

SoftSide™

TWO DOLLARS
AND FIFTY CENTS

Your BASIC Software Magazine • VOLUME III • NUMBER SEVEN • APRIL 1981

**SKUNK
DOMINOES
ASSAULT IV
ATARI CONVOY
and much, much more!**

IBC... International
Bridge Contractors

**Can you oust
the Chairman
of the Board?**

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 Tardus, and commissioned you to eliminate the evil "Master". Your resources include clones who will fight for you, the formidable CRASER weapons of the Tardus, and magic weapons such as Fusion Grenades and Borelian Matrix Crystals.

Traveling 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.

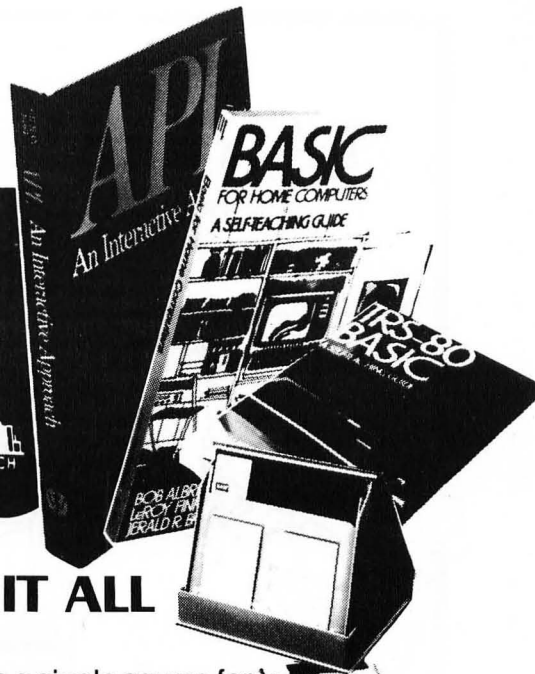
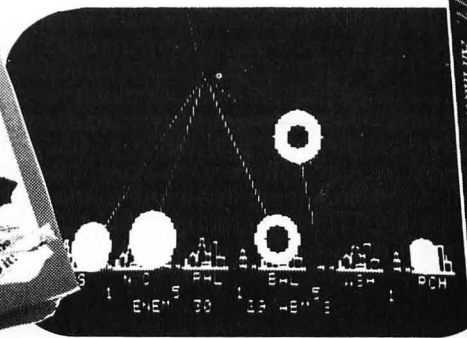
And once you find The Master can you destroy him?


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



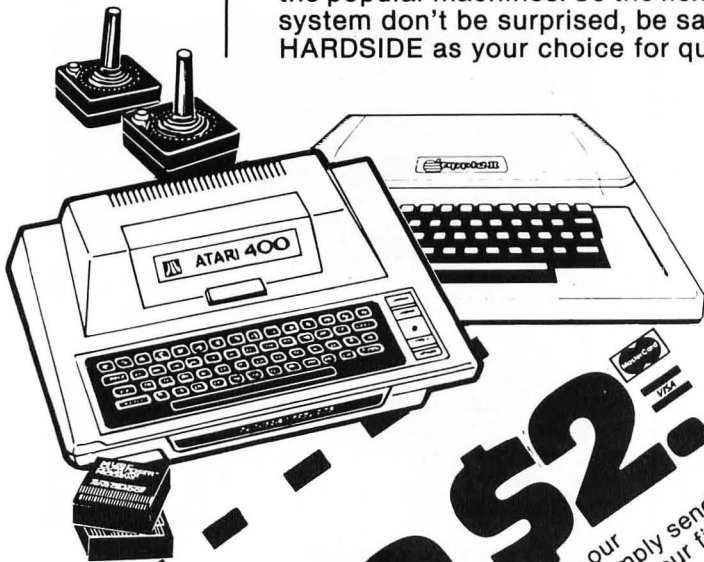
Based on Dr. Who of PBS fame.
Apple Integer Basic,
Disk, 48K ... \$29.95





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.



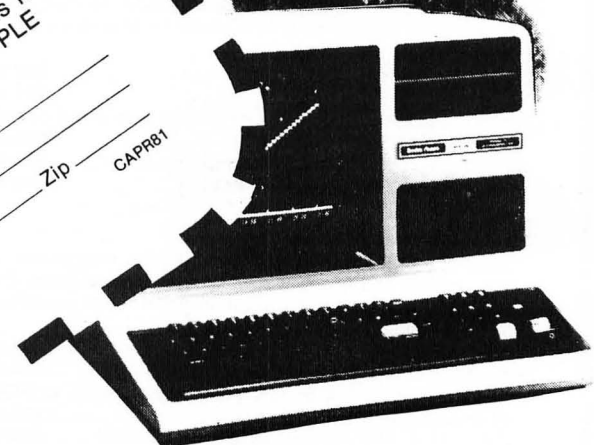
TSE-HARDSIDE
6 South St. Milford, NH 03055 (603) 673-5144
TOLL FREE OUT-OF-STATE 1-800-258-1790



Save \$2.00

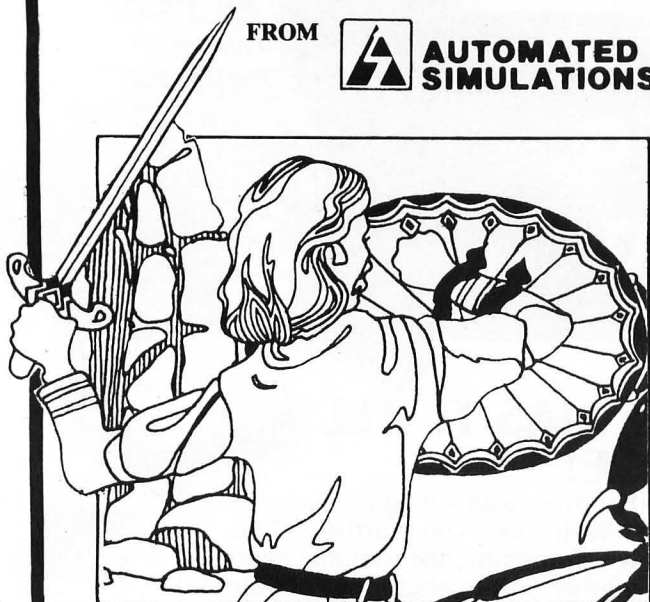
If you or a friend would like to subscribe to our TSE/HARDSIDE Complete Computer Catalog, simply send \$1.00 and receive a \$2.00 certificate good toward your first purchase of software, hardware, books or specialty items from TSE/HARDSIDE. I have a (check one) TRS-80, APPLE ATARI PET. A photocopy of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

Name _____
Address _____
City _____ State _____ Zip _____ CAPR81
MAIL TO: TSE/HARDSIDE
6 South Street
Milford, NH 03055



FANTASY AT YOUR FINGERTIPS

FROM  **AUTOMATED SIMULATIONS**



TEMPLE OF APSHAI

Limber up your sword arm and don your breastplate for a journey to an underground dungeon loaded with fierce monsters and exotic treasures. Equip your character and head for the subterranean ruins of an insect-worshipping culture.

S-80 Cassette: 16K, Level II \$39.95
 S-80 Disk: 32K, TRSDOS \$39.95
 APPLE Disk: 48K with APPLESOFT in ROM \$39.95
 PET Cassette: 32K (old or new ROMs) \$24.95

DATESTONES OF RYN

Rex the Reaver and his band of cutthroats have absconded with the precious datestones from the ducal calendar and your job is to recover them. Not only must you explore the cavernous lair of the robber band, but you must also face an assortment of monsters all hungering for your blood.

S-80 Cassette: 16K, Level II \$19.95
 S-80 Disk: 32K, TRSDOS \$19.95
 APPLE Cassette: 32K with APPLESOFT in ROM or 48K with APPLESOFT on cassette \$19.95
 APPLE Disk: 48K with APPLESOFT in ROM \$19.95
 PET Cassette: 16K (old or new ROMs) \$14.95

MORLOC'S TOWER

You have to thwart the evil necromancer Morloc, who has vowed to destroy your hometown of Hagedorn at the rise of the new moon tonight!

S-80 Cassette: 16K Level II \$19.95
 S-80 Disk: 32K \$19.95
 APPLE Cassette: 32K with APPLESOFT in ROM or 48K with APPLESOFT on cassette \$19.95
 APPLE Disk: 48K with APPLESOFT in ROM \$19.95
 PET Cassette: 20K (old or new ROMs) \$14.95

STARFLEET ORION

Starfleet Orion provides you with the framework of space combat and a dozen scenarios to choose from. Exhaust those scenarios and you can begin to create your own. Your own imagination is your only limit.

S-80 Cassette: 16K Level II \$24.95
 S-80 Disk: 32K \$24.95
 APPLE Cassette: 16K or 32K with integer BASIC in ROM (2 program versions supplied) \$24.95
 APPLE Disk: 32K with integer BASIC in ROM \$24.95
 PET Cassette: 8K (old or new ROMs) \$19.95

INVASION ORION

Your computer would like to battle you in outer space. You name the time, place, and scenario, the computer will take you on, no questions asked. You command a starfleet and must pit all of your tactical skills to stay on top of your opponent.

S-80 Cassette: 16K, Level II \$24.95
 S-80 Disk: 32K \$24.95
 APPLE Cassette: 32K with APPLESOFT in ROM \$24.95
 APPLE Disk: 48K with APPLESOFT in ROM \$24.95
 PET Cassette: 16K (old or new ROMs) \$19.95

RESCUE AT RIGEL

You are entrusted with the rescue of human hostages who have been seized by the High Tollah and are being held on a secret base circling the planet Rigel where dastardly experiments are being performed upon them.

S-80 Cassette: 16K Level II \$29.95
 S-80 Disk: 32K \$29.95
 APPLE Cassette: 32K with APPLESOFT in ROM or 48K with APPLESOFT on cassette \$29.95
 APPLE Disk: 48K with APPLESOFT in ROM \$29.95
 PET Cassette: 20K (old or new ROMs) \$19.95
 Atari 32K Cassette \$29.95



HELLFIRE WARRIOR

Four more levels of dungeon have been discovered beneath the Temple of Apschai region. These four levels are populated by Undead, Demons, and an assortment of thoroughly nasty types. There is also a good deal of treasure down there, but not many have survived to report about it.

S-80 Cassette: 16K Level II \$39.95
 S-80 Disk: 32K \$39.95
 APPLE Disk: 48K with APPLESOFT in ROM \$39.95


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

TABLE OF CONTENTS

ARTICLES

- 10 **MODIFYING STRING VARIABLES**
Tricks with your VARPTR..... S-80..... Benjamin Junge
- 26 **COMPUTER GRAPHICS**
Perspective Drawing Apple..... Joan Truckenbrod
- 54 **APPLESOFT CHAINING**
Quick & Dirty vs. Elegant..... Apple..... Murray Summers
- 69 **WHAT TO DO AFTER THE PROGRAM LOADS**
With apologies to Sherry Taylor Ed Ting
- 83 **PADDLE SOUND EFFECTS**
The name says it all Apple..... James Dwyer
- 84 **REVIEWS**
Temple of Apshai; Apple Machine Language; Dragonquest;..... Michael Humes, Jon Voskuil, Dave Albert
- 93 **AUTO MODPROG**
Programming perspectives..... S-80..... Shane Causer

PROGRAMS

- 20 **MESSIAH**
An Easter greeting..... Atari..... William Morris & John Cope
- 22 **INTERNATIONAL BRIDGE CONTRACTORS**
Cross this one when you come to it S-80..... Phillip Case
- 34 **CONVOY**
The long-awaited Atari version Atari..... William Morris & John Cope
- 42 **ASSAULT IV**
The Allied forces await your command S-80..... Jerald Uptain
- 48 **DOMINOES**
Three versions of a classic game S-80..... Peter Kirsch
- 58 **SKUNK**
Watch out for those furry fellows Apple..... John Daoust
- 60 **STRATEGY STRIKE**
Brush up on tactics..... Atari..... David Steenson, Rich Bouchard
- 66 **MATH DECATHLON**
A mathletic maelstrom..... Apple..... Jon Voskuil
- 70 **ROBOT BUILDER**
A game, not a tutorial..... S-80..... Arnold Gross
- 74 **FLAGS**
How's your geography? Apple..... William Morris & John Cope
- 76 **BATTLE AT SEA**
An old game with a new twist Apple..... Jon Voskuil

DEPARTMENTS

- 4 **EDITORIAL**..... Jon Voskuil
- 6 **INPUT**..... From our readers
- 7 **OUTGOING MAIL**..... Dave Albert
- 14 **I DON'T THINK WE'RE IN KANSAS ANYMORE**..... Mark Pelczarski
- 19 **ABOUT THIS ISSUE**..... The Munchkin Bunch
- 29 **SAY YOHO**..... Scott Adams
- 31 **BASICALLY SPEAKING CONTEST**..... Editors
- 77 **WORD PUZZLE**..... Editors
- 87 **BUGS, WORMS, & OTHER UNDESIRABLES**..... Kay Pasa
- 88 **HARDWARE CORNER**..... Ed Umlor
- 92 **WHAT'S NEW**..... Ed Umlor

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



PET

*TRS-80, Apple, Atari, and Pet are registered trademarks of Tandy Corporation, Apple Computer Company, Warner Communications, and Commodore Business Machines.

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
Elaine Cheever
Lynda Fedas
Lauri Miller

STAFF

Patricia Acampora
Lester Anderson
Ruth Anderson
Brian Berkebile
Diana Bishop, **Subscriptions**
Kathleen Boucher
Suzanne Breton
Phillip Brown
Brenda Cookingham
Donna Cookingham
Jeffrey Carroll
Pam Demmons
Mary Edwards, **Software Information**
Mark Eric, **Atari Submissions**
Anthony Fraser
Mary George
William F. Gollan, **Advertising**
Mylene Grigas
Kate Hannon
Pam Horne
Robert Hunsacker
Dave Hutchings
Donna Jean
Janice Johnson
Bette Keenan, **Customer Service**
Bea Kimball, **Software Shipments**
Karen Lawrence
Kathy Maloof
Jean Mathews
Dick Melhorn, **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**
Cristine 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

by Jon Voskuil

“How do I code thee? Let me count the ways....”

It's true that programming in BASIC is often more of an art than a well-defined science. There's room for much difference of style among programmers, much variety of “artistic expression” in the way we code ideas into program lines.

Nonetheless, the end result of our programming ought to be that our ideas and logic are EXPRESSED through the code we write rather than being OBSCURED by it. There is a difference between subtle programming and obscure programming.

Most of the programs which appear in **SoftSide** are written by you, our readers — not by professional programmers. The work you do reflects a tremendous amount of creative energy. This is just encouragement to do what you do...with style. Following are some specific suggestions for polishing your programming style — whether or not you submit the results to us for publication.

1. Use lots of subroutines. Some programs are short and simple enough not to need them, but most could be greatly improved with more of them. Think of your program in terms of a fairly short main control routine, which calls one subroutine after another to do the various tasks within the program. Each subroutine may, in turn, call other subroutines. If you can reduce your program to building blocks like this, it shows that you have a good grasp of the logical structure of the whole. It makes programming a LOT easier, greatly simplifies debugging, and results in a growing supply of subroutines which can be transplanted with little revision into another program to do similar tasks. Not only that, but other programmers will find it easy to follow the logic of the program — as will you, when you go back to revise it a few months later. And those further additions and revisions (even translation into another BASIC dialect) will be far easier to do when the flow of the program is revealed, rather than

hidden, by the coding.

2. Use line numbers to help structure your program. For example: A typical pattern for me is to begin the main program on line 100, and try to END it by line 990. Lines 10-98 are reserved for short, frequently used subroutines beginning at multiple-of-10 line numbers. (The statement GOTO 100 must therefore precede line 10.) The lines 1000 and up are for the subroutines that do the real work — usually beginning at multiple-of-1000 line numbers. Line numbers are generally incremented by ten, with these common exceptions: short subroutines and text-printing sections which might be incremented by only two or five, and logical breaks in the program where I might jump to the next multiple-of-100 line number. A renumbering utility can be a terrific asset in cleaning up a program once it's completely coded.

3. This is really derived from the above two suggestions, but bears a separate number all by itself: Group ALL subroutines at the beginning and/or end of the program. In my humble opinion, practically nothing makes a program harder to follow than mixing subroutines into the flow of the main control routine. (Yes, there are exceptions.)

4. Use memory-saving and keystroke-saving techniques in moderation. Specifically:

a. One-letter variable names are short and sweet, but longer ones are often worth the extra space. A name that suggests the use of the variable is a form of documentation that can make the whole program easier to follow (although this can be overdone too).

b. Long lines of coding are occasionally necessary, but often make it very difficult to follow what's going on. Use them sparingly; the few extra bytes of memory for another line number are seldom significant.

c. Lack of spaces — especially in those long lines — can make a program just about unreadable. Some computers (such as Apple and Atari) put spaces in program lines whether you type them or

continued on page 11



NATIONAL TRS-80[®] MICROCOMPUTER SHOW

**New York Statler
Exposition Hall
(opposite Pennsylvania Railroad Station
and Madison Square Garden)
7th Ave. & 33rd Street**

**May 21, 22, 23, 1981
Thursday Noon to 6 PM
Friday 11 AM to 6 PM
Saturday 10 AM to 4 PM**

For the Businessman, Educator, Professional and Hobbyist.

- Commercial Exhibits and Sales of Microcomputers, Software, Books, Magazines, Supplies, Parts, Printers, Etc.
- Free Seminars
- Famous Name Guest Speakers.
- User Groups.
(Mod I, II, III, Color and Pocket Computers).
- Door Prizes - A TRS-80[®] Computer will be given away free each day, plus other prizes.

Avoid standing on line - Send in your registration today!

Radio Shack and TRS-80 are registered trademarks of Tandy Corporation, which has no relationship to Kengore Corporation or The National TRS-80 Microcomputer Show.

REGISTRATION FOR TRS-80[®] MICROCOMPUTER SHOW

May 21, 22, 23, 1981

New York Statler Hotel

Name _____ Title _____

Company Name _____

Address _____

City, State, Zip _____

Please send _____ registrations at \$10.00 each.

(Registration Badge will be sent to you on May 1st.)

Send To:

Kengore Corporation, Dept. 80
3001 Route 27
Franklin Park, N.J. 08823

Co-sponsored By



(Be sure to enclose
check or money order.)

INPUT

Dear **SoftSide**:

In the May 1980 issue of **SoftSide** there was a program listing called "Fifteen Game". Since the listing was short and I was learning how to use my new Apple, I tried typing in and running the program. The program ran, but there seemed to be a bug in it. The number which was being moved to a new location was not blanked out in the old location. I kept waiting for someone to send you a correction to the program, but either no one did, or someone did and you didn't print it, or somehow I missed it. Well, in any event I finally felt compelled to sit down and figure out a solution to the problem which I now share with you and which you may share with the rest of the world.

```
510 IF N(I,J) = 0 GOTO 640
640 HTAB 8 / I * 5 - 1
650 VTAB 2 / J * 4
660 PRINT " "
670 GOTO 540
```

David Royce
Hazelwood, MO

Dear **SoftSide**,

The following is a small idea that I have found very useful, so I thought I would share it with you and my fellow readers of **SoftSide**.

Every time I type in a program from a magazine, such as **SoftSide**, I will either lose my place and type half of a line wrong or mess up a data statement

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.

so bad that I end up scrapping the whole thing and starting over again. Then it occurred to me, why not get my brother to read the lines to me as I type. Well this works, but not too well. I find it difficult to tell his "b" from his "d" or "e". To solve this little problem I dug up the tape recorder that I had cast aside when I bought my Disk II. Now when I wish to type in a long program I read the listing with the recorder on, then replay the recording when I'm ready to type. Now all I have to do is watch the screen and not look back and forth from listing to screen to listing again! On my recorder I have a pause button which I find very helpful. If you happen to have a nearby stereo that plays cassettes you may wish to put on the headphones and ignore all the knocking at the door, telephone calls, and the numerous other things which seem to distract you every time you sit down to type in a program.

Randy Reeves
Cypress TX

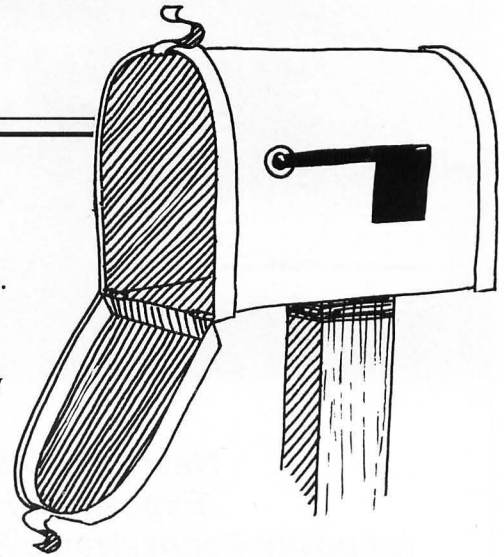
Dear **SoftSide**:

I am the district media supervisor for a school district of 3200 students and 406 staff members. Early in August we requested an evaluation copy of **SoftSide** magazine. The issue we received was the September issue. We were so impressed with the single magazine instead of one for each computer area that my staff recommended that we subscribe. I put it into my evaluation file. Later that week I took the time to go through the magazine. Wow! Was I impressed. Each article was well-written and easy to understand. I also liked the idea of telling the reader which article dealt with the different types of computers.

The article which really caught my eye was the one by Mark Pelczarski titled "**SoftSide's** Developing Data Base". We put this program into our Apple and started to use it that very day. That article helped us decide to subscribe to your magazine. Mark has done a super job with this data base. I have called him several times with problems and he has always been more than willing to assist me with the problem at hand.

I talked to Mark the other day and he informed me that the series will end in March. He also said that there was a great deal more he would like to add to the data base but two more articles will not be enough space to make all the additions he wants.

We have made great use of the data base program and many others in the magazine. At this time I feel it is one



of the best computer magazines on the market. Keep up the good work and we will continue to subscribe.

Thank you.

Don G. Erickson
Lebanon, OR

Dear **SoftSide**,

"Little Brick-Out" is one of the extra goodies that comes with the purchase of an Apple II+. Alas, once you've aced-out the game (knocked out all of the bricks on the first ball) it loses much of its appeal.

The easy modifications listed below introduce a new dimension — "dead spaces". A dead space is an area at the top and bottom of the usual paddle range which the paddle cannot reach. Depending upon the "Skill Level" selected, you will have to guide your shots to avoid having the ball hit in a dead area. At Skill Level 2, two ball-widths at the top and bottom become unreachable; at Level 3, three ball-widths; and so forth up to five ball-widths at Skill Level 5. (At the 0 Level there are no dead spaces and the game plays the same as the original version.)

Since any dead space significantly increases the game's difficulty, this is reflected in the scoring. The higher the Skill Level selected, the greater the points/brick and the larger the bonus for remaining balls.

Restrain paddle travel up (TS0) and down (BS1)

```
30 OQ= Q: Q= INT(39- Q/6)
32 IF Q < TS0 THEN Q= TS0
40 IF Q > BS1 THEN Q= BS1
```

Update score based upon skill level

```
280 K9= 2 * INT(K/2): VLIN
K9, K9 + 1 AT I
282 I9= 1/2 - 9
283 I9= I9 + I9 - TS0
284 S= S + I9: VTAB 24:
HTAB 8: PRINT S;
VTAB 23: HTAB 1
```


Decide if all bricks are gone (SS = Basic Max Score)

```
340 IF S < SS THEN 250
```

Award bonus points based upon skill level (TS0 also = level)

```
1540 FOR Q = 0 TO L * (50 + TS0 * 30) STEP 5
```

Enter skill level (TS0), set top and bottom dead space (TS0 and BS1), and set Basic Max Score (SS)

```
2224 HTAB 9: PRINT "SKILL LEVEL? (0-5) ";  
INPUT "gg"; TS0
```

Note: "gg" here indicates two CONTROL G (bell) characters

```
2226 TS0 = ABS(INT(TS0))  
2227 IF TS0 > 5 THEN TS0 = 5  
2228 SS = 720 + 720 * TS0  
2229 BS1 = 34 - TS0  
2230 PRINT
```

Jeff Hurlburt
Houston, TX

Dear **SoftSide**:

In your February 1981 issue, there is a column by Scott Adams ostensibly concerning software pirating. I hope Mr. Adams is more careful in his programming than he is in his thinking about this problem.

Pirating is defined in the lawsuit he mentions in his postscript. It has nothing to do with the argument he had with the man who offered to give a friend a copy of a program. Pirating involves taking the creation of an author, copying and then selling it without the authorization of the author.

When Mr. Adams contracted to write a column for **SoftSide**, the contract was not only with the publisher but with the public that reads his column. If he actually perceived the "English High Court" decision to be on a case analagous to the one he argued with the fellow copying a program for a friend, then I suggest that he stick with programming in his column and leave the heavier subjects to people who know what they are talking about. If, on the other hand, he knew full well that the case did not apply, then he owes an apology to his readers for his attempt to mislead them.

Did he ever lend a friend a book and then feel guilty about the money that was not going to the author? Would he suggest that a snow thrower company sue the three neighbors that chipped in and share one machine? Would he close all libraries? Would he outlaw VCRs and tape recorders? How much does Mr. Adams pay Tandy for creating a market for his programs? Or is he leeching off of their huge gamble?

The problem with Mr. Adams and his ilk is that they fail to see anything but their own parochial interest. They easily forget what they owe to their contemporaries and predecessors. I'm sure Mr. Adams never inquired of Microsoft if he could use their BASIC

in his programs. How much has he contributed to Dartmouth?

The fact is that Mr. Adams lives in a community and not by himself, and he gets from and gives to this community. I would suggest that, to-date, he has taken much more than he has given.

I believe program pirates should be arrested and jailed. But, let's be quite precise about this; it does not one any good to completely confuse the issue. The pirate attempts, in an organized manner, to seize the market of the legitimate producer. What one friend gives or lends to another is neither the government's nor Mr. Adams' business.

Eli Passin
New York, NY

Dear **SoftSide**,

I discovered a problem in one of the games featured in the November issue. The program was "Meteor Storm", and the problem is that the player can render his spaceship impervious to meteors by placing it over his score in the upper left-hand corner of the screen. This problem is encountered on the S-80 version of the program.

This problem can be solved by the following changes:

```
120 PRINT @G, " "; PRINT @  
RND(62) + 960, " +"  
150 IF PEEK(L) = 43 or  
PEEK(L + 1) = 43 or  
PEEK(L + 2) = 43 then 195  
155 Print @ 0, P0;  
195 Print @ 0, P0;
```

These changes should solve the problem.

Dan Singer
Washington, DC

Dear **SoftSide**:

I own an Apple II computer, and have been a faithful reader of your magazine, **SoftSide: Apple Edition**, since it came out. In your days as separate magazines, there was only one adventure game for the Apple computer, that being "Dog Star Adventure" in the very first issue. Seeing your new format, one larger magazine, I hoped that this would mean some better games for the Apple.

Indeed, in one respect, there have been many very good games for the Apple computer. Unfortunately, I have seen no adventure-type games. But, almost every issue there has been an adventure game for the S-80 (I am refraining from using the common term, TRASH-80). Some of these games are: Dr. Livingston, In Search Of; Mad Scientist; and Kidnapped, in only FIVE magazines.

I hope I speak for many Apple owners when I say that I am disappointed that there have not been any more adventure games since the very first issue. I hope this letter will influence a change in policy. If it does, I, for one, will be overjoyed.

Ben Cohen
Jersey City, NJ

continued on bottom of next page

OUTGOING MAIL

by Dave Albert

Hello. I'm a new face here, if you'll pardon the expression. Perhaps I should say a new voice. Whatever, this is my space for a while to talk back at you, the reader. With all due respect, you need it sometimes. What I would like to do in this space is open the door to the editorial office and invite you into the inner workings of **SoftSide**. I don't promise a cook's tour every month, but I would like you to get to know us, what we are and how we work.

To begin with, I'll introduce myself. I'm Dave Albert. For lack of a better title, I'm referred to as the Managing Editor. Depending upon the extent to which you read our magazine, you may have run across my name at the bottom of an editorial or at the end of a review. You have not, and will not, find my name at the top of a program for the simple reason that I am not a programmer. I am an editor. My background is in newspapers primarily, with freelance writing experience tossed in somewhere along the line. What I actually do is unclear to most, and an absolute mystery to me. But if the magazine doesn't get out every month, I know who's in trouble.

Some of my predecessors, most notably James Garon, have made attempts at explaining just what goes into putting this magazine out. But their explanations just dealt with the mechanical aspects of the process. In future columns I will try to let you in on how we decide what goes in the magazine (a dart board), how we get the submissions ready for print (trained spiders in inkwells), and why some programs don't run (so that the ones that do run will look better).

I will also try to answer some of the more frequent complaints and/or queries from you folks. With any luck I might even disabuse you of any notions you may have about our 347 person editorial staff! You can forget about the last two digits of that figure if you want to get close to

continued on next page

continued from previous page

the truth. To show my sincerity, here's one for you:

We have received a number of complaints about an incomplete listing for a program called "Boing!" It was a valid complaint, there were some lines missing...however, there are some extenuating circumstances involved. To understand them you first need to understand our procedure in getting a program ready to publish.

We start out by getting a working version of the program up and running on a computer. Once that is accomplished, (not an easy task), we then document it by using another program that puts the one we're working on into a word processor file and allows us to splice the documentation right into the listing. That program, call it "64", had some problems, one of which was that it tended to get hungry and eat the last few lines of a program. We don't splice Atari documentation into the program itself any more, and we did get the S-80 side of the program straightened out, but not before losing lines from "Convoy" and a couple of other programs. Like anyone else, we learn from our mistakes, but we do make them.

Now, if the people who actually paste up the magazine were programmers, things like this would get caught...but if you hire programmers, it is so that they will do programming, not paste up. The production department gets program listings handed to them

ready to be photographed and pasted into the magazine.

The editor who handles the documentation just splices it into a WORKING program and dumps out a listing on a lineprinter. Until the complaints started coming back to us, we had no reason to suspect that lines were being dropped at the end of the listings. Now we know. The procedure has always been to make sure the program ran, then to splice in the documentation. Now we check them out AFTER we document them, as well as before.

Furthermore, the editor that prepared "Boing!" for publication has since departed for sunnier climes. He wasn't around to supply the fix. The disk with the working program got lost in our moving the editorial office to another location. We simply did not have a working copy of the program. As soon as we got it straightened out, we started mailing it to those who had written in. Others managed to fix it themselves. For those that didn't, it is published in this issue in "Bugs, Worms and Other Undesirables".

But there's yet another side to this affair. About 90% of the complaints we receive are based on bugs that DO NOT exist. I know that sounds self-righteous, but I sit here all day reading complaints and running them down. They are, by and large, spurious. People don't type the programs in precisely and then complain that they don't run right. After a few

hundred of those complaints flash by, it's easy to assume that the latest Bug letter is probably due to a bad job of typing.

The point I tried to make in responding to a letter on the Input page was simply that we do not type programs in. The errors are not the result of poor proofreading or typing. When we find mistakes, we try to correct them. But some mistakes are totally unanticipated, and until we know how they occur, there is not much we can do about them. That is where you, the readers, come in. Write to us; let us know the problems you encounter. But when you do so, please give us as much information as you can: What system you use, what medium (disk or cassette), where you think the problem is, etc. Don't fly off the handle, it doesn't help you or us. We are trying to put out the best magazine that we can, and we do not have a large staff at all. We will continue to make mistakes, it is just the human thing to do. But we try to fix them and we try to keep you happy. Sometimes in a moment of frustration, after spending hours typing in a program, one can get angry when it doesn't work. But remember that at the other end of the line there are plain, ordinary (if somewhat talented in one respect or another) people, working as best they can. An angry letter won't get any faster results than a calm one. You catch more flies with honey than with vinegar.



continued from previous page

Dear **SoftSide**:

I wonder if you have ever had reports on how well your tapes work on the Model III. I have a Model III with 48K and two disk units as well as my 48K Model I with disks. I have recently received three tapes from you and I have tried them on the Model III. First, the Model III doesn't give a hoot where you set the volume control on the recorder. Get it within wide limits and the tape loads! Here is my report on the tapes:

November:

Everything went fine until I got to the "Card Draw." It wouldn't run. I listed it. Lo and behold! The first line says: 999 CLS:END. The book doesn't give this. Eliminating this and running "Out of String Space in 1030" came up. So I didn't get it working. On "Cards", it worked but it would not start or stop the recorder.

December:

Everything worked fine until

"Kidnapped." Listing and comparing with the book showed a comma replacing a paren in line 200. I changed this and everything worked fine. So this one was 100%.

January:

The first program "Convoy" gave trouble. It initialized forever. I listed it and line 100 was goofed up. I reran the tape. Line 100 is still goofed up. I checked with the book. The 22 slashes were missing and were replaced with the gibberish. I retyped this line and it still initialized forever. So this one was a loss. The other programs worked perfectly.

Now it may be something other than the Model III that prevents these few programs from running properly. But these are the results I got with them. I like the Model III very much. While one does not have to "CMD" "T" anymore it is necessary to POKE 500 baud. I have had some difficulty converting Model I disks to Model III, basically Machine Language programs although

I have converted many machine programs okay. It seems that if it is along program, trouble ensues. Tapes convert to disk excellently. I like it but it would have been nice if I had the Model III to begin with.

I thank you for providing these tapes with the magazine. The cost is well within reason, and the results are excellent. I might tell you that I have two Model I Level I which I loan to school children to get them interested. It has worked in almost (but not all) cases.

Again thanks. I just thought you might like a report.

Andy Anderson
Holton, KS

Editor's Reply: Thanks for the report, it's always good to get such items. As far as "Convoy" is concerned: Whoops! We dropped a few lines. They are printed in the "Bugs" section of the February issue. It should run fine if you add them.



Ten reasons why your floppy disk should be a BASF FlexyDisk.[®]



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[®] 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.

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

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.



BASF

Floppy Disks Mag Cards Cassettes Computer Tapes Disk Packs Computer Peripherals

SoftSide, April 1981

MODIFYING STRING VARIABLES

by Benjamin Junge

Have you ever created a drawing on your video screen that was so fantastic that you had to save it, one way or another?

Unfortunately, this exasperating situation has hit me many times, especially when I don't expect to come out with a good picture and I am totally unprepared. There are a few ways to do it.

First of all, you can place the ASCII values of each character on the screen into an array of subscripted variables. But in order to do this, you have to put a DIM statement somewhere in your program before the preservation is done, which takes an immense amount of memory. Then when you turn off the computer, the variables are cleared and your masterpiece is lost forever.

The second way to "save the screen" is to print the ASCII values to tape or disk, but that takes so much time that you would be asleep when the saving finished. There are some other methods, but most of them are either time-consuming, or insufficient in doing the job.

The one way that will accurately, and fairly quickly, save the screen, is called "Super Graphics". The term, Super Graphics, refers to using string variables to print out some sort of display. This process may be accomplished by determining the location of the string in memory, and then POKEing the ASCII values from the video into the memory occupied by the variable.

In TRS-80™ BASIC, finding the section of memory at which a string variable is located is done with the VARPTR(X\$) function, where X\$ is the variable being examined. VARPTR does not return the actual variable storage memory location, but a series of three values that will help you find the correct memory location. This system of storing one memory location in another is called a pointer, or variable pointer (VARPTR).

The value stored in VARPTR(X\$) is the length of X\$. The second and third memory locations (VARPTR(X\$) + 1) and

(VARPTR(X\$) + 2) contain the Least Significant and Most Significant Bytes of the actual variable storage area. In order for us to retrieve the actual memory location for X\$, we must manipulate an arithmetic formula properly. The formula is:

```
10 A = VARPTR(X$)
20 B = PEEK(A + 1) + 256
*PEEK(A + 2)
```

This example will return the actual location in memory where X\$ is stored in the variable B. So now that the exact storage area for X\$ has been established, we can actually alter the contents of the variable.

For example, if you wanted to preserve the first line of the video screen (63 characters), here's what you would do:

1. Establish a string variable, let's say A\$, so that it has a length greater than 63 characters.

2. Now determine the memory location of the storage area for A\$, with the formula shown earlier.

3. Set up a FOR-NEXT loop that will read the character codes from the first line of the screen and POKE them into A\$.

The following program illustrates the procedure described above in more detail.

```
10 'THIS PROGRAM WILL
```

PUT THE FIRST LINE OF THE SCREEN

```
20 'INTO THE VARIABLE, A$
30 '**PRINT SAMPLE
  GRAPHICS PATTERN**
40 CLS:FOR A=1 TO 100
50 SET(RND(127), RND(3)-1)
60 NEXT A
70 CLEAR 64 'SET ASIDE
  MEMORY FOR A$
80 A$=""
  ''
90 A = VARPTR(A$)
100 B = PEEK(A + 1) + 256
*PEEK(A + 2)
110 FOR I=15360 TO 15422
120 POKE B, PEEK(I)
130 B = B + 1
140 NEXT I
150 PRINT @ 192, "NOW A$
  CONTAINS THE GRAPHICS
  SHOWN ABOVE. A$="
160 PRINT A$
```

After you have altered a string, it will appear as BASIC commands. Do not be alarmed. This is normal. Just one suggestion: Try to avoid EDITing lines that contain Super Graphics. If you do so, the string will no longer print the specified graphics, but the words that show when looking at the variable in a listing. (Check **SoftSide**, December, 1980 for a method of editing Super Graphics. The Editors.) Play around a little! This is a pretty fast method, and you may find it very useful.

S-80 ONE LINERS

```
1 IF DX=0 THEN CLS:DX=1:H=RND(23):P=RND(0):GOTO1 ELSE X=X+DX:IF
  ABS(X)>63 THEN H=RND(23):P=1+RND(4):DX=-DX:GOTO1 ELSE Y=H*SIN(.0
  5*P*X):SET(X+63,Y+23):GOTO1
```

Terry Higman
Arlington, WA

```
1 CLEAR 900:CLS:FORG=1TO127:S$=S$+CHR$((RND(63)+128))+",":NEXTG:
  FORQ=1TO1000:PRINTS$;:NEXTQ:RUN
```


Ray Dashner
Pasadena, CA

```
1 CLEAR100:CLS:FORG=1TO5:S$=S$+CHR$((RND(63)+128))+",":NEXTG:FOR
  Q=1TO93:PRINT@1013-P,S$:P=P+11:NEXTQ:P=0:FORX=1TO1500:NEXT:PRINT
  @0" "":FORV=1TO16:PRINTSTRING$(64,3EXTV:RUN
```

Ray Dashner
Pasadena, CA

continued from page 4
not, but others (such as the TRS-80™) don't.

d. Use REMarks. Although we purge programs of REMs before we publish them in SoftSide, that's not because we're philosophically opposed to them. We think it's clearer, for publication purposes, to document them as we do. But you may well want to add REMs to programs that you copy from our pages, inserting the notes and even the variable lists that we provide. I generally reserve the line numbers ending in "9" for REMarks, to state the purpose of the following lines. Although some of these end up being deleted when the program is finished, they do help during development, and the most helpful ones are left in for future reference.

None of the above suggestions are chiseled in stone. Programs are written every day which ignore every one of these, and which nevertheless work perfectly. The issue for us as programmers is whether we're content just to do it any old way, or whether we want to aim for the satisfaction of doing it with style. 

Programming Hints

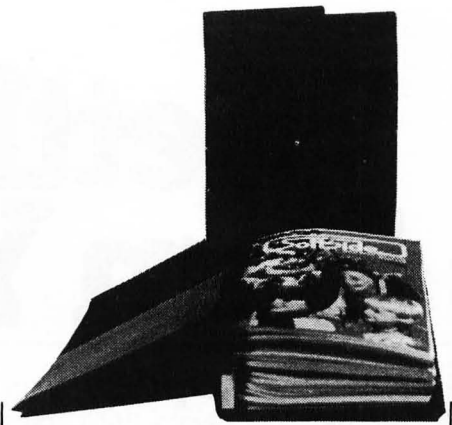
For a really micro micro-word processor: use the following routine:

```
1 A$=INKEY$: IFA$=""
THEN 1 ELSE LPRINT A$:
GOTO 1
```

While this processor doesn't have any editing commands, it is extremely useful for adding notes to program printout or as a short memorandum.

When using string packing techniques for graphics or machine language routines, you don't have to type the traditional
A\$="////////////////////".
The program will still work with a simple A\$=STRING\$(20,75) or whatever the conditions you have are. This cuts down on screen "garbage", and saves memory as well.

Shane Causer
Brunswick, GA



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).


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

Sports fans

Enjoy two of your favorites without leaving home!



written by
David Bolke

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/Atari 16K Cassette \$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/Apple/Atari 16K Cassette \$9.95
Apple 32K Disk \$14.95




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

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



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

I DON'T THINK WE'RE IN KANSAS ANYMORE

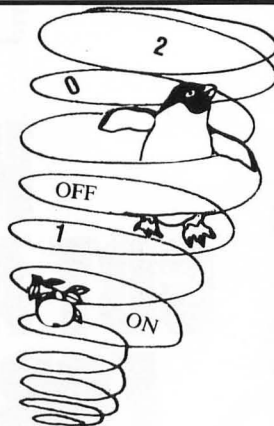
by Mark Pelczarski

The first time I encountered the real inner workings of a computer was in a programming course. We learned Fortran on an IBM 1130, punching our programs on cards, one line each, then putting them in a submissions bin where they'd eventually be whisked away into another room. A few hours later our output would arrive, stacks of computer printouts that would tell us what happened. Most of the time something went wrong, as expected among beginners, and some cryptic, handwritten scrawl from the heralded computer operator would appear at the top. There was a chart that we could look at to see what it meant, and more often than not the scrawl said "F00-9". This allegedly is what the computer spit back out from our program, and it meant something like "I just don't understand". Foo on it.

This computer was a magnificent piece of machinery; so magnificent that it even had its own air-conditioned room. I remember going in there occasionally and standing in awe of all those huge cabinets and the wires running all over the floor. There was a cabinet for the card reader, one for the printer, one for the memory, one for the disk drives, and a control console with flashing lights, a keyboard, and lots of switches. It had all of 16K bytes of storage. This was not in the dark ages; this was in 1970.

I learned later that F00-9 was actually F009; the 0's were actually zeroes. The flashing lights on the control console actually showed the contents of a word (two bytes) of storage, and when a program bombed, this storage location would show a certain value relating to the type of error that occurred. In the case of the dreaded F009, it actually displayed:

111100000001001 (in a fit of binary convenience, I've used 1s for lights that were on, 0s for lights that were off). Somehow F009 was an abbreviation for that pattern of ONs and OFFs. That was my first brush with hexadecimal.



ON AND OFF

All those little ONs and OFFs, all that a computer can really store, are organized into little bunches. For convenience, instead of trying to write ON, OFF, OFF, ON, OFF, etc., people started abbreviating, using 1 for ON and 0 for OFF. Now they could write 10010, etc. This excited mathematicians quite a bit, since they'd always been trying to come up with uses for numbering systems other than base 10, which they say we have because of the number of protuberances on our hands. Surely many of you remember in 7th or 8th grade when they taught you the marvels of base 7, base 4, and other wonderful bases, and remember wondering why in the world they are doing that and who anywhere would care what $4 + 5$ was in base 6? (It's 13.) The best explanation I ever heard was the one about "If someone ever came up to you in a dark alley with a gun and told you to tell him the answer to 4 base $7 + 6$ base 7 or he'd shoot you, it would be good to know." I've tried that one on students too, and it didn't work with them either.

BITS AND BYTES

Well, now 1s and 0s come along, and mathematicians get all excited since those are the digits used in base 2, binary. You know, base 10 uses 0-9, binary uses 0-1. They get all excited and call these things bits, for Binary digITS. Clever. Then they figure that it would be nicer to have larger units, like base 10 has hundreds, thousands,

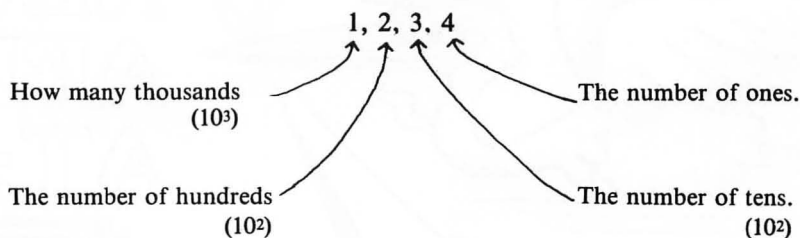
millions, and things like that, so they say that eight bits in strung together will be called a byte. Notice the unique spelling. Someone wanted to be cute, I guess. Depending on what computer you were talking about, there were also bigger units called words, which were two or four or some number of bytes. It varied from one machine to another. The IBM 1130 mentioned earlier was actually referred to as having 8K of storage, but those were two byte words. "K" is interesting, too. In usual talk, it means a thousand, but with computers it's actually 1,024. Strange number? Not really. It's actually 2 to the 10th power. Base 2, remember?

POWERS OF 2

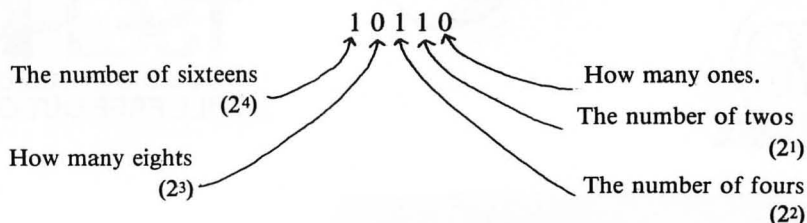
Powers of 2 appear often when using computers. Failing that, powers of 2 minus 1. $32767? 2$ to the 15th minus 1. $255? 2$ to the 8th minus 1. $127? 2$ to the 7th minus 1. It's just like powers of 10 in base 10. If you have room for 4 digits, the largest number you can use in 9999. 10 to the 4th minus 1. And on and on...

But people got tired of writing 1001101011001, etc. It's monotonous and boring. The number 9 in base 2 become 1001, for example, a bit more cumbersome in the writing. So they started looking for abbreviations. They first tried octal, which is base 8. To really understand it, though, you first have to know a little about binary. Any base is sort of like base 10. The place farthest right in base 10 represents the number of 1s. The second place is the number of 10s, which is 10 to the 1st power. The third place is the number of 100s, or 10 to the 2nd, followed by 1000s (10 to the 3rd), and so on. In base 2, the digit farthest right is still the number of 1s, but the 2nd digit is the number of 2s, which is 2 to the 1st. The third digit is 4s, or 2 to the 2nd. The fourth is 8s, or 2 to the 3rd, and so on. A definite pattern, and that's how it works. Figure 1 on the next page may help if the explanation wasn't clear enough.

Base 10 - Decimal



Base 2 - Binary



Base 16 - Hexadecimal

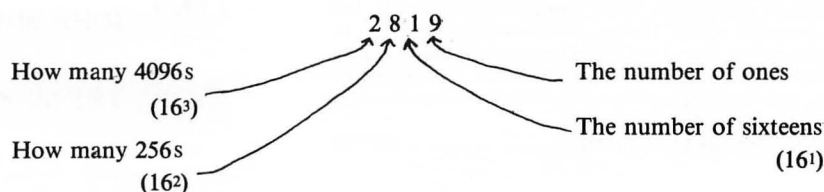


Figure 1

Well, since with three bits you can store the numbers 0 through 7 (000, 001, 010, 011, 100, 101, 110, 111), using sets of three bits you can refer to octal numbers. Octal, base 8, happens to use the digits 0-7, so a binary number like:

101 110 010 011 111 can be written an octal as: 5 6 2 3 7.

You match the sets of three binary digits with their actual values to get octal. Octal became a convenient shorthand for binary.

“But, alas,” someone proclaimed, “it is not really convenient because you cannot break a byte into groups of three! It has eight bits, so you must break it into groups of four!”

Well, with four bits you can represent the numbers 0 to 15 in binary. (15 is 1111 in binary. Try it.) “We can use base 16 for our abbreviations,” they said. This is all fine, but you know and I know that writing numbers like:

1 14 5 12

is going to get confusing. You get 114512, and suddenly you don't know whether it was:

11 4 5 12, or

1 14 5 1 2,

or what. So they improvised. They said “We'll count 1, 2, 3, 4, 5, 6,

Base 10 Decimal	Base 2 Binary	Base 8 Octal	Base 16 Hex- adecimal
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A

Figure 2

7, 8, 9, A, B, C, D, E, F, 10, 11, and so on.” Okay. So that's where hexadecimal comes in: base 16.

With it you can abbreviate all kinds of binary numbers.

0101 1001 1010 0111 becomes 5 9 A 7.

1011 1110 1110 1111 becomes B E E F.

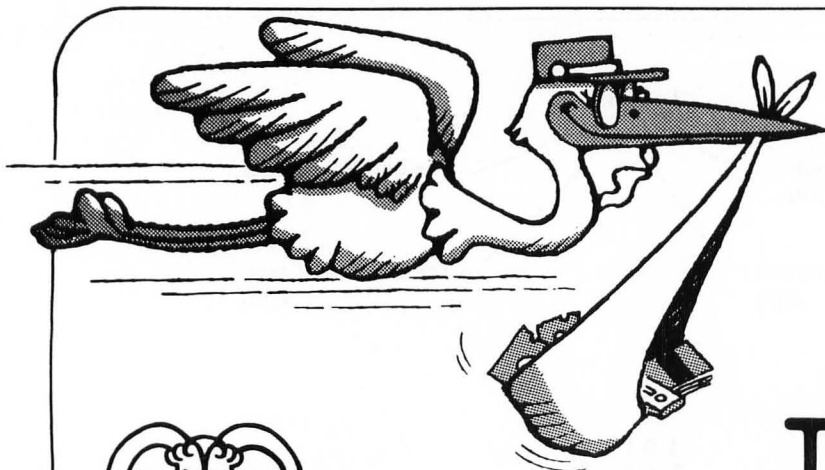
1111 0000 0000 1101 becomes F O O D.

NYBBLE NYBBLE

I'm getting rather hungry now. It's almost dinner time actually, so I think I'll leave off here. One interesting footnote is that at some point someone felt the need to name the unit of memory that can be stored by one hexadecimal digit: 4 bits, or a half byte. In ultimate cuteness, they named it a nybble. (Yes, the magazine by that name is spelled wrong, at least for what I believe is their intention.) I think nybble's a fairly new term; I couldn't find anything about who coined the term, although I did try. It wasn't even listed in my Penguin Dictionary of Computers (Penguin Books, 1970) (No joke, Dave). The Gold Star Award goes to anyone with clues as to its originator.

Dinner's awaitin'.





NEW ARRIVALS AT:



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



TIME LORD

The elders of Waldrom have supplied you with the hyperspace-worthy vessel Tardus, and commissioned you to eliminate the fallen Time Lord. Your resources include clones who will fight for you, the formidable CRASER weapons of the Tardus, and magic weapons such as Fusion Grenades and Borelian Matrix Crystals.

Apple 48K Disk Integer #47-191001D \$29.95

The Programmer's Guild

DRAGONQUEST!

In a desperate race against the sun you search for Smaegor, Monarch of Dragonfolk, who has kidnapped the Princess of the Realm. In a quest for honor and glory, you must search the land, seeking out the tools needed for the ultimate confrontation.

S-80 Tape #26-221001T \$15.95

S-80 Disk #26-221001D \$21.95

Brøderbund Software

GALAXY WARS

Full-color graphics, Machine Language sound effects and action make this an eerie challenge for you pilots anxious to maneuver your rocket through enemy fire and space fish to destroy enemy saucers.

Apple II 32K Disk Integer or Applesoft #47-230007D... \$24.95

TAWALA'S LAST REDOUBT

The cruel Emperor Tawala has been forced from his throne on the world of Galactia and has fled for his life to the planet of Farside, where he and a small band of adherents prepare to make their last stand. You are the rebel leader who must lead the assault against the Emperor's stronghold.

S-80 16K Tape Level II #26-230005T \$19.95

S-80 32K Disk #26-230005D \$24.95

Apple 48K Disk Applesoft #47-230005D \$29.95



JOHN WILEY & SONS, Inc.

MORE TRS-80 BASIC

This new interactive guide to programming in S-80 Level II BASIC teaches scores of practical applications — both recreational and educational. You'll get a clear demonstration of different methods to display graphics: file-handling techniques applied to both cassette and disk files plus: animation, games, and fun.

#65-233002B \$9.95

DATA FILE PROGRAMMING IN BASIC

With your own microcomputer, minimal knowledge of BASIC, and this easy-to-use guide, you can learn how to use data files to keep track of billings, customer inventory, and expenses, maintain mailing lists, process numerical and statistical information, and much more. Sample programs and lots of practical advice assist you every step of the way.

#65-233001B \$9.95

INTRODUCTION TO 8080/8085 ASSEMBLY LANGUAGE PROGRAMMING

Use your microcomputer to write fast, sophisticated, complex programs with the help of this guide. You'll learn what Machine Language is, how to perform routine operations, input/output, conditional commands, machine language logic stack routines, and more. Sample programs illustrate every technique and concept.

#65-233003B \$8.95

8080/Z80 ASSEMBLY LANGUAGE

Learn assembly language swiftly and easily by developing a powerful system monitor in a step-by-step approach. Ten appendices contain all the necessary material to write 8080 or Z80 Assembly Language programs.

#65-233004B \$9.95

DATA CAPTURE 3.0

Anything that appears on the screen can be captured, saved, printed, or edited. You can use "Data Capture" to compose text off-line for later transmission to another computer. Designed to be used with the Micromodem II.

Apple II Disk Applesoft #47-200073D \$29.95

ADVENTURE HINT BOOK

Having problems with one of Scott Adams' Adventures? Can't figure out how to get out of the bog or where the pharaoh's heart is? These and other clues are given in a special format designed to help you as little as possible but still let you solve the Adventure yourself.

#65-200082B \$7.95

CAPTURE/SIMON

Block your opponent's moving wall, but don't hit anything. You'll need fast reflexes to win this fast-moving Arcade game which features graphics and sound effects. Also includes the memory game "Simon".

S-80 Mod. 1 & 3 16K Tape Machine Language
#26-200054T \$12.95

S-80 Mod. 1 32K Disk Machine Language
#26-200054D \$17.95

SIMUTEK II

Simutek II Package One includes the following programs:
Graphicrek 2000 — Navigate the Enterprise to dock with the giant space stations as well as avoid Klingon torpedoes.

Invasion Worg — As general of Earth's forces in 3099 your job is to stop the Worg invasion and destroy their outposts.

Star Wars — Maneuver your space fighter deep into the nucleus of the Death Star, drop your bomb, then escape via the only exit.

Space Target — Shoot at enemy ships with your missiles.

Saucers — Can you be the commander to win the distinguished cross within the time limit?

S-80 16K Tape BASIC #26-200024T \$14.95

POKER TOURNAMENT

This program allows one person to play five card draw poker with five computer controlled opponents. The computer has bluff and anti-bluff functions and uses an attractive combination of graphics and alphanumerics.

S-80 Mod. 1 16K Tape #26-200093T \$12.95

Apple II 16K Tape Applesoft #47-200093T \$14.95

Apple II 16K Disk Applesoft #47-200093D \$20.95

ANGLE WORMS/CROLON DIVERSION

Players attempt to prevent growing worms from hitting an obstacle. Each worm may fire a projectile from its head to try to shorten its own length. For one or two players.

Also included is "Crolon Diversion", a space target game.

Atari 8K Tape #36-200092T \$9.95

TANK COMMAND

Two armies of tanks fight repeated battles until one army scores enough points to win the war. Players choose which tank to fire and the target is randomly selected from within the tank's range.

Apple II Tape #47-200033T \$9.95

'TWAS THE NIGHT BEFORE CHRISTMAS AND MATCH MAKER

A terrific Kid Venture version of the popular Yuletide poem with James Talley's special sounds and graphics. Perfect for your 4-10 year old. Also included is a memory game called "Match Maker". This game uses graphics and sound effects and is perfect for even non-readers.

S-80 Mod. 1 Tape #26-20083T \$14.95

SIX MICRO STORIES

This program offers an introduction to Interactive Fiction where the computer sets the scene with a fictional situation and you become a character in the story. You are an American spy in Hitler's Third Reich, the pilot of a doomed 747 and more. Now available for the Apple II.

Apple II Disk Applesoft #47-200021D \$14.95

E-Z SOUNDS

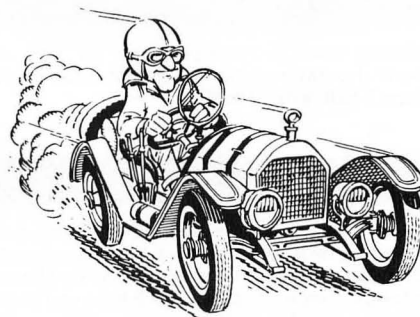
A sound routine for enhancing operation of BASIC programs. Easily patched into your program and executed upon command. No prior musical training is necessary and a short tutorial on music terminology is included. You may sell your own programs with "E-Z Sounds" routines, provided a credit line is given to "E-Z Sounds."

S-80 Mod. 1 16K Tape Machine Language
#26-200095T \$24.95

THE GREAT RACE

Try to finish the 600-mile race before your opponents, or before they stop you with wrecks or flat tires. Race against the computer or one to three players.

S-80 16K Tape BASIC #26-200035T \$9.95



PLANETOIDS

A real-time Hi-Res graphics game with super sound effects. Features three different skill levels — Novice, Average, and Expert where the asteroids are attracted to your ship.

Apple II or II +, 32K Tape Machine Language
#47-200015T \$14.95

Apple II or II +, 32K Disk Machine Language
#47-200015D \$19.95

LUNAR LANDER

Features a real-time lunar lander which you attempt to maneuver to safe landing sights. Great sound effects.

S-80 Mod. 1 & 3 16K Tape Machine Language
#26-200094T \$14.95
S-80 32K Disk Machine Language
#26-200094D \$20.95

LITTLE RED RIDING HOOD

In this Kid Venture, your child may interact in the story of Little Red Riding Hood. Cardboard cutouts mask all keys except the second row and the space bar to allow use by both readers and nonreaders. Includes musical sound effects and a play-along cassette tape.

S-80 Mod. 1 16K Tape BASIC #26-200030T \$14.95
Apple II 16K Tape BASIC #47-200030T \$19.95

LYING CHIMPS

The old game of "I Doubt It" or "Liar," only you play with four chimps who love to cheat. Excellent animated graphics with sound.

S-80 16K Tape BASIC #26-200037T \$9.95

STRIP DICE and CONCENTRATION

These sexually-oriented games are guaranteed to make for some very interesting parties. Humorous sound effects and graphics. NOTE: EXPLICIT SEXUAL LANGUAGE MAY BE OFFENSIVE TO SOME USERS.

S-80 16K Tape BASIC #26-200097T \$14.95
S-80 32K Disk BASIC #26-200097D \$20.95

SILVER FLASH PINBALL

This extremely realistic pinball simulation has four flippers, shake control, and even a tilt mechanism. Features bumpers, blockades, graphics, and sound effects.

S-80 Mod. I 16K Tape Machine Language
#26-200086T \$14.95
S-80 Mod. 1 32K Disk Machine Language
#26-200086D \$19.95

MUSICAL YAT-C

A self-scoring musical strategy game for one or more players which also has a certain degree of luck involved.

S-80 16K Tape BASIC #26-200090T \$12.95

OWL TREE

In this game you find a tree with nine playing positions. Eight are filled by owls and one by a bat. The object of the game is to attract owls to all nine positions by shooting the bats in the fewest number of plays. Each time you shoot a bat you will also scare off its neighbors which will be replaced by the opposite creature.

S-80 16K Tape BASIC #26-200034T \$9.95

MOUNTAIN SHOOT

Pick your own powder and your angle as you plug away over the mountain. Great sound and superb colors. Has special limited powder version for greater challenge.

Atari 16K Tape #36-200079T \$9.95

STAR TREK 3.5

This improved Star Trek version includes action, sound effects, 10% faster execution, up to 30% faster execution for long or short-range sensor scans, smoother command flow, multiple moves when using impulse engines during non-combat situations, and improved Klingon battle logic for a great challenge.

Atari 32K Tape #36-200025T \$14.95

DEFLECTION/SIMON SAYS

Deflection is a fast-moving Arcade game with simple rules and many skill levels. Also included is the memory game "Simon Says".

Atari Tape #36-200078T \$9.95

FROG

You are a hungry frog on a log in a pond. You're trying to catch your dinner on the fly. Great for ages 5 and up.

S-80 16K Tape #26-200055T \$9.95



BASKETBALL

One-on-one basketball against a friend or the computer. Steal the ball, duck around your opponent, and slant towards the basket for a lay up. The graphics are based on a three-dimensional depiction of a basketball court and dribbling sounds add to the realism.

S-80 Tape #26-234010T \$14.95
S-80 Disk #26-234010D \$20.95

SOFTWARE, ETC

DOUBLE ZAP II FOR NEWDOS/80

"Double ZAP II" is a major enhancement from Double ZAP I. It carries all of the features of D-Z I, as well as Automatic Density Recognition. Using this feature allows any density diskette to be used on any drive.

S-80 32K Disk 1-4 Disk Drives #15-275003D \$49.95

DOUBLE ZAP II FOR NEWDOS AND NEWDOS+

This Double Zap modifies the original NEWDOS (version 2.1) for operation with the Percom "Doubler". It also adds a mechanism to the DOS for managing different-sized drives in the system and adds some useful utilities as well. Contains Automatic Density Recognition.

S-80 32K Disk 1-4 Disk Drives #15-275002D \$39.95

ABOUT THIS ISSUE

Egg, egg, who's got the egg? Some weird guy with long ears showed up the other day and stashed an Easter egg on our cover. Can you find it? If so, write to the munchkins (that's us) and tell us where it is. We'll duly take note and mention you next time.

Well, let's see what lies between the covers this month. On the cover is the bridge that ate Schenectady, but has since retired to a harmless existence in northern Massachusetts. No teeth left, but the gums still show in spots. The program that goes with the photo is "IBC", written by the since-departed Missouri wonder, Phil Case. He flew south to work for Indenture Extranational, or somesuch.

And for those of you who aren't tired of sinking ships yet, there's the Apple version of the classic game of "Battleship", courtesy of our latest RAMwizard, Jon Voskuil. He's added a twist to it, now not only can you and a friend play, but four more friends (or the computer itself) can join in for a true free-for-all on the high seas. Avast ye lubbers! (A vast what!!!!? We don't even know what a small ye lubber looks like). And then there's a chance to get sunk instead of doing the sinking for you Atari folk: "Convoy" in yet another incarnation from those overachievers in Canada, Morris & Cope.

And as long as we're taking care of translations and such, we've included an Atari version of a game we published last month for the S-80 and Apple: "Strategy Strike". A tip of the stocking cap to Rich Bouchard, for another hard-fought battle with the translation poltergeists that seem to plague him.

Equations for all Occasions
Dept: That overworked Voskuil fellow maintains the pinnacle of a mathlete's aspirations with yet another installment of "Math Decathlon". Once you've used this gem to learn to recognize numbers, then you'll be ready for Peter Kirsch's venture into the boneyard: "Dominoes". It's a three-in-one package of the game that lent its metaphor to those that fear the

Red Menace, no relation to Dennis. Cousin Fred got so excited when he saw all those tiles up on the screen that he tried to see if he could tip one over and get the rest to fall...since then cousin Fred has been trying to glue back the pieces of the munchkin monitor, with little success. Our fearless publisher has vowed to keep him on bread and water until he succeeds.

After Fred finishes with the monitor we're going to let him build robots. Wanna try? It's easy, just key in the program "Robot Builder" and away you go. Your S-80 will try to build one faster than you can, which doesn't seem fair to us... the computer already knows more about robots than we do.

What's black and white, Hi-Res and odiferous all over? If you thought the answer was a newspaper printed on scented paper, you're wrong. It's "Skunk", a program by John Daoust, the fellow that brought you the upside down Apple "Darts" game a while back. This time he has given us a game that is an exercise in careful judgement and pushing your luck. We munchkins do well in this game as we always push our luck, right boss?

Plus, we've got the Penguin fellow continuing his exploration of what makes a computer tick. This time he examines all the different ways you can count. Elizabeth Barrett Browning tried that before, but she got hung up at 47. Mark's done better, he doesn't count that far, but boy does he do it in a lot of ways, at least until dinner reared its head.

That Yoho fellow takes a look at a couple of the newest kids on the Adventure block, and likes what he finds. No surprise. There are also articles inside about chaining in Applesoft (without getting sticky) and modifying string variables. Ms. Truckenbrod moves on to perspective in computer graphics and Ed Ting tells you what to do AFTER your program loads, with apologies to Sherry Taylor. Enjoy! 'Til next month, a merry munchkin farewell.



X-WING

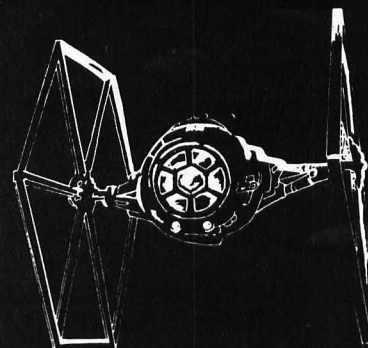


by
Chris Freund



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



TSE HARDWARE

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

MESSIAH



by William Morris and John Cope

"Messiah" is an ATARI program requiring 16K of memory.

In keeping with the time of year, here's a familiar piece of music, courtesy of George Frederick Handel, adapted for your Atari by our neighbors to the north, William Morris and John Cope. We find the combination of sound and graphics to be clever and most appropriate. As mid-April approaches load this into your computer and let it add to the atmosphere of the occasion.

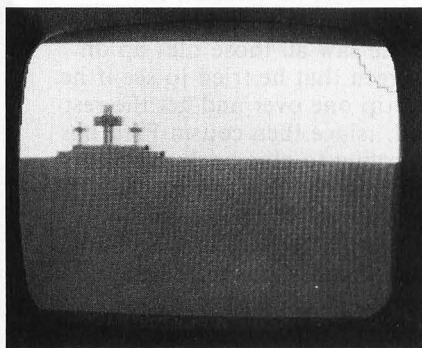
VARIABLES

J, K, L, M: Variables used to set the sound for each of the four voices and vertical position of note display.

Z: Horizontal plot position of the notes.

DOCUMENTATION

Lines 10-30: Title display.



Line 100: Set the color registers.
Line 110: Set the trap routine to end the chorus.

Line 120: Color the sky.

Line 130: Color the field.

Line 140: Draw the sun.

Line 150: Draw the hill.

Line 160-170: Draw the three crosses.

Line 200: Read data statements

and set the tone for each of the

four voices.

Line 210: Sets the horizontal screen position for each of four notes to be plotted. If Z is located at the extreme right boundary of the screen it is set back to the left and the luminescence of the sky is increased by 1.

Lines 211-214: Sets the vertical position for rests (note value of 0).

Line 220: Plots each of the four notes on the screen as well as an adjacent note to make the display more prominent. Dividing each note by 10 and adding 26 restricts the display to the lower portion of the screen.

Lines 230-240: Short delay loop.

Line 800: Sound routine.

Line 900: Trap routine. End of data error sends program here. All of the sound registers are turned off before starting again.

Lines 1000-1630: Data statements.

Lines 30200-30220: Routine to display mixed graphics mode of the title.

```

10 GOSUB 30200:POKE 87,2:POSITION 5,1:
? #6;"hallelujah"
20 POKE 87,1:POSITION 1,7:? #6;"hande
l's messiah":POKE 752,1
30 POKE 87,0:POSITION 5,14:? "(c) Wm.
Morris & J. Cope 1981":FOR Z=1 TO 300
0:NEXT Z
100 GRAPHICS 21:SETCOLOR 0,7,0:SETCOLO
R 1,12,0:SETCOLOR 2,15,10:SETCOLOR 4,0
,0
110 TRAP 900
120 COLOR 1:PLOT 79,20:DRAWTO 79,0:DR
AWTO 0,0:POSITION 0,20:POKE 765,1:XIO 1
8,#6,0,0,"S:"
130 COLOR 2:PLOT 79,47:DRAWTO 79,21:DR
AWTO 0,21:POSITION 0,47:POKE 765,2:XIO
18,#6,0,0,"S:"
140 COLOR 3:FOR Z=0 TO 8:READ X:PLOT X
,Z:DRAWTO 79,Z:NEXT Z
150 COLOR 2:PLOT 10,20:DRAWTO 30,20:PL
OT 13,19:DRAWTO 27,19
160 COLOR 4:PLOT 14,15:DRAWTO 14,18:PL
OT 13,16:DRAWTO 15,16:PLOT 25,15:DRAWT
O 25,18:PLOT 24,16:DRAWTO 26,16
170 PLOT 17,14:DRAWTO 22,14:PLOT 17,15
:DRAWTO 22,15:PLOT 19,12:DRAWTO 19,18:
PLOT 20,12:DRAWTO 20,18:Z=0
180 W=0
200 READ J,K,L,M:SOUND 0,J,10,4:SOUND
1,K,10,4:SOUND 2,L,10,4:SOUND 3,M,10,4
210 Z=Z+2:IF Z>78 THEN Z=2:W=W+2:SETCO
LOR 0,7,W
211 IF J=0 THEN J=10
212 IF K=0 THEN K=10
213 IF L=0 THEN L=10

```

```

214 IF M=0 THEN M=10
220 COLOR 3:PLOT Z,INT(J/10+26):PLOT Z
+1,INT(J/10+26):PLOT Z,INT(K/10+26):PL
OT Z+1,INT(K/10+26)
225 PLOT Z,INT(L/10+26):PLOT Z+1,INT(L
/10+26):PLOT Z,INT(M/10+26):PLOT Z+1,I
NT(M/10+26)
230 FOR T=1 TO 25:NEXT T
240 COLOR 2:PLOT Z,INT(J/10+26):PLOT Z
+1,INT(J/10+26):PLOT Z,INT(K/10+26):PL
OT Z+1,INT(K/10+26)
245 PLOT Z,INT(L/10+26):PLOT Z+1,INT(L
/10+26):PLOT Z,INT(M/10+26):PLOT Z+1,I
NT(M/10+26):GOTO 200
800 READ J,K,L,M:SOUND 0,J,10,4:SOUND
1,K,10,4:SOUND 2,L,10,4:SOUND 3,M,10,4
:FOR Z=1 TO 100:NEXT Z:GOTO 800
900 SETCOLOR 4,0,15:FOR Z=0 TO 3:SOUND
Z,0,0,0:NEXT Z:FOR Z=1 TO 2000:NEXT Z
:RUN
999 DATA 71,71,72,72,73,73,74,76,78
1000 DATA 53,72,42,217,53,72,42,217,53
,72,42,217,72,72,53,173
1010 DATA 64,81,53,162,72,85,53,217,0
,0,0,0,0,0,0
1020 DATA 53,72,42,217,53,72,42,217,53
,72,42,217,72,72,53,173
1030 DATA 64,81,53,162,72,85,53,217,0
,0,0,0,53,72,53,173
1040 DATA 53,64,40,162,53,72,42,217,0
,0,0,0,53,72,53,173
1050 DATA 53,64,40,162,53,72,42,217,0
,0,0,0,53,72,53,173
1060 DATA 57,81,47,193,53,85,72,217,53
,96,72,144,57,96,72,144

```

```

1070 DATA 53,85,72,217,53,85,72,217,0
,0,0,0,0,0,0
1080 DATA 47,72,57,144,47,72,57,144,47
,72,57,144,72,72,47,114
1090 DATA 42,72,53,108,47,72,57,144,0
,0,0,0,0,0,0
1100 DATA 47,72,57,144,47,72,57,144,47
,72,57,144,72,72,47,114
1110 DATA 42,72,53,108,47,72,57,144,0
,0,0,0,47,72,47,114
1120 DATA 42,72,53,108,47,72,57,144,0
,0,0,0,47,72,47,114
1130 DATA 42,72,53,108,47,72,57,144,0
,0,0,0,47,72,47,114
1140 DATA 42,72,53,108,47,72,47,114,53
,72,42,128,53,76,64,128
1150 DATA 57,72,47,144,57,72,47,144,0
,0,0,0,0,0,0
1160 DATA 72,72,72,144,72,72,72,144,72
,72,72,144,72,72,144
1170 DATA 64,128,64,128,64,128,64,128
,57,114,57,114,57,114,57,114
1180 DATA 53,108,53,108,108,108,108,21
7,53,108,53,108,53,108,53,108
1190 DATA 53,108,53,108,53,108,53,108
,57,114,57,114,57,114,57,114
1200 DATA 64,128,64,128,64,128,64,128
,64,128,64,128,64,128,64,128
1210 DATA 72,144,72,144,72,144,72,144
,0,0,0,0,47,72,72,114
1220 DATA 53,72,42,108,57,72,47,144,0
,0,0,0,47,72,47,114
1230 DATA 53,72,42,108,57,72,47,144,0
,0,0,0,47,72,47,114

```

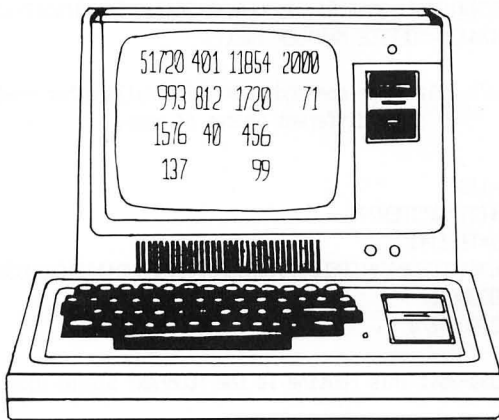
continued on next page

continued from previous page

1240 DATA 42,72,53,108,47,72,57,144,0,0,0,0,47,72,47,114
 1250 DATA 42,72,53,108,47,72,57,144,0,0,0,0,0,0,0
 1260 DATA 0,108,53,217,0,108,53,217,0,108,53,217,0,108,53,217
 1270 DATA 0,96,47,193,0,96,47,193,0,85,42,173,0,85,42,173
 1280 DATA 0,81,40,162,0,162,81,162,0,81,81,162,0,81,162,0,81,162,0,85,85,173,0,85,85,173
 1300 DATA 0,96,96,193,0,96,96,193,0,96,96,193,0,96,96,193,0,96,96,193,0,96,96,193
 1310 DATA 0,108,108,217,0,108,108,217,0,0,0,0,53,72,53,173
 1320 DATA 53,64,40,162,53,72,42,217,0,0,0,0,53,72,53,173
 1330 DATA 53,64,40,162,53,72,42,217,0,0,0,0,53,72,53,173
 1340 DATA 53,64,40,162,53,72,42,217,0,0,0,0,53,72,53,173
 1350 DATA 53,64,40,162,53,72,42,217,0,0,0,0,0,0,0
 1360 DATA 53,0,0,0,53,0,0,0,53,0,0,0,53,0,0,0,53,0,0
 1370 DATA 47,0,57,0,47,0,72,0,42,0,0,0,42,0,53,0
 1380 DATA 40,0,64,0,81,0,64,0,40,0,0,0,40,0,47,0

1390 DATA 40,0,57,0,40,0,72,0,53,0,53,0,53,72,53,0
 1400 DATA 47,81,53,0,47,96,53,0,47,72,57,0,47,72,57,144
 1410 DATA 53,72,53,173,53,72,53,217,0,85,35,0,53,108,40,0
 1420 DATA 57,0,72,144,72,96,72,144,0,14,72,144,57,144,72,144
 1430 DATA 53,0,64,128,53,85,64,128,47,96,57,114,47,81,57,114
 1440 DATA 72,85,53,108,72,85,108,217,53,64,53,108,57,72,53,108
 1450 DATA 64,76,53,108,64,96,53,108,0,72,57,114,47,72,57,114
 1460 DATA 53,72,64,108,57,72,64,108,53,76,64,108,53,76,64,108
 1470 DATA 57,72,72,144,57,72,47,144,47,0,57,0,57,96,72,144
 1480 DATA 0,72,0,173,72,85,53,217,53,0,53,0,72,85,53,108
 1490 DATA 0,96,0,114,57,114,47,144,47,0,57,0,57,0,72,0
 1500 DATA 0,108,53,0,72,108,53,0,42,108,53,0,53,108,53,108
 1510 DATA 0,96,47,114,57,96,47,144,53,85,42,0,72,85,42,108
 1520 DATA 0,81,40,128,64,162,81,162,47,81,40,0,47,81,40,193

1530 DATA 0,81,40,144,57,81,40,144,53,85,42,0,53,85,42,128
 1540 DATA 53,96,47,162,53,96,47,193,57,96,47,144,57,96,47,144
 1550 DATA 53,108,53,173,53,72,53,173,53,85,53,0,53,108,42,108
 1560 DATA 53,0,40,128,53,64,40,162,53,64,40,128,53,64,40,114
 1570 DATA 53,72,42,108,53,72,42,108,53,72,42,108,53,72,42,108
 1580 DATA 53,81,64,162,53,81,64,162,53,81,64,162,53,81,64,162
 1590 DATA 53,81,64,162,53,81,64,162,53,81,64,162,53,81,64,162
 1600 DATA 53,85,72,217,53,85,72,217,53,85,72,217,53,85,72,217
 1610 DATA 53,85,72,217,53,85,72,217,53,85,72,217,53,85,72,217
 1620 DATA 53,85,72,217,53,85,72,217,53,85,72,217,53,85,72,217
 1630 DATA 53,85,72,217,53,85,72,217,53,85,72,217,53,85,72,217
 30200 GRAPHICS 0:SETCOLOR 2,6,1:SETCOLOR 4,6,1: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



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.



COLUMN CALCULATOR 4.1

by David T. Gray

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

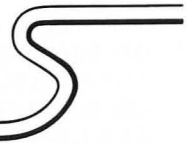
The information can be printed out on the line-printer in a compressed format at any stage in the development of a data base. Thus, it can be used as a finished report or as a copy of the worksheet to permit the filling in of additional data for later entry into the data base. The data base can be saved 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 \$39.95

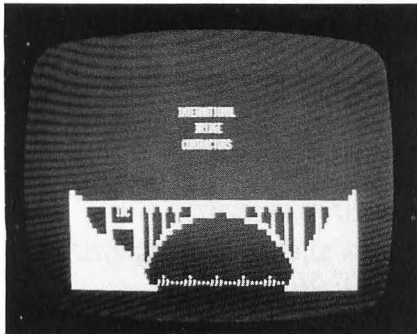

TSE HARDWARE
 6 South St. Milford, NH 03055 (603) 673-5144
 TOLL FREE OUT-OF-STATE 1-800-258-1790

INTERNATIONAL BRIDGE CONTRACTORS



by Phillip Case

"IBC" is an S-80 program which requires 16K of memory.



As you read the company newsletter, you can't believe your eyes. The chairman of the board is on vacation in Bermuda. While he's basking in the tropical sun, you're freezing in the frozen New Hampshire wilderness.

Your goals are set, but can you attain them? From your present position of a lowly office manager, you have to achieve and advance in position until you are Chairman of the Board of IBC, Inc.

International Bridge Contractors is a corporation which specializes in bridge construction all around the world. You must make the decisions governing how successful IBC, Inc. becomes, from hiring workers, to purchasing materials, to actually making bids for construction contracts.

Can you handle the requirements of running a large international corporation? Find out in IBC.

```

5 ' *****
  XX INTERNATIONAL BRIDGE CONTRACTORS  XX
  XX                                     XX
  XX          (C) COPYRIGHT 1979          XX
  XX          BY WILLIAM PHILLIP CASE     XX
6 ' *****

Line 10-130: Clear string space and initialize graphics.
10 CLEAR1050:CLS
20 L1$=CHR$(191)+CHR$(188)+CHR$(180)+CHR$(26)+STRING$(3,24)+STRIN
NG$(3,191)+CHR$(181)+CHR$(26)+STRING$(4,24)+STRING$(4,191)+CHR$(
189)+CHR$(180)+STRING$(2,176)+CHR$(26)+STRING$(8,24)+STRING$(9,1
91)+CHR$(189)+CHR$(188)+CHR$(180)+CHR$(176)+CHR$(26)
30 L1$=L1$+STRING$(13,24)+STRING$(14,191)+CHR$(188)+CHR$(176)+CH
R$(26)+STRING$(16,24)+STRING$(16,191)+CHR$(189)+STRING$(2,176)
40 L2$=CHR$(160)+STRING$(2,176)+CHR$(184)+CHR$(190)+CHR$(191)+ST
RING$(8,24)+CHR$(160)+CHR$(188)+STRING$(5,191)+STRING$(10,24)+CH
R$(160)+CHR$(184)+CHR$(188)+STRING$(7,191)+STRING$(12,24)+CHR$(1
76)+CHR$(190)+STRING$(10,191)+STRING$(14,24)+CHR$(184)
50 L2$=L2$+STRING$(13,191)+STRING$(16,24)+CHR$(176)+CHR$(190)+ST
RING$(14,191)
60 B1$=STRING$(14,176)+CHR$(26)+STRING$(9,24)+CHR$(191)+CHR$(32)
+"IBC"+CHR$(32)+STRING$(2,191)+CHR$(26)+STRING$(8,24)+CHR$(191)+
STRING$(5,131)+STRING$(2,191)+CHR$(26)+STRING$(2,24)+STRING$(2,1
91)+CHR$(26)+STRING$(2,24)+STRING$(2,191)+CHR$(140)
70 B2$=STRING$(11,176)+CHR$(26)+STRING$(10,24)+STRING$(2,191)+ST
RING$(3,32)+CHR$(191)+CHR$(32)+CHR$(160)+CHR$(191)+CHR$(26)+STRI
NG$(9,24)+STRING$(2,191)+STRING$(2,32)+CHR$(160)+CHR$(191)+CHR$(
26)+STRING$(6,24)+STRING$(2,191)+CHR$(26)+STRING$(3,24)
80 B2$=B2$+CHR$(139)+CHR$(191)
90 B3$=STRING$(32,176)+CHR$(26)+STRING$(32,24)+CHR$(171)+CHR$(12
9)+CHR$(32)+CHR$(171)+CHR$(129)+CHR$(32)+CHR$(171)+CHR$(161)+STR
ING$(2,176)+STRING$(5,143)+STRING$(3,131)+STRING$(4,143)+STRING$(
2,176)+CHR$(178)+CHR$(151)+CHR$(32)+CHR$(130)
100 B3$=B3$+CHR$(151)+CHR$(32)+CHR$(130)+CHR$(151)+CHR$(26)+STRI
NG$(32,24)+CHR$(170)+STRING$(2,32)+CHR$(186)+CHR$(140)+CHR$(134)
+CHR$(131)+CHR$(129)+STRING$(17,32)+STRING$(2,131)+CHR$(140)+CHR
$(181)+CHR$(144)+CHR$(32)+CHR$(149)+CHR$(26)+STRING$(32,24)
110 B3$=B3$+CHR$(170)+CHR$(140)+CHR$(131)+CHR$(26)+STRING$(3,24)
+CHR$(129)+CHR$(27)+STRING$(28,25)+CHR$(131)+CHR$(137)+CHR$(181)
120 W$(1)="....."
130 W$(2)="....."

```

Lines 170-190: Display introduction graphics.

```

170 PRINT@280,"INTERNATIONAL";PRINT@348,"BRIDGE";PRINT@409,"CONT
RACTORS"

```

```

180 PRINT@576,L1$;;PRINT@635,L2$;;PRINT@579,B1$;;PRINT@625,B2$;;
PRINT@593,B3$;
190 FORA=1T010;FORB=1T02;FORC=1T090;NEXTC;PRINT@915,W$(B);;NEXTB
,A;CLEAR400;U$="$$$$,###,###,##";LK$="$$$$###,##";A$=STRING$(6
3,42)

```

Lines 250-260: Initialize player variables.

```

250 CLS:INPUT"ENTER # OF PLAYERS (1-4)";P;IFP<1ORP>4THEN250
255 FORZ=1TOP;PRINT"PLAYER #";Z;" ENTER YOUR NAME";;INPUTNA$(Z);
C$(Z)=4000000;P$(Z)="LIGHT";PC(Z)=75000;PS(Z)=3000;PM(Z)=5000;B
$(Z)="LIGHT-MED";BC(Z)=210000;BS(Z)=100;EM(Z)=150;T$(Z)="MEDIUM"
;TC(Z)=300000;TS(Z)=1000;TM(Z)=2000;A$(Z)="MEDIUM"
260 AC(Z)=345000;AS(Z)=1200;AM(Z)=2200;C$(Z)="HEAVY";CC(Z)=52500
0;CS(Z)=2500;CM(Z)=3500;S$(Z)="HEAVY";SC(Z)=998000;SS(Z)=3000;SM
(Z)=6000;R$(Z)="OFFICE MANAGER";NEXT

```

Lines 270-290: Major command loop which calls other routines for different phases of game.

```

270 FORZ=1TOP
273 IFNA$(Z)="X"THEN290
275 TI(Z)=TI(Z)+1
280 GOSUB300;GOSUB400;GOSUB500;GOSUB3000;GOSUB4000;GOSUB5000;GOS
UB5500;GOSUB6000
290 NEXTZ;GOTO270

```

Lines 300-350: This routine is the 'COMPANY STATUS DISPLAY'.

```

300 CLS:PRINT"NAME: ";NA$(Z);TAB(30);"POSITION: ";R$(Z);PRINTA
$;PRINT"COMPANY STATUS DISPLAY";TAB(50);"TURN: ";TI(Z);PRINTA$;
FORX=9T056;SET(X,17);SET(X,22);SET(X,42);NEXTX;FORY=1T042;SET(9
,Y);SET(10,Y);SET(36,Y);SET(35,Y);SET(56,Y);SET(55,Y)
305 PRINT@259,"BRIDGE CONSTRUCTION STATUS:";
310 NEXT;PRINT@390,"TYPE:";PRINT@403,"COMPLETE";PRINT@518,"PON
TOON";PRINT@582,"SINGLE BEAM";PRINT@646,"TRUSS";PRINT@710,"AR
CH";PRINT@774,"CANTILEVER";PRINT@838,"SUSPENSION";
320 PRINT@534,B1(Z);PRINT@598,B2(Z);PRINT@662,B3(Z);PRINT@726
,B4(Z);PRINT@790,B5(Z);PRINT@854,B6(Z);
330 FORX=15651T016355STEP64;POKEX,191;NEXT;PRINT@295,"COMPANY AS
SETS STATUS:";PRINT@356,STRING$(27,42);PRINT@485,"NUMBER OF WO
RK CREWS=";CR(Z);PRINT@613,"TOTAL R&D=";USINGU$;RD(Z);PRINT@74
1,"MATERIALS=";USINGU$;M(Z);
335 PRINT@869,"CASH RES.=";USINGU$;C$(Z);
340 PRINT@970,"<PRESS ENTER>";
341 AA$=INKEY$;IFAA$=""THEN341
350 RETURN

```



```

Lines 400-450: This routine is the 'COMPANY PURCHASES DISPLAY'
400 CLS:PRINT"NAME: ";NA$(Z);TAB(30);"POSITION: ";R$(Z):PRINTA
$:PRINT"COMPANY PURCHASES DISPLAY";TAB(50);"TURN: ";TI(Z):PRINT
A$:PRINT:PRINT"1. HIRING OF 5 ADDITIONAL WORK CREWS      $ 1,0
00,000"
405 PRINT"2. PURCHASE OF CONSTRUCTION MATERIALS      $ 2,000,00
0"
410 PRINT"3. RESEARCH & DEVELOPMENT INVESTMENT      $ 5,000,00
0":PRINT:PRINTA$:PRINTTAB(21);"CASH RESERVE = ";:PRINTUSINGU$;C#
(Z):PRINTA$:PRINTTAB(16);"ENTER CHOICE (0 = CONTINUE)";:INPUTCH:
IFCH=0THENRETURNELSEIFCH<0ORCH>3THEN400
420 ONCHGOSUB430,440,450
425 GOTO400
430 CR(Z)=CR(Z)+5:C#(Z)=C#(Z)-1000000:RETURN
440 M(Z)=M(Z)+2000000:C#(Z)=C#(Z)-2000000:RETURN
450 RD(Z)=RD(Z)+5000000:C#(Z)=C#(Z)-5000000:RETURN

```

Lines 500-540: This routine is the 'SECRETARY REPORTS' routine
 Here random numbers are generated and DATA
 elements are are displayed affecting current
 cash amounts.

```

500 RESTORE:CLS:X=RND(40):PRINT"YOUR SECRETARY REPORTS:";PRINTA$
:IFX<11PRINT"NOTHING OF IMPORTANCE IS IN THE NEWS,":PRINTA$:PRIN
T:GOSUB2000:RETURN
501 X=X-10:FORRK=1TOX:READE$,MP:NEXT:PRINTE$:IFX<15THENC#(Z)=C#(
Z)-MPELSEC#(Z)=C#(Z)+MP
502 PRINTA$:PRINT"PRESENT CASH RESERVE = ";:PRINTUSINGU$;C#(Z):P
RINTA$:PRINT:GOSUB2000:RETURN
511 DATA"CONGRESS HAS JUST PASSED A TARIFF
WHICH COST THE COMPANY $1,000,000.",1000000
512 DATA"AN ACCOUNTING ERROR HAS JUST BEEN
DISCOVERED COSTING THE COMPANY $10,000.",10000
513 DATA"THE CHAIRMAN OF THE BOARD HAS JUST
ORDERED YOU TO SPEND $5000 FOR
A WILDLIFE CONSERVATION PROJECT.",5000
514 DATA"A FORMER EMPLOYEE HAS COLLECTED $100,000
IN AN INSURANCE CLAIM AGAINST THE COMPANY.",100000
515 DATA"A WORKER SLOWDOWN HAS COST THE COMPANY
$3000.",3000
516 DATA"THE HEALTH & SAFETY DEPT. HAS CONDEMNED
ONE OF YOUR REGIONAL OFFICES, IT COSTS
YOU $300,000 TO REBUILD.",300000
517 DATA"NEW GOVT. REGULATIONS JUST ADOPTED
REGARDING FOREIGN TRADE COSTS YOU
$2000.",2000
518 DATA"VANDALS HIT YOUR MAIN OFFICE CAUSING
$500 IN DAMAGE.",500
519 DATA"WORKER CARELESSNESS COST THE
COMPANY $1000.",1000
520 DATA"A LABOR STRIKE CAUSES $50,000 IN
LOST PRODUCTION.",50000
521 DATA"A LARGE PRODUCTION CRANE HAS BEEN
STOLEN IT COSTS $1,000,000 TO REPLACE.",1000000
522 DATA"A COMMUNICATIONS BREAKDOWN COST
THE COMPANY $7000 IN MISC. PROBLEMS.",7000
523 DATA"THE PRESENT CHAIRMAN OF THE BOARD HAS
DIED, IT COSTS $30,000 TO REPLACE HIM.",30000
524 DATA"A REGIONAL OFFICE HAS HAD A FIRE!!!!
IT COSTS $300,000 TO REPLACE.",300000
525 DATA"THE GOVERNMENT HAS CHANGED THEIR POSITION
TOWARDS COMPETITION IN THE INDUSTRY, IT
SHOULD NOT AFFECT OUR BUSINESS.",0
526 DATA"A FORMER CLIENT HAS GIVEN THE
COMPANY $20,000 BECAUSE
OF ITS PERFORMANCE ON THE
JOB.",20000
527 DATA"NEW LABOR CONTRACT ACTUALLY SAVES
THE COMPANY $50,000.",50000
528 DATA"COMPANY CAPTURES CORPORATE SPY AND
RECEIVES $50,000 IN REWARDS FROM
APPRECIATIVE COMPANIES.",50000

```

```

529 DATA"COMPANY MAKES EQUIPMENT PURCHASE ON FOREIGN
MARKET AND SAVES $2,000,000",2000000
530 DATA"COMPANY RESEARCH DISCOVERS NEW COMMUNICATIONS
DEVICE WHICH SELLS FOR $1,000,000.",1000000
531 DATA"ACCOUNTING DEPT. REPORTS PREVIOUS EARNINGS
UNDERSTATED, COMPANY GAINS $3000.",3000
532 DATA"GOVT. LOBBY FORCES LEGISLATION THROUGH CONGRESS
WHICH SAVES COMPANY $10,000.",10000
533 DATA"STOCKS WHICH THE COMPANY OWNS EARN DIVIDENDS
OF $800,000.",800000
534 DATA"NEW ACCOUNTING SYSTEM SAVES COMPANY
$4000.",4000
535 DATA"MAJOR COMPETITOR GOES BANKRUPT
COMPANY GAINS $50,000 IN NEW
REVENUES.",50000
536 DATA"COMPANY INSTALLS NEW COMPUTER SYSTEM
AND SAVES $50,000 IN LABOR.",50000
537 DATA"THERE HAS BEEN SOME PROTEST OF THE
COMPANY'S ACTIVITIES IN THE EVERGLADES,
HOWEVER, IT SHOULD NOT AFFECT BUSINESS.",0
538 DATA"GOVERNMENT LAUNCHES INVESTIGATION INTO
ALLEGED UNDERGROUND ACTIVITIES BY SOME
COMPANY OFFICIALS, SHOULD NOT AFFECT BUSINESS.",0
539 DATA"HER DAUGHTER HAS JUST HAD A BABY, THIS
SHOULD NOT AFFECT BUSINESS.",0
540 DATA"SHE IS FEELING POORLY AND WISHES TO GO HOME.",0
  Lines 2000-2001: INKEY$ wait routine.
2000 PRINTTAB(28);"<PRESS ENTER>"
2001 AA$=INKEY$:IFAA$=""THEN2001ELSERETURN
  Lines 3000-3028: This routine is the 'R & D Phase'.
3000 CLS:PRINT"NAME: ";NA$(Z);TAB(30);"POSITION: ";R$(Z):PRINT
A$:PRINT"RESEARCH & DEVELOPMENT STATUS";TAB(50);"TURN: ";TI(Z):
PRINTA$:PRINT:PRINT
3010 X=RND(30):X#=#X*1000000:IFRD(Z)>X#THEN3020
3015 PRINT@455,"RESEARCH & DEVELOPMENT REPORTS NO PROGRESS THIS
TURN,":PRINT:GOSUB2000:RETURN
3020 X=RND(6):PRINT"RESEARCH & DEVELOPMENT REPORTS A COST BREAK
THROUGH,":PRINT:PRINT"THE COST OF A ";ONXGOSUB3023,3024,3025,302
6,3027,3028:GOSUB2000:RD(Z)=RD(Z)-RND(RD(Z)/5000000)*5000000:RET
URN
3023 PRINT"PONTOON BRIDGE HAS GONE FROM
";PC(Z);" TO ";PC(Z)*.75;" PER 100/FT,":PC(Z)=PC(Z)*.75:RETURN
3024 PRINT"SINGLE BEAM BRIDGE HAS GONE FROM
";BC(Z);" TO ";BC(Z)*.75;" PER 100/FT,":BC(Z)=BC(Z)*.75:RETURN
3025 PRINT"TRUSS BRIDGE HAS GONE FROM
";TC(Z);" TO ";TC(Z)*.75;" PER 100/FT,":TC(Z)=TC(Z)*.75:RETURN
3026 PRINT"ARCH BRIDGE HAS GONE FROM
";AC(Z);" TO ";AC(Z)*.75;" PER 100/FT,":AC(Z)=AC(Z)*.75:RETURN
3027 PRINT"CANTILEVER BRIDGE HAS GONE FROM
";CC(Z);" TO ";CC(Z)*.75;" PER 100/FT,":CC(Z)=CC(Z)*.75:RETURN
3028 PRINT"SUSPENSION BRIDGE HAS GONE FROM
";SC(Z);" TO ";SC(Z)*.75;" PER 100/FT,":SC(Z)=SC(Z)*.75:RETURN
  Lines 4000-4020: This routine displays bridge specs.
4000 CLS:PRINT"NAME: ";NA$(Z);TAB(30);"POSITION: ";R$(Z):PRINT
A$:PRINT"PRESENT BRIDGE SPECIFICATIONS DISPLAY";TAB(50);"TURN:
";TI(Z):PRINTA$:PRINT" TYPE:";TAB(15);"TRAFFIC:";TAB(26);"COST/
100FT,":TAB(39);"SAFE DIST,":TAB(51);"MAX. DIST."
4010 PRINT"PONTOON";TAB(15);P$(Z);TAB(26);USINGLK$;PC(Z);:PRINT
AB(40);PS(Z);TAB(53);PM(Z)
4011 PRINT"SINGLE BEAM";TAB(15);B$(Z);TAB(26);USINGLK$;BC(Z);:PR
INTTAB(40);BS(Z);TAB(53);BM(Z)
4012 PRINT"TRUSS";TAB(15);T$(Z);TAB(26);USINGLK$;TC(Z);:PRINTTAB
(40);TS(Z);TAB(53);TM(Z)
4013 PRINT"ARCH";TAB(15);A$(Z);TAB(26);USINGLK$;AC(Z);:PRINTTAB(
40);AS(Z);TAB(53);AM(Z)
4014 PRINT"CANTILEVER";TAB(15);C$(Z);TAB(26);USINGLK$;CC(Z);:PRI
NTTAB(40);CS(Z);TAB(53);CM(Z)
4015 PRINT"SUSPENSION";TAB(15);S$(Z);TAB(26);USINGLK$;SC(Z);:PRI
NTTAB(40);SS(Z);TAB(53);SM(Z)

```

continued on next page

continued from previous page

```
4016 PRINTA$:PRINT"CASH RESERVE =" ;USINGU$;C$(Z);:PRINTTAB(35);"  
MATERIALS =" ;USINGU$;M(Z):PRINTA$  
4020 GOSUB2000:RETURN
```

Lines 5000-5050: This routine is the contract bidding phase.

```
5000 CLS:PRINT"NAME: ";NA$(Z);TAB(30);"POSITION: ";R$(Z):PRINT  
A$:PRINT"CONTRACT BIDDING PHASE";TAB(50);"TURN: ";TI(Z):PRINTA$  
:PRINT"A CLIENT YOU ARE DEALING WITH NEEDS A BRIDGE WITH THESE S  
PECS.":PRINTA$  
5001 X=RND(6);ONXGOTO5002,5003,5004,5005,5006,5007  
5002 X$="PONTOON":GOTO5008  
5003 X$="SINGLE BEAM":GOTO5008  
5004 X$="TRUSS":GOTO5008  
5005 X$="ARCH":GOTO5008  
5006 X$="CANTILEVER":GOTO5008  
5007 X$="SUSPENSION":GOTO5008  
5008 IFX=1THENY=RND(PM(Z)):C=1:ELSEIFX=2THENY=RND(BM(Z)):C=2:ELS  
EIFX=3THENY=RND(TM(Z)):C=3:ELSEIFX=4THENY=RND(AM(Z)):C=4:ELSEIFX  
=5THENY=RND(CM(Z)):C=5:ELSEY=RND(SM(Z)):C=6  
5009 IFX=1THENX=75000:M=5:F=PC(Z):ELSEIFX=2THENX=210000:M=8:F=BC  
(Z):ELSEIFX=3THENX=300000:M=15:F=TC(Z):ELSEIFX=4THENX=345000:M=2  
5:F=AC(Z):ELSEIFX=5THENX=525000:M=34:F=CC(Z):ELSEX=998000:M=50:F  
=SC(Z)
```

```
5010 PRINTTAB(10);"TYPE = ";X$;TAB(40);"LENGTH = ";Y;"FT.":PRINTA  
$:PRINT"YOUR COST FOR THIS BRIDGE IS ";USINGU$;Y*F/100:PRINTA$  
5015 IFCR(Z)<(Y/100)*C THENPRINT"YOU DO NOT HAVE ENOUGH WORK CREW  
S FOR THIS BRIDGE.":PRINT:GOSUB2000:RETURN  
5020 IFM(Z)<(Y/100)*M*10000 THENPRINT"YOU DO NOT HAVE THE CONST  
RUCTION MATERIALS TO BUILD THIS BRIDGE.":PRINT:GOSUB2000:RETURN  
5025 INPUT"WHAT PRICE DO YOU WISH TO BID FOR THIS BRIDGE";BD:PRI  
NTA$:IFBD>(RND(11)+1)*(X*Y/100)*.9 THENPRINT"YOUR PRICE WAS TOO  
HIGH, CUSTOMER TAKES BUSINESS ELSEWHERE.":PRINT:GOSUB2000:RETUR  
N  
5030 PRINT"YOU GET THE CONTRACT AND MAKE ";USINGU$;BD-(F*Y/100):
```

```
C$(Z)=C$(Z)+BD-(F*Y/100):M(Z)=M(Z)-(Y/100)*(M*10000):IFM(Z)<0THE  
NM(Z)=0  
5040 IFX$="PONTOON"THENB1(Z)=B1(Z)+1ELSEIFX$="SINGLE BEAM"THENB2  
(Z)=B2(Z)+1ELSEIFX$="TRUSS"THENB3(Z)=B3(Z)+1ELSEIFX$="ARCH"THENB  
4(Z)=B4(Z)+1ELSEIFX$="CANTILEVER"THENB5(Z)=B5(Z)+1ELSEB6(Z)=B6(Z  
) +1  
5050 GOSUB2000:RETURN
```

Lines 5500-6020: This routine checks for advancement in position.

```
5500 X#=RND(300):IFC$(Z)>0 THENRETURNELSEC$(Z)=C$(Z)*1.5:IFC$(Z)  
<X*1000000*-1 THENGOSUB9000  
6000 IFC$(Z)>50000000ANDCB(Z)<5 THENR$(Z)="DISTRICT MANAGER":CB(Z  
)=5:ELSEIFC$(Z)>100000000ANDCB(Z)<10ANDCB(Z)>4 THENR$(Z)="REGIONA  
L SUPERVISOR":CB(Z)=10:ELSEIFC$(Z)>150000000ANDCB(Z)<15ANDCB(Z)>  
9 THENR$(Z)="CORPORATE ADVISOR":CB(Z)=15  
6010 IFC$(Z)>200000000ANDCB(Z)<20ANDCB(Z)>14 THENR$(Z)="COMPANY R  
EPRESENTATIVE":CB(Z)=20:ELSE IFC$(Z)>300000000ANDCB(Z)<30ANDCB(Z  
)>19 THENR$(Z)="V.P. OF PRODUCTION":CB(Z)=30:ELSEIFC$(Z)>40000000  
0ANDCB(Z)<40ANDCB(Z)>29 THENR$(Z)="COMPANY PRESIDENT":CB(Z)=40  
6020 IFCB(Z)=50 THEN7000:ELSEIFC$(Z)>500000000ANDCB(Z)=40 THENCB(Z  
)=50:RETURN:ELSERETURN  
6999 ' WINNER!!!!  
7000 CLS:PRINT"
```

" ;TAB(16);NA\$(Z);" HAS JUST BEEN DECLARED THE

CHAIRMAN OF THE BOARD

(<GAME IS OVER>):FORA=1T02000:NEXT:R

UN

Line 9000: Bankrupt routine.

```
9000 CLS:PRINTTAB(15);"UN FORTUNATE NEWS ";:PRINT  
AB(15);STRING$(33,45):PRINT:PRINT:PRINTTAB(10);NA$(Z);" HAS GONE  
BANKRUPT AND IS OUT OF THE GAME.":NA$(Z)="X":PRINT:GOSUB2000:FO  
RA=1T0F:IFNA$(A)<>"X" THENRETURNELSENEXT  
9010 CLS:PRINTTAB(15);"NO MORE ACTIVE PLAYERS, GAME ENDS.":PRINT  
:PRINT:GOSUB2000:RUN
```

S-80 ONE LINERS

```
1 DEFINTJ,N:A$=CHR$(RND(158)+33):B$=CHR$(RND(158)+33):FORN=1T014  
:J=ABS(N-7):PRINTTAB(7-J)A$TAB(8+J)B$TAB(22-J)A$TAB(23+J)B$TAB(3  
7-J)A$TAB(38+J)B$TAB(52-J)A$TAB(53+J)B$:NEXT:GOTO1
```

Brian Schipper
Holland, MI

```
10 RANDOM:FORZ=0T010:A(Z)=RND(127):B(Z)=RND(47):NEXTZ:FORY=1T010  
:SET(A(Y),B(Y)):SET(A(Y),47-B(Y)):SET(127-A(Y),B(Y)):SET(127-A(Y  
,47-B(Y)):NEXTY:R=RND(2):IFR=1 THENGOTO10 ELSEFORU=1T0250:NEXTU:  
CLS:GOTO10
```

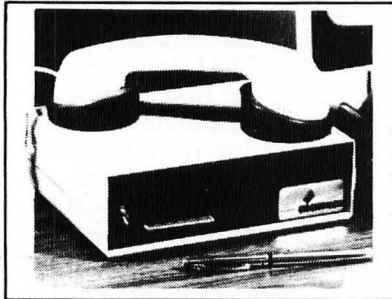
Mark Nelson
Palmyra, WI

```
0 PRINT@543,CHR$(190):FORT=1T050:NEXT:PRINT@543,CHR$(189):FORT=1  
T050:NEXT:PRINT@543,CHR$(183):FORT=1T050:NEXT:PRINT@543,CHR$(159  
) :FORT=1T050:NEXT:PRINT@543,CHR$(175):FORT=1T050:NEXT:PRINT@543,  
CHR$(187):FORT=1T050:NEXT:RUN
```

Brian Berkebile
Amherst, NH



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.\$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\$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.\$79.95

ST80* III

ST-80* D with extra utility programs. 32K disk program.\$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.

COMPUTER GRAPHICS

Perspective Drawing

by Joan R. Truckenbrod

Perspective drawing can be constructed with the aid of the microcomputer to show an object as it appears from different viewing points in space. The object can be viewed from the top, side, or bottom. For example, a figure or object can be tilted up to view the bottom or tilted down to view the top. The view of the figure in a perspective drawing is determined by the orientation of the figure to the X, Y and Z axes. In constructing a perspective it is helpful to be familiar with the effects of these rotations on the appearance of the figure.

Sequential rotations of an object in space on these three axes were illustrated in the February and March issues of **SoftSide**. The program that rotates three-dimensional figures in space and thus allows you to create perspective drawings, is listed and described in the February issue. This program is a valuable tool for creating three-dimensional drawings of a figure that are sometimes difficult and always time-consuming to construct by hand.

Objects drawn in perspective can be represented by lines, planes, or volumes. Stick figures are used for quick three-dimensional sketches of an object. The examples in Figure 1 illustrate the word LINE drawn in line with various X, Y and Z rotations in space. Figure 2 illustrates the use of two-dimensional planes to represent the same figure. The change in the figure's orientation in space is obvious here as the length and width of the planes appear to change as the figure is rotated. Objects can also be represented as volumes as shown in Figure 3. In these examples the object has height, width and depth and is rotated in three dimensions. Volumes require more lines to define the object and consequently are more complex figures. Thus, when these wire frame figures are rotated in space, they can become difficult to visualize due to the overlap of lines. Since this perspective program does not

remove the hidden lines, all of the lines in the figure are drawn. One technique for clarifying these more complex drawings is to introduce shading or color to differentiate various planes in the figure. Figure 3f shows an object in which three different gray values were used to identify the three frontal planes in order to clarify the object. In drawings of this complexity, gray values or different colors help the viewer understand the drawing. When using complex figures which require a large number of coordinate points, it is necessary to include a LOMEM:16500 statement in the program so that the program uses memory space other than that of the graphics screen and does not interfere with the drawing.

There are advantages and disadvantages to using perspective drawing techniques to illustrate an object. Perspective drawings can be used to realistically describe an object. These drawings can emphasize one prominent feature of an object or can show different views of an object. The appearance of the figure can be enhanced or distorted as various aspects of the figure can appear to be stretched or compressed. In various perspective views the appearance of the figure may become distorted or ambiguous, as is evident in some of the examples shown here. Care must be taken in perspective drawings to minimize distortion and to present the object in the clearest manner possible. Since the orientation of a figure in a perspective drawing is specified in this program by a combination of X, Y and Z rotations, it is helpful to be familiar with the effect of X, Y and Z rotations independently, and in combination, so that you may create the desired view in your perspective drawings. Rotation of a figure around the X axis will provide views of the bottom or top of an object, together with the front and/or one of the sides. If you would like to show a feature or detail on the top or bottom planes of the figure, use an X rotation. An X rotation of zero or 180 degrees shows the horizontal planes (top and bottom) as straight horizontal lines. 90 degree X

rotation shows the top of the figure as a vertical plane, and a 270 degree X rotation shows the bottom of the figure as a vertical plane. An X rotation reduces the apparent height or vertical dimension of a figure.

Consequently it is best to use a 20-45 degree X rotation to show the figure realistically. Figures with an emphasis on the top plane are illustrated in figures 2a, 2c, 3b and 3f. The bottom of an object can be shown realistically by employing a 315-340 degree rotation on the X axis. In order to present a realistic three-dimensional view of an object it is best to rotate the figure on at least two axes. The figure can be rotated on the Y axis to show the side of the figure with the front, or the side and the back of the figure. A Y rotation will show the right side of the figure with the front as it is rotated between zero and 90 degrees on the Y axis. This side and the back of the figure will be in view as the figure is rotated between 90 and 180 degrees on the Y axis.

Between 180 and 270 degrees the back and the left side of the figure will be in view, and between 270 and 360 degrees the left side and the front will be in view. In appearance a Y rotation can reduce the width or the horizontal dimension of the figure. A 20 degree Y rotation is illustrated in figure 3a, a 30 degree rotation in figure 2a, and a 60 degree Y rotation is illustrated in figure 3g. Rotations on the Y axis create the effect that the object is either coming towards the view or projecting back into space away from the viewer. Combinations of X and Y rotations can be used to create effective perspective drawings as illustrated in figure 2c. Since rotations around the Z axis turn the object around in a circle in the same plane as the paper or the video screen, they can be used to give direction to the figure. As shown in figures 1c, 2b and 3c, a 30 degree Z rotation points the figure down towards the lower right-hand corner. A 90 degree rotation on the Z axis points the figure directly towards the bottom of the screen, as in figure 1f. Combinations of all three types of

rotation can be used to create unusual perspective views. For example, figures can be drawn upside down as in figures 2e and 3d, backwards as is shown in figure 2f, or both upside down and backwards as in figure 3e. In constructing your own perspective drawings, experiment with various orientations of your own object in the manner demonstrated here.

**FIGURES CONSTRUCTED WITH LINES:
ROTATION IN THREE DIMENSIONAL SPACE**



Figure 1a. 330 Z, 30 Y, 0 X

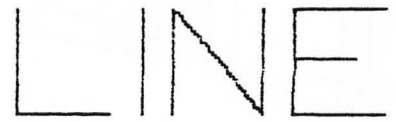


Figure 1b. 0 Z, 0 Y, 0 X

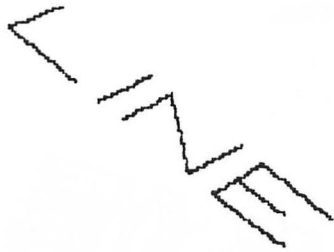


Figure 1c. 30 Z, 30 Y, 60 X

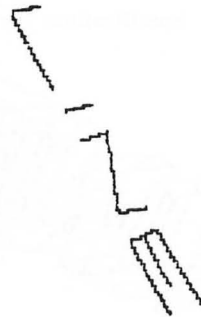


Figure 1d. 30 Z, 60 Y, 60 X



Figure 1e. 0 Z, 60 Y, 60 X



Figure 1f. 90 Z, 60 Y, 30 X



Figure 1g. 180 Z, 60 Y, 30 X



Figure 1h. 0 Z, 20 Y, 180 X

PERSPECTIVE DRAWINGS OF FIGURES CONSTRUCTED WITH PLANES

Specifications for Z, Y and X rotations.

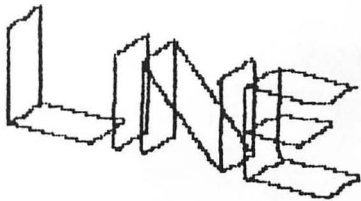


Figure 2a. 0 Z, 30 Y, 30 X Rotation

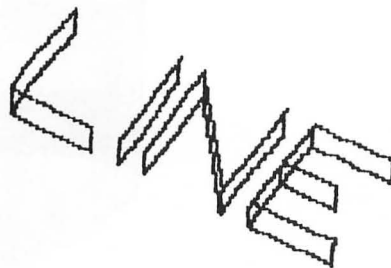


Figure 2b. 30 Z, 0 Y, 30 X Rotation

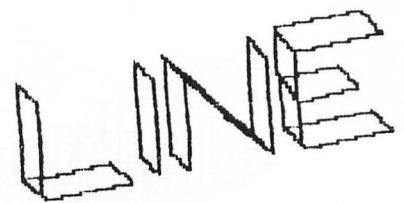


Figure 2c. 0 Z, 340 Y, 30 X Rotation

continued on next page

continued from previous page

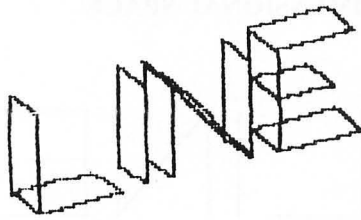


Figure 2d. 0 Z, 330 Y, 30 X Rotation



Figure 2e. 180 Z, 20 Y, 30 X Rotation

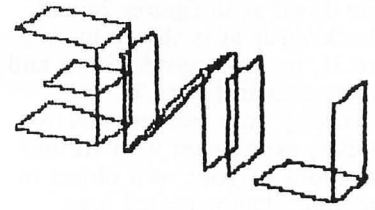


Figure 2f. 0 Z, 210 Y, 30 X Rotation

PERSPECTIVE DRAWINGS OF THREE - DIMENSIONAL FIGURES

Specifications for X, Y and Z rotations.

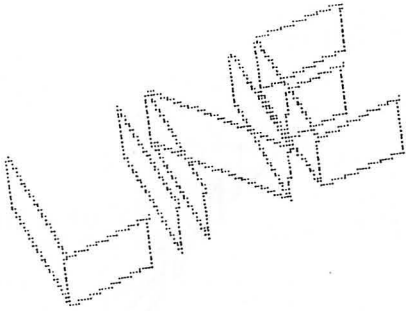


Figure 2g. 340 Z, 0 Y, 45 X Rotation

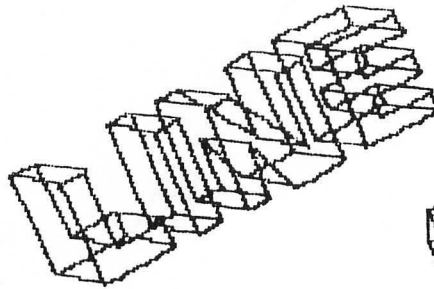


Figure 3a. 330 Z, 20 Y, 0 X Rotation

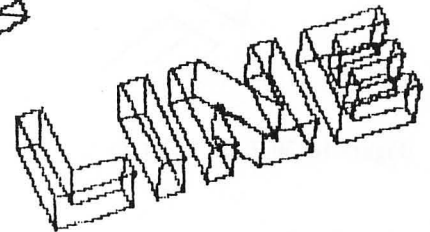


Figure 3b. 340 Z, 0 Y, 30 X Rotation

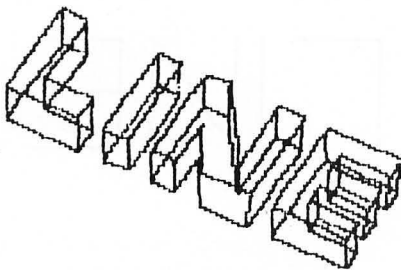


Figure 3c. 30 Z, 0 Y, 30 X Rotation

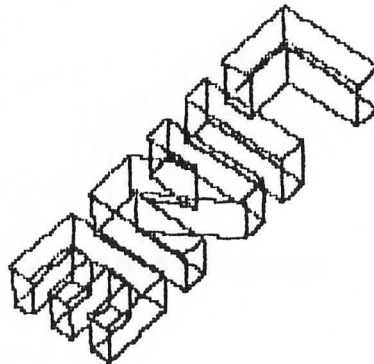


Figure 3d. 130 Z. 0 Y, 30 X Rotation

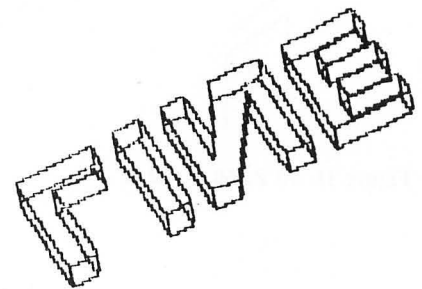


Figure 3e. 200 Z, 180 Y, 20 X Rotation

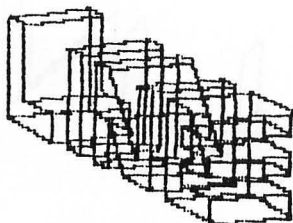
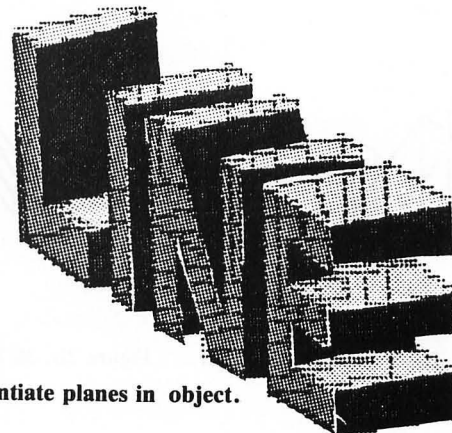


Figure 3f. 0 Z 60 Y, 20 X Rotation

Shading used to differentiate planes in object.



SAY YOHO

by Scott Adams

This month we will take a look at two milestone programs recently released for the small computers. "Zork"™ and "Classic Adventure"™.

I had the privilege to play a version of "Zork"™ (back when I earned an honest living as a programmer at Stromberg Carlson) on a large DEC PDP machine. The credits at the beginning said it was a wildcat conversion by a slightly crazed DEC employee who wished to remain anonymous as he feared for the sanctity of his job.

"Zork"™ was an ambitious undertaking, where Crowther & Woods had blazed a trail, this program put in a paved highway. Gone was the restriction of one or two word baby talk sentences and in its place you could speak full English! (Though, to be honest, the format looked for was "verb, object, indirect object, such as: "Hit the hammer with the nail".) Also there appeared a whole new and different world to conquer, complete with mind-boggling puzzles, evil denizens, and, of course, fabulous treasures.

Some of the problems, I'm afraid to say, were difficult only because of errors made in the conversion to the PDP, but on the whole the program was as addicting as "Original Adventure" was.

Well now "Zork"™ is available for use on your S-80 Model I or Apple II. It does require disk, (from which it makes frequent access) but as you descend deep into the bowels of the underground empire you will quickly forget you even have a disk drive! I had only a short time to play before deadline on my column, but the beginning play appears to be true to the original version. I have heard that a few minor alterations were made to get it to fit on a mere 90K diskette but I'm sure that if you like Compunovels you'll love "Zork"™! "Zork"™ is written by Infocom and is published by Personal Software and should be available wherever fine software is sold. I highly recommend it.

The other piece of major Compunovel software that was just

released is "Classic Adventure"™. This is, as the name implies, yet another version of Crowther & Woods "Original Adventure" for the Apple II. Why another version, you ask? Well this is the first version to say it is complete and totally unmodified from Crowther & Woods' version. Everything is there, from Breathtaking View to the final Grand Master Game! All other versions of "Original Adventure" have left out locations, sections, descriptions, and many have added embellishments which did not exist in the original. Not so in "Classic Adventure"™. It is complete right down to the "save" feature. And the second and even more stunning piece of news is it does not require a disk to run! The entire program is in Machine Language and through heavy use of Page 0 calls and text compression manages to fit entirely in 48K. You can purchase it on tape or disk. Each version can be transferred either to tape or disk, and will also run on 3.2 or 3.3. The best part is the price: \$14.95 for tape and \$20.95 for disk. So even if you have already played another version of Adventure, it is but a minor investment to have a true classic (remember, no disk accesses). I recommend getting the tape version and transferring it yourself to disk as it only takes six minutes to do originally and saves you \$6. "Classic Adventure"™ is written by John Rausch and published by Adventure International, a division of Scott Adams, Inc.

For those of you who haven't noticed, the computer shows are starting to make the rounds again. If you get a chance to go to one, stop by the AI booth and say hi. If you live in California or can get there, the one show that promises to be the biggest and best this year is the annual San Francisco Computer Faire. It is held the first week in April in Brooke's Hall and is so large that the exhibits are also downstairs in the parking garage under the plaza. This show attracts exhibitors from as far away as England and Japan. If you can, don't miss this one.

So until next month I'll just say Happy Adventuring (and Zorking) and YOHO...



SPECIAL DELIVERY with EXTRACT

A 100% Machine Language Word Processor from:
Quality Software Distributors

We can't stop improving and expanding the capabilities of your TRS-80. By using SPECIAL DELIVERY with EXTRACT and either Electric Pencil or Radio Shack's Scripsit you can get even more out of your computer. From just one package you will get all this:

MAILFORM

Create MAILFILE: The ONLY complete name and address list entry/editor program written in machine language.

MAILRITE

Print letters written with either the Electric Pencil or Radio Shack's Scripsit inserting information from a MAILFILE into the letter for personalizing and addressing.

EXTRACT

Take out information from the MAILFORM, the machine language mailing list. Find the names you need by Zip Code, Street Address, Gender, Age or any other way!

SORT

In-Memory sort on an entire address list using any field as the key.

LABEL

Prints labels from MAILFILE.

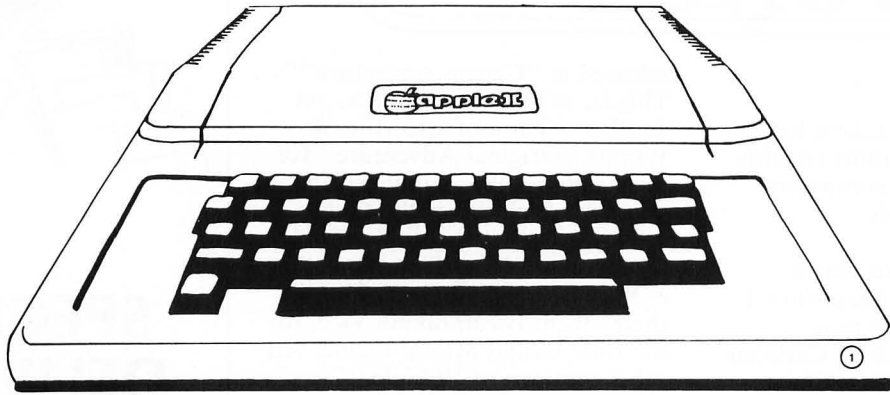
CONVERT

Make MAILFILE from RS mail list.

SPECIAL DELIVERY will run on your S-80 with TRSDOS, NEWDOS or any other TRS-like DOS.

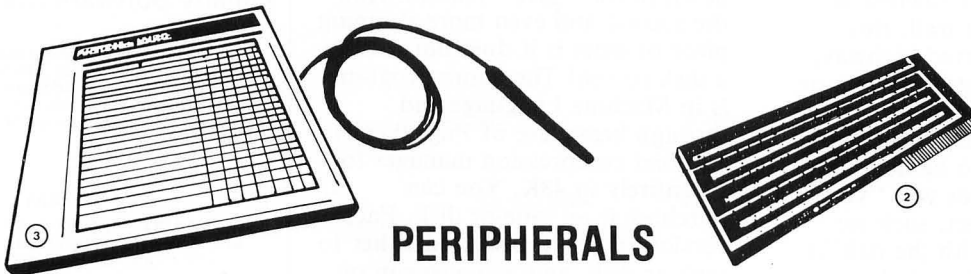
\$125.00


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



COMPUTERS

- ① 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).....\$579.00
 APPLE II Disk Drive (#47-005).....\$439.00
 LOBO Disk Drive (#47-3101).....\$379.00
 MICROSOFT Z-80 SoftCard (#47-80).....\$279.00
 MICROSOFT RAMCard (#47-81).....\$169.00
 AppleSoft BASIC Language Card (#47-ASC).....\$195.00
 Integer BASIC Language Card (#47-IC).....\$195.00
 PASCAL Language Card (#47-PAS).....\$469.00
 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 Chasis (#47-MH024).....\$609.00
 HARDSIDE Memory Upgrade Kit (#5-1102).....\$59.00
 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
 ② CCS Calendar/Clock Module (#47-7424A).....\$125.00
 LYNX Communications Interface (#19-85).....\$239.00
 ③ 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


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: 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.

BASICALLY SPEAKING

THE MOST UNUSUAL PROGRAMMING CONTEST YOU'VE EVER HEARD OF

Can you write English? Can you write BASIC? Can you do both at the same time? (Can you wiggle your ears, too??)

IF YOU = CAN THEN STOP

...to finish reading this before turning the page.

Announcing: The **SoftSide** "BASICally Speaking" Programming Contest! The object is to write one or more lines of proper BASIC code which also make sense as English. The above one-liner is one example; here are a few others:

"Valentine"

```
1 IF NOT YOU THEN END
```

"Las Vegas"

```
1 LET BET = VAL(UE$(1))
```

```
2 IF LUCKY AND WEWIN(10) THEN ON WE GOTO  
100, 1000, 10000
```

"Happily Ever After"

```
1 FOR TIME = NOW TO ETERNITY
```

```
2 LOVE = YOU AND I
```

```
3 IF NOT YOU AND I THEN NEXT TIME
```

```
4 REMIND ME TO
```

```
5 GET SUSIE
```

"Love Story"

```
1 TUESDAY = THEDAY
```

```
2 CALL TUESDAY
```

```
3 LET ITRING = 5 * MINIMUM
```

```
4 IF AMAN THEN STOP
```

```
5 IF NOT AMAN THEN FOR ALL = OUR TO
```

```
DAYS STEP BYME
```

```
6 LET US = MARRIED
```

```
7 NEXT ALTAR
```

```
8 GET MOVING
```

```
9 GOTO 1 : REMOTE PLACE
```

```
10 ALL = LOVE AND JOY : END
```

Here are the Official Rules:

1. All code must conform to proper BASIC syntax on at least one of these three computer systems: Apple II, Atari, or TRS-80™ (any version, model, or level). Submissions will be judged EXACTLY as submitted, with no allowances for typographical errors.

2. The "programs" need not do anything useful. However, programs which CAN be run without hanging, crashing, or generating error messages will have a competitive edge. And programs which actually DO something unique when run will have an additional advantage.

3. The BASIC line numbers need not fit into the English message of your program. They will be ignored by the judges, unless you do something unique and creative with them which contributes to the message.

4. The ENGLISH syntax of your code need NOT be perfect. Entries will be judged on creativity and originality as well as grammatical sense. But you can expect to lose points if the English meaning is TOO forced or obscure.

5. Entries may be any length. The English message may be humorous or serious, ridiculous or sublime. You may submit as many entries as you like, but every entry should be on a separate sheet of paper and should be labelled with your name, address, phone number, and the computer for which it is written.

6. All entries must be postmarked no later than June 30, 1981. Entries are not returnable, and become the property of **SoftSide**.

7. Judging will be by members of the editorial staff of **SoftSide**. Winners will receive certificates good toward the purchase of software from TSE, and will have their entries published in **SoftSide**. Other entries deserving honorable mention will also be published.

IF YOU = GOOD THEN YOU = WINNER

Good luck!



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 \$15.95
S-80 32K Disk \$21.95

Dealer Inquiries Invited

TAPE \$15.95

DISK \$21.95


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

S-80 One Liners

```
1 FORC=26TO1STEP-1;FORC=1TO50;A=A+.25;PRINTTAB(28+SIN(A)*B);STRIN
GC$(10,191);FORZ=1TO10;NEXTZ;NEXTC;B
```

Dan Cobb
Bucyrus, OH

```
0 IFC=0CLS;C=1;GOTOELSEPRINT@192,CHR$(31);FORG=0TO9;PRINT@192+5*
64+RND(10)*S,"*";NEXT;FORT=1TO900;NEXT;IFZ=0PRINT@0,"YOUR PERSO
NALITY TRAITS WILL BE REVEALED BY WHAT YOU SEE IN THESE CONSTELL
ATIONS.I, MERLIN THE SHRINK, SAY IT IS SO.":Z=1;GOTOELSEGOTO
```

Bill Burnham
Redwood City, CA

```
1 CLS;FORX=129TO191;FORY=129TO191;D#=CHR$(X)+CHR$(Y);A=RND(895)+
64;PRINT@A,D#;NEXTY;NEXTX;GOTO1
```

Janelle Keberle
Chagrin Falls, OH

```
1 IFW>0GOTO1ELSECLS;D=400;B=20;L=B;S=1;E=.4;C=.05;F=.5;FORW=1TO5
2;FORX=LTO10STEP5;Y=SQR(ABS(D-X*X));IFD<X^2 Y=F;SET(Y*S+B,X+20);
B=B+C;D=D-E;NEXT;L=L;S=-S;NEXT;ELSESET(Y*S+B,X+20);B=B+C;D=D-E;
NEXT;L=L;S=-S;NEXT;GOTO1
```

Michael Lyon
Kansas City, MO

```
1 S#=CHR$(140);CLS;FORJ=1TO16;P=RND(56)+67+64*RND(13);PRINT@P,CH
R$(140);FORT=1TO80;NEXT;PRINT@P,CHR$(191);PRINT@P+1,CHR$(157);
PRINT@P-1,CHR$(174);FORT=1TO20;NEXT;PRINT@P-64,CHR$(143);PRIN
T@P+64,CHR$(188);PRINT@P+3,S#+S#;PRINT@P-4,S#+S#;NEXTJ;GOTO1
```

L.O. RexRode
Willoughby, OH

Programming Hint

In many programs the ' is often used as the abbreviation of REM. This is incorrect. The ' is actually an abbreviation of :REM (see pages 1-8 of the Level II Manual).

The ' takes three bytes of memory to store, in Hex. (3A 93 FB). REM only takes one byte of memory to store, in Hex. (93).

When ' is used at the beginning of a line, it is using two more bytes of memory than REM would. For example:

```
10 ' THIS IS A REMARK.
```

vs .

```
10 REM THIS IS A REMARK.
```

When ' is used at the end of a line, it is using one more byte of memory than REM would.

For example:

```
10 A = B + C' A IS THE SUM.
```

vs .

```
10 A = B + C:REM A IS THE SUM.
```

Note that the often-used combination of : ' is redundant, as that would stand for ::REM (a waste of yet another byte).

If you want optimum memory usage, and comments are a necessity, then avoid the short term benefits of ' and get long term benefits by typing out REM.

Stephen Milliken
Randolph, MA

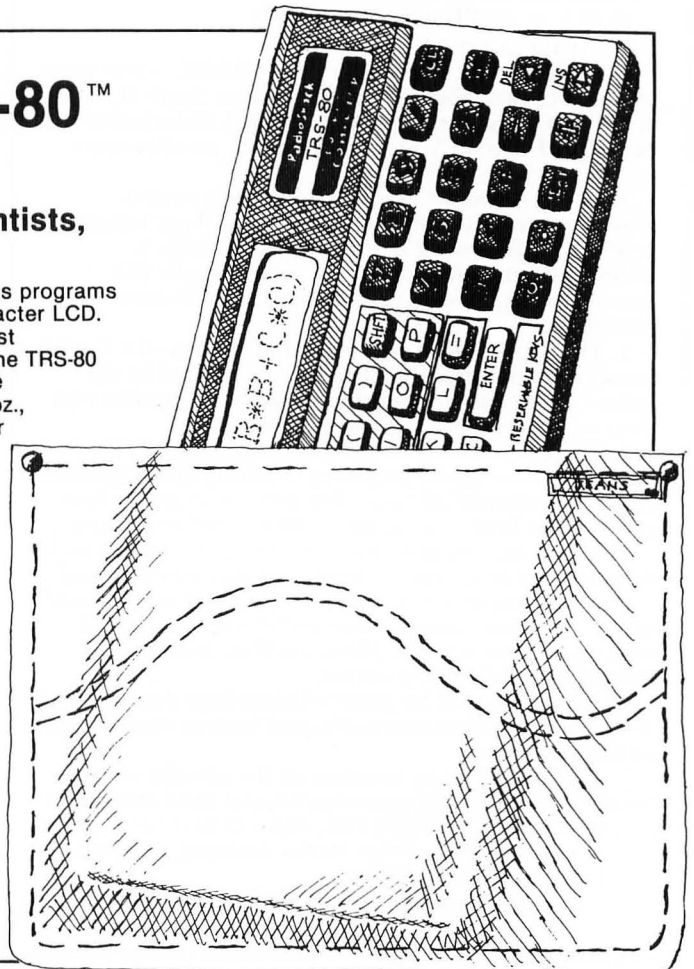
Radio Shack's TRS-80™ Pocket Computer

**For Businessmen, Engineers, Scientists,
Educators!**

Library of prerecorded programs available. Memory retains programs even with power off. Typewriter-format keyboard, 24-character LCD. 15 arithmetic functions, 10 digit accuracy. One of the most remarkable technological achievements of the decade! The TRS-80 Pocket Computer actually has most of the features of the famous Level I TRS-80 microcomputer in a pocketable 6 oz., 2 3/4 x 6 7/8 x 11/16" size. You can program it in BASIC or use the growing library of prerecorded programs for Business Statistics, Business Finance, Real Estate, Civil Engineering, Personal Finance, and Aviation. There's even an educational Math Drill and Games Pack for fun. Programs and data are retained even with power off. Case, long-life (300 hours) batteries included. Cassette interface included, connects Pocket Computer to cassette recorder for loading prerecorded software, storing and saving programs. Batteries extra. With cable. 26-3501 +

\$259.⁰⁰


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



COMPUTER SCIENCE PRESS

THE NATURE OF COMPUTATION: AN INTRODUCTION TO COMPUTER SCIENCE

Ira Pohl and Alan Shaw

An exciting new book covering the events in chess playing computer programs since 1975. It includes a description of the microcomputer revolution, its impact on computer chess, and a complete description of the challenge match between David Levy and Chess 4.5, the world champion program. \$16.95

PASCAL: AN INTRODUCTION TO METHODICAL PROGRAMMING

William Findlay and David Watt

This book, intended for use in a first course in programming, is based on the Pascal language. It assumes no prior knowledge of computing and only elementary mathematics. Emphasizing programming principles, good style, and a methodical approach to program development, it is an excellent introduction to Pascal. \$12.95

STRUCTURED BASIC AND BEYOND

Wayne Amsbury

An outstanding teaching and self-educational book on the BASIC language, emphasizing structured programming principles. Details of the language are presented in an easily understood manner oriented toward interactive terminal use. \$10.95

CHES AND COMPUTERS

David Levy

This is the first really comprehensive book on computer chess. It contains a history of computer chess, describes early chess playing machines, how computers play chess, and an account of early Soviet attempts at chess programming, plus a record of computer chess tournaments. *Paper*, \$9.95

MORE CHES AND COMPUTERS: THE MICROCOMPUTER REVOLUTION THE CHALLENGE MATCH

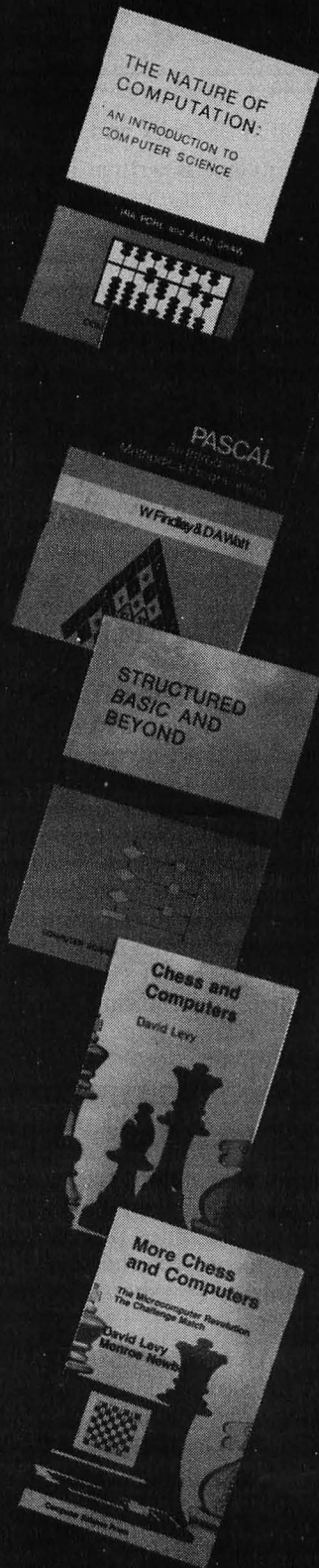
David Levy and Monroe Newborn

An excellent new text which deals with the core concepts of computer science. It addresses three major objectives: a survey of the field; an initial literacy in the language and methods of computer science; and a historical, philosophical, and social perspective. \$12.95

All prices are subject to change without notice. Handling and postage charge of \$1.25 per book.



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



CONVOY

by William Morris and John Cope

"Convoy" is written in Atari BASIC and requires 24K of RAM and joysticks.

Atari "Convoy" is a translation of the "Convoy" that appeared in the January issue of *SoftSide*, and there you will find a more detailed description of the game than the one below. The game plays much the same as the other versions.

"Convoy" can be divided into seven major sections:

INITIALIZATION

This is the entry of personal data concerning the player. If this is not the player's first game, additional information is read from cassette.

CONVOY CONSTRUCTION

Allows the player to select the content of the convoy, with a maximum of 15 vessels. Certain cargos, such as fuel and ammunition, carry a high risk content and are more easily destroyed. These cargos may be ignored during initial games, but these requirements must be satisfied to permit successful completion of the simulation. To specify a cargo, type the first two letters of the cargo.

BEGINNING THE CONVOY

A player enters the position of each ship within the convoy. The computer evaluates the setup, and assigns experience points accordingly. Ships placed first are easiest to protect and so the most valuable cargos should be entered early.

MAP AND MOVEMENT PHASE

A map of the North Sea area is drawn, and the convoy's position is shown. The player is asked to choose a destination port. Murmansk is closer and is the prime goal, but Archangel will bring more chances to earn experience through combat.

Using the joystick, the player moves the convoy. Normally three moves are permitted each day. After this movement, an on-screen submarine search is made, and if no contact is made, a Luftwaffe air search takes place. Contact with either of these forces puts the player into a combat situation. If no contact occurs, the player has

the opportunity to conduct air/sea drills or reorganize the convoy.

SUBMARINE CONTACT

Initially, surface contact is made with a graphic display modelled after the radar screens of World War II. The player, represented by a solid dot, must move to intercept the submarine within a certain time period or the submarine will be able to fire a torpedo at the convoy.

Upon interception, the submarine dives, which requires the player to interpret an animated and sometimes inaccurate sonar screen to determine the depth of the U-boat. Once the player has determined the depth charge setting using the joystick, and pressed the fire button on the joystick to release the charge, the underwater explosion is shown. Failure to sink the submarine allows it to again fire a torpedo at the convoy.

AIR CONTACT PHASE

A plane flies across the screen, and the player must use the joystick to center a sight on the plane in order to down it. To fire, press the fire button on the joystick. Should the plane penetrate the defenses, bombs will be dropped on the convoy. Ships with high-risk cargos such as fuel and ammunition can absorb only one hit, while others may absorb more. If any ship is struck, the player must decide if the convoy will be slowed down for repairs or, if the damaged ship will be left alone where its survival is doubtful.

TERMINATION

Should you reach your destination, a final summary evaluation is made. Point status will determine the commander's promotion or demotion. At the same time, an updated requirement list for the Eastern Front is shown. Should the player choose, all of the important information will be saved onto cassette for use at a later time as part of a more involved scenario.

DOCUMENTATION

Lines 0-40: Initialization and Title Display — Dimensions all strings and opens the keyboard.

Lines 100-190: Data statements for graphic map display or graphic string — For the Atari version the variable J\$ is used for graphic strings. Note the inclusion of cursor control characters within this version.

Lines 200-400: Player Input — If this is the initial game, the string A1\$ assigns a first-time player the rank of Lieutenant while setting the variables K, S, TS, YE and MM to their starting values. If this is not the first voyage, routines are called to input data from tape in lines 210-230. Line 330 checks to see if data entered from either the keyboard or data file is correct.

Lines 340-510: Constructing the convoy — The variables R(1) through R(8) contain the "requirements" for the Eastern front. These are set to a maximum value in line 340 if this is the first game. The variable L(M) keeps track of the number of ships in the convoy with each vessel being part of the string BB\$. Line 510 initiates the first point evaluation. Note the difference in how the lack of string dimensioning capability in the Atari BASIC affects the program format with the use of BB\$. The last line has L1 assume the size of the convoy, L2 assumes the size of the convoy while K is calculated on the basis of the construction process.

Lines 700-900: Designing the Convoy — Using the portion of the data statements alluded to somewhat earlier, a loop of variable length (z) outlines the physical shape of the convoy. Line 720 sets the limit on the data statements to be read for each vessel's position on the screen. Line 750 checks for an illegal entry. Line 780 automatically



places each escort vessel into its location. Lines 800-810 allow for alteration of data entry. This section concludes within algorithm to calculate once again, your K points in line 800. Lines 830-840 place any damaged vessel at the last place in the convoy.

Lines 900-1250: Map Phase — Line 900 sets the Atari's attract memory location to 0 to prevent alteration in the screen colors. The lines between 910 and 920 are required to deal with alterations in the variables SP that can occur later in the program. The Atari uses the same method to "draw" the European theater in line 950 employed by the Apple. The drawing continues until a value of less than 0 is read (line 940). The variable S determines in turn the season (E\$) and Z the length of the ice flows across the top of the screen. The variable S in line 1000 can be a floating point value. Setting it to an integer value was necessary in view of the Atari's tendency to do unusual things such as rounding up to the higher value in the "ice flows" across the top of the screen. Only the background color returns a value of 0 in line 1050 thus avoiding a plot over any land areas. P1 and P2 determine the position of the convoy on the screen for each cycle.

Lines 1100-1400: Movement Phase — Line 1110 uses P2 as a flag to ship to line 1300 if this is not Day 1. Line 1170 — if the keyboard buffer is empty the loop continues. Line 1190 sets the screen position of your destination (XA,YA) based on the value returned from the keyboard. Line 1330 branches to 2060 for joystick entry for movement. If the screen position of the convoy and destination port are identical, the program jumps to 3000. Line 1360 guards against "wrap around" screen movement. Line 1370 continues the loop if the counter P is less than the number of permitted moves. SP controls the number of moves per day. With the Atari it is set at 3 because of the use of the joystick for diagonal moves.

Lines 1550-1600: Submarine Search — Z is used to randomly generate screen locations to be plotted. Before plotting, however, the Atari version checks that location for a 0 value in order to permit the plot (line 1510). Notice

how T is used as a flag to increase the possibility of contact in line 1550.

Lines 1600-1800: Air Search — S controls the number of air bases plotted with the screen position of your convoy also determining whether or not a particular "base" is plotted. R1 and R2 control the movement of the subsequent plots. As with the submarine search a plot at the convoy's position terminates the search with PR being set to zero indicating a combat situation. Line 1630 alters the search pattern locations if the horizontal screen position exceeds 26. Line 1644 guards against a search over the ice flow (line 1650).

Lines 1800-2000: No Contact Phase — The flag PR is set to connote a drill situation when the program branches to the appropriate location. The value returned from the keyboard is reduced by 48 to permit the "on GOTO" procedure in 1830.

Lines 2000-2399: Aircraft Run — X1, the left starting point of the aircraft JX\$, is determined by your K points. The joystick values in lines 2050-2090 control the screen position of the gunsight JT\$ in line 2040. The subroutine is accessed by two other segments of the program (Map Movement and Radar Screen). Line 2050 checks to see if you have pressed the joystick button to initiate firing at line 2130. Lines 2150-2160 print explosion if hit is made. Depending on the status of PR the remaining lines in this section adjust both the size of the convoy and its speed as well as your K points. The logic for this procedure is essentially the same as the other versions.

Lines 2400-2630: Radar Screen — Lines 2400-2430 set the position of the enemy sub on the radar screen. Lines 2465-2470 print the sub (club) and your pursuit vessel (diamond) only after the bar (J6\$) reaches their horizontal position. The pursuit method accesses the common joystick routine (2500). Lines 2580-2600 control movement of the sub on the screen. The pursuit procedure is similar to the methodology used in lines 1300-1400. Different algorithms at the end of this section attends to the size of variables — L2, SP and the array B\$.

Lines 2640-2830: Sonar Screen — V and W are set at this point in line 2640. They control the screen location of the graphic "sub" in the next section as well. FOR/NEXT loops are used to create the animated bar. Line 2690 determines whether the gauge will be printed as an aid in determining the sub's depth. Line 2795 and 2890 check for joystick reading to alter depth setting (53) (lines 2920, 2950).

Lines 2970-2999: Depth Charge Launch — V and W determine the vertical and horizontal position of each graphic figure. The animated plot of the depth charge figure is preset except for the final vertical position. As with the other versions, Atari "Convoy" uses the ABS function to compare W and V to the final plot position, to determine if a "hit" has occurred.

Lines 3000-3470: L1 through L6 are used to summarize player progress. Report Section — This section is accessed at the end of each "day" and the voyage. Line 3027 is employed at the end of the simulation. Line 3400 increments the season variable S\$. This allows for three voyages for each season. This line also changes if the year after the third voyage. Line 3405 — End of simulation check. Lines 3410-3430 — Algorithms to attend to updating Soviet requirements based on your performance. Lines 3440-3470 — Save to Tape Section.

Lines 4000-5000: Evaluation Algorithms — This section is the same as the others.

Lines 6000-7000: End of game evaluation for promotion or demotion.

Lines 7000-7030: Final Summation.

Lines 8000-9000: Daily Log Entries — Once again the same algorithms are employed in earlier versions.

Lines 30000-30030: Sound Routines.

Lines 30200-30220: Delay Routines.

Lines 170-190, and line 2980 in Atari "Convoy" consist of graphic strings which cannot be printed on our printer, and are therefore

continued on next page

continued from previous page
 represented by symbols in the listing. The symbols should be replaced by:

Symbol	Meaning
b	Space
@	A reverse video space (full graphics block)
d	ESCAPE followed by SHIFT equal (looks like a down-arrow).
l	ESCAPE followed by SHIFT plus (looks like a left-arrow).
u	ESCAPE followed by SHIFT minus (looks like an up-arrow).
*	An asterisk character.(*)

Capital letters in these lines should be typed with the Control key to get special graphic characters.

Underlined characters in this listing should be typed in reverse video using the Atari logo key. The curled right bracket character that appears in some of the PRINT statements is a clear screen character, generated by typing ESCAPE followed by a SHIFT CLEAR.

ON SPLITTING THE CONVOY

"Convoy" has been structured in such a manner as to permit splitting it into two sections for machines with 16K of memory. The optimum point at which to

perform the 'surgery' is just prior to line 900. Upon doing so, however, certain modifications will have to be made to both segments. (hereafter called Parts I and II). We hope the following suggestions will aid in the process.

1. The data statements in lines 105 to 135 are accessed in Part II. To allow this without any additional memory overhead try the following:

LINE PROGRAM STATEMENTS

```
890 For X=1 to 37: Read Z:
    Next X REM-Read through
    CONVOY design data

891 For X=1536 to 1689: Read
    Z: Poke X,Z: Next X
```

This procedure permits the program to access page six of memory (which we are using as a storage area for the map data). To use this from Part II try the following:

LINE STATEMENT

```
940 For X=1536 to 1689: Step 3:
    A=Peek X: B=Peek (X+1):
    D=Peek (X+2): Next X
```

You will of course, have to modify the Read statement accordingly, eliminating 930 and

940 as outlined in the above example. This step allows the program to utilize almost 1K of data without the equivalent memory charge.

2. Move the load from the tape section for subsequent "voyages" from Part II into Part I. This permits access to data files generated in Part II while opening up some additional memory in the latter area. Listen — everytime another byte bites the dust the happier you should be.....

3. Delete option 3 — "reorganizing the CONVOY" from Part II as this accesses a key routine in Part I. (Line 1800-1830).

4. If necessary, you might modify the report section in lines 3000 to 3380, 6040 to 6075 and 7000 to 7030 to merely state daily progress in point form. (eg. subs sunk---, points ---- etc.)

5. Some sound and delay routines in lines 30000-30220 will have to be duplicated in Part I. This is not a problem as you have more than sufficient space here.

6. Should all of these operations still leave you short of memory (which if highly doubtful), you might consider placing all of the evaluations procedures used at the end of each voyage (lines 4000-7000) as a third segment to be chained at the appropriate time.

7. Be sure that you use the Atari chaining routine as outlined on pages 30-31 of the manual. Using this procedure, variables defined in Part I can be utilized in Part II.

<pre>0 REM *** CONVOY *** 1 REM *** (C) J. Cope & Wm Morris *** 10 GRAPHICS 2:POKE 752,1:DIM A\$(16),A1 \$(21),A2\$(21),A3\$(21),B\$(2),BB\$(16),B(17),E\$(6),L(9),R(8),RE(8) 20 DIM JB\$(10),JC\$(5),JD\$(38),JE\$(20), JF\$(20),JG\$(18),JH\$(36),JK\$(36),JL\$(18),JT\$(17),JX\$(3) 30 FOR Z=1 TO 9:L(Z)=0:NEXT Z:BB\$="FUA HTAFLORFOGRTI":OPEN #2,4,0,"K:" 40 POSITION 7,5:? #6;"CONVOY":FOR Z=0 TO 1:SETCOLOR 4,3,0:GOSUB 30000:SETCOL OR 4,8,0:GOSUB 30000:NEXT Z 100 DATA 26,13,30,13,28,11,28,15,22,13 ,34,13,24,11,32,11,24,15,32,15,28,9,28 ,17,24,17,32,17,24,9,32,9,26,7,30,7,28 105 DATA 0,0,6,1,0,6,2,0,4,3,0,4,4,0,3 ,5,0,3,6,0,3,7,0,1,8,0,0,3,10,12,4,10, 12 110 DATA 5,11,11,6,11,12,7,17,19,8,16, 19,9,18,19,10,19,19,16,0,1,17,0,0,19,1 ,1</pre>	<pre>120 DATA 36,1,3,37,1,1,37,3,3,19,12,12 ,19,15,17,19,19,19,20,12,17,21,11,16 125 DATA 22,10,19,23,9,19,24,7,19,25,5 ,17,26,5,16,27,5,11,28,6,11,29,6,14 130 DATA 30,6,13,31,6,8,31,11,13,32,6, 8,32,11,19,33,5,8,33,11,19,34,11,19,35 ,8 135 DATA 19,36,7,19,37,5,19,38,5,19,39 ,5,19,30,16,19,28,18,18,-1,0,0 160 DATA LIEUTENANT ,LIEUTENANT COMMAN DER ,COMMANDER ,CAPTAIN ,COMMODORE ,Z 170 JE\$="@@@@@@@@@@@@@@@@@@@@":JF\$="eb bbbbbbbbbbbbbbbbb":JG\$="RRRRRRRRRRRRRR RRRRR" 180 JC\$="UUUUU":JX\$="UUU":JE\$="JeeHu1 11eF":JD\$="MMMMMMMMMMMMMMMMMMMMMMMMMM MMMMMMMMMM" 185 JH\$="@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@@":JK\$="ebbbbbbbbbbbbbbbbbbbbbbb bbbbbbbbbbbbb" 190 JT\$="QMEd111ASDd111ZX":JL\$="bbbb bbbbbbbbbbbbb"</pre>	<pre>200 A\$="":A1\$="LIEUTENANT "?: CHR\$(29) ;" Is this your first voyage? (Y/N)": GOSUB 30010:GET #2,U:IF U=89 THEN 300 210 ? "):Insert data tape and press 'RE TURN":GET #2,U:OPEN #1,4,0,"C:" 220 INPUT #1,A\$:INPUT #1,A1\$:INPUT #1, K:FOR Z=1 TO 8:INPUT #1,Y:R(Z)=Y:RE(Z) =R(Z):NEXT Z 230 INPUT #1,S:INPUT #1,YE:INPUT #1,N: INPUT #1,TS:CLOSE #1:POKE 764,255 300 GRAPHICS 0:POKE 752,1:SETCOLOR 2,9 ,0:?: CHR\$(29);" BRITISH ADMIRALTY - AR CTIC COMMAND":GOSUB 30010:IF K>0 THEN 320 310 ? CHR\$(29);CHR\$(29);"What is your name":INPUT A\$:IF A\$="" THEN A\$="HORN BLOWER" 320 ? :? :? A1\$;A\$;":":? " Your expe rience points are ";K:?"and you are t aking out CONVOY #";TS+1;". 330 ? CHR\$(29);"IS THIS CORRECT (Y/N)? ":GET #2,U:IF U=78 THEN K=0:S=0:TS=0:Y E=1941:N=0:GOTO 300</pre>
--	--	--

340 IF K=0 THEN FOR Z=1 TO 8:R(Z)=INT(RND(0)*199)+800:NEXT Z
400 SETCOLOR 2,13,0:FOR Z=1 TO 8:RE(Z)=R(Z):NEXT Z
405 ? CHR\$(125);CHR\$(29);" CO
STRUCTION PHASE":? ? "RUSSIAN REQUIR
EMENTS":GOSUB 410:GOTO 430

410 ? ? "Fuel ";RE(1);" ";? "Ammo
";RE(2);" ";? "Tanks ";RE(3);"
";? "Planes ";RE(4);" "
420 ? "Ore ";RE(5);" ";? "Food
";RE(6);" ";? "Grain ";RE(7);" ";
? "Timber ";RE(8);" ";RETURN
430 ? CHR\$(29);CHR\$(29);"NUMBER OF SHI
PS IN YOUR CONVOY IS ";L:?"(Enter @ t
o end,)":IF L=15 THEN 490

440 ? CHR\$(29);"SHIP # ";L+1;" CARGO "
;:INPUT B\$:IF B\$="@" AND L>0 THEN 490
442 IF LEN(B\$)<2 THEN GOTO 446
444 M=0:GOSUB 450:IF RE(M)>0 AND M<0
THEN GOSUB 30020:GOTO 470
446 GOSUB 30030:GOTO 405
450 IF M>14 THEN M=0:RETURN
455 M=M+2:IF B\$=BB\$(M-1,M) THEN M=M/2:
RETURN
460 GOTO 450
470 L(M)=L(M)+1:RE(M)=RE(M)-40:IF RE(M)
<0 THEN RE(M)=0

480 L=L+1:GOTO 405
490 POSITION 2,15:?" Your CONVOY is
now ready to sail,":L(9)=1+INT(K/1000
0):IF L(9)>4 THEN L(9)=4
500 ? ? " You are being escorted by
";L(9):?" destroyer(s).":? ,,"GOOD LUC
K":GOSUB 30010:GOSUB 30210
510 L1=L:L2=L(9):K=K+L1*10+L2*20:SP=3
700 GRAPHICS 0:SETCOLOR 2,7,0:PRINT CH
R\$(29);" DESIGNING THE CONVOY":
? CHR\$(29);CHR\$(29);"CARGO VESSELS"

701 FOR Z=1 TO 8:RE(Z)=L(Z):NEXT Z
710 RESTORE :GOSUB 410:?" Escorts ";L(
9)
720 POKE 752,1:FOR Z=1 TO L+L(9):READ
X,Y:POSITION X,Y:?" CHR\$(8);CHR\$(10);:G
OSUB 30020:NEXT Z
721 RESTORE :FOR Z=1 TO L:READ X,Y
730 POSITION X,Y:?" CHR\$(8);CHR\$(10);:G
OSUB 30020:POSITION X,Y:?" CHR\$(136);CH
R\$(138);:GOSUB 30020
740 POSITION 2,20:?" WHICH VESSEL"::IN

POT B\$:POSITION 2,20:?"
":M=0:GOSUB 450
750 IF M=0 OR RE(M)=0 THEN GOSUB 30030
:GOTO 730
760 RE(M)=RE(M)-1:POSITION X,Y:?" B\$;:P
OSITION 2,5:GOSUB 410:?" Escorts ";L(9
):B(Z)=M:NEXT Z
780 FOR Z=1 TO L(9):READ X,Y:POSITION
X,Y:?" ES":GOSUB 30020:NEXT Z:POSITION
10,14:?" 0"

790 POSITION 1,20:?" Is this the prope
r configuration (Y/N)?":GET #2,U:IF U
=78 THEN 700
800 GOSUB 30010:FOR Z=1 TO L:K=K+R(B(Z
) /10000*(9-B(Z)))*(15-Z):NEXT Z
810 IF T=0 THEN POSITION 2,22:?" Your
experience points are now ";INT(K):GOS
UB 30200
820 IF RR=0 THEN 890
830 FOR Z=1 TO L:IF B(Z)=RR THEN B(Z)=
B(L):B(L)=RR:Z=15

840 NEXT Z
890 IF S=0 THEN S=1:YE=1941
900 POKE 77,0:T=T+1:GOSUB 8000:RESTORE
910 IF SP=2 AND RND(0)<0.5 THEN SP=3:?"
":Damaged vessel has been repaired,":
?" Resuming normal speed":GOSUB 30210
920 GRAPHICS 3:SETCOLOR 0,12,0:SETCOLO
R 1,0,8:SETCOLOR 2,3,0:SETCOLOR 4,7,2:
COLOR 1:PR=0:IF SP=1 THEN SP=2
930 RESTORE :FOR X=1 TO 37:READ Z:NEXT
X

940 READ A,B,D:IF A<0 THEN 1000
950 PLOT A,B:DRAWTO A,D:GOTO 940
1000 ON INT(S) GOTO 1010,1020,1030,104
0
1010 E\$="FALL":Z=2:GOTO 1050
1020 E\$="WINTER":Z=3:GOTO 1050
1030 E\$="SPRING":Z=2:GOTO 1050
1040 E\$="SUMMER":Z=1
1050 COLOR 2:FOR Z1=0 TO Z:FOR Z2=0 TO
39:LOCATE Z2,Z1,X:IF X=0 THEN PLOT Z2
,Z1

1060 NEXT Z2:NEXT Z1
1100 POKE 752,1:COLOR 3:IF P2=0 THEN G
=2:H=18:PLOT G,H
1110 ? "3":POKE 657,10:?" E\$;" OF ";YE;
" DAY ";T:IF P2=1 THEN 1300
1130 ? CHR\$(29);" Your convoy is west
of Britain":P2=1

1140 FOR Z=1 TO 5:GOSUB 30000:COLOR 0:
PLOT G,H:GOSUB 30220:COLOR 3:PLOT G,H:
NEXT Z
1150 POKE 656,2:?" Is your destination
ARCHANGEL (A)? ":COLOR 2:PLOT 35,8:G
OSUB 30020:GOSUB 30210:COLOR 1

1160 PLOT 35,8:POKE 656,2:POKE 657,21:
?" OR MURMANSK (X)?":COLOR 2:PLOT 31,
6:GOSUB 30020
1170 GOSUB 30210:COLOR 1:PLOT 31,6:IF
PEEK(764)=255 THEN 1150
1190 U=PEEK(764):XA=31:YA=6:IF U<37 T
HEN XA=35:YA=8
1300 FOR Z=1 TO 3:COLOR 1:PLOT XA,YA:G
OSUB 30020:COLOR 2:PLOT XA,YA:GOSUB 30

220:NEXT Z:P=0
1310 POKE 656,2:?" Use JOYSTICK to re
position Convoy ":COLOR 3:PLOT G,H
1330 X1=G:Z1=H:GOSUB 2060:IF X1=XA AND
Z1=YA THEN 3000
1350 LOCATE X1,Z1,Z:IF Z<0 THEN 1370
1360 COLOR 0:PLOT G,H:IF X1<39 AND X1>
0 AND Z1<20 THEN P=P+1:G=X1:H=Z1:GOSUB
30010
1370 COLOR 3:PLOT G,H:IF P<37 THEN 133
0

1500 P=0:?" 3":CHR\$(29);" NAZI SUBMARIN
E SEARCH":GOSUB 30000:FOR X=1 TO 20:Z=
INT(5+(34*RND(1))):Y=INT(19*RND(1))
1510 LOCATE Z,Y,Z1:IF Z1<0 THEN 1530
1520 COLOR 2:PLOT Z,Y:GOSUB 30020:GOSU
B 30220:COLOR 0:PLOT Z,Y:GOTO 1530
1525 IF Z=G AND Y=H THEN GOSUB 30000:C
OLOR 2:PLOT Z,Y:GOSUB 30000:COLOR 0:PL
OT Z,Y

1526 IF X<21 THEN X=20:NEXT X
1528 PA=1:PR=0:GOTO 2400
1530 NEXT X
1540 IF PA=1 THEN PA=0:IF INT(RND(1)*2
)=0 THEN Z=G:Y=H:GOTO 1525
1550 IF INT(RND(0)*2)=0 AND (T=4 OR T=
5) THEN Z=G:Y=H:GOTO 1525
1600 ? CHR\$(125);CHR\$(29);" LUFTWAFFE A
IR SEARCH":ON INT(S) GOTO 1604,1606,16
04,1602

1602 Y=23:Z=13:GOSUB 1620:Y=21:Z=14:GO
SUB 1620
1604 Y=28:Z=7:GOSUB 1620
1606 Y=29:Z=6:GOSUB 1620:Y=27:Z=6:GOSU
B 1620:Y=23:Z=9:GOSUB 1620:Y=22:Z=15:G
OSUB 1620:Y=20:Z=17:GOSUB 1620
1610 GOTO 1800
1620 IF H>Z THEN RETURN
1625 COLOR 2:PLOT Y,Z:FOR W=1 TO 16:R1
=INT(RND(0)*2):R2=INT(RND(0)*2)

1630 Y=Y-R1:Z=Z-R2:IF G>26 THEN Y=Y+R1
*2
1640 IF Y>38 THEN Y=38
1642 IF Z<0 THEN Z=0
1644 LOCATE Y,Z,Z2:Z3=0:IF Z2<0 THEN
Z3=1
1646 IF Z2=2 THEN 1680

continued on next page

continued from previous page

1650 GOSUB 30030:COLOR 2:PLOT Y,Z:IF Y
=G AND Z=H THEN PR=0:FA=1:W=16:NEXT W:
GOTO 2000

1660 GOSUB 30220:COLOR Z3:PLOT Y,Z
1680 NEXT W:RETURN
1800 ? "3. Air Defence Drill 3. Regro
up Convoy":? "2. Submarine Drill 4.
Continue"

1810 PR=1:PRINT "ORDERS?";
1820 GET #2,U:U=U-48:IF U<1 OR U>4 THE
N 1820

1830 ON U GOTO 2000,2400,700,900
2000 GRAPHICS 0:SETCOLOR 2,3,0:POKE 75
2,1:POSITION 11,11:IF PR THEN ? "AIR A
TTACK DRILL":GOTO 2020
2010 ? "COMBAT STATIONS!"
2020 GOSUB 30000:Z=INT(23*RND(0)+1):X=
INT(K/1500):GRAPHICS 0:SETCOLOR 1,0,15
:SETCOLOR 2,7,2:IF X>20 THEN X=20
2025 X1=17:Z1=11:Y=X+3
2030 ? "3":POKE 752,1:IF X>36 THEN 220
0

2040 POSITION X1,Z1:JF:POSITION X,Z
:JX:GOSUB 2050:GOTO 2092
2050 IF STRIG(0)=0 THEN 2130
2060 IF STICK(0)>4 AND (STICK(0)<8) TH
EN X1=X1+1
2070 IF STICK(0)>8 AND (STICK(0)<12) T
HEN X1=X1-1
2080 IF STICK(0)=6 OR STICK(0)=14 OR S
TICK(0)=10 THEN Z1=Z1-1
2090 IF STICK(0)=5 OR STICK(0)=9 OR ST
ICK(0)=13 THEN Z1=Z1+1

2091 RETURN
2092 IF Z1<1 THEN Z1=1
2094 IF Z1>18 THEN Z1=18
2096 IF X1<1 THEN X1=1
2098 IF X1>34 THEN X1=34
2100 IF XJ=3 THEN XJ=0:GOTO 2030
2105 XJ=XJ+1:X=X+1:R=INT(RND(0)*4-2):Z
=Z+R:IF Z<1 THEN Z=1
2110 IF Z>23 THEN Z=23
2120 GOTO 2030

2130 FOR ZK=1 TO 2:POSITION X1,Z1:CH
R\$(29):"xxx":GOSUB 30030:NEXT ZK
2135 IF X>X1 OR Z>Z1+1 THEN 2050
2150 FOR ZK=1 TO 3:POSITION X1,Z1:CH
R\$(29):" x ":GOSUB 30020
2160 POSITION X1,Z1:CHR\$(29):"x x":
GOSUB 30020:NEXT ZK:?:?:?:GOTO 2392
2200 SETCOLOR 2,3,0:IF PR THEN ? "Air
Attack Drill was unsuccessful!":K=K-7:
GOTO 900

2250 POKE 84,20:?"BOMBERS HAVE PENETR
ATED YOUR DEFENSES!":GOSUB 30210

2260 ? "Bombs are being dropped on you
r convoy":GOSUB 30210:R=L:IF SP<3 THE
N 2290

2280 R=INT(L*RND(0)+1):B(L+1)=B(R)
2285 FOR Z=R TO L+1:B(Z)=B(Z+1):NEXT Z
2290 RR=B(L):IF INT(4*RND(0))=2 OR SP<
3 OR B(L)<3 THEN 2380

2295 ? "SHIP # ";R;" HAS BEEN DAMAGED"
:K=K-15:L4=L4+1

2297 POKE 764,255:?"Will you reduce s
peed for repairs?":GET #2,U:IF U=89 TH
EN SP=1:GOTO 900

2300 L(B(R))=L(B(R))-1:L=L-1:IF INT(2*
RND(0))>1 THEN 2310

2305 ? "The damaged vessel returned to
England":L3=L3+1:GOTO 900

2310 IF INT(2*RND(0))=1 THEN K=K-30:L3
=L3+1:L4=L4-1:?"Vessel sinks en route
":GOTO 900

2320 K=K+20:?"The damaged vessel will
arrive"

2380 K=K-80:PR=0:L(RR)=L(RR)-1:L3=L3+1
:IF SP<3 THEN SP=3

2385 ? "SHIP # ";R;" HAS BEEN SUNK":L=
L-1:GOTO 900

2392 SETCOLOR 2,11,2:IF PR THEN ? "Air
Attack Drill was successful!":K=K+20:
GOTO 900

2395 ? "Aircraft was shot down!":K=K+1
00:L6=L6+1:GOSUB 30220

2397 LL=LL-1:IF LL<1 THEN R=INT(2*RND(
0)):IF R=1 THEN 900

2398 R=INT(2*RND(1)):IF R=1 THEN 2020

2400 XZ=0:LM=L(9):GRAPHICS 0:SETCOLOR
2,3,0:POKE 752,1:POSITION 8,10:Z1=9:X1
=19:IF PR=0 THEN 2410

2405 PRINT "SUBMARINE PURSUIT DRILL":P
OSITION 10,12:?"Radar Search Phase":G
OSUB 30210

2410 IF PR=0 THEN ? " COMBAT STATIONS
!":GOSUB 30000:GOSUB 30000

2420 XS=INT(RND(0)*20):IF XS<5 OR XS>1
3 THEN 2420

2430 YS=INT(RND(0)*20):IF YS<12 OR YS>
28 THEN 2430

2440 SETCOLOR 2,13,0:?"":POSITION 10
,2:JF:FOR ZK=3 TO 15:POSITION 10,ZK
:JF:NEXT ZK:POSITION 10,16:JF\$
2450 POSITION 14,20:?"RADAR PHASE":?
:"Use joystick to pursue enemy vess
el":

2460 FOR ZZ=4 TO 14:POSITION 11,ZZ:J
G\$:GOSUB 30020:POSITION 11,ZZ:JL\$
2465 IF ZZ=Z1 THEN POSITION X1,Z1:CH
R\$(96);

2470 IF ZZ=XS THEN POSITION YS,XS:CH
R\$(16);

2480 NEXT ZZ

2490 IF STICK(0)=15 THEN 2490

2492 XZ=XZ+1:IF XZ=8 AND PR=0 THEN ? "
":GOTO 2989

2494 IF XZ=8 THEN XZ=0:POSITION 7,20:?
"Drill was unsuccessful!":K=K-25:GOTO
900

2500 GOSUB 2060

2510 IF Z1<4 THEN Z1=4

2520 IF Z1>14 THEN Z1=14

2530 IF X1<11 THEN X1=11

2540 IF X1>29 THEN X1=29

2560 IF XS<Z1 OR YS<X1 THEN 2580

2570 FOR ZK=3 TO 15:POSITION 10,ZK:J
F\$:NEXT ZK:GOSUB 30030:POSITION YS,XS:
? CHR\$(96);

2575 GOSUB 30030:POSITION YS,XS:POSITI
ON YX,XS:CHR\$(16):GOTO 2640

2580 R=INT(RND(0)*2):IF R=1 THEN 2600

2585 R=INT(RND(0)*3-1):XS=XS+R:IF XS<4
THEN XS=4

2590 IF XS>13 THEN XS=13

2600 R=INT(RND(0)*3):YS=YS+R:IF YS<12
THEN YS=12

2610 IF YS>28 THEN YS=28

2620 IF XS=Z1 AND YS=X1 THEN 2570

2630 GOTO 2460

2640 V=INT(RND(0)*18):SS=6:W=INT(RND(0
)*)9):YY=V+10-W

2680 ? "3":POSITION 2,12:JH\$:FOR Z=1
TO 4:JH\$:NEXT Z:JH\$

2690 IF K>11000 AND INT(RND(0)*6)=4 TH
EN POSITION 17,7:?"GUAGE MALFUNCTION!
":POSITION 14,20:GOTO 2720


```
2700 FOR Z=3 TO 36:POSITION Z,16: CHR
$(24);:NEXT Z:POSITION 14,20
2720 ? "SONAR REPORT":? "Press BUTTON
to launch depth charge":? "JOYSTICK al
ters depth SETTING: 10";
2730 FOR YM=YY TO V+1 STEP -1:ZW=4:FOR
ZZ=ZW TO (ZW+YM):POSITION ZZ,14: CHR
$(160);
2750 NEXT ZZ:GOSUB 30020:GOSUB 30220
2790 IF STRIG(0)=0 THEN YM=V:ZJ=1:GOTO
2820
```

```
2795 IF STICK(0)<>15 THEN GOSUB 2920
2800 FOR ZS=ZW TO (ZW+YM):POSITION ZS,
14: ? " ";:NEXT ZS:W=W+1:IF W>28 THEN W
=28
2820 NEXT YM:IF ZJ THEN ZJ=0:GOTO 2970
2830 FOR Y=V+1 TO 30:Q=Y:FOR VV=0 TO Q
:ZZ=ZW+W:POSITION ZZ,14: CHR$(160);:
NEXT VV
2840 GOSUB 30020:GOSUB 30220
2880 FOR XX=ZW TO (ZW+Q):POSITION XX,1
4: ? " ";:NEXT XX:IF STRIG(0)=0 THEN Y=
36:ZJ=1:GOTO 2910
```

```
2890 IF STICK(0)<>15 THEN GOSUB 2920
2900 W=W+1:IF W>28 THEN W=28
2910 NEXT Y:ZJ=0:GOTO 2970
2920 IF STICK(0)=14 THEN SS=SS-1
2930 IF STICK(0)=13 THEN SS=SS+1
2940 IF SS>21 THEN SS=21
2950 IF SS<5 THEN SS=5
2960 POSITION 34,22: ? " ";:POSITION
34,22: ? SS-5;"0":IF STICK(0)<>15 THEN
GOSUB 30220:GOTO 2920
```

```
2965 RETURN
2970 GRAPHICS 0:POKE 752,1:SETCOLOR 2,
7,2:POSITION 0,5: ? JD$:POSITION 5,4: ?
JB$:V=V+5:POSITION W,V: ? JC$
2971 A=9:B=3:GOSUB 2975:A=10:B=2:GOSUB
2975:A=11:B=1:GOSUB 2975:FOR ZM=B TO
B+(SS-2):B=ZM:GOSUB 2975
2972 NEXT ZM:GOTO 2977
2975 POSITION A,B:PRINT CHR$(20);:GOSU
B 30220:POSITION A,B: ? " ";:POSITION A
,5: ? CHR$(13);:RETURN
```

```
2977 IF W>7 AND W<15 AND (ABS(V-SS)=0
OR (ABS(V-SS)=1 AND K<10001)) THEN 298
0
2979 POSITION 25,10: ? "A MISS!":NM=1
2980 FOR ZP=1 TO 5:POSITION 10,SS: ? "u
*bd111b*bd111b*":GOSUB 30020:POSITIO
N 10,SS: ? "ub*bd111b*bd111b*b"
2984 GOSUB 30020:NEXT ZP:IF PR=0 AND N
M=1 THEN NM=0:GOTO 2989
2986 IF PR AND NM THEN NM=0:K=K-25:PRI
NT "PRACTICE WAS UNSUCCESSFUL":GOTO 900
2987 POSITION 3,10: ? "A HIT!":K=K+40:IF
PR=0 THEN K=K+460:L5=L5+1
2988 GOTO 900
```

```
2989 POSITION 20,10: ? "TORPEDO LAUNCHE
D!":GOSUB 30210:K=K-30:RR=0:R=L
2990 L(B(R))=L(B(R))-1:L3=L3+1: ? "SHIP
# ";L;" HAS BEEN SUNK": ? :GOSUB 30220
2991 L=L-1:GOSUB 30020:IF L=0 THEN 300
0
2992 IF SP<3 THEN L4=L4-1:SP=3
2993 B(L+1)=0:IF XZ=8 THEN XZ=0:GOTO 2
440
2994 LM=LM-1:IF LM=0 THEN 2998
2995 ? "DO YOU WANT ANOTHER RUN (Y/N)?"
:GET #2,U:IF U=89 THEN 2640
```



```
2998 IF INT(RND(0)*3)=1 THEN ? "ANOTHE
R SUBMARINE IS APPROACHING":GOSUB 3021
0:GOTO 2400
2999 GOTO 900
3000 GRAPHICS 0:SETCOLOR 2,9,0: ? "ARCT
IC CONVOY COMMAND HEADQUARTERS"
3010 GOSUB 4000:Z=1+K/10000:IF Z>5 THE
N Z=5
```

```
3020 IF A1$<>A2$ THEN 6000
3027 A2$="":IF (G=XA AND H=YA) OR L=0
THEN ? ? "Voyage Termination File"
3030 ? ? "LOG SUMMARY FOR:" ? A1$;" "
;A$;:GOSUB 3340:GOTO 3380
3340 ? ? ? "Your initial convoy stre
ngth was ";L1: ? "and you were escorted
by ";L2
3345 ? "destroyer(s).": ? "Of these shi
ps you lost ";L3;" and had ";L4: ? "dam
aged."
```

```
3350 ? ? ? "During your voyage you sank
";L5: ? "submarine(s) and downed ";L6;
" plane(s)"
3360 IF K<0 THEN K=0
3370 ? ? ? "Your experience points are
now ";INT(K);":":RETURN
3380 POKE 764,255: ? CHR$(29);"PRESS RE
TURN TO CONTINUE":GET #2,U
3400 S=5+0.34:IF S>3 THEN YE=1942:IF S
>5 THEN S=5
3405 TS=TS+1:IF TS+1=18 THEN ? "3": ? :
? :GOSUB 410:GOTO 7000
```

```
3410 ? ? ? "3":FOR Z=1 TO 8:R(Z)=R(Z
)+INT(50*RND(0)+1)
3415 IF R(Z)>999 THEN R(Z)=999
3420 RE(Z)=R(Z):NEXT Z:GOSUB 410:X=0
3425 FOR Z=1 TO 8:IF RE(Z)>999-TS*40 T
HEN X=X+1
3430 NEXT Z:IF X>2 THEN N=1: ? ? " MU
RMANSK HAS FALLEN DUE TO SUPPLY": ? "SH
ORTAGE"
3440 ? ? ? "Do you want to save your ga
me to tape?":GET #2,U:IF U=78 THEN END
3450 ? ? ? "Insert data tape and press 'R
ETURN':GET #2,U:OPEN #1,8,0,"C":
```

```
3460 PRINT #1;A$:PRINT #1;A1$:PRINT #1
,K:FOR Z=1 TO 8:Y=R(Z):PRINT #1,Y:NEXT
Z
3470 PRINT #1,S:PRINT #1,YE:PRINT #1,N
:PRINT #1,TS:CLOSE #1:END
4000 IF T<10 THEN T=20
4010 K=K-(T-10)*250:IF N=1 THEN K=K+50
0
4020 FOR Z=1 TO 8:K=K+R(Z)/(999-TS*50)
*100
4025 R(Z)=R(Z)-40*L(Z):IF R(Z)<0 THEN
R(Z)=0
4030 K=K-R(Z)/100^2:NEXT Z:K=INT(K):IF
K<1 THEN K=1
4040 RETURN
6000 RESTORE :FOR X=1 TO 193:READ Q:ME
XT X:FOR X=1 TO Z:READ A2$:NEXT X:IF Z
=5 THEN 6030
```

```
6010 FOR X=Z+1 TO 5:READ A3$:IF A3$=A1
$ THEN X=5:GOTO 6060
6020 NEXT X
6030 IF A1$=A2$ THEN 3027
6040 ? ? ? ? "Congratulations!": ? ? ? "
you have been promoted to"
6050 POKE 85,5:PRINT A2$:A1$=A2$:GOSUB
30200:POKE 85,0:GOTO 3027
6060 IF K>Z*10000-2000 THEN 3027
6070 ? ? ? ? "BAD NEWS!": ? ? ? "Because
of your poor showing you"
6075 ? "have been demoted to the rank
of": ? A2$:A1$=A2$:GOSUB 30200:GOTO 302
7
7000 ? CHR$(29);"PRESS 'RETURN' TO CON
TINUE":GET #2,U
7010 ? "3CONGRATULATIONS! You have com
pleted": ? "the simulation!"
```

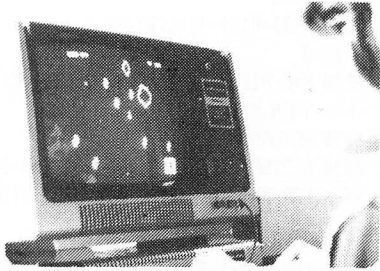


```
7015 ? ? ? "You began the game as a Lie
utenant": ? "and you have now attained
the rank of": ? A1$;"."
7030 ? "You have accumulated ";K;" vic
tory": ? "points.": ? ? :END
8000 IF T=1 THEN RETURN
8002 IF U<4 THEN GOSUB 30200
8003 IF L=0 THEN 3000
8004 U=0:GRAPHICS 0:SETCOLOR 2,15,0:IF
L=0 THEN 3000
8010 PRINT ">LOG ENTRY FOR CONVOY ";TS
+1;" ON DAY ";T-1;"."
8020 POKE 752,1: ? ? A1$;" ";A$: ? " IN
COMMAND.":GOSUB 3340: ? ? ? "PRESS '
BUTTON' TO CONTINUE"
8030 IF STRIG(0)=1 THEN 8030
30000 FOR U=200 TO 50 STEP -4:SOUND 0,
U,10,8:NEXT U:SOUND 0,0,0,0:RETURN
30010 FOR U=1 TO 4:SOUND 0,20,10,8:GOS
UB 30220:SOUND 0,0,0,0:NEXT U:RETURN
30020 U=INT(RND(0)*100)+50:SOUND 0,U,1
0,8:GOSUB 30220:SOUND 0,0,0,0:RETURN
30030 SOUND 0,1,4,8:GOSUB 30220:SOUND
0,0,0,0:RETURN
30200 FOR TT=1 TO 350:NEXT TT
30210 FOR TT=1 TO 70:NEXT TT
30220 FOR TT=1 TO 10:NEXT TT:RETURN
```

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

TRS-80™ HOME ARCADE

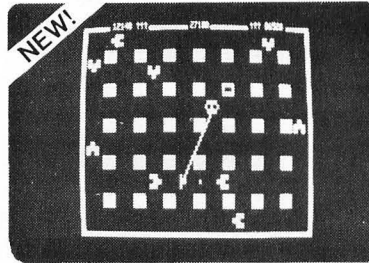
SUPER NOVA®



"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

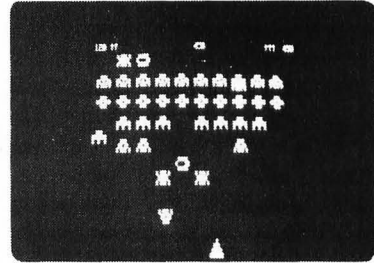
ATTACK FORCE®



"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

PERSONAL SOFTWARE VISICALC™



Students and professors, small business owners and tycoons, all rave about Visicalc. Dispense with pencil, paper and calculator — instead, you can have an "electronic sheet" to do your calculations, projections, and planning. You can calculate sales projections, income taxes, personal budget, cost estimates, engineering changes — even balance your checkbook. Visicalc is partitioned into a matrix of rows and columns. At each position in this matrix you can define a title, formula, or number. By writing on your "electronic sheet," you can set up individualized charts, tables, and records. For example, you can design your own sales forecasting format to assist in making the important "What if?" business decisions such as: "What would happen if sales increased by 10%?" "What if delivery time were decreased by two weeks?" "What would be the result if I produced 500 widgets this month instead of 600? What if I produced 700?" With Visicalc to assist you in performing these calculations, you can save countless hours.

Apple II 32K Disk..... \$149.95
Atari 32K Disk..... \$199.95
S-80 32K Disk..... \$95.00



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

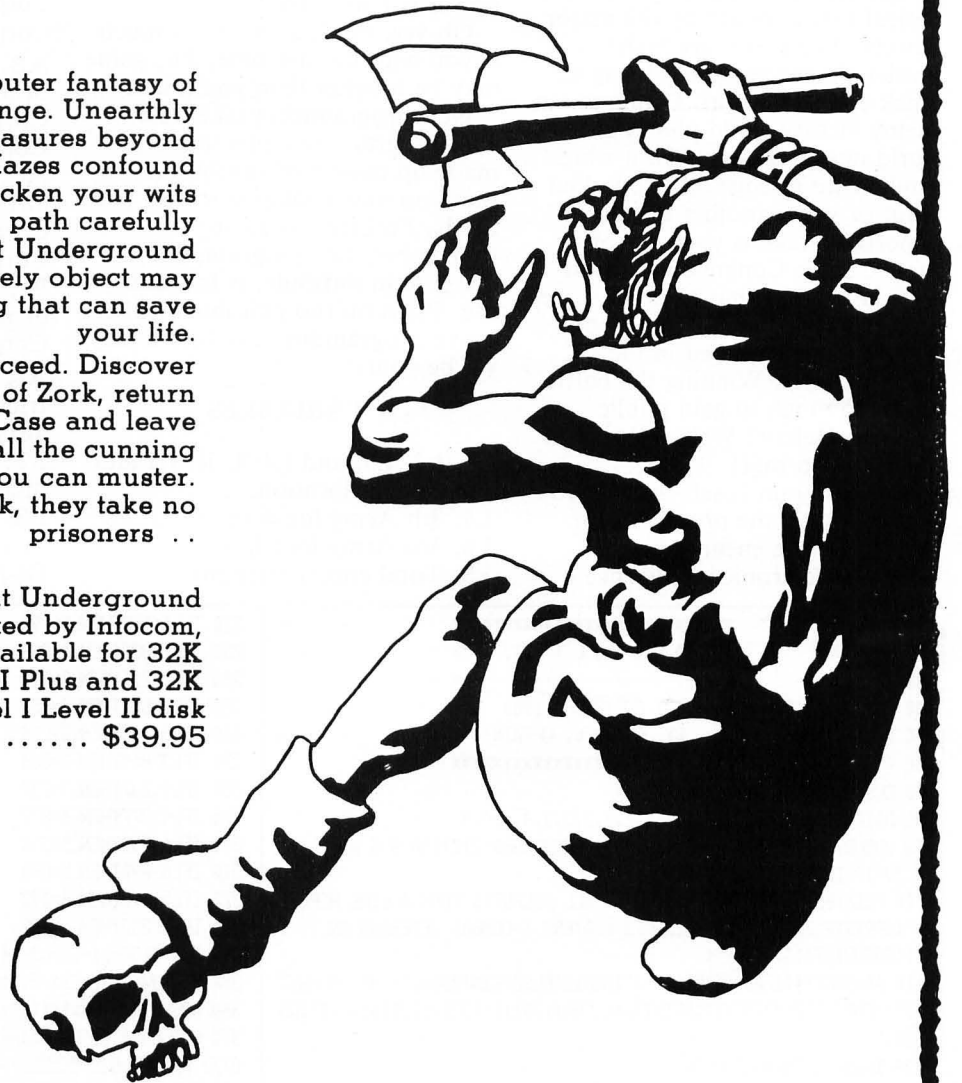


ZORK™

Zork™ is a computer fantasy of ultimate challenge. Unearthly creatures guard treasures beyond your imagination. Mazes confound your quest. So quicken your wits and pick your path carefully through the Great Underground Empire. The least likely object may be the only thing that can save your life.

Yet, you can succeed. Discover the 20 treasures of Zork, return them to the Trophy Case and leave alive. But bring all the cunning and courage you can muster. Because in Zork, they take no prisoners . .

Zork, The Great Underground Empire, was created by Infocom, Inc., and is available for 32K Apple II and II Plus and 32K S-80 Model I Level II disk systems. \$39.95




TSE-HARDSIDE

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

FROM
**PERSONAL
SOFTWARE**



ASSAULT IV



by Jerald Uptain

"Assault IV" is an S-80 program requiring 16K of RAM.

Welcome to the war-ravaged world of 1993! Things are not too pleasant these days, but you are about to become (in this game at least!) a key figure in making them better! How? By fighting the most critical battle fought by the major powers. The stakes are high! Battlefield success could bring a quick end to the war. However, an enemy victory could plunge the world into a darkness from which it might never emerge! From your point of view, another equally important stake is your personal career. Since Congress approved your last star, it has been no secret that you have political ambitions noted and appreciated in the highest circles. Winning the battle could do much to gain public support. Defeat? Well, let's try to not dwell on that!

This program is self-instructive — just watch the prompts. But remember, the enemy has an excellent electronic intelligence

system and is highly mobile. He will not hesitate to attack when he believes it to his advantage. You will also find that, as in real life, not everything on your side is always under your immediate control. For example, some of your battlefield commanders are prima donnas! I'd be careful giving some of them too much power if I were you!

Oh yes, don't assume too much if you win the first time. The game may be tougher than you think!

New programmers take note! This program was intentionally made up mostly of subroutines so that you may change it to suit your needs. Feel free to do so. Remember, this program, like all material in **SoftSide**, is for your use. Press on (no pun intended!), brave programmer, the future may yet be yours!

VARIABLES

L1, L2, L3, and L4: I, II, III and IVth Corps location.
L5: 8th Army location.
L6: 3rd Army location.
ES: Total enemy strength.

EC: Enemy strength per corps area.
ED: Estimated number of enemy divisions in battle zone.
CG: Net friendly forces ground gain.
SK: Player's score.
MS: Number of player errors.
AA, AD, AG, AJ, AM: Number of infantry division in reserve, I Corps, II Corps, III Corps, and IV Corps.
AB, AE, AH, AK, AN: Number of armored division in reserve, I Corps, II Corps, III Corps, and IV Corps.
AC, AF, AI, AL, AP: Number of airborne division in reserve, I Corps, II Corps, III Corps, and IV Corps.
F1, F2, F3, F4: Total number of divisions in I Corps, II Corps, III Corps, and IV Corps.
A1, A2, A3: Total number of friendly infantry, armored and airborne divisions.
FT: Total number of friendly divisions, all types.
LS: Corps battle loss.
S(-): Corps battle strength.
D(-): Initialization array variable.

```

10 ' *****
20 ' * ASSAULT IV *
30 ' *-----*
40 ' * (C) COPYRIGHT 1980 *
50 ' * BY JERALD V. UPTAIN *
60 ' *****
70 CLS: CLEAR600: DIMD(20)
80 DATA 1,1,1,1,1,1,1,1,1,1,1,2,2,2,3,3,4,4,4,4
90 GOSUB2400: PRINTSTRING$(64,45);: PRINTCHR$(212); "ASSAULT
  IV";: PRINTSTRING$(128,45);
100 PRINT@394, "1 - NEW PLAYERS ( ALL RECRUITS FORM A LINE HERE!
  )";: PRINT@522, "3 - EXPERIENCED PLAYERS ( COMBAT VETERANS ONLY! )
  ";: GOSUB1070: Q$=INKEY$
110 A$=INKEY$: IFA$="1" OR A$="3" THEN 120 ELSE PRINT@666, "( HIT NUMBER
  )";: FORT=1 TO 100: NEXT T: PRINT@666, CHR$(206);: FORT=1 TO 100: NEXT T: GO
  TO 110
120 IFA$="1" THEN GOSUB570
130 GOSUB410: GOSUB440
140 GOSUB1160: CLS: PRINT@462, "BATTLE IN PROGRESS
  !"
  LINES 150-400: Battlefield routine, and end of game routines.

150 '
160 GOSUB1610: KE=ES
170 C=1: L=L1: X=AD: T=AE: A=AF: GOSUB1660: L1=L: AD=X: AE=T: AF=A
180 C=2: L=L2: X=AG: T=AH: A=AI: GOSUB1660: L2=L: AG=X: AH=T: AI=A
190 C=3: L=L3: X=AJ: T=AK: A=AL: GOSUB1660: L3=L: AJ=X: AK=T: AL=A
200 C=4: L=L4: X=AM: T=AN: A=AP: GOSUB1660: L4=L: AM=X: AN=T: AP=A
210 L6=L4-115+64*RND(3): L5=L1-134+64*RND(3)

220 IFL1<198 THEN L1=198
230 IFL1>838 THEN L1=838
240 IFL2<211 THEN L2=211
250 IFL2>851 THEN L2=851
260 IFL3<224 THEN L3=224
270 IFL3>864 THEN L3=864
280 IFL4<237 THEN L4=237
290 IFL4>877 THEN L4=877
300 IFL6<250 THEN L5=250
310 IFL6>890 THEN L5=890
320 IFL5<192 THEN L6=192
330 IFL5>832 THEN L6=832
340 GG=(2150-L1-L2-L3-L4)/64: GG=INT(GG)
350 ED=(4*EC-KE+ES)/8.5: ED=INT(ED): IF ED<0 THEN ED=0
360 GOSUB1470: SK=10*(GG+INT((136-ES)/8.5)-INT(16-FT))-MS
370 GOSUB410: GOSUB440: PRINT@0, E$;: IF (L1=838 OR L2=851 OR L3=864 OR L4=
  877) THEN GOSUB1870 ELSE IF FT<4 THEN GOSUB1970 ELSE IF (GG<-14 OR MS>4) THEN
  GOSUB2080 ELSE IF (GG>120 OR ES<28) THEN GOSUB2180 ELSE GOTO140
380 CLS: PRINT@339, "DO YOU WANT TO TRY AGAIN?";: Q$=INKEY$
390 A$=INKEY$: IFA$="Y" OR A$="N" THEN 400 ELSE PRINT@471, "( HIT 'Y' OR
  'N' )";: FORT=1 TO 100: NEXT T: PRINT@471, CHR$(210);: FORT=1 TO 100: NEXT
  T: GOTO390
400 IFA$="Y" THEN CLS: PRINT@398, "ONE MOMENT PLEASE
  !";: RESTORE: CLEAR600: DIMD(20): GOSUB2400: GOSUB1070: GOTO130 ELSE IF
  A$="N" THEN CLS: PRINT@342, "END OF GAME";: PRINT@468, "TO P
  LAY AGAIN, ENTER 'RUN'";: END

  Lines 410-430: Draw situation map.

410 '
420 CLS: X1$="." + STRING$(12,32): FORT=1 TO 16: PRINTCHR$(197): X1$: X1$
  
```

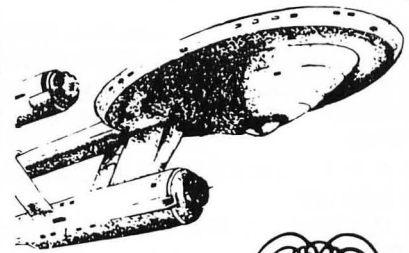
```
;X1$;X1$;" "CHR$(198));NEXTY:PRINT@L1,L1$;PRINT@L2,L2$;PRINT@
L3,L3$;PRINT@L4,L4$;PRINT@L5,L5$;PRINT@L6,L6$;
430 PRINT@960,CHR$(149);" ENEMY DIVS:";ED;PRINT@977,"? "CHR$(1
49);" RESERVE:";PRINT@1005,CHR$(149);" SCORE: ";SK;PRINT@1022
,CHR$(149);RETURN
```

Lines 440-560: Division Locator

```
440 '
450 X=AD;Y=AE;Z=AF;P=L1+64;GOSUB460;X=AG;Y=AH;Z=AI;P=L2+64;GOSUB
460;X=AJ;Y=AK;Z=AL;P=L3+64;GOSUB460;X=AM;Y=AN;Z=AP;P=L4+64;GOSUB
460;X=AA;Y=AB;Z=AC;P=991;GOSUB460;RETURN
460 '
470 A$="";IFX<0THENA$=A$+STRING$(X,88)
480 IFY<0THENA$=A$+STRING$(Y,84)
490 IFZ<0THENA$=A$+STRING$(Z,65)
500 L=LEN(A$);IFL=0THENPRINT@P,CHR$(204);RETURNELSEFORT=1TOL:B$
(T)=MID$(A$,T,1);NEXTT;FORT=1TOL
510 R=RND(L);IFB$(R)=" "THEN510ELSEA$(T)=B$(R);B$(R)=" "NEXTT
520 IFL=1PRINT@P," ";A$(1);" ";
530 IFL=2PRINT@P," ";A$(1);" ";A$(2);" ";
540 IFL=3PRINT@P," ";A$(1);" ";A$(2);" ";A$(3);" ";
550 IFL=4PRINT@P," ";A$(1);" ";A$(2);" ";A$(3);" ";A$(4);" ";
560 FORT=1T04;A$(T)="";B$(T)="";NEXTT;RETURN
```

Lines 570-1060: Program orientation. Gives background information about the situation being simulated.

```
570 '
580 CLS:PRINT@0,STRING$(63,144);PRINT@960,STRING$(63,144);PRIN
T@198,"IT IS MARCH 20, 1993. THE WORLD HAS BEEN AT WAR FOR TWO"
590 PRINT"YEARS. SO FAR, ONLY 'CONVENTIONAL' WEAPONS HAVE BEEN
USED. AT"
600 PRINT"PRESENT, THE CONFLICT HAS BEEN REDUCED TO A EUROPEAN L
AND WAR."
610 PRINT"MOST ALLIED COMMANDERS BELIEVE THAT VICTORY IS CLOSE A
T HAND."
620 PRINT"HOWEVER, INTELLIGENCE HAS CONFIRMED THAT THE ENEMY IS
MASSING"
630 PRINT"HIS STRONGEST COMBAT DIVISIONS IN AN AREA APPROXIMATEL
Y 100"
640 PRINT"KILOMETERS NORTH OF DRESDEN IN PREPARATION FOR A SLASH
ING DRIVE"
650 PRINT"ACROSS GERMANY, NORPAC COMMAND HAS DECIDED TO CRUSH T
HE ENEMY"
660 PRINT"BUILDUP WITH AN ATTACK BY THE '5TH ARMY GROUP.' 8TH A
ND 3RD"
670 PRINT"ARMIES WILL PROVIDE FLANK SUPPORT."
680 PRINT:GOSUB1130:CLS
690 PRINT@0,STRING$(63,144);PRINT@960,STRING$(63,144);PRINT@19
8,"YOU ARE THE COMMANDING GENERAL OF THE US SOUTHERN DESERT"
700 PRINT"COMMAND. EARLIER IN THE DAY, YOU WERE TOLD BY THE COM
MANDER OF"
710 PRINT"NORPAC THAT YOU WOULD BE PICKED TO COMMAND THE 5TH ARM
Y GROUP"
720 PRINT"BECAUSE OF YOUR PROVEN ABILITY DURING THE PAST MONTH.
AS YOU"
730 PRINT"PONDER THE CHALLENGE, AN AIDE BRINGS YOU THE FOLLOWING
LETTER,"
740 PRINT"WHICH HAS JUST ARRIVED:";PRINT:GOSUB1130
750 '
760 CLS:PRINTCHR$(213);"S T R A T S E C";PRINTCHR$(213);FORT=1T
015:PRINTCHR$(131);NEXTT:PRINTCHR$(13);
770 PRINTCHR$(214);"HEADQUARTERS"
780 PRINTCHR$(209);"NORPAC COMMAND, EUROPE"
790 PRINT:PRINT"DEL: ARMED COURIER";CHR$(220);"17 MARCH 199
3"
800 PRINT"SUBJECT: CONFIRMATION OF ASSIGNMENT" continued on next page
```



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 \$14.95

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

continued from previous page

```
810 PRINT:PRINT"TO:      COMMANDING GENERAL"
820 PRINTCHR$(202);"SOUTHERN DESERT COMMAND"
830 PRINTCHR$(202);"APO 765, US FORCES":PRINT
840 PRINT"  1. PROCEED IMMEDIATELY TO SITCON LOC. 4B AND ASSUME
  COMMAND"
850 PRINT"OF 'STRIKEDOWN' PER OUR CONVERSATION THIS DATE."
860 GOSUB1130
870 CLS:PRINT"  2. IMPLEMENT PLAN 'VICTOR' NOT LATER THAN 0700
  HOURS,"
880 PRINT"10 APRIL 1993. YOU WILL BE SUPPORTED BY 'BIG RED' AND
  'BLUE'"
890 PRINT"BOY,' AS AGREED."
900 PRINT"  3. BE PREPARED TO EXECUTE PLAN 'QUICKFLASH' ON THE
  COMMAND"
910 PRINT"OF THIS HEADQUARTERS, SHOULD USE OF ATOMICS BECOME NEC
  ESSARY."
920 PRINT"  4. REPORT LATEST BATTLEFIELD INTELLIGENCE TO THIS"
930 PRINT"HEADQUARTERS NOT LATER THAN 2400 HOURS THIS DATE, VIA
  'BABELX'"
940 PRINT"CODE ONLY."
950 PRINT:PRINTCHR$(198);"FOR THE COMMANDER:"
960 PRINTCHR$(217);"HARRISON C. SMYTHE, GENERAL"
970 PRINTCHR$(217);"DEPUTY COMMANDER, NORPAC COMMAND"
980 PRINT:PRINTCHR$(213);"S T R A T S E C":PRINTCHR$(213);:FORT=
  1T015:PRINTCHR$(131);:NEXTT:PRINTCHR$(13);
990 GOSUB1130
1000 '
1010 CLS:PRINT@,STRING$(63,144);:PRINT@960,STRING$(63,144);:PRI
  NT@196,"YOU HAVE JUST ARRIVED AT SITCON LOC. 4B (YOUR COMMAND PO
  ST).";
1020 PRINT"YOU ARE BEING BRIEFED ON THE BATTLEFIELD SITUATION,
  BEFORE YOU"
1030 PRINT"IS YOUR COMMAND BOARD, WHICH IS MUCH LIKE THE EARLY"
1040 PRINT"MICROCOMPUTERS USED LAST DECADE. FROM HERE, YOU WIL
  L"
1050 PRINT"CONTROL THE ALLIED FORCES UNDER YOUR COMMAND.":PRINT:
  GOSUB1130
1060 RETURN
```

Lines 1070-1120: Sets up initial battle positions.

```
1070 '
1080 GOSUB1090:AA=A:AB=B:AC=C:GOSUB1090:AD=A:AE=B:AF=C:GOSUB1090
:AG=A:AH=B:AI=C:GOSUB1090:AJ=A:AK=B:AL=C:GOSUB1090:AM=A:AN=B:AP=
  C:RETURN
1090 '
1100 A=0:B=0:C=0:FORK=1T04
1110 R=RND(20):IFD(R)=0THEN1110ELSEIFD(R)=1THENA=A+1ELSEIFD(R)=2
  THENB=B+1ELSEIFD(R)=3THENC=C+1
1120 D(R)=0:NEXTK:RETURN
```

Lines 1130-1150: Delay program execution until the Enter key is hit.

```
1130 '
1140 Q$=INKEY$
1150 AZ$=INKEY$:IFAZ$=CHR$(13)THENRETURNELSEPRINTCHR$(212);"(PRE
  SS 'ENTER' TO CONTINUE)";CHR$(29);:FORT=1T0100:NEXTT:PRINTCHR$(2
  54);CHR$(29);:FORT=1T050:NEXTT:GOTO1150
```

Lines 1160-1460: Battle command center. Allows repositioning of forces.

```
1200 PRINT@,E$;:PRINT@,"TYPE OF DIVISION YOU WISH TO MOVE?":PR
  INT@64," X = INFANTRY T = ARMORED A = AIRBORNE";:Q$=INKEY$
1210 A$=INKEY$:IFA$="X"ORAS$="T"ORAS$="A"THEN1220ELSEPRINT@149,"(H
  IT 'X', 'T', OR 'A')";:FORT=1T0100:NEXTT:PRINT@149,CHR$(214);:FO
  RT=1T050:NEXTT:GOTO1210
```

```
1160 '
1170 GOSUB1470:PRINT@,E$;:PRINT@,"YOU HAVE";FT;" DIVISIONS, LO
  CATED AS SHOWN.":PRINT@69,"X = INFANTRY T = ARMORED A = AIRB
  ORNE":PRINT@128,"DO YOU WANT TO RE-POSITION THEM?";:Q$=INKEY$
1180 A$=INKEY$:IFA$="Y"ORAS$="N"THEN1190ELSEPRINT@168,"(HIT 'Y' O
  R 'N')";:FORT=1T0100:NEXTT:PRINT@168,CHR$(208);:FORT=1T050:NEXTT
  :GOTO1180
1190 XX=(F1=0ORF2=0ORF3=0ORF4=0):IFXXANDA$="N"THEN1450ELSEIFA$="
  N"PRINT@,E$;:RETURN
1220 IFA$="X"THENV=1ELSEIFA$="T"THENV=2ELSEIFA$="A"THENV=3
1230 IF(V=1ANDA1=0)OR(V=2ANDA2=0)OR(V=3ANDA3=0)PRINT@64,"YOU HAV
  E NO ;DV$(V);" DIVISIONS LEFT!";GOSUB890:MS=MS+1:GOSUB1700:GOTO9
  10
1240 R=RND(30):IFR=13THENGOSUB2320:RETURN
1250 R=RND(100):IFR<5THENGOSUB2350:IFKK=1THENRETURN
1260 PRINT@,E$;:PRINT@,"WHERE DOES THE ";DV$(V);" DIVISION MOV
  E FROM?":PRINT@64,"(1=I CORPS, 2=II CORPS, 3=III CORPS 4=IV CORP
  S, 5= RESERVE)";:Q$=INKEY$
1270 B$=INKEY$:IFB$="1"ORB$="2"ORB$="3"ORB$="4"ORB$="5"THEN1280E
  LSEPRINT@154,"(HIT NUMBER)";:FORT=1T0100:NEXTT:PRINT@154,CHR$(20
  4);:FORT=1T050:NEXTT:GOTO1270
1280 X=(A$="X"):Y=(A$="T"):Z=(A$="A");L=VAL(B$):K1=0;K2=0;K3=0
1290 K1=((XANDL=1ANDAD<1)OR(XANDL=2ANDAG<1)OR(XANDL=3ANDAJ<1)OR(
  XANDL=4ANDAM<1)OR(XANDL=5ANDAA<1));K2=((YANDL=1ANDAE<1)OR(YANDL=
  2ANDAH<1)OR(YANDL=3ANDAK<1)OR(YANDL=4ANDAN<1)OR(YANDL=5ANDAP<1))
1300 K3=((ZANDL=1ANDAF<1)OR(ZANDL=2ANDAI<1)OR(ZANDL=3ANDAL<1)OR(
  ZANDL=4ANDAP<1)OR(ZANDL=5ANDAC<1))
1310 IFK1ORK2ORK3PRINT@128,"YOU HAVE NO ";DV$(V);" DIVISIONS IN
  ";LO$(L);"!";GOSUB1130:MS=MS+1:GOSUB2280:GOTO1260
1320 PRINT@,E$;:PRINT@,"WHERE DOES THE ";DV$(V);" DIVISION MOV
  E TO?":PRINT@64,"(1=I CORPS, 2=II CORPS, 3=III CORPS, 4=IV CORPS
  , 5=RESERVE)";:Q$=INKEY$
1330 R=RND(30):IFR=13THENGOSUB2320:RETURN
1340 R=RND(100):IFR<5THENGOSUB2350:IFKK=1THENRETURN
1350 C$=INKEY$:IFC$="1"ORC$="2"ORC$="3"ORC$="4"ORC$="5"THEN1360E
  LSEPRINT@154,"(HIT NUMBER)";:FORT=1T0100:NEXTT:PRINT@154,CHR$(20
  4);:FORT=1T050:NEXTT:GOTO1350
1360 C=VAL(C$):IF(C=1ANDF1>3)OR(C=2ANDF2>3)OR(C=3ANDF3>3)OR(C=4A
  NDF4>3)OR(C=5ANDFR>3)PRINT@,E$;:PRINT@,"SORRY, GENERAL, THERE'
  S ROOM FOR ONLY 4 DIVISIONS!";GOSUB1130:MS=MS+1:GOSUB2280:GOTO13
  20
1370 IFC=LTHENPRINT@,E$;:PRINT@,"THE DIVISION IS ALREADY IN ";
  LO$(C);"!";GOSUB1130:MS=MS+1:GOSUB2280:GOTO1320
1380 R=RND(30):IFR=13THENGOSUB2320:RETURN
1390 R=RND(100):IFR<5THENGOSUB2350:IFKK=1THENRETURN
1400 ONV GOSUB1490,1530,1570
1410 GOSUB440:PRINT@,E$;:PRINT@,"THE DIVISION IS RELOCATED! DO
  YOU WANT TO RE-POSITION OTHERS?";:Q$=INKEY$
1420 A$=INKEY$:IFA$="Y"ORAS$="N"THEN1430ELSEPRINT@88,"(HIT 'Y' OR
  'N')";:FORT=1T0100:NEXTT:PRINT@88,CHR$(208);:FORT=1T050:NEXTT:G
  OTO1420
1430 GOSUB1470:XX=(F1=0ORF2=0ORF3=0ORF4=0)
1440 IFA$="Y"THEN1200
1450 IFA$="N"ANDXXPRINT@,E$;:PRINT@,"YOU CANNOT LEAVE A CORPS
  AREA UNDEFENDED!";GOSUB1130:MS=MS+1:GOSUB2280:GOTO1200
1460 PRINT@,E$;:RETURN
```

Lines 1470-1480: Total friendly divisions.

```
1470 '
1480 A1=AA+AD+AG+AJ+AM:A2=AB+AE+AH+AK+AN:A3=AC+AF+AI+AL+AP:FR=AA
+AB+AC:F1=AD+AE+AF:F2=AG+AH+AI:F3=AJ+AK+AL:F4=AM+AN+AP:FT=A1+A2+
  A3:RETURN
```

Lines 1490-1520: Infantry move verifier.

```
1490 '
1500 IFL=1THENAD=AD-1ELSEIFL=2THENAG=AG-1ELSEIFL=3THENAJ=AJ-1ELS
  EIFL=4THENAM=AM-1ELSEIFL=5THENAA=AA-1
```

continued on page 63

Do You Have What it Takes to be a **SoftSide Editor?**

Are you computer literate?
Are you just plain literate?
Are you both? If so, then you
may be the person **SoftSide**
is looking for. We need
people who like to write both
in English and in BASIC to
work on our editorial staff.

As a leader in the field of
software publishing,
SoftSide is committed to
providing its readers with
clear, concise, and easy to
understand articles and
programs for the TRS-80™,
Apple, and Atari
microcomputers. Our growth
rate is such that our staff is
constantly expanding and we
are always on the lookout for
those with the special skills
required to put out a quality
magazine.

While New Hampshire wages are not the highest
around, we offer a better scale of pay than most of
our competitors. Furthermore, our fringe benefits are
excellent. Another bonus is that New Hampshire has
no personal income or sales tax.

Located in a beautiful area of southern New
Hampshire, Milford is within an hour's distance of
Boston, the Atlantic Ocean, and the White
Mountains. Nashua and Manchester, New
Hampshire's two largest cities, are but 15 minutes
from our offices. Ski slopes abound in the region, as
do fine restaurants, arts and crafts centers, and
countryside the likes of which must be seen to be
believed.

If you are a hard-working, skilled person with
experience in the fields of computers and magazines,
and if you enjoy the relaxed country lifestyle with the
convenience and cultural opportunities of major
urban centers nearby, drop us a line. We might be
good for each other.

If you feel that you fit the bill, please write to us
and let us know who you are and what you can do. A
formal resume isn't a necessity, just send a letter
outlining your skills, qualifications, and experience
to:

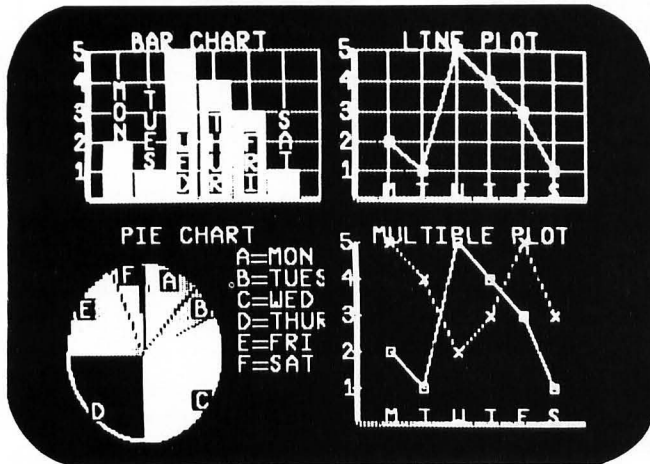
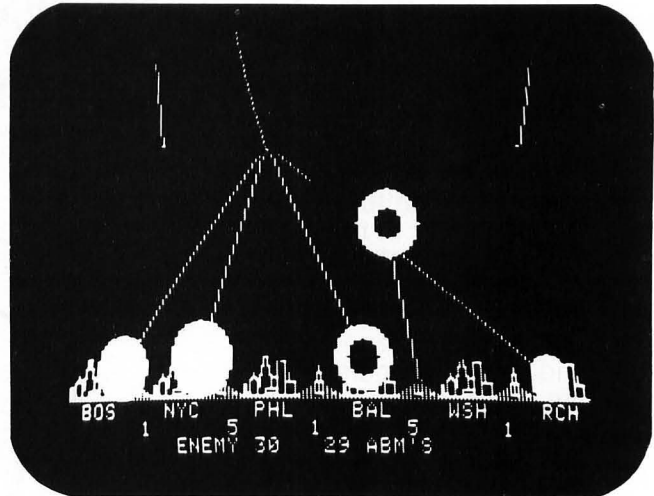
Dave Albert,
SoftSide Publications,
6 South Street,
Milford, New Hampshire
03055



TWO NEW ONES FOR THE APPLE II® OR II plus®

ABM

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)



DATA PLOT

Easy editing features allow you to create and modify a wide variety of full color graphic representations of numerical information. Bar charts, including additive bars, as well as single and multiple line charts may be plotted individually or cumulatively. Pie charts are easily sliced. All figures may be output to a graphics printer or saved as hi-res "pictures" for dramatic full color recall as visual aids during presentations. Basic statistics are displayed automatically. On disk, requires 48 k and Applesoft ROM. (\$59.95)

from the leader in quality software

MUSE SOFTWARE™

Apple II is a trademark of Apple Computer Corp.



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



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. \$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.

..... \$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, domino, 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.



DOMINOES

by Peter Kirsch

"Dominoes" is an S-80 program requiring 16K of memory.

Take a stroll through the boneyard with your S-80. Here for your delight are three versions of the classic game of dominoes. Your computer is an astute opponent, so be ready for a hard fought game, no matter which version you choose.

All the rules for play are included in the program itself, as well as some thoughts on the game by the author. But the best way to explore domino theory, short of watching newsreels from the '50s, is to play. Type it in and find out for yourself.

VARIABLES

A1: In Draw game, the number of pips in the tile the computer drew to select starter.

A2: Same as A1 above, but for player.

A3: Used to determine starting double.

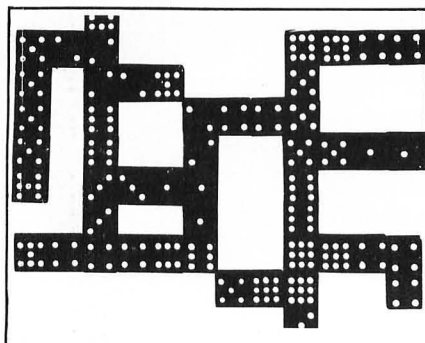
A\$(): Boneyard.

B1: In Draw game, the number of pips in the tile the player drew to select starter.

BE: Flag used in Bergen. BE = 1 if the current player has already drawn a bone from the boneyard in this turn.

BN: Number of bones in yard.

C(): Quantity of each set of pips computer currently holds. Needed



for best play of computer.

D: Game number being played.

D(): List of screen positions for bottom of screen tile display.

DM\$: Contains the left and right ends of the domino chain.

G: Graphic block for drawing bones. CHR\$(140)

G\$(): Names of each game.

GM: Set to 1 after the first time through the game. When equal to 1, the instruction routine is skipped.

GO: Player going out. 1 = Player in game 1, 2 = Computer in game 1, 3 = Player in game 2, 4 = Computer in game 2.

GT: Score needed to win.

H: Number of pips in either the lowest or highest double in the player's hand at the start of the game. Used to determine starter in Block and Bergen games.

HM: 1 if human player has a double at start of game.

L: Number of pips in a certain side of a bone.

LD: One if left tile is a double.

LT: Number of pips on left side of chain.

MD\$: Contains the right and left ends of the domino chain.

P: Top of screen memory (15360).

P\$(): Player's tiles.

P1\$(): Temporary storage of player's tiles.

PB: Number of bones player has.

PL: Player control variable.

1 = Player's turn, 2 = Computer's turn.

Q1: Used to determine where computer's tile should be placed. 1 = on left of chain, 2 = on right side.

RD: One if right tile is a double.

RT: Number of pips on right side of chain.

S: Value of right portion of computer's tile.

SP: Total number of pips player has at end-of-hand.

ST: Total number of pips computer has at end-of-hand.

T: Same as H above, but for the computer.

T\$(): Computer's tiles.

TB: Number of bones computer has.

TP: Player's score.

TR: Used if computer has a double at start of game.

TT: Computer's score.

W: For/Next variable to cycle through each bone in the computer's hand.

W1: Computer's play to be used if no others plays exist.

X\$: A short graphic line used to underline current player on screen display.

X1\$: A string of blanks used to erase the line created by X\$ above.

<pre> Lines 5-40: Heading display and variable initialization. 5 CLS:PRINTCHR\$(23)," DOMINOES VERSION 2.1 6 ' BY PETER KIRSCH JULY 1980 10 CLEAR1000:RANDOM:GT=100:G=140:P=15360:DIM(28),D(20),P\$(20) ,T\$(20),P1\$(20) 15 G\$(1)="BLOCK GAME":G\$(2)="DRAW GAME":G\$(3)="BERGEN ""X\$=ST RING\$(3,131):X1\$="" ""CH=-1:PB=7:TB=7:BN=14:DB=1 20 C=1:FORA=0TO6:FORB=0+ATO6:A\$(C)=STR\$(A)+STR\$(B):C=C+1:NEXT:NE XT 22 IFGM=1THEN190ELSEGM=1 25 FORA=1TO16:D(A)=896+(A*3):NEXT </pre>	<pre> 30 GOSUB2000 40 PRINT@330,"DOMINOES";:PRINT@342,"DOMINOES"; Lines 50-66: Game selection. 50 PRINT@512,"CHOOSE GAME";:FORA=1TO3:PRINT@530+A*64,ACHR\$(24)", "G\$(A);:NEXT 60 A\$=INKEY\$:FORB=1TO125:NEXT:POKE16007,63:FORA=1TO125:NEXT:POKE 16007,32:IFA\$=""THEN60 65 D=VAL(A\$):IFD<1ORD>3THEN60ELSEIFD=3THENGT=15ELSEIFD=1THENGT=5 0 66 IFD=2GS=1 Lines 70-99: Instructions. 70 PRINT@512,G\$(D)" "";:PRINT@595,"DO YOU WANT "";:PRINT@659," </pre>
--	--

```

RULES? ";;PRINT@723," (Y/N) ";;PRINT@787,"
";
71 A$=INKEY$:IFA$=""THEN7ELSEIFA$="N"THEN10ELSEIFA$="Y"THEN80E
LSE71
80 CLS:PRINT"DOMINOES ARE PLAYED WITH A DOUBLE-SIX PACK OF 28 TI
LES (CALLED";PRINT"BONES) WITH PIPS (OR SPOTS) RANGING FROM 0/0
TO 6/6. PLAYERS";PRINT"ARE DEALT 6 OR 7 BONES (DEPENDING ON GAME
). THE REST OF THE"

82 PRINT"PACK IS CALLED THE BONEYARD (OR JUST YARD) AND IS SET A
SIDE FOR";PRINT"LATER, THE PRIMARY OBJECT OF DOMINOES IS TO BE T
HE FIRST TO";PRINT"PLAY ALL HIS TILES AND GO OUT, OR HAVE THE LO
WEST PIP COUNT"
84 PRINT"LEFT IN HAND WHEN PLAY ENDS. PLAYERS, IN TURN, LAY TILE
S IN A";PRINT"LONG CHAIN, MATCHING ADJACENT ENDS. ";PRINT:PRINT"THE
LARGE DOMINO IN THE DISPLAY SHOWS BOTH ENDS OF THE CHAIN. ";PR
INT"YOU NEED MERELY TO KEY IN THE NUMBERS OF THE TILE YOU WISH
TO"
86 PRINT"PLAY AND IT WILL BE PLACED FOR YOU. YOU NEED NOT PRESS
ENTER. ";PRINT"YOUR DOMINOES WILL BE DISPLAYED ALONG THE BOTTOM O
F THE SCREEN. ";PRINT:PRINT"STARTING (SETTING) AND SCORING DIFFER
S WITH GAME PLAYED. ";GOSUB5000

87 CLS:PRINTG$(D):PRINT:ONDGOTO93,96,88
88 PRINT"PLAYERS ARE DEALT 6 BONES INSTEAD OF 7. GAME IS PLAYED
TO 15";PRINT"POINTS. THE LOWEST DOUBLE MUST BE SET AND THE HOLDE
R SCORES 2";PRINT"POINTS. THE OBJECT OF THE GAME IS TO MATCH BOT
H ENDS OF THE"
89 PRINT"CHAIN, MAKING BOTH ENDS ALIKE. HE SCORES 2 POINTS FOR A
";PRINT"DOUBLE-HEADER. IF ONE END OF THE CHAIN HAS A DOUBLE OR I
F THE";PRINT"PLAYER PLAYS A DOUBLE, HE SCORES 3 POINTS FOR A TRI
PLE-HEADER."

90 PRINT"A SMALL 'D' INSIDE A CUBICLE ON THE DOMINO ON THE DISPL
AY";PRINT"INDICATES A DOUBLE ON THAT END OF THE CHAIN. IT'S JUST
A";PRINT"REMINDER. IF A PLAYER GOES OUT, HE GETS 1 ADDITIONAL P
OINT. ALL"
91 PRINT"BONES IN THE YARD ARE PLAYED. IF YOU CAN NOT PLAY YOU M
UST";PRINT"DRAW ONE BONE AND ONLY ONE. IF YOU STILL CAN NOT PLAY
, YOU MUST";PRINT"PASS. IF THE YARD IS USED UP, 2 CONSECUTIVE PA
SSES ENDS PLAY. ";GOSUB5000;GOSUB2000;GOTO100
93 PRINT"PLAYER HAVING THE HIGHEST DOUBLE MUST START AND SET IT.
THERE";PRINT"IS NO DRAW FROM THE YARD. PLAYERS PLAY ONLY WITH T
ILES DEALT. ";PRINT"WHEN YOU CAN'T PLAY, YOU MUST PASS. TWO CONSE
CUTIVE PASSES"
94 PRINT"ENDS PLAY. GAME IS PLAYED TO 50 POINTS. ";GOSUB98;GOSUB2
000;GOTO100

96 PRINT"PLAYERS DRAW BONES TO SEE WHO SETS FIRST, HIGHEST TILE
(TOTAL";PRINT"PIPS) STARTING AND PLACING ANY HE OWNS. IF A PLAYE
R CAN NOT";PRINT"PLAY A TILE, HE MUST DRAW ONE OR MORE FROM THE
YARD UNTIL HE"
97 PRINT"CAN PLAY AND PLAY IT. THE LAST 2 BONES IN THE YARD ARE
FROZEN";PRINT"AND MAY NOT BE PICKED. A PLAYER MAY NOT PASS UNTIL
THEN. TWO";PRINT"CONSECUTIVE PASSES ENDS PLAY. GAME IS PLAYED T
O 100 POINTS."
98 PRINT:PRINT"SCORING: IF ONE PLAYER GOES OUT, HE SCORES THE SU
M OF THE PIPS";PRINT"ON HIS OPPONENT'S UNPLAYED TILES. IF NO ONE
GOES OUT, THE";PRINT"PLAYER WITH THE FEWEST REMAINING PIPS WINS
AND SCORES THE"
99 PRINT"DIFFERENCE BETWEEN HIS AND HIS OPPONENT'S. ";GOSUB5000;I
FD=1THENRETURNELSEGOSUB2000

100 IFD>2THEN190ELSEPRINT@512," ";;GOSUB150:PRINT@583,"
WE'LL DRAW TO SEE WHO STARTS";;FORA=1TO2000:NEXT:CH=0
110 A=RND(28):B=RND(28):IFA=BTHEN110ELSEA1=VAL(MID$(A$(A),2,1))+
VAL(RIGHT$(A$(A),1)):B1=VAL(MID$(B$(B),2,1))+VAL(RIGHT$(B$(B),1
)):PRINT@583,"I DREW A"A$(A)", YOU DREW A"A$(B);
120 IFA1=B1PRINT@647,"WE BOTH HAVE"A1"PIPS";;PRINT@711,"WE'LL RE
DRAW";;FORA=1TO2000:NEXT:GOSUB150:FORA=1TO999:NEXT:GOTO110

130 IFA1>B1PRINT@647,"I HAVE"A1"AND YOU HAVE"B1"PIPS";;PRINT@711
,"I START";;FORA=1TO3000:NEXT:GOSUB150:Z2=2:PL=1;GOTO190
140 IFB1>A1PRINT@647,"YOU HAVE"B1"AND I HAVE"A1"PIPS";;PRINT@711
,"YOU START";;FORA=1TO3000:NEXT:Z2=1:PL=2;GOSUB150;GOTO190
150 PRINT@583,STRING$(30,32);;PRINT@647,STRING$(30,32);;PRINT@71
1," ";;RETURN
190 IFD=3THENTB=6:PB=6:BN=16
199 '
200 FORA=1TOPB
210 C=RND(28):IFA$(C)=""THEN210ELSEP$(A)=A$(C):A$(C)=""NEX:FOR
A=1TOTB

220 C=RND(28):IFA$(C)=""THEN220ELSEP$(A)=A$(C):A$(C)=""GOSUB300
0:NEXT:PRINT@43,C$(D);;PRINT@0,"SCORE:";;PRINT@14,"TRS-80:";;PRI
NT@27,"YOU:";;PRINT@172,"TILES:";;PRINT@243,"TRS-80:";;PRINT@307
,"YOU:";;IFD>1PRINT@371,"YARD:";
225 PRINT@492,"LAST PLAYED:";;PRINT@563,"TRS-80:";;PRINT@627,"YO
U:";;PRINT@748,"CHAIN LENGTH:";;PRINT@812,"GAME: "GT"PTS";
230 GOSUB1100:FORA=1TOPB:PRINT@D(A),MID$(P$(A),2,1);;PRINT@D(A)+
64,RIGHT$(P$(A),1);;NEXT:PRINT@951,"< BONES";;PRINT@330,"
";;PRINT@342," ";;PRINT@512,STRING$(16,32);;PRINT@662,"
";;PRINT@726," ";;ONDGOTO235,231,260

231 IFZ2=1PRINT@586,"YOUR TURN TO SET ";;FORA=1TO1500:NEXT:PL
=2;GOTO300
232 IFZ2=2PRINT@586,"MY TURN TO SET ";;FORA=1TO1500:NEXT:PL=
1;GOTO300
235 A3=7:PRINT@586,"HIGHEST DOUBLE STARTS";;GOTO280
260 A3=1:PRINT@586,"LOWEST DOUBLE STARTS";
280 A3=A3-1:FORA=1TOPB:A2$=STR$(ABS(A3)):A2$=MID$(A2$,2,1):IFMID
$(P$(A),2,1)=A2$ANDRIGHT$(P$(A),1)=A2$H=ABS(A3):H#1:GOTO288
285 IF((A3=0)*(D=1))+((A3=-6)*(D=3))THEN288ELSENEXTA:GOTO280
288 IFD=1THENA3=7ELSEIFD=3A3=1

290 A3=A3-1:FORH=1TOTB:A2$=STR$(ABS(A3)):A2$=MID$(A2$,2,1):IFMID
$(T$(H),2,1)=A2$ANDRIGHT$(T$(H),1)=A2$T=ABS(A3):TR=1;GOTO296
295 IF((A3=0)*(D=1))+((A3=-6)*(D=3))THEN900ELSENEXT:GOTO290
296 GOSUB9100:PRINT@586,"";;IFD=1IF(H>T)+(TR=0)THENL=H:PRINT"YOU
'RE HIGH WITH "H"/"H";PRINT@636,"SET";;PL=2;GOTO900ELSEL=T:PRINT
"I HAVE A "T"/"T" ";;PRINT@572,"SET";;PL=1:C(T)=C(T)-2;GOT
O900
297 IFD=3IF((H>T)*(TR=1))+((H#0)*(TR=1))THENL=T:PRINT"I HAVE A
"T"/"T" ";;TT=TT+2:C(T)=C(T)-2;GOSUB1400:PRINT@572,"SET";;P
L=1;GOTO900ELSEIFH#1L=H:PRINT"YOU'RE LOW WITH A "H"/"H";TP=TP+2
;GOSUB1400:PRINT@636,"SET";;PL=2;GOTO900
299 REM PRINT@128,"";;FORJ=1TOTB:PRINTT$(J) " ";;NEXT:PRINT"
";;TO SEE COMPUTER PLAY ITS DOMINOES FACE UP, REMOVE REM

Lines 300-392: Asks player what he would like to play,
validates the play, and determines where it will be placed.

300 Q1=0:DB=0;GOSUB1100;GOSUB1300:IF((TB=0)+(PB=0))+((Q6=1)*(Q7=
1))THEN1200ELSEPRINT@586,STRING$(23,32);;IFPL=1THENPRINT@80,X$;
PRINT@91,X1$;GOTO500ELSEPRINT@91,X$;PRINT@80,X1$;PRINT@704,"O
PTIONS:";
303 PRINT@715,"PLAY (##) OR";;PRINT@730,"PASS (P) ";;IF (
D>1)*(BN=2)+(D=3)*(BE=0)PRINT@730,"DRAW (D) ";;

```

continued on next page

Dominoes

continued from previous page

```
305 Q6=0:PRINT@320,"PLAYED ";:PRINT@384,"? / ?";
310 A$=INKEY$:IFA$=""THEN310ELSEIFA$="P"THENIFD>1ANDBN<2ANDBE=0P
RINT@586,"YOU MUST DRAW";:GOTO310ELSE380

312 IFA$="D"IF(BN=0)+(BE=1)PRINT@586,"SORRY, YOU CAN'T";:GOTO310
315 IFA$="D"IFD=1THENPRINT@586,"NOT IN THIS GAME";:GOTO310ELSEIF
(D=2)*(BN<3)PRINT@586,"* YARD IS FROZEN *";:GOTO310ELSE390
320 IFASC(A$)<48ORASC(A$)>54THEN310
322 PRINT@384,A$;
325 B$=INKEY$:IFB$=""THEN325ELSEIFASC(B$)<48ORASC(B$)>54THEN325
330 C$=A$+B$;D$=RIGHT$(C$,1)+LEFT$(C$,1);FORA=1TODB:Y$=MID$(P$(A
),2,1);F$=RIGHT$(P$(A),1);E$=Y$+F$;IF(E$=C$)+(E$=D$)THEN335ELSE
EXT:PRINT@586,"YOU DON'T HAVE THAT ONE";:GOTO305

335 PRINT@636,A$+"B$ ";:PRINT@388,B$;IF(RT=LT)+(GS=1)THEN339E
LSEIF((VAL(A$)=LT)*(VAL(B$)=RT))+((VAL(A$)=RT)*(VAL(B$)=LT))THEN
PRINT@704," AT WHICH END, PLEASE, L OR R ? ";:POKE15885
,76;POKE15899,82ELSE339
338 Z$=INKEY$:IFZ$=""THEN338ELSEPOKE15885,32;POKE15899,32;IFZ$="
L"THENL=RT;GOTO900ELSEIFZ$="R"THENL=LT;GOTO950ELSE338

339 PRINT@388,B$;:IFD=2IFPL=2IFZ2=1IFGS=1GS=0:L=VAL(A$);CH=-1:HA
=1;Z2=2;GOTO900
340 IFVAL(A$)=LTHENL=VAL(B$);GOTO900ELSEIFVAL(A$)=RTTHENL=VAL(B
$);GOTO950
350 IFVAL(B$)=LTHENL=VAL(A$);GOTO900ELSEIFVAL(B$)=RTTHENL=VAL(A
$);GOTO950
360 PRINT@586,"YOU CAN'T MATCH IT ";:GOTO305
380 Q6=1:PRINT@636,"PASS";:PL=1:BE=0:IFD=3IFBN<0THENQ6=0
385 GOTO300
390 BN=BN-1:PB=PB+1:PRINT@636,"DRAW";:IFD=3BE=1
392 C=RND(28):IFA$(C)=""THEN392ELSEP$(PB)=A$(C):A$(C)=""A=PB:PR
INT@D(A),MID$(P$(A),2,1);:PRINT@D(A)+64,RIGHT$(P$(A),1);:GOTO300
```

Lines 500-540: Computer's play, like human's above.

```
500 Q7=0:PRINT@320," ";:PRINT@384," ";:PRINT@704,STRING
$(35,32);:IFD=2IFGS=1IFZ2=2Z2=1:GS=0:HA=1:H=RND(6)+1:L=VAL(MID$(
T$(W),2,1);S1=VAL(RIGHT$(T$(W),1));PL=1;GOSUB980;GOTO900
501 IF(D=3)*(DB=0)THEN600
502 FORA=1TODB:S=VAL(RIGHT$(T$(W),1))
503 IFVAL(MID$(T$(W),2,1))=LTIFF(S)>1THENL=S:S1=LT;GOSUB980;GOTO
900ELSEL=S:Q1=1:S1=LT;W1=W
504 NEXT
505 FORA=1TODB
506 S=VAL(RIGHT$(T$(W),1))

507 IFVAL(MID$(T$(W),2,1))=RTIFF(S)>1THENL=S:S1=RT;GOSUB980;GOTO
950ELSEL=S:Q1=2:S1=RT;W1=W
508 NEXT
509 FORA=1TODB
510 S=VAL(MID$(T$(W),2,1))
511 IFVAL(RIGHT$(T$(W),1))=LTIFF(S)>1THENL=S:S1=LT;GOSUB980;GOTO
900ELSEL=S:Q1=1:S1=LT;W1=W
512 NEXT
513 FORA=1TODB
514 S=VAL(MID$(T$(W),2,1))
515 IFVAL(RIGHT$(T$(W),1))=RTIFF(S)>1THENL=S:S1=RT;GOSUB980;GOTC
950ELSEL=S:Q1=2:S1=RT;W1=W
516 NEXT
517 IFQ1<0THENH=W1;GOSUB980;IFQ1=1THEN900ELSE950
```

```
518 IF((BN<3)*(D<3))+(D=1)+(BE=1)+(BN=0)THEN530ELSEPRINT@572,"D
RAW";:PRINT@586,"I'LL DRAW FROM THE YARD";:BN=BN-1:TB=TB+1:IFD=3
BE=1
519 C=RND(28):IFA$(C)=""THEN519ELSEFORA=1TODB:IFT$(A)=""THEN$(A
)=A$(C)ELSENEXT:T$(TB)=A$(C):A=TB
520 A$(C)=""GOSUB3000:FORA=1TOD2000:NEXT;GOTO299
530 Q7=1:PRINT@572,"PASS";:PRINT@586,"I'LL HAVE TO PASS";:BE=0:P
L=2:FORA=1TOD2000:NEXT;IFD=3IFBN<0THENQ7=0
540 GOTO300

600 DM$=STR$(LT)+STR$(RT);MD$=STR$(RT)+STR$(LT);FORA=1TODB:IFT$(
W)=DM$THEN610ELSENEXT
605 FORA=1TODB:IFT$(W)=MD$THEN620ELSENEXT;GOTO502
610 L=VAL(RIGHT$(DM$,1));S1=LT;GOSUB980;GOTO900
620 L=VAL(MID$(MD$,2,1));S1=LT;GOSUB980;GOTO900

900 BE=0;POKE15690,32;CH=CH+1;LT=L;PRINT@266," ";:POKE1
5694,32;PRINT@394," ";:IFL=1ORL=3ORL=5THENGOSUB1000ELSE
IFL<0GOSUB1010
902 GOSUB1510;LD=0
903 IFHA=1HA=0:L=VAL(B$);S1=L;GOTO950
905 IFDB=1IFPL=1GOSUB981

910 IFDB=1THEN950ELSE960
950 BE=0;POKE15710,32;CH=CH+1;RT=L;PRINT@278," ";:POKE1
5706,32;PRINT@406," ";:IFL=1ORL=3ORL=5THENGOSUB1040ELSE
IFL<0GOSUB1050
955 GOSUB1510;RD=0
960 GOSUB1500:A4=A:IFPL=1THENPL=2:FORA1=1TOD2000:NEXT:TB=TB-1;GOT
0299ELSEIFPL=2FORA2=1TOD7;FORA1=1TOD125:NEXT:PRINT@D(A4),MID$(P$(A
4),2,1);:PRINT@D(A4)+64,RIGHT$(P$(A4),1);:FORA1=1TOD125:NEXT:PRIN
T@D(A4)," ";:PRINT@D(A4)+64," ";:NEXT:P$(A4)=""
962 A3=0:FORA=1TODB:IFF$(A)=""THENNEXT:PB=PB-1:PL=1;GOTO300ELSEA
3=A3+1:P1$(A3)=P$(A):NEXT:P$(PB)=""

965 FORA=1TODB:P$(A)=P1$(A):NEXT:FORA=1TODB:PRINT@D(A),MID$(P$(A
),2,1);:PRINT@D(A)+64,RIGHT$(P$(A),1);:NEXT:PRINT@D(A-1)," "
;:PRINT@D(A-1)+64," ";:PL=1:PB=PB-1;GOTO300
980 A$=MID$(T$(W),2,1);B$=RIGHT$(T$(W),1);PRINT@586,"I'LL PLAY T
HE "A$ / "B$";:PRINT@572,A$+"B$ ";
981 T$(W)=""FORA=1TODB:IFT$(A)=""THENFORA3=ATODB:T$(A3)=T$(A3+1
);NEXT:T$(A3+1)=""ELSENEXT
982 IFDB=0IFA$=B$THENC(L)=C(L)-2ELSEC(L)=C(L)-1:C(S1)=C(S1)-1
984 RETURN
```

Lines 1000-1070: Bone drawing routines. Lines 1000-1030 are for the left square on the screen, lines 1040-1070 are for the right. The first line of each routine is called via a GOSUB if the number of dots to be displayed (L) is odd, other-wise the second line of the routine is called.

```
1000 POKE15694,G:IFL=1RETURN
1010 POKE15626,G:POKE15762,G:IFL<4RETURN
1020 POKE15634,G:POKE15754,G:IFL<6RETURN
1030 POKE15630,G:POKE15758,G:POKE15694,32:RETURN
1040 POKE15706,G:IFL=1RETURN
1050 POKE15638,G:POKE15774,G:IFL<4RETURN
1060 POKE15646,G:POKE15766,G:IFL<6RETURN
1070 POKE15642,G:POKE15770,G:POKE15706,32:RETURN
```

Lines 1100-1110: Update data on the video screen.

```

1100 PRINT@21,TT;:PRINT@31,TP;:PRINT@251,TB;:PRINT@315,PB;:PRINT
@764,CH;:IFD>1PRINT@379,BN;
1103 IFD=2IFBK<3POKE15738,42:POKE15742,42
1105 IF(DB=1)+(GS=1)PRINT@764,ZZ;
1110 RETURN

```

Lines 1200-1295: End of hand routines. Handles end-of-hand scoring.

```

1200 ST=0:SP=0:FORA=0TOTB:ST=ST+VAL(MID$(T$(A),2,1))+VAL(RIGHT$(
T$(A),1)):NEXT:FORA=1TOPB:SP=SP+VAL(MID$(P$(A),2,1))+VAL(RIGHT$(
P$(A),1)):NEXT
1205 PRINT@80,X1$;:PRINT@91,X1$;:POKE15738,32:POKE15742,32:PRINT
@586,"** HANDPLAY OVER **";:PRINT@704,STRING$(40,32);:PRINT@320,
" ";:PRINT@384," ";:FORA=1TO2500:NEXT

```

```

1207 IFD=3IFTB=0THENPRINT@586,"I GET 1 POINT FOR GOING OUT";:TT=
TT+1ELSEIFPB=0THENPRINT@586,"YOU GET 1 POINT FOR GOING OUT";:TP=
TP+1
1210 IFD=3THEN1290ELSEIFTB=0THENTT=TT+SP:GO=1ELSEIFPB=0THENTP=TP
+ST:GO=2ELSEIFST>SPTHENTP=TP+(ST-SP):GO=3ELSEIFSP>STTHENTT=TT+(S
P-ST):GO=4ELSEPRINT@580,"DRAWN HAND -- WE BOTH HAVE "SP"PIPS";:GO
TO1290

```

```

1220 GOSUB1100:PRINT@580,"";:ONGOGOTO1230,1240,1250,1260
1230 PRINT"I WIN THE HAND - HA HA ";:GOTO1290
1240 PRINT"I GUESS YOU WIN THE HAND ";:PRINT@644,"I HAVE"ST"PIP
S";:GOTO1290
1250 PRINT"YOU HAVE"SP"PIPS AND I HAVE"ST"PIPS";:PRINT@644,"YOUR
COUNT IS LOWER THAN MINE";:PRINT@708,"YOU WIN THE HAND";:GOTO12
90

```

```

1260 PRINT"I HAVE"ST"PIPS AND YOU HAVE"SP"PIPS";:PRINT@644,"MY C
OUNT IS LOWER THAN YOURS";:PRINT@708,"I WIN THE HAND";
1290 GOSUB1100:FORA=1TO999:NEXT:GOSUB1300:PRINT@320,"TO DEAL";:P
RINT@384,"PRESS";:PRINT@448,"ENTER";:PRINT@636," ";:PRINT@572
," ";:PRINT@266," ";:PRINT@330," ";:PRINT@3
94," ";:IFD=2GS=1
1291 PRINT@278," ";:PRINT@342," ";:PRINT@406,"
";

```

```

1292 A$=INKEY$:IFA$=""THEN1292ELSEIFASC(A$)=13THENST=0:FORA=896T
O936:PRINT@A," ";:PRINT@A+64," ";:NEXT:SP=0:Q7=0:Q6=0:PRINT@580,
STRING$(37,32);:PRINT@644,STRING$(35,32);:PRINT@708,STRING$(35,3
2);:PRINT@320," ";:PRINT@448," ";:PRINT@384," ";

```

```

1295 IFASC(A$)<13THEN1292ELSEPRINT@590,"* SHUFFLING *";:FORA=1T
O20:P$(A)=""T$(A)=""NEX:FORA=0TO6:C(A)=0:NEXT:POKE15738,32:PO
KE15742,32:GOTO15

```

Lines 1300-1350: Check for end of game.

```

1300 IFTT>=GTTTHENH$="I";GOTO1350ELSEIFTP>=GTTTHENH$="YOU";GOTO135
0
1310 RETURN
1350 FORA=1TO7:FORB=1TO300:NEXT:PRINT@708," ";:IF
ORB=1TO300:NEXT:PRINT@708,H$ " WIN THE MATCH";:NEXT:PRINT@256,"TO
PLAY";:PRINT@320,"AGAIN ";:PRINT@384,"PRESS";:PRINT@448,"ENTER"
;:INPUTA$:RUN
1400 FORK=1TO999:NEXT:PRINT@586,"2 POINT STARTER "
;:RETURN

```

Lines 1500-1505: If playing Bergen, this checks to see if the tile placed is a double, if it is set the proper flag (LD or RD).

```

1500 IFD<>3THENRETURNELSEIFA$=B$IFL=VAL(A$)THENLD=1:POKE15690,6
BELSEIFRT=VAL(B$)RD=1:POKE15710,68
1505 RETURN

```

Lines 1510-1550: If playing Bergen, these lines check for double and triple headers.

```

1510 IFD<>3ORDB=1THEN1515ELSEIFRT=LTHEN1520
1515 RETURN
1520 IFPL=1IFLD=1ORRD=1ORA$=B$THENTT=TT+3:H$="TRIPLE";GOTO1550EL
SETT=TT+2:H$="DOUBLE";GOTO1550
1530 IFPL=2IFLD=1ORRD=1ORA$=B$THENTP=TP+3:H$="TRIPLE";GOTO1550EL
SETP=TP+2:H$="DOUBLE";GOTO1550
1550 PRINT@704,STRING$(35,32);:FORA5=1TO5:FORA6=1TO300:NEXT:PRIN
T@586,H$ " HEADER ";:FORA6=1TO300:NEXT:PRINT@586,"
";:NEXT:RETURN

```

Line 2000: Subroutine to draw graphic display.

```

2000 CLS:FORA=0TO768STEP64:POKEA+P+41,191:NEXT:FORA=0TO63:POKEA+
P+832,131:NEXT:FORA=8TO32:POKEA+P+192,176:POKEA+P+448,131:NEXT:F
ORA=256TO384STEP64:POKEA+P+8,191:POKEA+P+20,191:POKEA+P+32,191:N

```

```

EXT:RETURN
3000 C(VAL(MID$(T$(A),2,1)))=C(VAL(MID$(T$(A),2,1)))+1:C(VAL(RIG
HT$(T$(A),1)))=C(VAL(RIGHT$(T$(A),1)))+1:RETURN

```

Line 5000: Program delay subroutine.

```

5000 PRINT@960,"* HIT ENTER * ";:INPUTA$:RETURN

```

Lines 9000-9050: Displays an error message if neither player has a double in either Block or Bergen game.

```

9000 IFD<>2IFTR=0ANDH$=0PRINT@586,"NO DOUBLE IN PLAY - REDEAL";:
FORA=1TO2000:NEXT:GOSUB150:PRINT@896,STRING$(30,32);:PRINT@960,S
TRING$(30,32);:GOTO15
9050 GOTO296

```

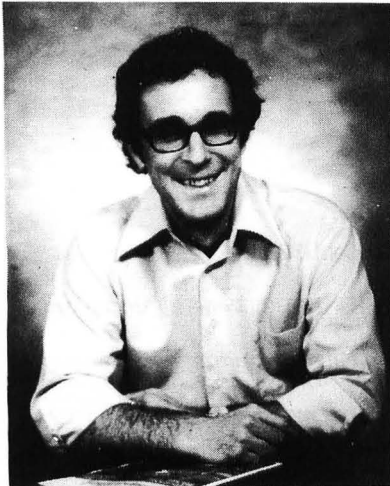
Lines 9100-9500: Subroutine to check for a proper deal.

```

9100 IFD=1IFH>TTHENIFMID$(P$(A),2,1)<>RIGHT$(P$(A),1)GOTO9500
9150 IFD=1IFT>HIFMID$(T$(W),2,1)<>RIGHT$(T$(W),1)THEN9500
9160 IFD=3IFH<TIFMID$(P$(A),2,1)<>RIGHT$(P$(A),1)GOTO9500
9170 IFD=3IFT<HIFMID$(T$(W),2,1)<>RIGHT$(T$(W),1)GOTO9500
9200 RETURN
9500 PRINT@586,"COMPUTER ERROR -- REDEAL";:FORA=1TO2000:NEXT:GOS
UB150:PRINT@896,STRING$(30,32);:PRINT@960,STRING$(30,32);:GOTO15
' POSSIBLE ONLY ON SET PLAY AND IS RARE

```





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.

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)

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	Hockey
Basics	Horserace
Acey Ducey	Hurkle
Amazing	Kinema
Animal	King
Awari	Letter
Bagels	Life
Banner	Life For Two
Basketball	Literature Quiz
Batnum	Love
Battle	Lunar LEM Rocket
Blackjack	Master Mind
Bombardment	Math Dice
Bombs Away	Mugwump
Bounce	Name
Bowling	Nicomachus
Boxing	Nim
Bug	Number
Bullfight	One Check
Bullseye	Orbit
Bunny	Pizza
Buzzword	Poetry
Calendar	Poker
Change	Queen
Checkers	Reverse
Chemist	Rock, Scissors, Paper
Chief	Roulette
Chomp	Russian Roulette
Civil War	Salvo
Combat	Sine Wave
Craps	Slalom
Cube	Slots
Depth Charge	Splat
Diamond	Stars
Dice	Stock Market
Digits	Super Star Trek
Even Wins	Synonym
Flip Flop	Target
Football	3-D Plot
Fur Trader	3-D Tic-Tac-Toe
Golf	Tic Tac toe
Gomoko	Tower
Guess	Train
Gunner	Trap
Hammurabi	23 Matches
Hangman	War
Hello	Weekday
Hexapawn	Word

APPLESOFT CHAINING

by Murray Summers

So you know all there is to know about chaining in Applesoft, right? It's simple. You just 'BLOOD CHAIN, A520' and then 'CALL 520"XXXX" '. Easy, huh? Well, as long as you can tolerate about a five to ten second delay each time a chaining operation is carried out then you really do know all you need to know about that subject. But if you want to make sure that the overall operation of a multi-module program is as fast as possible, or if you just want to learn a little about how Apple works, then read on.

First of all we should define exactly what is done when you chain one program to another. Simply put, chaining two programs gives them common variables. This means that all simple, array, and string variables are passed from one program to the next. This is frequently done when one needs lots of data space and can conveniently break a program into distinct pieces. In order to understand how a chain operates, we must first examine how Apple handles variables.

There are three kinds of variables that can be processed by Apple (not counting the distinction between integer and FP variables): simple (i.e. not subscripted), array (i.e. subscripted), and string. Each of these variables is stored in different ways by Apple, as described and illustrated in the Applesoft manual on pages 127 and 137. It is a slight oversimplification but the general scheme is that simple variables are stored just above the end of the program, followed by arrays, free space, and finally strings. As more and more strings are defined, string storage grows down from the top of memory, and free space contracts commensurately. When strings collide with arrays, Apple "goes away" for what seems an interminable amount of time to "houseclean". If there are no old strings to be deleted, thereby freeing up more free space, then an OUT OF MEMORY error is generated with predictably fatal results. The important thing to

remember is that the order of memory allocation, from bottom to top, is program, then simple variables, arrays, free space, and strings. Whew!

Now, let's suppose you want to load another program which uses the same variables as the previous program. It's fairly simple as long as the second program, and all subsequent chainees, are all smaller than the original, because all you have to do is load the new program and then expand the free space until the descending simples bounce off the top of the program. The more general case would be to chain two programs, regardless of their relative size. For this, we first eliminate the free space altogether, by smushing the simples and arrays up to the bottom of strings. Then load the next program. Of course, if there's not enough space below the simples to load the program you get the familiar "OUT OF...." reminder. Once the new program has been loaded, the free space is expanded as in the first case above.

Having covered chaining we can now get back to the original point — how do we avoid loading the chain routine each time we chain? There are two ways to do this. One of these is the quick and dirty way and the other is much more elegant. I will deal with the quick and dirty first.

There is one prerequisite for quick and dirty chaining, and that is that the first module must be the largest of all of the modules. This is not too much of a restriction, since the first command of the first module could reset LOMEM so that it is effectively the largest by occupying the most memory (e.g. a program that ends at decimal location 8000 could be made effectively 8K larger by resetting LOMEM to 16000 with the LOMEM:16000 command). At any rate, all you need to do to chain this program to any number of smaller programs is to PEEK out the pointers that define the variable space (\$69 to \$70) (105 to 112, decimal), store them someplace safe, load the next program, restore the above pointers to their previous values, and voila! You have just chained

the two programs. The hard part is coming up with someplace safe in memory to store the pointers. There are two places that serve very nicely for this purpose. The first of these is at the very end of the keyboard buffer area (\$2D0 to \$2FF) (720 to 767, decimal). The second place is in the peripheral slot scratchpad area, as described on page 83 in the **Apple II Reference Manual**, better known as the "New Red Book". I will use the keyboard buffer area in an example:

PROGRAM #1

```
10 DIM A$(100),A(25,25),
   B%(20)
20 DEF FN(X)=C↑X + 256 :
   C = 27
30 REM ALL OF THE ABOVE
   STUFF IS NONSENSE
   WHICH WAS JUST STUCK
   IN TO SEEM IMPORTANT
40 REM HERE COMES THE
   GOOD STUFF
50 POKE 720, PEEK (105) :
   POKE 721, PEEK (106) :
   POKE 722, PEEK (107) :
   POKE 723, PEEK (108) :
   POKE 724, PEEK (109) :
   POKE 725, PEEK (110) :
   POKE 726, PEEK (111) :
   POKE 727, PEEK (112) :
60 REM WE HAVE JUST
   STORED AWAY THE
   NECESSARY POINTERS
70 REM AND NOW FOR THE
   KICKER
80 PRINT CHR$(4)"RUN
   PROGRAM #2"
```

PROGRAM #2

```
10 POKE 105, PEEK (720) :
   POKE 106, PEEK (721) :
   POKE 107, PEEK (722) :
   POKE 108, PEEK (723) :
   POKE 109, PEEK (724) :
   POKE 110, PEEK (725) :
   POKE 111, PEEK (726) :
   POKE 112, PEEK (727) :
15 REM NOW PUT IT ALL
   BACK
20 REM THAT'S ALL,
   FFFFOLKS!
30 REM WE HAVE JUST
   DONE THE QUICK
   DIRTY CHAIN.
40 PRINT C : REM JUST TO
   VERIFY THAT THE
   VARIABLES GOT PASSED
```


That's about all there is to a quick and dirty chain. There is one very important detail! Any strings that are defined within quotes in the program, (like A\$ = "ABCD") will not be passed correctly using this technique. To correct this problem, every string so defined must be processed as follows:

```
A$ = MID$(A$,1)
```

Now all of the variables will be passed correctly, and we have avoided accessing the disk altogether. Note that the above fix does not need to be done with strings which were defined using the "CHR\$()" command.

You might be tempted to ask what are the drawbacks of using the quick and dirty chain technique, and you should well be concerned, because it was too easy. There is one important disadvantage, and that is that your free space is only as big as your largest program allows it to be. If the free space is very small, and the programs involve lots of string manipulations, you might think that your Apple had gone to Bermuda because it will spend a great deal of time just doing housecleaning, since the string storage space can't grow too much before it crashes into the array storage, even though there is lots of empty room above your program that is not being used because you had to reset LOMEM! If speed of execution is your major concern, then you may be dissatisfied with the quick and dirty chain.

All of these problems are solved by the next technique I shall describe, which I feel is a more elegant approach to chaining. It, however, is not used without paying the price of about 1.3K worth of overhead. The concept of its use is pretty simple.

A modified chain routine is loaded just below DOS and HIMEM is reset under it. In this location the chain routine is completely safe, and need not be reloaded unless the disk is rebooted, or unless the system crashes. The modification is a short machine code prefix which calls the monitor move subroutine so that it is relocated to its normal running location and then jumps to the relocated address for a conventional chain. I think that

this is an elegant way to do the chain. It requires no longer to run than the program being chained requires to load, and is virtually foolproof. All that is necessary to do is to implement the following code by entering lines 10000 through 10090 into your program, and construct a modified CHAIN routine.

PROGRAM #3

```
10000 H1 = PEEK (115) + 256
      * PEEK (116)
10010 H2 = H1 - 464
10020 PRINT CHR$(4)
      "BLOAD CHAIN1, A" H2
10030 LN = H2 - INT(H2/256)
      * 256 : HN = INT
      (H2/256)
10040 LO = H1 - INT (H1/256)
      * 256 : HO = INT
      (H1/256)
10050 POKE 115, LN : POKE
      116, HN : POKE III, LN :
      POKE 112, HN
10055 RETURN
10060 POKE 60, LN + 8 :
      POKE 61, HN
10070 POKE 62, LO : POKE
      63, HO
10080 POKE 66, 8 : POKE 67, 2
10090 RETURN
```

To make the modified CHAIN routine, insert the DOS MASTER and type the following commands:

```
BLOAD CHAIN, A2008
POKE 2000, 160
POKE 2001, 0
POKE 2002, 32
POKE 2003, 44
POKE 2004, 254
POKE 2005, 76
POKE 2006, 8
POKE 2007, 2
```

Then insert an initialized disk and type "BSAVE CHAIN1, A2000, L464".

Now, let's examine how all this mystery works! The first two pokes load the 6502's "Y" register with a zero (because, just because). The next three pokes are "JSR FE2C" which causes a jump to a subroutine in the monitor which moves memory located between the address contained in location 60 and 61 (high byte last, of course) and 62 and 63, to the address beginning at the location contained in 66 and 67. The final three pokes

are "JMp 208" which causes execution to begin with the routine at \$208 (our old friend decimal location 520!). In other words, we just use the monitor to relocate the chain routine from someplace up high in memory down to where it needs to be run. But what happens if we need to store strings or something up high in memory? Does the chain routine get clobbered? Well, it would unless we did something to protect it, and that is what the pokes in program #3 are doing, among other things. Line 10000 reconstructs HIMEM and the next line calculates where HIMEM should be reset to if the chain routine is loaded right at the top of memory (just under DOS, of course). Then line 10020 loads the modified chain routine right where we calculated (H2). Lines 10030 and 10040 are to break the old HIMEM (H1) and the new HIMEM (H2) into their low and high bytes, respectively. These values are then poked into the correct locations for the monitor memory move routine to find them. The reason that line 10060 adds 8 to LN is so that the memory move routine will skip over the first 8 bytes of the chain routine (i.e. the eight pokes done to modify the chain routine to begin with). These eight bytes don't need to be moved since they are not part of the chaining algorithm (I think we are about to get there!).

Now all we need to do to chain is to include lines 10000 to 10090 in any program (the line #'s can be changed as necessary), and initialize things at the beginning of the first program with a:

```
'GOSUB 10000'
```

When you are ready to chain all that is necessary is:

```
'GOSUB 10060 : CALL H2' 'XXX'
```

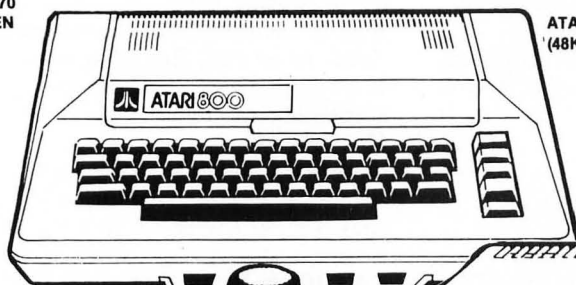
All subsequent programs need have only lines 10060 to 10090 to correctly chain to other modules. Note the syntax of the CALL in the above lines. There is to be no space between the 'H2' and the first quotation mark.

It will be up to you to decide which routine better suits your purposes. Do you require speed or elegance? Personally, I'll take the slow Mercedes every time.



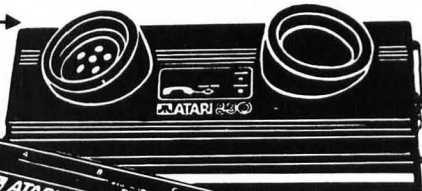
The Atari* Connection.

ATARI CX70
LIGHT PEN
\$74.95



ATARI 800
(48K) \$1019.00

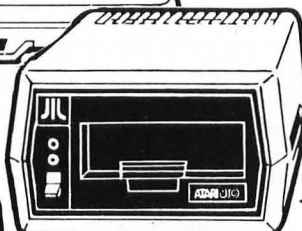
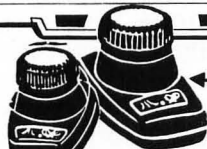
ATARI 850
INTERFACE
\$179.00



ATARI 830
MODEM
\$179.00

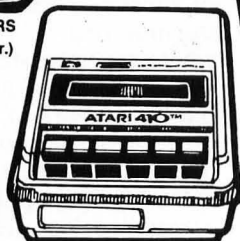
ADDITIONAL
RS232 PORTS

PADDLE
CONTROLLERS
\$19.95 (pr.)

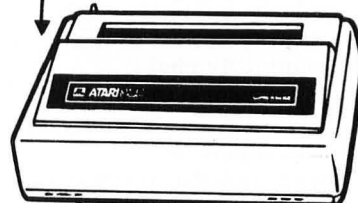


ATARI 810 DISK DRIVE \$499.00

JOYSTICK
CONTROLLERS
\$19.95 (pr.)



ATARI 410
RECORDER \$69.00



ATARI 825 PRINTER \$769.00

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

Hardware

ATARI 400 Computer System, 8K RAM . . .	\$439.00 (#36-400)
ATARI 400 Computer System, 16K RAM . .	\$499.00 (#36-401)
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 . .	\$1019.00 (#36-802)
ATARI 410 Program Recorder	\$69.00 (#36-803)
ATARI 810 Disk Drive	\$499.00 (#36-810)
ATARI 820 Printer (40-col)	\$369.00 (#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	\$119.50 (#36-854)
32K RAM Module for the ATARI	\$189.00 (#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	\$39.95 (#36-BASK)
Chess	\$39.95 (#36-CHS)
Editor/Assembler	\$59.95 (#36-ASE)
Educational System Master	\$29.95 (#36-ESMC)
Music Composer	\$59.95 (#36-MUSE)
Star Raiders	\$59.95 (#36-STRDS)
Super Breakout	\$39.95 (#36-SUPB)
Tele-Link	\$24.95 (#36-TEL)
3D Tic-Tac-Toe	\$39.95 (#36-3TTT)
Video Easel	\$39.95 (#36-VIDEO)

Software on Disk

VISICALC from Personal Software \$199.95 (#36-VICL)

Software on Cassette

Basic Algebra*	\$29.95 (#36-ALGE)
Basic Electricity*	\$29.95 (#36-ELEC)
Basic Psychology*	\$29.95 (#36-PSYCH)
Basic Sociology*	\$29.95 (#36-SOC)
Biorhythms	\$14.95 (#36-BIOR)
Blackjack	\$14.95 (#36-BLACK)
Business Communications*	\$29.95 (#36-BUS)
Effective Writing*	\$29.95 (#36-WRIT)
Great Classics*	\$29.95 (#36-CLAS)
Hangman	\$14.95 (#36-HANG)
Intro to Programming in BASIC	\$19.95 (#36-IPB)
Kingdom	\$14.95 (#36-KING)
Physics*	\$29.95 (#36-PHYS)
Principles of Accounting*	\$29.95 (#36-ACCT)
Principles of Economics*	\$29.95 (#36-ECON)
Spelling*	\$29.95 (#36-SPELL)
Supervisory Skills*	\$29.95 (#36-SUPSK)
3-Dimensional Graphics Package	\$29.95 (#36-3D-G)
U.S. Government*	\$29.95 (#36-USG)
U.S. History*	\$29.95 (#36-USH)
World History*	\$29.95 (#36-WRLD)

*Requires Educational System Master Cartridge

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.



A Warner Communications Company

Enter our world of microcomputing...

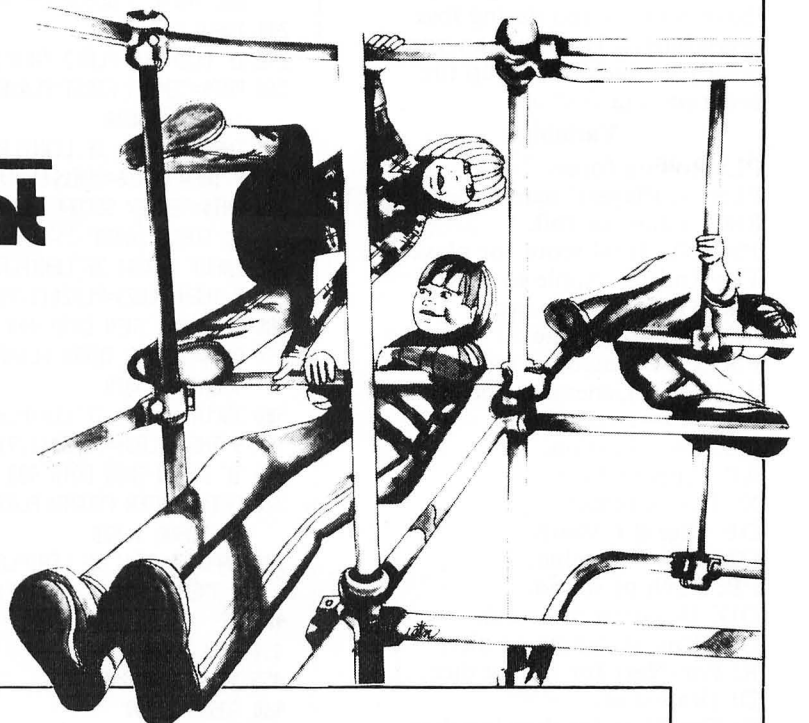
Discover our exciting world of microcomputing by reading SoftSide Magazine, the monthly that helps you learn to use your S-80, Apple or Atari more effectively and enjoyably.

In each issue you'll find helpful programming hints, follow regular columns by such well-known authors as Scott Adams, and explore the wide range of software and hardware for your particular needs.

The focus of SoftSide is on lots of entertaining software — great games, simulations, graphics and educational programs all ready to type right into your computer, each one fully documented and carefully explained. Each month you'll find at least nine programs — one reader told us that he estimates the value of the software in a single issue of SoftSide at \$18 — the price of a 12-issue subscription! Order your subscription to SoftSide today and become a part of our world of microcomputing!

SoftSide™
our viewpoint
is strictly
BASIC.

CALL OUR TOLL FREE HOTLINE
1-800-258-1790
(IN NH CALL 673-5144)



Here's my ticket to SoftSide's world of microcomputing.
12 issues for only \$18 (a savings of 40% over cover price)

Check enclosed Bill Me Bill my Visa MasterCard

Card # _____ Expiration date _____

I own a(n) _____ computer

Name _____

Address _____

City _____ State _____ Zip _____

SoftSide Publications 6 South Street Milford, NH 03055

SS41

SKUNK

by John Daoust

"Skunk" is an Apple program written in Integer BASIC which requires at least 16K of RAM.

"Skunk" is based on a dice game that's been around for many years. The game is played by two to four players, one of whom (which?) can be the Apple itself. The object is simple: Reach a predetermined point total before the others do. The problem is that if you roll a pair of ones (skunks) you must start all over again. A single skunk merely wipes out your score for a single round, rather than for the entire game.

Only the most daring win, but those who are too daring lose everything... How far can you push your luck? Load up the program and find out.

Variables:

PL: Rolling force.
 PLRn\$: Players' names.
 RSC: Total for roll.
 TSC(X): Total score for player.
 TR: Rnd for Apple to quit.
 MK: High score to beat.
 SC: Total for game.
 PNT\$: Statement to type.
 A\$, T, O: General purpose.
 PLR: Player number.
 DL: Dice location.
 AP: Apple's turn.
 X: Turn counter.
 D1: Dice # 1 value.
 D2: Dice # 2 value.
 PP: Pitch of sound.
 DD: Duration of sound.
 UU: Rnd sketching of dice.
 R: For/Next for rolling dice.
 D: Draw dice.
 K: For/Next for drawing dice.

```
20 REM 'SKUNK' BY JOHN C. DAoust
21 REM 17-OCT-79
```

Initialization and set-up.

```
35 DIM PLR1$(20),PLR2$(20),PLR3$(20),PLR4$(20)
40 DIM A$(5),TSC(4),PNT$(100)
45 GOSUB 22100
50 TEXT : CALL -936
70 FOR T=1 TO 4:TSC(T)=0: NEXT T
100 VTAB 10: TAB 17:PNT$="SKUNK": GOSUB 16020
105 TAB 6:PNT$="PROGRAMMED BY JOHN C. DAoust": GOSUB 16020
```



```
150 FOR A=1 TO 600: NEXT A
200 PNT$="WHAT SCORE DO YOU WISH FOR GAME ": GOSUB 16020
210 INPUT SC
230 PNT$="HOW MANY PLAYERS (2 TO 4) ARE THERE": GOSUB 16020
240 INPUT PLR
245 IF PLR>4 OR PLR<2 THEN 230
260 PNT$="ENTER FIRST PLAYERS NAME ": GOSUB 16020
265 INPUT PLR1$: IF LEN(PLR1$)>9 THEN PLR1$=PLR1$(1,9)
270 PNT$="ENTER SECOND PLAYERS NAME ": GOSUB 16020
275 INPUT PLR2$: IF LEN(PLR2$)>9 THEN PLR2$=PLR2$(1,9)
280 IF PLR=2 THEN GOTO 400
290 PNT$="ENTER THIRD PLAYERS NAME ": GOSUB 16020
300 INPUT PLR3$: IF LEN(PLR3$)>9 THEN PLR3$=PLR3$(1,9)
310 IF PLR=3 THEN GOTO 400
320 PNT$="ENTER FOURTH PLAYERS NAME ": GOSUB 16020
330 INPUT PLR4$: IF LEN(PLR4$)>9 THEN PLR4$=PLR4$(1,9)
400 GR
450 CALL -936
455 X=1:RSC=0:G=1
460 GOSUB 21000
```

Main control loop.

```
500 AP=0:G=1
550 IF X=1 THEN GOSUB 10000
560 IF X=2 THEN GOSUB 10100
570 IF X=3 THEN GOSUB 10200
580 IF X=4 THEN GOSUB 10300
620 GOSUB 13000
700 IF AP=1 THEN GOSUB 11000
720 IF AP=1 THEN GOTO 835
800 PL= PDL (0)/25
810 POKE 34,22: VTAB 23
815 TAB 20: PRINT "ROLLING FORCE=" ;PL;" "
830 IF PEEK (-16287)<=127 THEN GOTO 800
```

```
835 IF PL=0 THEN 1100
850 GOSUB 12000
860 RSC=RSC+D1+D2
880 IF D1=1 OR D2=1 THEN RSC=0
890 IF D1=1 AND D2=1 THEN RSC=TSC(X)
892 PF=50:DD=127
900 IF D1=1 OR D2=1 THEN GOSUB 22000
905 PP=30:DD=255
910 IF D1=1 AND D2=1 THEN GOSUB 22000
915 IF D1=1 AND D2=1 AND TSC(X)=0 THEN GOTO 1300
920 IF D1=1 AND D2=1 THEN G=-1
930 IF D1=1 AND D2=1 THEN GOTO 1100
958 POKE 34,23
959 VTAB 24: PRINT
960 VTAB 24: PRINT "TOTAL FOR ROLL=" ;RSC;" SCORE WILL BE ";TSC(X)+RSC;
970 IF D1=1 OR D2=1 THEN GOTO 1300
980 IF D1<>1 AND D2<>1 THEN GOTO 1480
990 GOTO 1300
1100 POKE 34,21
1101 PF=15:DD=5
1102 FOR T=1 TO RSC
1103 IF RSC=0 THEN GOTO 1480
1105 TSC(X)=TSC(X)+G
1107 FOR O=1 TO 15: NEXT O
1108 VTAB 22: TAB 4+(10*(X-1)): PRINT " "
1109 GOSUB 22000
1110 FOR O=1 TO 15: NEXT O
1113 VTAB 22: TAB 4+(10*(X-1)): PRINT TSC(X)
1115 NEXT T
1120 G=1
1300 X=X+1
1310 IF X>PLR THEN X=1
1315 FOR T=1 TO 1200: NEXT T
1320 IF TSC(X)>=SC THEN GOSUB 14000
1321 POKE 50,63
1322 POKE 34,23: VTAB 24
1323 PRINT " NEXT PLAYERS TURN ";
1324 POKE 50,255
1330 RSC=0
1480 FOR T=1 TO 500: NEXT T
1500 GOTO 500
2000 END
```

Check to see if the Apple is playing.

```

10000 IF "APPLE"=PLR1$ THEN AP=1
10030 RETURN
10100 IF "APPLE"=PLR2$ THEN AP=1
10130 RETURN
10200 IF "APPLE"=PLR3$ THEN AP=1
10230 RETURN
10300 IF "APPLE"=PLR4$ THEN AP=1
10330 RETURN

```

Apple's rolling logic.

```

11000 PL= RND (9)+1
11004 MK=0
11010 TR= RND (4)
11015 IF RSC=0 THEN GOTO 11065
11020 FOR T=1 TO 4
11025 IF TSC(T)>=SC THEN GOTO 11032
11030 NEXT T
11031 GOTO 11050
11032 IF MK<=TSC(T) THEN MK=TSC(T)
11035 GOTO 11030
11050 IF TSC(X)+RSC<=MK AND MK<>0
    THEN GOTO 11065
11055 IF MK<>0 AND TSC(X)+RSC>MK THEN
    PL=0
11057 IF MK=0 AND TSC(X)+RSC>=SC THEN
    PL=0
11060 IF TR=0 THEN PL=0
11065 POKE 34,22: VTAB 23
11067 TAB 20: PRINT "ROLLING FORCE= "
    ;PL;" "
11070 RETURN

```

Roll dice.

```

12000 FOR R=1 TO PL
12050 UU= RND (2)
12060 D1= RND (6)+1:D2= RND (6)+1
12070 FOR K=1 TO 2
12071 PF=1:DD=3: GOSUB 22000
12075 IF UU=0 THEN 12085
12077 D=D2:DL=22: GOSUB 12220
12080 GOTO 12100
12085 D=D1:DL=4: GOSUB 12200
12100 IF D=1 THEN GOSUB 12250
12110 IF D=2 THEN GOSUB 12300
12120 IF D=3 THEN GOSUB 12350
12130 IF D=4 THEN GOSUB 12400
12140 IF D=5 THEN GOSUB 12450
12150 IF D=6 THEN GOSUB 12500
12155 GOTO 12185
12170 NEXT K
12175 NEXT R
12180 RETURN
12185 IF UU=1 THEN GOTO 12192
12187 IF UU=0 THEN UU=1
12189 GOTO 12170
12192 UU=0: GOTO 12170

```

Draw dice.

```

12200 COLOR=15
12205 FOR Y=10 TO 30: HLIN 4,17 AT
    Y: NEXT Y
12210 RETURN

```

```

12220 COLOR=15
12225 FOR Y=10 TO 30: HLIN 22,35 AT
    Y: NEXT Y
12230 RETURN

```

Draw skunk.

```

12250 COLOR=0
12255 PLOT DL+3,15: VLIN 19,20 AT
    DL+3: PLOT DL+3,26: VLIN 15
    ,16 AT DL+4: PLOT DL+4,19: VLIN
    24,26 AT DL+4

```

```

12260 VLIN 13,26 AT DL+5: VLIN 14
    ,26 AT DL+6: VLIN 23,25 AT
    DL+7

```

```

12265 VLIN 16,24 AT DL+8: PLOT DL+
    9,15: PLOT DL+9,23: PLOT DL+
    10,15

```

```

12267 PLOT DL+4,23

```

```

12270 VLIN 17,22 AT DL+10: PLOT DL+
    11,16

```

```

12275 COLOR=2: PLOT DL+4,14

```

```

12280 RETURN

```

Draw two spot.

```

12300 COLOR=0

```

```

12315 VLIN 13,15 AT DL+2: VLIN 25
    ,27 AT DL+10

```

```

12320 VLIN 13,15 AT DL+3: VLIN 25
    ,27 AT DL+11

```

```

12330 RETURN

```

Draw three spot.

```

12350 COLOR=0

```

```

12360 VLIN 13,15 AT DL+10: VLIN 19
    ,21 AT DL+6: VLIN 25,27 AT
    DL+2

```

```

12365 VLIN 13,15 AT DL+11: VLIN 19
    ,21 AT DL+7: VLIN 25,27 AT
    DL+3

```

```

12380 RETURN

```

Draw four spot.

```

12400 GOSUB 12300: GOSUB 12350

```

```

12410 COLOR=15

```

```

12420 VLIN 19,21 AT DL+6: VLIN 19
    ,21 AT DL+7

```

```

12430 RETURN

```

Draw five spot.

```

12450 GOSUB 12300: GOSUB 12350

```

```

12460 RETURN

```

Draw six spot.

```

12500 GOSUB 12400

```

```

12510 COLOR=0

```

```

12520 VLIN 19,21 AT DL+2: VLIN 19
    ,21 AT DL+10

```

```

12530 VLIN 19,21 AT DL+3: VLIN 19
    ,21 AT DL+11

```

```

12550 RETURN

```

Draw player marker.

```

13000 COLOR=4

```

```

13020 HLIN 0,39 AT 39

```

```

13030 COLOR=13

```

```

13050 IF X=1 THEN HLIN 3,5 AT 39

```

```

13060 IF X=2 THEN HLIN 13,15 AT 39

```

```

13070 IF X=3 THEN HLIN 23,25 AT 39

```

```

13080 IF X=4 THEN HLIN 33,35 AT 39

```

```

13100 COLOR=4

```

```

13300 RETURN

```

Announce winner.

```

14000 TEXT : CALL -936

```

```

14010 MJ=0

```

```

14020 FOR T=1 TO 4

```

```

14030 IF MJ<TSC(T) THEN GOTO 14200

```

```

14050 NEXT T

```

```

14100 PNT$="THE RESULTS OF THIS GAME O
    F SKUNK ARE:" : GOSUB 16020

```

```

14110 PRINT : PRINT

```

```

14120 PRINT PLR1$;" = ";TSC(1)

```

```

14125 PRINT : PRINT PLR2$;" = "

```

```

;TSC(2)

```

```

14127 IF PLR=2 THEN GOTO 14150

```

```

14130 PRINT : PRINT PLR3$;" = "

```

```

;TSC(3)

```

```

14135 IF PLR=3 THEN GOTO 14150

```

```

14140 PRINT : PRINT PLR4$;" = "

```

```

;TSC(4)

```

```

14150 PRINT : PRINT

```

```

14160 PRINT "THE WINNER HAS THE HIGH S
    CORE OF ";TSC(MN)

```

```

14165 IF MN=1 AND PLR1$="APPLE" THEN
    GOTO 14180

```

```

14170 IF MN=2 AND PLR2$="APPLE" THEN
    GOTO 14180

```

```

14175 IF MN=3 AND PLR3$="APPLE" THEN
    GOTO 14180

```

```

14177 IF MN=4 AND PLR4$="APPLE" THEN
    GOTO 14180

```

```

14179 GOTO 14184

```

```

14180 F= RND (4): IF F=0 THEN GOSUB
    17000

```

```

14181 IF F=1 THEN GOSUB 17100

```

```

14182 IF F=2 THEN GOSUB 17200

```

```

14183 IF F=3 THEN GOSUB 17300

```

```

14184 PRINT :PNT$="WOULD YOU CARE TO P
    LAY AGAIN": GOSUB 16020: INPUT
    A$

```

```

14186 IF "Y"=A$(1,1) THEN GOTO 50

```

```

14190 END

```

```

14200 MJ=TSC(T):MN=T: GOTO 14050

```

Typing routine.

```

16020 PF=3:DD=3

```

```

16050 FOR T=1 TO LEN(PNT$)

```

```

16060 IF T= LEN(PNT$) THEN GOTO 16080

```

```

16070 PRINT PNT$(T,T);

```

```

16075 GOTO 16090

```

```

16080 PRINT PNT$(T,T)

```

```

16090 GOSUB 22000

```

continued on page 76

STRATEGY STRIKE



by David Steenson
Atari translation by Rich Bouchard

“Strategy Strike” is an Atari program requiring at least 16K of RAM.

Atari “Strategy Strike” is a translation of the Apple “Strategy Strike” published last month. It is a game of logic and memory, where players try to manipulate their forces to defeat their opponent.

INSTRUCTIONS

The object of the game is to move militarily ranked pieces across a battle board while trying to capture the opposing player’s flag. Each player starts with forty pieces, consisting of:

# OF PIECES	NAME	RANK
1	Marshal	1
1	General	2
2	Colonel	3
3	Major	4
4	Captain	5
4	Lieutenant	6
4	Sergeant	7
5	Miner	8
8	Scout	9
1	Spy	10
6	Bomb	UNMOVEABLE
1	Flag	UNMOVEABLE

The above pieces are in order of military rank, the Marshal being the highest ranked, and the Spy being the lowest. However, the spy may remove the Marshal from the board only if it is attacking the Marshal. The Bombs are immobile pieces that will remove any piece attempting to “strike” it, except for the Miner, which can dismantle and remove the Bomb.

The game is played on a ten-by-ten board which will be fully displayed throughout the game. Player 1 sets up pieces on the top four rows of the board, and player 2 sets up pieces on the bottom four rows.

SETUP

When the game starts, it will allow each player to set up his or her forces. To place any of the moveable pieces except the Spy on the board, simply type the rank of the desired piece and the square marked by a circle on the screen will then contain that piece. To place a Spy, Bomb or Flag, type S,

B, or F respectively. After the initial setup you will be allowed to make any needed changes. To make a change, type in the coordinates of two different pieces, and the position of those pieces will be switched on the board. To enter the coordinate of a piece, type its column (the letter underneath the square) and its row (the number to the left of the square). As in the Apple version, the Delete key may be used to correct mistakes while entering coordinates, and no Return is needed.

MOVEMENT

1). Pieces may move only one space at a time, and there are no diagonal moves. To move, type the coordinates of the piece to be moved, and the coordinates of the square the piece is to be moved to.

2). Pieces may not move onto the lakes in the center of the board.

3). Two pieces may not occupy the same space at one time, and pieces may not be jumped.

4). The Flag and Bomb pieces may not be moved.

A player’s pieces are revealed only when it is that player’s turn. The opposing player should not be facing the screen when an opponent is moving. Only during the combat phase should both players be looking at the screen.

STRIKE OR ATTACK RULES

To strike an opponent, move one of your pieces into an adjoining square containing an opponent’s piece. Combat will then commence, with the losing piece being removed from the board and replaced by the victor. If both pieces have the same rank, both are removed. If the flag is attacked, the attacker wins the games.

VARIABLES

A: Temporary variable for a piece to be printed.

A1: Piece occupying “from” square.

A2: Piece occupying “to” square.

A(X,P*10+Y): Contains player P’s pieces at coordinates X,Y.

BOMB: Numerical designation for bomb (12).

FLAG: Numerical designation for flag (11).

I: General loop variable.

IN(I): Number of pieces of type I already entered.

NU(I): Quota of pieces of type I.

P: Player number (0 or 1).

PASS: Logical variable. If 1 then skip turn.

R: Argument of GET#1,R used to get a character from the keyboard.

R1, R2: Coordinates of “from” square.

R3, R4: Coordinates of “to” square.

RR: Numerical value of the character contained in R.

S1\$, S2\$: Strings used to draw board.

SPY: Numerical value of Spy (10).

X, Y: Horizontal and vertical coordinates.

Z: Time delay and color loop variable.

DOCUMENTATION

Line 25: Clear the right-hand part of the screen where messages are displayed.

Lines 30-48: Subroutine to input move coordinates.

Lines 100-320: Initialization.

Lines 150-260: Routine to draw pictures of each piece.

These lines included in the listing contain symbols for what should be typed in. The following table explains what should be typed in place of these symbols.

Symbol Meaning

b	Blank
d	ESC followed by CTRL = (looks like down arrow)
1	ESC followed by CTRL + (looks like left arrow)
=	SHIFTed equal sign (appears as vertical line)

Any capital letter should be typed with the CTRL key held down, except on line 240, where the capital letters should be typed as normal letters. In addition, the four underlined characters in line 260 should be typed in reverse video as well. Thank James Garon and his “Checker Challenge” (October 1980 *SoftSide*) for most of these graphic figures.

Lines 400-560: Set up each player’s pieces.

Lines 600-700: Main playing loop. Begins a new turn.

Lines 710-830: If player P has attacked...

Lines 900-940: End of game routine, after flag is captured.

Lines 1000-1060: Subroutine to draw playing board.

Lines 2000-2040: Subroutine to print player P's pieces on board.

Lines 2500-2540: Subroutine to print an individual piece on the board.

Lines 3000-3040: Subroutine to cover player P's pieces on the board with blanks.

Lines 4000-4030: Subroutine to switch pieces during setup.

Lines 5000-5160: Attack subroutine. Uses lines 150-260 to

draw each piece, then carries out the attack.

Lines 6000-6040: Subroutine to display bomb flash.

Line 7000: Dummy Return statement where Apple sound routine was.

Lines 8000-8020: Subroutine to play fanfare.

When typing in Atari "Strategy Strike", the following should be typed in reverse video:

Line 46: "Y/N"

Line 270: "STRATEGY STRIKE", excluding the spaces.

Lines 910, 915, 5000, and 5015: Every string in a PRINT statement.

Line 1010: Both character strings.

Line 4010: "ILLEGAL MOVE!"

```
3 OPEN #1,4,0,"K:"
5 GOTO 100
25 FOR T=0 TO 18:POSITION 26,T:PRINT "
    ";;NEXT T:RETURN
30 POKE 752,0:PASS=0:POSITION 26,2:PRI
    NT "(P TO PASS)";:POSITION 26,3:PRINT
    "FROM: ";:GET #1,R
31 IF CHR$(R)="P" THEN PRINT "PASS";:P
    ASS=1:POKE 752,1:RETURN
32 PRINT CHR$(R);:R1=R-65:IF R1<0 OR R
    1>9 THEN PRINT CHR$(253);:GOTO 30
34 POSITION 33,3:PRINT ", ";:GET #1,R:I
    F R=126 THEN 30
36 PRINT CHR$(R);:R2=R-48:IF R2<0 OR R
    2>9 THEN PRINT CHR$(253);:GOTO 34
38 POSITION 26,4:PRINT "TO: ";:GET #1,
    R:IF R=126 THEN 34
40 PRINT CHR$(R);:R3=R-65:IF R3<0 OR R
    3>9 THEN PRINT CHR$(253);:GOTO 38
42 POSITION 33,4:PRINT ", ";:GET #1,R:I
    F R=126 THEN 38
44 PRINT CHR$(R);:R4=R-48:IF R4<0 OR R
    4>9 THEN PRINT CHR$(253);:GOTO 42
46 POSITION 26,5:PRINT "CORRECT (Y/N)"
    ;;GET #1,R:IF CHR$(R)="" THEN 30
48 PRINT :POKE 752,1:SOUND 0,40,10,10:
    FOR T=1 TO 10:NEXT T:SOUND 0,0,0,0:RET
    URN
100 GRAPHICS 0:DIM A(9,19),NU(12),W(1)
    ,IN(12),S1$(30),S2$(30),R$(1),Q(1)
110 NU(1)=1:NU(2)=1:NU(3)=2:NU(4)=3:NU
    (5)=4:NU(6)=4:NU(7)=4:NU(8)=5:NU(9)=8:
    NU(10)=1:NU(11)=1:NU(12)=6
120 SPY=10:FLAG=11:BOMB=12
140 GOSUB 7000
145 GOTO 270
150 PRINT "bbEd111bb=d111bbX";:RETURN
160 PRINT "qEd111qRcd111zRC";:RETURN
170 PRINT "qEd111bRdDd111zRC";:RETURN
180 PRINT "wbWd111zRSd111bbX";:RETURN
190 PRINT "qEd111zREd111zRC";:RETURN
200 PRINT "qEd111AREd111zRC";:RETURN
```

```
210 PRINT "qEd111bb=d111bbX";:RETURN
220 PRINT "qEd111ARdDd111zRC";:RETURN
230 PRINT "qEd111zRdDd111zRC";:RETURN
240 PRINT "bSbd111SPYd111bSb";:RETURN
250 PRINT "qEd111ARcd111=bb";:RETURN
260 PRINT "bFbd111HbJd111JbH";:RETURN
262 PRINT :PRINT
270 GRAPHICS 0:POSITION 2,6:PRINT "
    S T R A T E G Y   S T R I K E":POSITI
    ON 12,8:PRINT "BY DAVID STEENSON"
272 POSITION 6,10:PRINT "ATARI VERSION
    BY RICH BOUCHARD"
280 GOSUB 8000
290 FOR Y=0 TO 3:FOR X=0 TO 9:A(X,Y)=
    1:NEXT X:NEXT Y
292 FOR Y=4 TO 15:FOR X=0 TO 9:A(X,Y)=
    0:NEXT X:NEXT Y
300 FOR Y=16 TO 19:FOR X=0 TO 9:A(X,Y)
    =1:NEXT X:NEXT Y
310 GRAPHICS 0:POKE 752,1:GOSUB 1000
320 FOR P=0 TO 1:GOSUB 3000:NEXT P
400 FOR P=0 TO 1:GOSUB 25:FOR I=1 TO 1
    2:IN(I)=0:NEXT I
410 FOR Y=P*6 TO P*6+3:FOR X=0 TO 9
420 GOSUB 25:SOUND 0,0,0,0:POSITION 26
    ,2:PRINT "PLAYER "P+1";":POSITION 2
    6,3:PRINT "WHAT PIECE DO";
425 POSITION 26,4:PRINT "YOU WANT IN";
    :POSITION 26,5:PRINT "THIS POSITION";:
    POSITION 26,6:PRINT "(1-9,S,B,F)";
430 POSITION X*2+5,Y*2+2:PRINT CHR$(20
    ):POSITION X*2+5,Y*2+2:GET #1,R:RR=R-4
    8
440 IF CHR$(R)="S" THEN RR=SPY
450 IF CHR$(R)="B" THEN R=42:RR=BOMB
460 IF CHR$(R)="F" THEN RR=FLAG
470 IF RR<1 OR RR>12 THEN 430
480 IF NU(RR)=IN(RR) THEN 430
490 IN(RR)=IN(RR)+1
500 POSITION X*2+5,Y*2+2:PRINT CHR$(R)
    ;
```

continued on next page

AN ART STUDIO IN 48K

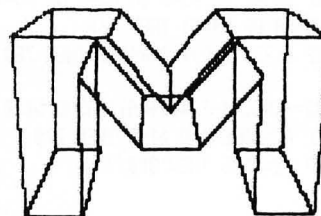
TWO PROGRAMS
BY MARK PELCZARSKI
FROM



Magic Paintbrush

Design shape tables with keystrokes or paddles. Also included are Applesoft Invaders and Slot Machine — colorful variations of well-known diversions whose graphics were created with this package. Save your graphics on disk and put them in your own programs.

32K Applesoft, ROM Diskette \$29.95



Three-D

CREATE
3-DIMENSIONAL GRAPHICS!

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

48K Applesoft ROM Diskette \$29.95



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

continued from previous page

```
510 SOUND 0,RND(0)*100+10,10,10
520 A(X,P*10+Y)=RR
530 NEXT X:NEXT Y:SOUND 0,0,0,0

540 GOSUB 25:POSITION 26,2:PRINT "ANY
CHANGES?";:POSITION 26,3:PRINT "(Y OR
N)";:GET #1,R

545 IF CHR$(R)<>"N" THEN GOSUB 4000:GO
TO 540
550 IF CHR$(R)<>"N" THEN GOSUB 4000:GO
TO 540

560 GOSUB 3000:NEXT P
600 P=0
610 GOSUB 25:POSITION 26,2

620 PRINT "PLAYER ";2-P; ", YOU";:POSIT
ION 26,3:PRINT "MUST GO AWAY";

630 POSITION 26,5:PRINT "PLAYER ";P+1;
 ", HIT";:POSITION 26,6:PRINT "RETURN T
O GO.";:GET #1,R:IF R<155 THEN 630

640 GOSUB 25:POSITION 26,0:PRINT "PLAY
ER ";P+1;:POSITION 26,1:PRINT "-----
-";
650 GOSUB 2000

660 GOSUB 30:IF PASS=1 THEN 820
670 A1=A(R1,P*10+R2):A2=A(R3,P*10+R4)

680 IF A1=0 OR A2<0 OR A1=BOMB OR A1=
FLAG OR ABS(R1-R3)+ABS(R2-R4)>1 THEN 6
60

690 IF (R3=2 OR R3=3 OR R3=6 OR R3=7)
AND (R4=4 OR R4=5) THEN 660
700 IF A(R3,(1-P)*10+R4)=0 THEN 780

710 A2=A(R3,(1-P)*10+R4):GOSUB 5000
730 IF A2=BOMB AND A1=8 THEN 770
740 IF A2=BOMB THEN 800

750 IF A1=A2 THEN POSITION R3*2+5,R4*2
+2:PRINT CHR$(160);:A(R3,(1-P)*10+R4)=
0:GOTO 800
760 IF NOT (A2<A1 AND NOT (A2=1 AND
A1=SPY)) THEN 770

762 A(R1,(1-P)*10+R2)=A2:A(R3,(1-P)*10
+R4)=0:POSITION R3*2+5,R4*2+2:PRINT CH
R$(160);:POSITION R1*2+5,R2*2+2

764 A=A2:GOSUB 2500:A(R1,P*10+R2)=0:GO
TO 820
770 A(R3,(1-P)*10+R4)=0

780 POSITION R3*2+5,R4*2+2:A=A(R1,P*10
+R2):GOSUB 2500
790 A(R3,P*10+R4)=A(R1,10*P+R2)
800 A(R1,P*10+R2)=0

810 POSITION R1*2+5,R2*2+2:PRINT CHR$(
160);
820 FOR Z=1 TO 1000:NEXT Z:GOSUB 3000
```

```
830 P=1-P:GOTO 610
900 GOSUB 25

910 POSITION 26,4:PRINT " PLAYER ";CH
R$(P+177);" ";:POSITION 26,5:PRINT "
 ";
915 POSITION 26,6:PRINT " HAS WON! "
 ;

920 GOSUB 8000:GOSUB 6000:GOSUB 8000:G
OSUB 6000
930 FOR P=0 TO 1:GOSUB 2000:NEXT P

940 POSITION 0,21:END
1000 GRAPHICS 0:POKE 752,1
1010 S1$="++++++";S2$="
!!!!!!!!!!!!"

1015 PRINT
1020 FOR I=1 TO 10:PRINT " ";S1$:PRIN
T " ";S2$:NEXT I:PRINT " ";S1$

1030 FOR I=0 TO 9:POSITION 3,I*2+2:PRI
NT I;:NEXT I
1040 POSITION 5,22:PRINT "A B C D E F
G H I J"

1050 FOR Y=10 TO 12:POSITION 9,Y:PRINT
"((( ";:POSITION 17,Y:PRINT "((( ";:NEX
T Y

1060 RETURN
2000 FOR Y=0 TO 9:FOR X=0 TO 9:SOUND 0
,110-Y*10-X,10,10
2010 A=A(X,P*10+Y):IF A=0 THEN 2040

2020 POSITION X*2+5,Y*2+2
2030 GOSUB 2500

2040 NEXT X:NEXT Y:SOUND 0,0,0,0:RETUR
N

2500 IF A<10 THEN PRINT A;
2510 IF A=SPY THEN PRINT "S";

2520 IF A=FLAG THEN PRINT "F";
2530 IF A=BOMB THEN PRINT "B";
2540 RETURN

3000 FOR Y=0 TO 9:FOR X=0 TO 9
3010 IF A(X,P*10+Y)=0 THEN 3040
3020 POSITION X*2+5,Y*2+2:PRINT " ";

3030 SOUND 0,X*10+20,10,10
3040 SOUND 0,0,0,0:NEXT X:NEXT Y:RETUR
N

4000 GOSUB 25:GOSUB 30
4010 IF A(R1,P*10+R2)=0 OR A(R3,P*10+R
4)=0 THEN POSITION 26,5:PRINT "ILLEGAL
MOVE!";:FOR Z=1 TO 1000:NEXT Z

4011 IF A(R1,P*10+R2)=0 OR A(R3,P*10+R
4)=0 THEN 4000

4020 A=A(R1,P*10+R2):A(R1,P*10+R2)=A(R
3,10*P+R4):A(R3,P*10+R4)=A

4030 GOSUB 2000:RETURN
```

```
5000 GOSUB 3000:GOSUB 8000:A=A(R1,P*10
+R2):POSITION R1*2+5,R2*2+2:GOSUB 2500
:A=A(R3,(1-P)*10+R4)

5005 POSITION R3*2+5,R4*2+2:GOSUB 2500
5010 GOSUB 25:POSITION 26,7:PRINT "PLA
YER ";CHR$(178-P);" ";:POSITION 26,8
:PRINT "FACE SCREEN";

5015 POSITION 26,9:PRINT "FOR ENEMY "
 ;:POSITION 26,10:PRINT "ATTACK "
 ;
5020 FOR Z=1 TO 500:NEXT Z

5030 POSITION 26,2+P*10:GOSUB 140+A1*1
0:POSITION 26,2+(1-P)*10:GOSUB 140+A2*
10

5040 FOR Z=1 TO 500:NEXT Z
5050 W(0)=0:W(1)=0:IF A2=FLAG THEN POP
:GOTO 900

5060 IF (A1=SPY AND A2=1) OR (A2=BOMB
AND A1=8) THEN W(P)=1:GOTO 5110
5070 IF A2=BOMB THEN W(1-P)=1:GOSUB 60
00:GOTO 5110

5080 IF A1=A2 THEN 5110
5090 IF A2<A1 THEN W(1-P)=1
5100 IF A2>A1 THEN W(P)=1

5110 FOR I=2 TO 4:FOR J=1 TO 5
5120 IF W(P)=0 THEN POSITION 25+J,P*10
+I:PRINT " ";SOUND 0,RND(0)*100+150,6,
14

5125 FOR T=1 TO 10:NEXT T:SOUND 0,0,0,
0
5130 IF W(1-P)=0 THEN POSITION 25+J,(1
-P)*10+I:PRINT " ";:SOUND 1,RND(0)*100
+150,6,14
5135 FOR T=1 TO 10:NEXT T:SOUND 1,0,0,
0

5140 NEXT J:NEXT I:FOR Z=1 TO 500:NEXT
Z
5150 GOSUB 25

5160 RETURN
6000 FOR T=1 TO 3:FOR Z=0 TO 15
6010 SETCOLOR 2,Z,14

6020 SOUND INT(RND(0)*4),RND(0)*100+15
0,12,12
6030 NEXT Z:NEXT T

6040 FOR T=0 TO 3:SOUND T,0,0,0:NEXT T
:SETCOLOR 2,9,4:RETURN
7000 RETURN

8000 RESTORE :FOR I=1 TO 7:READ Z1,Z2:
SOUND 0,Z1,10,10:FOR T=1 TO Z2/4:NEXT
T:NEXT I

8010 DATA 96,50,72,50,57,50,48,100,57,
50,48,200,0,0
8020 RETURN
```



continued from page 44

```
1510 IFC=1THENAD=AD+1ELSEIFC=2THENAG=AG+1ELSEIFC=3THENAJ=AJ+1ELS
EIFC=4THENAM=AM+1ELSEIFC=5THENAA=AA+1
1520 RETURN
```

Lines 1530-1560: Armored move verifier.

```
1530 '
1540 IFL=1THENAE=AE-1ELSEIFL=2THENAH=AH-1ELSEIFL=3THENAK=AK-1ELS
EIFL=4THENAN=AN-1ELSEIFL=5THENAB=AB-1
1550 IFC=1THENAE=AE+1ELSEIFC=2THENAH=AH+1ELSEIFC=3THENAK=AK+1ELS
EIFC=4THENAN=AN+1ELSEIFC=5THENAB=AB+1
1560 RETURN
```

Lines 1570-1600: Airborne move verifier.

```
1570 '
1580 IFL=1THENAF=AF-1ELSEIFL=2THENAI=AI-1ELSEIFL=3THENAL=AL-1ELS
EIFL=4THENAP=AP-1ELSEIFL=5THENAC=AC-1
1590 IFC=1THENAF=AF+1ELSEIFC=2THENAI=AI+1ELSEIFC=3THENAL=AL+1ELS
EIFC=4THENAP=AP+1ELSEIFC=5THENAC=AC+1
1600 RETURN
```

Lines 1610-1630: Determine enemy strategy.

```
1610 '
1620 GOSUB1470:GOSUB1640:EC=(ES)*(FT-FR)/(4*FT):IFGG>0THENECEC+
(ES-4*EC)*(GG/80)
1630 EC=INT(EC):RETURN
```

Lines 1640-1650: Battleline strength tally.

```
1640 '
1650 S(1)=8*AD+10*AE+6*AF:S(2)=8*AG+10*AH+6*AI:S(3)=8*AJ+10*AK+6
*AL:S(4)=8*AM+10*AN+6*AP:RETURN
```

Lines 1660-1860: Corps battles.

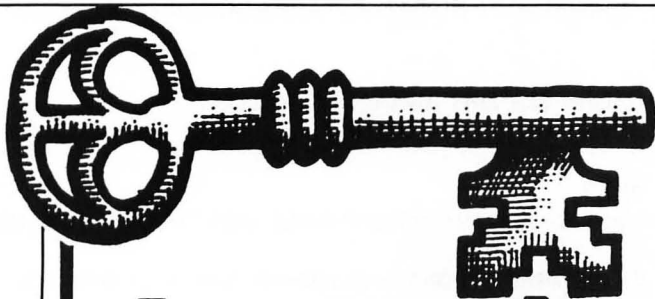
```
1660 '
1670 MV=S(C)-EC:LS=0
1680 IFMV=2ORMV=3THENL=L-64
1690 IFMV=-2ORMV=-3THENL=L+64
1700 IFMV=4ORMV=5THENL=L-64:ES=ES-8
1710 IFMV>5THENL=L-64:ES=ES-16
1720 IFMV=-4ORMV=-5THENL=L+64:LS=1
1730 IFMV<-5THENL=L+64:LS=2

1740 IFLS=0THENRETURN
1750 A$="":IFX<>0THENA$=A$+STRING$(X,88)
1760 IFT<>0THENA$=A$+STRING$(T,84)
1770 IFA<>0THENA$=A$+STRING$(A,65)
1780 LN=LEN(A$):IFLN=0THENRETURN
1790 IFLN<LSTHENL$=LN

1800 FORW=1TOLN:K$(W)=MID$(A$,W,1):NEXTW
1810 FORW=1TOLS
1820 R=RND(LN):IFK$(R)="/"THEN1820
1830 IFK$(R)="X"THENX=X-1
1840 IFK$(R)="T"THENT=T-1
1850 IFK$(R)="A"THENA=A-1
1860 K$(R)="/" :NEXTW:RETURN
```

Lines 1870-1960: Enemy breakthrough. Displays a random message.

continued on next page



The key to
your TRS-80*.

Learning Level II

by David A. Lien

Written by David A. Lien, the author of your Level I User's Manual, LEARNING LEVEL II picks right up where the Level I Manual leaves off. It also supplies the changes needed to make the Level I Manual compatible with your Level II TRS-80*.

LEARNING LEVEL II covers all Level II BASIC beyond Level I, plus much more. It shows you how to use the Editor, explains what the many error messages are really saying, and leads you thru conversion of Level I programs to Level II.

Dual cassettes, printer, the expansion interface with clock and other features are explained in the same easy-to-learn style that made the Level I Manual famous. LEARNING LEVEL II was created specifically for your Level II TRS-80*!

\$15.95

*TRS-80 is a registered trademark of Radio Shack and Randy Corporations.


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

continued from previous page

```
1870 '  
1880 PRINT@0,"THE ENEMY HAS BROKEN THROUGH YOUR LINES!"  
1890 R=RND(6)  
1900 IFR=1PRINT@64,"YOU ARE KILLED BY A GRENADE!"  
1910 IFR=2PRINT@64,"YOU ARE TAKEN PRISONER AND USED FOR PROPAGAN  
DA BROADCASTS!"  
1920 IFR=3PRINT@64,"THE COMMANDER OF NORPAC INTENDS TO HAVE YOU  
SHOT!"  
  
1930 IFR=4PRINT@64,"THEY ARE RIPPNG YOUR REMAINING FORCES TO SHR  
EDS!"  
1940 IFR=5PRINT@64,"YOU REPORT YOURSELF 'MISSING IN ACTION' AND  
HEAD FOR THE HILLS!"  
1950 IFR=6PRINT@64,"YOU SUSPECT TROUBLE!"  
1960 GOSUB1130:RETURN
```

Lines 1970-2070: Total Defeat. Your forces have been destroyed, and the computer picks a random insult to throw at the player.

```
1970 '  
1980 PRINT@0,"YOUR REMAINING FORCES ARE RETREATING IN PANIC BEFO  
RE THE ENEMY!"  
1990 R=RND(7)  
2000 IFR=1PRINT@64,"YOU HAVE LOST ALL CONTROL!"  
2010 IFR=2PRINT@64,"THE WAR IS LOST!"  
2020 IFR=3PRINT@64,"ONE OF YOUR JUNIOR OFFICERS SHOOTS YOU!"  
2030 IFR=4PRINT@64,"YOU ARE FORCED TO FLEE FOR YOUR LIFE!"  
2040 IFR=5PRINT@64,"YOU BEGIN TO WORRY ABOUT YOUR CAREER!"  
2050 IFR=6PRINT@64,"YOU FEAR THIS WILL AFFECT YOUR POLITICAL AMB  
ITIONS!"  
2060 IFR=7PRINT@64,"ONE OF YOUR TANK COMMANDERS SEES YOU AND RUN  
S OVER YOU!"  
2070 GOSUB1130:RETURN
```

Lines 2080-2170: Relief of command.

```
2080 '  
2090 PRINT@0,"YOU HAVE BEEN RELIEVED OF COMMAND DUE TO INCOMPETE  
NCY!"  
2100 R=RND(6)  
2110 IFR=1PRINT@64,"A COURT-MARTIAL HAS REDUCED YOU TO PRIVATE!"  
2120 IFR=2PRINT@64,"YOU HAVE BEEN FORCED TO RESIGN AND SENT HOME  
IN DISGRACE!"  
  
2130 IFR=3PRINT@64,"YOUR MILITARY CAREER IS OVER!"  
2140 IFR=4PRINT@64,"YOUR FRIENDS AT NORPAC NO LONGER SPEAK TO YO  
U!"  
2150 IFR=5PRINT@64,"YOUR SONS HAVE CHANGED THEIR LAST NAMES!"  
2160 IFR=6PRINT@64,"THE PRESIDENT HAS CALLED YOU "A TURKEY IN TH  
E MIDST OF EAGLES!"  
2170 GOSUB1130:RETURN
```

Lines 2180-2270: Victory!

```
2180 '  
2190 PRINT@0," YOU HAVE WON! THE WAR IS  
OVER!"  
2200 R=RND(6)
```

```
2210 IFR=1PRINT@64,"YOU ARE A NATIONAL HERO!"  
2220 IFR=2PRINT@64,"THE PRESIDENT WANTS YOU AS HIS 1996 ELECTION  
RUNNING MATE!"  
2230 IFR=3PRINT@64,"HOLLYWOOD WANTS TO MAKE A MOVIE ABOUT YOUR L  
IFE!"  
2240 IFR=4PRINT@64,"A STATUE OF YOU NOW RESTS ON THE PARADE GROU  
ND AT WEST POINT!"  
2250 IFR=5PRINT@64,"YOUR MEMOIRS ARE SELLING LIKE HOTCAKES!"  
2260 IFR=6PRINT@64,"YOUR SONS HAVE DECIDED TO FOLLOW IN YOUR FOO  
TSTEPS!"  
2270 GOSUB1130:RETURN
```

Lines 2280-2310: Warnings given for repeated mistakes.

```
2280 '  
2290 IFMS=3PRINT@0,E$;:PRINT@0,"YOUR MISTAKES HAVE BEEN REPORTED  
TO NORPAC COMMAND!";  
2300 IFMS=4PRINT@0,E$;:PRINT@0,"NORPAC COMMAND WILL NOT PERMIT F  
URTHER MISTAKES!";  
2310 IFMS<3THENRETURNELSEGOSUB1130:RETURN
```

Lines 2320-2340: Enemy attack along all of front line.

```
2320 '  
2330 PRINT@0,E$;:PRINT@0,"THE ENEMY IS ATTACKING ALONG ALL CORPS  
FRONT LINES!!!";  
2340 FORT=1TO20:PRINT@L1+3,"*";:PRINT@L2+3,"*";:PRINT@L3+3,"*";  
PRINT@L4+3,"*";:FORTA=1TO20:NEXTTA:PRINT@L1+3," ";:PRINT@L2+3,"  
";:PRINT@L3+3," ";:PRINT@L4+3," ";:FORTA=1TO20:NEXTTA,T:RETURN
```

Lines 2350-2390: Unauthorized offensives.

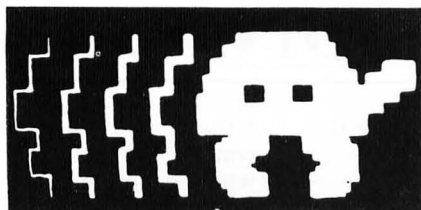
```
2350 '  
2360 GOSUB1470:GOSUB1640:KK=0:X1=((AE>1ANDS(1)>26)OR(S(1)>28)):X  
  
2=((AH>1ANDS(2)>26)OR(S(2)>28)):X3=((AK>1ANDS(3)>26)OR(S(3)>28))  
:X4=((AL>1ANDS(4)>26)OR(S(4)>28))  
2370 IFR=1ANDX1THENBB=L0$(1):LL=L1:KK=1ELSEIFR=2ANDX2THENBB=L0  
$(2):LL=L2:KK=1ELSEIFR=3ANDX3THENBB=L0$(3):LL=L3:KK=1ELSEIFR=4A  
NDX4THENBB=L0$(4):LL=L4:KK=1
```

```
2380 IFKK=1PRINT@0,E$;:PRINT@0,"THE ";BB$;" COMMANDER IS LAUNCHI  
NG AN ATTACK!";ELSEReturn  
2390 FORT=1TO20:PRINT@LL+3,"*";:PRINT@LL+7,"*";:FORTA=1TO20:NEXT  
TA:PRINT@LL+3," ";:PRINT@LL+7," ";:FORTA=1TO20:NEXTTA,T:RETURN
```

Lines 2400-2420: Variable initialization.

```
2400 '  
2410 L0$(1)="I CORPS":L0$(2)="II CORPS":L0$(3)="III CORPS":L0$(4  
)="IV CORPS":L0$(5)="RESERVE":DV$(1)="INFANTRY":DV$(2)="ARMORED"  
:DV$(3)="AIRBORNE":ES=136  
2420 L1$="--- I ---":L2$="--- II ---":L3$="--- III ---":L4  
$="--- IV ---":L5$="- 8 -":L6$="- 3 -":L1=518:L2=531:L3=544:L4  
=557:L5=512:L6=570:E$=STRING$(192,32):FORT=1TO20:READD(T):NEXTT  
ED=11+RND(5):RETURN
```



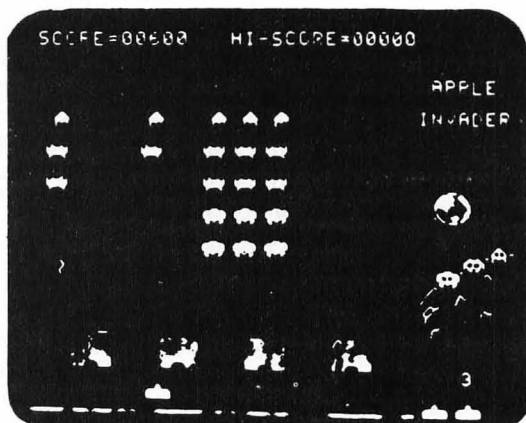


FROM

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



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

MATH DECATHLON



Part 2

by Jon Voskuil

Welcome back to the Olympics! This second installment of "Math Decathlon" will add two more events and the music-generating routines.

The two purposes stated in the first article continue into this and the remaining installments: To develop a fun and interesting math game program, and to learn good programming techniques along the way. (If you missed the first installment, back issues are available for a mere pittance.) I'll continue to explain commands and syntax that are unique to the Apple, for the benefit of those who are working with different computers.

The first thing you'll need to do, before sitting down to type in the lines which follow, is to load the first part of the program, which you typed in last time. Failing to do this will void all warranties — not to mention generating endless insults from your computer such as UNDEF'D STATEMENT ERROR and BAD SUBSCRIPT ERROR.

Having done that, the first new thing to be done is to change lines 310 and 320 to include the names of the next two events, as shown in the listing. Line 350 is also added, calling the subroutine at line 12000, which pokes into memory a short Machine Language routine to generate musical notes.

[Translation note: This and all other music routines are designed for the Apple; if you're translating for another computer, you'll have to create your own. Or you can omit the routines entirely, and the rest of the program will work fine. In that case you should omit lines 350, 11090-11190, and 12000-13425, and change lines 420 and 430 appropriately.]

EVENT #3: SPEED MATH

In the third event, which begins at line 3000, a player's score depends on how quickly the correct answer can be typed in. This is the only event which functions in this way, and it calls for a rather different kind of input

routine. How do you measure the delay between displaying the problem on the screen, and receiving the answer from the keyboard? One answer is to invest a fewscore-and-seven dollars in a real-time clock for one of those gaping slots inside your Apple. A less expensive (nay, free!) answer is to use a loop with a keyboard peek and a GET statement in it, such as you find in lines 3070-3150. [Translation note: GET works just as INPUT does, except that it can input only one character, and pressing RETURN (that's ENTER to you Radio Shack fans) is not needed. It's similar to the S-80's INKEY\$, except that GET waits for a keystroke.]

Here's how the input routine works, Line 3080 sets up a loop which shrinks in length at higher skill levels. The cursor is positioned at PSN, and then the PEEK takes a look at the keyboard to see if a key has been pressed. If not, control skips to the end of the loop at 3150, whence it returns to 3080. If a key has been pressed, however, control falls through to line 3090 where the pressed key is trapped in D\$ using GET, after which the keyboard is cleared with the POKE. The captured character is examined and, if it's not a backspace, control skips to 3120. There it's checked again to see if it's a RETURN. If so, the player has finished the entry and control breaks out of the loop to the scoring routine.

If the key pressed was neither the backspace nor RETURN, then in line 3130 its value is stored in NUM(DI) — the next DIgit of the NUMber — and printed on the screen. The position of the cursor is then incremented in line 3140, and control jumps back to the beginning of the loop again to start searching for the next digit to be entered. Simple, eh? This is a routine which you can transplant into one of your own programs whenever you want to limit or monitor the amount of time taken for a user response.

The scoring routine begins in line 3160, and makes use of the value of T (the loop counter from above) to determine the player's

score. Finally, lines 3210-3230 contain the three subroutines which generate values (and assign the math function symbol, +, -, or x) for the three types of problems.

EVENT #4: STARRY STARRY NIGHT

The next event has an unlikely title for a math game, but one which is very descriptive nevertheless. The object is for each player to estimate how many points of light the computer flashes on the screen. The score depends on how close the guess is.

The "stars" are simply periods printed out on the text screen, in a more-or-less random arrangement. The method used to do this is to build a string of spaces and a random number of periods, and then to print that string several times to fill the screen. This is accomplished in lines 4050-4100. Line 4060 determines how often a period is to be stuck in, using a somewhat obscure formula derived mostly by trial and error. X\$ is built up to 137 characters, and then printed seven times — odd numbers which make the display look fairly random. Lines 4110-4170 compute the player's score, based on percentage error; the "standard" scoring subroutine at line 30 is not used here. And that's all the coding there is to event #4.

THE MUSIC ROUTINES

Each time the scoreboard is displayed, the players' scores for the event just completed are compared. If there is a single winner (not a tie), then a part of that player's national song is played. This wondrous feat is accomplished by lines 11090-11190, which must be inserted into the existing scoreboard subroutine. Lines 11100-11120 find the high points (HP) for this event, and lines 11130-11150 check for a tie. If there is a winner, then lines 11160-11190 play the appropriate song. This is done by assigning values to the variables SKIP and PLAY, which specify how many musical notes in the DATA statements are to be skipped over,

and then how many are to be played.

Subroutine 13000 is called to accomplish that. After RESTORE, pitch/time data pairs are idly (but quickly) read until reaching the ones to play, and then these are POKEd into the appropriate memory locations and the music subroutine is CALLED. This is the Machine Language routine which is initially set up by subroutine 12000, which is listed here for your typing pleasure. The remaining lines, from 13100 on contain all the data necessary for the national songs, and of course must be typed in without error to produce sweet music.

Next month: The next three exciting events!

Variables (Part 2)

A, B: Numbers generated for speed-math problem.

C: Correct answer for speed-math problem.

D\$: Input character.

DI: Place-value position for entered digit.

F\$: Arithmetic operation symbol: +, -, or x.

HP: High points for this event.

K: Loop counter.

NUM: Number of stars.

NUM(di): Storage array for entered digits.

P%: Temporary variable for points scored.

PITCH: Memory address for music routine.

PK: Loop counter.

PLAY: Number of notes to play.

POK\$: String of poke values for music routine.

PSN: Tab position for input routine.

R: Random number which affects number of stars.

SKIP: Number of notes to skip.

T: Input routine timing counter.

TIME: Memory address for music routine.

TYE: Used to determine if tie score.

WNR: Number of winning player for this event.

X\$: String of "stars".

New lines to replace previous lines 310 and 320.

```
310 E$(1) = "MISSING #";E$(2) = "
    GUESS A #";E$(3) = "SPEED MA
    TH";E$(4) = "STARS"
```

```
320 EE$(1) = " 1. FIND THE MISSIN
    G NUMBER";EE$(2) = " 2. GUES
    S A NUMBER";EE$(3) = " 3. SP
```

```
EED MATH";EE$(4) = " 4. STAR
    RY STARRY NIGHT"
```

Call subroutine to poke in machine-language tone generator.

```
350 GOSUB 12000
```

Event #3:

Speed Math

Print instructions.

```
3000 E = 3:NXT = 0
```

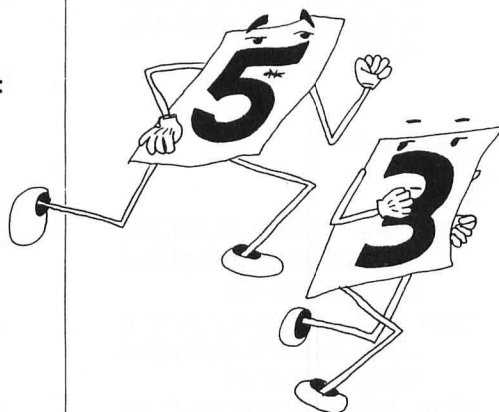
```
3010 FOR P = 1 TO NP: GOSUB 10
```

```
3020 PRINT "IN THIS EVENT I WILL
    GIVE YOU A SERIES OF THREE
    MATH PROBLEMS. THE MORE QU
    ICK-LY YOU CAN TYPE IN THE C
    ORRECT ANSWER, THE HIGHER Y
    OUR SCORE WILL BE."
```

```
3030 PRINT : PRINT "IF YOU TYPE
    IN THE WRONG ANSWER, OR RUN
    OUT OF TIME, YOUR SCORE FOR
    THAT PROBLEM WILL BE ZERO."
```

```
3040 GOSUB 90: IF NXT THEN 4000
```

Generate and display an addition, subtraction, or multiplication problem.



```
3050 FOR PROB = 1 TO 3: ON PROB GOSUB
    3210,3220,3230
```

```
3060 HOME : VTAB 5: HTAB 19 + (A
    < 10) + (A < 100): PRINT A:
    PRINT : HTAB 17 + (B < 10) +
    (B < 100): PRINT F$;" ";B: PRINT
    TAB( 17)"-----": PRINT TAB(
    21)"?": PRINT : PRINT : PRINT
    : PRINT TAB( 10)"ANSWER:";
```

Timed input routine.

```
3070 DI = 0:A1 = 0:PSN = 20: FOR
    T = 1 TO 300: NEXT T
```

```
3080 FOR T = 1 TO 600 - S(P) * 1
    00: HTAB PSN: IF PEEK ( - 1
    6384) < 127 THEN 3150
```

```
3090 GET D$: POKE - 16368,0: IF
    ASC (D$) < > 8 THEN 3120
```

```
3100 PSN = PSN - 1:DI = DI - 1: IF
    DI < 0 THEN DI = 0
```

```
3110 GOTO 3150
```

```
3120 IF ASC (D$) = 13 THEN 3170
```

```
3130 DI = DI + 1: PRINT VAL (D$)
    ;NUM(DI) = VAL (D$): IF DI
    = 4 THEN 3170
```

```
3140 PSN = PSN + 1
```

```
3150 NEXT T
```

Scoring routine.

```
3160 PRINT : PRINT : PRINT "TIME
    'S UP. . ."
```

```
3170 FOR K = 1 TO DI:A1 = A1 + N
    UM(K) * 10 ^ (DI - K): NEXT
    K
```

```
3180 A2 = C:PX = 10 - INT (10 *
    T / (600 - S(P) * 100)): GOSUB
    40
```

```
3190 GOSUB 100
```

```
3200 NEXT PROB,P: GOSUB 11000: GOTO
    4000
```

Generate addition problem.

```
3210 A = INT ( RND (1) * 10 ^ (S
    (P) / 2 + 1)) + 1:B = INT (
    RND (1) * 10 ^ (S(P) / 2 +
    1)) + 1:C = A + B:F$ = "+": RETURN
```

Generate subtraction problem.

```
3220 A = INT ( RND (1) * 10 ^ (S
    (P) / 2 + 1)) + 5:B = INT (
    RND (1) * (A - 3 * (S(P) +
    1))) + 3:C = A - B:F$ = "-":
    RETURN
```

Generate multiplication problem.

```
3230 A = INT ( RND (1) * 10 * (S
    (P) + 1)):B = INT ( RND (1)
    * 6,9) + 3:C = A * B:F$ = "
    X": RETURN
```

Event #4:

Starry Starry Night

Print instructions.

```
4000 E = 4:NXT = 0
```

```
4010 FOR P = 1 TO NP: GOSUB 10
```

```
4020 PRINT "IN THIS EVENT I WILL
    BRIEFLY SHOW YOU A BLACK NI
    GHT SKY WITH A NUMBER OF STA
    RS"
```

```
4030 PRINT "IN IT. YOU ARE TO G
    UESS HOW MANY STARS THERE AR
    E. THE CLOSER YOUR GUESS, T
    HE MORE POINTS YOU'LL GET,"
```

```
4040 GOSUB 90: IF NXT THEN 5000
```

Choose and display random number of 'stars'.

```
4050 FOR PROB = 1 TO 3: HOME :NU
    M = 0:X$ = ""
```

```
4060 R = RND (1) * .09 * (S(P) +
    1) + .03 + .02 * S(P)
```

continued on next page

continued from previous page

```
4070 FOR I = 1 TO 137: IF RND (
1) > R THEN X$ = X$ + " ": NEXT
: GOTO 4090
4080 X$ = X$ + " ": NUM = NUM + 1:
NEXT
4090 FOR I = 1 TO 7: PRINT X$:: NEXT

4100 NUM = NUM * 7: FOR I = 1 TO
2000: NEXT I
```

Input and score player's guess.

```
4110 HOME : VTAB 8: INPUT "HOW M
ANY STARS? "; AN$: AN = VAL (
AN$)
4120 PZ = 10 - INT (20 * ABS (N
UM - AN) / NUM): IF PZ < 0 THEN
PZ = 0
4130 PRINT : PRINT "THE ACTUAL N
UMBER WAS "; NUM; ";"
4140 PRINT : PRINT "YOUR SCORE I
S "; PZ; " OUT OF 10."
4150 PZ(P,E) = PZ(P,E) + PZ
4160 GOSUB 100
4170 NEXT PROB: NEXT P: GOSUB 11
000
```

Event #5 will begin at line 5000.
For now, end here.

5000 END

Routine to play a portion of the
national song of this event's
winner.

```
11090 IF NP = 1 THEN 11200
11100 HP = 0: WNR = 0
11110 FOR P = 1 TO NP: IF PZ(P,E
) > HP THEN HP = PZ(P,E): WNR
= P
11120 NEXT P
11130 TYE = - 1
11140 FOR P = 1 TO NP: IF PZ(P,E
) = HP THEN TYE = TYE + 1
11150 NEXT P: IF TYE THEN 11200
11160 SKIP = 17: PLAY = 26: IF C(W
NR) = 2 THEN SKIP = 117: PLAY
= 28
11170 IF C(WNR) = 3 THEN SKIP =
145: PLAY = 18
11180 IF C(WNR) = 4 THEN SKIP =
261: PLAY = 27
11190 GOSUB 13000
```

Subroutine to poke in
machine-language tone generator.

```
12000 POK$ = "173,048,192,136,208
,004,198,001,240,008,202,208
,246,166,000,076,000,003,096
"
12010 FOR PK = 0 TO 18: POKE 768
+ PK, VAL ( MID$(POK$,PK *
4 + 1,3)): NEXT PK
```

12020 PITCH = 0: TIME = 1: RETURN

Subroutine to read proper notes
from DATA statements and play them.

```
13000 RESTORE : IF SKIP = 0 THEN
13020
13010 FOR I = 1 TO SKIP: READ PT
, TM: NEXT
13020 FOR I = 1 TO PLAY: READ PT
, TM: POKE PITCH, PT: POKE TIM
E, TM: CALL 768: NEXT
13030 RETURN
```

Data for Great Britain.

```
13100 DATA 64,128,64,128,57,128
,68,192,64,64,57,128,51,128,
51,128,48,128,51,192
13105 DATA 57,64,64,128,57,128
,64,128,68,128,64,255, 1,128
,43,128,43,128,43,128
13110 DATA 43,192,48,64,51,128,
48,128,48,128,48,128,48,192,
51,64,57,128,51,128
13115 DATA 48,64,51,64,57,64,64
,64,51,192,48,64,43,128,38,6
4,43,32,48,32,51,128,57,128,
64,255
```

Data for U.S.A.

```
13200 DATA 72,96,86,32,108,128,
86,128,72,128,54,255,43,96,4
8,32,54,128,86,128
13205 DATA 76,128,72,255,72,64
,72,64,43,192,48,64,54,128,5
7,255,64, 64,57,64
13210 DATA 54,128,54,128,72,128
,86,128,108,128,72,96,86,32,
108,128,86,128,72,128
13215 DATA 54,255,43,96,48,32,
54,128,86,128,76,128,72,255,
72,64,72,64,43,192
13220 DATA 48,64,54,128,57,255,
64,64,57,64,54,128,54,128,72
,128,86,128,108,128
13225 DATA 43,64,43,64,43,128,
40,128,36,128,36,255,40,64,4
3,64,48,128,43,128
13230 DATA 40,128,40,255,40,64,
40,64,43,192,48,64,54,128,57
,255, 64,64,57,64
13235 DATA 54,128,86,128,76,128
,72,255,72,128,54,128,54,128
,54,64,57,64,64,128
13240 DATA 64,128,64,128,48,128
,40,64,43,64,48,64,54,64,54,
128,57,255,72,64
13245 DATA 72,64,54,192,48,64,
43,64,40,64,36,255,54,64,48,
64,43,192,40,64,48,128,54,25
5
```

Data for France.

```
13300 DATA 86,22,86,68,86,22,64
,90,64,90,57,90,57,90,43,135
,51,45,64,68
13305 DATA 64,22,51,68,64,22,76
,90,48,180,57,68,68,22,64,22
5,1,45,64,68
13310 DATA 57,22,51,90,51,90,51
,90,48,68,51,22,51,45,57,45,
57,135,1,45
13315 DATA 57,68,51,22,48,90,48
,90,48,90,43,68,48,22,51,180
,1,90,43,68
13320 DATA 43,22,43,90,51,68,64
,22,43,90,51,68,64,22,86,180
,1,68,86,22
13325 DATA 86,45,68,45,57,180,4
8,90,57,45,68,45,64,90,64,90
,72,180,76,90
13330 DATA 64,45,64,45,64,90,68
,45,64,45,57,180,1,90,57,90,
54,135,54,45
13335 DATA 57,45,54,45,48,45,43
,45,57,180,1,90,54,45,57,45,
64,135,64,45
13340 DATA 64,45,54,45,57,45,64
,45,64,45,68,45,68,90,1,158,
43,22,43,248
```

```
13345 DATA 43,22,51,68,64,22,57
,255,1,68,43,22,43,248,43,22
,51,68,64,22
13350 DATA 57,180,1,90,86,90,64
,255,57,90,51,255,48,90,43,9
0,38,90,57,255
13355 DATA 38,90,43,248,51,22,4
8,68,57,22,64,255
```

Data for Scotland.

```
13400 DATA 72,96,72,32,72,96,86
,32,72,96,64,32,54,128,64,96
,64,32,64,96
13405 DATA 72,32,64,96,57,32,54
,96,48,32,43,96,43,32,48,96,
72,32,72,96
13410 DATA 48,32,43,128,54,96,6
4,32,64,96,72,32,72,255,43,9
6,43,32,43,96
13415 DATA 48,32,43,96,40,32,36
,128,48,96,48,32,48,96,54,32
,48,96,43,32
13420 DATA 40,128,36,96,43,32,4
8,96,54,32,54,96,48,32,43,12
8,54,96,64,32
13425 DATA 64,96,72,32,72,255
```



What to Do

AFTER

the Program Loads

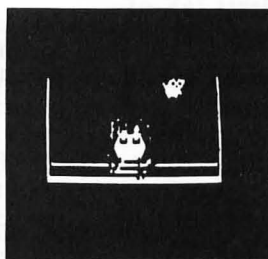
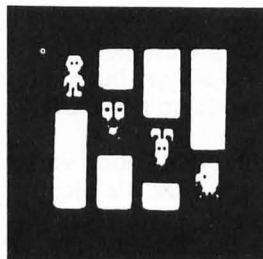
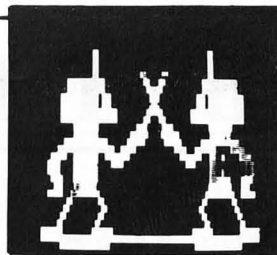
by Edward Ting

OK, you've got your program loaded, as per Sherry Taylor's instructions in December 1980 **SoftSide**. Now what? Sit there and stare at the asterisks left on the screen? Here, in a sequel to "What to Do While the Program Loads", is a step-by-step guide.

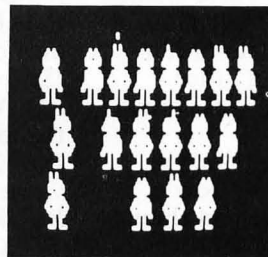
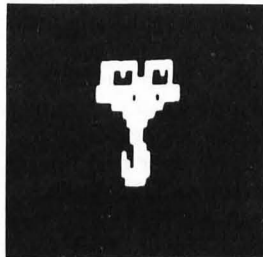
1. In wild anticipation, type in R-U-N and ENTER.
2. Hit ENTER again in response to the MEMORY SIZE question.
3. Rewind the tape and load it again.
4. See page 28 of Dec. **SoftSide**.
5. Again, type in R-U-N and ENTER.
6. Read the response.
7. Grit your teeth after confirming that there is no syntax error in line 65536.
8. Blame someone else; a child is best.
9. Rewind the tape and CLOAD again. Stay in the room this time.
10. Flare your nostrils after hearing your CTR-80 automatically stop at the end of the tape.
11. Walk up to the keyboard.
12. Hit it.
13. Feverishly attempt to replace the keys that have fallen out.
14. Call a Radio Shack Service Center.
15. Try to stop Radio Shack serviceman from laughing. Hang up.
16. Pick a fight with a neighbor.
17. Ice the area of your anatomy where he hit you.
18. Ax your TRS-80™ in two, salvaging what parts you can.
19. Bring the original tape to a fellow computerist's house.
20. Write a letter to the software house, asking for a replacement for the defective tape.
21. Have fun reassembling your computer. The Radio Shack warranty does not cover damages inflicted by ax.



FIVE



FROM CHRISTOPHERSON



ANDROID NIM

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

BEE WARY

This fast-paced real time action game is a contest between a Bee operated by the player and a Spider operated by the computer. Machine language subroutines, but loads as Level II for easy operation. 16K, S-80, Cassette \$14.95

SNAKE EGGS

Here is a computerized reptilian version of 21 complete with arrogant snakes and appropriate sound. 16K, S-80, Level II, Cassette .. \$14.95

LIFE TWO

Two in one: Game of Life, at an astounding 100 generations a minute, plus Battle of Life with animated creatures and sound.
16K, S-80, Level II, Cassette \$14.95

DUEL-N-DROIDS

You are the Fencing Master and all you need to do is train your Android by making him duel the machine's androids. After he is trained, enter him in the Tournament and sit back and enjoy the fun. Features included in this game are: sound effects, extensive graphic displays and multiple playing levels.

16K, S-80, Level II, Cassette \$14.95
S-80, Disk \$20.95



TSE-HARDSIDE

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

ROBOT BUILDER



by Arnold Gross

"Robot Builder" is an S-80 program requiring at least 16K of RAM.

If you've always wanted to build your own robot, but your mechanical aptitude is such that you not only have all thumbs but all your thumbs have hangnails, "Robot Builder" may be just the program for you. The process is simplicity itself: Just push the space bar and the computer will do the rest. Sound easy? There is one restriction, however. The robot must be built in a certain order, starting with a body and then progressing to wheels, neck, arms, head, and scanner.

Not the most mentally taxing game we've published, "Robot Builder" is nicely done and should provide many hours of amusement for the little ones clustered around the keyboard.

Notice how parts of this listing are in lower case. If you have a lower case modification, or if you plan to use "Robot Builder" on a system that does, type in these lower case letters while holding down the SHIFT key. The letters will appear the same on the screen,

but when the program is run they will become lower case!

VARIABLES

A\$: Either "YOU" or "I", used for printing messages.
 B\$: As A\$ above, but contains possessive form ("YOUR", "MY").
 CL: FOR/NEXT loop variable used to erase the die picture from the screen.
 D0, D1, D2, D3, D4, D9, DN, DR, E\$: Various graphic and control strings.
 I: FOR/NEXT loop variable to POKE lower case driver into memory.
 I\$: INKEY\$ loop variable.
 LC: Various values associated with the lower case option.
 L\$: Code to clear one line of the video screen -CHR\$(30).
 R: Die rolled.
 T: Current player. 1 = Human
 2 = Computer.
 X, Y: Used in graphic FOR/NEXT loops.
 Z: PRINT@ screen position. Takes on a different value for each player to instruct the program where to print messages for that player.
 B: 1 = Player's robot has a body, 0 = it doesn't.

N: 1 = Player's robot has a neck, 0 = it doesn't.
 H: 1 = Player's robot has a head, 0 = it doesn't.
 A: Number of arms current player's robot has (0, 1 or 2).
 W: Number of wheels current player's robot has (0 to 6).
 S: 1 = Player's robot has a scanner, 0 = it doesn't.
 A(16): Stores the six variables above for each player.
 A(1-6) = B, N, H, A, W, S for human player.
 A(11-16) = B, N, H, A, W, S for computer.

Variables C, E, F, G, J, and K are used as pointers to the Y axis position of various parts of the robots. Each variable can have two different values, one for each player's robot. These variables are used by the various robot part routines (body, neck, arms, etc.) to determine where a certain part should be drawn on the screen. The routines these variables are used in are:

C: Scanner routine.
 E: Neck routine.
 F: Neck and Arm routines.
 G: Body and Arm routines.
 J: Wheel routine.
 K: Head routine.

```

100 ' *****
    * -- THE ROBOT BUILDING CONTEST -- *
    * Version 3.1 (9/80) TRS-80 16K *
    * Arnold Gross (212-666-4903) *
105 ' * 771 West End Ave., N.Y. 10025 *
    * *****

110 CLS: CLEAR: POKE16561,210: CLEAR200

    Line 115: Checks to see if lower case is installed, and
    if it is a routine at 820 is called to POKE in a
    lower case driver.

115 POKE15361,97: LC=PEEK(15361): CLS: IFLC=97: THEN GOSUB820

    Lines 120-135: Asks if player wants instructions. If
    so, branch to a routine at line 685 which prints them.

125 PRINT@656,"Do you need instructions (Y/N) ?";
130 I$=INKEY$: IF I$="Y" OR I$="y" THEN CLS: GOSUB685: GOTO140
135 IF I$="" OR I$="N" OR I$="n" THEN 130

    Lines 140-175: Variable initialization.

140 CLEAR200: DEFSTRD: DIMA(16): L$=CHR$(30)
145 D0=CHR$(130)+STRING$(3,143)+CHR$(129)
150 DB=CHR$(26)+STRING$(4,8): D9=DB+D0
155 D1=CHR$(136)+STRING$(3,191)+CHR$(132)

160 D2=CHR$(160)+STRING$(3,188)+CHR$(144)+DB+STRING$(3,131)
165 D3=STRING$(3,176)+CHR$(26)+STRING$(4,8)+D0
170 D4=CHR$(160)+STRING$(3,188)+CHR$(144)+DB+STRING$(3,179)+D9

175 DN="" do not need another ":DR=" robot has no ":E$=CHR$(212)

    Lines 180-215: Set up initial playing board.

180 CLS: PRINT@3,"Your robot: ";: PRINT@515,"My robot: ";
185 PRINT@367,CHR$(188)STRING$(15,140)CHR$(188);
190 FORY=64 TO 256 STEP 64: PRINT@367+Y,CHR$(191)CHR$(207)CHR$(191);
195 NEXT: PRINT@368,STRING$(15,140):; PRINT@687,STRING$(17,131);
200 PRINT@451,STRING$(44,176);
205 FORY=1970 TO 9799 STEP 64: PRINT@,CHR$(191):; NEXT
210 T=2: PRINT@790,"(press the space bar for your first roll)";
215 GOSUB785: PRINT@790,L$;

    Line 220: Rolls a six-sided die and graphically displays
    the result on the screen.

220 R=RND(6): ON R GOSUB260,265,270,275,285,290

    Lines 225-245: Determines whose turn it is, and branches to
    the proper routine. The routine at line 230 sets up the
    needed variables for when it is the computer's turn, while
    the routine as line 240 handles the player's turn.

225 IFT=2 THEN T=1: GOTO240 ELSE T=2
230 A$="I": B$="My": Z=599: C=28: K=31: E=37: F=38: G=39: J=44
235 B=A(11): N=A(12): H=A(13): A=A(14): W=A(15): S=A(16): GOTO250
240 A$="You": B$="Your": Z=87: C=4: K=7: E=13: F=14: G=15: J=20
245 B=A(1): N=A(2): H=A(3): A=A(4): W=A(5): S=A(6)

    Lines 250-255: Branches to the proper body part routine
    for the number rolled (body, neck, head, arm, wheel or
    scanner),
    
```



```
250 PRINT@Z-64,B$;" roll: ";R;:PRINT@Z+1,;
255 ONRGOTO370,420,465,520,575,640
```

Lines 260-295: Six subroutines to display each of the possible rolls of a six-sided die. The routine for the six rolls start at 260, 265, 270, 275, 285 and 290.

```
260 PRINT@501,D2;:RETURN
265 PRINT@498,D2;:PRINT@504,D2;:RETURN
270 PRINT@501,D2;:PRINT@562,D3;:PRINT@441,D1;:RETURN
275 PRINT@434,D2;:PRINT@440,D2;:PRINT@562,D2;
280 PRINT@568,D2;:RETURN
285 GOSUB270;PRINT@433,D1;:PRINT@570,D3;:RETURN
290 PRINT@434,D1;:PRINT@440,D1;
295 PRINT@498,D4;:PRINT@504,D4;:RETURN
```

Line 300: Erases the die display from the screen.

```
300 FORCL=432TO624STEP64;PRINT@CL,CHR$(206);:NEXT;GOTO220
```

Lines 305-320: Branch to line 325 if game has been won, otherwise clean up the display for the next turn.

```
305 IFA=2ANDS=1ANDW=6THEN325ELSEGOSUB780;PRINT@919,L$;
310 IFT=1THEN300ELSEPRINT@22,L$;:PRINT@86,L$;:PRINT@214,L$;
315 PRINT@278,L$;:PRINT@406,E$;:PRINT@534,E$;:PRINT@598,E$;
320 PRINT@726,L$;:PRINT@790,L$;:GOTO300
```

Lines 325-365: Win routine. The message "I WIN" or "YOU WIN" is flashed in lines 330-340, then lines 345-365 move the eyes of the winning player's robot.

```
325 B$=CHR$(205)
330 IFA$="I"THENA$="I WIN !! "ELSEA$="YOU WIN !! "
335 FORN=1TO10;PRINT@Z+322,B$;:FORT=1TO50;NEXT
340 PRINT@Z+322,A$;:FORT=1TO50;NEXTT,N
345 FORX1=1TO10;E=K+2;B=G+2;RESET(18,E);SET(22,E);RESET(19,E)
350 SET(23,E);FORX=12TO20;SET(X,B);SET(X+1,B);SET(X+2,B)
355 RESET(X,B);NEXTX;RESET(22,E);SET(18,E);RESET(23,E)
360 SET(19,E);FORX=21TO29;SET(X,B);SET(X+1,B);SET(X+2,B)
365 RESET(X,B);NEXTX,X1;GOTO790
```

Lines 370-415: Body routine. Determines if a body is needed or if one is already present. If one is needed, it is placed on the screen.

```
370 PRINT" (body) ...";
375 IFB=1THENPRINT@Z+128,A$DN"body,";:GOTO305
380 PRINT@Z+128,A$;" now have a body,";
385 GOSUB395;B=1;IFT=1THEN(1)=B;GOTO305
390 A(11)=B;GOTO305
```

```
395 FORX=9TO32;SET(X,G+4);NEXT;FORX=11TO30;SET(X,G);NEXT
400 FORY=G+1TOG+3;SET(10,Y);SET(11,Y);SET(30,Y);SET(31,Y);NEXTY
405 FORY=G+1TOG+3;SET(13,Y);SET(15,Y);SET(16,Y);SET(18,Y)
410 SET(20,Y);SET(21,Y);SET(23,Y);SET(25,Y);SET(26,Y)
415 SET(28,Y);NEXT;RETURN
```

Lines 420-460: Neck routine. Much the same as the body routine above, but it also checks to see if there is a body to place the neck on.

```
420 PRINT" (neck) ...";
425 IFN=1THENPRINT@Z+128,A$DN"neck,";:GOTO305
430 IFB=1THENPRINT@Z+128,A$;" now have a neck,";:GOTO445
435 PRINT@Z+128,"I cannot draw a neck!";
440 PRINT@Z+192,B$DR"body,";:GOTO305
445 GOSUB455;N=1;IFT=1THEN(2)=N;GOTO305
450 A(12)=N;GOTO305
455 FORY=ETOF;SET(18,Y);SET(19,Y);SET(22,Y);SET(23,Y)
460 NEXT;RETURN
```

Lines 465-515: Head routine.

```
465 PRINT" (head) ...";
470 IFH=1THENPRINT@Z+128,A$DN"head,";:GOTO305
475 IFN=1THENPRINT@Z+128,A$;" now have a head,";:GOTO490
480 PRINT@Z+128,"I cannot draw a head!";
485 PRINT@Z+192,B$DR"neck,";:GOTO305
490 GOSUB500;H=1;IFT=1THEN(3)=H;GOTO305
495 A(13)=H;GOTO305
500 FORX=14TO27;SET(X,K);SET(X,K+5);NEXT;FORY=K+1TOK+4
505 SET(14,Y);SET(15,Y);SET(26,Y);SET(27,Y);NEXT
510 SET(18,K+2);SET(19,K+2);SET(22,K+2);SET(23,K+2)
515 SET(20,K+4);SET(21,K+4);RETURN
```

Lines 520-570: Arm routine.

```
520 PRINT" (arm) ...";
525 IFA=2THENPRINT@Z+128,A$DN"arm,";:GOTO305
530 IFB=1THENPRINT@Z+128,"I will draw an arm,";:GOTO540
535 PRINT@Z+128,"I cannot draw an arm!";:GOTO440
540 A=A+1;ONAGOSUB555,565
545 IFT=1THEN(4)=A;GOTO305
550 A(14)=A;GOTO305
555 FORX=6TO7;FORY=FTOG+1;SET(X,Y);NEXTY,X
560 SET(8,G+1);SET(9,G+1);RETURN
565 FORX=34TO35;FORY=FTOG+1;SET(X,Y);NEXTY,X
570 SET(32,G+1);SET(33,G+1);RETURN
```

Lines 575-635: Wheel routine.

continued on next page



RAYWARE has the right tool for the job!

1. STAD

Unlock the power of the Z-80 with STAD (Symbolic Trace and Debug) a powerful monitor with special Debugging. Single step through the Machine Language programs. For tape and disk systems.
16-48K on one S-80 Cassette \$24.95

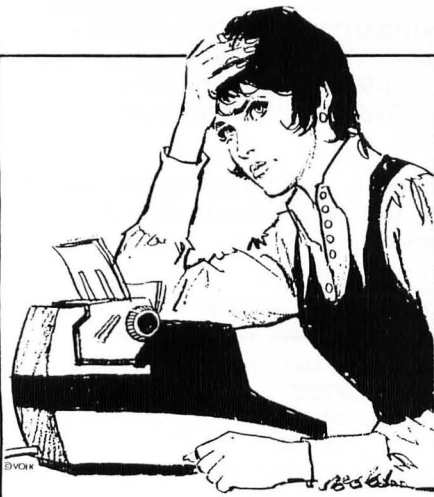
2. Z80 ZAP CMD

Powerful disk modification utility in Machine Language allows you to READ, DISPLAY, MODIFY, WRITE, and COMPARE disk sectors. It will calculate Hash Index Codes. You can recover killed disk files. Search for a byte and have it identified with a flashing cursor. Do disk backups, apply patches and fixes, and explore your disk. Instruction manual Included.
S-80 16K Disk Machine Language..... \$29.95



6 South St., Milford, NH 03055 (603) 673-5144

TOLL FREE OUT-OF-STATE 1-800-258-1790



Typing Tutor

by Roy Groth from . . .



Wish you were a better typist, but don't want to take (or pay for) a class? Teach yourself to type with the aid of your microcomputer. With TYPING TUTOR you will be quizzed and graded, but you set the pace at which you learn. TYPING TUTOR is a set of programs that lets you become as good a typist as you wish, allowing you to advance from one level to the next when you feel comfortable with your skills.

Let "hunt and peck" slip into the past, teach yourself speed and accuracy on the keyboard with TYPING TUTOR.

S-80/16K/Cassette/Level II
 \$19.95


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

continued from previous page

```

575 PRINT" (wheel) ...";
580 IFW=6THENPRINT@Z+128,A$DN"wheel.";;GOTO305
585 IFB=1THENPRINT@Z+128,"I will draw a wheel.";;GOTO595
590 PRINT@Z+128,"I cannot draw a wheel!";;GOTO440
595 W=W+1;ONWGOSUB610,615,620,625,630,635
600 IFT=1THENA(5)=W;GOTO305
605 A(15)=W;GOTO305
610 SET(10,J);SET(11,J);SET(10,J+1);RETURN
615 SET(14,J);SET(15,J);SET(14,J+1);RETURN
620 SET(18,J);SET(19,J);SET(18,J+1);RETURN
625 SET(22,J);SET(23,J);SET(23,J+1);RETURN
630 SET(26,J);SET(27,J);SET(27,J+1);RETURN
635 SET(30,J);SET(31,J);SET(31,J+1);RETURN
  
```

Lines 640-680: Scanner routine.

```

640 PRINT" (scanner) ...";
645 IFS=1THENPRINT@Z+128,A$DN"scanner.";;GOTO305
650 IFH=1THENPRINT@Z+128,A$;" now have a scanner.";;GOTO665
655 PRINT@Z+128,"I cannot draw a scanner!";
660 PRINT@Z+192,B$DR"head.";;GOTO305
665 GOSUB675;S=1;IFT=1THENA(6)=S;GOTO305
670 A(16)=S;GOTO305
675 FORX=17TO24;SET(X,C);NEXT;FORY=C+1TOC+2
680 SET(20,Y);SET(21,Y);NEXT;RETURN
  
```

Lines 685-785: Instruction subroutine.

```

685 PRINT@64,"As everyone knows, a robot has 12 basic parts:"
690 PRINT:PRINT" ", "1 body", "1 head"
695 PRINT" ", "2 arms", "1 scanner"
700 PRINT" ", "1 neck", "6 wheels";PRINT
705 PRINT" When you assemble a robot, you must start with"
710 PRINT"the body, otherwise you have nowhere to attach the"
715 PRINT"arms, wheels, or neck. Once you have attached the"
720 PRINT"neck, you can then screw the head to the neck, and"
725 PRINT"then put the scanner on top of the head."
730 GOSUB780;CLS
735 PRINT@132,"The object of this game is for you to finish"
740 PRINT"building your robot before I finish building mine."
745 PRINT" I will roll a die for each of us in turn. Then"
750 PRINT"I will tell you which part of the robot's body the"
755 PRINT"number rolled stands for, and whether or not that"
760 PRINT"part can be used. If it can, I will then draw that"
765 PRINT"section of the robot, in order that you may watch"
770 PRINT"to see which of us is closest to the completion of"
775 PRINT"building the robot."
  
```

Lines 780-785: Subroutine to delay program execution.

```

780 PRINT@919,"(press the space bar to continue)";
785 I$=INKEY$;IFI$=""THEN785ELSERETURN
  
```

Lines 790-815: Gives the option of playing another game.

```

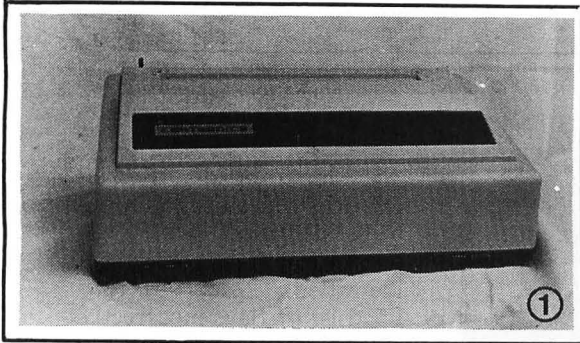
790 PRINT@406,E$;PRINT@534,E$;PRINT@598,E$;
795 PRINT@726,L$;PRINT@790,L$;PRINT@918,L$;
800 PRINT@790,"Do you want to play again (Y/N) ?";
805 I$=INKEY$;IFI$="Y"ORL$="y"THEN140
810 IFI$<"N"ANDI$>"n"THEN805
815 CLS;PRINT@128,"Good-bye!";PRINT:END
  
```

Lines 820-850: Subroutine to POKE lower case driver into RAM.

```

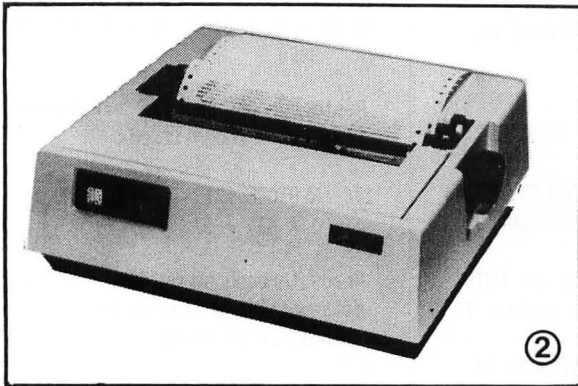
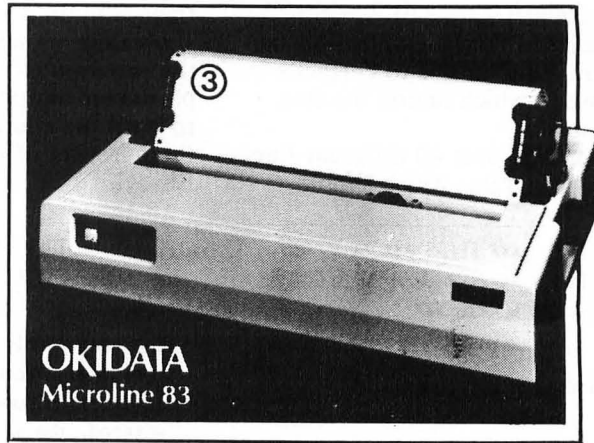
820 LC=(PEEK(16561)+1)+PEEK(16562)*256
825 IFLC>32767THENLC=(65536-LC)*-1
830 FORI=LC TO LC+40;READB;POKEI,B;NEXT
835 POKE16414,PEEK(16561)+1;POKE16415,PEEK(16562);RETURN
840 DATA 221,110,3,221,102,4,218,154,4,221,126,5,183,40
845 DATA 1,119,121,254,32,218,6,5,254,128,210,166,4,195
850 DATA 125,4,82,105,107,0,80,105,101,112,99,101,33
  
```





PICK YOUR PRINTER FROM TSE-HARDSIDE TODAY!

- RS Plotter/Printer (26-1190) \$1339.00
- RS Quick Printer II (26-1155) \$209.00
- TYPRINTER 221 (9-221) \$2495.00
- NEC 5510 SpinWriter (9-5520) \$3095.00
- NEC 5530 SpinWriter (9-5530) \$2595.00
- ① CENTRONICS 730 (9-730) \$659.00
- CENTRONICS 737 (9-737) \$769.00
- OKIDATA Microline-80 (9-80) \$489.00
- ② OKIDATA Microline-82 (9-82) \$669.00
- ③ OKIDATA Microline-83 (9-83) \$999.00
- OKIDATA SL300 (300 1pm) (9-SL300) \$3795.00
- EPSON MX-70 (9-MX70) \$419.00
- EPSON MX-80 (9-MX80) \$519.00
- EPSON MX-80FT (9-MX80FT) \$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) \$129.95
- Centronics Zip-Pack Ribbons (3-pack) . (21-01) \$15.95
- SpinWriter Multi-Strike Ribbons (3) . . . (21-02) \$18.95
- Microline Printer Ribbon (3-pack) (21-04) \$12.95
- EPSON Printer Ribbons (2-pack) (21-05) \$25.95

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. HARDSIDE accepts VISA & MASTERCARD. Certified checks and Money Orders; Personal checks accepted (takes 3 weeks to clear). HARDSIDE 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.

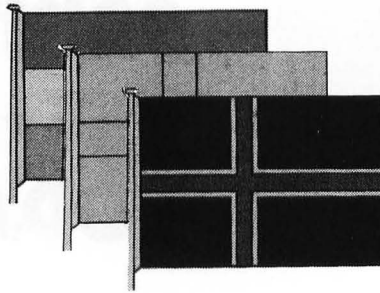
FLAGS

by John Cope and William Morris

"Flags" is an Apple program written in Applesoft and requiring at least 16K of RAM.

Does your sense of geography extend beyond the borders of the United States? If not, or even if it does, this program could prove interesting. It will display flags from different nations on the screen and give you the names of four countries, one of which is represented by the emblem on the screen. Your task is to correctly choose to which nation the flag pertains.

There are over 40 different flags in this program, which should



provide a considerable variety of choices. While some of the actual flags contain detailed emblems not shown on the screen, we feel that the program overcomes this particular limitation quite well due to both the educational value and the pure fun of competition. Enjoy!

String Variables

A\$: Flag to be identified.
 D\$: Reference list for comparisons to be identified.
 E\$: List of flags correctly identified.
 F\$: List of flags incorrectly identified.

Numerical Variables

C0, C1, C2: Color variables.
 NU: Number of flags to be identified.
 CG: Number of correct identifications.
 TS: Total score.
 R, R1: Random numbers.

```

0 REM   *** FLAGS ***
1 REM   *** (C) JOHN COPE & WILL
      IAN MORRIS ***

      Dimension reference list.

10 DIM D$(50): GOSUB 20000

      Set background color for graphic
      display and draw flag outline.

20 SC = 14: GOSUB 11000: GOSUB 21
      000: VLIN 5,39 AT 4: HLIN 5,
      38 AT 5: VLIN 5,30 AT 38: HLIN
      5,38 AT 30: IF FG THEN 70
25 HLIN 8,11 AT 7: VLIN 7,25 AT
      8: HLIN 8,10 AT 15: COLOR= 1
      1: VLIN 7,25 AT 13: HLIN 13,
      17 AT 25: COLOR= 4: VLIN 7,2
      5 AT 19: VLIN 7,25 AT 23: HLIN
      19,23 AT 7
26 HLIN 19,23 AT 15: COLOR= 6: VLIN
      7,25 AT 25: VLIN 17,25 AT 29
      : HLIN 25,29 AT 25: HLIN 27,
      29 AT 17: HLIN 25,29 AT 7: VLIN
      17,19 AT 27
27 COLOR= 13: HLIN 31,34 AT 7: HLIN
      31,34 AT 25: HLIN 32,34 AT 1
      5: VLIN 7,15 AT 31: VLIN 15,
      25 AT 34
    
```

If the variable SF=1 then the program skips over the anthem routine. The flag is set only if the instruction routine is called.

```

28 IF SF THEN 70
29 GOSUB 21000: GOSUB 21030

      The game paddle button is used to
      register an answer to the
      instruction question. If a 'yes' is
      entered, the program branches to
      21040.

30 PRINT "INSTRUCTIONS REQUIRED?"
      "
40 PRINT : INVERSE : HTAB 3: PRINT
      "PRESS GAME PADDLE BUTTON TO
      CHOOSE": NORMAL
50 FLASH : VTAB 21: HTAB 27: PRINT
      "YES":X = PEEK ( - 16286): IF
      X > 127 THEN 70
60 GOSUB 21090: VTAB 21: HTAB 27
      : CALL - 868: PRINT "NO": NORMAL
      :X = PEEK ( - 16286): IF X >
      127 THEN GOSUB 21040:SF = 1
      : GOTO 20
65 GOSUB 21090: GOTO 50
    
```

Sets the display for the text window. Game paddle (1) is used to set the number of flags to be displayed (NU). Line 80 continues the loop until the button is pressed.

```

70 HOME : GOSUB 21010: INVERSE :
      PRINT "NUMBER OF FLAGS TO B
      E IDENTIFIED IS ": NORMAL : PRINT
      "USE PADDLE (1) TO ADJUST TH
      IS TOTAL": HTAB 7: FLASH : PRINT
      "PRESS BUTTON TO CONTINUE"
    
```

```

75 VTAB 21: HTAB 37: CALL - 868
      : PRINT PDL (1)
80 NU = PDL (1): IF PEEK ( - 16
      286) < 128 THEN 75
85 DIM E$(NU),F$(NU): GOSUB 2101
      0
90 PR = PR + 1: IF PR = NU + 1 THEN
      8000

      Restores data pointer and chooses the
      flag to be indentified.

100 RESTORE :R1 = INT (( RND #1
      ) * 46) + 1)

      Reads through sound data and flag
      data until the flag to be
      identified is found.

105 FOR X = 1 TO 33: READ Z: NEXT

110 FOR Z = 1 TO R1: READ A$,C0,
      C1,C2: NEXT

      If the flag has vertical bars branch
      to 2000.

130 IF R1 < 18 THEN 2000

      Horizontal flag stripe routine.

1000 COLOR= C0: FOR Z = 6 TO 13:
      HLIN 5,37 AT Z: NEXT : COLOR=
      C1: FOR Z = 14 TO 21: HLIN 5
      ,37 AT Z: NEXT : COLOR= C2: FOR
      Z = 22 TO 29: HLIN 5,37 AT Z
      : NEXT
1040 IF R1 < 27 THEN GOSUB 3000
    
```

```

1050 IF R1 > 36 THEN GOSUB 4000
1990 GOTO 9000

```

Vertical flag stripe routine.

```

2000 COLOR= C0: FOR Z = 5 TO 15:
  VLIN 6,29 AT Z: NEXT : COLOR=
  C1: FOR Z = 16 TO 26: VLIN 6
  ,29 AT Z: NEXT : COLOR= C2: FOR
  Z = 27 TO 37: VLIN 6,29 AT Z
  : NEXT
2050 IF R1 < 8 THEN GOSUB 3000
2990 GOTO 9000

```

Routine to draw emblem.

```

3000 COLOR= 1: HLIN 18,24 AT 15:
  HLIN 18,24 AT 20: VLIN 16,2
  0 AT 18: RETURN

```

Routine to draw non-standard flags (Haiti, Japan, Malta, Spain and Portugal).

```

4000 ON (R1 - 36) GOTO 4010,4020
  ,4030,4040,4050,4060,4070,40
  80,4090,4100
4010 COLOR= 0: FOR Z = 5 TO 20: VLIN
  6,29 AT Z: NEXT : RETURN
4020 COLOR= 9: FOR Z = 13 TO 21:
  HLIN 18,24 AT Z: NEXT : COLOR=
  15: PLOT 18,13: PLOT 18,21: PLOT
  24,13: PLOT 24,21: RETURN
4030 COLOR= 9: FOR Z = 21 TO 37:
  VLIN 6,29 AT Z: NEXT : COLOR=
  5: VLIN 7,12 AT 9: HLIN 7,11
  AT 9: COLOR= 9: PLOT 9,9: RETURN
4040 COLOR= 15: FOR Z = 9 TO 12:
  VLIN 6,29 AT Z: NEXT : FOR
  Z = 16 TO 20: HLIN 5,37 AT Z
  : NEXT : COLOR= 9: GOTO 4055
4050 COLOR= 15
4055 FOR Z = 10 TO 11: VLIN 6,29
  AT Z: NEXT : FOR Z = 17 TO
  19: HLIN 5,37 AT Z: NEXT : RETURN
4060 COLOR= 7: VLIN 6,29 AT 15: HLIN
  5,37 AT 14: GOTO 4085
4070 COLOR= 13: GOTO 4085
4080 COLOR= 15
4085 FOR Z = 16 TO 18: VLIN 6,29
  AT Z: NEXT : FOR Z = 15 TO
  18: HLIN 5,37 AT Z: NEXT : IF
  A$ = "NORWAY" THEN COLOR= 3
  : VLIN 6,29 AT 17: HLIN 5,37
  AT 16
4086 RETURN

```

```

4090 COLOR= 13: FOR Z = 13 TO 23
  : HLIN 5,37 AT Z: NEXT : GOSUB
  3000: RETURN
4100 COLOR= 12: FOR Z = 5 TO 17:
  VLIN 6,29 AT Z: NEXT : COLOR=
  1: HLIN 15,20 AT 15: HLIN 15
  ,20 AT 20: VLIN 16,20 AT 15:
  RETURN

```

Sound and other data.

```

5000 DATA 173,48,192,136,208,5,2
  06,1,3,240,9,202,208,245,174
  ,0,3,76,2,3,96,81,118,96,80,
  121,160,96,160,81,160,60,255
5001 DATA MEXICO,12,15,9
5002 DATA BARBADOS,3,13,3
5004 DATA GUATEMALA,7,15,7
5006 DATA PERU,9,15,9
5008 DATA CAMEROON,12,9,13
5010 DATA RWANDA,9,13,12
5012 DATA SENEGAL,12,13,9
5014 DATA IRELAND,12,15,9
5016 DATA ITALY,12,15,9
5018 DATA ANDORRA,6,13,9
5020 DATA CHAD,3,13,9
5022 DATA GUINEA,9,13,12
5024 DATA IVORY COAST,9,15,12
5026 DATA MALI,12,13,9
5028 DATA NIGERIA,12,15,12
5030 DATA BELGIUM,0,13,9
5032 DATA FRANCE,3,15,9
5034 DATA INDIA,9,15,12
5036 DATA EGYPT,9,15,0
5038 DATA BOLIVIA,9,13,12
5040 DATA NIGER,11,15,12
5042 DATA VENEZUELA,13,3,9
5044 DATA BULGARIA,15,12,9
5046 DATA EAST GERMANY,0,9,13
5048 DATA YUGOSLAVIA,3,15,9
5050 DATA PARAGUAY,9,15,7
5052 DATA LUXEMBOURG,9,15,7
5054 DATA NETHERLANDS,9,15,7
5056 DATA WEST GERMANY,0,9,13
5058 DATA ETHIOPIA,12,13,9
5060 DATA GABON,12,13,7
5062 DATA SIERRA LEONE,12,15,7
5064 DATA SUDAN,3,13,12
5066 DATA AUSTRIA,9,15,9
5068 DATA UPPER VOLTA,0,15,9
5070 DATA EL SALVADOR,7,15,7
5080 DATA HAITI,1,1,1
5090 DATA JAPAN,15,15,15
5100 DATA MALTA,15,15,15
5110 DATA ICELAND,3,3,3
5120 DATA DENMARK,9,9,9
5130 DATA FINLAND,15,15,15
5140 DATA SWEDEN,7,7,7
5150 DATA NORWAY,9,9,9
5160 DATA SPAIN,9,9,9
5170 DATA PORTUGAL,9,9,9

```

Sets the top of the screen window, checks to see if the screen limit is exceeded and toggles the speaker.

```

8000 TEXT : HOME : HTAB 15: INVERSE
  : GOSUB 21010: PRINT "GAME S
  UMMARY": NORMAL : POKE 34,4:
  IF V = 0 THEN 8040
8005 VTAB 5: PRINT "YOU CORRECTL
  Y IDENTIFIED THE FOLLOWING":
  PRINT "EMBLEMS"
8010 VTAB 10: FOR X = 1 TO V: PRINT
  X;" "E$(X): IF PEEK (37) <
  17 THEN 8030
8020 GOSUB 21110: HOME : NORMAL
8030 NEXT : GOSUB 21110
8040 FOR X = 1 TO 150: NEXT : IF
  W = 0 THEN 8090
8045 HOME : PRINT "YOU DID NOT I
  DENTIFY THE FOLLOWING": PRINT
  "EMBLEMS": GOSUB 21020
8050 VTAB 10: FOR Z = 1 TO W: PRINT
  Z;" "F$(Z): IF PEEK (37) <
  17 THEN 8080
8060 GOSUB 21110: HOME
8080 NEXT : GOSUB 21110
8090 POKE 34,0: HOME : VTAB 12: PRINT
  "PRESS BUTTON FOR ANOTHER SE
  QUENCE": FOR Z = 1 TO 9: X =
  PEEK ( - 16336): FOR A = 1 TO
  75: NEXT : NEXT : GOSUB 2113
  0: RUN
8100 END

```

Main game loop section. A common centering routine is used in lines 9050 and 9200 with the paddle button being used to register input.

```

9000 IF R > 32 THEN RESTORE : FOR
  X = 1 TO 33: READ Z: NEXT
9010 D$(1) = A$: FOR X = 2 TO 4
9015 Z = INT (( RND (1) * 152) /
  4 + 1)
9018 IF D$(Z) = D$(X - 1) THEN 9
  015
9020 D$(X) = D$(Z): NEXT
9030 HOME : INVERSE : HTAB 6: PRINT
  "WHAT COUNTRY FLIES THIS FLA
  G?": PRINT : PRINT ">>>>>>>>
  >>>>": HTAB 28: PRINT "<<<<
  <<<<<<<<<<": NORMAL : PRINT
  " PRESS GAME PADDLE BUTTON
  TO CHOOSE";
9040 R = INT ( RND (1) * 4 + 1)
9050 HTAB (20 - ( LEN (D$(R)) /
  2)): VTAB 23: PRINT D$(R): GOSUB
  21090: VTAB 23: HTAB 14: PRINT
  " "X = PEEK ( -
  16286): IF X > 127 THEN 9070

```

continued on next page

continued from previous page

```

9060 R = R + 1: IF R > 4 THEN R =
1
9065 GOTO 9050
9070 TS = TS + 1: FLASH : IF D$(R
) = A$ THEN CG = CG + 1: GOSUB
12000: VTAB 22: HTAB 16: PRINT
"CORRECT!": GOSUB 21010: GOTO
9220
9200 VTAB 22: HTAB 16: PRINT "IN
CORRECT": GOSUB 13000: GOSUB
21020: VTAB 22: HTAB 13: PRINT
"CORRECT ANSWER": VTAB 23: HTAB
(20 - ( LEN (A$) / 2)): NORMAL
: PRINT A$
9220 GOSUB 21090: GOSUB 21090: VTAB
22: FLASH : PRINT "CORRECT "
;CG;: CALL - 868;: HTAB 25:
PRINT "TOTAL GUESSES ";TS: NORMAL
: GOSUB 21090: NORMAL : GOTO 90

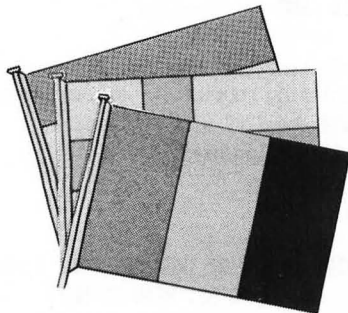
```

Machine language routine to set background color.

```

11000 REM *** SCREEN COLOR
11020 POKE 768,216: POKE 769,160
: POKE 770,0: POKE 771,76: POKE
772,44: POKE 773,254
11050 GR
11100 POKE 1024,SC + SC * 16: POKE
60,0: POKE 61,4: POKE 62,254
: POKE 63,7: POKE 66,1: POKE
67,4: CALL 768
11110 HOME : RETURN

```



Increments list of correct identifications.

```
12000 V = V + 1: E$(V) = A$: RETURN
```

Increments list of incorrect identifications.

```
13000 W = W + 1: F$(W) = D$(R): RETURN
```

Establishes reference list D\$ to be used in main game loops.

```

20000 FOR X = 1 TO 33: READ Z: NEXT
: FOR X = 1 TO 46: READ B$,C
0,C1,C2:D$(X) = B$: NEXT : RESTORE
: RETURN

```

Initialize sound routine.

```

21000 RESTORE : FOR Z = 770 TO 7
90: READ X: POKE Z,X: NEXT :
RETURN
21010 FOR X = 100 TO 50 STEP -
2: POKE 768,X: POKE 769,9: CALL
770: NEXT : RETURN
21020 FOR X = 25 TO 225 STEP 5: POKE
768,X: POKE 769,9: CALL 770:
NEXT : RETURN
21030 FOR X = 1 TO 6: READ Z,Y: POKE
768,Z: POKE 769,Y: CALL 770:
NEXT : RETURN

```

Instructions.

```

21040 TEXT : HOME : HTAB 12: PRINT
"WELCOME TO ";; INVERSE : PRINT
"FLAGS": NORMAL
21050 VTAB 5: PRINT "UPON ENTERI
NG THE NUMBER OF NATIONAL": PRINT
"EMBLEMS YOU WISH TO IDENTIF
Y, A FLAG": PRINT "WILL BE D
ISPLAYED ON THE SCREEN."
21060 PRINT "YOU WILL BE PRESENT
ED WITH 4 POSSIBLE": PRINT "
CHOICES. PRESSING THE PADDLE
BUTTON": PRINT "WILL INDICA
TE YOUR CHOICE. AFTER AN": PRINT
"INCORRECT GUESS, THE EMBLEM
WILL BE"
21070 PRINT "IDENTIFIED FOR YOU
BEFORE PROCEEDING": PRINT "T
O THE NEXT FLAG.": PRINT : PRINT
"A STANDARD FORM OF CREST WI
LL BE": PRINT "PLACED IN THE
CENTER OF ALL EMBLEMS": PRINT
"WITH THAT FEATURE": GOSUB 2
1110
21080 P = 1: RETURN
21090 FOR X = 1 TO 1500: NEXT : RETURN
21100 FOR Z = 1 TO 250: NEXT : RETURN

```

Paddle button read routine. This subroutine is accessed by several sections which require an input from the button to continue.

```

21110 VTAB ( PEEK (37) + 6)
21120 FLASH : PRINT "PRESS GAME
PADDLE BUTTON TO CONTINUE": NORMAL
21130 IF PEEK ( - 16286) < 128 THEN
21130
21140 POKE - 16286,0
21150 RETURN

```

continued from page 59

```

16095 NEXT T
16100 PRINT
16300 RETURN

```



Apple statements.

```

17000 PRINT :PNT$=" THANK YOU FOR THE
EXCITING GAME,": GOSUB 16020
17010 PNT$="LET'S PLAY AGAIN SOMETIME.
": GOSUB 16020
17030 RETURN
17100 PRINT :PNT$="HA HA HA, ADMIT IT,
I PLAY A PRETTY ": GOSUB 16020
17110 PNT$="GOOD GAME OF SKUNK. I'M HA
RD TO BEAT,": GOSUB 16020
17140 RETURN
17200 PRINT :PNT$="I BEAT YOU, HOW ABO
UT ANOTHER": GOSUB 16020
17210 PNT$="GAME? YOU'RE NOT A BAD PL
AYER,": GOSUB 16020
17240 RETURN
17300 PRINT :PNT$="HOW ABOUT ANOTHER R
OUND. I ENJOYED ": GOSUB
16020
17310 PNT$="THAT GAME TREMENDOUSLY.
": GOSUB 16020
17330 RETURN

```

Set-up screen.

```

21000 COLOR=4
21010 FOR T=0 TO 39
21020 VLIN 0,39 AT T
21030 NEXT T
21050 VTAB 21: TAB 5-( LEN(PLR1$)
/2): PRINT PLR1$
21060 VTAB 21: TAB 15-( LEN(PLR2$
)/2): PRINT PLR2$
21070 IF PLR=2 THEN GOTO 21150
21080 VTAB 21: TAB 25-( LEN(PLR3$
)/2): PRINT PLR3$
21090 IF PLR=3 THEN GOTO 21150
21100 VTAB 21: TAB 35-( LEN(PLR4$
)/2): PRINT PLR4$
21150 VTAB 23: TAB 4: PRINT "GAME="
;SC
21180 RETURN

```

Poke sound routine.

```

22000 POKE 0,PP: POKE 1,DD: CALL
2: RETURN
22100 POKE 2,173: POKE 3,48: POKE
4,192: POKE 5,136: POKE 6,208
: POKE 7,4: POKE 8,198: POKE
9,1: POKE 10,240
22150 POKE 11,8: POKE 12,202: POKE
13,208: POKE 14,246: POKE 15
,166: POKE 16,0: POKE 17,76
: POKE 18,2: POKE 19,0: POKE
20,96: RETURN

```

WORD PUZZLE

```

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

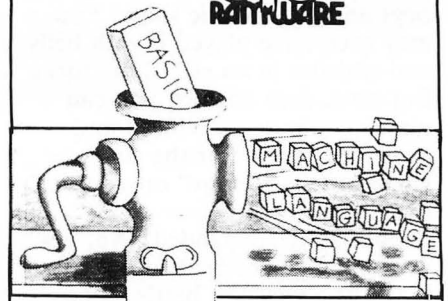
WORD LIST

CHIP	BASIC	KEYBOARD
RESOLUTION	PASCAL	OUTPUT
VECTOR	ASSEMBLER	MICROPROCESSOR
FLOWCHART	COMPILER	SOFTSIDE
CONCATENATION	JOYSTICK	BYTE
STRING	VARIABLE	NYBBLE
COMMAND	ADDRESS	BINARY HEXADECIMAL
EXECUTION	DEBUG	OCTAL
RANDOM	SUBROUTINE	DATA
ERROR	FORMAT	PRINTOUT
GRAPHICS	PARAMETER	INTERFACE
KEYWORD	TERMINAL	DISK
INITIALIZATION	PERIPHERAL	MONITOR

Solution in next month's issue.

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 \$19.95
32K, S-80, Disk \$24.95

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

BATTLE AT SEA

by Jon Voskuil

“Battle at Sea” is written in Applesoft and requires 16K of RAM.

You may have played it with pencil and paper. You may have played it with red and white plastic pegs and grey plastic ships. You may even have played it with bells and whistles in an electronic form. But now, dear friends, you can play a computer-enhanced and expanded version of the classic game of “Battleship” on your very own Apple.

Enhanced? Expanded? Ah, yes, you heard correctly. It used to be that this classic sea battle game was just for two players. Now, you can play with as many as six. And one of the fleet commanders can be the Apple itself, as it strives to prove that computers will indeed take over the world — beginning not in space, as some would have us believe, but in the mid-Atlantic.

Such introductory hoopla over with, let’s look at the play of the game and the operation of the program.

PLAYING THE GAME

After typing in the code and commanding the program to RUN (and then ridding it of all the bugs which infest such keyboard-entered entities), you will be greeted with the title page, along with a tune reminiscent of moonlit waterfronts, tall sailing ships, sinking destroyers, and burning oil freighters. The instructions will be displayed, and then all the players enter their names. (These are truncated to eight letters, so if your name is Michelangelo you’d be better off to enter “Mike” than to end up with “Michaela”.) You enter “Apple” as one of the names if you want the computer to play along.

The next step is for everyone to position their ships on their 10-by-10 grid. This is done by one player at a time, while the others turn their backs. You just enter the endpoint coordinates for each ship. For example, a submarine (three units long) could be placed from

A2 to A4 vertically, or maybe from C5 to E7 diagonally. The Apple also positions its ships in turn, hiding behind the TV screen as it does so. Incidentally, even though part of the Apple knows where all the ships are for all the players, its left RAM doesn’t know what its right RAM is doing when it comes to aiming shots at its opponents.

Following these preliminaries, the battle gets under way. Each player, in turn, first presses the paddle button to show that he’s ready. The reason for this is that with more than two human players, the paddles must be shared. So this step just lets the computer know who’s who. The player then selects (via the spacebar) the grid upon which to wreak destruction, and then aims the sights (via either paddle) and fires (using the paddle button). The wailing of a falling bomb is followed either by an ineffectual splash or a monitor-shattering explosion, with an appropriate printed message.

Play continues from player to player until one emerges victorious. Players whose fleets are entirely destroyed are automatically removed from play, and must watch from their floating debris while the remaining fleets battle it out. The final winner is rewarded with a little melody reminiscent of moonlit waterfronts, oil freighters, etc., and (as you might expect) the little blinking cursor finally returns to await your next command. (This is the way the world ends...?)

PROGRAM NOTES

The program is set up so that a main control section (lines 100-340) calls various subroutines which do the different tasks required. These are all appropriately labelled in the line listing which follows.

A considerable amount of the coding is involved with positioning the cursor (the gun sight) at the proper place on the proper grid. This can get a little involved, especially since different grid arrangements are used for different numbers of players. (The display

looks so much nicer when it’s arranged symmetrically.) The upper-left-corner coordinates for each of the grids are determined in lines 3000-3070, which are then stored in the H() and V() arrays. Positioning the cursor within a given grid, then, simply involves adding these base values to the “relative” cursor coordinates determined by the paddle position.

The Apple’s move strategy (subroutine 5000) is the single most involved part of the program. I hope I’m not giving away any military secrets by telling you that it will choose to fire upon the grid which has the least number of hits. It will then determine if there are any wounded (hit but not sunk) ships, and if so will fire upon one of them. If the ship already has multiple hits, then it will try another shot along the same line. If the ship has only one hit, then it will determine the placement of its bomb by finding the longest stretch of uninterrupted sea and firing on that side of the first hit. You clever strategists out there ought to be able to figure out a way to use this military intelligence against your friends when you’re all playing with the Apple.

The Machine Language sound generating routine first appeared in **SoftSide** last August. It makes use of the Apple’s ampersand (&) to jump to the routine and to plug in the pitch and duration of the tone that you want. The advantage of this method, over the more common one that requires a couple of POKES and a CALL, is speed. You get less of a pause or “click” between notes, and it seems to be the best way to make melodies — short of a complete Machine Language routine. For more on the ampersand, see the Applesoft and Apple II reference manuals, and the April, 1980, Apple Edition of **SoftSide**.

One additional note. Occasionally the Apple will really get clever and set up its ships so that two of them overlap diagonally. (Computers tend to have an unrealistic world-view.) Just thought I’d better tell you that before you found it out yourself.

VARIABLES

A\$: Used to draw grids.
 APPLE: Apple's player number.
 AX,AY: Absolute values of X, Y.
 CHOICE: Type of ship which Apple chooses to fire upon.
 DEAD: Number of dead (destroyed) fleets.
 G: Grid number.
 G%(x,y,p): Array giving ship status at coordinates x, y on player p's grid, as follows:
 5 - aircraft carrier
 4 - battleship
 3 - submarine
 2 - cruiser
 1 - destroyer
 0 - open sea
 -1 - hit destroyer
 -2 - hit cruiser
 -3 - hit submarine
 -4 - hit battleship
 -5 - hit aircraft carrier
 -6 - hit open sea (i.e., a miss)

H,HH,HO,OH: Used in horizontal positioning of cursor.
 H(g): Horizontal location of grid g.
 HITS: Number of hits sustained.
 HITS(s,p): Hits sustained by player p's ship type s.
 I,J: General loop counters.
 KEY: ASCII value of key pressed.
 LEAST: Used to find grid with least hits.
 MAX: Used to find direction of maximum open sea from hit ship.
 N: General-purpose temporary variable.
 NP: Number of players.
 OUT(i): Array containing numbers of players whose fleets are destroyed.
 P: Player number.
 P1,P2: Values for sound routines.
 PL\$(p): Player p's name.
 PN: Paddle value.
 POK\$: Used to contain data to poke into memory.

R, R1, R2, R3, R4: Random numbers.
 R\$: String used to read data.
 S: Ship type (see G%).
 S\$(s): Initial letter of ship type s.
 SIZ(s): Size of ship type s.
 SS\$(s): Name of ship type s.
 T1, T2: Values for sound routines.
 V, VV, VO, OV: Used in vertical positioning of cursor.
 V(g): Vertical location of grid g.
 X: Used for x-coordinate calculations.
 X\$: X-coordinate input variable.
 X1, X2, XD, XI, XX: Used for x-coordinate inputs and calculations.
 X(i): X-coordinates of hits on a ship, used in calculating Apple's move.
 Y, Y\$, Y1, Y2, YD, YI, YY, Y(i): All correspond to respective x-variables.
 Z: Time-delay loop counter.
 Z\$: General input variable.

```
10 HOME : GOTO 100
```

Subroutine to wait for keypress.

```
20 VTAB 24: PRINT " >>> PRESS
  ANY KEY TO CONTINUE <<<";
30 IF PEEK ( - 16384) < 128 THEN
  30
40 POKE - 16368,0: RETURN
```

Initialization.

```
100 DIM GX(9,9,6),HITS(5,6),SIZ(
  5),S$(6),PL$(6),V(6),H(6),SS
  $(5),OUT(4),X(5),Y(5)
110 SS$(1) = "DESTROYER " : SS$(2)
  = "CRUISER " : SS$(3) = "S
  UBMARINE " : SS$(4) = "BATTLE
  SHIP " : SS$(5) = "AIR,CARRIER
  "
120 S$(1) = "D" : S$(2) = "C" : S$(3)
  = "S" : S$(4) = "B" : S$(5) = "
  A" : S$(6) = " " : S$(0) = "+"
130 SIZ(1) = 2 : SIZ(2) = 3 : SIZ(3) =
  3 : SIZ(4) = 4 : SIZ(5) = 5
140 A$ = "+++++++"
150 GOSUB 7700
160 GOSUB 1000
170 GOSUB 2000
180 HOME : VTAB 8: FLASH : PRINT
  SPC( 40) : PRINT " ALL H
  ANDS TO BATTLE STATIONS!!
  " : PRINT SPC( 40) : NORMAL
190 GOSUB 7600 : GOSUB 7600
200 HOME
```

```
210 GOSUB 3000
220 P = 1
```

Main control loop.

```
230 IF P = (APPLE) THEN GOSUB 5
  000: GOTO 250
240 GOSUB 4000
250 GOSUB 6000
260 P = P + 1: IF P > NP THEN P =
  1
270 IF DEAD = 0 THEN 230
280 FOR I = 0 TO DEAD - 1: IF OU
  T(I) = P THEN P = P + 1: IF
  P > NP THEN P = 1
290 NEXT I
300 GOTO 230
```

End of game.

```
310 VTAB 22: HTAB 1: PRINT SPC(
  119) : VTAB 23: HTAB 7
320 FLASH : PRINT PL$(P) : " IS TH
  E VICTOR!!" : NORMAL
330 GOSUB 7500
340 VTAB 1: END
```

Print instructions.

```
1000 HOME : VTAB 8
1010 PRINT TAB( 8) : "B A T T L E
  A T S E A"
1020 VTAB 12: PRINT TAB( 13) : "B
  Y JON VOSKUIJL"
1030 GOSUB 7500
1040 FOR Z = 1 TO 1000: NEXT Z
```

```
1050 HOME : VTAB 3
1060 PRINT " EACH PLAYER HAS
  A 10-BY-10 GRID ON": PRINT
1070 PRINT "WHICH TO PLACE FIVE
  SHIPS: A DESTROYER,"
1080 PRINT "A CRUISER, A SUBMARI
```

continued on next page



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.

continued from previous page

```
ME, A BATTLESHIP,"; PRINT
1090 PRINT "AND AN AIRCRAFT CARR
IER,"; PRINT
1100 PRINT " PLAYERS TAKE TUR
NS TRYING TO BLAST"; PRINT
1110 PRINT "EACH OTHER'S SHIPS O
FF THE MAP, FROM"; PRINT
1120 PRINT "TWO TO FIVE HITS ARE
NEEDED TO SINK A"; PRINT
1130 PRINT "SHIP, DEPENDING ON I
TS TYPE, THE WINNER"
1140 PRINT "IS THE PLAYER WHO SU
RVIVES THE LONGEST,"
1150 GOSUB 20
1160 HOME : VTAB 3
1170 PRINT " UP TO SIX PLAYER
S MAY PARTICIPATE,"; PRINT
1180 PRINT "ENTERING 'APPLE' FOR
ONE NAME WILL MAKE"; PRINT

1190 PRINT "HE ONE OF THE PLAYER
S, PRESS 'RETURN'"; PRINT
1200 PRINT "BY ITSELF WHEN THERE
ARE NO MORE NAMES,"
1210 P = 1; APPLE = 0; PRINT : PRINT
```

Input players' names.

```
1220 PRINT "PLAYER #"; P; : INPUT
"; : PL$(P)
1230 IF PL$(P) < > "" THEN 1260

1240 IF P < 3 THEN 1220
1250 P = P - 1; GOTO 1310
1260 IF LEFT$(PL$(P),5) < > "
APPLE" THEN 1290
1270 IF APPLE > 0 THEN 1220
1280 APPLE = P
1290 IF LEN(PL$(P)) > 8 THEN P
L$(P) = LEFT$(PL$(P),8)
1300 IF P < 6 THEN P = P + 1; GOTO
1220
1310 NP = P; HOME : RETURN
```

Subroutine to set up ships on grids.

```
2000 FOR P = 1 TO NP; HOME
2010 IF APPLE = P THEN GOSUB 25
00; GOTO 2460
2020 PRINT " DES-
BATTLE- AIR, TROYER
CRUISER SUB SHIP CAR
RIER"
2030 VTAB 4; INVERSE : HTAB 3; PRINT
"DD"; HTAB 12; PRINT "CCC";
: HTAB 20; PRINT "SSS"; HTAB
26; PRINT "BBBB"; HTAB 35; PRINT
"AAAA"; NORMAL
```

```
2040 H = 20; V = 8; GOSUB 3500
2050 VTAB 7; HTAB 20; PRINT "ABC
DEFGHIJ";
2060 FOR I = 0 TO 9; VTAB 8 + I;
HTAB 19; PRINT I; : HTAB 30;
PRINT I; : NEXT I
2070 VTAB 18; HTAB 20; PRINT "AB
CDEFGHIJ";
2080 VTAB 21; HTAB 1; PRINT PL$(
P); ", IT'S YOUR TURN TO SET
UP YOUR"
2090 PRINT "SHIPS, TYPE IN TWO
LETTER/NUMBER PAIRS TO POSIT
ION THE END POINTS OF EACH S
HIP IN TURN, (OTHER PLAYERS
, TURN AWAY!!!)";
2100 FOR Z = 1 TO 1500; NEXT Z
2110 FOR S = 1 TO 5
2120 VTAB 8; HTAB 3; PRINT SS$(S
);
2130 VTAB 9; HTAB 2; PRINT "____
____";
2140 VTAB 11; HTAB 3; PRINT "FRO
M:";
2150 VTAB 11; HTAB 9; GET X%;X1 =
ASC(X%) - 65
2160 IF X1 < 0 OR X1 > 9 THEN 21
50
2170 VTAB 11; HTAB 9; PRINT X%;
2180 VTAB 11; HTAB 11; GET Y%;Y1
= ASC(Y%) - 48
2190 IF Y1 = - 40 THEN 2150; REM
BACKSPACE
2200 IF Y1 < 0 OR Y1 > 9 THEN 21
80
2210 VTAB 11; HTAB 11; PRINT Y%;
2220 VTAB 13; HTAB 3; PRINT "TO:
";
2230 VTAB 13; HTAB 9; GET X%;X2 =
ASC(X%) - 65
2240 IF X2 = - 57 THEN 2180
2250 IF X2 < 0 OR X2 > 9 THEN 22
30
2260 VTAB 13; HTAB 9; PRINT X%;
2270 VTAB 13; HTAB 11; GET Y%;Y2
= ASC(Y%) - 48
2280 IF Y2 = - 40 THEN 2230
2290 IF Y2 < 0 OR Y2 > 9 THEN 22
70
2300 VTAB 13; HTAB 11; PRINT Y%;

2310 VTAB 15; HTAB 1; PRINT "COR
RECT? (Y/N)"; GET Z%; IF Z%
= "N" THEN 2140
2320 IF Z% < > "Y" THEN 2310
```

Check validity of input.

```
2330 XD = X2 - X1; YD = Y2 - Y1; X =
ABS(XD); Y = ABS(YD); IF
X = 0 AND Y = SIZ(S) - 1 THEN
2380
```

```
2340 IF Y = 0 AND X = SIZ(S) - 1
THEN 2380
2350 IF Y = X AND X = SIZ(S) - 1
THEN 2380
2360 VTAB 15; HTAB 1; FLASH : PRINT
"> TRY AGAIN! <"; : NORMAL
2370 FOR Z = 1 TO 2000; NEXT Z; VTAB
15; HTAB 1; PRINT SPC(14);
GOTO 2140
2380 IF GZ(X1,Y1,P) < > 0 OR GZ
(X2,Y2,P) < > 0 THEN 2360
2390 IF SIZ(S) = 2 THEN 2420
2400 FOR I = 2 TO S - 1; IF GZ(X
1 + (I - 1) * XD / (SIZ(S) -
1),Y1 + (I - 1) * YD / (SIZ(
S) - 1),P) < > 0 THEN 2360
2410 NEXT I
```

Draw ship on grid and place it into memory.

```
2420 GZ(X1,Y1,P) = S; GZ(X2,Y2,P) =
S; VTAB Y1 + 8; HTAB X1 + 20
: PRINT S$(S); : VTAB Y2 + 8;
HTAB X2 + 20; PRINT S$(S); :
IF SIZ(S) = 2 THEN 2440
2430 FOR I = 2 TO SIZ(S) - 1; XX =
X1 + (I - 1) * XD / (SIZ(S) -
1); YY = Y1 + (I - 1) * YD /
(SIZ(S) - 1); GZ(XX,YY,P) = S
: VTAB YY + 8; HTAB XX + 20;
PRINT S$(S); : NEXT I
2440 VTAB 11; HTAB 3; PRINT SPC(
10); : VTAB 13; HTAB 3; PRINT
SPC(10); : VTAB 15; HTAB 1;
PRINT SPC(14);
2450 NEXT S
2460 FOR Z = 1 TO 2500; NEXT Z; NEXT
P
2470 RETURN
```

Subroutine for setting up Apple's ships.

```
2500 HOME : VTAB 15; PRINT " APP
LE IS NOW SETTING UP ITS SHI
PS, . ."
2510 FOR S = 1 TO 5
2520 X = INT ( RND (1) * 10); Y =
INT ( RND (1) * 10)
2530 XI = INT ( RND (1) * 3) - 1
; YI = INT ( RND (1) * 3) -
1; IF XI = 0 AND YI = 0 THEN
2530
2540 FOR I = 1 TO SIZ(S); XX = X +
I * XI; YY = Y + I * YI
2550 IF XX < 0 OR XX > 9 OR YY <
0 OR YY > 9 THEN 2520
2560 IF GZ(XX,YY,P) > 0 THEN 252
0
```

```

2570 NEXT I
2580 FOR I = 1 TO SIZ(S):GZ(X +
  XI * I,Y + YI * I,P) = S: NEXT
  I
2590 NEXT S
2600 FOR Z = 1 TO 1000: NEXT Z
2610 RETURN

```

Subroutine to position and draw the proper number of grids.

```

3000 FOR I = 1 TO 4:V(I) = 1:H(I
  ) = 2: NEXT I
3010 IF NP < 4 THEN V(1) = 6:V(2
  ) = 6:V(3) = 6: GOTO 3040
3020 V(4) = 12: IF NP = 4 THEN V(
  3) = 12: GOTO 3040
3030 V(5) = 12:V(6) = 12
3040 IF NP = 2 OR NP = 4 THEN H(
  2) = 30:H(4) = 30: GOTO 3080

3050 IF NP = 3 THEN H(2) = 16:H(
  3) = 30: GOTO 3080
3060 H(2) = 16:H(3) = 30: IF NP =
  5 THEN H(5) = 30: GOTO 3080
3070 H(5) = 16:H(6) = 30
3080 FOR P = 1 TO NP
3090 H = H(P):V = V(P): GOSUB 350
  0
3100 NEXT P: RETURN

```

Subroutine to draw a single grid.

```

3500 INVERSE : FOR I = 0 TO 9
3510 VTAB V + I: HTAB H: PRINT A
  $;
3520 NEXT I
3530 NORMAL : FOR J = 1 TO LEN
  (PL$(P))
3540 VTAB V + J: HTAB H - 1
3550 PRINT MID$( PL$(P),J,1);
3560 NEXT J
3570 RETURN

```

Subroutine to input a player's move.

Which paddle is player using?

```

4000 VTAB 23: HTAB 1: PRINT PL$(
  P);", PRESS YOUR PADDLE BUTT
  ON": PRINT "WHEN YOU'RE READ
  Y,";
4010 IF PEEK ( - 16287) > 127 THEN
  N = 0: GOTO 4040
4020 IF PEEK ( - 16286) > 127 THEN
  N = 1: GOTO 4040
4030 GOTO 4010
4040 IF PEEK ( - 16287 + N) > 1
  27 THEN 4040

```

Which grid does player want to fire upon?

```

4050 VTAB 23: HTAB 1: PRINT SPC(
  79);: VTAB 22: HTAB 1
4060 IF NP > 2 THEN PRINT "PRES
  S SPACE BAR TO SELECT GRID."

4070 PRINT : PRINT "PRESS PADDLE
  BUTTON TO FIRE,";

4080 G = 1
4090 IF G = P THEN G = G + 1: IF
  G > NP THEN G = 1
4100 IF DEAD = 0 THEN 4140
4110 FOR I = 0 TO DEAD - 1: IF G
  < > OUT(I) THEN NEXT I: GOTO
  4140
4120 G = G + 1: IF G > NP THEN G =
  1
4130 GOTO 4090

```

Position cursor on chosen grid using paddle input.

```

4140 HO = H(G) + 5:VO = V(G) + 5:
  OH = 5:OV = 5
4150 PN = PDL (N) / 2: IF PN > 9
  9 THEN PN = 99
4160 WV = INT (PN / 10):HH = PN -
  WV * 10:V = WV + V(G):H = HH
  + H(G)
4170 IF H = HO AND V = VO THEN 4
  240
4180 INVERSE : VTAB VO: HTAB HO:
  S = GZ(OH,OV,G): IF S > - 1
  THEN PRINT "+": GOTO 4210

4190 IF S > - 6 THEN NORMAL
4200 PRINT S$( ABS (S));
4210 FLASH : VTAB V: HTAB H:S =
  GZ(HH,WV,G): IF S > - 1 THEN
  PRINT "+": GOTO 4230
4220 PRINT S$( ABS (S));
4230 R = RND (1) * 100 + 100: &
  TR,B

```

Has the space bar been pressed?

```

4240 KEY = PEEK ( - 16384): IF K
  EY < 128 THEN 4310
4250 POKE - 16368,0: IF KEY < >
  160 THEN 4310
4260 VTAB V: HTAB H: INVERSE :S =
  GZ(HH,WV,G): IF S > - 1 THEN
  PRINT "+": GOTO 4290
4270 IF S > - 6 THEN NORMAL
4280 PRINT S$( ABS (S));
4290 G = G + 1: IF G > NP THEN G =
  1

```

4300 GOTO 4090

Has the paddle button been pressed?

```

4310 IF PEEK ( - 16287 + N) < 1
  28 THEN HO = H:VO = V:OH = H
  0 - H(G):OV = VO - V(G): GOTO
  4150
4320 IF PEEK ( - 16287 + N) > 1
  27 THEN 4320
4330 S = GZ(HH,WV,G): IF S < 0 THEN
  PRINT """: GOTO 4150
4340 NORMAL : VTAB 22: HTAB 1: PRINT
  SPC( 119);
4350 RETURN

```

Subroutine to execute Apple's move. Choose a grid.

```

5000 VTAB 22: HTAB 1: PRINT "APP
  LE'S TURN"
5010 IF NP = 2 THEN G = 3 - APPL
  E: GOTO 5070
5020 LEAST = 17
5030 FOR I = 1 TO NP: IF I = (AP
  PLE) THEN 5060
5040 HITS = 0: FOR J = 1 TO 5:HIT
  S = HITS + HITS(J,I): NEXT J

5050 IF HITS < LEAST THEN LEAST =
  HITS:G = I
5060 NEXT I

```

Are there any ships that are hit but not sunk?

```

5070 CHOICE = 0: FOR I = 1 TO 5
5080 IF HITS(I,G) > 0 AND HITS(I
  ,G) < SIZ(I) AND HITS(I,G) >
  CHOICE THEN CHOICE = I:HITS =
  HITS(I,G)
5090 NEXT I

```

5100 IF CHOICE = 0 THEN 5270

Calculate attack on wounded ship.

```

5110 J = 1: FOR Y = 0 TO 9: FOR X
  = 0 TO 9: IF GZ(X,Y,G) = -
  CHOICE THEN X(J) = X:Y(J) =
  Y:J = J + 1
5120 NEXT X,Y: IF HITS < > 1 THEN
  5190

```

continued on next page

continued from previous page

```
5130 MAX = 0: FOR YI = - 1 TO 1:
  FOR XI = - 1 TO 1: I = 1: IF
  XI = 0 AND YI = 0 THEN 5170
5140 XX = X(1) + I * XI: YY = Y(1)
  + I * YI: IF XX < 0 OR XX >
  9 OR YY < 0 OR YY > 9 THEN 5
  160
5150 IF GZ(XX,YY,G) > - 1 THEN
  I = I + 1: GOTO 5140
5160 IF I > MAX THEN MAX = I: XD =
  XI: YD = YI
5170 NEXT XI, YI
5180 X = X(1) + XD: Y = Y(1) + YD:
  GOTO 5330
5190 IF ABS (X(HITS) - X(1)) =
  HITS - 1 OR ABS (Y(HITS) -
  Y(1)) = HITS - 1 THEN 5230
5200 FOR I = 2 TO HITS: AX = ABS
  (X(I) - X(I - 1)): AY = ABS
  (Y(I) - Y(I - 1)): IF AX = 1
  OR AY = 1 THEN NEXT I
5210 X = INT ((X(I) + X(I - 1)) /
  2 + .5): Y = INT ((Y(I) + Y(
  I - 1)) / 2 + .5)
5220 GOTO 5330
```

```
5230 X = X(1) - (X(HITS) - X(1)) /
  (HITS - 1): Y = Y(1) - (Y(HIT
  S) - Y(1)) / (HITS - 1)
5240 IF X > - 1 AND X < 10 AND
  Y > - 1 AND Y < 10 THEN IF
  GZ(X,Y,G) > - 1 THEN 5330
5250 X = X(HITS) + (X(HITS) - X(1
  )) / (HITS - 1): Y = Y(HITS) +
  (Y(HITS) - Y(1)) / (HITS - 1)
5260 GOTO 5330
```

If no wounded ship, then choose
random location.

```
5270 X = INT ( RND (1) * 10): Y =
  INT ( RND (1) * 10)
5280 IF GZ(X,Y,G) < 0 THEN 5270
5290 N = 0: FOR YI = - 1 TO 1: FOR
  XI = - 1 TO 1: IF XI = 0 AND
  YI = 0 THEN 5320
5300 XX = X + XI: YY = Y + YI: IF
  XX < 0 OR XX > 9 OR YY < 0 OR
  YY > 9 THEN 5320
5310 IF GZ(XX,YY,G) > - 1 THEN
  N = N + 1
5320 NEXT XI, YI: IF N < 4 THEN 5
  270
```

Position the cursor and fire.

```
5330 FOR I = 1 TO RND (1) * 8 +
  4
5335 H = RND (1) * 10: V = RND (
  1) * 10
5340 IF GZ(H,V,G) < 0 THEN 5335
5350 HTAB H + H(G): VTAB V + V(G
  ): FLASH : PRINT "+";
5360 R = RND (1) * 100 + 100: &
  TR,8
5370 FOR Z = 1 TO RND (1) * 500
  : NEXT Z
5380 HTAB H + H(G): VTAB V + V(G
  ): INVERSE : PRINT "+";
5390 NEXT I
5400 HH = X: VV = Y: H = HH + H(G):
  V = VV + V(G): S = GZ(HH,VV,G
  )
5410 HTAB H: VTAB V: FLASH : PRINT
  "+";
5420 R = RND (1) * 100 + 100: &
  TR,8
5430 FOR Z = 1 TO RND (1) * 100
  0 + 500: NEXT Z
5440 HD = 0: VD = 0
5450 NORMAL : RETURN
```

Subroutine to evaluate the results
of the attack.

```
6000 GOSUB 7000
6010 VTAB 22: HTAB 1: PRINT SPC(
  119);
6020 IF S > 0 THEN 6120
```

Missed!

```
6030 GOSUB 7200
6040 VTAB 22: HTAB 1: IF APPLE <
  > P THEN 6070
6050 RESTORE : FOR I = 1 TO RND
  (1) * 10 + 1: READ R$: NEXT
  I
6060 PRINT R$: GOTO 6080
6070 PRINT "SORRY, "; PL$(P); ", Y
  OU MISSED."
6080 GZ(HH,VV,G) = - 6
6090 VTAB V: HTAB H: INVERSE : PRINT
  ";;; NORMAL
6100 FOR Z = 1 TO 2000: NEXT Z
6110 VTAB 22: HTAB 1: PRINT SPC(
  39): GOTO 6250
```

A hit!

```
6120 GOSUB 7400
6130 VTAB 22: HTAB 1: PRINT "A H
  IT ON "; PL$(G); "'S "; SS$(S);
  "!"
```

```
6140 GZ(HH,VV,G) = - 5
6150 VTAB V: HTAB H: PRINT S$(S)
  ;
6160 HITS(S,G) = HITS(S,G) + 1
6170 IF HITS(S,G) = SZ(S) THEN
  VTAB 23: HTAB 1: PRINT ; SS$(
  S); " SUNK!!!"; GOSUB 7600
6180 HITS = 0: FOR I = 1 TO 5: HIT
  S = HITS + HITS(I,G): NEXT I
6190 FOR Z = 1 TO 2000: NEXT Z: VTAB
  22: HTAB 1: PRINT SPC( 119)
  ;
6200 IF HITS < 17 THEN 6250
6210 FLASH : VTAB 22: HTAB 1: PRINT
  PL$(G); "'S FLEET IS DESTROYE
  D!"; NORMAL
6220 GOSUB 7500: FOR Z = 1 TO 10
  00: NEXT Z
6230 VTAB 22: HTAB 1: PRINT SPC(
  39);
6240 OUT(DEAD) = G: DEAD = DEAD +
  1: IF DEAD = NP - 1 THEN POP
  : GOTO 310
6250 RETURN
```

Apple's comments.

```
6260 DATA "RATS, MISSED AGAIN."
  ,"I THOUGHT I HAD YOU THAT T
  IME!"; "I WAS SURE I SAW A BL
  IP ON RADAR. . ."; "JUST WAIT
  TILL NEXT TIME."; "THE BATTL
  E ISN'T OVER YET!!"
6270 DATA "I'LL BET THAT WAS CL
  OSE, THOUGH!"; "I HOPE YOU'RE
  A GOOD SWIMMER."; "THAT WAS
  JUST TO GET YOUR ATTENTION."
  ,"ONE LESS PLACE TO HIDE. .
  ."; "NEXT TIME I'M GONNA AIM.
  . ."
```

Sound subroutines:

Falling bomb.

```
7000 N = 5: TM = 10
7010 FOR I = 1 TO 255 STEP N: &
  TI, TM: NEXT
7020 RETURN
```

Plop!

```
7200 P1 = 200 + RND (1) * 56: P2 =
  80 + RND (1) * 58: T1 = 10 +
  RND (1) * 6: T2 = 18 + RND
  (1) * 6
7210 FOR Z = 1 TO 500: NEXT Z
7220 & TP1, T1: & TP2, T2
7230 RETURN
```

Explosion.

```

7400 POKE - 16302,0
7410 R1 = 200 + RND (1) * 55;R2 =
3 + RND (1) * 3;R3 = 18 + RND
(1) * 7;R4 = 3 + RND (1) *
3
7420 FOR J = 1 TO 4 + RND (1) *
3: FOR I = 1 TO 8 + RND (1)
* 3: POKE - 16304,0
7430 & TR1 / I,R2: & TR3 * I,R4
7440 POKE - 16303,0: NEXT : NEXT
7450 RETURN

```

Musical theme.

```

7500 & T192,200: & T153,100: & T
128,100: & T114,150: & T153,
50: & T114,200: & T96,200: &
T86,100: & T128,100: & T96,2
50
7510 RETURN

```

Klaxon.

```

7600 FOR J = 1 TO 3
7610 FOR I = 1 TO 30: & T255,3: NEXT
I

```

```

7620 FOR I = 1 TO 30: & T150,8: NEXT
I
7630 FOR I = 1 TO 20: & T255,3: NEXT
I
7640 FOR I = 1 TO 300: NEXT I
7650 NEXT J
7660 RETURN

```

Poke in the machine language sound-generation routine.

```

7700 POK$ = "201,084,208,015,032,
177,000,032,248,230,138,072,
032,183,000,201,044,240,003,
076,201,222,032,177,000,032,
248,230,104,134,003,134,001,
133,000"
7710 FOR I = 1 TO 35: POKE I + 7
67, VAL ( MID$ (POK$,I * 4 -
3,I * 4 - 1)): NEXT I
7720 POK$ = "170,160,001,132,002,
173,048,192,136,208,004,198,
001,240,007,202,208,246,166,
000,208,239,165,003,133,001,
198,002,208,241,096"
7730 FOR I = 1 TO 33: POKE I + 8
02, VAL ( MID$ (POK$,I * 4 -
3,I * 4 - 1)): NEXT I
7740 POKE 1013,76: POKE 1014,0: POKE
1015,3
7750 RETURN

```

PADDLE SOUND EFFECTS

by James D. Dwyer
Dandelion Micro Products

If you are into computer-generated sounds, this is one program you will want to add to your collection. It requires an Apple II with a "revision 1" board, or an Apple II Plus. You do NOT have a revision 1 board if your Apple can only muster four colors (including black and white) in Hi-Res graphics, if you must press "reset" each time you power up, and if your letters are always tinged with color even in TEXT mode.

This routine uses two speakers, one of which must be connected via the cassette port as follows:

1. Connect the cable from the "cassette out" port on the Apple to the microphone jack on your cassette recorder, in the normal manner. Do not connect the other part of the cable to the cassette monitor jack.

2. Connect an 8-ohm speaker to the cassette monitor jack.

3. To put the speaker into operation, place the cassette recorder into "record" mode. You can either insert an unused cassette, or reach into the back left of the cassette compartment and push on the little metal "finger" as you press the record key.

Both the on-board speaker and this added one play tones which are controlled by the two paddles. Rotating one or both of them provides an endless variety of sounds — weird, exotic, bizarre, and even harmonious.

Here is the Applesoft program which pokes in the routine that does it. (If you have only Integer BASIC, you'll need to skip the READ and DATA statements, and use 19 separate POKES: POKE 768,173 : POKE 769,48 :, etc.) Just type in the program, RUN it, and start twisting the paddles.

```

10 FOR I = 768 TO 786: READ
D: POKE I,D: NEXT I
20 DATA 173,48,192,162,0,32,30,
251,173,32,192,162,1,32,30,2,51
76,0,3
30 CALL 768

```

APPLE ONE LINERS

```

10 HGR : FOR X = 1 TO 10000: HCOLOR=
6 * RND (1) + 1: D = INT ( RND
(1) * 278) + 1: E = INT ( RND
(1) * 158) + 1: H$ = D,E TO
D + 1,E TO D + 1,E - 1 TO D,
E - 1: NEXT X

```

Matthew Levie
New York, NY

```

10 HGR : FOR X = 1 TO 10000: HCOLOR=
6 * RND (1) + 1: H$ = D,E TO
D + 1,E TO D + 1,E - 1 TO D,
E - 1: NEXT X

```

Mathew Levie
New York, NY

```

1 HOME : HGR : FOR L = 1 TO 2: X =
RND ( PEEK (78) + PEEK (79)
) * 256: X = RND ( - X): A =
INT (209 * RND (1)) + 35: B
= 159: D = 35: HCOLOR= INT
(H / 10): FOR P = B TO D STEP
- 1: H$ = A,B TO A,P: NEXT
P: FOR T = 1 TO H: G = INT (
70 * RND (1)): H = INT (70 *
RND (1)): G = G - 35 + A: H$ =
A,D TO G,H: L = 1: NEXT T,L

```

Mal Holman
Lexington, MA

REVIEWS



Temple of Apschai

Automated Simulations

Ever moved into an apartment only to find it already occupied by ants, roaches and other chitinous vermin the size of Chryslers? They aren't that hard to banish (once you begin to understand their traffic patterns, anyway), but it usually takes a long time and can get pretty messy. But if you think evicting kitchen-loving sextopeds is a chore, try traipsing through "Temple of Apschai", Automated Simulations' Fantasy Role Playing (FRP) game-derived computer adventure. NOW we're talking big bugs: giant ants, giant mosquitoes, giant termites, even the dread giant bombardier beetles. Certainly not things you'd like to find hiding behind the stove.

To play "Temple of Apschai", you need a character. You can make one up, investing him or her with superhuman capabilities in the characteristics of intelligence, strength, intuition, ego, constitution and dexterity, and equipping your character with magic armor, weapons, and enough healing potions to open up a hospital. Some people would call this taking the path of least resistance, but the old sticks in the mud who review games might be moved to call it cheating. But whatever you call it, it does tend to cut down enjoying the game because it makes everything too easy. If you aren't so into instant ego gratification, you can ask the Innkeeper who you meet at the beginning of the game to select a character for you by randomly generating characteristics and determining the number of silver pieces you have. He will also sell you the equipment you need. (Be warned that he loves to dicker, and will try to take advantage of those with a low intelligence; he once asked a slow-witted fellow I know named Dumbeaux for 13,000 silver pieces for a dagger.)

After you are equipped and named, it is off to the awful temple of the insect god to fight monsters, find treasure (although some of the treasure you find is trash), and in all likelihood

provide some antman with a late-night nosch.

There are limitations to the game, but none of them are really fatal. The pace can get pretty slow, since as you progress down the halls and from room to room, the computer pauses to draw what you see, and it sometimes takes its own sweet time doing it. And while there are a fair number of non-insects around to battle, all those damn bugs can get a bit tedious. It is difficult to feel very threatened by a giant termite. What's he gonna do, eat your clogs? It is also a little disconcerting to have monsters suddenly materialize out of nowhere or to swing your sword through thin air and hear something go "crunch". The scope of the game is also limited in that the player can assume no role other than fighter.

Nevertheless, "Temple of Apschai" is enjoyable and comes closer than most computer games to the flavor of FRP games. It doesn't wear thin quickly like some games, if for no other reason than that it takes rather a long time to explore all four levels of the temple, and if you get tired of one character, the Innkeeper is more than happy to generate another one for you. All in all, it is a reasonably good investment.

Michael Humes

Apple Machine Language

by Don Inman and Kurt Inman
Reston Publishing Co.

If you've pretty well mastered Apple BASIC and need more challenge, or have become frustrated with some of BASIC's limitations, then the next logical step is to learn Machine Language. Not necessarily to write all your programs henceforth on that level, but at least to start peppering your programs with Machine Language subroutines.

If you've seen cryptic listings such as

0300-A2 05 LDX #05

0302-A9 22 LDA #22

and have a mortal fear of departing the more or less refined, English world of BASIC to enter a primitive foreign region, take heart! **Apple Machine Language**,

by Don and Kurt Inman, takes a lot of the fear out of foraging into the realm of hexadecimal addresses, mnemonic codes, and monitor subroutines.

The authors take a step-by-step approach to learning Machine Language, assuming nothing more than reasonable competence in BASIC. The pace seems about right, neither painfully slow nor breathtakingly fast. The book is intended to be used hand-in-hand with your Apple, so that you learn not just by reading but by continually applying what you read. If you just read through the pages, you might have the impression that the authors are repeating themselves, but that repetition should be an asset in the learning-by-doing process.

One minor stylistic gripe about the book is its periodic use of the word "data" as a singular noun (e.g., "the data has been placed in the memory location...", p. 29). I know I'm fighting a losing battle, but in my book "data" is still the plural of "datum". Also, there's a minor error in the memory map on page 26: "64K installed" should read "48K installed." These are indeed LITTLE things, and shouldn't detract from the overall very good quality of the book's written style and visual appearance.

Jon Voskuil

Dragonquest

Programmer's Guild

Calling all worthy heroes, calling all valiant warrior types! There's a dragon trying to create an international incident. Having been turned away at the embassy gates, scaly old Smaegor has snatched the King's daughter and made off for parts unknown. If you ask the King, he'll tell you that the aforementioned reptilian raptor has set a deadline of nightfall before he subjects the lovely princess to a trial by mastication. Here's a chance for fame and fortune, not to mention a nifty bit of adventuring.

"Dragonquest" is a Machine Language adventure written by Charles Forsythe and put out by the Programmer's Guild. It opens

with a cute little routine right out of obedience training that's nearly worth the price of admission by itself. From then on, all the de rigueur adventure stuff pops out at you from every direction: There's the water travel segment, the find a weapon routine, the Dr. Doolittle let's talk to the animals bit, and much, much more.



There are also some design factors in "Dragonquest" that are of interest. For one, you can pick up or drop a whole bunch of stuff at one blow, rather than singly, item by item. And there's the use of the shift key for entering the direction you wish to go: SHIFT E would help you move east, etc. Asking for help also produces some interesting results, but if I told you what they were then you'd miss out on the fun of finding out for yourself.

The adventure itself follows a fairly predictable logic. There are no glaring incongruities in it, nor is there any required esoteric knowledge. You are not out for treasure per se, but rather you have a specific task to accomplish within a limited amount of time. If one were to seek out the weak points of "Dragonquest", one would have to look at the lack of red herrings, the lack of blind alleys and dead ends.

"Dragonquest" is very trim, there are not many twists and turns in it. But for many, this would hardly be considered a drawback, but rather an advantage. This reviewer prefers obscurity, but then this reviewer's sanity has been questioned not infrequently.

In the end there's only one way to find out.. Pick up a copy and try to rescue the princess. She needs your help.

Dave Albert 



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.

TAPE VERSION, 16K (min.), LEVEL II \$24.95
32K DISK VERSION \$39.95
WITH INVOICING ON DISK \$59.95



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

FROM

MUSE

SUPER TEXT II...

Language of Super-Text II software — 6502 Assembly . . . gives extremely fast editing response.

Program Disk Protection — A backup copy of the program disk is provided.

Text Creation and Editing Features — Super-Text II includes a character-oriented, floating cursor edit with Add, Change, Math, Print and Preview modes. Among the standard features retained from the original Super-Text are:

- single key cursor control
- automatic word overflow
- character, word, and line insertion
- forward and backward scrolling
- automatic on-screen tabbing
- automatic paragraph indentation
- character, word, and line deletion
- ditto key
- single key for entering "the"
- block copy, save and delete
- advanced file-handling
- global (multi-file) search and replace
- column decimal alignment
- left and right justification of columns
- automatic user-positioned page numbering
- chapter relative page numbering
- complete printer tab controls in text files
- line centering
- superscripting and subscripting
- on-screen lower case with MUSE Data Shift Lower Case Adapter (Paymar also supported)
- shift key modification compatible
- Split-Screen allows the user to view and edit two portions of a file simultaneously.
- Math Mode provides a built-in 15 digit floating point calculator which performs on-screen calculations, column totals and verified numeric data in statistical documents.

Trade in your old word processor for Super Text II and receive the following credits: Documentation and proof of purchase must be included with return.

Super Text — \$100 credit
 Apple Writer — \$50 credit
 P.I.E. — \$50 credit
 Easy Writer — \$50 credit

Length and Linkage of Files — Maximum file size equals 20,049 characters. Automatic file linkage (Autolink) provides unlimited file size when doing find, find and replace, and preview or printing operations.

Page Preview Provided — Yes. Preview mode allows viewing a file exactly as it would print for inspection of page breaks and text formatting. The preview mode carries through AutoLink operations allowing inspection of extremely large documents.

Text Capacity of Screen — 23 lines by 40 characters.

Storage of Text on Disk — Manually at any time, automatically during AutoLink.

File Backup — Dual disk copy program is included on the program disk. Files may also be loaded and saved on a blank disk individually.

Automatic Pagination and Page Numbering Features — Page breaks are automatic based upon number of lines per page and may be forced at any point using a control character in the text file. Page numbering is automatic but may be suppressed. Location of the page number is user selectable. Chapter relative page numbering is also provided.

Printers Supported — Any printer which will interface with an Apple computer.

Right Justification Technique — Provided by variable between word spacing.

Special Printing Features Available — Features include: underline, boldface, superscript, subscript. Other special printer features, including multiple character fonts, may be defined by the user.

On-screen Display of Special Printer Function — Inverse characters are displayed on the screen to indicate special features. These control characters are not displayed in preview mode.

Control Characters — All printer control characters are inverse so there is no conflict with printable characters. However, a user may define any printer control sequence.

Apple II and Apple II + 48K Disk \$150.00



BUGS, WORMS,

and other undesirables

In the S-80 Disk version of "Convoy" published in the January, 1981, issue of **SoftSide**, Line 5020 should be changed to:



```
5020 OPEN"0",1,LD$:PRINT#1,A:PRINT#1,A1:PRINT#1,K,R(1),R(2),R(3),R(4),R(5),R(6),R(7),R(8),S,YE,N,TS:CLOSE:GOTO 3110
```

In the S-80 version of "Tanks A-Lot" published in the February, 1981, issue of **SoftSide**, the left commander's control keys should be:

```
FORWARD — S  
FIRE — D  
ROTATE RIGHT — A  
ROTATE LEFT —
```



In the Atari program "Changing Hearts" published in the February, 1981, issue of **SoftSide**, there should be a "control comma" added to the end of the character string in Line 40.

DataBase continues...There is a problem with the delete routine in **SoftSide's** Data Base that has been brought to our attention (boy has it!). For example, if you try to delete the first three records, the first will be deleted fine, and the remaining data will be moved to fill up the resulting space. But when record two is deleted, the program is really deleting what started as record three, and record two has now become record one! Below are corrections to this problem for all three systems. After this correction is made, the above example would delete record one first, but after a record is deleted the record number pointer is not moved forward, so no records are skipped.

ATARI

```
200 TRAP 200:CLOSE #3:RL=(NH+1)*IL:PRI  
NT CHR$(125);"(S) SAVE CURRENT DATA"  
6140 NI=NI-1:SS=0:I=I-1  
8250 I=I-1  
8252 FOR I9=I1 TO I2:I=I+1  
8375 IF RS=1 THEN I9=I2  
8380 NEXT I9
```

APPLE and S-80

```
6140 NI = NI - 1:SS = 0:I = I - 1  
8250 I = I1 - 1  
8252 FOR I9 = I1 TO I2:I = I + 1  
8375 IF RS = 1 THEN I9 = I2  
8380 NEXT I9
```

The Apple translation of "Strategy Strike" published in the March, 1981, **SoftSide** has one mistake which needs correction in two places. Lines 730 and 5060 are supposed to check for a mine, but erroneously use the number 5 rather than the number 8. The lines should be as follows:



```
730 IF A2 = BOMB AND A1 = 8 THEN  
770
```

```
5060 IF (A1 = SPY AND A2 = 1) OR  
(A2 = BOMB AND A1 = 8) THEN  
W(P) = 1: GOTO 5110
```



HARDWARE CORNER



by Edward E. Umlor

This month we are going to talk about some of the disk drives and cables that are available for your S-80. There have been a lot of people jumping on the disk bandwagon. I intend to cover several disk drives and their compatibility, and also the suppliers of these drives. First we need some definitions.

Drive 0: That drive which comes on when power is applied or the reset button is pushed. This is the drive your Disk Operating System (DOS) must be in to go into disk operation.

Flippy Drive: This drive has two sets of write protect and beginning of track sensors. This allows you to take the disk out, turn it over, reinsert it into the drive, and use it like a brand new blank disk.

Tracks: The number of steps the read/record head can move. Also, the number of concentric circles of data that can be put on the disk.

Speed: The rotation speed in revolutions per minute (rpm) of the magnetic media inside the disk envelope. This should be 300 rpm for 5¼" disk drives.

Terminator: This is a DIP (Dual In-line Package) circuit that contains resistors for matching the drive electronics to the cable. It is necessary to minimize the amount of interference between two adjacent wires carrying signals.

Density: You can get twice as much information on a double density disk as on single density. (For a more complete discussion of density, see "What's New" in the February 1981 *SoftSide*).

There are a few manufacturers of the drive mechanisms

themselves and many brand names of drives. These few manufacturers supply drives and the brand name advertiser builds a cabinet and power supply. These may take several forms: One will do all the programming of the drive (0,1,2...etc.) on a connector extender card, another will connect directly to the drive circuit board, and some will allow multiple drives in one overall cabinet.

DRIVE 0

One of the very first things you are confronted with is which connector is drive 0? The Tandy way is the connector closest to the expansion interface is drive 0, and Hardside's cable places drive 0 on the connector farthest from the expansion interface. You should receive directions with the cable or the drives on the matter, but if you don't, you need not panic. Hook up the system and place a drive on the first connector (closest to the expansion interface). If the red light does not come on and the motor doesn't run, then move the drive to the last connector and try again. You will not hurt your drive. **BE SURE TO REMOVE THE DISK FROM THE DRIVE BEFORE TURNING OFF THE POWER!** If you don't follow this cardinal rule, your diskette can receive a TANDY ZAP. This is the common name for spurious data being placed on the disk (usually the directory track or track 0) as various signals are generated during the power down.

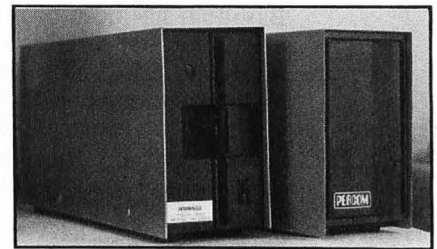
SETTING UP DRIVES

The next thing you are going to be faced with is setting up your drives for the specific position they are to occupy in your system. You should receive directions on how to do this with the drives. The terminator should remain in the drive that is positioned farthest from the interface, and be removed from all the other drives in the system. Too many terminators on line can load the output stages of the drives and the interface and cause damage to these circuits. You might also have to change a jumper's position or

change the setting of a dip switch to tell the drive it is not drive 0. All the information you will need should come with the drives. Now let's look at some of the names available.

KINDS OF DRIVES

HARDSIDE 40-Track: This is actually a TEAC drive mechanism. The cable attaches directly to the drive circuit board and requires the removal of the cover. The newer drives require a dip switch (DS3 position) to be changed from ON to OFF for drive positions 1, 2, and 3. It is completely compatible with the Model I S-80. We have not had any problems in mixing the TEAC with drives from other manufacturers. I have been using these drives with the Doubler for several months with very solid double density operation. The formatted storage capacity is 102K bytes single density and 180K bytes double density.



**HARDSIDE 40-TRACK DRIVE (left)
PERCOM TFD-40 DRIVE (right)**

HARDSIDE 80-Track: This is also a TEAC drive mechanism and does all the 40-track will do. We have checked out the double density operation and found it just as solid as the 40-track. The main difference is the formatted storage is 204K bytes single density and 360K bytes double density. Three of these and you have one MEGABYTE on-line storage in double density.

PERCOM TFD-40: This is a 40-track drive using the MPI drive mechanism. It is also rated for double density operation with the same storage capacity as the HARDSIDE 40. The cable is internal to the cabinet (connects to the drive circuit board). You don't have to change anything to set up for a different drive position.

PERCOM TFD-100: This is the first flippy drive we have talked

about and is a SIEMENS drive mechanism. The track count is 40 tracks and sets up similar to the TFD-40. The biggest advantage of a floppy is the use of both sides of the disk. Each box of ten diskettes you buy is like buying 20 diskettes. If you do a lot of work with large amounts of data being stored, you can save money in the long run. The initial cost of this drive is \$20.00 more than the TFD-40, but this can be saved on your first box of diskettes. All PERCOM drives are double density rated (after all it is their Doubler).

CCI-100(T/M): These drives are 40-track TEAC (T) or 40-track MPI (M). The difference here is the cables connect to an extender and there isn't a need to change any switches to set up position. You just plug them in and go after you remove the extra terminators.

There are many other brand names and it would take a book to cover them all. The hookup, set-up, and operation of all of them are similar. You will have to choose your DOS (Disk Operating System) carefully for the drives you are using. TRSDOS, DOUBLEDOS, and NEWDOS+ do not allow the mixing of 35-track, 40-track, 77-track, or 80-track drives (either 5¼ or 8 inch) on the same system. NEWDOS/80 will allow the mixing of drives (it has a special command PDRIVE that allows you to specify diskette size and number of tracks for each position in the system.

TRACK COUNT

One of the interesting things I have found out in playing with my drives is that the track count can be extended by one or two tracks. I used NEWDOS/80 zapped to double density (see "What's New", February 1981 *SoftSide*) and experimented with formatting. The PERCOM drives give 41 tracks solid and the TEAC drives give 42 tracks solid. Since I am using the TEACs, that gives me 40 full tracks of data with 18 sectors per track. Those two extra tracks (36 sectors) are enough to hold this entire article with a little room to spare. One track or even two tracks don't sound like much, but you can see by the length of this article, it does add up.

Well, that's all for this trip around the keyboard. I am looking forward to receiving some questions in the mail bag. **HAPPY COMPUTING FROM OLD GRANITE KNOGGIN.**



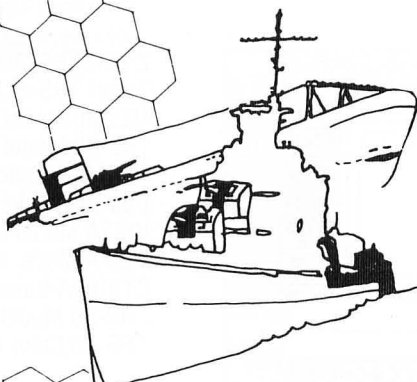
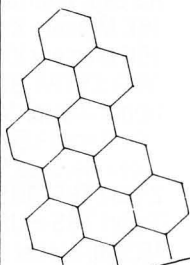
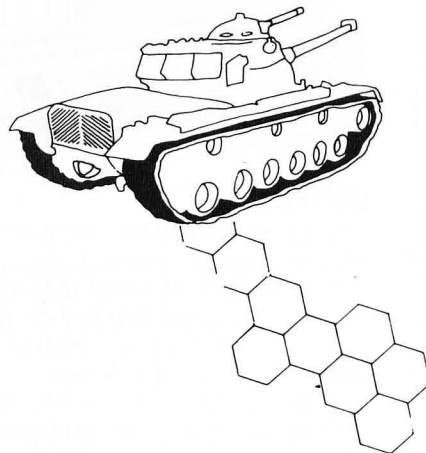
Wargamer's delight

Three from Potkin

1). 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



2). Up Periscope

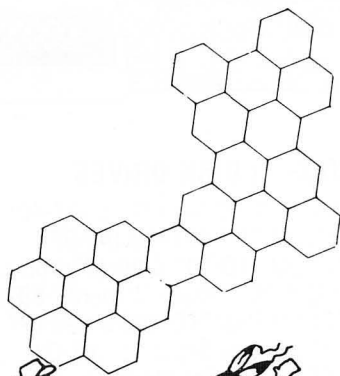
The author of the popular Kriegspiel II has done it again. This time the action takes place at sea with one player controlling the submarines while the other attempts to sail around RADSHA Island, with at least three of his fleet surviving the attempt. This realistic wargame includes sonar, depth charges,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

S-80 Level II, 16K cassette \$14.95

3). Warpath

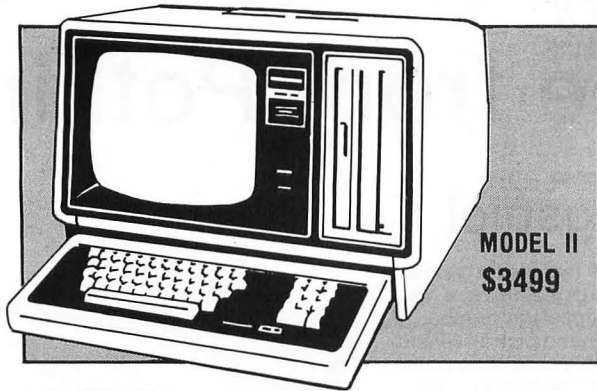
The Indians are on the warpath! The Chief, along with 24 braves, is out to take the garrison at the fort, or at least to stop reinforcements from entering the stockade. The General, with his 14 troopers, is trying to relieve the garrison before the flag is captured. The player determines the scenario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

S-80 Level II, 16K cassette \$14.95



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

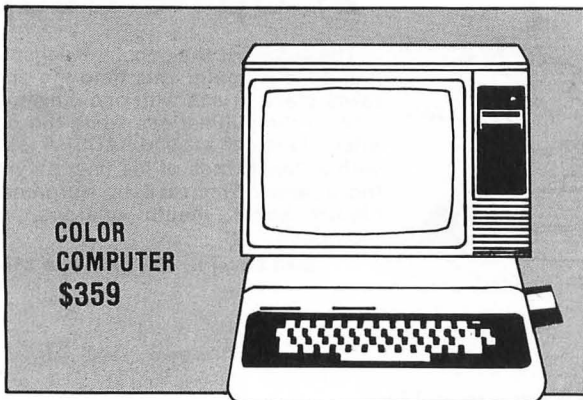




MODEL II
\$3499

COMPUTERS

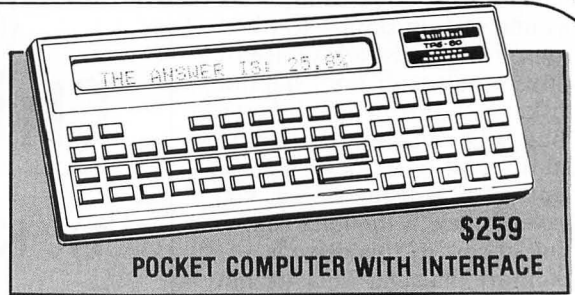
- TRS-80 Model II 64K RAM (#26-4002) \$3499.00
- TRS-80 Model III, 16K RAM (#26-1062) \$919.00
- TRS-80 Model II, 48K RAM (#26-1062+) . . . \$1019.00
- TRS-80 Mod. III 48K RS232 2-dr. (#26-1063) \$2299.00
- TRS-80 Pocket Comp. w/Interface (#26-3501+) . \$259
- TRS-80 Video Tex (#26-5000) \$359.00
- TRS-80 Color Computer, 4K RAM (#26-3001) . \$359.00
- TRS-80 Color Comp. 16K RAM (#26-3001+) . \$399.00
- TRS-80 Color Comp. Ext. BASIC (#26-3002) . \$529.00



COLOR
COMPUTER
\$359

MODEL I DISK DRIVES

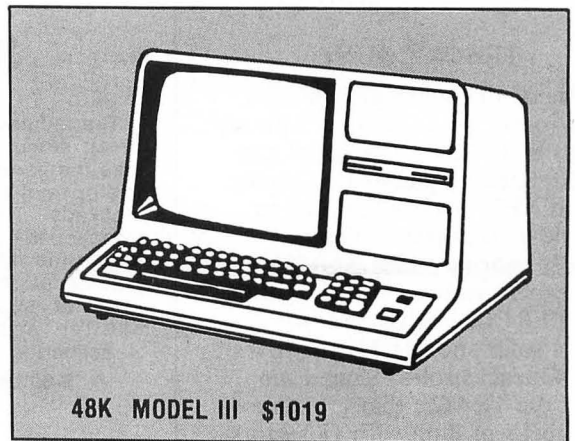
- HARDSIDE 40-track Drive (#7-40) \$319.00
- PERCOM TFD-40 Drive (#7-99) \$379.00
- PERCOM TFD-100 Drive (#7-100-1) \$399.00
- HARDSIDE 80-track Drive (#7-80) \$449.00
- PERCOM Dual TFD-100 Drives (#7-100-2) . . . \$799.00
- PERCOM Data Separator (#7-03) \$29.95
- PERCOM Doubler (#7-07) \$199.95
- HARDSIDE Extender Cable (#7-02) \$15.95
- HARDSIDE 2-Drive Cable (#7-04) \$29.00
- HARDSIDE 4-Drive Cable (#7-05) \$39.00



\$259
POCKET COMPUTER WITH INTERFACE

MODEL I PERIPHERALS

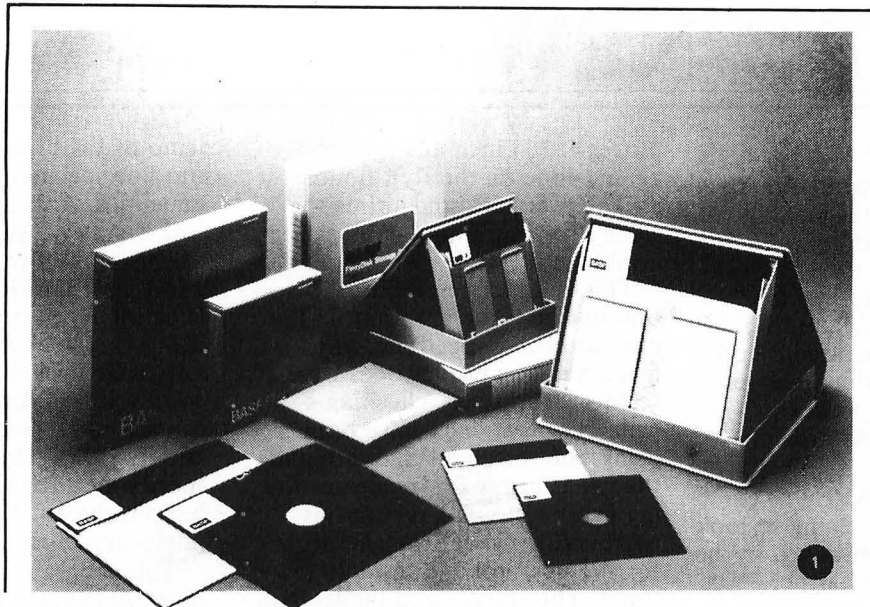
- COMM-80 Interface (#4-80) \$159.00
- CHATTERBOX Interface (#4-81) \$239.00
- Disk-80 Interface, 32K RAM (#4-82) \$339.00
- DISK-80+ Interface, 32K RAM (#4-83) \$369.00
- BUSY BOX Interface (#4-01) \$89.95
- LYNX Communications Interface (#19-80) . . . \$229.00
- RS Exp. Interface, 32K RAM (#26-1140-32) . . \$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) \$24.95
- 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) \$39.95
- 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
- VISTA Model II 8" Disk Drive 1(#7-4001) . . . \$939.00
- VISTA Model II 8" Disk Drive, 3(#7-4002) . . \$1795.00
- 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



48K MODEL III \$1019

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.



We have all the supplies you need for your micro.

BASF DISKETTES

Box of 10, 5¼" Sgl. Density \$34.95
 Box of 10, 5¼" Dbl. Density \$44.95
 Box of 10, 8" Dbl. Density \$49.95

DYSAN DISKETTES

104/1 One-sided single density (one side certified error free at a single density of 3979 bpr.) In track and between track testing is performed to ensure that the entire recording surface of every disk is 100% error-free.
 Box of 5 diskettes \$29.95

3-M SCOTCH DISKETTES

Encased in a rough jacket which resists handling damages. 100% certified error-free performance. Low modulation provides better signal stability.
 Box of 10. \$39.95

DIGITAL CASSETTES

Premium quality digital recording tapes. These five-screw cassettes come packaged in boxes of ten. They are offered in 10 and 20 minute lengths.
 C-10 \$6.95
 C-20 \$7.95

DISKETTE HEAD CLEANING KIT

3-M Scotch 7400 head cleaning kit is simple and easy to use. You simply saturate the write head cleaning fabric in the cleaning diskette with the cleaning solution, insert the diskette into the drive and turn it on. The rotating cleaning fabric alternately wipes the heads with the solution and the dry surface, removing contamination from the read and write head. Each kit contains two cleaning diskettes which will allow you a total of 100 cleanings.
 \$29.95

DISKETTE LIBRARY CASE

Allows you to store up to 15 mini-diskettes in a strong molded plastic carrier.
 \$5.00

FLOPPY DISK SAVER

PREVENTS: Computer drive's cleaning clamping hub from tearing disk's center hole;
 Coating removal, scuffing, dimpling;
 Data loss caused by improper rotation.
 \$14.95
 Refills \$7.95

RECORDING HEAD TAPE ALIGNMENT KIT

It consists of a template for locating the adjustment screw. An alignment tape is also included and a special screwdriver as well. The kit will allow you to align the heads to an industrial standard, ensuring consistent loading results.
 \$9.95

call us today!




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

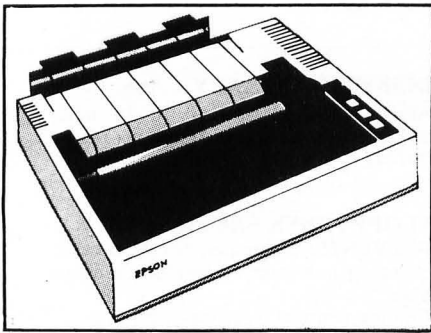
WHAT'S NEW

by Edward E. Umlor

Hey!! I haven't heard from any of you yet. We are still looking for submissions to this column. Each submission used will carry its own byline. We sure would like to publish your material, and that goes for you manufacturers. How about some new product releases or write-ups on your newest software. Looking forward to hearing from you.

HARDWARE

This month we are going to look at a unit that was mentioned in the February "Hardware Corner." That's right. We are going to review the MX-80 printer. This little guy seems to be very popular and with good reason I think. This



EPSON MX-80 PRINTER

relatively inexpensive printer does as much, if not more, than a lot of the BIG GUYS. Here are some of the items from the spec sheet:

Print Method: Impact dot matrix printing using 9 x 9 for ASCII characters (yes — descenders for lower case), and 6 x 12 for graphics (done in a two-pass print).

Speed: 80 characters per second (normal) bidirectional.

Paper Drive: Tractor only. Adjustable from four to ten inches. Accepts three-part paper (original plus two copies using carbon paper). There is some scuttlebutt about a MX-80TF (tractor/friction) on its way.

Line Spacing: Software selectable 6 or 8, 1/6 inch or 7/72 inch. Your choice!

Line Length: 132, 80, 66, or 40 characters per line, software selectable.

Print Density: Four software selectable print densities — normal single strike, emphasized (head

moves 1/264 inch and strikes again, done on the fly), double strike (goes back and prints the same line again), and emphasized double strike.

This is a nice selection of options from the program. If the question has popped into your head, yes, the printer does have a self-test feature for checking general operation. I have been using one for about a month and the use is heavy (doing demos, all my letters, etc.) To date it has performed without fault (except for the operator's from time to time). The emphasized mode gives excellent quality for doing letters and the emphasized double strike makes excellent masters for photocopiers.

One of the big questions people have is the setup for S-80s. The unit comes S-80 compatible with the graphics offset to 160-223 decimal. There is an internal switch for S-80 direct compatibility. This will change the graphics codes, but will also defeat some of the neat features of this little printer. I do not recommend changing the switches to only S-80 compatible. I have too much fun discovering new combinations of size and density, etc. The printer also has a selectable bell (a three-second beep) that I put in at the end of long printing sequences. This allows me to go do other things until the job is done, and it will cry for my attention.

The printer comes with a complete list of control and escape (ESC) codes. Some people get confused and put an ESC in front of every code. This will defeat the control code sequences. The control codes are output as straight CHR\$(value) and the ESC codes can be output as CHR\$(27);CHR\$(value) or CHR\$(27)+"letter". As an example to line feed and go into emphasized mode output:
LPRINTCHR\$(10);CHR\$(27);
CHR\$(69); or
LPRINTCHR\$(10);CHR\$(27)+"E";

Both of the above lines will do the job. In the May "Hardware Corner", I will try to cover some programming hints to make life faster and easier when using these types of codes. I have done a little

demo of the printer without going into line spacing. I hope it gives you some idea of what it can do.

SOFTWARE

This month we are going to take a look at "ULTRA-MON." This is an intelligent utility for debugging your Assembly Language programs or seeing how your S-80 ROMS do things. You can set break points both in RAM and in ROM. The latter is done by specifying a ROM address as a jump to break. It allows interpretive single-step execution of a program.

This program allows for the quick debugging of your assembly programs by helping you display registers, interpreting instruction being performed, and being able to track relative position in the program when bomb-out occurs. Single step through that area to find the goof and change bytes to correct the problem. You can then rerun the program to verify corrected operation.

I like the disassembly feature. It will give a symbolic dump with jumps shown as + or - relative locations. It will also give an ASCII character for being compared for program decisions. You can dump to the display and print the display on your printer. You can modify data in register pairs, blocks of memory, or program bytes.

It would take a long time to be able to tell of all the goodies in this program, and even longer to discover all the uses of the subroutines. For the money it is one of the best Assembly Language manipulation programs I have reviewed. If you are into assembly programming, this program could sure pay for itself with time saved in debugging and building working modules. I would give it at least a ★★★ rating with★★★★ being tops.

Well that's all for this month. If you have a utility program or a piece of hardware you would like reviewed, just send a sample to:

WHAT'S NEW
SOFTSIDE MAGAZINE
6 SOUTH STREET
MILFORD, NH 03055
ATTN: E. UMLOR

Please be sure to include the documentation for the item.

AUTO MODPROG

by Shane Causer

Shortly after I saw "Modprog" published in *SoftSide* (August, 1980), I thought about how nice it would be if you could let the computer modify a statement Level II users wouldn't use, like a LSET statement, into a statement that could be used on the TRS-80™ Level II computer.

First of all, I should explain the method behind program modification, for all of you that came in late. Every BASIC keyword (FOR, RESET, CLS, etc.) has a one-byte code assigned to it. These codes are stored in ROM, along with the keyword. The following program brings them out:

```
10 A=128
20 FOR S=5712 TO 6180
30 IF PEEK(S)>128 THEN
PRINT: PRINT
A;" ";CHR$(PEEK(S)
-128);:A=A+1:GOTO 50
40 PRINT CHR$(PEEK(S));
50 NEXT S
```

Each keyword has its code, and when the interpreter comes across one of the codes, it translates it into the proper word. By using the POKE command, you can add statements, change statements, and delete a keyword. We shall work with this principle.

How will we set up a test for the statement that we wish to find? The obvious way is by setting up a program loop. But to set a loop properly, we need to know where the loop begins and ends. By referring to the memory map in the Level II manual, we find that RAM starts at 17129, and ends at locations that vary according to memory size (4K ends at 20479, 16K ends at 32767, 32K ends at 49151, and 48K ends at 65535).

We can easily see that a loop from 17129 to 32767 (assuming a 16K machine) would take a long, long, long time. An easy way around this is to take a guess (or type ? MEM) to see how much memory the program takes up, and set up the loop accordingly.

Once we have a loop set up, we need to put a PEEK statement in the program. This PEEK statement would interrogate (like that word?) the memory location set up by the

loop. If the location holds the code of the keyword, we POKE in the new value for the keyword we want.

Here's a simple program to demonstrate all that we have covered so far.

```
10 FOR P=17129 TO 18500
20 IF PEEK(P)=171 THEN 40
30 NEXT P:END
40 POKE P,141
50 END
60 LSET 10
```

In the above program, line 10 sets up a loop. This loop is larger than need be, but it's just an example. Line 20 examines the contents of the memory location set up in the loop. If the contents are equal to 171 (the code for the LSET command), then we branch to line 40 which POKES memory location P with 141 (the code for the GOTO command), then ENDS the program. If location P isn't equal to 171, the loop goes back through the next location. You may notice that line 60 doesn't make any sense. LSET will bring about a L3 error in Level II BASIC.

Type the program in, and LIST it. Look at line 60. That's going to be your last look at the LSET command. RUN the program. When the computer comes back with a "READY" prompt, LIST the program, and line 60 should make sense. The line should look like this:

```
60 GOTO 10
```

You are probably saying "Very cute. But so what?" Simple. Let's suppose you write a neat program that requires a line to be entered, or modifications need to be done to it that would eat up memory. This would confuse any users other than yourself, because they:

a) May not understand programming, or

b) they may understand but don't want to go to all the trouble, or

c) they don't know what the heck you're talking about!

Look at this program for an idea:

```
10 CLS
20 INPUT "PRESS 'S' FOR
SINE OF 56, 'C' FOR
COSINE"; A$
30 FOR P=17129 TO 17900
40 IF PEEK(P)=171 THEN 50
ELSE NEXT:END
50 IF A$="S" THEN POKE
P,226 ELSE POKE P,225
60 PRINT LSET(56);
```

If we look at this program, we find out something very important: The POKE action only changes the keyword. It doesn't insert or remove parentheses.

What we could do, very simply, is place two LSETs together, modify our program so that it searches for two LSETs together, then modify it so it would print "B=". This is the the easiest way to illustrate how not only keywords are subject to modifications.

Modify our second program by adding another LSET immediately before the other LSET. DO NOT leave a space between them.

Change line 20 to read:

```
20 IF PEEK(P)=171 AND IF
PEEK(P+1)=171 THEN 40
```

Change line 40 to read:

```
40 POKE P,66:POKE P+1,61
```

RUN the program, and when you list line 60, it should read B=10.

These programs have barely touched on what can be achieved if you put your mind to it. Imagine how much time could be saved if you wrote a program that solves for equations, but writes in its own intrinsic functions where you specify. Hmmm.....

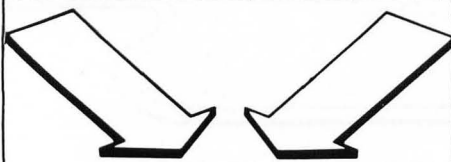
S-80 ONE LINERS

```
1 CLEAR:FORT=1T06:A$=A$+CHR$(RND(59)+128):NEXT A$;A$+CHR$(32):FO
RT=1T0300:PRINTA$;:NEXT:GOTO1
```

Dan Singer
Washington, DC

```
1 RANDOM:CLS:FORY=1TORND(23):FORT=15360T016383STEP:RND(63):OUT255
,4:OUT255,8:A=PEEK(14351):A=A+RND(13):POKET,A:POKE16445,RND(16):
OUT255,0:NEXT T:FOR T=16383T015360STEP-RND(64):OUT255,4:OUT255,8
:A=PEEK(14351):POKET,A:POKE16445,RND(16):OUT255,0:NEXTT,Y:RUN
```

Steve Rejto
Greenwick, CT



OPTIONS MONITOR

by W. J. Kutlever

THE INVESTOR'S KEY TO OPTIONS MARKETING!!!

Make informed trading decisions in minutes with this easy to use package.

WHO: Investors, Option Writers or Purchasers use this program.

WHAT: Analyzes option prices and responds to current market prices as soon as you receive them.

WHY: Assesses the value of a given option. Monitors the value of your investments. Helps you decide when to purchase or to close out your options.

HOW: Use S-80 Disk.

Next to your broker this program is your best option in creative investing. Analyze option prices before risking any money. Convenient features are data entry and flexibility in handling a wide variety of situations.

Available for Level II, 16K S-80 Microcomputers.

\$29.95



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

STATPAL

by Bruce Chalmers

02
8 6 4 6
3 4 7 3
9 8 4 8
5 2



Stumped by statistics? Here's the program for you!

Written by a statistician but designed for use in the real world. Helps you create files; examine and edit data; and understand descriptive statistics.

Sophisticated enough for the working statistician. This powerful interactive statistical package features complete error diagnostic, missing value specification, and sophisticated graphics.

S-80 Disk 32K and 48K versions on one disk \$29.95

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

Nine Games for Preschool Children

by George Blank

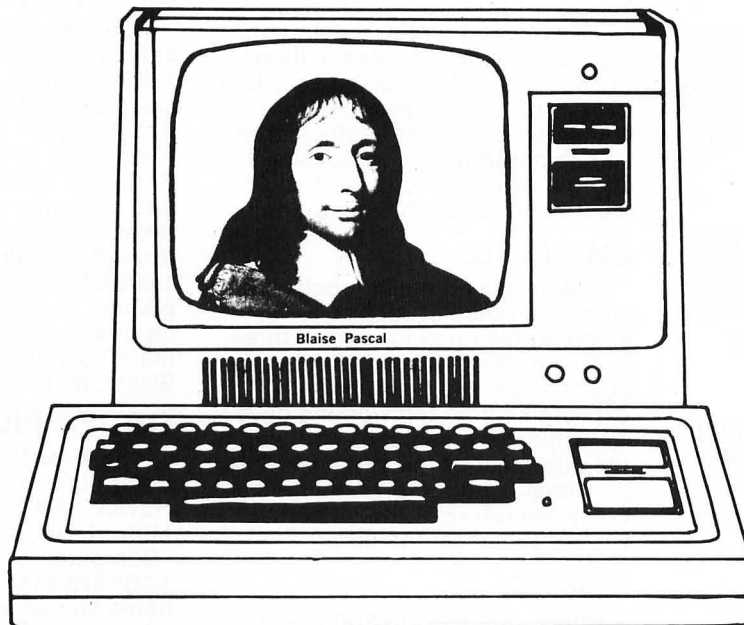
Even preschoolers deserve a shot at the wonders of microcomputing. With these nine games, they not only will have a chance to tickle the keyboard, but learn letters and numbers to boot. And if that isn't enough, they'll have a good time doing so. What more could a parent ask for? Here are education and entertainment for the very young in a single package!

S-80, Level II, 16K, Cassette \$9.95



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

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 \$99.95



TERMS AND CONDITIONS

DOMESTIC:

TSE/HARDSIDE accepts VISA and MASTERCARD, certified checks and money orders. Personal checks are accepted (takes 3 weeks to clear). TSE/HARDSIDE pays all shipping charges (within the 48 states) on all PREPAID orders OVER \$100.00. On all PREPAID orders UNDER \$100 a handling charge of \$2.50 must be added.

ALASKA/HAWAII:

All orders from Alaska or Hawaii will be charged actual shipping charges. Orders under \$100.00 are subject to a \$2.50 handling charge in addition to actual shipping charges.

C.O.D.:

C.O.D. orders are accepted for U.S. shipment only (orders over \$250.00 require a 25% deposit). There is a \$5.00 handling charge in addition to normal shipping charges.

PURCHASE ORDERS:

Purchase orders are accepted only from U.S. schools and universities. The minimum purchase is \$100.00. Our terms are NET: 10 days. There is a \$10.00 handling charge plus actual shipping charges.

CANADA/MEXICO:

Canadian and Mexican orders are accepted. However, we cannot ship C.O.D. to Canada or Mexico. The preferable method of payment is by bank credit card (VISA or MASTERCARD). NO PERSONAL OR COMPANY CHECKS. We will accept a bank check if it has been preprinted for payment in U.S. DOLLARS. The handling charge on all Canadian or Mexican orders is \$5.00 plus actual shipping charges.

FOREIGN:

Payment must be by a BANK CHECK drawn on a U.S. BANK, payable in U.S. DOLLARS. We will also accept affiliated bank credit cards of VISA or MASTERCARD. All shipping and duty charges are the customer's responsibility. All overseas orders are subject to a \$10.00 handling charge plus actual shipping charges.

DAMAGE:

Claims must be made within three working days of receipt of product. All requests must be made by telephone to the TSE/HARDSIDE Customer Service Department at 1-603-673-5144.

RETURNS:

Any unauthorized returns will be refused upon delivery. If you wish to return an item for ANY reason, CALL FIRST! All returns must have a Return Authorization Number clearly marked on the outside of the package. Any hardware items to be restocked will be subject to a 10% restocking charge. Any software items to be restocked will be subject to a 20% restocking charge. Any item to be restocked must be in resaleable condition.

SHIPPING:

Orders are normally processed within 24 hours of receipt, with shipping taking place within three to five working days. If the item is out-of-stock and delivery is not expected within ten working days we will send a postcard notifying the customer of the expected delay (in which case there would be a complete refund if the customer wishes to cancel the order).

PRICES:

Prices and specifications are subject to change without notice. We are not responsible for typographical errors including prices.

WARRANTY:

TSE/HARDSIDE Limits the liability of its warranty to the repair or replacement (our option) of a defective product, and can assume no responsibility for the way the products are used. Software products will be replaced on a direct exchange basis, if they are found to be defective within 10 days of receipt. Hardware products carry the warranty listed in the product description. No other warranty is to be assumed.

SERVICE:

HARDSIDE maintains a service department for repair of items it covers by the HARDSIDE warranty. Our service department will also repair items not covered by the

HARDSIDE warranty (or out-of-warranty) on a fee basis. To contact the service department call 1-603-673-5144.

TO ORDER:

You may place your order by calling our toll-free order line at 1-800-258-1790. In New Hampshire call 1-673-5144. The hours are 9 a.m. to 9 p.m. Monday through Friday and 12 noon to 4 p.m. on Saturday. Technical assistance and customer service calls are taken from 9 a.m. to 5 p.m. Monday through Friday on the 1-603-673-5144 number only. When placing an order have your credit card ready. You may also order by mail: TSE/HARDSIDE, 6 South Street, Milford, NH 03055.

ORDER PICK-UP:

TSE/HARDSIDE maintains an order pick-up center in "Computer Haven" at 4 South Street in Milford, NH. It is best to first call "Computer Haven" at 1-603-673-5144, to confirm if the items you wish are in stock. "Computer Haven" hours are from 2:30 p.m. to 5:30 p.m. Monday through Friday and from 10 a.m. to 5 p.m. on Saturday. Please remember that since "Computer Haven" sells all TSE/HARDSIDE products at mail-order prices we cannot always offer all retail or demonstration services at this facility.

READ CAREFULLY

If you live within the Continental U.S. and are prepaying your order: If your order is less than \$100.00, you must include \$2.50 for handling. If your order is more than \$100.00 and you are prepaying your order, we will pay all shipping and handling charges (Continental U.S. only).

CATALOG:

For detailed descriptions of our software, hardware, and supplies send \$2.00 for the TSE Catalog and receive a certificate worth \$4.00 off any purchase.

?? MOVING ??

TO CORRECT OR CHANGE YOUR ADDRESS ATTACH LABEL FROM YOUR LATEST COPY HERE AND PRINT NEW ADDRESS BELOW:

Name.....
Address.....
City.....
State.....
Zip.....

MAIL TO: SoftSide Publications, P.O. Box 68, Milford, NH 03055

Let our fingers do the typing!

With a SoftSide Disk or Cassette subscription you can get each month's TRS-80, APPLE, or ATARI programs delivered on disk or cassette with every issue of your SoftSide. No more hours of typing. No hunting for typing mistakes. The programs for your computer are tested and ready to go the day you receive them.

If you already receive SoftSide magazine, you will receive credit for the remainder of your subscription toward your new cassette or disk subscription.

SPECIAL:

Magazine \$18/year
Magazine and Cassette \$39/6 months
Magazine and Cassette \$75/year
Magazine and Disk \$64/6 months
Magazine and Disk \$125/year

Make your SoftSide library complete with back issues of SoftSide: APPLE, SoftSide: S-80, and PROG/80. Programs, games, complete documentation and lots more! If you have missed out on any past issues, now is the time to order.

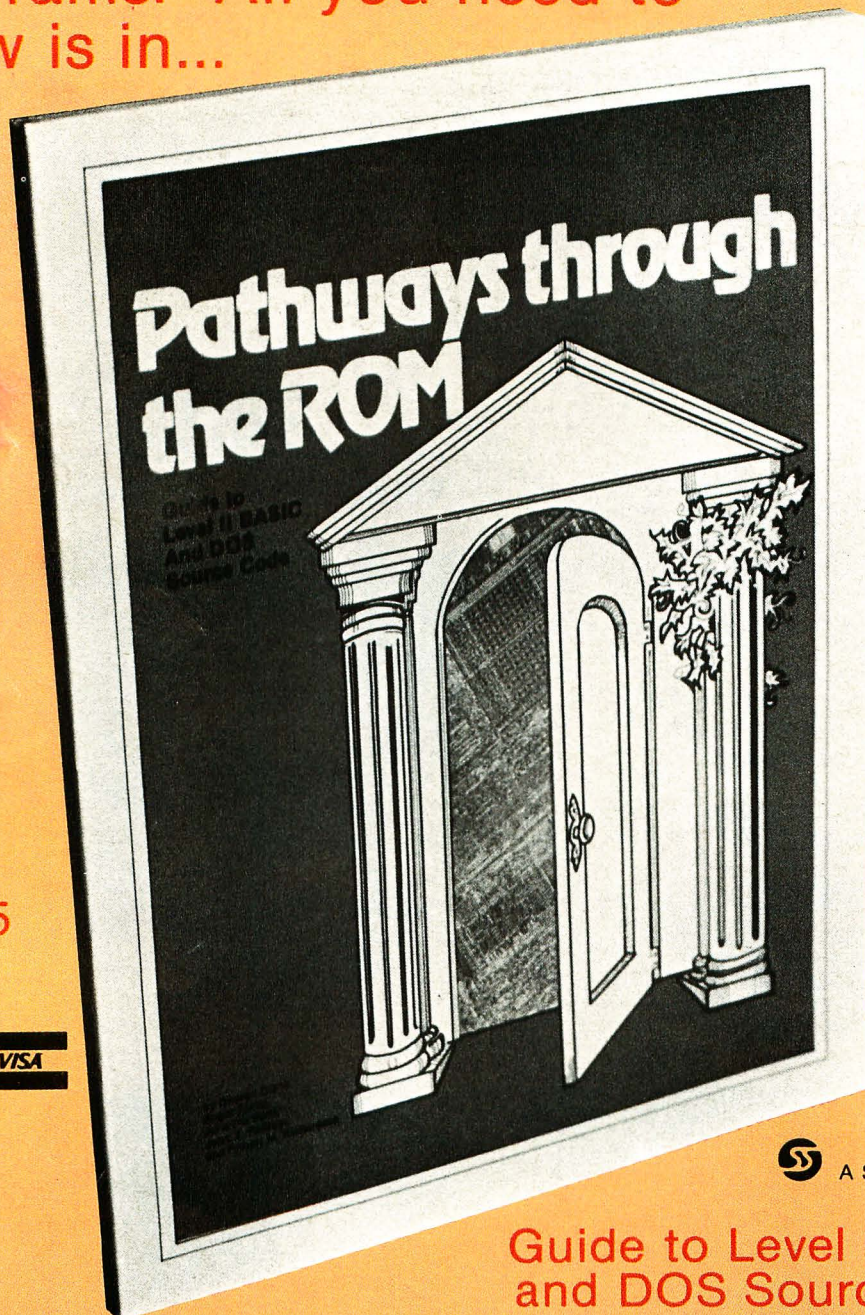
SoftSide: S-80, and APPLE Back Issues (magazine only) \$2.50
PROG/80 Back Issues \$4.00
Magazine with Cassette programs \$9.95
Magazine with Disk programs \$14.95
New Super SoftSide Back Issues (Magazine Only)
August or September, 1980 \$3.00
October, 1980 to present \$3.50


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

SoftSide
cassette & diskette



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...



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

ALL
ONLY
\$19.95

plus \$1 shipping



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.

ORDER TOLL-FREE
(In NH call 673-5144)

1-800-258-1790

The Software Exchange