are under the control of the first three fingers of your right hand:

```
U I O
H J K L
, . /
```

The H is the "hyperspace" key (use it cautiously!), and the other nine correspond to their relative physical positions on the keyboard: U is up-left, I is up, O is up-right, etc., with K being the "stay here" position.

The game becomes more addicting as you play it and get better at it. You can consider yourself pretty good if you survive more than half the time. And if you start getting too relaxed, just throw in a few more bats....

## VARIABLES

A(x,y): Array containing field locations of bats, fences, and you.

B: Number of bats.

F: Number of random fence sections.

G: Number of games played.

H: Number of "hyperspaces" used.

K: Number of bats killed.

M: Number of moves used.

Q\$: Player's move.

X, Y: Horizontal and vertical coordinates on field.



Initialization and instructions.

```
15 RANDOM
20 CLEAR 1000
25 DIM A(20,12):DIM X(50):DIM Y(50)
60 F=38:CLS
70 F$=CHR$(153)+CHR$(191)+CHR$(166)
100 PRINT "BATS >V< THE ELECTRIC VAMPIRE NIGHTMARE!":
PRINT
101 INPUT "DO YOU NEED INSTRUCTIONS (Y/N)";Q$
102 PRINT:INPUT "HOW MANY BATS DO YOU CHALLENGE";B
106 IF LEFT$(Q$,1)="Y" THEN GOTO 110 ELSE PRINT:PRINT:PRINT:PRIN
T " .....PLEASE STAND BY WHILE I LOCATE A BAT CAVE.....":GOTO
1000
110 CLS:PRINT "YOU AWAKE TO FIND YOURSELF TRAPPED IN A FIELD OF
ELECTRIFIED"
120 PRINT "WIRE FENCES WITH";B;"BATS. THIS IS NO ROD SERLING TW
ILIGHT"
130 PRINT "ZONE TELEVISION PROGRAM...THIS PROGRAM IS REAL!"
131 PRINT:PRINT "THE BATS CANNOT SEE THE FENCES AND WILL BE DEST
ROYED IF"
133 PRINT "THEY FLY INTO ONE OR INTO EACH OTHER."
134 PRINT:PRINT "THE BATS WILL MINDLESSLY MOVE TOWARD YOUR POSIT
ION"
135 PRINT "AND IF ONE BITES YOU, OR YOU BLUNDER INTO A FENCE - Y
OU LOSE!"
136 PRINT "YOUR ONLY HOPE IS TO LURE THE BATS INTO THE FENCES."
137 PRINT:PRINT "YOU WIN ONLY WHEN ALL THE BATS ARE DESTROYED."
138 PRINT:LINE INPUT "(DEPRESS 'ENTER' TO CONTINUE)";Q$
139 CLS
140 A$="+-----+-----+-----+":PRINT@6,A$;:PRINT@192,"+-----";A$;:
PRINT@384,"+-----";A$;:PRINT@582,A$;
```

```
160 PRINT@95,"YOU CAN MOVE IN ANY DIRECTION";:PRINT@159,"BY PRES
SING THESE KEYS:";
162 PRINT@297,"U I O";:PRINT@357,"H J K L";:PRINT@425,
"M , .";
164 PRINT@543,"WHICH CORRESPOND TO THE";:PRINT@607,"DIRECTIONS S
HOWN AT LEFT.";
170 PRINT @70,"!LEFT ! UP !RIGHT!";:PRINT@134,"! UP ! ! UP
!";
180 PRINT@256,"!HYPER!LEFT !WAIT !RIGHT!";:PRINT@320,"!SPACE!
!HERE ! !";
190 PRINT @ 454,"!LEFT ! !RIGHT!";:PRINT@518,"!DOWN !DOWN !D
OWN !";
205 PRINT @ 704,STRING$(64,37);
210 PRINT@770,">V< - A BAT "F$;" - A FENCE YOU -
A VICTIM";
```

Set up outer fences.

```
1000 FOR I=0 TO 12:FOR J=0 TO 20
1010 A(J,I)=0:A(0,I)=1:A(20,I)=1:NEXT J:NEXT I
1020 FOR I=0 TO 20:A(I,0)=1:A(I,12)=1:NEXT I
```

Set up random inner fences.

```
1030 FOR I=1 TO F:A(RND(19),RND(11))=1:NEXT I
```

Place bats in random locations.

```
1090 FOR I=1 TO B
1100 X(I)=RND(19):Y(I)=RND(11):IF A(X(I),Y(I))<>0 THEN GOTO 1100
ELSE A(X(I),Y(I))=2
1200 NEXT I
```

Place victim in random location.



```

1300 IF A(X1,Y1)=0 THEN GOTO 1305 ELSE X1=RND(19)
1301 Y1=RND(11):GOTO 1300
1305 A(X1,Y1)=3

```

Reset counters.

```

1310 H=0:K=0:M=0:PRINT @896,CHR$(30):PRINT @896," ";

```

Beginning of game.

```

1320 INPUT "READY TO PLAY (Y/N)";Q$
1340 IF Q$="N" THEN GOTO 9000 ELSE CLS

```

Print game field.

```

1400 FOR J=0 TO 12:FOR I=0 TO 20
1450 ON A(I,J) GOTO 1510,1520,1530
1500 GOTO 1600
1510 PRINT @(I*3)+(J*64),F$;
1515 GOTO 1600
1520 PRINT @(I*3)+(J*64),">V<";
1525 GOTO 1600
1530 PRINT @(I*3)+(J*64),"YOU";
1600 NEXT I:NEXT J

```

Timed input routine.

```

1800 PRINT @896,CHR$(30);
2000 PRINT @896,"YOUR MOVE: ";
2002 FOR TM=1 TO 100
2004 Q$=INKEY$
2006 IF Q$="" THEN NEXT TM
2008 IF Q$="" THEN Q$="K"
2010 Q=ASC(Q$): PRINT @896,CHR$(30);

```

Clear last position from memory and display.

```

2020 A(X1,Y1)=0:PRINT @(X1*3)+(Y1*64)," ";CHR$(15);

```

Hyperspace jumpoff point.

```

2030 IF Q=72 THEN H=H+1: PRINT@896,"<<HYPERSPACE>>";X1=RND(19):
Y1=RND(11): GOTO 2200

```

Determine new position based on player's input.

```

2032 IF Q=77 THEN X1=X1-1:Y1=Y1+1:GOTO 2200
2034 IF Q=44 THEN Y1=Y1+1:GOTO 2200
2036 IF Q=46 THEN X1=X1+1:Y1=Y1+1:GOTO 2200
2038 IF Q=74 THEN X1=X1-1: GOTO 2200
2040 IF Q=76 THEN X1=X1+1: GOTO 2200
2042 IF Q=85 THEN X1=X1-1: Y1=Y1-1: GOTO 2200
2044 IF Q=73 THEN Y1=Y1-1:GOTO 2200
2046 IF Q=79 THEN X1=X1+1: Y1=Y1-1: GOTO 2200
2080 A(X1,Y1)=0

```

Update counters and check for collisions.

```

2200 M=M+1:IF A(X1,Y1)=0 THEN GOTO 2300
2210 ON A(X1,Y1) GOTO 2310,2320
2250 GOTO 2320
2300 PRINT @(X1*3)+(Y1*64),"YOU";A(X1,Y1)=3:GOTO 2400
2310 PRINT @(X1*3)+(Y1*64),"ZAP";:GOTO 4000
2320 PRINT @(X1*3)+(Y1*64),"BAT";:GOTO 4000
2400 J=0
2410 FOR I=1 TO B:IF X(I)=0 THEN GOTO 2700
2500 PRINT @(X(I)*3)+(Y(I)*64)," ";A(X(I),Y(I))=0
2510 X(I)=X(I)+SGN(X1-X(I)):Y(I)=Y(I)+SGN(Y1-Y(I))

```

```

2520 ON A(X(I),Y(I)) GOTO 2620,2620,2650
2530 IF A(X(I),Y(I)) <> 0 THEN GOTO 2650
2600 PRINT @(X(I)*3)+(Y(I)*64),">V<";A(X(I),Y(I))=2:J=J+1:GOTO
2700
2620 X(I)=0:K=K+1:GOTO 2700
2650 PRINT @(X(I)*3)+(Y(I)*64),"BIT";:GOTO 4000
2700 NEXT I

```

Recycle to next move.

```

2800 IF J<>0 THEN GOTO 1800

```

Game won.

```

3000 PRINT @896,CHR$(30):PRINT @896,"YOU WIN";
3050 G=6+1:W=W+1:IF W*2>6 THEN B=B+1
3100 IF H=0 AND M<=K THEN B=B+1
3200 IF M*2<K THEN B=B+1
3300 GOTO 5000

```



Game lost.

```

4000 PRINT @896,CHR$(30):PRINT @896,"YOU LOSE";
4100 G=6+1:IF W*2<6 THEN B=B-1
4500 IF K*2>B AND M<K AND H<2 THEN B=B+1 ELSE IF B=0 THEN GOTO 9
100

```

Print scoreboard.

```

5000 IF M>1 THEN PRINT " IN";M;"MOVES.":GOTO 5005
5001 PRINT " IN";M;"MOVE."
5005 PRINT@960,"NO. BATS:";B;
5010 L=6-W
5110 PRINT "      GAMES:";G;"      WON/LOST:";W;"/";L;

```

"Computer fatigue" check.

```

6000 IF G>=100 THEN GOTO 9100 ELSE GOTO 1000

```

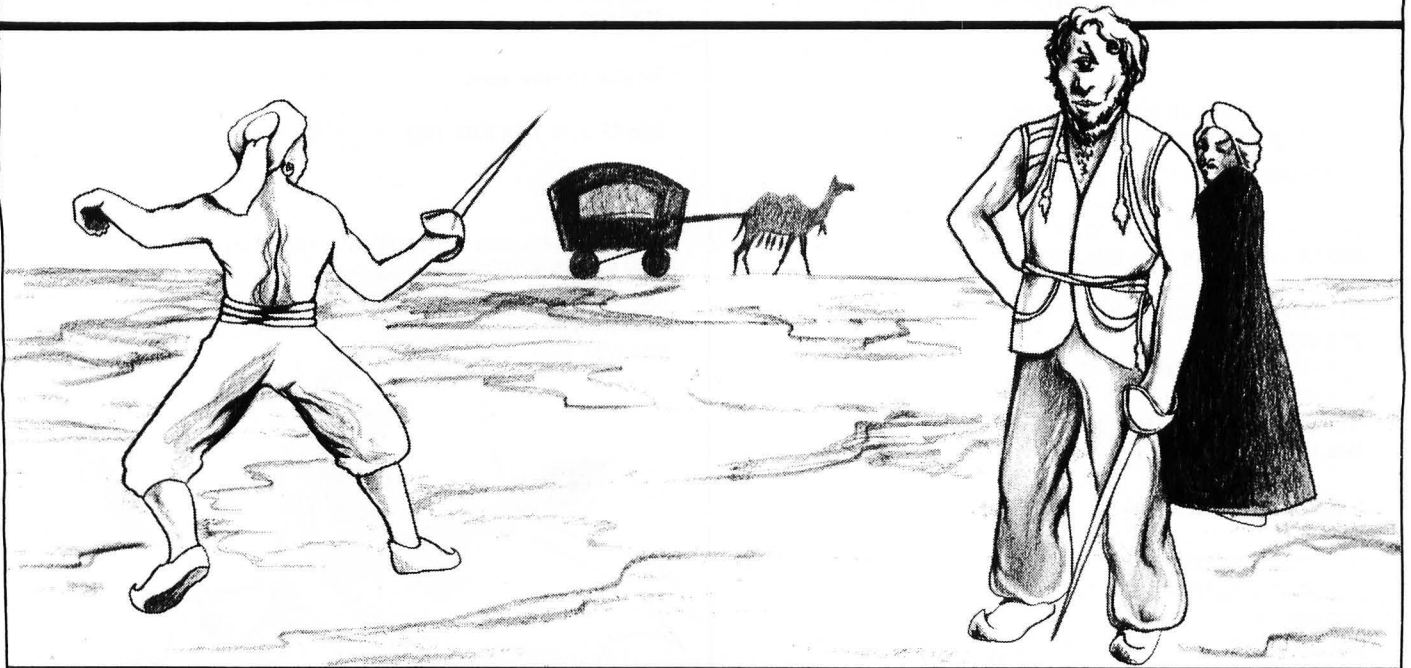
Exit routine.

```

9000 CLS:PRINT TAB(11) "TOO BAD YOU COULDN'T WORK UP ENOUGH NERV
E!":PRINT
9010 PRINT "IF YOU EVER GET SOME GUTS, JUST COME BACK AND I WILL
"
9015 PRINT "TRY TO FIND ANOTHER CAVE."
9017 PRINT @896;
9020 END
9100 CLS:PRINT "THAT IS QUITE ENOUGH!":END

```

# ADVENTURE OF THE MONTH CLUB



## ARABIAN ADVENTURE

### JUNE ADVENTURE OF THE MONTH

As Sinbad, the mightiest sailor in ancient Arabia, it is your mission to rescue Princess Jasmine from the clutches of Rex, the Wizard of Darkness. Your search takes you across the Seven Seas to the deadly Cyclops Mountain and you'll do battle with frightening skeletons, a one-eyed beast, a hairy tarantula and more monsters who try to thwart your noble pursuit.

**You can get great Adventures by...**

**Paying up to \$35 each (and give up lunch for a month)**

**Or by typing them in yourself (and eat at the computer for a week)**

**OR you can sign up to join SoftSide's...ADVENTURE OF THE MONTH CLUB.**

Everybody likes Adventures — they're challenging and entertaining every time you play. But too often, pre-programmed cassettes and disks cost upwards of \$35, a price the manufacturer must charge to defray promotional and packaging costs.

On the other hand, you can enter Adventures yourself, but when you do, you type away all the surprises. As a result, the game loses some of its challenge.

At **SoftSide**, we've found a way to beat the high cost of Adventuring without having to miss out on any of the fun. We're offering no-frill Adventures — high quality Adventures — on cassette or disk at an almost unheard-of price: **\$5 for cassette, \$8 on disk.**

We save you money by only advertising this offer to **SoftSide** readers (you won't see us anywhere else) and by foregoing fancy packaging and documentation — you'll get the software and only the software, but we believe it's as good as the \$50 packages.

You'll save even more by joining the Adventure of the Month Club.

Here's how it works: **SoftSide's** editorial department will select an original BASIC language Adventure each month and make it available to you on a subscription basis:

6 months on cassette: just \$27

6 months on disk: just \$45

Every month we'll tell you about the Adventure you'll be getting in **SoftSide** Magazine. To order, use the convenient order card in this issue — fill it out and send, with payment to:

**Adventure of the Month Club  
Department 681  
6 South Street  
Milford, NH 03055**

# The Atari\* Connection.

ATARI CX70  
LIGHT PEN  
\$74.95

ATARI 800  
(48K) - \$999.00

ATARI 850  
INTERFACE  
\$179.00

ATARI 830  
MODEM  
\$179.00

PADDLE  
CONTROLLERS  
\$19.95 (pr.)

ATARI 810 DISK DRIVE \$499.00

ADDITIONAL  
RS232 PORTS

JOYSTICK  
CONTROLLERS  
\$19.95 (pr.)

ATARI 410  
RECORDER \$89.00

ATARI 825 PRINTER \$789.00



## Hardware

ATARI 400 Computer System, 8K RAM . . .	\$439.00 (#36-400)
ATARI 400 Computer System, 16K RAM . .	\$499.00 (#36-401)
ATARI 400 Computer System, 32K RAM . .	\$599.00 (#36-402)
ATARI 800 Computer System, 16K RAM . .	\$829.00 (#36-800)
ATARI 800 Computer System, 32K RAM . .	\$949.00 (#36-801)
ATARI 800 Computer System, 48K RAM . .	\$999.00 (#36-802)
ATARI 410 Program Recorder . . . . .	\$69.00 (#36-803)
ATARI 810 Disk Drive . . . . .	\$499.00 (#36-810)
ATARI 822 Thermal Printer . . . . .	\$389.00 (#36-820)
ATARI 825 Printer (80-col) . . . . .	\$769.00 (#36-825)
ATARI 830 Acoustic Modem . . . . .	\$179.00 (#36-830)
ATARI 850 Interface . . . . .	\$179.00 (#36-850)
16K RAM Module for the ATARI . . . . .	\$109.00 (#36-854)
32K RAM Module for the ATARI . . . . .	\$179.00 (#36-855)
MACROTRONICS Printer Interface (36-pin)	\$69.95 (#36-936)
MACROTRONICS Printer Interface (40-pin)	\$69.95 (#36-940)
ATARI Joystick Controllers . . . . .	\$19.95 (#36-3005)
ATARI Paddle Controllers . . . . .	\$19.95 (#36-3004)
ATARI CX-70 Light Pen . . . . .	\$74.95 (#36-70)
Dust Cover for ATARI 400 . . . . .	\$7.95 (#16-40)
Dust Cover for ATARI 800 . . . . .	\$7.95 (#16-03)

## ROM programs

Basketball . . . . .	\$34.95 (#36-BASK)
Chess . . . . .	\$34.95 (#36-CHS)
Editor/Assembler . . . . .	\$49.95 (#36-ASE)
Music Composer . . . . .	\$49.95 (#36-MUSE)
Star Raiders . . . . .	\$49.95 (#36-STRDS)
Super Breakout . . . . .	\$34.95 (#36-SUPB)
Tele-Link . . . . .	\$24.95 (#36-TEL)
3D Tic-Tac-Toe . . . . .	\$34.95 (#36-3TTT)
Video Easel . . . . .	\$34.95 (#36-VIDEO)

## Software <sup>on</sup> Disk


VISICALC from Personal Software . . . . .	\$199.95 (#36-VICL)
MAILING LIST . . . . .	\$24.95 (#36-279002D)

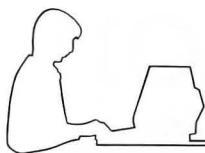
## Software <sup>on</sup> Cassette

Star Trek 3.5 . . . . .	\$14.95 (#36-200025T)
Deflection/Simon Says . . . . .	\$9.95 (#36-200078T)
Mountain Shoot . . . . .	\$9.95 (#36-200079T)
Angle Worm/Crolon Diversion . . . . .	\$9.95 (#36-200092T)
Fleet . . . . .	\$6.95 (#36-27701BC)
Connect Four . . . . .	\$6.95 (#36-277020C)
Letters . . . . .	\$6.95 (#36-277022C)
Reverse . . . . .	\$6.95 (#36-277023C)
Zap . . . . .	\$6.95 (#36-277024C)
Lander . . . . .	\$6.95 (#36-277019C)
Chase . . . . .	\$6.95 (#36-277017C)
Computer Acquire . . . . .	\$20.00 (#36-237002C)
Conflict 2500 . . . . .	\$15.00 (#36-237001C)
Lords Of Karma . . . . .	\$20.00 (#36-237001C)
Biorhythms . . . . .	\$14.95 (#36-BIOR)
Blackjack . . . . .	\$14.95 (#36-BLACK)
Hangman . . . . .	\$14.95 (#36-HANG)
Kingdom . . . . .	\$14.95 (#36-KING)
3-Dimensional Graphics Package . . . . .	\$29.95 (#36-3D-G)

TERMS: Prices and specifications are subject to change. HARDWARE accepts VISA & MASTERCARD. Certified checks and Money Orders. Personal checks accepted (takes 3 weeks to clear). HARDWARE pays all shipping charges (within the 48 states) on all PREPAID orders over \$100.00. On all orders under \$100 a \$2.50 handling charge must be added. COD orders accepted (orders over \$250 require 25% deposit), there is a \$5.00 handling charge. UPS Blue Label and Air Freight available at extra cost.



A Warner Communications Company 



# DIVIDE AND CONQUER

by Phillip C. Soine

**“Divide and Conquer” is an S-80 program which requires at least 16K RAM.**

“Divide and Conquer” is written in two separate parts, a sound routine and the main program. The sound routine must be executed to play the game, but once it is executed it need not be executed again as long no disks are booted or the machine is shut off. If you have disk, store the two programs on the same disk, calling the sound program “DIVIDE” and the main program “CONQUER”. To run the program, run the program DIVIDE which will then run CONQUER. If you have cassette, place the sound program before the main program on the cassette. The sound program will then load the main program when it is done.

If you do not want sound, you may simply enter the main program, adding the following line:

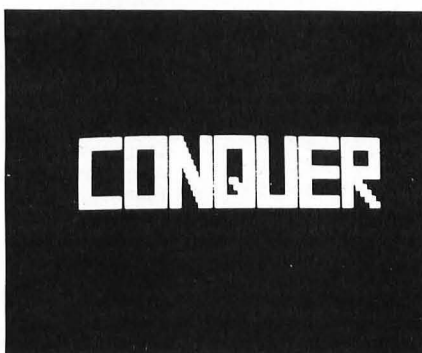
```
5 POKE 16782,201
```

The sound routine uses the NAME function instead of the USER(0) statement to make the program compatible with both Level II and Disk BASIC. The line above for those of you without audio amplifiers simply cancels the NAME routine from functioning, BASIC will simply ignore any such statements encountered.

When designing this program, the author ran into the problem of “garbage collecting”, which caused the computer to pause for as much as 30 seconds while it cleared out string space. A friend, who is credited in the program, suggested to the author that he use the variable pointer function so that changes in strings could be poked directly into the strings, thus avoiding the stalling problem. This is what the blank strings at the beginning of the program are for. They are the strings that store the four complete screens, one for each player, and they are continuously changing. If you break the program while a game is in progress and list it, whatever was on the screen at that time will show up in these strings.

In “Divide and Conquer” you have been waiting just outside the orbit of the Moon for hours, but somehow you know something is

about to happen. Soon waves of aliens will be bearing down on you and Earth is depending on you to drive them off. These aliens are mutants from the now destroyed planet of Krypton (yes, that’s where Superman was from). They have lived in space for years and are now looking to take over Earth. They don’t need spaceships or life-support systems and they propel themselves through space using an advanced levitation technique. They are nearly indestructible so they don’t bother with weapons, they will try to destroy you by ramming you.



Your weapons are armed with the only thing known to kill the aliens, kryptonite (what else?). Hitting an alien may only make matters worse for you because a characteristic of their mutation is that they divide up into separate aliens when hit. But, they can only divide up so much — eventually you will get rid of them.

The battle will be recreated on your ship’s computer screen. This allows you to view yourself in relation to the aliens. Your ship, if visible, will always be shown at the top of the screen with the aliens coming up at you. You use the “less than” and “greater than” keys to move your ship and the space bar to fire.

Just because you are destroyed, it doesn’t mean it’s all over with. The computer automatically teleports you back to Earth when your ship is being destroyed. To begin with you are allotted three ships. You are allowed more only if you do well enough. (Earth doesn’t want to waste expensive ships on inferior pilots.) The computer judges how well you do through the following scoring system:

Alien type	Points
*****	10
####	25
+ +	50
-	100

For each 10,000 points you earn, you get an extra ship. Your score is shown in the upper right corner and the number of ships left in the upper left.

The aliens get angry and speed increases as you hit smaller aliens. As a result their speed will frequently change as you hit the different kinds of aliens.

The first wave of aliens consists of four large aliens. If you can get rid of them, another wave of aliens will come, with four additional large ones. The waves will continue to increase by four up to a maximum of sixteen large aliens.

If the aliens don’t hit you, then, when they go off the top of the screen, they appear again at the bottom. You are allowed a certain number of these “passes” before there is an invasion. (How many depends on the level you chose and how many large aliens there were in the wave.) If there is an invasion your ship isn’t destroyed but you do receive penalty points. The total penalty points are determined by the kind and the amount of aliens that invade. The times at which each kind of alien will invade and the penalty points for each one that invades is as follows:

Alien type	Fraction of total passes allowed before invasion	Penalty points each
*****	1/4	1,000
####	1/2	500
+ +	3/4	250
-	1	100

You will receive a warning when you are three passes away from an invasion. An arrow will appear next to your score and you will hear a high pitched tone.

If, at any time, you have a negative score, the aliens take over Earth. But if, on the other hand, you receive a score of 100,000 or more then you have single-handedly defeated the aliens. Otherwise your contribution to defeating the aliens is proportionate to your score.

The skill level ranges from 1 to 20, (low to high) and controls, as already explained, the number of passes allowed by the aliens before they invade. There are also four versions which are made up of combinations of the following two factors:

1. You are able to refire at any time or you must wait until the missile hits something or goes off the screen.
2. Whether your ship is visible or not.

These factors are arranged into the four versions as follows:

Version	Ship visible	Refire anytime
1	Yes	Yes
2	Yes	No
3	No	Yes
4	No	No

One to four players can play at one time.

Now, if you're sure you have read these instructions carefully and thoroughly then you are ready to attempt to "Divide and Conquer".

## VARIABLES

A(3,16): Storage of each person's screen.  
 CT(3,3): For each person and for each type of alien it stores the total hit or invaded.  
 LC(3,16): POKE locations for each line of each person's screen.  
 V(3,16): Location of large alien for each line of each person's screen.  
 BH(3): Points needed for next bonus ship, each person.  
 D(0)-D(2): Ship printout.  
 D(3)-D(8): Destruction of ship.  
 GT(3): Each person's score.  
 H(3): Ships left for each person.  
 IN(7): Separate letters for invasion graphics.  
 IV(3): Point of invasion, each person.  
 L(3): Next line to print for each person.  
 N(3): Name of each person.  
 RF(3): Refire capability of each person.  
 SA(3): Stores first invasion time for each person, which is transferred to IV(3).  
 SK(3): Skill level for each person.

T(3): Total aliens in the wave, for each person.  
 TI(3): Count to invasion, for each person.  
 VB(3): Whether or not each person's ship is visible.  
 VN(3): Version, for each player.  
 D: For firing shot.  
 DR: Prints up arrow if close to an invasion.  
 F: Print location of the shot.  
 FR: Distance shot is away from the ship.  
 IP: Points penalized for an invasion.  
 I, J: Loop variables.  
 LK: Stores which alien you hit by a PEEK to the screen.  
 LV: Line of screen at which the alien was hit.  
 P: Current player number.  
 Q: When U reaches this point, the aliens advance.  
 R: Stores whether the hit was detected at the current point of the shot or one space ahead of it.  
 SP: Space across at which the hit occurs.  
 T: Location of sound subroutine.  
 TP: Total players.  
 U: Count that decides when the aliens advance.  
 X: Ship's position.  
 Y: For proper trailing blank, if needed.  
 Y1: Modifies ship print position.

```

50000 ' MACHINE LANGUAGE LOADER
50010 POKE16553,255
50020 TM=PEEK(16561)+256*PEEK(16562)
50030 N=16783:ML=TM+9:GOSUB50210
50040 GOSUB50240
50050 CLS:PRINT#384,"MEMORY SIZE MUST BE SET TO AT LEAST 125 BYTES LESS THAN MAX"
50060 PRINT#E.G. 32642 FOR 16K. IF IT'S NOT, POWER OFF AND START AGAIN!!"
50070 PRINT#593,"PRESS ENTER TO CONTINUE!!!";
50080 A$=INKEY$:IFA$=""THEN50080
50090 CLS:PRINTCHR$(23):PRINT#454,"THANKS TO CAP ELECTRONICS"
50100 PRINTTAB(3)"FOR THE SOUND SUBROUTINE"
50110 READA$:IFA$<"ML$*"THEN50110
50120 FORN=TM+2TOTM+200
50130 READML:IF ML>255THENN=64*1024:GOTO50160
50140 IFML<0THENGOSUB50190:GOTO50160
50150 POKEN,ML
50160 NEXT:RESTORE:CLS:PRINT#468,"STARTING ADDRESS =" ;TM+2
50170 PRINT#534,"LOADING MAIN PROGRAM"
50180 RUN"CONQUER": 'IF YOU HAVE CASSETTE, REPLACE THIS
'LINE WITH: 50180 CLOAD
'AND PLACE THE MAIN PROGRAM DIRECTLY
'AFTER IT ON THE CASSETTE

50190 ML=TM-ML+1
50200 IFML>TM+9THENML=ML+1
50210 IFML<0THENML=ML+65536
50220 Z1=INT(ML/256):Z2=ML-Z1*256
50230 POKEN,Z2:POKEN+1,Z1:N=N+1:RETURN
50240 IFTM>32767THENTM=TM-65536
50250 RETURN
  
```



```

50260 DATAML$,50,0,15,0,0,0,0,229,219,255,230,64,238,64,15,15,15,246
50270 DATA 1,95,243,58,-6,87,58,-7,254,0,194,-37,122,50,-7
50280 DATA 42,-1,34,-53,42,-3,34,-50,33,15,0,1,50,0,11,120,177
50290 DATA 194,-55,123,238,3,211,255,95,43,124,181,194,-52,58
50300 DATA -5,254,0,202,-104,61,50,-5,58,-53,130,50,-53,58,-7
50310 DATA 103,122,50,-7,84,195,-49,50,-7,50,-4,50,-2,62,15
50320 DATA 50,-3,62,50,50,-1,251,225,201,300
  
```

Lines 10-40: Set up arrays, define variables to their respective types, and find the entry point for the sound routine. Line 30 sets variable T equal to the beginning of protected memory using BASIC's pointer at locations 16561 and 16562.

```

10 CLEAR 1000:DEFSTR D,N:DEFINT H-L:GOSUB 30:GOSUB 9000
20 CLEAR 100:DEFSTR A,D,N,D:DEFINT C,F,H,I,J,L,P-Y:X=131:FR=0:I=0:U=0:Y=0:Y1=0:D="" :DR="" :LK=0:F=0:Q=0:R=0:DIM A(3,16),H(3),GT(3),VB(3),RF(3),L(3),D(8),TI(3),IV(3),LC(3,16),CT(3,3),BH(3),V(3,16),SA(3),SK(3),T(3),IN(7),N(3),VN(3)
30 T:=PEEK(16561)+2*PEEK(16562)*256:IF T!>32767 THEN T=T!-65536 ELSE T=T!
40 IF X=0 RETURN
  
```

Lines 50-260: Empty graphic strings to contain saved screen displays. Each line contains 64 spaces, which may be entered using the tab or right arrow key to speed the process up.

```

50 A=""
":
A(0,1)="
":
  
```



continued on next page

continued from previous page

A(0,2)="

60 A(0,3)="

A(0,4)="

A(0,5)="

70 A(0,6)="

A(0,7)="

A(0,8)="

80 A(0,9)="

A(0,10)="

A(0,11)="

90 A(0,12)="

A(0,13)="

A(0,14)="

100 A(0,15)="

A(0,16)="

A(1,1)="

110 A(1,2)="

A(1,3)="

A(1,4)="

120 A(1,5)="

A(1,6)="

A(1,7)="

130 A(1,8)="

A(1,9)="

A(1,10)="

140 A(1,11)="

A(1,12)="

A(1,13)="

150 A(1,14)="

A(1,15)="

A(1,16)="

160 A(2,1)="

A(2,2)="

A(2,3)="

170 A(2,4)="

A(2,5)="

A(2,6)="

180 A(2,7)="

A(2,8)="

A(2,9)="

190 A(2,10)="

A(2,11)="

A(2,12)="

200 A(2,13)="

A(2,14)="

A(2,15)="

210 A(2,16)="

A(3,1)="

A(3,2)="

220 A(3,3)="

A(3,4)="

A(3,5)="

230 A(3,6)="

A(3,7)="

A(3,8)="

240 A(3,9)="

A(3,10)="

A(3,11)="

250 A(3,12)="

A(3,13)="

A(3,14)="

260 A(3,15)="

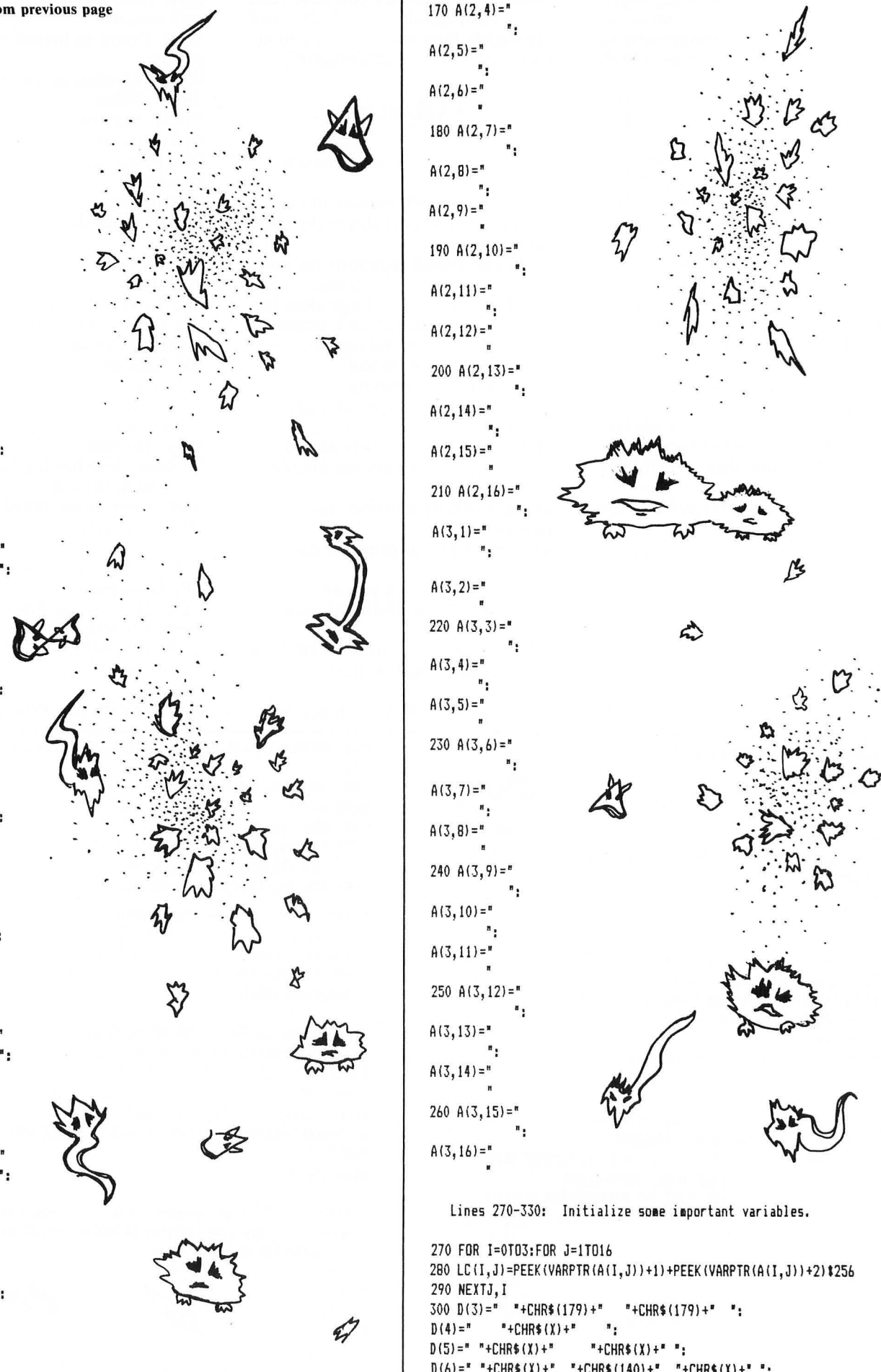
A(3,16)="

Lines 270-330: Initialize some important variables.

```

270 FOR I=0TO3:FOR J=1TO16
280 LC(I,J)=PEEK(VARPTR(A(I,J))+1)+PEEK(VARPTR(A(I,J))+2)*256
290 NEXTJ,I
300 D(3)=" "+CHR$(179)+" "+CHR$(179)+" ":
D(4)=" "+CHR$(X)+" ":
D(5)=" "+CHR$(X)+" "+CHR$(X)+" ":
D(6)=" "+CHR$(X)+" "+CHR$(140)+" "+CHR$(X)+" ":

```



```
D(7)=CHR$(X)+STRING$(7,32)+CHR$(X);
D(8)=CHR$(140)+" "+CHR$(176)+" "+CHR$(140)
310 O="INVASION"
320 GOSUB 10000
330 D(0)=CHR$(143)+CHR$(188)+CHR$(143):D(1)=D(0)+" ":D(2)=" "+D(0)
```



Lines 500-550: Increment variable P to start next player. If variable P is more than the number of players, it is reset to make it the first player's turn again. The bracket character in line 550 is an up-arrow character--CHR\$(91)

```
500 P=P+1:IF P>TP THEN P=0
510 IF H(P)=0 THEN 500
520 CLS:PRINTCHR$(23):PRINT@472-INT(LEN(N(P))/2)*2,N(P)"'S TURN"
530 FOR I=1TO5000:NEXT
540 D=INKEY$:D=4:FR=0:X=30:Y=1:CLS
550 IF TI(P)+4>IV(P) THEN DR="[" :POKE T,15:POKE T+3,1:NAME ELSE DR=" "
```

Lines 1000-1100: Main game loop. Checks for shots being fired, ship movement, ship collisions, and hits scored on aliens. In line 1020, the test PEEK(14368)=16 checks if the "<" key is depressed, and PEEK(16548)=64 checks for the ">" key.

```
1000 '
1010 D=INKEY$:IF D=" " AND (FR=0 OR RF(P)=1) THEN POKE T,60:POKE T+2,1:POKE T+4,30:POKE T+5,5:NAME:IF FR<>0 PRINT@F, " ";FR=1:F=X+65
1020 IF PEEK(14368)=16 THEN X=X-1:Y=Y-1:Y1=0 ELSE IF PEEK(14368)=64 THEN X=X+1:Y=2:Y1=1 ELSE Y=0:Y1=0
1030 IF X<0 THEN X=0:Y=0 ELSE IF X>61 THEN X=61:Y=0:Y1=0
1040 IF FR>1 PRINT@F, " ";F=F+64:IF FR=16 THEN FR=0
1050 U=U+1:IF U=Q THEN U=0:Y=0:Y1=0:PRINT@1023, " ";A(P,L(P));L(P)=L(P)+1:POKE T,Q*50:NAME:IF L(P)=17 THEN L(P)=1:TI(P)=TI(P)+1:IF TI(P)=IV(P) GOTO 4000ELSE IF TI(P)+4>IV(P) THEN DR="[" :POKE T,15:POKE T+3,1:NAME ELSE DR=" "
1060 FOR I=15360+XT015362+X:IF PEEK(I)>32 AND PEEK(I)<46 THEN 3000ELSE NEXT
1070 PRINT@0,USING"##";H(P);:PRINT@58,DR;USING"#####";GT(P);:IF VB(P)<>1 PRINT@X-Y1,D(Y);
1080 IF FR<>0 THEN LK=PEEK(15296+F):IF LK<>32 AND FR>1 THEN R=1:GOTO 2000ELSE LK=PEEK(15360+F):IF LK<>32 THEN R=0:GOTO 2000
1090 IF FR<>0 PRINT@F, " ";FR=FR+1
1100 GOTO 1010
```

Lines 2000-2850: Routines to destroy and divide aliens, adding to the score appropriately.

```
2000 '
2010 LV=L(P)+FR-16-R:IF LV<1 THEN LV=LV+16
2020 SP=F-FR*64-1
2030 IF LK=42 GOSUB 2810ELSE IF LK=35 GOSUB 2610ELSE IF LK=43 GOSUB 2410ELSE GOSUB 2200
2040 FR=0:IF GT(P)>BH(P) THEN H(P)=H(P)+1:BH(P)=BH(P)+5000:IF GT(P)>99999 THEN 6000ELSE FOR I=1TO3:POKE T,10:POKE T+3,2:NAME:FOR J=1TO10:NEXTJ,I
2050 IF CT(P,0)=T(P)*8 GOSUB 8000:DR=" ":GOTO 540ELSE 1010
2200 '- DASHES -
2210 POKE T,60:NAME:PRINT@F-R*64, " ";
2220 POKE LC(P,LV)+SP+1,32
2230 GT(P)=GT(P)+100
2240 Q=1:U=0:CT(P,0)=CT(P,0)+1
2250 POKE T,40:NAME:RETURN
2400 '++ PLUSES ++
2410 IF PEEK(15359+F-R*64)=43 THEN SP=SP-1
2420 POKE T,125:NAME:PRINT@F-R*64+SP, "- ";
2430 POKE LC(P,LV)+SP,45:POKE LC(P,LV)+SP+1,32:POKE LC(P,LV)+SP+
```



```
2,32:POKE LC(P,LV)+SP+3,45
2440 Q=2:U=1:CT(P,1)=CT(P,1)+1:IF CT(P,1)=T(P)*4 THEN IV(P)=SA(P)*4:DR=" "
2450 GT(P)=GT(P)+50
2460 POKE T,75:NAME:RETURN
2600 '#### NUMBERS ####
2610 IF V(P,LV)+8<SP THEN RN=V(P,LV)+8 ELSE RN=V(P,LV)-2
2620 POKE T,175:NAME:PRINT@F-R)*64+RN, "++ ++";
2630 POKE LC(P,LV)+RN,43:POKE LC(P,LV)+RN+1,43:FOR I=2TO5:POKE LC(P,LV)+RN+I,32:NEXTI:POKE LC(P,LV)+RN+6,43:POKE LC(P,LV)+RN+7,43
2640 Q=3:U=1:CT(P,2)=CT(P,2)+1:IF CT(P,2)=T(P)*2 THEN IV(P)=SA(P)*3:DR=" "
2650 GT(P)=GT(P)+25
2660 POKE T,125:NAME:RETURN
2800 '##### STARS #####
2810 POKE T,225:NAME:PRINT@F-R)*64+V(P,LV), "#### ####";
2820 FOR I=0TO3:POKE LC(P,LV)+V(P,LV)+I,35:NEXTI:FOR I=4TO9:POKE LC(P,LV)+V(P,LV)+I,32:NEXTI:FOR I=10TO13:POKE LC(P,LV)+V(P,LV)+I,35:NEXTI
2830 Q=4:U=1:CT(P,3)=CT(P,3)+1:IF CT(P,3)=T(P) THEN IV(P)=SA(P)*2:DR=" "
2840 GT(P)=GT(P)+10
2850 POKE T,175:NAME:RETURN
```

Lines 3000-3100: Handles the explosion of the player's ship after a crash, and checks for end of game.

```
3000 '
3010 FOR I=3TO7 STEP2
3020 IF X>58 PRINT@X-3,LEFT$(D(I),67-X);:PRINT@X+61,LEFT$(D(I+1),67-X);:GOTO 3050
3030 IF X<3 PRINT@0,RIGHT$(D(I),X+6);:PRINT@64,RIGHT$(D(I+1),X+6);:GOTO 3050
3040 PRINT@X-3,D(I);:PRINT@X+61,D(I+1);
3050 POKE T,I*25:POKE T+2,1:POKE T+4,30:POKE T+5,2:NAME
3060 NEXTI
3070 H(P)=H(P)-1:IF H(P)=0 CLS:PRINTCHR$(23):PRINT@462-INT(LEN(N(P))/2)*2,N(P)"'S MISSION IS OVER":FOR I=1 TO 2000:NEXT
3080 FOR I=0T0TP
3090 IF H(I)>0 THEN 500ELSE NEXT I
3100 GOTO 7000
```

Lines 4000-4930: Invasion routine. Called after the aliens have passed all the way to the top of the screen a certain number of times, based on the skill level. The "INVASION" message displayed is contained in variable O\$.

The routine starting at lines 4300, 4500, 4700 and 4900 are routines for each type of alien to determine such things as how many points the invasion will cost.

```
4000 '
4010 CLS:PRINT CHR$(23)
4020 DN IV(P)/SA(P) GOSUB 4300,4500,4700,4900
4030 FOR I=0TO7:IN(I)=464+I*4:NEXT
4040 FOR J=1TO8
4050 L=RND(8)-1:IF IN(L)=0 THEN 4050
4060 IB=(64+RND(3)-2)*(3-2*RND(2))
4070 FOR I=IN(L)+IB*7TOIN(L) STEP-IB
4080 PRINT@I,0;
4090 POKE T,40+I/ABS(IB)*10:NAME
4100 PRINT@I, " ";
4110 NEXTI
4120 PRINT@IN(L),MID$(O,L+1,1);:IN(L)=0
4130 NEXTJ
4140 FOR I=100 TO 200 STEP5:POKE T,I-50:NAME:POKE T,I:NAME:POKE T,300-I:NAME:POKE T,250-I:NAME:NEXT
4150 PRINT@450,"YOU ARE PENALIZED";IP;"POINTS";
4160 FOR I=1TO2000:NEXT
```



continued on next page

continued from previous page

```

4170 GT(P)=GT(P)-IP
4180 IF GT(P)<0 THEN 5000
4190 CLS:FR=0:Q=4:U=1:DR=" "
4200 IF CT(P,0)=T(P)*8 GOSUB 8000:GOTO 540ELSE 1010
4300 ***** STARS *****
4310 FOR I=1TO16
4320 IF A(P,I)<>A THEN IF PEEK(LC(P,I)+V(P,I)+4)=42 THEN FOR J=4
TO9:POKE LC(P,I)+V(P,I)+J,32:NEXT
4330 NEXTI
4340 IP=T(P)-CT(P,3):IV(P)=SA(P)*2
4350 CT(P,2)=CT(P,2)+IP*2:CT(P,1)=CT(P,1)+IP*4:CT(P,0)=CT(P,0)+I
P*8
4360 IP=IP*1000:O1="*"
4370 RETURN
4500 ***** NUMBERS *****
4510 FOR I=1TO16
4520 IF A(P,I)=A THEN 4550
4530 LC=LC(P,I)+V(P,I):IF PEEK(LC)=35 THEN FOR J=0TO3:POKE LC+J,
32:NEXT
4540 IF PEEK(LC+10)=35 THEN FOR J=0TO3:POKE LC+10+J,32:NEXTJ
4550 NEXTI
4560 IP=T(P)*2-CT(P,2):IV(P)=SA(P)*3
4570 CT(P,1)=CT(P,1)+2*IP:CT(P,0)=CT(P,0)+4*IP
4580 IP=IP*500:O1="*"
4590 RETURN
4700 '++ PLUSES ++
4710 FOR I=1TO16
4720 IF A(P,I)=A THEN 4770
4730 LC=LC(P,I)+V(P,I):IF PEEK(LC-2)=43 THEN POKE LC-2,32:POKE L
C-1,32
4740 IF PEEK(LC+4)=43 THEN POKE LC+4,32:POKE LC+5,32
4750 IF PEEK(LC+8)=43 THEN POKE LC+8,32:POKE LC+9,32
4760 IF PEEK(LC+14)=43 THEN POKE LC+14,32:POKE LC+15,32
4770 NEXTI
4780 IP=T(P)*4-CT(P,1):IV(P)=SA(P)*4
4790 CT(P,0)=CT(P,0)+2*IP
4800 IP=IP*250:O1="+"
4810 RETURN
4900 '- DASHES -
4910 IP=100*(T(P)*8-CT(P,0))
4920 CT(P,0)=T(P)*8:O1="-"
4930 RETURN

```

Lines 5000-5080: Ending routine when a negative score is achieved.

```

5000 '
5010 H(P)=1:GT(P)=0
5020 CLS:PRINT CHR$(23)
5030 PRINT@330,"YOUR SCORE IS NEGATIVE"
5040 PRINT@456,"THANKS TO YOUR INEPTITUDE"
5050 PRINT@512,"THE ALIENS HAVE TAKEN OVER EARTH"
5060 PRINT@646,"PREPARE TO PERISH IN SPACE"
5070 X=3:Y=2:FOR I=1TO30:Z=X:X=Y:Y=Z:POKE T+1,X:POKE T+2,100/X:N
AME:NEXT
5080 GOTO 3070

```

Lines 6000-6080: Ending routine if score has reached 100,000.

```

6000 '
6010 H(P)=1:GT(P)=100000
6020 CLS:PRINT CHR$(23)
6030 PRINT@336,"CONGRATULATIONS!"
6040 PRINT@456,"YOU HAVE SINGLE HANDEDLY"
6050 PRINT@524,"CONQUERED THE ALIENS"
6060 PRINT@644,"THE SURVIVORS ARE RETREATING"
6070 FOR I=50TO250:POKE T,I:NAME:POKE T,300-I:NAME:NEXT
6080 GOTO 3070

```

Lines 7000-7100: Display the final scores, and allow the game to be restarted.

```

7000 '
7010 CLS:PRINTCHR$(23)
7020 PRINT@256,"NAME";TAB(11)"VERSION";TAB(20)"LEVEL";TAB(27)"SC
ORE"
7030 FOR I=0TOTP
7040 PRINTN(I);TAB(13)VN(I);TAB(21)USING"###";23-SK(I);:PRINTTAB(
25)USING"###,###";GT(I);
7050 NEXTI
7060 PRINT:PRINTTAB(1)"DO YOU WANT TO PLAY AGAIN";:INPUT D
7070 IF LEFT$(D,1)="Y" RUN 20
7080 CLS:PRINT CHR$(23)
7090 PRINT@464,"EARTH IS DOOMED":PRINT@832,""
7100 END

```

Lines 8000-8180: Set up a wave of aliens.

```

8000 '
8010 CLS:PRINT CHR$(23):PRINT@398,"HERE COMES ANOTHER"
8020 PRINT@530,"WAVE OF ALIENS"
8030 FOR I=0TO3:CT(P,I)=0:NEXT
8040 SA(P)=SA(P)+SK(P):IV(P)=SA(P)
8050 T(P)=T(P)+4:IF T(P)>16 THEN T(P)=16:SA(P)=4*SK(P)
8060 TI(P)=0
8070 FOR I=1TO16
8080 IF A(P,I)=A THEN 8120
8090 FOR J=0TO62
8100 POKE LC(P,I)+J,32
8110 NEXTJ
8120 NEXTI
8130 FOR I=1TOT(P)
8140 L=RND(16):X=RND(43)+7
8150 IF A(P,L)=A THEN FOR J=0TO5:POKE LC(P,L)+X+J,42:NEXT ELSE 8
140
8160 V(P,L)=X-4
8170 NEXTI
8180 DR=" ":L(P)=0:RETURN

```

Lines 9000-9350: Print the graphic introduction to the program. The data from line 9140-9240 contains the data for the graphic strings. Each item in data is one graphics character in the string. A value of 128 is added to each piece of data read to get the proper graphics code. By adding the 128, all the data items are reduced in length from three digits each to only one or two, saving time.

```

9000 '
9020 CLS:PRINTCHR$(23):PRINT@456,"IF YOU HAVE THE SKILL"
9030 FOR J=1TO2:FOR I=1TO106:IF I=54 THEN D(J)=D(J)+STRING$(11,3
2)
9040 READ L:D(J)=D(J)+CHR$(L+128):NEXTI,J
9050 CLS:PRINTCHR$(23):PRINT@466,"AND THE GUTS"
9060 FOR J=3TO4:FOR I=1TO54:IF I=28 THEN D(J)=D(J)+STRING$(37,32
)
9070 READ L:D(J)=D(J)+CHR$(L+128):NEXTI,J
9080 PRINT@462,"YOU'RE READY FOR"
9090 FOR J=5TO6:FOR I=1TO127:READ L:D(J)=D(J)+CHR$(L+128):NEXTI,
J
9100 CLS:FOR I=0TO384 STEP64:PRINT@389-I,D(1);:PRINT@517+I,D(2);
:IF I=0 THEN FOR J=1TO2000:NEXT ELSE POKE T,243-I/4:POKE T+2,5:P
OKE T+4,32:POKE T+5,255:NAME
9110 CLS:NEXTI:PRINT@5,RIGHT$(D(1),53);:PRINT@965,LEFT$(D(2),53)
;:POKE T,131:POKE T+2,5:POKE T+4,32:POKE T+5,255:NAME:CLS:FOR J=
1TO100:NEXT:POKE T,115:POKE T+2,5:POKE T+4,32:POKE T+5,255:NAME:
FOR J=1TO500:NEXT
9120 PRINT@402,D(3);:PRINT@530,D(4);:FOR J=1TO2000:NEXT

```



```

9130 CLS:PRINT@0,RIGHT$(D(5),63);:PRINT@960,LEFT$(D(6),63);:POKE
T,84:POKE T+2,5:POKE T+4,32:POKE T+5,1:NAME:CLS:FOR I=384TO0 ST
EP-64:PRINT@384-I,D(5);:PRINT@512+I,D(6);
9140 POKE T,196-I/4:POKE T+2,5:POKE T+4,32:POKE T+5,1:NAME:IFI(<
0 CLS:NEXTI ELSE FOR J=1TO2000:NEXT
9150 DATA 63,63,15,15,15,60,48,0,0,15,15,15,63,63,15,15,15,0,63,
63,0,0,0,63,63,0,15,15,15,63,63,15,15,15,0,63,63,15,15,15,60,4
8,0,0,63,63,15,15,15,15,15
9160 DATA 63,63,0,0,0,2,63,61,0,0,0,63,63,0,0,0,63,63,0,0,0,
0,63,63,0,0,0,63,63,0,0,0,63,63,0,0,0,2,63,61,0,63,63,48,48,
48,48,48,48
9170 DATA 63,63,0,0,0,32,63,31,0,0,0,63,63,0,0,0,2,47,61,16,
32,62,31,1,0,0,0,63,63,0,0,0,63,63,0,0,0,32,63,31,0,63,63,3,
3,3,3,3,3
9180 DATA 63,63,60,60,60,15,3,0,0,60,60,63,63,60,60,60,0,0,0,
11,63,63,7,0,0,0,60,60,60,63,63,60,60,60,0,63,63,60,60,60,15,3,0
,0,63,63,60,60,60,60,60
9190 DATA 42,63,31,15,15,15,47,63,21,42,63,63,52,0,0,0,63,63,0,6
3,63,15,15,15,60,48,0,42,63,53,48,48,48,58,63,21,42,63,23,47,61,
16,0,63,63,0,63,63,0,0,0,2,63,61
9200 DATA 42,63,23,3,3,3,43,63,21,42,63,21,0,11,63,52,63,63,0,63
,63,0,0,0,32,63,31,42,63,21,0,0,0,42,63,21,42,63,21,0,0,2,47,63,
63,0,63,63,60,60,60,15,3,0
9210 DATA 42,63,31,15,15,15,15,15,42,63,31,15,15,15,47,63,21,4
2,63,63,52,0,0,0,63,63,0,63,63,15,15,15,15,15,63,63,0,63,63,0,0,0,
63,63,0,63,63,15,15,15,15,15,15,0,63,63,15,15,15,15,63,63,0
9220 DATA 42,63,21,0,0,0,0,0,42,63,21,0,0,0,42,63,21,42,63,23,
47,61,16,0,63,63,0,63,63,0,0,0,63,63,0,63,63,0,0,0,63,63,0,6
3,63,48,48,48,48,48,0,63,63,48,48,48,48,63,63
9230 DATA 42,63,21,0,0,0,0,0,42,63,21,0,0,0,42,63,21,42,63,21,
0,11,63,52,63,63,0,63,63,0,12,52,16,63,63,0,63,63,0,0,0,63,63,
0,63,63,3,3,3,3,3,0,63,63,15,63,51,3,3,3,0
9240 DATA 42,63,61,60,60,60,60,60,2,63,61,60,60,60,62,63,21,
42,63,21,0,0,2,47,63,63,0,63,63,60,60,60,63,63,63,0,63,63,60,60,

```

```

60,60,63,63,0,63,63,60,60,60,60,60,60,0,63,63,0,0,3,15,60,48
9250 CLS:PRINTCHR$(23):PRINT@192,"COMPLETELY CREATED AND DEVELOP
ED"
9260 PRINT@350,"BY"
9270 PRINT@466,"PHILLIP SOINE"
9280 PRINT@526,"905 N. 15TH STREET"
9290 PRINT@584,"MOUNT VERNON, WA. 98273"
9300 PRINT@716,"COPYRIGHT JULY 1980"
9310 FOR I=1TO5000:NEXT
9320 CLS:PRINTCHR$(23):PRINT@398,"SPECIAL THANKS TO"
9330 PRINT@466,"JOHN MCKNIGHT"
9340 PRINT@516,"FOR THE STRING STORAGE IDEA"
9350 RETURN

```

Lines 10000-10120: Set the playing parameters for each player, and initializes the first wave of aliens.

```

10000 '
10010 CLS:PRINT CHR$(23):PRINT@454,"HOW MANY PLAYERS (1-4)";
10020 INPUT TP:IF TP<1 OR TP>4 THEN 10010ELSE TP=TP-1
10030 FOR P=0TOTP
10040 CLS:PRINTCHR$(23):PRINT@448,"NAME OF PLAYER NO. ";P+1;:INPU
T N(P)
10060 CLS:PRINTCHR$(23):PRINT@458,"SKILL LEVEL (1-20)";
10070 INPUT SK(P):IF SK(P)>20 OR SK(P)<1 THEN 10060
10080 H(P)=3:BH(P)=4999:SK(P)=23-SK(P)
10085 PRINT@458,"WHICH VERSION (1-4)";:INPUT VN(P):IF VN(P)<1 OR
VN(P)>4 THEN 10085 ELSE IF VN(P)>2 THEN VB(P)=1
10087 IF VN(P)/2<>INT(VN(P)/2) THEN RF(P)=1
10090 CLS:PRINTCHR$(23):PRINT@472,"WORKING"
10100 GOSUB 8030
10110 NEXTP
10120 RETURN

```



### S-80 ONE LINERS

```

1 INPUT:CLS:G=42:FOR T=0TO127:FOR D=GTO47:SET(T,D):NEXT:G=G+RND(3)
)-2+1*(G=47):NEXT:FOR F=0TO99-H*5:P=PEEK(14400):H=H+.5*(P=32)-.5*(
P=64):V=V+.25+.5*(P=8):RESET(X,Y):X=X+H:Y=Y+V:IF POINT(X,Y)=0SET
(X,Y):NEXT ELSE IF V<=11 AND H=0 PRINT@349,"LANDED" ELSE RESET(X,Y):RUN

```

John Boyer  
Anaheim, CA

```

10 CLS:J=27:P=15391:FOR M=1TO9E9:T=RND(99):FOR L=RND(T)TOTSTEP RND(
9)/9:J=50-ABS(ABS(J+SIN(L))-50):A=PEEK(14400)/32:P=P-(A AND 1)-(A
/2 AND 1)*(PEEK(14464)+1):PRINTTAB(J)!!!! , !!!":IF PEEK(P)=33
PRINT"SCORE:"SELSEPOKEP,191:S=S+1:NEXTL,M

```

James Petivan  
New Orleans, LA

```

10 CLS:PRINT"HIT SPACE BAR TO SEE A NEW PATTERN":FOR X=1TO700:NEX
T X:FOR T=1TO50000:CLS:FOR Z=1TO1160:X=RND(64)-1:Y=RND(24)-1:SET(X,
Y):SET(X,47-Y):SET(127-X,47-Y):SET(127-X,Y):IF INKEY$="" THEN NEXT
Z,T:GOTO10 ELSE NEXTT:GOTO10

```

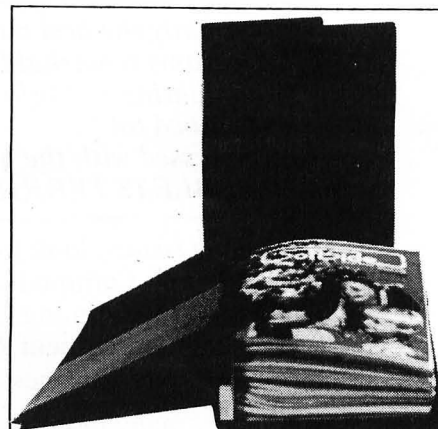
Mark Soupen  
Hamilton, OH

```

10 CLEAR 255:FOR X=1TO255:PRINTSTRING$(X,191),:NEXT:FOR X=255TO1ST
EP-1:PRINTSTRING$(X,191),:NEXT:GOTO10

```

Brian Yamauchi  
Oxford, OH



## VINYL BINDERS

Collectors! Protect your SoftSide back issues, Volumes I and II, or any publication of your choice, with these durable wood-grain vinyl binders with inside pocket and clear spine sleeve for easy identification. Holds and protects 12 back issues. A regular \$4.95 value, SALE priced at \$3.95\*. FREE (while supply lasts) with the purchase of Volume I or II (12 issue collection of SoftSide).

SMALL #75-401001 \$3.95  
8 1/2 x 11 #75-401002 \$7.95



6 South St. Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790

# "NIBBLE<sup>®</sup> IS TERRIFIC" (For Your Apple)



**NIBBLE IS:** *The Reference for Apple computing!*

**NIBBLE IS:** One of the Fastest Growing new Magazines in the Personal Computing Field.

**NIBBLE IS:** Providing Comprehensive, Useful and Instructive Programs for the Home, Small Business, and Entertainment.

**NIBBLE IS:** A Reference to Graphics, Games, Systems Programming Tips, Product News and Reviews, Hardware Construction Projects, and a host of other features.

**NIBBLE IS:** A magazine suitable for both the Beginner and the Advanced Programmer.

Each issue of NIBBLE features significant new Programs of Commercial Quality. Here's what some of our Readers say:

- "Certainly the best magazine on the Apple II"
- "Programs remarkably easy to enter"
- "Stimulating and Informative; So much so that this is the first computer magazine I've subscribed to!"
- "Impressed with the quality and content."
- "NIBBLE IS TERRIFIC!"

*In coming issues, look for:*

- Stocks and Commodities Charting
- Assembly Language Programming Column
- Pascal Programming Column
- Data Base Programs for Home and Business
- Personal Investment Analysis
- Electronic Secretary for Time Management
- The GIZMO Business Simulation Game

And many many more!

NIBBLE is focused completely on the Apple Computer systems.

Buy NIBBLE through your local Apple Dealer or subscribe now with the coupon below.

**Try a NIBBLE!**

**nibble**

Box 325, Lincoln, MA. 01773 (617) 259-9710

**I'll try nibble!**

**Enclosed is my \$17.50 (for one year).  
(Outside U.S., see special rates on this page.)**

**check**     **money order**

Your subscription will begin with the next issue published after receipt of your check/money order.

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_

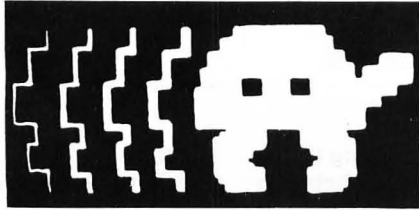
State \_\_\_\_\_ Zip \_\_\_\_\_

**NOTE:**

First Class or Air Mail is required for all APO, FPO and all foreign addresses with the following additional amounts

**Air Mail Postage Rates**    12-14 oz. x 8  
 Europe \$32.00  
 Mexico and Central America \$21.00  
 South America \$32.00  
 Middle East \$35.00

Africa: North \$32.00  
 Central \$43.00  
 South \$43.00  
 Far East, Australia \$43.00  
 Canada \$18.00  
 APO FPO \$7.50



FROM

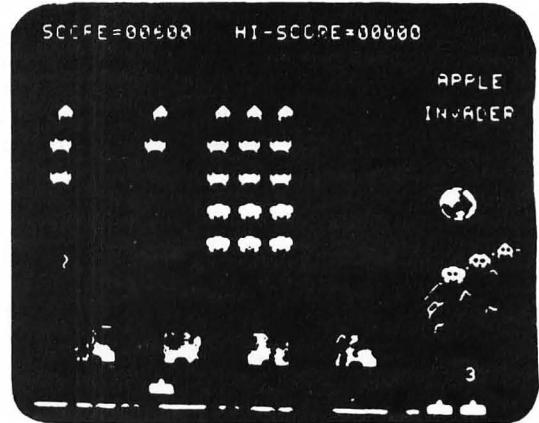
**sensational  
software**

### SPACE WAR

You're in command in SPACE WAR! Destroy your opponent's ship by forcing him to collide with the sun or to explode upon re-entry from hyperspace . . . or challenge him to fight face to face with missile fire. You're in command of the speed and direction of your ship. You control the timing of your missiles. You select the game mode from five options, including Reverse Gravity, and the battle begins . . . Accelerate to place your shots — and escape into hyperspace before your opponent comes within range. But be wary, he (or she!) may circle out of sight and reappear on the opposite side of the galaxy! (This is the classic MIT game redesigned especially for the Apple.)

### SUPER INVASION

SUPER INVASION is the original invasion game, with the original moon creatures and faster action than any other invasion game. Features superb high resolution graphics, nail-biting tension and hilarious antics by the moon creatures! Self-running "attract mode" of operation for easy learning and demonstration of the game. As good in every way as the famous Invaders arcade game. High speed action! Sound effects! Runs on the Apple II and the Apple II Plus.



- Super Invasion **Only**
- 32K, Apple II, Cassette . . . . . \$19.95
- Space War and Super Invasion **Combined**
- 48K, Apple II, Disk . . . . . \$29.95

### AIR TRAFFIC CONTROLLER

In AIR TRAFFIC CONTROLLER you assume responsibility for the safe flow of air traffic within a 15 x 25 mile area up to 5,000 feet in altitude. During your shift as a controller in charge of this airspace, 26 aircraft become active and under your control. Jets and prop planes have to be guided to and from the two airports, navigational beacons and ten entry/exit fixes. The aircraft enter the controller's airspace at various altitudes and headings whether or not you are ready.

- 16K, Apple II and Apple II + Cassette . . . . . \$11.95
- 16K, S-80, Cassette . . . . . \$11.95





# KRAZY TALK

by David Gash

**“Krazy Talk” is an S-80 game requiring 16K of memory.**

With all the emphasis on “order” and “logic” in today’s world (especially within the computer field), we sometimes tend to forget just how much fun plain old silliness can be! With that thought in mind, let’s leave the realm of most computer games which require a) skill, b) endurance, or even worse, c) actual thought, and take a look at a new game which requires d) none of the above (only a slightly warped sense of humor) — “Krazy Talk”!

“Krazy Talk” is a friendly competition between two players, using a S-80 16K Level II (or up) as playing surface and scorekeeper. Given a random sampling of 20 words chosen from nouns, verbs, articles, etc., the players try to use as many words as they can to form interconnecting sentences, crossword-style, by moving the words around on the screen, within a given time limit. Most of the words refer to parts of the body (such as knees, teeth, etc.), while most of the verbs supplied seem rather odd when used with the rest of the words. So, usually, very few of the sentences make much real sense, but therein lies the most enjoyable part of the game: depending on how the players construct their sentences, the outcome can range from mildly hilarious to downright ridiculous! (There is even a “BLANK”, which may be used as any word the player wishes!)

The program has a number of especially entertaining and useful features, such as visually interesting displays which operate even during internal operations. This means the player never has to sit through a blank screen while the computer does something known only to itself — there is always some kind of screen activity. There is, of course, a built-in rules/instructions routine, available at the players’ discretion.

The program keeps track of each player and uses his or her name throughout the game, which may be played on any of three levels — the higher the level, the less time is allotted.

During each player’s turn, the program maintains an accurate, constantly-running clock timer which

ticks off the seconds remaining to the player. This timer runs even between keystrokes of the same move, but does NOT require an expansion interface.

The program makes extensive use of the keyboard strobe (INKEY\$), thereby eliminating the need to press ENTER to register a move, when split seconds count. ENTER is used only to end a turn early or to terminate a series of unused words at game’s end. This feature allows automatic error trapping — the program catches and disallows any attempt to move a word from an empty position or to an occupied one. When a valid move is made, the word immediately jumps to its new position, and the program is ready for another move.

When a player’s time runs out (or if he feels he can’t use any more words and ends his turn early), he must tell the program the positions of any unused words. The program then automatically calculates the player’s score, displays it, and prepares the other player’s grid. The player receives five points for each word used once, 10 points for each word used twice (in two directions), and loses three points for each unused word.

When both players have completed their turns, the program displays the scores and an appropriate remark, and of course, there is an option to play again.

The word grid is different every time the game is played — the sentence combinations possible from each random set of words are endless; this eliminates the rapid onset of boredom so common with computer games after they have been played awhile.

These are the main features of the

game — please give it a workout. I believe you will find it funny and enjoyable; but don’t be sensible or logical — remember, KRAZY is the name of the game!

## INTEGER VARIABLES:

A,B/C,D/X,Y: Subscripts for word positions in array WD\$.

L: Counter in timing loop.

I: Counter in graphics print loop.

M: Minutes to play (set by playing level).

P1, P2: Players’ scores.

PA: Array containing “PRINT AT” positions for each word.

PL: Playing level.

PN: Current player number (1 or 2).

Q: Random print position (1 - 1023).

R: Random word from group just read (1 - 6).

S: Seconds left to play.

S1: Score for words used once.

S2: Score for words used twice.

S3: Score for unused words (negative).

T: Temporary counter used in various loops.

TS: Total score for current player (S1 + S2 - S3).

WU: Counter for words used once.

WD: Counter for words used twice.

UN: Counter for unused words.

## STRING VARIABLES

C\$: One character picked up from keyboard via “INKEY\$”.

D1\$, D2\$: Players’ names in order of final scores.

P\$(1), P\$(2): Players’ names as input at start of game.

PN\$: Current player’s name.

W\$: Array to hold each group of six words as read in.

WD\$: Array to hold and manipulate word grid as shown on screen.

Initialization and introductory display.

```
100 ' "KRAZY TALK" SENTENCE BUILDING GAME
110 ' COPYRIGHT (C) 1981 : CREATIVE MICRO SYSTEMS
120 ' 1820 S. MAIN, SUITE F
130 ' BROKEN ARROW, OK. 74012
140 ' 918 / 258 - 3442
150 CLEAR 1000
160 DIM W$(6), WD$(13, 8), PA(12, 7)
170 CLS
180 FOR Z=1 TO 3
190 RESTORE
200 FOR X=1 TO 20
```

```

210 FOR Y=1 TO 6
220 READ W$(Y)
230 NEXT Y
240 PRINT @ RND(1023), W$( RND(6)) " ";
250 NEXT X
260 NEXT Z
270 GOSUB 290
280 GOTO 360

```

Subroutine to draw a frame around the screen.

```

290 PRINT @ 0, CHR$(191)CHR$(143)STRING$(60, 131)CHR$(143)STRING
$(2, 191),
300 FOR I=127 TO 959 STEP 64
310 PRINT @ I, STRING$(2, 191);
320 NEXT I
330 PRINT CHR$(188)STRING$(60, 176)CHR$(188);
340 POKE 16383, 191
350 RETURN

```

Draw title box and display the title.

```

360 PRINT @ 391, CHR$(191)STRING$(47, 131)CHR$(191);
370 PRINT @ 455, CHR$(191)" GREETINGS! THIS IS THE GAME OF 'KRAZ
Y TALK'!! " CHR$(191);
380 PRINT @ 519, CHR$(191)" WOULD YOU LIKE TO SEE THE INSTRUCTIO
NS? (Y/N) " CHR$(191);
390 PRINT @ 583, CHR$(143)STRING$(47, 140)CHR$(143);
400 C$=INKEY$
410 IF C$="" OR(C$<>"Y" AND C$<>"N")THEN 400
420 IF C$="Y" THEN GOSUB 2260

```

Input the players' names and the playing level.

```

430 CLS
440 GOSUB 290
450 PRINT @ 82, "***** KRAZY TALK! *****";
460 PRINT @ 330, "";
470 INPUT "WHAT IS THE NAME OF PLAYER # 1";P$(1)
480 PRINT @ 383, STRING$(2, 191);
490 PRINT @ 447, CHR$(191);
500 PRINT @ 394, "";
510 INPUT "WHAT IS THE NAME OF PLAYER # 2";P$(2)
520 PRINT @ 448, CHR$(191);
530 PRINT @ 511, CHR$(191)CHR$(191);
540 PRINT @ 525, "PLEASE SELECT YOUR PLAYING LEVEL";
550 PRINT @ 582, "(THE HIGHER THE LEVEL, THE LESS TIME YOU'LL HAV
E)";
560 PRINT @ 660, "1. SLIGHTLY KRAZY";
570 PRINT @ 724, "2. DEFINITELY KRAZY";
580 PRINT @ 788, "3. TOTALLY KRAZY";
590 PRINT @ 919, "----> <---";
600 C$=INKEY$
610 IF C$="" OR C$<"1" OR C$>"3" THEN 600
620 PRINT @ 924, C$;
630 FOR X=1 TO 500
640 NEXT X
650 IF C$="1" THEN PL=5 ELSE IF C$="2" THEN PL=4 ELSE PL=3

```

Beginning of the main game loop.

```

660 FOR PN=1 TO 2
670 CLS
680 PN$=P$(PN)

```

Choose random words for the playing grid.

```

690 PRINT @ 455, "STAND BY, " PN$ " -- I'M SETTING UP YOUR GRID...

```

```

700 FOR A=1 TO 12
710 FOR B=1 TO 7
720 WD$(A, B)=""
730 PA(A, B)=0
740 NEXT B
750 NEXT A
760 RESTORE
770 FOR X=1 TO 20
780 FOR Y=1 TO 6
790 READ W$(Y)
800 NEXT Y
810 R=INT(RND(36)/6)
820 IF R=0 THEN R=1
830 CLS
840 Q=RND(1023)
850 PRINT @ Q, W$( RND(6));
860 FOR T=1 TO 10
870 NEXT T
880 PRINT @ Q, " ";
890 A=RND(12)
900 B=RND(7)
910 IF WD$(A, B)<>" " THEN 890
920 WD$(A, B)=W$(R)
930 NEXT X
940 PRINT @ 470, "GOT IT! HERE WE GO!"
950 P=121
960 FOR A=1 TO 12
970 FOR B=1 TO 7
980 PA(A, B)=P+(B*9)
990 PRINT @ 470, CHR$(30);
1000 NEXT B
1010 P=P+64
1020 PRINT @ 470, "GOT IT! HERE WE GO!";
1030 NEXT A

```

Words for setting up the playing grid.

```

1040 DATA CHEW, STEAL, TICKLE, SNIFF, HURT, BLANK, BLUE, GROW, B
ITE, GIVE, CHOKE, CHASE, SEE, MAKE, HEAR, LOSE, CHANGE, CHOOSE,
PLAY, GUESS, REMOVE, FIGHT, MASH, FLY, RACE, PAINT, BLANK
1050 DATA THE, THAT, ANY, THIS, WHICH, OUR, HIS, MY, THEIR, YOUR
, BLANK, A, THAT, THE, MY, THEIR, YOUR, HIS, HER, A, ANY, SOME,
MANY
1060 DATA AT, DOWN, OUT, FOR, WITH, INTO, IN, ON, UP, OFF, TO, F
ROM, UNDER
1070 DATA YOU, HE, SHE, WE, I, THEY, ME, IT, US, THEM, HER, HIM,
YOU, ME, HE, HER, WE, BLANK
1080 DATA WARPED, KNOBBY, CRAZY, GREEN, GOOFY, HAIRY, SPOT, BROK
EN, PICKLED, ITCHY, FAST, BLANK, BIG, SLOW, DUMB, WEIRD, FUNNY, F
URRY
1090 DATA TEETH, NOSE, TOE, NAIL, HAIR, HEAD, EYE, EAR, NECK, CH
IN, ARM, BLANK, BODY, TUMMY, LEG, FEET, FINGER, WRIST, KNEE, BON
E, ELBOW

```

Display the playing grid on the screen.

```

1100 PRINT @ 25, PN$"'S TURN";
1110 PRINT @ 63, "";
1120 FOR X=1 TO 7
1130 PRINT " " X " ";
1140 NEXT X
1150 PRINT CHR$(13);
1160 FOR X=1 TO 12
1170 PRINT CHR$(64+X)CHR$(94)STRING$(60, " ")CHR$(93)CHR$(64+X);
1180 NEXT X
1190 FOR X=1 TO 7
1200 PRINT " " X " ";
1210 NEXT X
1220 PRINT @ FROM: TO: (PRESS 'ENTER' TO END);
1230 PRINT @ 1010, "TIME: ";

```

continued on next page

continued from previous page

```
1240 FOR A=1 TO 12
1250 FOR B=1 TO 7
1260 PRINT @ PA(A, B), WD$(A, B);
1270 NEXT B
1280 NEXT A
```

Player PN's turn to play, by rearranging the words on his grid.

```
1290 M=PL
1300 S=0
1310 C=0
1320 PRINT @ 1015, M:" S;
1330 FOR T=1 TO 4
1340 C$=INKEY$
1350 IF C$<>" THEN IF ASC(C$)=13 THEN 1590 ELSE 1430
1360 C=C+1
1370 IF C<35 THEN 1340
1380 C=0
1390 S=S-1
1400 IF S=-1 THEN M=M-1: S=59:IF M=-1 THEN 1590
1410 PRINT @ 1015, M:" S;
1420 GOTO 1340
1430 IF T=1 OR T=3 THEN IF ASC(C$)<65 OR ASC(C$)>76 THEN 1360 EL
SE X=ASC(C$)-64:IF T=1 THEN PRINT @ 966, C$; GOTO 1570 ELSE PRI
NT @ 974, C$; GOTO 1570
1440 IF ASC(C$)<49 OR ASC(C$)>55 THEN 1360
1450 Y=VAL(C$)
1460 IF T=2 THEN PRINT @ 967, C$; ELSE PRINT @ 975, C$;
1470 IF T=2 THEN IF WD$(X, Y)="" THEN PRINT @ 966, " "; GOTO 13
30 ELSE A=X: B=Y: PRINT @ 974, " "; GOTO 1570
1480 IF WD$(X, Y)<>" THEN PRINT @ 974, " "; T=3: GOTO 1340
1490 C=X
1500 D=Y
1510 WD$(C, D)=WD$(A, B)
1520 PRINT @ PA(C, D), WD$(C, D);
1530 WD$(A, B)=""
1540 PRINT @ PA(A, B), " ";
1550 PRINT @ 966, " TO: ";
1560 GOTO 1330
1570 NEXT T
1580 END'ERROR IF THIS IS REACHED
```

Time is up; eliminate all the words not yet used.

```
1590 IF M=-1 THEN PRINT @ 1014, "S UP!!"; FOR X=1 TO 2000: NEXT
X
1600 WU=0
1610 DW=0
1620 UN=0
1630 PRINT @ 960, CHR$(30)"PLEASE TYPE IN ANY UNUSED WORD'S POSI
TION ('ENTER' TO END): ";
1640 FOR T=1 TO 2
1650 C$=INKEY$
1660 IF C$="" THEN 1650
1670 IF C$=CHR$(13) THEN 1780
1680 IF T=1 THEN IF ASC(C$)<65 OR ASC(C$)>76 THEN 1650 ELSE X=AS
C(C$)-64: PRINT C$; GOTO 1760
1690 IF ASC(C$)<49 OR ASC(C$)>55 THEN 1650
1700 Y=VAL(C$)
1710 PRINT C$;
1720 IF WD$(X, Y)="" THEN 1630
1730 WD$(X, Y)=""
1740 PRINT @ PA(X, Y), " ";
1750 UN=UN+1
1760 NEXT T
1770 GOTO 1630
```

Calculate the player's score.

```
1780 PRINT @ 960, CHR$(30)TAB(5)"STAND BY, " PN$ -- I AM CALCUL
ATING YOUR SCORE...";
1790 FOR A=1 TO 12
1800 FOR B=1 TO 7
1810 IF WD$(A, B)="" OR WD$(A, B)="BLANK" THEN 1850
1820 IF WD$(A, B)<>" THEN WU=WU+1 ELSE 1850
1830 IF WD$(A-1, B)<>" AND(WD$(A, B+1)<>" OR WD$(A, B-1)<>"")T
HEN DW=DW+1: GOTO 1850
1840 IF WD$(A+1, B)<>" AND(WD$(A, B-1)<>" OR WD$(A, B+1)<>"")T
HEN DW=DW+1
1850 NEXT B
1860 NEXT A
1870 S2=DW*5
1880 S1=WU*5
1890 S3=UN*3
1900 TS=S1+S2-S3
1910 IF PN=1 THEN P1=TS ELSE P2=TS
1920 PRINT @ 960, CHR$(30)TAB(15)PN$, " YOUR TOTAL SCORE IS:" TS;
1930 FOR X=1 TO 2000
1940 NEXT X
1950 S1=0
1960 S2=0
1970 S3=0
1980 WU=0
1990 DW=0
2000 UN=0
2010 TS=0
2020 NEXT PN
```

End of the game.

```
2030 CLS
2040 PRINT @ 192, TAB(15)"THE GAME IS OVER! THE SCORES ARE:"
2050 PRINT
2060 PRINT TAB(25)P$(1):" P1
2070 PRINT TAB(25)P$(2):" P2
2080 IF P1>P2 THEN D1$=P$(1): D2$=P$(2) ELSE IF P1<P2 THEN D1$=P
$(2): D2$=P$(1) ELSE 2130
2090 PRINT
2100 PRINT TAB(12)"LOOKS LIKE " D1$ " IS KRAZIER THAN " D2$"!!!"
2110 PRINT
2120 GOTO 2160
2130 PRINT
2140 PRINT TAB(12)"LOOKS LIKE YOU TWO ARE EQUALLY KRAZY!!!"
2150 PRINT
2160 PRINT TAB(12)"DO YOU WANT TO PLAY ANOTHER GAME? (Y/N)";
2170 GOSUB 290
2180 C$=INKEY$
2190 IF C$="" OR(C$<>"Y" AND C$<>"N") THEN 2180
2200 IF C$="Y" THEN RUN
2210 PRINT @ 776, "THANKS FOR BEING SO KRAZY -- SEE YOU NEXT TIME
!";
2220 FOR X=1 TO 1500
2230 NEXT X
2240 CLS
2250 END
```

Subroutine to print the instructions.

```
2260 CLS
2270 PRINT @ 84, "***** KRAZY TALK! *****"
2280 PRINT
2290 PRINT " THE OBJECT OF THIS GAME IS TO MAKE AS MANY COMPL
ETE"
2300 PRINT "SENTENCES AS YOU CAN FROM THE WORDS YOU ARE GIVEN IN
THE BRIEF"
2310 PRINT "TIME ALLOTTED (THREE TO FIVE MINUTES, DEPENDING ON YO
UR LEVEL"
```

2320 PRINT"OF EXPERIENCE WITH NONSENSICAL LANGUAGE!)."
 2330 PRINT" YOU ARE GIVEN A TOTAL OF TWENTY WORDS TO WORK WITH,"
 2340 PRINT"CHOSEN AT RANDOM FROM MY 'DICTIONARY OF KRAZY WORDS'. THEY ARE"
 2350 PRINT"PRESENTED ON A 12-BY-7 GRID, AND YOU CAN MOVE A WORD ANYWHERE"
 2360 PRINT"YOU WISH ON THE GRID BY USING THE LETTER-NUMBER COMBINATION"
 2370 PRINT"OF THE WORD ITSELF AND THE SPACE YOU WISH TO MOVE IT TO."
 2380 PRINT"(YOU DON'T HAVE TO PRESS 'ENTER' -- JUST PRESS THE RIGHT KEYS.)"
 2390 PRINT
 2400 PRINT"SPECIFIC RULES ARE ON THE NEXT PAGE; PRESS ANY KEY TO CONTINUE."
 2410 IF INKEY\$="" THEN 2410
 2420 CLS
 2430 PRINT TAB(20)"\*\*\*\*\* RULES \*\*\*\*\*"
 2440 PRINT
 2450 PRINT"1. WHILE THE SENTENCES DON'T NECESSARILY HAVE TO BE REALISTIC,"
 2460 PRINT" THEY SHOULD BE GRAMMATICALLY CORRECT."
 2470 PRINT
 2480 PRINT"2. ALL SENTENCES MUST INTERCONNECT VERTICALLY AND HORIZONTALLY,"
 2490 PRINT" LIKE A CROSSWORD PUZZLE'S LETTERS."
 2500 PRINT
 2510 PRINT"3. THE WORD 'BLANK' MAY BE USED AS ANY WORD YOUR WARPED LITTLE"
 2520 PRINT" HEART DESIRES; HOWEVER, IF IT IS USED IN TWO DIRECTIONS AT"
 2530 PRINT" ONCE, IT MUST BE USED AS THE SAME WORD BOTH WAYS."
 2540 PRINT
 2550 PRINT"4. YOU MAY CHANGE VERB TENSES, OR ADD ENDINGS SUCH AS '-LY',"
 2560 PRINT" '-S', '-ED' OR '-ING' TO MAKE YOUR SENTENCES READ CORRECTLY."
 2570 PRINT
 2580 PRINT"EXPLANATION OF SCORING IS ON NEXT PAGE; PRESS ANY KEY TO GO ON."
 2590 IF INKEY\$="" THEN 2590
 2600 CLS
 2610 PRINT @ 128, TAB(20)"\*\*\*\*\* SCORING \*\*\*\*\*"
 2620 PRINT
 2630 PRINT" YOUR TURN IS OVER WHEN YOUR TIME RUNS OUT. IF YOU ARE"
 2640 PRINT"FINISHED EARLY, PRESS 'ENTER' FOR ANY 'FROM' OR 'TO' POSITION."
 2650 PRINT"I WILL THEN ASK YOU TO TELL ME THE LOCATIONS OF ANY WORDS"
 2660 PRINT"YOU COULDN'T USE. WHEN YOU HAVE DONE THAT, I'LL FIGURE YOU"
 2670 PRINT"SCORE FOR YOU. YOU GET 5 POINTS FOR EACH WORD USED ONCE, 10"
 2680 PRINT"POINTS FOR EACH WORD USED TWICE, AND YOU LOSE 3 POINTS FOR"
 2690 PRINT"EACH WORD YOU COULDN'T USE. THE WORD 'BLANK' ISN'T WORTH"
 2700 PRINT"ANY POINTS TOWARD YOUR TOTAL SCORE."
 2710 PRINT
 2720 PRINT" THAT'S ABOUT IT -- PRESS ANY KEY TO BEGIN THE GAME!"
 2730 IF INKEY\$="" THEN 2730
 2740 RETURN



# BUGS, WORMS,

## and other undesirables

The following line should be substituted for Line 10 in the S-80 program "Dominoes" published in the April SoftSide.

```
10 CLEAR1000:RANDOM:GT=100:G=140:P=15360:DIMA$(28),D(20),P$(20),T$(20),P1$(20)
```



The following lines should be substituted for Lines 900 and 950 in the S-80 program "Kidnapped" published in the December, 1980 SoftSide.

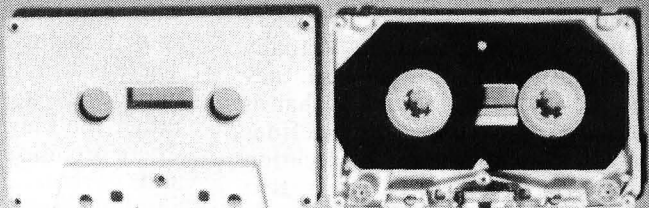
```
900 IF D(B)=11THENIFA=19ANDE$="KEY"ANDI(21)=1THENA$(56)="CRUDE KEY":H$(56)=A$(56):A(56)=19:GOTO4900 ELSEIFE$="OVE"ANDA=37THENPRINT"YOU "A$"WITH THE"LEFT$(A$(35),16)"S!":GOTO1050
950 IF D(B)=13THENIFI(25)=IIFI(24)=1QRI(24)=AQRI(24)=QANDA(24)=AIFE$="DON"K3=25:R$="LARGE INFLATED BALLOON":GOSUB1100:H$(25)=R$:BO=1:GOTO4900
```

The Apple version of "Miner" (January, 1981) has a bug in line 480. It should be replaced by the following lines:

```
480 IF NOT (X > 240 AND S = EP) THEN 490
482 GOSUB 830: IF Y > 32 THEN 650
484 GOTO 230
```



## BLANK CASSETTES



Premium quality digital recording tapes. These 5-screw cassettes come packaged in boxes of 10. They are offered in 10 and 20 minute lengths.

C-10 #15-C10 .....	\$6.95
C-20 #15-C20 .....	\$7.95

  
**TSE-HARDSIDE**  
 6 South St., Milford, NH 03055 (603) 673-5144  
 TOLL FREE OUT-OF-STATE 1-800-258-1790



# REVIEWS

## LORDS OF KARMA from Avalon Hill

"Lords of Karma" represents a new venture on the part of the venerable Avalon Hill Game Company. Although they have produced microcomputer games before, such as "Nukewar", "B-1 Nuclear Bomber", "Planet Miners", and "North Atlantic Convoy Raiders"; this time they have tackled a new genre: Adventure. It seems fitting that they should do so; their microcomputer games division is following the same pattern that the games division has established over the years — wargames and then a stab at fantasy games.

Avalon Hill virtually defined the art of wargaming, going into business in 1958 with the release of "Gettysburg" and "Tactics II". They pioneered the use of hexagonal patterns on game boards, and began publishing **The General**, a magazine devoted to wargaming and military history. In 1974, when TSR Games came out with "Dungeons and Dragons", a new type of game rose to popularity — the Fantasy Role-Playing Game (FRP).

While closely related to wargaming, FRPs differ in two respects: theme and execution. Most FRPs are set either in a world rife with magic and totally lacking in machine technology, or in the far future where technology and magic are virtually indistinguishable. In either case, they lack the historical accuracy that is demanded in and that defines wargames. Furthermore, in execution FRPs differ from wargames in that they are open-ended, i.e., there is no definite end to the game. An FRP can go on for months or even years with the same cast of characters who grow in power and ability. In FRPs the goal is survival, not victory.

There have been attempts to capture FRP games on microcomputers, principally in two forms: The graphics form pioneered by Automated Simulations ("Temple of Apschai", "Hellfire Warrior"), and the Adventure (Scott Adams' work, "Zork", "Original Adventure"). Neither approach truly captures the spirit of FRPs; each method has its advantages and disadvantages. The graphics or Apschai approach centers almost exclusively around combat and



survival, with little or no cerebral effort required on the part of the player. "Adventure", on the other hand, is far more mentally taxing, requiring the player to figure his way out of various predicaments and puzzles, but is visually quite dull.

With "Lords of Karma", Avalon Hill has tackled a fantasy theme for microcomputer games, using the Adventure form to do so. The result is a good Adventure, but a disappointment to those who expected more creativity and ingenuity from the country's premier adult games company. "Lords of Karma" is a high level Adventure, on a par with "Zork" or Scott Adams' work. The flavor of the program is fantasy with all the trappings...there are goblins with axes, slimy worms and giant spiders racing about on sticky webs. There is a princess who has been kidnapped by a scurvy varlet, and her father the king who wants her back. There are mazes and swamps, sewers and caverns...in short, just about anything you could want short of lakes of molten lava!

The twist, if you care to call it that, is that this Adventure is for the do-gooders of the world, the Paladins, the embodiments of the chivalric code. The whole point of the Adventure is to do good deeds. By so doing, you ensure yourself of a place in heaven, via the AH Express. The way that all this is carried out is in the scoring system — the more good that you do, the more "karma points" you accumulate — an ironic "westernization" of the concept of karma. There are all sorts of ways to get karma points — you can slay monsters, rescue the princess, donate to the charity of your choice...I wouldn't be

surprised if there was even a little old lady to escort across a busy street. If you prefer activities of the evil persuasion, then be prepared to evaluate your performance on a negative scale, for you will get -X karma points if you're nasty.

Winning the game is quite simple: When you reach a certain total of karma points (over 200) you will be elevated to heaven upon performance of certain acts. Should you get killed in your wanderings, don't despair, you will be promptly resurrected on a mountain top, empty-handed, of course. But the karma points accumulated in all previous incarnations during the playing of the game will still be there. Should your karma point total be in the negative range, be prepared to burn in purgatory prior to rebirth. If anything, this might be the principal failing of "Lords of Karma": You are always reborn. Thus, if you have the patience to keep plugging away, you can win in a single sitting. There is really no challenge to winning, then...just patience. Conceivably you could win by repeating a single action enough times, merely shuffling back and forth between the mountain peak and the chapel. But then all you hardy Adventurers would never stoop to that, would you?

There are no major differences between this Adventure and the vast majority of others as far as execution is concerned. Simple two-word commands are entered for actions. There is not the elaborate pedagogy of "Zork", although there is one option that is appearing more and more frequently in Adventures — you can GET or DROP more than one object at a time by specifying ALL. One drawback is that upon arriving at a new location you must specify an examination, otherwise you get only the most cursory description of the scene and no mention of possible exits.

All in all, "Lords of Karma" is a good Adventure. There are plenty of locations and a fairly wide variety of situations to face. The program is done in Machine Language and executes quickly. It is well-written and thematically consistent, and does not require any sort of esoteric knowledge on the part of the user. If you enjoy this type of computer game, then I would recommend "Lords of Karma"; it is really quite



enjoyable. However, if you are expecting more than a simple Adventure, then you may be disappointed. Avalon Hill has not broken any new ground, they have only covered known territory well.

Dave Albert

### POKER TOURNAMENT from Adventure International

There you sit, an icebox full of beer and scads of munchies, two brand new decks of Bicycle playing cards and enough chips to rebuild Reno...and the phone rings. You answer it and suddenly the vision of five-handed poker rears its head. Three more quick phone calls and you begin to wonder how two-handed poker would work. One more phone call and you resign yourself to an evening of watching "Love Boat" reruns.

If this scenario sounds vaguely familiar, then perhaps it's time to consider a poker alternative: "Poker Tournament" from the folks at Adventure International. For one, you never have to worry about no shows, and besides, "Slim", "Tex", "Bart", "Doc", and "The Kid" don't drink very much and they eat even less!

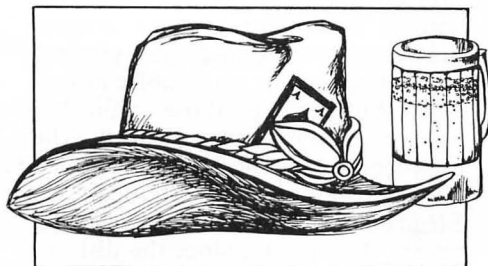
Slim and the boys are your taciturn companions for an evening of \$20-ante five-card draw poker. I say "taciturn" because Slim et al. don't have a lot to say...mostly they just curse their bad hands or say they'll see yours. Nevertheless, they are not bad playing buddies. For one, they never cheat, and they're terrible bluffers.

If one of the boys loses his shirt and all of his chips, he leaves the table and goes home, wherever that may be. (My theory is Redondo Beach, Calif., but that's only MY theory.) The table is then redrawn with one less player. The game continues until all but one player has been eliminated; that remaining player being then named champion with all due hoopla.

The only drawback to playing with these electronic fellows is that they only know how to play five card draw... one misses the bizarre variations of stud poker advocated by more human opponents. No wild cards, no high-low, no Cincinnati...nor do you get to take your winnings home, but then you don't drop a bundle either, so all in all the exchange seems fair. Certainly

"Poker Tournament" is a game for purists.

John Warshawer, the author of "Poker Tournament" has done a good job. He has chosen the Old West saloon for his thematic backdrop, as demonstrated by the names of the participants and their



vocabulary. The graphics display, while not on a level with programs like "Olympic Decathlon", is nonetheless quite adequate. Your cards are displayed numerically, while the table is drawn with the names of the surviving players in their respective chairs. The deal rotates around the table, as it should, and the betting protocol is correct.

All in all, a tidy little package that delivers just what it promises, and that isn't anything to complain about.

Dave Albert

### THRILY OF GAMES from On-Line Systems

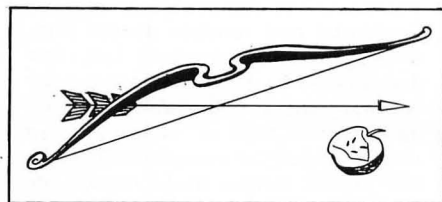
This "Thrily" of games by Warren Schwader includes two versions of popular arcade games and an original, and quite cute, archery game based on the William Tell legend. The games are written in Integer BASIC and Assembly Language and are all done with Hi-Res graphics. The games are:

1) "Smash Up" which is the perennial favorite "Collision" (or "Head On" or "Dodge" or whatever) in which you must race around a set of concentric squares, picking up dots worth a certain point value each, while the computer has a car going around in the opposite direction, intent on mayhem, destruction, and such. Your car can go faster than the computer's, but the computer car is quite astute and will make last-second lane changes in the pursuit of auto-destruction (pun intended). There are four skill levels in this game, ranging from "beginner" to "so you think you're good?" Oh yes, picking up all the dots results in a bonus condition giving you more points for additional dots, and add-

ing a car to your beginning allowance of five.

2) "Bustout" is another video favorite, a.k.a "Breakout", "Little Brickout", "Ricochet", etc. It's solitaire ping-pong, trying to break through a wall of bricks by keeping a ball going with a paddle on the screen. The beginner's version has part of the ball escape area sealed off, so it's easier to keep the ball on the screen, punching out those bricks.

3) "William Tell" is my favorite program of the three, although as a game it is readily mastered and then becomes somewhat dull. The gist of it is that you are William Tell, with five arrows to shoot. Your targets are three apples, two of which are on a tree and the third, (which you must hit three times for a perfect score) rests atop the head of your son, who is leaning against the aforementioned tree, contemplating the state of pin-cushionhood, I'm sure. You control your shooting by virtue of deciding how far back to pull the bowstring, and when to release the arrow.



Should you miss the apples in the tree, the little wisecrack leaning against it will question your marksmanship and competency quite readily, greatly increasing the temptation to "misjudge" and put an arrow into him, son or no. All in all, this is a nice little program, well-conceived and executed. The only drawback, as I mentioned before, is that there is not enough challenge in it to hold the player's interest for very long.

"Thrily" is on the whole a nice package, although there is nothing new or exciting in it. If you don't already own other versions of the games included in it, it is worth the price, as the games run well and are bug-free.

Dave Albert

### MICRO-PAINTER from DataSoft, Inc.

If you don't have a color TV or monitor hooked up to your Apple,

continued on next page



## INVENTORY'S

by Roger W. Robitaille, Sr. for...



**Inventory 'S'** is an exciting advance in small business software for the TRS-80. Its in-memory system of data storage solves the problems of both sequential and random access files, while providing extremely fast, random access to any record. Other advantages include the ability to use any combination of characters for stock number; an exceptionally flexible record format (field names are user definable); and the ability to store data to tape or disk and upgrade at any time. Up to 150 items can be stored per 16K of available memory, with stock number, description, cost, vendor, reorder, and profit data in each record. Use your present stock numbers (a sort function is included), unlike competing systems which force you to use a different "record number". User-definable screen and printer reports let you see just the data you need, when you need it.

**Inventory 'S'** can be used effectively with a 16K tape based system or a 48K disk and printer system.

S-80 16K Tape Level II #26-IVST... \$24.95

S-80 32K Disk #26-IVSD... \$39.95

With INVOICING on Disk

#26-IVSI... \$59.95



continued from previous page

forget about "Micro-Painter". But if you DO have color, you really ought to have this superb high-resolution graphics program. It's not actually a sketching or drawing program, but one which makes you feel like an artist as you paint or color in a previously-created line drawing. This you do by moving a cursor around the screen on the displayed picture, and filling in with your choice of colors. Rather like those paint-by-number kits, but without the numbers (or the messy paints!).

"Micro-Painter" is subtitled "An Electronic Coloring Book". That's an apt description, since the disk includes nine finely detailed, full screen Hi-Res pictures all ready for your artistic coloring-in. These range from still lifes and nature scenes to portraits of the distinguished Einstein and the heroic Micro-Man. You can also draw your own, using the simple 30-line Micro-Draw program listed in the documentation. (Why this wasn't put on the disk is something of a mystery to me.) For that matter, you can create a Hi-Res line drawing using ANY program or method that you choose, and then use "Micro-Painter" to color it in. In addition to a "Negative" command which transforms a white-line drawing into a black-line drawing (and reverses all other colors as well), there's a special "Fix It Up" command which helps prepare such drawings for painting.

By using two-color mixes of Apple's six Hi-Res colors, the program is able to produce 21 distinct colors for your artistic use. Any of these can be defined with two keystrokes (such as "BG" for blue-green or "VV" for solid violet), and then a press of the paddle button will fill in the part of the drawing in

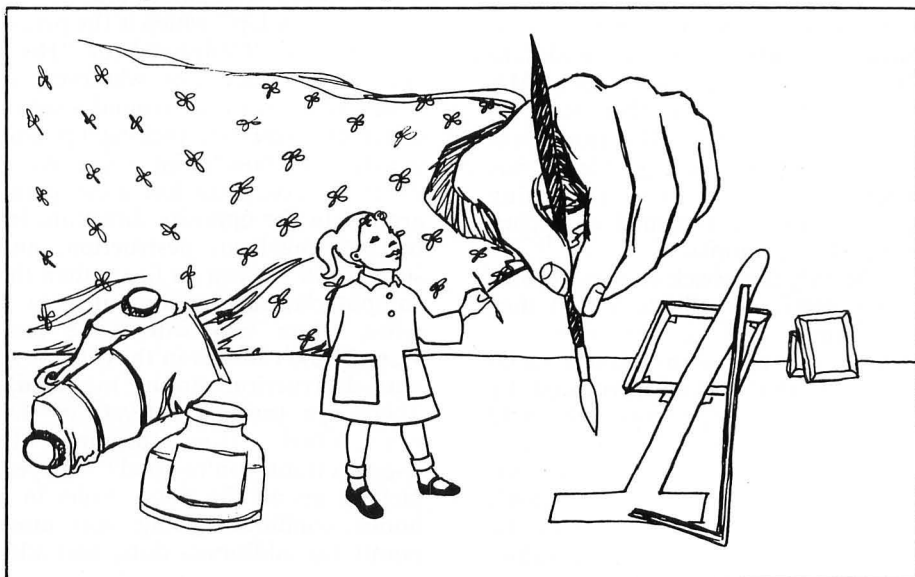
which the cursor is currently located. This is always an intriguing thing to watch, as the color spreads out in a diamond-shaped pattern to fill in every cranny up to a boundary line. The cursor is moved either with the two paddles or, for very precise step-by-step movements, with the I-J-K-M keys.

One of the most fascinating features of "Micro-Painter" is its magnified viewing and painting mode. Pressing the space bar at any time instantly zooms you in to a roughly 10-X microscopic view of the immediate vicinity of the cursor. The cursor then stays in the center of the screen, and the paddles or keys move the picture left, right, up, or down. Each individual "dot" of the picture (displayed in this mode as a rectangle) can then be viewed and colored at will, instead of filling in whole areas at once.

Other features allow saving and loading pictures at any stage of development, protection of black areas of the picture (normally the boundary lines) from being inadvertently colored over, and the ability to exercise DOS commands without having to exit the program.

The documentation of all aspects of the program is excellent. The 32-page booklet contains thorough, well-illustrated, and easy-to-understand instructions, plus print-outs of the nine pictures which are on the disk. You even get a nice colorful box, which contains not only the disk and booklet, but six crayons and a magnifying glass as well! Bob Bishop and DataSoft Inc. have put together a very attractive, entertaining, and useful graphics tool which gets my highest recommendation.

Jon Voskuil





# HARDWARE CORNER

by Edward E. Umlor

Well here it is time for the old GRANITE KNOGGIN to bend your eyeballs again. Where is everybody out there? I haven't received any mail at all. I know there are people with computers out there, but you would never know it by my mailbag. It was so lonely, that it ran off with a UPS bag, and didn't even leave a note for me. I would really like to be answering specific questions in this column. It is the only way of knowing that I am really helping people solve their computing problems.

Some people have been having problems with a modification called the CLOCK CONTROL BOARD. This is a modification that would help only those people with extremely long programs, or programs that require many complex math formulas to run. The installation should be done by a competent technician as there are etches to cut and quite a few wires to be soldered in.

The biggest problem seems to be broken wires. If the wires are not stripped properly, they are weakened and break very easily from any vibration. This can cause intermittent operation and a great deal of frustration. The second problem in line is — installation by a person who does not know how to use the tools of the trade. This has caused several logic boards to be scrapped, because too many etches, etc., have been damaged. There comes a point when it is cheaper to replace the logic board than try to resurrect it. The third most popular problem is miswiring the modification. This can cause a cascade effect among the chips on the board as one pops, then another. This not only makes the control board useless, but can also render the logic board something less than desirable. These can all be very costly to correct, and you will be without your system for some time. I would not recommend this speed-up mod unless you definitely need to increase the computer's operation by more than 100% (double the present speed).

The easiest and most reliable speed-up method I have heard of so far will result in a 100% increase

(takes 1/2 the time to run the same program). This is done by replacing the RAM chips with 200 ns chips, replacing the Z-80 with a Z-80A, and replacing the crystal in the keyboard with one of double the frequency. Unsolder the present 10.6445 MHz crystal and solder in a 21.2890 MHz crystal. Remove the Z-80 and insert the Z-80A in the same socket (be careful of static electricity). The RAM memory chips are also in sockets and can be replaced very easily (again watch out for static). I have heard of several people that have made this conversion with good results.

This is for all you Atari owners that want to use the Epson MX-80 or other eight-bit word printer with the MACROTRONICS interface cable. This cable works very well with the MX-70 or other seven-bit word printers, but not with eight-bit word printers. DO NOT DESPAIR — ALL IS NOT LOST! There is a simple fix that can be done to allow the MX-80 and others to work. Yes you will be giving up something on the MX-80. It is known as the block graphics (the upper end codes of the eight-bit word).

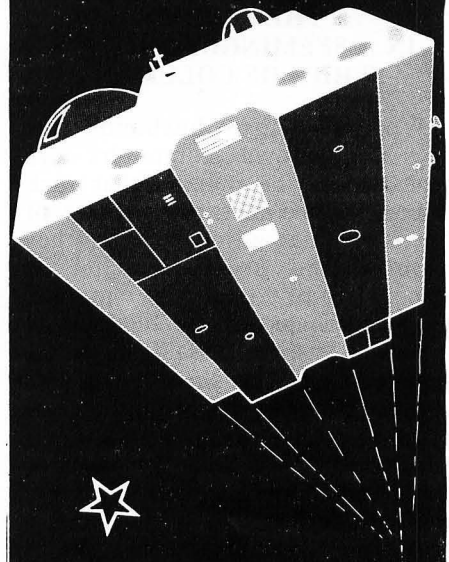
The fix can be done without any soldering. The connector used is a self-stripping type and cuts through the insulation as you press the wire in place. Open the 36-pin connector housing to expose the wire contact area. If you look at the face of the 36-pin connector you will be able to see the pin numbers. Cut a two inch piece of 26 guage (AWG) insulated solid wire. Press one end into pin 9 slot. Be sure to press it down fully into the slot. Press the other end into the slot exactly opposite pin 8. Check for continuity between the two pins and if everything is OK, reassemble the connector housing.

What we have done is to ground bit 8 on the connector. The MX-80 now considers the incoming word as the lower 128 ASCII codes. I am sure you tried the cable unmodified and found that the only thing it would print was graphics (the upper 128 ASCII codes). I hope that this has been of help to you.

Well, that's about all for now. Any questions...etc., please write to GRANITE KNOGGIN. I kinda miss my mailbag.



# INVASION



Now you can continue to play this popular arcade game even after you run out of quarters! Shoot down the invading aliens, but protect your guns from their bombs.

Great sound and graphics. (Sound requires external amplifier.)

S-80 16K Cassette #26-ALIN ..... \$9.95

S-80 32K Disk #26-ALID ..... \$14.95



From



**TSE-HARDSIDE**

6 South St. Milford, NH 03055 (603) 673-5144  
TOLL FREE OUT-OF-STATE 1-800-258-1790



# THE SENSUOUS PROGRAMMER

by "J"

## THE SECOND IN A SEEMINGLY ENDLESS SERIES OF COLUMNS

Women whose husbands (or boyfriends) own computers are generally of the opinion that such silicon-infested plastic boxes are of the feminine gender. Why they should hold to this belief is something of a mystery to me. Unless perhaps they're thinking of the way in which said husbands sometimes caress their keyboards and cuddle their minifloppies. Or maybe the occasional late-night rendezvous where man and machine exchange sweet nothings in a language that only they understand. Or, possibly, the continuous financial drain caused by the machine's unrelenting demands for soft things to ware, more and more exotic memories, and foreign language lessons. Still, these small things don't seem to justify the derogatory way in which some wives refer to their husband's computer as his "mistress".

As I tried to convey last month, a computer program is nothing more or less than a set of instructions which tells how to solve a problem in a logical way. And the computer her . . . er, ITself is nothing more or less than the collection of circuits and switches and accessory devices which follows these problem-solving instructions. So what rational reason is there to be jealous of such a piece of machinery?

I guess the first part of the answer to that is another question: "Since when does jealousy have to be rational?" And the second part of the answer is that certain kinds of problems can be so much fun to solve, and be so altogether engrossing in terms of time and concentration (and sometimes money), that there may well be ample justification for feelings of jealousy.

This ongoing column is written in the interest of communication — not only between human and machine, but between human and human. If you're a human interested in making your computer understand you better, you might find some helpful goodies here. If you're a human interested in understand-



ing a fellow-human who has a computer for a "mistress", you may get some insights along the way. If you're a computer interested in understanding your human better, and concerned with bug-free interfacing with your fellow-machine, . . . er, maybe you'd be better off reading **Byte**.

Communicating with computers used to be a tedious and highly-specialized process, involving much flipping of switches and constant watching of blinking lights. You still CAN communicate with computers like that, and I guess some primitive, Neanderthal types still do. Cartoons still picture computers in that way — either that, or in the other extreme of communication through conversational, spoken English a la Star Trek, Buck Rogers, et al. But most civilized people in the real world of 1981 communicate with computers through Pidgin English languages which fall somewhere between switch-flipping and street slang. And most of us micro-minded people (did I say that right?) deal almost exclusively with BASIC — the Beginner's All-purpose Symbolic Instruction Code. (If you take offense at the word "beginner" in the above acronym, you may want to think of it as the Boolean Algorithmic Simulation for Intelligent Computerists.)

Speaking of BASIC (ah, we finally get to the subject at hand) let's take a look at the ways in which this particular language allows humans and computers to communicate with each other. Communication, of course, is a two-way street. And computer-oriented types invariably refer to the two sides of that street as input and output. Input is what

you tell the computer, and output is what the computer tells you. There are different layers of input and output going on inside the computer all the time, but the layer we're concerned with here is what takes place during the running of a BASIC program that has been entered into the computer's memory. There are several ways available to the programmer to enable computer and human to interact, so let's take a look at some of them. This month we'll scrutinize input methods, and next month tackle output.

BASIC's most commonly used input instruction is INPUT (I wonder why they picked that word?). The statement

**INPUT A**

in a program causes the computer to do three things: (1) print a question mark and cursor on the screen, (2) wait — forever, if necessary — for someone to press RETURN or ENTER, and (3) store the number which was typed in before RETURN or ENTER in a memory location which it labels "A". INPUT itself doesn't do anything to the number except to store it; but from then on, it's available to any other part of the program that needs to use it.

INPUT isn't limited to handling numbers; it can handle non-numbers such as "ABRACADABRA", "R2D2", AND "#%&?/\*!" as well. The only thing is that you have to tell the computer whether to expect non-numbers as part of the input or not. If you don't, you will get an embarrassing error message at best, or at worst cause the program to croak prematurely in its prime of life. Telling the computer what kind of input to expect is as simple as using the right kind of variable name with the INPUT statement. Just add a dollar sign to the end of your variable name (e.g., A\$), and you can type in all the non-numerical (or, for that matter, numerical) garbage your heart desires.

Well, almost all. On most computers (Atari owners, please ignore), there are a few bits of garbage that an INPUT statement won't accept as characters in their own right. These are commas, colons, and leading quotation marks. When the computer finds either a comma or a colon in the string of characters that

was typed in before ENTER/RETURN, it acts just as though there were another ENTER/RETURN right there in the middle of the input string. So it takes the characters up to that point and stores them in its memory and labels that memory location A\$ (or whatever) — and then it blatantly ignores all the rest of your wonderful garbage, and tells you so by printing an unsightly “EXTRA IGNORED” message on your screen.

Now, you can take advantage of this peculiar treatment of commas and colons, if there happen to be several different things that you want to input all at the same time. If you want to have somebody type in his full name, and then store the first and last names in different locations, you can do so with a statement like

```
INPUT L$, F$
```

where L\$ will be the storage location for the last name and F\$ for the first name. Then, if the user types in

```
EINSTEIN, ALBERT
```

and hits RETURN/ENTER, A1's first and last names will be filed away just where you want them and no sloppy error messages will clutter up the screen. You can do the same with number inputs, or even an assortment of number and string inputs (e.g. INPUT A\$, B\$, X, Y) — as long as you know exactly how many inputs are expected, and don't try to input a string when a number is expected. Typing in too FEW entries in such a case will not generate an error message, but will clutter up the screen with another question mark or two as the computer prompts you for more input. Generally, multiple inputs clumped together like this are not a very good idea because of all the possibilities for confusion and errors on the part of the user; but occasionally they are helpful.

Quotation marks in an input string are another part of the story, and in fact may be used to defeat the function of the comma and colon. If you begin and (optionally) end your string of input garbage with a quotation mark, you can put anything you want, including commas, colons, and the kitchen sink, into the computer's memory by means of your INPUT statement. EXCEPT (there's always one in the crowd) for another quotation mark. So if you typed in

```
“EINSTEIN, ALBERT:  
PHYSICIST”
```

then the entire string of characters, except for the quotation marks themselves, would be treated as one

input string. Without the leading quotation mark, the input would be split into three parts and would require three variables in the INPUT statement to capture all three.

Incidentally, you can stick quotation marks themselves into the computer's memory using an INPUT, as long as some non-space character precedes the first quotation mark (e.g., Q“SD2”“”“\*”R). Then all quotes will be treated just like any other characters. But also note that beginning the input with a character other than a quotation mark will NOT then allow you to include commas and colons in the string. There are just some things in life that you can't have both ways, and with an INPUT statement that's one of them.

The Atari's INPUT statement, as noted above, works just a bit differently. Any characters, including commas, colons, and quotation marks, can be entered via INPUT using a string variable. Multiple variables using a single INPUT statement are still allowed; however, only when you are entering multiple NUMERICAL variables can you type them in separated by commas. If you're entering multiple STRING variables, each entry must be separated from the next with a RETURN, since commas and colons would simply be accepted as part of the input.

To add to the fun, S-80 Disk BASIC users also have another statement available to them, which functions as Atari's INPUT does when inputting a string. Its name is LINE INPUT, and it will accept all characters as part of the input line. It must be used with one and only one string (not numerical) variable name, such as

```
LINE INPUT A$  
and is terminated by pressing  
ENTER.
```

There will be a little more to say about INPUT next month, under the topic of output (ah, the subtle logic of it all). But first let's look at some other options that the BASIC programmer has for feeding stuff into a program. Here we have to get even more machine-specific. Apple, Atari, and (in Disk BASIC only) the S-80 all have the statement GET. What can be confusing is that all three use it in different ways. The S-80 uses it as a means of reading a record from a random-access disk file, which is not really germane to the present discussion. The Atari's GET is used to retrieve a single character from a file — a file which

might be on a disk, but which might also be (lo and behold) in the keyboard's input buffer. (That's the place where disreputable characters loiter around waiting to be used by the computer after being typed in.) And the Apple's GET is different yet, having the exclusive job of snatching a single character ONLY from the aforementioned keyboard buffer.

The Atari and Apple GETs, then, can be used to capture any character — one and only one — which is typed in at the keyboard. The statement is simplest on the Apple, where it takes either of the forms

```
GET A or GET A$
```

depending on whether you want to get a number or a non-numerical character. On the Atari, prior to using the GET you need to assign what's called an Input/Output Control Block (IOCB) to the keyboard. This done with a statement such as

```
OPEN #2,4,0,“K:”
```

which assigns IOCB #2 to the keyboard (K:). The number 4 specifies an input-only device, and the number 0 is just a filler for an unused “auxiliary code” specifier. Once this is done, then a statement such as

```
GET #2, A
```

will do the trick whenever you want to grab a character.

With both the Atari and the Apple, when a GET is encountered, the computer will print the cursor on the the screen and wait for you to press any key (NOT followed by RETURN). After you do that with the Atari, the variable you specified (A, in the above example) will be assigned the ATASCII value of the character you entered (0-255). The Apple, on the other hand, will assign to the variable the actual value of the character typed in: 0 through 9 for a numerical variable, or the appropriate one-character string for a string variable. Then, with both machines, the cursor will disappear and the program will go on to the next statement. Note that the character you typed will NOT be printed on the screen by the GET statement, unlike INPUT which displays characters as you type them.

The S-80 has a statement called INKEY\$ which, like Apple's and Atari's GET, allows you to capture a single character from the keyboard. But unlike said GET, INKEY\$ won't print the cursor on the screen, and it won't stand around waiting for you

continued on next page

continued from previous page

to tap a key. It whizzes by, glancing at the keyboard buffer on the run; and reports back to your program what it saw, even if it saw nothing at all. For this reason, INKEY\$ is frequently used in a loop that keeps on sending it back to look again until it finds something other than an empty room. For example:

```
10 A$ = INKEY$: IF A$ = ""
```

```
THEN 10
```

where there is no space between the quotation marks, indicating the "null" or empty string (i.e., no keypress).

You can do the same thing on the Apple that INKEY\$ does on the S-80, using a PEEK statement. (More on such voyeuristic goodies in general in a later article.) A PEEK into memory location -16384 inside the hidden recesses of the Apple (which you don't even have to lift the cover to do) will reveal a number equal to the ASCII value of the last key

struck, plus 128. If no character at all is in the buffer, then the number you find will be less than 128. As with INKEY\$, no cursor is printed on the screen when you do the PEEK, and there's no hanging around waiting for a character to show up. So again, this statement is usually found in a loop which keeps checking for a keypress (although such loops may well be quite a bit longer than the one-line examples given here). After finding a character in the buffer with this PEEK, it's good practice to clear the buffer with the statement POKE -16368,0.

The following four bits of coding, then, are pretty much equivalent, all of them used to capture a single character from the keyboard in A\$.

On the S-80:

```
10 A$ = INKEY$: IF A$ = ""
```

```
THEN 10
```

On the Apple:

```
10 GET A$
```

or


```
10 A = PEEK(-16384): IF A < 128  
THEN 10
```

```
20 POKE -16368,0: A$ =  
CHR$(A-128)
```

On the Atari:

```
10 GET #2,A: A$ = CHR$(A)
```

If all this seems a little confusing...well, it can be. But since most of us don't have more than one computer system, it's not all that bad once you get used to your own.

One final note on input before abandoning you for another month. YOUR input in response to this column is most welcome. Suggestions (keep them nice), arguments (keep them rational), additional observations (keep them simple so I can understand them), and contributions (cash, check, or money order) will all be graciously and gratefully received. But as far as other types of letters go — no, I'm sorry, I'm busy next Saturday night. 

continued from page 55

which may be added to, deleted from, or updated by your programs. This information is stored as files. Files may be composed of one or more data records and may be contained on any storage media such as magnetic tape or magnetic disk. In our Payroll application example, we may have one file which contains identification and control information relative to the Client (company). This file would only consist of one record, since there is only one company involved. We would also have a file of Employee Records. This file would have a separate record for each employee; hence, it will contain multiple records. Collectively, these files, as well as any other files that may be required, constitute our data base.

A Record is composed of data elements and our efforts now should be to ascertain which of the data elements are to be included in each of the Records. In addition to the input and output data elements required by the system, our System Design analysis may turn up the requirements for "intermediate" data elements. For example; in our Payroll application we most probably have identified input data elements such as "Salary" and "Hours Worked". Since the data element "Salary" might be passed directly to the output data element "Gross Wages", we would not require an intermediate element. However, we must have some method of converting "Hours Worked" into "Gross

Wages". To do this, we will probably be required to add the intermediate data element "Hourly Rate of Pay" if we do not already have it on our list of data elements.

After our list of input, intermediate, and output data elements has been completed through analysis of the elements (or at least as complete as we can visualize at this point), we are ready to start planning our Record Layouts.

A Record Layout is simply a "picture" of the conceptualized format of the records we will be using in our data base files. These completed Record Layouts; our list of data elements and such notes as may be required relative to the source, content, or use of the data elements (we will call this the Data Element Dictionary and discuss it at more length in Section III); and, our Flow Charts which we prepared in the Problem Definition phase and updated in the Systems Analysis phase constitutes our System Design documentation.

## PROGRAM DEVELOPMENT

Now is when it starts coming together. If you have done your planning well, the actual coding of programs can be fun and if it is not fun, get out of the business right now! You may be startled with an observation that I have made. That observation is that most of the programmers I have met in the last twenty years were not programmers at all. They were simply people working at programming. I didn't keep figures dur-

ing this time but I will venture a guess that the great majority of these people were in the business for one of two reasons; either they found a secure little niche in life that paid rather well or they were simply using it as a stepping stone to bigger and better things. For some it was a very slippery stone and for others the Peter Principle was appros. I think that the advent of the microcomputer will change this programmer profile dramatically.

Let's assume you do like to program. If you haven't yet learned, this booklet will not attempt to teach you. There are too many good books on that subject already. However, there are several rules to remember, whether you are coding for profit or for your own joy and amusement. These rules are not mandatory, your programs will run whether you use them or not, but if you employ them, you will find that the coding can be speeded up, the chance of error reduced, and you will have far less debugging to do.

Common Subroutines. During the System Design phase you must have noticed that at several points in your logic the same routine would be required. Examples of this would be Input/Output routines against the same data file or a routine to convert dates from computational mode (numeric) to display mode (string) or vice versa. These common routines may be required at several points in program coding. Therefore, to save coding and to reduce the chance of error, it would be prudent to write the

subroutine coding, debug it, and save it away for later inclusion in your program(s). Remember that every byte of program code is a potential error and to reduce coding is tantamount to reducing errors. Besides, multiple usage of common routines within the same program will reduce memory requirements.

**Simplicity.** For some reason, I guess it is human nature, programmers feel that unless the coding is esoteric they will be scoffed at by their peer group who might read the program. Bull! The name of the game is to write code that works and can be easily deciphered when modification are required. **KISS** (Keep It Simple, Sugar) is the byword.

**Use Copious Comments.** I learned something in pistol shooting many years ago — that the human mind can concentrate on a given object (in that case the front sight), to the exclusion of all other things, for only a few seconds. I don't care how brilliant you are (Mensa take note), you cannot retain the knowledge of where or why you wrote a particular set of code for very long. A few minutes of coding comments here will save countless hours later. Besides, we will be showing you how to take them out when they are no longer needed.

the more frustrating aspects of debugging a program is the "lost variable" problem. This is when you inspect the contents of a variable and say to yourself, "I wonder how that got there?". One of the prime faults of BASIC is the restriction placed on variable naming conventions. True, many versions allow the use of an eight-byte variable name but with only the first two bytes having precedence. I have always chosen to shy away from using these longer names because the inadvertant duplication of variable names is a trap of the first magnitude.

**Write Your Code.** I write out my code before I start keying it in. There are many who would disagree with this modus operandi as being unnecessarily time-consuming. I guess I don't have the retentive powers or native genius to code directly from the keyboard — or I am writing bigger programs that must interrelate with other programs — or I don't write "structured" programs. Whichever is my problem, I find it necessary to commit my code to paper before keying.

However, there are some definite benefits to the way I do business. First, I know where my GOTOs go

to. Second, and very important, I trap many little syntax errors and even some logic problems by being forced to take a second look.

**Keep a Notebook.** I guess I subscribe to most all of the popular publications devoted to the TRS-80 because that is my machine and I firmly believe that the world is made up of many programmers that are smarter than I am. Fortunately, a small amount of this intelligence creeps into these trade publications in the form of a few lines of code that are sheer genius. I pay a lot of attention to these coding examples and, when I find one of particular interest, I write a little routine which incorporates the idea, test it for authenticity (some publishers are notorious for typographical error), and incorporate it into my notebook.

**Consider the User.** While you are programming, think about the person who is going to have to operate your system. If you are going to write a line of code that will solicit a "YES" or "NO" response, add the extra code that will enable the program to ascertain the response from the first character of the input. I have read a lot of adverse comment about Microsoft BASIC not reinitializing the contents of the variable on an INPUT statement. I consider this a boon. For example, if I were to solicit a negative or affirmative response, my code might look something like this.

```
500 X$="Y" : INPUT"WANT  
TO CONTINUE (Y/N)";X$  
510 IF LEFT$(X$,1)="N"  
THEN STOP ELSE IF LEFT$(X$,1)  
<>"Y" THEN 500  
520 REM RESUME PROCESS  
ING WITH EITHER "Y" OR  
NULL RESPONSE
```

In this example, the variable X\$ is set to "Y" prior to the solicitation because "YES" is assumed to be the most frequently used alternative. Therefore, when responding to the question, the operator may either enter the character "Y", "YES", or simply depress the (ENTER) key. Line 510 will either direct the negative action or trap erroneous entries and cause resolicitation of the correct entry. Frankly, I detest programs that require me to enter a "1" for "YES" and a "2" for "NO" or some similar Level I convention. After paying the tariff for Level II, I want to be able to exercise its capability.

**Plan Ahead.** In the last paragraph, I made an innuendo about the restrictiveness of TRS-80 Level I BASIC. Now let me come right out and tell

you what is wrong with Level II. It lets you get away with murder in two areas: the first being the graphics characters and the second the IF-THEN-ELSE statement. This was fine as long as we didn't have any place else to go with our software, but look what happened when Radio Shack announced the Model II.

First, the ASCII codes for graphics characters have been severely restricted. Now, for those of us who disdain games, this no big thing. However, I do like to dress up my screen displays with some lines of graphics. For example, I was using the statement PRINT STRING\$(64,CHR\$(138)); to display a nice heavy line across the screen. This is not so good on the Model II so I have changed it to PRINT STRING\$(64,"="); so that we can all be compatible.

Second, Level II allowed us to be very sloppy with the IF-THEN-ELSE. You could get away with the statement IF A=B GOTO 500 ELSE GOTO 600. This statement makes the Model II regurgitate. Well, friends, the Model II BASIC dialect (at least in this case) is industry standard so you had better consider it if you intend to "up-line" your programs some day. Just for the record, the rule is that IF must be followed by THEN and may or may not be followed by ELSE. The GOTO is assumed in either case.

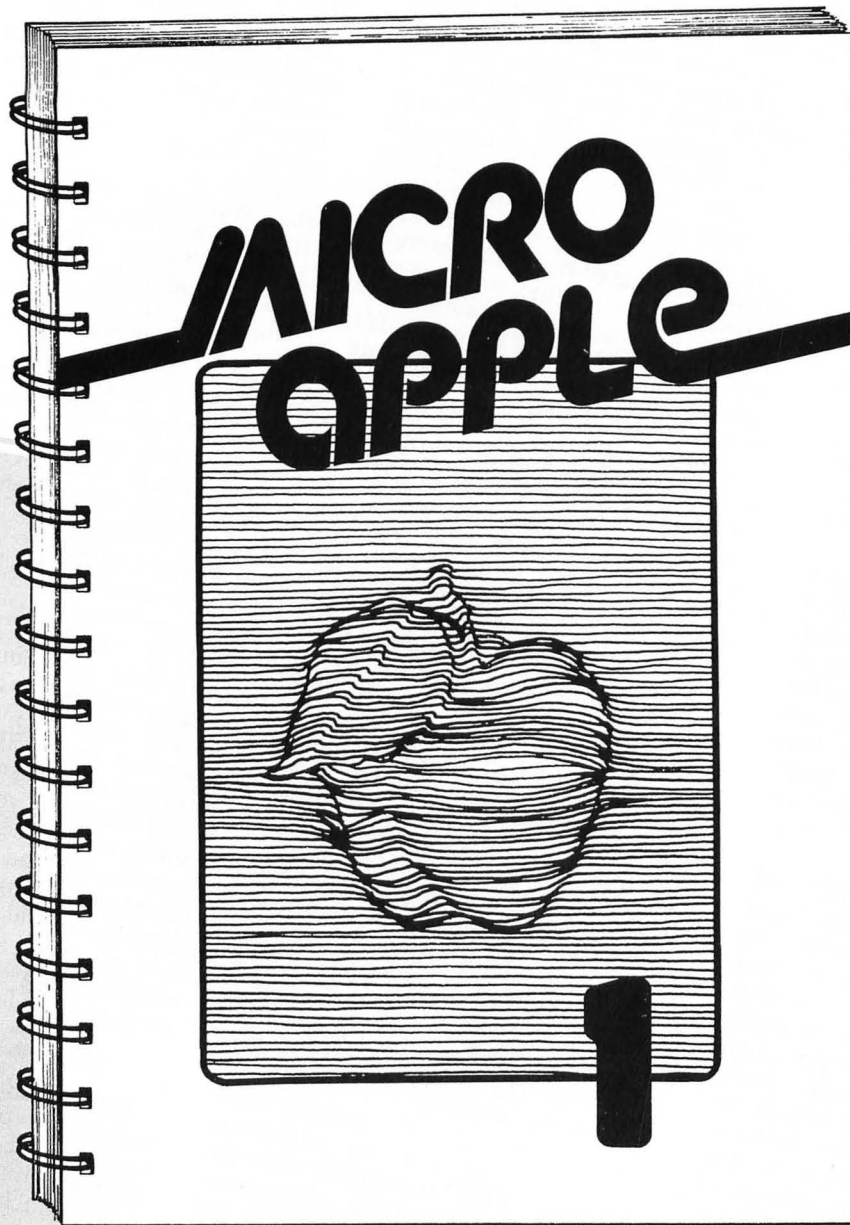
Of course, if you are like me, you violate all of these rules most of the time anyway and that is why we offer the Mod I to Mod II FILTER program at the end of this booklet. This is a program which can materially assist anyone who wishes to transfer programs up-line to the Mod II.

## UNIT/SYSTEM TESTING

Unit Testing is the testing of each individual program while system testing is a test of the data flow throughout the entire system. Normally, much of the unit testing can be performed as the program is being keyed in. Because of the interpretive nature of the BASIC language, a subroutine or a few contiguous lines of code can be tested as an entity. In fact, it is a good idea to always consider the order in which a program is to be keyed in — it will enable you to take full advantage of the "piecemeal" testing.

As I stated earlier, I commit my code to paper before keying. This enables me to key in any part of a program and perform a test on it before proceeding. In this manner, a

continued on page 87



**\$24.95  
With  
diskette!**

**MICRO/Apple 1**

This first volume in our new series contains 30 articles selected from MICRO, 1977-1980, updated by the authors or our staff. The MICRO staff has added introductory material and re-entered, listed, and tested the programs and put them on diskette.

Every user will want this highly practical work next to his Apple, with its chapters on BASIC Aids, Graphics, Education, Games, I/O Enhancements, Runtime Utilities, and References.

Get MICRO/Apple 1 at your local computer store.  
224-page book and diskette

\$24.95\*

**More Than 30 Programs on Diskette  
— For Less Than \$1.00 Apiece!**

**No Need to Type in  
Hundreds of Lines of Code!**

**MICRO™**  
P.O. Box 6502  
Chelmsford, MA 01824

\* On mail orders, add \$2.00 for surface  
or \$5.00 for air shipment.



program can be built in increments and tested in larger and larger segments until, when it is completely keyed in, it is virtually "bug-free". One of the shortcomings of TRS-80 BASIC is that it does not syntax check your instructions while they are being entered, and this method of keying and testing is an excellent way to overcome this shortcoming.

The delivery of "bug-free" software is the programmer's Great American Dream. The best way to come even close to this dream is to conceive and execute a test plan that will cause each of the lines of code to be executed at least twice. Yes, twice! This is because the language you may be using may have some idiosyncrasy that the second iteration of a logic path could cause inconsistent results. For example, the FOR-NEXT loop — does it exit on the FOR or the NEXT? Also, you could have overlooked some type of switch setting that would fail on the second iteration.

We have all heard of Murphy's Law which states that "anything that can go wrong, will". Well, when it comes to installing software, you will soon learn that Murphy was an optimist. Remember, if you don't adequately test and de-bug your software, the Client will do it for you — and usually at the oddest hour!

A good test plan is a combination of hard data and common sense. You might best construct a test plan by gathering all of the input documents that your system requires and making representative entries that will accommodate all of the input Data Elements contained thereon. (You did save those completed documents that you picked up at the Client's office six months ago, didn't you?) Do not test with just what you know to be valid Data Element values. For example, if a Data Element is titled "Unit Price" and it is assumed that a positive value is always entered by the Operator, then test with positive numbers (including extra decimal places), negative numbers, null values, and even alphabetic characters to see what will happen. Not only are we interested in insuring that data is processed correctly, but we must insure that, during day-to-day operation, the computer will never come to an undocumented "READY". You may as well patch your Client's operating system so that the "READY" message is followed by your home phone number — it will save the Client time in looking it up!

## DOCUMENTATION

The subject of Documentation will be treated in three parts; Internal Documentation, External Documentation, and Backup Documentation. Internal Documentation concerns those instructions to the operator that are intrinsic to the programs such as alternative responses to prompting messages displayed on the video monitor during operation. External Documentation is that documentation which is provided to the Client which explains the operation of the system and might also be used for sales promotion of other prospective Clients. Backup Documentation consists of the forms, notes, and listings used during systems development.

One of the popular trade publications recently published an impressive statement. It said, in substance, that one did not have to be a programmer to operate a computer system; provided that the system has been designed for the naive user. Internal Documentation should always consider that the user is naive. If this were not so, you would be out of business.

How can we best accommodate the naive user? Well, the method I advocate is that the user should need only know how to turn the system on and "boot" your application program. Ideally, in a multi-program system, this would be a "menu" program which would guide the operator to the function which is to be performed by simply selecting the applicable function code. Bear in mind that the more simple you make the system operate, in terms of meaningful prompting messages and responses, the less trouble you will have in installation and training and the less "nonprofit" calls you will receive subsequent to installation. Just a note about operator interaction with the prompting messages. Whenever you ask the operator for a response, trap out invalid response possibilities. Don't give the computer the chance to come to the "READY" simply because someone entered something unexpected.

In many cases, good Internal Documentation will preclude those "I-did-this; it-did-that; what-do-I-do-now?" calls. I say "many cases" because no matter how good your documentation might be, some people just disdain reading instructions. This brings up a point to remember, when you are installing your system, install your documentation. That is, make sure the operator knows of its

existence and how it is used. It will save you much grief later.

Internal Documentation should consist of the following documents as a minimum:

...Program "boot" sequence. These are the specific instructions for initializing the system and getting the first function menu on the screen. It should be tailored to the specific hardware and Operating System that the Client will be using and consists of each individual operation that must be performed down to the level of depressing the (ENTER) key or its equivalent.

...Backup Operations. A description of just what the operator must do to back up the program and data files to provide for emergency conditions such as hardware failure, a mutilated diskette, fire, flood, or acts of a disgruntled employee.

...Auxiliary Messages. If program prompts or information messages, such as error conditions, requiring further clarification than can be displayed on the video screen, these should be keyed by a message number to the Internal Documentation where they should be amplified to whatever extent necessary.

External Documentation is mandatory if you intend to sell your software outside your commuting area or even within it if you will not personally be installing the software. This type of documentation is often called a Management Overview and its purpose is to spell out just what the capabilities of the system are, including options on how it may be run. As a minimum, External Documentation should include:

...A System Overview. This is a narrative abstract of system capabilities.

...A Detailed System Description. An amplification of the System Overview, this section of the External Documentation should include a detailed description of everything the system will do, what it will not do (such as file size restrictions), and Input/Output examples.

...System Options. If applicable, the External Documentation should include all of the various alternatives open to the Client for running the system.

Backup Documentation will be

continued on next page

continued from previous page

discussed in depth further along in this booklet; however, its importance will be discussed here. Simply, it is this: Without adequate Backup Documentation, you will have an extremely difficult (if not impossible) time in maintaining a system or modifying it to some other Client's requirements. You may feel that the system you have created will stand for years as a testimonial to your genius. Bosh! About the only software that withstood the test of time was the Ten Commandments and that is because it was documented in stone and there was no competition between developers. If there is one thing you can count on about your masterpiece (other than it will not be bug-free) is that the user will require modifications and enhancements. Count on it.

There is also another very good reason to be meticulous about the preparation of Backup Documentation. You have labored hard and long to create this system and there is no sense of re-inventing the wheel every time you want to sell something similar to another Client. However, be aware of another truism — whatever you have already developed and running will not satisfy the next

prospective Client you show it to. It will require modification because "his business is different". Remember something else when dealing with a prospective Client. His business is different! It is because he runs it differently. The successful software developer does not profess to know how to run the prospective Client's business better than he does (although you might). If you persist in this attitude and keep failing to sell your services, then have your business cards printed to read "Business Management Consultant" and get out of the way of us tigers!

### SYSTEM IMPLEMENTATION AND MAINTENANCE

System Implementation (or Installation if you prefer) can be the most traumatic experience of the System Development cycle because this is where the phrase "Oh, I thought you said ..." is most frequently heard. You will find, on occasion, a built-in animosity among employees of the business who will be using your new system. This animosity is usually quite subtle and is sometimes manifested by an apparent nervousness on the part of the employee or an apparent inability of the employee to grasp the operation of the system which you have so meticulously documented. On the other hand, you will sometimes find employees who are not so subtle with their animosity. I've even heard employees say, "I don't care what the Boss says, this isn't the way we do business. He doesn't know what is going on."

This animosity, if found, is most probably fostered by articles that the employee has read about automation replacing them; or it could be that people are simply adverse to change. If your system is to affect one or more employees, then it would be prudent to discuss the ramifications of System Implementation during early discussions with your Client in the Problem Definition phase. If a problem is anticipated, it may be a good idea to prepare the employees for the change early in the System Analysis phase by having them contribute to this effort (you will be surprised what you can learn from them). Simply by letting them talk themselves out, and being a sympathetic listener, you will steal their thunder, or at least temper it, for what is to come later. Remember this

point — the Client (Boss) who states, "My employees will damn well do what I tell them to do", may be a tiger in front of you but a pussycat when it comes to those employees.

At any rate, the Implementation of an automated system should be well thought out prior to its execution. Put your thoughts on paper. Time-phase the implementation so that it does not cause a more-than-necessary disruption to normal office procedures. Remember, you are probably not being paid in full for your efforts until you have implemented the system, so your economic life depends on the success of this phase.

Maintenance of the system will probably start on Day One of Implementation. There will always be some little "glitch" to be fixed. However, the major System Maintenance effort will follow implementation. It seems that the enhancement of a system is in direct proportion to how good a job you did on the initial system. If you did a bang-up job of interpreting the Client's requirements and installed the system to the satisfaction of all, they will quickly find things for the system to do that were not in the original specifications. I once heard a computer professional state that he was considering giving away software, just so that he could do enhancements, and modifications — and charge for those! You can always determine the quality of your system by the speed with which the Client gets back to you with add-on work.

The System Maintenance phase is where your Backup Documentation becomes vital. Obviously, you have been doing other things in the interim between the Program Development phase and the Maintenance phase and how quickly we forget the nitty-gritty of what we have done. A directory of program routines and subroutines, a data element to variable name cross-reference, and detailed record layouts are "musts" for System Maintenance.

### SUMMARY OF SECTION I

Section I has been mostly theory. It has, however, provided a structured approach to Systems Development and pointed out some of the pitfalls that might be experienced. In the next section, we will take a more pragmatic approach and describe some specific tools that might be used for Systems Development.

Welcome to . . .

## THE RACES



**THEY'RE OFF!**

Eight horses surge down the track, straining for the lead, with your horse struggling in the pack.

They round the turn and head into the stretch. Your horse shoots from behind, catching the lead horse. They cross the finish line.

The Win, Place, and Show horse results are printed on the screen, along with each bettor's race winnings and total daily winnings.

You collect your winnings and decide if you want just to watch, or bet on the next race. You study the odds, place your bets, and select the track speed—fast (dry), average, or slow (wet).

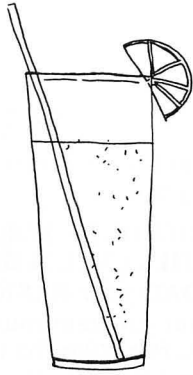
The horses are at the starting gate, jumping and snorting. You raise the gate, and the next race is underway.

Each horse gallops forward randomly. Spectators squirm and shout as they urge their horses to win.

You have all the track action and thrills. Plenty of winners—and losers! Now you can use your computer to find out what it takes to win at the track. Good Luck!

Requires 16K Send check, or charge it to Visa  
S-80 Tape—\$9.95 or MC. (Print charge number and  
S-80 Disk—\$14.95 expiration date—Phone 313-627-  
2877 for charge if you wish)

WE GIVE IMMEDIATE SERVICE!  
ECHO PRODUCTS INC., 335 MILL, ORTONVILLE  
MICH. 48462  
DEPT. S Dealer Inquires Invited



## SECTION II THE SYSTEM DESIGN MANUAL

Up to this point, we have discussed the "what", "where", "when" and "why" of Systems Development for microcomputer systems. Let's now turn to the "how" and provide some examples of documents which will materially assist your development efforts as well as serve as backup documentation when the system is complete and we enter the System Maintenance phase. I have attempted to minimize the number of forms, yet capture all of the pieces of information that will be required for both development and subsequent maintenance of the system. This is not to say that you will not have useful information on other documents, but these other documents are generally free-form, ranging from bond paper to the backs of envelopes and even an occasional brown paper lunch bag. The principle here is the same as my accountant keeps telling me; it really doesn't matter where you write it, just so long as you write it somewhere and don't lose it.

However, brown paper lunch bags are difficult to organize into meaningful files, so the five forms which will be described below are a suggested substitute. These forms serve as the basis for a System Design Manual. Not only does a manual of this nature materially assist the development of a system, but it will serve you extremely well throughout the life of the system. (That is the period of time when either you or someone else is actively using the system.) Using the forms we are about to describe, the System Design Manual would consist of the following:

...A System Description composed of completed Program Designed Description Forms.

...Backup Documentation ar-

ranged in five sections and referenced by the Program Design Descriptions.

- Section 1 contains Source Document Description Forms.

- Section 2 contains Video Display Format Forms.

- Section 3 contains Line Printer Format Forms.

- Section 4 contains File Layout Forms.

- Section 5 contains Program Flow Charts, the Data Element Dictionary, and Program Listings.

### PROGRAM DESIGN DESCRIPTION FORM

This form, which is shown in Figure 2-1, will provide you with a "picture" of the operational flow of the system. These forms are usually completed for each program in the system, but may also be completed for segments of programs where the additional detail would be required.

The heading of the form should contain the identification information necessary to relate the program

or program segment described on the form to its logical position in the system. This includes the name (or acronym) you have given the system, the program name (or acronym) and the module (or segment) name if further definition is required because the program is divided into logical modules.

As we said earlier, Program Design Descriptions should be prepared for each program, or program module, that is identified on your system flow chart that was prepared as part of your initial System Design (and probably updated many times since then). The "Synopsis" block of the form should provide a brief statement as to what this program/program module does and how it contributes to the overall system operation. As you can readily see, when you have completed the Program Design Description forms for the entire system, you will be able to discard that brown paper bag with all the erasures on it that has been so

continued on page 95

SOFTWARE DESIGN DESCRIPTION	SYSTEM	PROGRAM	MODULE
<b>SYNOPSIS:</b>			
<pre> graph TD     subgraph Inputs         I1[I/P DOC (Fmt 1)]         I2[I/P DOC (Fmt 2)]         I3[I/P DOC (Fmt 3)]     end     subgraph Outputs         O1[O/P DOC (Fmt 1)]         O2[O/P DOC (Fmt 2)]         O3[O/P DOC (Fmt 3)]     end     subgraph Storage         D0[DISK UNIT #0]         D1[DISK UNIT #1]         D2[DISK UNIT #2]         D3[DISK UNIT #3]     end     subgraph Other         T1[TAPE I/O #1]         T2[TAPE I/O #2]         CRT[CRT]         PROC[PROCESS]     end     I1 --&gt; PROC     I2 --&gt; PROC     I3 --&gt; PROC     T1 --&gt; PROC     T2 --&gt; PROC     PROC --&gt; O1     PROC --&gt; O2     PROC --&gt; O3     PROC --&gt; D0     PROC --&gt; D1     PROC --&gt; D2     PROC --&gt; D3     CRT --- PROC     </pre>			
<p>* FILE TYPE: P-Program; D-Permanent Data File; T-Temporary Data File; I-Interface</p>			

Figure 2-1



# ANALLIST

by Rod Fitzgerald

**"Anallist" is an S-80 utility for creating structured program listings and requires 8K.**

One of the advantages of Level II and Disk BASIC is their ability to process program code with more than one statement per line. This allows more versatility in writing programs, and can also save memory space. Even if a program is written with single statement lines, there are now programs available to "pack" your BASIC program into a multi-statement format.

But this very ability also gives rise to one of the disadvantages of BASIC. With as many as 15 or 20 statements crammed into one line, the "flow" of the program can become very hard to follow. (BASIC is not exactly a "structured" language to begin with!) This is especially true if there are "nested" IF-THEN-ELSE statements in a line. Trying to discern where a program is branching from a nested IF-THEN-ELSE can be very frustrating.

"Anallist" was written to alleviate this problem. It breaks down multi-statement lines into individual instructions, and "structures" nested IF-THEN-ELSE statements. See Figure 1 for an example of a section of a program before and after "Anallist". The output is directed to

a printer, and shown on the monitor as well. "Anallist" prints headings and page numbers automatically, and allows programmable skips to the top of the next page so that you can easily divide the program into sections (e.g., initialization, main body, subroutines, data, etc.).

There are certain steps that must be taken beforehand to fully utilize "Anallist". First, go through the program that you want to analyze, and make sure that every IF statement has a corresponding THEN. For example, a statement such as

```
IF Z = 10 GOTO 350
```

must be changed to

```
IF Z = 10 THEN GOTO 350
```

in order to be analyzed properly. Second, if programmed skips to top-of-page are desired, enter a line at the appropriate point consisting of a line number, a remark (REM or ') and a slash (/). Third, if for any reason you do not want a line analyzed (that is, you want it listed normally), enter a colon as the first character in that line. For instance, you may want to do this with lines where many variables are being initialized.

If these steps are followed, you should get a properly structured listing. Incidentally, if your printer can handle S-80 graphics, then they will be printed in the listing also. This

is useful in finding errors in a line of "packed graphics."

## PROCEDURE FOR USE WITH LEVEL II BASIC

1. CLOAD your BASIC program. Note: Your program must not have line numbers greater than 59999.

2. Enter "PRINT PEEK(16633)". If the number printed happens to be less than 2, you will get an error message when "Anallist" is run. If this is the case, just add two characters anywhere in your program and start again with this step.

3. Enter "PRINT PEEK(16548)". We'll refer to the number printed as A.

4. Enter "PRINT PEEK(16549)". We'll refer to this number as B.

5. Enter "POKE 16548, PEEK(16633)-2".

6. Enter "POKE 16549, PEEK(16634)".

7. CLOAD "Anallist".

8. Enter "POKE 16548, A (from step 3).

9. Enter "POKE 16549, B (from step 4).

10. Enter "RUN 60000".

## PROCEDURE FOR USE WITH DISK BASIC.

1. Save "Anallist" to disk in ASCII format (e.g., SAVE "ANALLIST/TXT",A).

FIGURE 1

### Before "Anallist"

```

3050 G=G+1:W=W+1:IF W*2>G THEN B=B+1
3100 IF H=0 AND M<=K THEN B=B+1
3200 IF M*2<K THEN B=B+1
3300 GOTO 5000
4000 PRINT @896,CHR$(30):PRINT @896,"YOU LOSE";
4100 G=G+1:IF W*2<G THEN B=B-1
4500 IF K*2>B AND M<K AND H<2 THEN B=B+1 ELSE IF B=0 THEN GOTO 9100
5000 IF M>1 THEN PRINT " IN";M;"MOVES.":GOTO 5005
5001 PRINT " IN";M;"MOVE."
5005 PRINT@960,"NO. BATS:";B;
5010 L=G-W
5110 PRINT "   GAMES:";G;"   WON/LOST:";W;"/";L;
6000 IF G>=100 THEN GOTO 9100 ELSE GOTO 1000

```

### After "Anallist"

```

3050  G = G + 1
:     W = W + 1
:     IF  W * 2 > G THEN
:         B = B + 1
3100  IF  H = 0 AND M < = K THEN

```

```

:         B = B + 1
3200  IF  M * 2 < K THEN
:         B = B + 1
3300  GOTO 5000
4000  PRINT @896,CHR$(30)
:     PRINT @896,"YOU LOSE";
4100  G = G + 1
:     IF  W * 2 < G THEN
:         B = B - 1
4500  IF  K * 2 > B AND M < K AND H < 2 THEN
:         B = B + 1
:     ELSE IF  B = 0 THEN
:         GOTO 9100
5000  IF  M > 1 THEN
:     PRINT " IN";M;"MOVES."
:     GOTO 5005
5001  PRINT " IN";M;"MOVE."
5005  PRINT @960,"NO. BATS:";B;
5010  L = G - W
5110  PRINT "   GAMES:";G;"   WON/LOST:";W;"/";L;
6000  IF  G > = 100 THEN
:     GOTO 9100
:     ELSE GOTO 1000

```

2. Load your BASIC program.  
 Note: Your program must not have line numbers greater than 59999.

3. Enter 'MERGE  
 "ANALLIST/TXT"'.  
 4. Enter "RUN 60000".

Since "Anallist" must examine every program byte looking for tokens, it's not terribly speedy. A 12K program, for example, will usually take about 30 minutes to print. However, the lack of speed is more than compensated for by the clarity and structure of the resulting listing. You may find it habit-forming to go to your "Anallist"!

### VARIABLES

C1: Length of remaining portion of program line.

CI: Subscript for If/Else array.  
 E!: Memory location of first byte in the current line.  
 F\$: Array of BASIC tokens.  
 FI: Array of If/Else print positions.  
 FO: "On" token flag.  
 G: For/Next loop counter.  
 G!: Memory location of current character/token in current line.  
 HS: Temporary storage of data items.  
 I\$: Name of file on disk/tape.  
 IN: Indentation from left margin.  
 L1: If/Else array subscript.  
 LP: Length of listing title.  
 N2!: Current line number.  
 P\$: Listing title.  
 P1: Indentation for listing title.  
 P9: For/Next loop counter.  
 Q: Page number.

QF: Quote flag.  
 R: Line counter.  
 RF: Remark flag.  
 S: Decimal value of current character.  
 SS: String value of current character.  
 S1: Decimal value of next character.  
 S9!: Memory location of the start of next program line.  
 T\$: Reformatted portion of current line.  
 T1\$: Remaining portion of current line.  
 X1\$: Constant - value ":".  
 X2\$: Constant - value "+".  
 X3\$: Constant - value "page".  
 X4\$: Page number string.  
 XX: End of program flag.  
 Y\$: Miscellaneous input string.

<pre> 60000 GOT060250 Subroutine to skip to top of page.  60050 FORP9=1T066-R 60060 IFPEEK(14312)=63THENPOKE14312,10ELSE60060 60070 IFP&lt;4THENPRINT 60080 NEXTP9:IFXX=1THENPOKE16425,0:END 60090 R=0:RETURN  Subroutine to print page headings.  60110 T\$=STRING\$(P1,32)+P\$:GOSUB60150:Q=Q+1:X4\$=X3\$+STR\$(Q):T\$=I \$+STRING\$(64-LEN(I\$)-LEN(X4\$),32)+X4\$:GOSUB60150:T\$=STRING\$(64, )=":GOSUB60150:RETURN  Subroutine to get a line of text (T\$).  60130 C1=LEN(T\$):IFC1&gt;64THENC1=C1-64:T1\$=RIGHT\$(T\$,C1):T\$=LEFT\$( T\$,64):GOSUB60150:T\$=X2\$+STRING\$(IN,32)+T1\$:RETURN  Subroutine to print T\$, increment line count, skip to top of page, and print headings.  60150 R=R+1:LPRINTT\$:PRINTT\$;:IFLEN(T\$)&lt;64THENPRINT 60160 T\$="":IFR&gt;59THENGOSUB60050:GOSUB60110 60170 RETURN  Point at return position.  60190 L1=13 60200 L1=L1-1:IFL1&gt;1ANDFI(L1)=0THEN60200ELSERETURN  Initialization.  60250 CLEAR1000:DEFINTA-Z 60260 E!=PEEK(16548)+256*PEEK(16549):IFE!&gt;32767THENE!=E!-65536 60270 :X1\$="":X2\$="":X3\$="PAGE"  60280 DIMF\$(122),FI(12):FI(1)=9:POKE16553,255:RESTORE 60290 READH\$:IFH\$&lt;&gt;"END" THEN60290 60300 F\$(0)=H\$:FORG=1T0122:READF\$(G):NEXT:IFF\$(122)&lt;&gt;"MID\$"THENC LS:PRINT"INCORRECT DATA ITEMS IN ANALLIST'S ARRAY":END 60310 CLS:INPUT"USUAL NAME OF FILE TO BE LISTED ";I\$ 60320 INPUT"LISTING TITLE (OPTIONAL) ";P\$:IFP\$=""THENP\$= "\$\$\$" 60330 LP=LEN(P\$):IFLP&lt;64THENP1=(64-LP)/2 60340 IFPEEK(14312)&lt;&gt;63THENINPUT"PRINTER READIED? (HIT ENTER) ";Y\$:GOTO60340 60350 PRINT:PRINT:GOSUB60110 </pre>	<pre> Beginning of main program loop.  60400 S9!=PEEK(E!)+256*PEEK(E!+1):IFS9!&gt;32767THENS9!=S9!-65536 60410 N2!=PEEK(E!+2)+256*PEEK(E!+3):IFN2!=60000THENXX=1:GOSUB600 50 60420 G!=E!+4:IN=6:RF=0:QF=0:CI=0:FORL1=2T012:FI(L1)=0:NEXT 60430 N2\$=STR\$(N2!):T\$=N2\$+STRING\$(8-LEN(N2\$),32)  Print the characters or tokens in the program line.  60450 S\$=CHR\$(PEEK(G!)):S=PEEK(G!):IFG!+1&lt;S9!-2THENS1=PEEK(G!+1) ELSESS=0 60460 IFS&lt;32ORS&gt;95ANDS&lt;128ORS&gt;250THEN60660ELSEIFRF=1ANDS=213TH ENT\$=T\$+CHR\$(G1):GOTO60660:ELSEIFQF=1ANDS&lt;&gt;34ANDS&gt;31THENT\$=T\$+S\$ :GOTO60660ELSEIFRF=1ANDLEN(T\$)=9ANDS=47THENT\$=T\$+S\$:GOSUB60150:G OSUB60050:GOSUB60110:GOTO60660  Set quote flag or remark flag if appropriate.  60480 IFS=34THENQF=1-QF:GOTO606590 60490 IFS=58ANDS1=147THENRF=1:IFLEN(T\$)&lt;9THEN60670ELSEGOSUB60130 :T\$=X1\$+STRING\$(IN,32):GOTO60670  60500 IFS=58ANDLEN(T\$)&lt;9THENT\$=T\$+"":RF=1:GOTO60670  60510 IFS=147ANDQF=0THENRF=1 60520 IFS=32ANDRF=0ANDQF=0THEN60660ELSEIFS&lt;&gt;58THEN60590  Print colon (S = 58).  60540 IFRF=0ANDQF=0ANDS1=58THEN60660 60550 IFS1=149THENGOSUB60130:GOSUB60190:T\$=X1\$+STRING\$(FI(L1)-3, 32):IN=FI(L1)+2:IFL1&gt;1THENFI(L1)=0:GOTO60660ELSE60660 60560 IFQF+RF=0THENGOSUB60130:T\$=X1\$+STRING\$(IN,32)ELSET\$=T\$+S\$ 60570 GOT060660  Print alphanumeric characters (32 - 95).  60590 IFS&lt;128THENT\$=T\$+S\$:GOTO60660  Print token or graphic characters (127 - 255).  60610 IFF0=1AND(S=141ORS=145)THENT\$=T\$+" ":FO=0 60620 IFS&lt;&gt;143ANDS&lt;&gt;149ANDS&lt;&gt;161ANDS&lt;&gt;202THENT\$=T\$+F\$(S-128):GOT 060660ELSEIFS=143THENC1=CI+1:FI(C1)=LEN(T\$)+1:T\$=T\$+F\$(15):IN=IN +5ELSEIFS=202THENT\$=T\$+F\$(74):GOSUB60130:T\$=X1\$+STRING\$(IN,32):G 0T060640ELSEIFS=161THENT\$=T\$+F\$(33):FO=1 60630 IFS=149THENIFS1=143THENT\$=T\$+F\$(21):GOSUB60130:IN=IN-5:T\$= X1\$+STRING\$(IN,32):GOTO60660ELSET\$=T\$+F\$(21):IFL1&gt;1ANDL1&lt;13THENF I(L1)=0 60640 IFS1&gt;48ANDS1&lt;58THENT\$=T\$+F\$(13)  Print full or partial line if ready, else go back. </pre>
---	--

continued on page 95



# WORD-SEARCH PUZZLE GENERATOR

by David W. Durkee (S-80 and Atari translations by Jon Voskuil)

**“Word-Search Puzzle Generator” is for an Apple, Atari, or S-80 with at least 8K RAM and a printer.**

If you’ve been wondering about the word-search puzzles that have appeared in the last three issues of **SoftSide** (including this one), here is the clever program that’s been responsible for generating them. With the computer’s help, an imbecile (yea, even a **SoftSide** editor) can put together one of these entertaining goodies, in little more than the time it takes to think up the words you want included.

Upon **RUNning** the program, you’ll be given the option of seeing the puzzle as it’s being created on the screen, or leaving the screen blank so that you yourself can enjoy working the puzzle later. Then you simply proceed to type in words to your heart’s content. The computer does the rest, placing the words in random orientations in the letter matrix.

The matrix size varies with the three versions of the program: 40 x 20 in the original Apple version, 37 x 20 in the Atari version, and 32 x 14 in the S-80 version. These dimensions are tailored to the screen display; if you’re concerned only with the print-

out, you could easily enlarge the matrix by changing the **DIMension** statement and the various loops which use those dimensions. For that matter, if you want a smaller, simpler puzzle, you could shrink the size as well. An interesting modification would be to allow the user to choose the puzzle dimensions each time the program is run.

After typing in all the words to be included, entering the word “**STOP**” will cue the computer that you’re finished, and it will proceed to generate an answer key, the completed puzzle with random letters filled in, and a list of all the words entered. (If you should happen to want the word “stop” as one of the words in the puzzle, as we did with the **BASIC Keywords** puzzle, just type it in backwards: “pots”.) You are then given the option of printing out another copy of the puzzle and word list. After you have all the copies you want, the program ends.

Besides the obvious entertainment value of the computer-generated puzzles, there’s also a great potential for educational applications. It’s an easy way to become familiar with a list of words — for spelling, vocabulary, geography, the sciences, almost any subject area. And in place of a word list, a clue list could be pro-

vided (as with crossword puzzles), to help learn the meaning of the words as well. Let your creativity take over!

## VARIABLES

**A%(\*,\*)** or **A(\*,\*)**: Array which stores ASCII value of letters in matrix.

**A\$**: Input variable. Also used in Atari version to assemble each line of puzzle to be printed out.

**B**: Counts directions that word may go in.

**B%(\*,\*)** or **B(\*,\*)**: Notes which directions the computer may write a word, given a random starting position. If **B%(X+2,Y+2)** is 1 then it is possible to write in that direction; if it is 2 or more, then the word shares one or more letters with other words if written in that direction.

**C**: Loop counter.

**D, R**: Used to select best direction in **B%** matrix.

**L, U**: Random starting coordinates for word.

**X, Y**: Indicate word direction along x and y axes; values can be -1, 0, or 1 (but not both 0), defining the eight different directions.

**X1, Y1**: Printing coordinates for individual letters; derived from **U, L** and **X, Y**.

### S-80 Version

Instructions and initialization.

```
5 CLEAR 500
10 CLS: PRINT@76, CHR$(23);"WORD SEARCH PUZZLE": PRINT@396, "BY
DAVID W. DURKEE": PRINT@518, "TRANSLATED BY JON VOSKUIL": PRINT@
716, "COPYRIGHT (C) 1981"
20 FOR I=1 TO 1000: NEXT I
40 CLS: PRINT CHR$(23);"TO CREATE A PUZZLE, SIMPLY TYPE IN A WOR
D AFTER THE '?' PROMPT, AND PRESS 'ENTER'. WHEN YOU'RE FINISHED
, ENTER 'STOP' AS YOUR LAST WORD, AND THE COMPUTER WILL DO THE R
EST."
50 PRINT:PRINT"PLEASE CHOOSE:": PRINT" 1 - FOR NORMAL DISPLAY D
URING ENTRY": PRINT" 2 - FOR BLANK DISPLAY (SO THAT
YOU CAN'T SEE THE PUZZLE)"
55 K$=INKEY$: K=VAL(K$): IF K<1 OR K>2 THEN 55
60 DEFINT A,B: DIM M$(200), A(32,14), B(3,3)
80 CLS:PRINT CHR$(23);
```

Beginning of word-entry loop.

```
90 Z=Z+1
92 PRINT@ 896, "WORD #";Z;: INPUT A$: IF A$="" THEN 92
95 M$(Z)=A$
100 IF A$="STOP" THEN 500
```

Choose random starting position.

```
120 U=RND(15): L=RND(32)
```

Check each direction to see if word may be written in that direction.

```
160 FOR X=-1 TO 1: FOR Y=-1 TO 1
170 IF X=0 AND Y=0 THEN 270
180 X1=L: Y1=U
190 FOR C=1 TO LEN(A$)
200 X1=X1+X: Y1=Y1+Y
210 IF X1>32 OR X1<1 OR Y1>14 OR Y1<1 THEN B(X+2,Y+2)=0: GOTO 27
0
220 IF A(X1,Y1)=0 THEN 250
230 IF A(X1,Y1)<>ASC(MID$(A$,C,1)) THEN B(X+2,Y+2)=0: GOTO 270
240 B(X+2,Y+2)= B(X+2,Y+2)+1
250 NEXT C
260 B(X+2,Y+2)=B(X+2,Y+2)+1: B=B+1
270 NEXT Y,X
280 IF B=0 THEN 120
```

Select direction to write word; if possible, choose one which will intersect another word.

```
310 R=2: D=2
```

```

320 FOR X=1 TO 3: FOR Y=1 TO 3
330 IF B(X,Y)>B(R,D) THEN R=X:D=Y
340 NEXT Y,X
350 X=R-2: Y=D-2
360 IF X=-1 AND Y=-1 AND B(1,1)=1 THEN 380
370 GOTO 400
380 X=RND(3)-2: Y=RND(3)-2
390 IF (X=0 AND Y=0) OR B(X+2,Y+2)=0 THEN 380
400 X1=L: Y1=U

```

Print word on screen (unless blank screen was chosen).

```

420 FOR C=1 TO LEN(A$)
430 X1=X1+X: Y1=Y1+Y
440 A(X1,Y1)=ASC(MID$(A$,C,1))
445 IF K=2 THEN 460
450 PRINT@ (Y1-1)*64 + (X1-1)*2, CHR$(A(X1,Y1));
460 NEXT C
470 B=0: FOR X=1 TO 3: FOR Y=1 TO 3: B(X,Y)=0: NEXT Y,X
480 PRINT@896, STRING$(32,32); GOTO 90

```

Prepare answer key.

```

500 FOR X=1 TO 32: FOR Y=1 TO 14
510 IF A(X,Y)<>0 THEN 530
520 A(X,Y)=45: PRINT@ (Y-1)*64 + (X-1)*2, "-";
530 NEXT Y,X
540 PRINT@896,""; INPUT"POSITION PAPER AND HIT ENTER";K$
547 GOSUB 670
550 LPRINT " ":LPRINT" WORD PUZZLE ANSWER KEY"
560 FOR I=1 TO 31: LPRINT " ":NEXT I

```

Fill in blanks with random letters.

```

570 PRINT@896, "PLEASE WAIT A FEW MOMENTS. . . ";
590 FOR X=1 TO 32: FOR Y=1 TO 14
600 IF A(X,Y)<>45 THEN 620
610 B=RND(26)+64: A(X,Y)=B
620 NEXT Y,X
630 GOSUB 670
640 LPRINT " ": LPRINT" COMPUTER-GENERATED": LPRINT" WORD-S
EARCH PUZZLE"
650 FOR I=1 TO 31: LPRINT " ": NEXT I: GOTO 720

```

Subroutine to print complete puzzle.

```

670 LPRINT " "
680 FOR X=1 TO 32: FOR Y=1 TO 14
690 LPRINT CHR$(A(X,Y));" ";
700 NEXT Y: LPRINT: NEXT X
710 RETURN

```

Print out word list.

```

720 LPRINT " ":LPRINT" WORD LIST": LPRINT " "
730 FOR I=1 TO Z-1: LPRINT W$(I): NEXT I

```

Another copy? If not, then end.

```

760 PRINT@ 896, STRING$(32,32); PRINT@ 896,""; INPUT "WOULD YOU
LIKE ANOTHER COPY";K$: IF LEFT$(K$,1)="N" THEN END
770 PRINT@ 896,""; INPUT"ADVANCE PAPER AND HIT ENTER";K$: GOTO
630

```

## Atari Version Documentation

Lines 90-165: Initialization and instructions.

Lines 170-200: Beginning of word-entry loop.

Line 210: Choose random starting position.

Lines 220-340: Check each direction to see if word may be written in that direction.

Lines 350-440: Select direction to write word; if possible, choose one which will intersect another word.

Lines 450-520: Print word on screen (unless blank screen was chosen).

Lines 530-590: Prepare answer key.

Lines 600-670: Fill in blanks with random letters.

Lines 680-720: Subroutine to print complete puzzle.

Lines 730-746: Print out word list.

Lines 760-770: Another copy? If not, then end.

```

90 POKE 752,1
100 PRINT " ":POSITION 10,5:PRINT "WORD SEARCH PUZZLE"
105 POSITION 10,8:PRINT "BY DAVID W. DURKEE":POSITION 10,10:PRINT "COPYRIGHT (C) 1981"
107 POSITION 7,13:PRINT "TRANSLATED BY JON VOSKUIL"
110 FOR I=1 TO 2000:NEXT I
120 PRINT " ":POSITION 2,5:PRINT "TO C

```

REATE A PUZZLE, SIMPLY ENTER A WORD WHICH YOU WOULD LIKE TO HAVE IN"

125 PRINT "THE PUZZLE AFTER THE '?' PROMPT."

130 PRINT :PRINT "WHEN YOU'VE ENTERED ALL THE WORDS YOU WOULD LIKE IN THE PUZZLE, ENTER THE"

135 PRINT "WORD 'STOP' AND THE ATARI WILL DO THE REST."

140 PRINT :PRINT "IF YOU WOULD LIKE A PUZZLE FOR YOUR- SELF (BLANK SCREEN), THEN TYPE '1';"

145 PRINT "OTHERWISE, TYPE '0' TO BEGIN: ";

150 INPUT BLANK:PRINT " ":Z=0

160 DIM W\$(1000),A\$(41),B(3,3),A(37,20)

165 W\$="":FOR I=1 TO 37:FOR J=1 TO 20: A(I,J)=0:NEXT J:NEXT I

170 Z=Z+1

180 POSITION 2,22:PRINT "WORD #";Z:"

":INPUT A\$:IF A\$="" THEN 180

190 IF A\$="STOP" THEN 530

200 W\$(LEN(W\$)+1)=A\$:W\$(LEN(W\$)+1)=" "

210 U=INT(RND(1)\*20)+1:L=INT(RND(1)\*37)+1

220 FOR X=-1 TO 1:FOR Y=-1 TO 1

230 IF X=Y AND Y=0 THEN 330

240 X1=L:Y1=U

250 FOR C=1 TO LEN(A\$)

260 X1=X1+X:Y1=Y1+Y

270 IF X1>37 OR X1<1 OR Y1>20 OR Y1<1

THEN B(X+2,Y+2)=0:GOTO 330

280 IF A(X1,Y1)=0 THEN 310

290 IF A(X1,Y1)<>ASC(A\$(C,C)) THEN B(X+2,Y+2)=0:GOTO 330

300 B(X+2,Y+2)=B(X+2,Y+2)+1

310 NEXT C

320 B(X+2,Y+2)=B(X+2,Y+2)+1:B=B+1

330 NEXT Y:NEXT X

340 IF B=0 THEN 210

350 R=2:D=2

360 FOR X=1 TO 3:FOR Y=1 TO 3

370 IF B(X,Y)>B(R,D) THEN R=X:D=Y

380 NEXT Y:NEXT X

390 X=R-2:Y=D-2

400 IF X=-1 AND Y=-1 AND B(1,1)=1 THEN 420

410 GOTO 440

420 X=INT(RND(1)\*3)-1:Y=INT(RND(1)\*3)-1

430 IF (X=0 AND Y=0) OR B(X+2,Y+2)=0 THEN 420

440 X1=L:Y1=U

450 FOR C=1 TO LEN(A\$)

460 X1=X1+X:Y1=Y1+Y

470 A(X1,Y1)=ASC(A\$(C,C))

480 IF BLANK THEN 500

490 POSITION X1+1,Y1:PRINT CHR\$(A(X1,Y1));

500 NEXT C

510 B=0:FOR X=1 TO 3:FOR Y=1 TO 3:B(X,Y)=0:NEXT Y:NEXT X

520 POSITION 2,22:PRINT "

":GOTO 170

530 FOR X=1 TO 37:FOR Y=1 TO 20

continued on next page

continued from previous page

```
540 IF A(X,Y)<>0 THEN 560
550 A(X,Y)=45:POSITION X+1,Y:PRINT "-"
;
560 NEXT Y:NEXT X
570 POSITION 2,22:PRINT "READY TO PRINT;
TURN ON PRINTER AND HIT 'RETURN'
";:INPUT A$:GOSUB 680
580 LPRINT :LPRINT "WORD PUZZLE ANSWER
KEY"
590 LPRINT :LPRINT :LPRINT
600 PRINT :PRINT "PLEASE WAIT A MOMENT
FOR ME TO CREATE PUZZLE. . ."
610 FOR X=1 TO 37:FOR Y=1 TO 20
620 IF A(X,Y)<>45 THEN 640
630 B=INT(RND(1)*26)+65:A(X,Y)=B
640 NEXT Y:NEXT X
650 GOSUB 680
660 LPRINT :LPRINT "COMPUTER GENERATED
WORD PUZZLE"
670 LPRINT :LPRINT :LPRINT :GOTO 730
680 LPRINT
690 FOR X=1 TO 37:A$=""
692 FOR Y=1 TO 20
695 A$(LEN(A$)+1)=CHR$(A(X,Y))
697 A$(LEN(A$)+1)=" "
700 NEXT Y
705 A$=A$(1,LEN(A$)-1)
710 LPRINT A$:NEXT X
720 RETURN
730 LPRINT :LPRINT :LPRINT "WORD LIST"
:LPRINT
740 J=1:FOR I=1 TO LEN(W$)
744 IF W$(I,I)="#" THEN LPRINT W$(J,I-
1):J=I+1
746 NEXT I
760 PRINT :PRINT "WOULD YOU LIKE ANOTH
ER COPY ";:INPUT A$
765 IF A$(1,1)="Y" THEN LPRINT :LPRINT
:LPRINT :LPRINT :GOTO 650
770 END
```

### Apple Version

Instructions and initialization.

```
100 HOME : VTAB 10: HTAB 14: PRINT
"WORD SEARCH PUZZLE": VTAB 1
2: HTAB 14: PRINT "BY DAVID
M. DURKEE": VTAB 14: HTAB 14
: PRINT "COPYRIGHT (C) 1981"
110 FOR I = 1 TO 2000: NEXT I
120 HOME : VTAB 10: PRINT "TO CR
EATE A PUZZLE, SIMPLY ENTER
A WORD YOU WOULD LIKE TO HAV
E IN THE PUZZLE AFTER THE
'?' PROMPT."
130 VTAB 14: PRINT "WHEN YOU'VE
ENTERED ALL THE WORDS YOU
WOULD LIKE IN THE PUZZLE, TY
PE 'STOP' AND THE APPLE WI
LL DO THE REST."
```

```
140 VTAB 18: PRINT "IF YOU WOULD
LIKE TO MAKE A PUZZLE FOR
YOURSELF (BLANK SCREEN), THE
N TYPE '1'; OTHERWISE TYPE '
0' TO BEGIN: ";
150 INPUT BLANK: HOME :Z = 0
160 DIM W$(200),BZ(3,3),AZ(40,20
)
```

Beginning of word-entry loop.

```
170 Z = Z + 1
180 VTAB 22: HTAB 1: PRINT "WORD
# ";Z;": ";: INPUT A$: IF A
$ = "" THEN 180
190 IF A$ = "STOP" THEN 530
200 W$(Z) = A$
```

Choose random starting position.

```
210 U = INT ( RND (1) * 20) + 1:
L = INT ( RND (1) * 40) + 1
```

Check each direction to see if word may be written in that direction.

```
220 FOR X = - 1 TO 1: FOR Y = -
1 TO 1
230 IF X = Y AND Y = 0 THEN 330
240 X1 = L:Y1 = U
250 FOR C = 1 TO LEN (A$)
260 X1 = X1 + X:Y1 = Y1 + Y
270 IF X1 > 40 OR X1 < 1 OR Y1 >
20 OR Y1 < 1 THEN BZ(X + 2,Y
+ 2) = 0: GOTO 330
280 IF AZ(X1,Y1) = 0 THEN 310
290 IF AZ(X1,Y1) < > ASC ( MID$
(A$,C,1)) THEN BZ(X + 2,Y +
2) = 0: GOTO 330
```

```
300 BZ(X + 2,Y + 2) = BZ(X + 2,Y +
2) + 1
310 NEXT C
320 BZ(X + 2,Y + 2) = BZ(X + 2,Y +
2) + 1:B = B + 1
330 NEXT Y: NEXT X
340 IF B = 0 THEN 210
```

Select direction to write word; if possible, choose one which will intersect another word.

```
350 R = 2:D = 2
360 FOR X = 1 TO 3: FOR Y = 1 TO
3
```

```
370 IF BZ(X,Y) > BZ(R,D) THEN R =
X:D = Y
380 NEXT Y: NEXT X
390 X = R - 2:Y = D - 2
400 IF X = - 1 AND Y = - 1 AND
BZ(1,1) = 1 THEN 420
410 GOTO 440
420 X = INT ( RND (1) * 3) - 1:Y
= INT ( RND (1) * 3) - 1
430 IF (X = 0 AND Y = 0) OR BZ(X
+ 2,Y + 2) = 0 THEN 420
440 X1 = L:Y1 = U
```

Print word on screen (unless blank screen was chosen).

```
450 FOR C = 1 TO LEN (A$)
460 X1 = X1 + X:Y1 = Y1 + Y
470 AZ(X1,Y1) = ASC ( MID$ (A$,C
,1))
480 IF BLANK THEN 500
490 VTAB Y1: HTAB X1: PRINT CHR$
(AZ(X1,Y1));
500 NEXT C
510 B = 0: FOR X = 1 TO 3: FOR Y =
1 TO 3:BZ(X,Y) = 0: NEXT Y: NEXT
X
520 VTAB 22: HTAB 1: PRINT SPC(
39): GOTO 170
```

Prepare answer key.

```
530 FOR X = 1 TO 40: FOR Y = 1 TO
20
540 IF AZ(X,Y) < > 0 THEN 560
550 AZ(X,Y) = 45: VTAB Y: HTAB X:
PRINT "-";
560 NEXT Y: NEXT X
570 VTAB 22: INPUT "READY TO PRI
NT: TURN ON PRINTER AND HIT
<RETURN>...":A$: PR# 1: GOSUB
680
580 PRINT : PRINT "WORD PUZZLE A
NSWER KEY"
590 PRINT : PRINT : PRINT
```

Fill in blanks with random letters.

```
600 PR# 0: PRINT : PRINT "PLEASE
WAIT A MINUTE FOR ME TO CRE
ATE PUZZLE...": PR# 1
610 FOR X = 1 TO 40: FOR Y = 1 TO
20
620 IF AZ(X,Y) < > 45 THEN 640
630 B = INT ( RND (1) * 26) + 65
:AZ(X,Y) = B
640 NEXT Y: NEXT X
650 GOSUB 680
```



<pre>660 PRINT : PRINT "COMPUTER GENE RATED WORD PUZZLE" 670 PRINT : PRINT : PRINT : GOTO 730</pre> <p>Subroutine to print complete puzzle.</p> <pre>680 PRINT</pre>	<pre>690 FOR X = 1 TO 40: FOR Y = 1 TO 20 700 PRINT CHR\$(AZ(X,Y));" "; 710 NEXT Y: PRINT : NEXT X 720 RETURN</pre> <p>Print out word list.</p> <pre>730 PRINT : PRINT : PRINT "WORD LIST": PRINT</pre>	<pre>740 FOR I = 1 TO Z - 1: PRINT M\$( I): NEXT I 750 PR# 0</pre> <p>Another copy? If not, then end.</p> <pre>760 PRINT : INPUT "WOULD YOU LIK E ANOTHER COPY? ";A\$: IF A\$ = "Y" THEN PR# 1: GOTO 650 770 END</pre>
--	---	--

**Anallist**  
continued from page 91

```
60660 IFLEN(T$)>64THENGOSUB60130
60670 G!:=G!+1:IF6!<=S9!-2THEN60450ELSEIFLEN(T$)>0THENGOSUB60130
60680 E!:=S9!:GOTO60400
```

Data items.

```
60730 DATA"END ", "FOR ", "RESET ", "SET ", "CLS ", "CMD ", "RANDOM "
, "NEXT ", "DATA ", "INPUT ", "DIM ", "READ ", "LET ", "GOTO ", "RUN ", "
IF ", "RESTORE ", "GOSUB ", "## RETURN ##", " ", "STOP", "ELSE ", "TR
ON", "TROFF"
60740 DATA"DEFSTR ", "DEFINT ", "DEFSNG ", "DEFDBL ", "LINE ", "EDIT
", "ERROR ", "RESUME ", "OUT ", "ON ", "OPEN ", "FIELD ", "GET ", "PUT "
```

```
, "CLOSE ", "LOAD ", "MERGE ", "NAME ", "KILL ", "LSET ", "RSET ", "SAVE
"
60750 DATA"SYSTEM ", "LPRINT ", "DEF ", "POKE ", "PRINT ", "CONT", "LI
ST ", "LLIST", "DELETE ", "AUTO ", "CLEAR ", "CLOAD ", "CSAVE ", "NEW",
"TAB(", " TO ", "FN", "USING ", "VARPTR", "USR", "ERL", "ERR"
60760 DATA"STRING$", "INSTR", "POINT", "TIME$", "MEM ", "INKEY$", " TH
EN", " NOT", " STEP ", " + ", " - ", " * ", " / ", " [ ", " AND ", " OR "
, " > ", " = ", " < ", "SGN", "INT", "ABS", "FRE", "INP", "POS", "SQR", "RN
D", "LOG", "EXP", "COS", "SIN", "TAN"
60770 DATA"ATN", "PEEK", "CVI", "CVS", "CVD", "EOF ", "LOC", "LOG", "MKI
$", "MKS$", "MKD$", "CINT", "CSNG", "CDBL", "FIX", "LEN", "STR$", "VAL ", "
ASC", "CHR$", "LEFT$", "RIGHT$", "MID$"
```

**Lemonade and Champagne**  
continued from page 89

valuable to you during system development.

Symbols have been provided on the Program Design Description form for definition of the Input, Output, and File Media that are to be used. Three Input (I/P) forms symbols have been provided to identify one-, two- or three- Source Documents from which the operator derives data for keyboard entry required by the system. These symbols are annotated with references to the specific Source Document Description Form which will be contained in Section 1.

The CRT (Cathode Ray Tube or, as we call it, Video Screen) symbol is used to identify the applicable Video Screen Format Form, or Forms, which will be contained in Section 2 or the manual.

Two symbols have been provided for Magnetic Tape Input/Output if tape storage is to be used. Tape Files, as well as Disk Files which will be discussed in the following paragraph, are referenced to Section 4 of the manual which will contain the Record Layout Form for the Data File(s) to be used.

Across the bottom of the Program Design Description form there are four blocks representing the Disk Drives. Of course, your system will probably be using less than four drives so use the number of blocks

that are required.

In the appropriate disk drive block you should include the filename of any Disk Data File with which the program or program module being described interfaces. In other words, any file that is read by or written to the program. Each of the filenames given should be identified by its File Type. The File Types we commonly use are:

...“P” (Program). This is simply the required program.

...“D” (Permanent Data File). This is a file of information (data) that is permanently maintained by the system. An example would be the Company Master File, or Employee Data File in our Payroll application from the previous Section.

...“T” (Temporary Data File). This is a file that is created by one program or program module and possibly passed on to another program or program module. Characteristically, when the file has been used for its intended purpose, it is “KILLed”. This type of file is sometimes known as either a “Transient” or “Intermediate” File.

...“I” (Intermediate File). In some cases, a program or program module will require some sort of special independent program such as a Machine Language Subroutine for sorting data. We have included the File Type “I” to accommodate such program files in our documentation.

On the extreme right side of each of the Disk Drive symbols is a column for “reference”.

Continued next month

## FREE 48-Page Catalog

A new, free 48-page catalog is free from Creative Computing and Peripherals Plus. To help buyers make intelligent purchasing decisions, the product descriptions are exceptionally comprehensive and include screen photos in the software section.

The catalog describes 20 books on programming, games, and educational applications; 160 software packages for Apple, Atari, TRS-80, PET, CP/M, TI, Sorcerer and Sol computers; 3 magazines (*Creative Computing*, *Microsystems*, and *SYNC*); 5 graphics and music peripherals; an LP record; board game; 8 T-shirts and an eclectic assortment of other products for the personal computer user.

To get your free copy, simply drop a card or note to the address below.

**creative computing**

Attn: Beverly  
39 East Hanover Avenue  
Morris Plains, NJ 07960

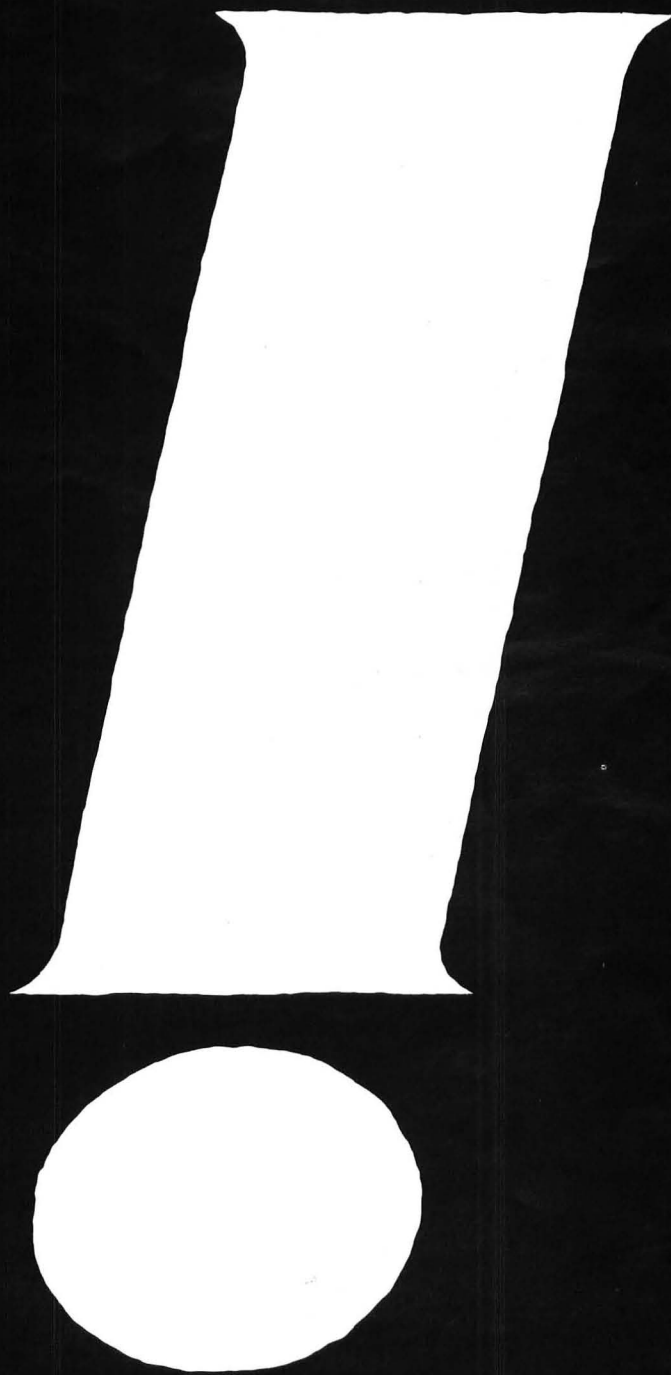
# Word-Search Puzzle #3

G S N F R U K H V C X F X N Q K A L I A  
 P G O S I N C L A I R F M F S E B W R K  
 Z R T X U L J O C M E M O R C G E M Q U  
 Y P O I P B K S O N P W J U I B G M M F  
 T R A G U H S I V V D X C Y E L P P A U  
 Y K O G K G S D U M G X E M M J S I H U  
 T N O R T A X E B E F O E A J S L X K Q  
 U T O P C Y F T L N O R T H S T A R J H  
 H T M C F U Y H R I C N B E I F H A P A  
 C S T Z O Y N A S G X R P I P U L F P P  
 Q Z Z S A K T K G I G F O K I D A T A U  
 I T H W I H T M I T A B R E V A Q I M S  
 Y P Q C E E R K B S R W H T A E H T U U  
 O L L D E D R C U I H E Z I I A Y V M D  
 U L Q P M I N O I T A V O N V C T D F Q  
 Z E R G U S I K L E X I C O N Q Q B M D  
 Y B I U Q D S O Z T W W S J Y Q Z H W M  
 Q R B O Y R V B A W C C Y T N K T I T E  
 J A T Y F A V N I P A R R E T U F Y Y Z  
 D T Z I H H Y I M M W U P F F S C O O J  
 A K V A R Q R P I M E F N U D I A B L O  
 S X T Q B H E B V A B F M G B F J X V I  
 K X S T U S Q J W C S V D O W E V M G P  
 A A C U C A B X Q R G W C A A G E S V H  
 S V I W P A G K X O B K M S D D R N F D  
 W D N L K U R T A T N A V M R G O E Z M  
 Y X O O X S B E L R J T C J F P D M D A  
 D D R R S N X P T O X A V X S I O E N X  
 N J T T U D D R T N W R E L A T M I O E  
 A N N M V C F A M I I I E D B J M S T L  
 T A E E R A X H H C U T L F O W O O A L  
 F S C B O E F S V S E K M K D T C P E S  
 G Y J B C T L C W C J I K K I L E B E H  
 W D P G W P N X X G K N O S P E U Y M J  
 E G E I G N V E F N Z E W K B T A X X C  
 L M R D E H D O B O L U H A N H W N E F  
 P X C R W E L W G P S M I W C T W S L G  
 H Z O K E P R X Y O X T R D J H G Y K X  
 P O M L T R E M N P H J V M K H B Q B T  
 S U A E X I D Y Y B Y P A Z N W T C H G

## WORD LIST

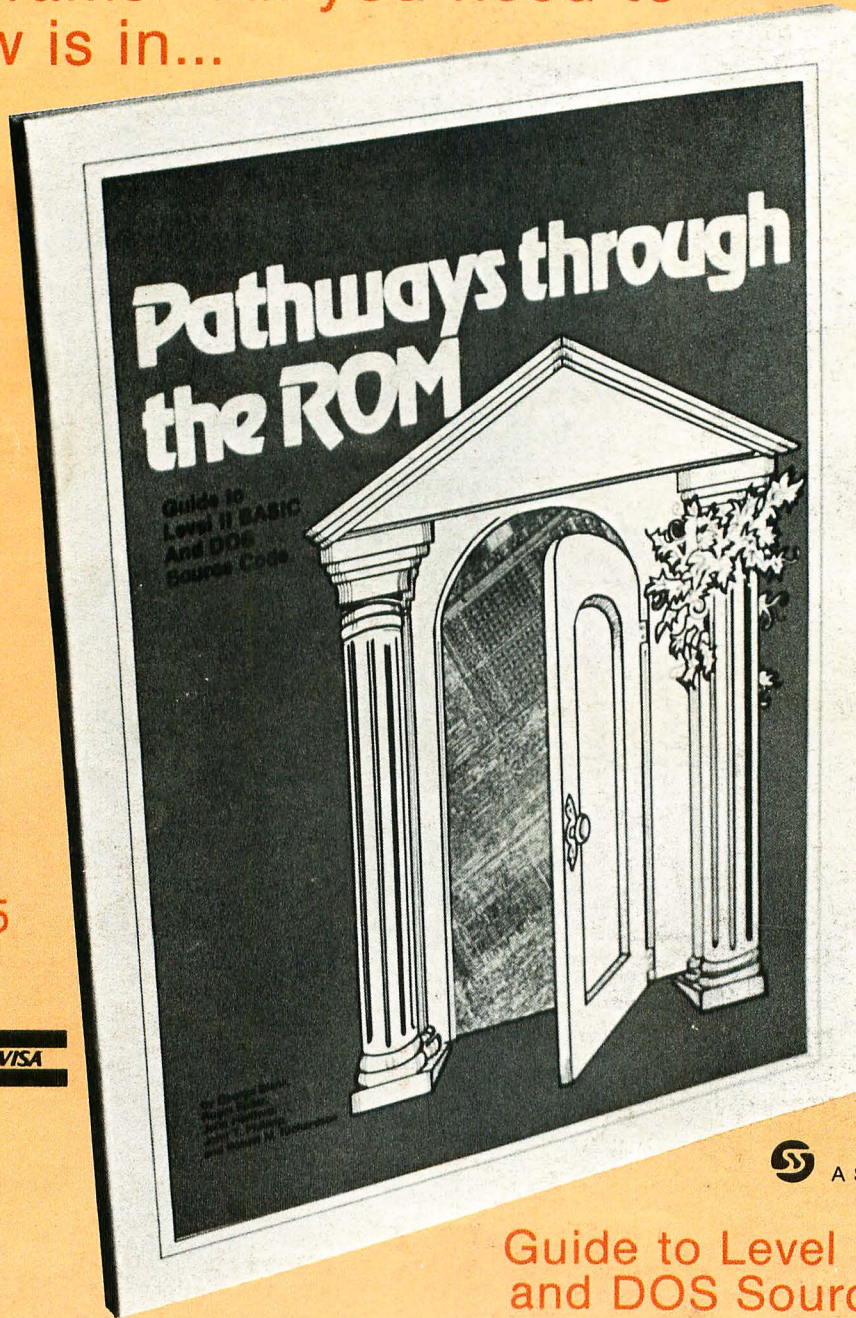
APPLE  
 TANDY  
 ATARI  
 COMMODORE  
 CENTRONICS  
 EPSON  
 OKIDATA  
 SHUGART  
 TEAC  
 PERCOM  
 LOBO  
 LEEDEX  
 EXIDY  
 VERBATIM  
 BASF  
 DYSAN  
 CROMEMCO  
 CORVUS  
 SIEMENS  
 HARDSIDE  
 EXATRON  
 SINCLAIR  
 INTERACT  
 CASIO  
 DIABLO  
 NORTHSTAR  
 NOVATION  
 LEXICON  
 DCHAYES  
 MACROTRONICS  
 EMTROL  
 KURTA  
 SHARP  
 TARBELL  
 TERRAPIN  
 HEATH  
 SONY  
 QUME  
 MAXELL  
 SANYO  
 EATON





***!-\$TRING is coming...***

Unlock the hidden power of your computer for fast and easy programming! Use ROM routines in your BASIC and Assembly Language programs! All you need to know is in...



ALL  
ONLY  
**\$19.95**  
plus \$1 shipping



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

**The Software Exchange**


INCLUDES:

**SUPERMAP**  
From Fuller Software (\$18.95)

**TRS-80  
DISASSEMBLED  
HANDBOOK**  
by Robert Richardson (\$10.00)

**HEX MEM**  
by John Phillip  
Monitor written in BASIC

**Z-80  
DISASSEMBLER**  
by George Blank

 A SoftSide Publication

## Guide to Level II BASIC and DOS Source Code

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