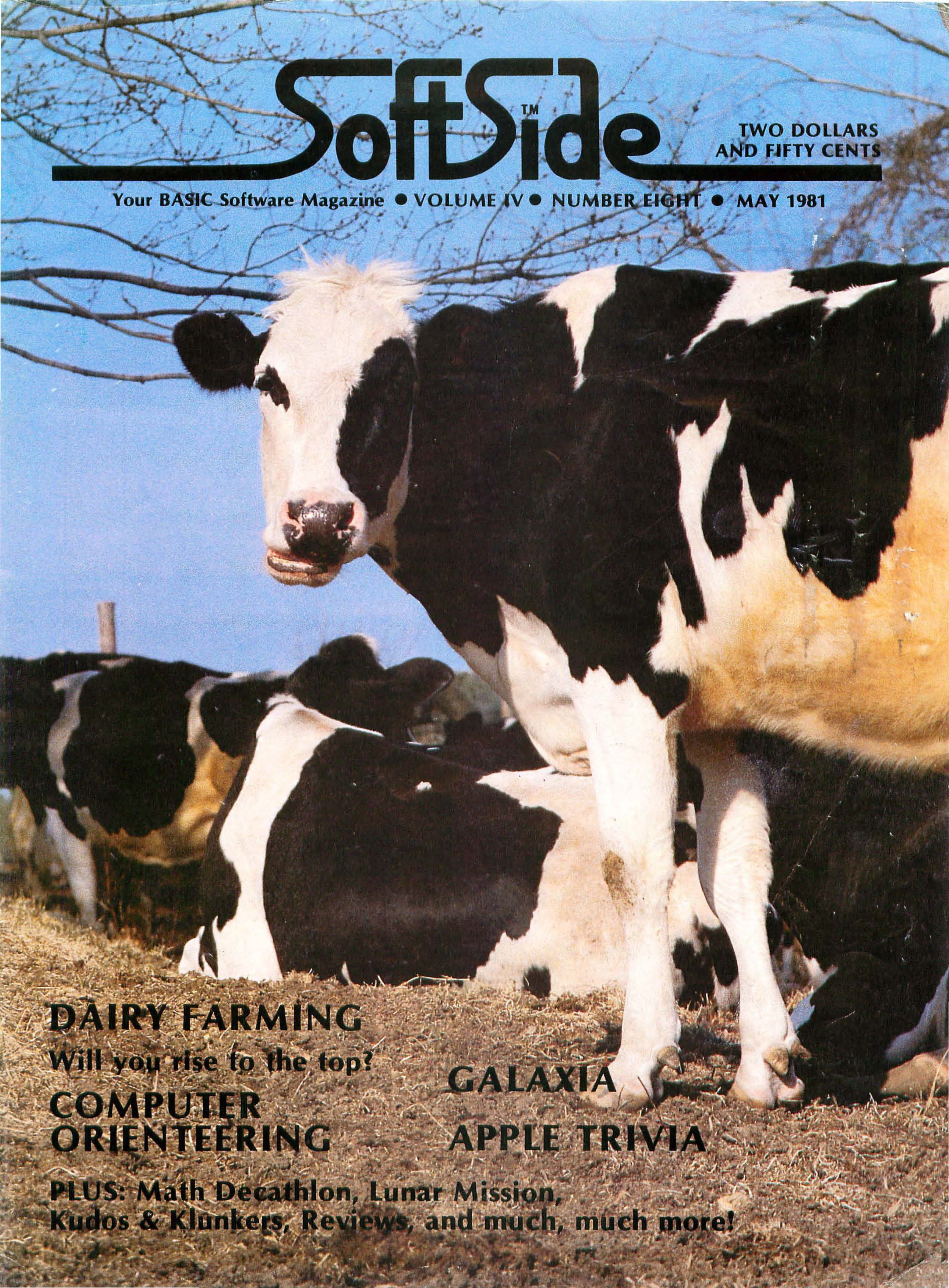


SoftSide™

TWO DOLLARS
AND FIFTY CENTS

Your BASIC Software Magazine • VOLUME IV • NUMBER EIGHT • MAY 1981

DAIRY FARMING

Will you rise to the top?

COMPUTER ORIENTEERING

GALAXIA

APPLE TRIVIA

PLUS: Math Decathlon, Lunar Mission,
Kudos & Klunkers, Reviews, and much, much more!

Hot pursuit through space and the vortices of time!

RAMWARE PRESENTS. . .

Time Lord

The fallen Time Lord, who presumptuously calls himself The Master, is at large.

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

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

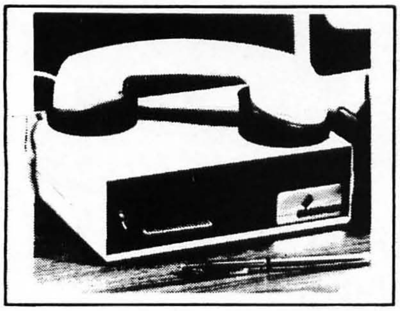
And once you find The Master can you destroy him?

Based on Dr. Who of PBS fame.
Apple Integer Basic,
Disk, 48K . . . $29.95
#47-191001D

MasterCard VISA

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

EXPAND YOUR COMPUTING CAPABILITIES!



THE CHATTERBOX

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

ST80* SMART TERMINAL PACKAGE

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

ST80* UC

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

ST80*

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

ST80* D

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

ST80* III

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

THE SOURCE

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

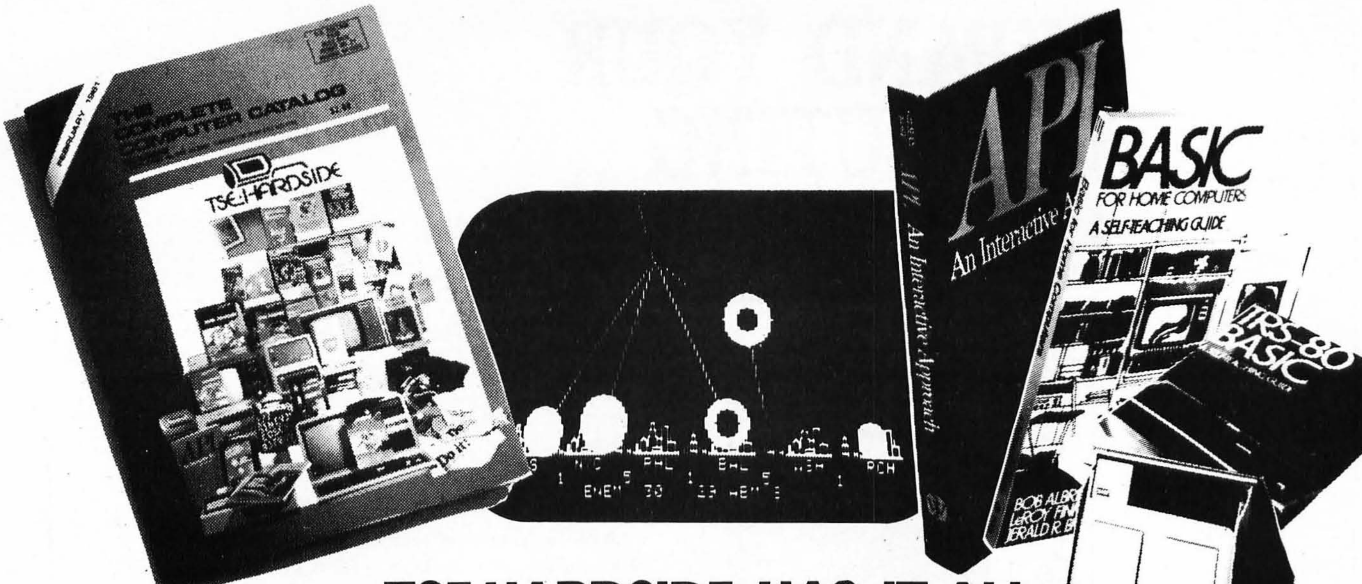
Connect your microcomputer to THE SOURCE! THE SOURCE is a bank of mainframe computers in McLean, Virginia connected by TYMNET and TELEMET data transmission services to hundreds of cities around the country. Most urban areas can access THE SOURCE by local phone call. Costs:

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

MasterCard VISA

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

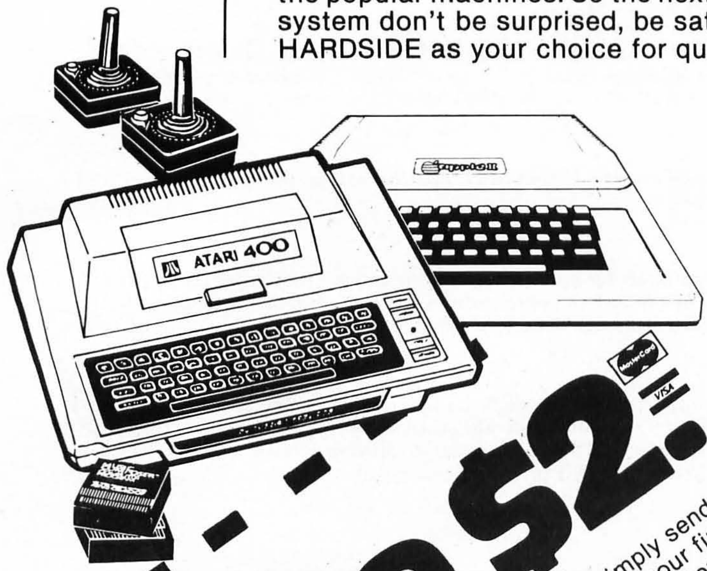
*ST80 is a trademark of Lance Micklus, Inc. TRS-80 is a trademark of Tandy Corp.



TSE·HARDSIDE HAS IT ALL IN ONE!

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

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



Save $2.00

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

Name ____________________

Address ____________________

City __________ State __________ Zip __________

CMAY81

MAIL TO: TSE/HARDSIDE
6 South Street
Milford, NH 03055

TABLE OF CONTENTS

ARTICLES

PROGRAMS

DEPARTMENTS

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

TRS-80 **APPLE** **ATARI** **PET**

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

STAFF

PUBLISHER:

Roger Robitaille Sr.

EDITORIAL DEPARTMENT

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

PRODUCTION DEPARTMENT

Donna Bennett
Cindy Boucher
Elaine Cheever
Lynda Fedas
Damian Henriques

STAFF

Patricia Acampora
Lester Anderson
Ruth Anderson
Judie Aveni
Brian Berkebile
Diana Bishop, **Subscriptions**
Kathleen Boucher
Suzanne Breton
Philip 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
Kathleen Hannon
Pamela Horne
Robert Hunsacker
Dave Hutchings
Donna Jean
Bette Keenan, **Customer Service**
Bea Kimball, **Software Shipments**
Randal Kottwitz
Karen Lawrence
Kathy Maloof
Jean Matthews
Dick Mehlhorn, **Hardware Information/Shipments**
Doris Miller
Clem Morey, **Apple Submissions**
Robin Moss
Glen Ohlund
Mary Reed, **Dealer Information**
Carol Roane
David Robitaille
Elizabeth Robitaille, **Personnel Administrator**
Cindy Schalk
Ken Sicard, **S-80 Submissions**
Christine Spade
Alan Thulander
Joanne Tracy
Anmar William
Nancy Wood
Ed Umlor
Gary Young
Cynthia Zawacki

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

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

EDITORIAL

by Dave Albert

By definition, a simulation is an attempt to imitate life, or some aspect of it. One can trace the roots of simulations back to the dawn of history, when people practiced sympathetic magic in the belief that by acting out the future one could literally shape it. The notion that by approximating an activity one somehow would be more the master of that activity shows up in the 18th and 19th century war simulations where commanders believed that acting out a battle beforehand would help to win it in actuality. Any simulation has the lure that play-acting is somehow analagous to the real thing.

In computing, simulations constitute a genre of games in which the player is placed in someone else's shoes and challenged to succeed. "Hammurabi" is perhaps the best-known microcomputer simulation. We have published several simulations in **SoftSide**, including "Imhotep", "Micro Millionaire", and "IBC". We enjoy simulations and will continue to publish them whenever we run across ones that we like. This month, however, we are publishing a pair of them, each with certain features that bear close scrutiny.

The feature program that lends its theme to our cover this month is "Dairy Farming". It is a nicely-executed simulation that allows you to try to succeed as, you guessed it, a dairy farmer! Success is judged on a point basis, calculated on the land you own, the size and type of your herd, how much milk you sell, etc. Sounds glamorous, huh? Well, not really... which is what makes it an interesting simulation. There's none of this fantastic stuff about becoming chairman of the board or ruler of a city-state, just a simple goal of being a good farmer. You must rely on shrewd business judgment and the ability to know when to choose a certain course of action. There is little whimsical in this program, and even less of the unbelievable.

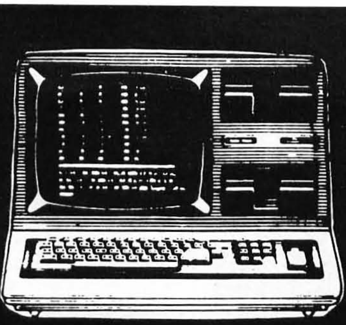
Which brings up the point of this editorial: Is fantasy necessary to a stimulating simulation? In one sense, the answer must be yes, for the whole point of a simulation is to be involved in a make-believe situation, but one must bear in mind the corollary that in so doing, one is somehow learning to master the subject being simulated. Once the point about fantasy as basis is conceded, then where does the line of fantasy versus approximated reality need to be drawn? In "Dairy Farming" a mundane topic is chosen and made demanding, and thus exciting. All kidding aside, "Dairy Farming" is as tough to win at as any simulation I've ever played. But it still places the player in command of resources that many of us only dream about, i.e., enough capital to purchase land and a herd of cows.

The other simulation in this issue goes well beyond that in attempting to approximate objective reality. "Orienteering at Jacque's Coulee" is about as close as you can come to actually going out into the country with a compass and a map and finding your way around. Michael O'Keefe has done an excellent job of first explaining the sport of orienteering and then emulating it on the S-80. Orienteering itself is a sport that requires virtually no equipment, and is thus accessible to all, rather than to the well-off. What O'Keefe has done is to choose a real spot and then put you there via your computer. Your goal is to find your way around this terrain, in much the same fashion as if you were physically there, armed with naught but a compass, map, and your wits. It would be hard to get any closer to the real thing than with his program.

Now it is my contention that what O'Keefe has done is infinitely harder to do than what his simulation-writing predecessors have done... He has applied his imagination in a very controlled and disciplined fashion to produce what is TRULY a simulation, rather than a flight of fantasy masquerading as an approximation of reality. To date, this is the ONLY such simulation I have encountered. Although "Dairy Farming" at least attempts to do something similar, *it* still relies on what are impossible circumstances for the vast majority.

The distinction I'm trying to make is that "Orienteering" really tries to approximate reality by introducing as few fantasy postulates as possible, while the general rule in simulations is to create a reality that is totally

continued on page 85



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.

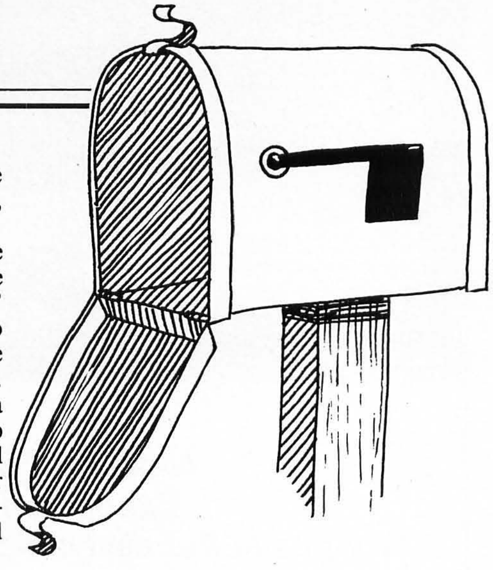
(Registration Badge will be sent to you on May 1st.)

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

Co-sponsored By
80 microcomputing™

(Be sure to enclose
check or money order.)

INPUT



Dear **SoftSide,**
Attn: Lance Micklus

We haved tried to gather enough "intelligence in our little group of phone-order takers to respond to certain statements in your most recent column.

We took your comments personally since you have seen us function and know that TSE was built on the customer service we provide. We pride ourselves on our knowledge here and want all to know that we take the time (our own) to learn about a product because most programmers don't take the time to answer questions for the customers.

We challenge you Lance, to sit in on a day of telephone order taking, to learn first-hand exactly what is required to put money in your pocket. We'll even pay your expenses if you can complete one day handling four WATS lines using many order blanks while keeping your "Apples" from mixing with your "Pets".

We have had many comments from customers complimenting us for the accurate software information we provide versus the lack of support received from a well-known computer store chain.

Taking your theory one step further, you could eliminate an answering service completely by installing a recorder — at the sound of the beep, the customer could specify the order and charge card number.

The Softhearted Women from
TSE (The Software Exchange).
Milford, NH

Dear **SoftSide,**

I have been following the letters, editorials, and articles presented in the last few months, not only in **SoftSide**, but in most of the computer magazines to which I subscribe. Needless to say I have grown weary of this piracy talk.... Please, enough is enough!!

I read Lance Micklus' "My Side of the Page" in January, 1981. His side of the page, "Getting a Bit Serious — Part 1," took three pages explaining marketing to me and that he would be happy to sell me "the" copy of "ST80-III" for a million. Now my point Lance, I don't have a million, but I may be able to come up with a lesser amount for a good terminal program. Why should I buy "ST80-D" or ST80-III"? All I see are full-page multi-thousand dollar ads in all magazines, and what do they say?

"ST80-UC" would cost me $24.95. OK, I already have Radio Shack's Communication Software and Videotex etc., so not knowing any better, I'll pass on "ST80-UC". Next comes "ST80" — reprograms my RS232 board for $49.95. Humm not bad, but why do I need to reprogram it? Pass again. Next is "ST80-D" for $79.95. Well now this could be something, but there are a lot of buzz words here:

User Created Translation Table
Gathering Pre-formated Data
Editing of Received Files
Job Logging
Device Driven I/O
Rub
Bell

Well I have had my system, less modem, for two+ years and most of that still leaves me in the in the dark. Next I see there is "ST80-III" and for $150.00 I can have, I quote, "ST80-D" with extra utility programs unquote. That's $70.05 for utility programs. Now even just one good utility program could be worth $70.05 to me, but remember I have utility programs sliding off my diskettes now so I would be just a little interested in know what these new ones can do.

Now Lance has used three full pages, nine columns, to tell me more about marketing than I cared to know. He could have said all that mattered to me about marketing, in one column, and could have used the other eight columns to sell me a new terminal program. In that article I could have been informed as to the meaning of said buzz words, and extra utility programs, and had an in-depth simulated contact using all those words and utility programs.

Sorry to pick on Lance, but this to me, is having what I am in the market for, but not selling it, just saying "Buy this. Look at the big bucks I spend on ads. It's good. I wrote it."

We all still love ya Lance. Keep writing the good stuff.

Paul Schmid
Orrville, OH

SoftSide™

INPUT POLICY

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

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

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

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

Dear **SoftSide,**

You seem to have missed the main point in my letter published in the 2/81 issue. I fully understand that the published listings are not TYPED in. But they still have errors. Yes, the programs are sometimes very complex and you may have missed some bugs. But that excuse is just a copout. It is admittedly very difficult for someone very close to a project to proofread it. Why not take my suggestion and let some readers try out programs from your "to be published" printout before you publish them. Sure does look like a simple thing to do to avoid a lot of grief on the part of your subscribers.

Your 2/81 issue illustrates my point perfectly. "Mini Golf" for the Atari needs to have " 's turn" added to line 1010. No big deal, but this could have been picked up. "Changing Hearts" has a flawless listing, but an error in the G$ string to be put in line 40. After some analysis I found that the string needs a "control comma" (graphics "heart") added on the end of the string to make the program work. I enjoy this sort of "fixing"but all your readers may not be of the same mind.

Richard Kushner
High Bridge, NJ

Dear **SoftSide,**

I would like to disagree with your response to a letter in the February, 1981, issue of **SoftSide**. The letter in question is on page 7 pointing out errors in programs listed in the magazine. The author pointed out an error in the "Boing!" program that appeared in the November, 1980, issue on pages 82-83. If you would have taken the time to carefully check the program you would have seen that the Atari program is absolutely useless as printed! The end of the program after line 320 is obviously missing. There are many references to line 400 (lines 260, 265, 270) which is NOT given. Also, line 5 attempts to read data from DATA statements that are not listed. How about publishing the complete program.....

Robert W. Baker
Atco, NJ

Editor's Reply: See "Bugs, Worms, and Other Undesirables" on page 13 for the correction to "Boing".

Dear **SoftSide:**

As a charter subscriber, I have received a mixture of pleasure and pain from **SoftSide**. Pleasure, from the interesting articles and useful programs. Pain, from the frustrating debugging of several Atari programs which won't run!

Now, I already know your answer to a subscriber with a similar complaint. That is, that your programs are printed directly from the computer on which they are running. Well, your logic is overwhelming, but repeated proofreading without success has led me to question it.

Proof of my position arrived in the form of Bouncing Barney in "Boing!" (November, 1980, page 83). Three lines in the Atari version (lines 260, 265, and 270) go to a subroutine on line 400, which does not exist. Since the page ends with your SS logo, presumably the program is complete. Indeed, it does not carry over to another page in that issue. The remark preceeding these lines describes the subroutine as the sound Barney makes when he hits either wall.

The missing sound subroutine is not the only problem. When the program is run, there is an immediately an error message indicating that it is out of data in line 5. It also fails to move the trampoline to the right, or erase its image after it's been moved.

Now, I realize that you feel that the mistakes are all on this side of the table. Perhaps the program is letter perfect, and just this one time the end was lost by the printer.

As far as I'm concerned, this is not the case. I suggest that you listen carefully to the "what happened" letters. And, as a regular service, provide a column of updates on previously published programs. Invite your readers to send in corrections and fixes, if they can find them. Believe me, the bugs are there!

Martin W. Molloy. Ph.D
Palo Alto, CA

Editor's Reply: Sorry if we gave the impression that we feel all mistakes come from somewhere else. Yes, we do make mistakes, *"Boing" being* a case in point. And yes, we welcome any and all corrections and fixes, which we carefully check when they come in. Without reader response we wouldn't know where we stand, much less where we're headed.

Dear **SoftSide,**

I have just finished reading a letter from Richard Kushner on page 7 of the February, 1981, issue of **SoftSide**. My feelings after reading this letter are twofold: (1) Complete agreement with Mr. Kushner's remarks concerning **SoftSide**'s errors in Atari programs; and (2) Anger at the dismissal by you of the points he raised in his letter. Actually, to be more precise, you didn't dismiss his remarks, you ignored them.

Let's take a specific program which Mr. Kushner and I have typed in from the November, 1980 issue of **SoftSide**. This is the program "Boing!". This program does not and cannot work as it appears in the magazine!!! Line 5 has a READ statement for which there is no corresponding DATA statement. Lines 260, 265, and 270 have a GOSUB 400 for which there is no line 400. The program ends at line 320. You may have "dumped directly to a line printer" but as Mr. Kushner pointed out and you ignored, you did not proof your copy to see that you had lost part of the program!

Incidentally, we examined the December issue of **SoftSide** looking for corrections to the "Boing!" program and found in the Table of Contents "Bugs, Worms, and other Undesirables" listed as being on page 94, but not only were they not on that page but were not to be found in that issue, and we have never seen a correction to the "Boing!" program.

I agree with Mr. Kushner's comments on "Trench" and "Meteor Storm" since neither of these programs worked as typed in from your magazine listing. We had to debug both programs to get them to execute logically and properly. My conclusion is that you either need to get your act together or adopt Mr. Kushner's solution and let Atari owner's help you insure the proper execution of your listings.

Thomas N. Wheeler
Charleston, WV

Dear **SoftSide;**

Attn: Mark Pelczarski

About "I Don't Think We're in Kansas Anymore":

I think you ain't never bin in Kansas. At least your writin's ain't from Kansas. If you ever growed up in Kansas your parents would have learnt you everything they knowed and you'd git shoved up the crick if you done wrong. So don't claim you was ever in Kansas 'cause you ain't hep enough to understand Kansas lingo.

Andy Anderson
Holton, KS

Dear **SoftSide,**

I received a free copy of your October, 1980, edition of your magazine. I am very impressed at the quality of the programs and articles in **SoftSide**, and I have already sent in my subscription form. However, is there any way to prevent the magazine's covers from getting torn in the mail (this always happens with my other magazines also)? Also, I would like a copy of your author's guide.

Brian Yamauchi
Oxford, OH

continued on next page

OUTGOING MAIL

by Dave Albert

Here we go again. Notice *anything* about the magazine you're holding? It should look somewhat different in that we've changed the format to justified type. The reason for this is that we believe it is easier to read this way, and also it looks a lot tidier. If you have feelings for or against this change, do let us know by writing a letter. We think it's an improvement, but then the ultimate decision lies with you, the reader.

This month I thought I would talk a bit about submissions — how we receive them and how we handle them. **SoftSide** depends on submissions from its readers for its content. In a sense we are a large software club, culling out the best from what we receive and then publishing it for all to use. We don't have a whole lot of control over how the stuff is written or coded. When we see something we like, but that we think could be done better, we either try to improve it if the faults are minor, or we send it back to the original author with an encouraging letter containing our suggestions.

For the sake of handling the amount of material that comes through the doors, we do not accept submissions in a printed form. They must come to us on a cassette or 5¼ inch floppy diskette, ready to load and run. We simply don't have the time or the personnel to type in programs and then debug, edit, and prepare them for printing. When programs arrive on a disk or cassette, but without documentation and variable listings, we curse the authors in a dozen different tongues and stick pins into wax images of what we think they look like.

What we look for in a submission is good programming technique, an attractive and fast-moving (within the limits of BASIC) program, and originality and flair in design. This applies to games, utilities, and any other type of program we receive. We are fond of printing little programs that help with chores, bring the cows home, wash the dishes, or any other of a myriad of useful functions. Of course the wider the application, the better we like the program. While a program that determines rivet placement on the fuselage of supersonic aircraft may be well-written, we feel

continued on page 10

continued from previous page

Dear **SoftSide,**

I have read your magazine from the beginning of **SoftSide: Apple Edition.** In general, it is excellent. But (there always has to be a catch somewhere!) I do have two minimal complaints. Number 1: the lack of Apple adventures that appear (or don't appear,) in your magazine. As I said, this is a very minimal complaint, but it would be nice if there were a few now and then. Number 2: I am afraid to say that the quality of the binding on your new magazine leaves quite a bit to be desired. I find my **SoftSide's** beloved pages falling out and scattering themselves around the house (not too handy when I want to type in a program from a past magazine and one of the pages is missing).

On the whole I find your magazine a pleasure to read and look at, and I highly recommend it to all my friends, with or without a computer. Keep up the good work, and happy SoftSiding (a legal word??!?)!!

Matthew Laurence
Lincoln, MA

Dear **SoftSide,**

I have been following the debate on the new vs. the old **SoftSide** in the "Input" columns. I find I must agree with Brian Thompson, John Trumbull and Clyde Elsasser, at least in part.

I too, liked the old **SoftSide** because it pertained only to the S-80 and I knew I could eventually use every article and program. I have considered not renewing, but I must admit, I'm afraid I'll miss something good. At least you note in the contents page which computer the article covers. That is a help. I have a suggestion for your other subscribers that may soothe some bad feelings.

When my new **SoftSide** arrives, the first thing I do, is to take a "Hi-Liter" pen (those pens college students use to mark passages in their textbooks) and mark each title that pertains to my computer. Then I go to that page and write a big "S-80" on the pages of the article. I use yellow, and you can read through it. Then when I sit down to my serious reading, I read only those pages with the marks. I also mark "Programming Hints", "One Liners", and "Bug Reports" that are for my computer, too. It takes a little extra time at first, but after you get the hang of it, you can do it quickly. At least when you start reading, you don't read something that has nothing to do with your system.

Sherry M. Taylor
Haines City, FL

Dear **SoftSide,**

I have been working with your "Developing Data Base" articles on my S-80, and since you mentioned several times your interest for reader input I thought I would send along the following to you. The TRSDOS disk I happened to be using when I first used your program (up through and including sort) was not write-protected. As you can imagine, although I had the program on drive No. 1, my data was stored on drive zero. In an effort to maintain my "system disk" clean I attempted to input at the end of the file name ":1" in an attempt to have the data written in drive 1. This was unsuccessful since the input query apparently perceives the colon as a delimiter.

Although I am quite unfamiliar with disk BASIC, in quickly reviewing my disk BASIC manual, I came across the command "line input" which does not "see" any delimiter except the enter key. By changing lines 1000, 1500, and 2030, "input" commands to "line input" commands I was able to append a ":1" to my file name and force it to be written on drive 1.

After having done all that it became apparent to me that a much simpler "fix" would have been to write-protect my system disk. Nevertheless, I pass my efforts along for whatever value you find in them.

Keep up the articles. I am particularly interested in print formatting. I intend to apply the data base management program to my office in a number of areas, the one which now has the highest priority is our docketing system.

For this application I am particularly interested in maintaining an audit trail. That is, when the operator inputs, changes, or deletes records, I would like to have a subroutine for printing out the data which was input, the data which was changed, or the data which was deleted.

Stanley B. Green
Washington, DC

Dear **SoftSide,**

Ooops!!!! I was just typing in the Apple version of "Miner" from the February issue of **SoftSide** (Whew!!), when I came across a most regretful error on the part of Mr. Steve Justus. The mistake is very understandable and easy to correct. In fact, the error is so obvious I'm sure you will have many, many letters such as this one. In any case I hate to be the one to point out errors in any program because no one to my knowledge (not a very large knowledge) has ever written a bug-free program, with the possible exception of A.L.F.

The error is on line 480, in Applesoft (or any Apple language) there is no "IF -THEN - ELSE" statement. In order to correct the mistake I just added the following line:

```
485 IF Y< >32 THEN 230
```

No big deal. Except that in line 480 then "ELSE" should be removed. That's all, hope I was of some help!

Randy Reeves
Cypress, TX

Dear **SoftSide;**

Loved the "Meteor Storm" program by Roy Harper, but I thought it would be nice to have something slightly more meteor-like than + signs. These following changes (for S-80 machines) will give a whole screen of random meteor-like objects to fly your spaceship through. Add or change the following lines:

```
115 R2=RND(191): IFR2<128THEN115
120 PRINT@G, " ";:PRINT @RND(62)+960,CHR$(r2): PRINT@0,P0;
150 IFPEEK(L) 1280RPEEK (L+1) >1280RPEEK(L+2) > 128 THEN200
190 P0=P0+1:GOTO115
290 FORI=1TO100:PRINT@RND (62)+960,CHR$(R2):PRINT:NEXT
310 FORI=1TO30:PRINT@RND (62)+960,CHR$(147):PRINT: NEXT:PRINTTAB(23);"METEOR STORM":PRINTTAB: "BY ROY HARPER"
```

William C. Clugston
Lorton, VA

Dear **SoftSide,**

This is to let you know of two bugs in "Tanks-A-Lot" (2/81 issue):

Line 240 — the FOR-NEXT Loop was omitted.

Between PRINT @ I, K;: and PRINT @ I, " ";: the following should be inserted:

For 0 = 1T03: NEXT:

Also in the text, the indicators, for the left commander are in error. They should read:

D = Fire
S = Forward
A = Rotate Right
↓ = Rotate Left

My apologies to the author for tampering with an otherwise excellent program.

Feel free to publish this information if you think it would be of any benefit to your subscribers.

Your magazine gets a ★★★★★rating from me.

It's fantastic. I don't know how you do it but keep doing it.

Raymond Nehilla
Ambridge, PA

Dear **SoftSide,**

I found your recent program "Convoy" (January, 1981) very interesting and entertaining but in view of your policy of requiring a program to run in both Level II and Disk BASIC it seemed unfair that the Apple version allowed multiple convoy play only on the disk version unless the program was modified.

It is nice to see multiple translations of major programs and would like to see this practice continued.

In the process of modifying the Apple translation to run without disk I found a few errors and ommissions.

First was an obvious error which I am surprised has not been reported yet.

6060 IF K > Z * 10000 - 2000 THEN 2973 should be 3060

Omitted was the use of the flag "N" as in the S-80 version line #1110.

```
1150 GOSUB 50010: IF N = 1 THEN VTAB 24: PRINT " YOUR DESTINA TION IS ARCHANGEL": X = 193: GOSUB 50010: GOTO 1240
```

continued on page 10

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.

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.

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

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

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

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

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

Packaging to suit your requirements – standard flip-top box, Kassette 10® storage case, or bulk pack.

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

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

BASF BASF

Floppy Disks Mag Cards Cassettes Computer Tapes Disk Packs Computer Peripherals

INPUT continued from page 8

I am enclosing a method to modify the program for continuous play for non-disk operation which may not be of interest and a list of errors revealed when the program is run without the disk option. The errors are leaving subroutines without using RETURN and leaving FOR-NEXT loops improperly.

```
DEL 3180,3200
DEL 50090,50150
3230 IF I$ = "Y" THEN 200
50160 TS = TS + 1: S = S + .34: IF S > 3 THEN YE = 1942
50170 P2 = 0: P = 0: T = 0: L1 = 0: L2 = 0: L3 = 0: L4 = 0: L5 = 0: L6 = 0: L = 0: RETURN
50300 FOR W = 1 TO 4: GOSUB 50070: COLOR = 3: PLOT Z, Y: GOSUB 50070: COLOR = 0: PLOTZ, Y: NEXT: PA = 1: PR = 0: RETURN
ERRORS:
1530 IF Z = G AND Y = H THEN GOSUB 50300: X = 20: NEXTX: GOTO 2400
1550 IF PA = 1 THEN PA = 0: IF INT (RND (1)*2) = 0 THENZ = G: Y = H: GOSUB 50300: GOTO 2400
1560 IF (T = 4 OR T = 5) AND INT (RND (1)*2) = 0 THEN Z = G: Y = H: GOSUB 50300: GOTO 2400
1650.........PA = 1: W = 16: NEXT W: POP: GOTO 2000
2120 POP: GOTO 2392
2981..........IF Y = 38 THEN ZG = 30: NEXT ZG: GOTO 2983
3160.........NEXT Y: NEXT Z:..........
```

Hilding Anderson, Jr.
Buchanan, NY

Outgoing Mail continued from page 7

its appeal to our readers is somewhat limited.

It usually takes anywhere up to a month subsequent to receipt of a program before we have a chance to thoroughly review it. When we are looking for a particular kind of submission to fill a space in the magazine, we may look at one program sooner than the others that arrived on the same day; but as a rule, we try to get to them as fast as we can.

Once we have decided to go with a program, we contact the author and make an offer. If accepted, we then go ahead and debug the program as far as we are able, splice in the documentation, whip up an illustration of some sort, and publish it. The amount we pay for a submission is determined by the editorial staff, based on the individual merits of a program, cover programs being worth more than shorts, etc.

Often we will not use a submission for several months, until the right spot turns up. So if you send a program in and don't hear from us right away, don't despair.

SAY YOHO

by Scott Adams

They're calling it the massacre at AI computer center! The night I single-handedly wiped out two computers and three printers in less than half an hour. It all started late one night at the lab....

I had just finished adding a new dunning letter feature to our in-house accounting system. The system runs on a 64K S-80 Model II with four drives, and takes care of everything from mailing lists to paying monthly royalties to our authors. The new dunning letter would automatically send out requests for payments to those dealers who were in arrears with us. I had set it up to run on a new Qume Sprint-5 daisy wheel printer we had just recently added to the lab.

Well, I proceeded to hook up the Qume to the Model II to check out the program and the first thing I knew the printer was spitting out garbage. I disconnected the Model II from the Qume and hooked it back up to the Anadex 9501 that it normally uses.

I then connected the Qume up to the Model I to see if it was a compatability problem between the printer and the Model II. It wasn't! Hooked up to the Model I, the Qume once again started spraying out garbage! At this point I realized that the Qume was dead, so I disconnected it and set it aside.

About this time Jeffe Jessee stopped by to get a listing of a new program he was working on, so I proceeded to hook the Atari system up to the Anadex 9501. When we tried to list his program, however, the Anadex stood mute! I then quickly connected the Model II back to the Anadex to see if the problem was in the Atari. It wasn't (at least not entirely, it turned out). The Anadex had nothing to say when hooked up to the Model II or the Model I. I then connected the Model II to a Quick printer to at least verify that the Model II was still working... It wasn't.

At this point I decided I could:

1) Try hooking up the Apple to the Anadex to see if the printer was really broken;

2) Try hooking up the Atari to the Quick printer to see if it was broken; or

3) I could go home.

I went home.

The next day we surveyed the damage and found the following units were indeed shot:

1) Qume Sprint-5,
2) Model II printer port,
3) Atari interface module,
4) Anadex 9501 printer, and
5) Centronics *quick-printer*.

What a disaster! I had single-handedly done in half the lab! Later as we reconstructed the mess we realized what must have happened.

1) The Qume was plugged into a Radio Shack line filter. It turns out this is a definite NO-NO and is what wiped out the Qume.

2) The Qume then took out the printer port on the Model II.

3) The Model II then axed the Anadex printer.

4) The Anadex then gave the old heave-ho to the Atari and, as a final coup-de-grace;

5) The Model II killed the Quick printer!

Ah well, it could have been worse, I could have tried hooking the Apple up too!

And now the good news: We have some new programs available for the Atari! A real-time "Lunar Lander" game, "Galactic Empire", and, at last, "Adventures"! "Lunar Lander" and "Adventures" require a minimum of 24K to run and "Galactic Empire" requires 32K; I'm sure if you flip through you'll find ads for them somewhere in this issue of **SoftSide** so for now I'll just SAY YOHO.......

ABOUT THIS ISSUE

As you can see from our friend on the cover, this month's feature program is an udder delight! It's "Dairy Farming", a simulation by Scott Tapley. We wouldn't steer you towards this one if we didn't think it was a veal good program. Why we munchkins could play it until the cows come home, so to speak. Don't be cowed by the length of the listing; it's fun. We think it beefs up the magazine's appeal in the heartland. (Okay Fred, you're milked this one for all it's worth, get on with telling the folks what's between the covers, already!)

For those of you S-80 owners who are fond of getting lost in the woods every time you go outdoors, we have just the program for you: "Computer Orienteering". Now you can get lost in the woods in the comfort of your own living room!

Wynne Keller from Downeast Digital in Maine has decided to grace our pages with a quick review of the Model III TRSDOS. Called "Kudos and Klunkers", her article explores the abilities of Tandy's latest, warts and all.

And who might the mysterious "J" be? Or the Klibanesque blonde lurking behind his disk drives, for that matter...No one told us munchkins there was just this strange package in the Editorial mail file with a scented card attached suggesting what to do with it... We decided to print it instead.

Here's a chance for you Apple folk to stop pouring quarters down the slot at your local arcade. We present "Galaxia", by Michael Prescott. Cut from the same cloth, we also have "Bombardment" for the Atari, courtesy of Alan Hartman, and an interesting lunar lander for the Atari called "Lunar Mission", by Matt Rutter.

Monsieur Granite Knoggin explores the world of lineprinters this month, telling us how to make the Centronics, Microline, and Epson printers do tricks. The only trick the munchkin printer knows is how to roll over and play dead, however.

And of course there's the tale of woe from that Yoho fellow in Florida...it seems he singlehandedly wiped out a computer lab in a frenzy of chain reaction. Plus more on computer graphics, on how these machines work, and scads of reviews.

'Til next month, a merry munchkin farewell.

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.

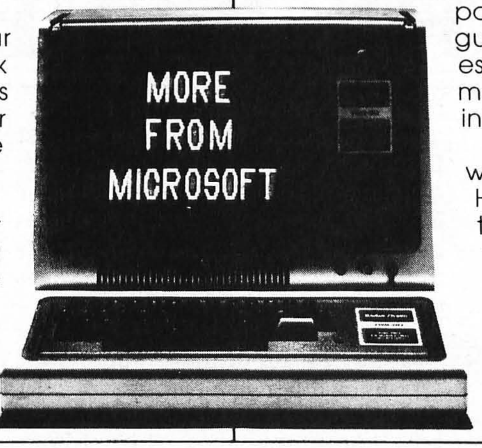
Simply type in and debug your program as usual, using the BASIC interpreter. Then enter a command line telling the computer what to compile and what options to use.

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

MasterCard VISA

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

WORD SEARCH PUZZLE #2

In April we gave you a word-search puzzle embedded with all sorts of computer-related words like "error", "peripheral", and "subroutine". If you had fun with that one, you may find this one ever more of a challenge, because (heh, heh) you don't get a word list this time. However, that should be no problem for you computer-literate **SoftSide** readers, because all 67 of the embedded words (at least all of them that we intentionally put in) are BASIC keywords, commands, and functions.

We'll give you the answer key and the word list next month, and later on we'll even give you the program by David Durkee that generated both of these puzzles, so that you can create them on your own to your heart's content!

By the way the solution to last month's puzzle is on page 33.

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

BUGS, WORMS,

and other undesirables

In the S-80 version of "Strategy Strike" published in the March issue of **SoftSide**, line 80 should be changed to:

```
80 NL$="1234567890123456789112345б7":E
9=VARPTR(NL$):E8=PEEK(E9+1)+256*PEEK(E
9+2):FORE7=E8 TO E8+26:READE6:POKE E7,
E6:NEXT:DATA 205,127,10,77,68,62,1,105
,211,255,45,32,253,13,16,238,175,211,2
55,201
```

In the Atari version of "Boing" published in the November issue of **SoftSide**, the following line should be added:

```
400 SOUND 1,200,10,8:FOR TIME=1 TO 10:
NEXT TIME:SOUND 1,0,0,0:RETURN
450 DATA 32,17,18,18,18,18,5,32,32
```

I would like to clear up the confusion surrounding my article, "Atari Memory Upgrade" which appeared in the March issue of **SoftSide**. First, there is an error in step 6. The memory board should always be inserted with the SOLDER side of the board facing toward the front of the computer, not the component side as was indicated in the article.

Second, I have heard from readers that some 8K modules have the jumper resistors installed in different locations from those on my two 8K boards. This has caused some confusion when readers tried to "remove resistor B" only to find that Atari had beaten them to it! Since writing the article, I have obtained from Atari the schematics for the 8K and 16K modules. (The schematics are contained in the book "Atari Personal Computer System User's Manual and Hardware Manual", Atari part number CO16555, a hefty volume which will fascinate the technically-minded and prove invaluable to Assembly Language programmers.) The schematic for the 8K board indicates that there are apparently four different 8K boards made by different manufacturers, and all bearing the same Atari part number! The only difference between the versions that I have been able to discern is in the placement of the two jumper resistors.

My interpretation of the schematics is that the modification described in the article should work for all four versions of the Rev. 3 board. In step 7, where you are told to remove the jumper resistors at "B" and "D", you should remove resistors "A", "B", "D", "E", and "F" (NOT "C"!) if present, and install one at "C" if there is not one there already. Perform the other steps as indicated (be sure to note the correction to step 6 above).

I must stress that I have NOT been able to try this modification on the other three versions of the board, as I have never seen one, but it appears from the schematics that there should be no problem.

Finally, I have heard rumors that the 8K Atari 400 contains the same 8K board inside it as the 800 (less the plastic case). If this is true, it should be possible to upgrade a 400 to 16K using the same techniques.

Paul Johnson

BOMBARDMENT

by Alan Hartman

"Bombardment" is an Atari program requiring 16K and a joystick.

Can you save your bases from the enemy's barrage of missiles? You control a modern ship with a powerful anti-missile cannon which you can use to destroy the enemy's incoming missiles, but you must be quick to catch them all before they destroy your bases with their explosive charges. The enemy keeps up the attack, gaining speed, and making it harder and harder to protect your bases. How long can you survive? And to make your job even harder, some of the enemy missiles are MIRVs (Multiple Independently targetable Reentry Vehicles, which can split up into more missiles.

You will have 15 missiles at the start of each round. The four bases you are assigned to protect are at the top of the screen, and your ship is in the middle of them. To destroy the enemy missiles rising from the bottom of the screen, move the joystick to position the crosshairs ahead of the missile. Press your fire button, and an explosion will occur. If the enemy missile hits the explosion, it will be destroyed. You must keep your defense going until all missiles have been destroyed.

For each enemy missile you destroy, you are awarded 25 points multiplied by the difficulty level. For each remaining missile of your 15, you get 5 points times the difficulty level. For each remaining base, you get 100 points times the difficulty level. The object of the game is to get the high score. GOOD LUCK!

VARIABLES

CI(A, B): Table for city and ship data (see below).
JB: Value from fire button.
JS: Value from joystick.
MI (A, B): Table for enemy missile data (see below).
N: Number of enemy missiles.
SC: Score.
SH: Missiles left to fire.
X, Y: Position of crosshairs.
A, B, C, D, E, I, T: Temporary usage: loops, timing delays, etc.

TABLE DATA
(variables MI(A, B) and CI(A, B)

MI(0, A): 1 = Active missile, 0 = destroyed.
MI(1, A): Present X location of missile.
MI(2, A): Present Y location.
MI(3, A): X offset to target.
MI(4, A): Y offset to target.
MI(5, A): Starting X location (used for erasing).
MI(6, A): Starting Y location.
CI(0,A): 1 = Active base, 0 = destroyed.
CI(1, A): X location of base.

DOCUMENTATION

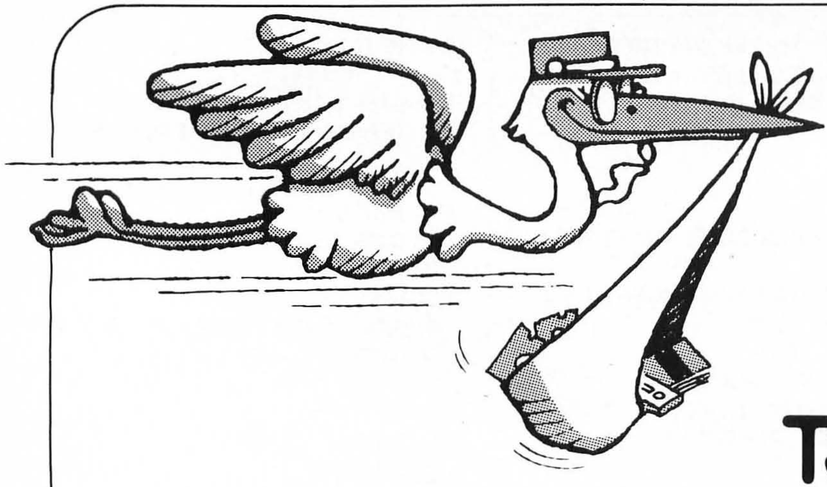
Lines 30-70: Initialize the tables for the bases and the ship.
Lines 80-90: Check to see if there are any bases left. If not, then go to the end of game routine.
Lines 100-130: Build a graphic mode with one line of text at the top of the screen, the 92 lines of graphics mode 7.
Lines 140-150: Place the remaining bases onto the screen.
Lines 160-220: Initialize the table for the enemy missiles.
Lines 230-350: Erase old crosshairs, process the joystick movements, draw the new crosshairs, and repeat.
Lines 360-570: Take care of the explosions of the missiles.
Lines 680-690: Move the enemy missiles, check to see if they hit anything, and deplete your missile supply if your ship was hit.
Lines 700-790: Erase a base hit by an enemy missile.
Lines 800-820: Update the score when an enemy missile is destroyed and erase the old missile.
Lines 830-840: Check to see if all missiles have been destroyed.
Lines 890-900: Draw the bases.
Lines 910-1010: Update the score at the end of one round.
Line 1020: Draws the crosshairs on the screen.
Lines 1030-1047: End of game display. Allow player to start a new game.
Lines 1050-1090: Set the colors depending on the difficulty level.
Lines 1100-1210: Break an enemy missile into more missiles.
Lines 1220-1250: Print the message to start each round.

```
5 REM *** BOMBARDMENT ***
6 REM BY Alan Hartman
7 REM
10 DIM CI(1,5),MI(6,15):M=1

20 X=INT(RND(0)*155+2):Y=INT(RND(0)*77
)+5:SC=0
30 DATA 26,52,104,130,79
40 RESTORE 30

50 FOR I=1 TO 5:READ T
60 CI(0,I)=1:CI(1,I)=T
70 NEXT I
80 FOR C=1 TO 4:IF CI(0,C)=1 THEN 100

90 NEXT C:GOTO 1030
100 GOSUB 1220:SP=0:GRAPHICS 23:POKE 7
52,1
110 I=PEEK(560)+PEEK(561)*256+4:POKE I
-1,66:POKE I+93,65:POKE I+94,PEEK(560)
:POKE I+95,PEEK(561)
115 REM IN LINE 120 POSITION 19,0 SHOU
LD HAVE A INVERSE 'CONTROL J & H' TO F
ORM A TRIANGLE POINTING DOWN

120 POKE 87,0:POSITION 19,0:PRINT #6;"
";:POSITION 2,0:PRINT #6;"SCORE : ";
SC;

130 GOSUB 1050:POKE 87,7
140 FOR T=1 TO 4:IF CI(0,T)=1 THEN C=1
:D=CI(1,T):GOSUB 890
150 NEXT T:SH=10:POKE 87,0:POSITION 22
,0:PRINT #6;"MISSLES LEFT : ";SH;:POKE
 87,7


160 N=INT(RND(0)*M*2)+4:IF N>10 THEN 1
60
170 GOSUB 1020
180 FOR I=1 TO N
190 MI(0,I)=1:MI(5,I)=INT(RND(0)*158)+
1:MI(6,I)=92
200 T=INT(RND(0)*5)+1:A=CI(1,T)+RND(0)
*5*-1^(INT(RND(0)*2)+1):D=SQR((A-MI(5,
I))^2+91^2)

210 MI(3,I)=(A-MI(5,I))/D:MI(4,I)=91/D
:MI(1,I)=MI(5,I):MI(2,I)=MI(6,I)
220 NEXT I
230 FOR JT=1 TO 9-M
240 SOUND 0,X+Y,10,4
250 JS=STICK(0):JB=STRIG(0):IF JB=0 TH
EN GOSUB 360

260 IF SH=0 THEN GOSUB 580:GOTO 260
270 IF JS=15 THEN GOTO 340
280 LOCATE X,Y,I:IF I=2 OR I=0 THEN CO
LOR 0:PLOT X-2,Y:DRAWTO X+2,Y:PLOT X,Y
-2:DRAWTO X,Y+2
290 IF JS>8 AND JS<12 THEN X=X-1:IF X<
3 THEN X=157
```

```
300 IF JS>4 AND JS<8 THEN X=X+1:IF X>1
57 THEN X=3
310 IF JS=9 OR JS=13 OR JS=5 THEN Y=Y+
1:IF Y>90 THEN Y=6
320 IF JS=10 OR JS=14 OR JS=6 THEN Y=Y
-1:IF Y<6 THEN Y=90


330 LOCATE X,Y,I:IF I=0 THEN GOSUB 102
0
340 NEXT JT
350 GOSUB 580:GOTO 230
360 IF SH=0 THEN GOSUB 1030:GOTO 910
370 POKE 87,0:POSITION 22,0:SH=SH-1:PR
INT #6;"MISSLES LEFT : ";SH;" ";


380 POKE 87,7:COLOR 3:PLOT 79,1:DRAWTO
 X,Y:COLOR 0:PLOT 79,1:DRAWTO X,Y
390 FOR C=1 TO 100 STEP 3:SOUND 0,C,10
,6:NEXT C:SOUND 0,0,0,0
400 FOR E=0 TO 5:A=X-E:B=X+E:C=Y-E:D=Y
+E
410 IF A<0 THEN A=0
420 IF B>159 THEN B=159
430 IF C<4 THEN C=4


440 IF D>92 THEN D=92
450 SOUND 0,6,0,E+3:SOUND 1,21,0,E+3:S
OUND 2,27,0,E+3:SOUND 3,40,0,E+3
460 COLOR 3:PLOT A,C:DRAWTO B,C:DRAWTO
 B,D:DRAWTO A,D:DRAWTO A,C:GOSUB 580
470 NEXT E


480 FOR E=5 TO 0 STEP -1:A=X-E:B=X+E:C
=Y-E:D=Y+E
490 IF A<0 THEN A=0
500 IF B>159 THEN B=159
510 IF C<4 THEN C=4
520 IF D>92 THEN D=92


530 COLOR 0:PLOT A,C:DRAWTO B,C:DRAWTO
 B,D:DRAWTO A,D:DRAWTO A,C:GOSUB 580
540 SOUND 0,6,0,E:SOUND 1,21,0,E:SOUND
 2,27,0,E:SOUND 3,40,0,E
550 NEXT E
560 GOSUB 1020

570 RETURN
580 POKE 77,0:FOR I=1 TO N:IF MI(0,I)=
0 THEN 680
590 IF MI(2,I)<60 AND RND(0)<0.1 AND S
P<>1 THEN SP=1:GOSUB 1100
600 FL=1:COLOR 2:PLOT MI(1,I),MI(2,I)


610 MI(1,I)=MI(1,I)+MI(3,I):MI(2,I)=MI
(2,I)-MI(4,I)
620 IF MI(2,I)<1.5 THEN MI(2,I)=1
630 LOCATE MI(1,I),MI(2,I),A:IF A<>0 T
HEN ON A GOTO 700,670,800
640 IF MI(2,I)=1 THEN IF (MI(1,I)>82 O
R MI(1,I)<76) THEN 810
```

```
650 IF MI(2,I)=1 THEN SH=0:FOR C=50 TO
 250:SOUND 0,C,10,8:NEXT C:GOTO 810
670 DRAWTO MI(1,I),MI(2,I)
680 NEXT I
690 RETURN


700 MI(0,I)=0:FOR B=1 TO 4:IF CI(0,B)=
0 THEN 780
710 IF MI(1,I)<CI(1,B)-4 OR MI(1,I)>CI
(1,B)+4 THEN 780
720 FOR T=0 TO 9
730 SOUND 0,99,8,8:SOUND 1,75,8,8:SOUN
D 2,60,8,8:SOUND 3,20,8,8
740 C=2:D=CI(1,B):GOSUB 890

750 FOR C=0 TO 3:SOUND C,0,0,0:NEXT C:
C=0:GOSUB 890
760 NEXT T
770 CI(0,B)=0:GOTO 810
780 NEXT B:GOTO 810
790 GOTO 680
800 SC=SC+INT(M)*25:POKE 87,0:POSITION
 2,0:PRINT #6;"SCORE : ";SC;:POKE 87,7
:SOUND 0,30,10,10

810 COLOR 0:PLOT MI(5,I)+1,MI(6,I):DRA
WTO MI(1,I)+1,MI(2,I):PLOT MI(5,I),MI(
6,I):DRAWTO MI(1,I),MI(2,I)
820 PLOT MI(5,I)-1,MI(6,I):DRAWTO MI(1
,I)-1,MI(2,I):MI(0,I)=0

830 FL=0:FOR NU=1 TO N:IF MI(0,NU)=1 T
HEN FL=1
840 NEXT NU:IF FL=0 THEN 910
850 FOR T=1 TO 4:IF CI(0,T)=1 THEN C=1
:D=CI(1,T):GOSUB 890
860 NEXT T


870 IF SH=0 THEN POKE 87,0:POSITION 22
,0:PRINT #6;"MISSLES LEFT : ";SH;" ";:
POKE 87,7
880 GOTO 680
890 COLOR C:PLOT D-4,1:DRAWTO D+4,1:PL
OT D-3,2:DRAWTO D+3,2:PLOT D-2,3:DRAWT
O D+2,3
900 RETURN


910 FOR T=0 TO 3:SOUND T,0,0,0:NEXT T
920 IF SH=0 THEN 970
930 FOR T=SH TO 1 STEP -1
940 POKE 87,0:POSITION 22,0:PRINT #6;"
MISSLES LEFT : ";T-1;" ";
950 SC=SC+5*INT(M):POSITION 2,0:PRINT
#6;"SCORE : ";SC;


960 SOUND 0,60,10,6:FOR C=1 TO 75:NEXT
 C:SOUND 0,0,0,0:FOR C=1 TO 25:NEXT C:
NEXT T
970 FOR T=1 TO 4:IF CI(0,T)=0 THEN 101
0
980 POKE 87,7:D=CI(1,T):C=0:GOSUB 890:
SC=SC+100*INT(M):POKE 87,0
```

```
990 POSITION 2,0:PRINT #6;"SCORE : ";S
C
1000 SOUND 0,120,10,6:FOR C=1 TO 150:N
EXT C:SOUND 0,0,0,0:FOR C=1 TO 50:NEXT
 C

1010 NEXT T:POKE 87,7:M=M+0.5:GOTO 80
1020 COLOR 2:PLOT X-2,Y:DRAWTO X+2,Y:P
LOT X,Y-2:DRAWTO X,Y+2:RETURN
1030 GRAPHICS 1+16:POSITION 0,11:PRINT
 #6;"FINAL SCORE : ";SC;
1035 POSITION 4,6:PRINT #6;"bombardmen
t";

1040 POSITION 0,13:PRINT #6;"congratul
ations !!!";
1045 POSITION 2,22:PRINT #6;"PRESS 'fi
re' FOR";:POSITION 5,23:PRINT #6;"NEXT
 GAME!";:S=1
1046 SOUND 0,S,10,4:IF STRIG(0)=1 THEN
 S=S+3:GOTO 1046
1047 RUN
1050 DATA 0,1,12,4,3,12,4,8,5,4,8,11,4
,8,5,12,8,6,11,3


1060 RESTORE 1050
1070 A=INT(M)-INT(INT(M)/5.01)*5-1:IF
A<>0 THEN FOR T=1 TO 4*A:READ C:NEXT T
1080 READ C:SETCOLOR 4,C,0:READ C:SETC
OLOR 0,C,4:READ C:SETCOLOR 1,C,6:READ
C:SETCOLOR 2,C,2
1090 RETURN
1100 A=INT(RND(0)*3)+1:FOR B=1 TO N:IF
 MI(0,B)=1 THEN 1150
1110 A=A-1:MI(0,B)=1:MI(5,B)=MI(1,I):M
I(6,B)=MI(2,I)


1120 T=INT(RND(0)*5)+1:C=CI(1,T)+RND(0
)*5*-1^(INT(RND(0)*2)+1):D=SQR((C-MI(5
,B))^2+(MI(6,B)-1)^2)
1130 MI(3,B)=(C-MI(5,B))/D:MI(4,B)=(MI
(6,B)-1)/D:MI(1,B)=MI(5,B):MI(2,B)=MI(
6,B)
1140 IF A=0 THEN RETURN
1150 NEXT B
1160 IF N+A>15 THEN A=15-N
1170 FOR B=N+1 TO A+N:N=N+1

1180 MI(0,B)=1:MI(5,B)=MI(1,I):MI(6,B)
=MI(2,I)
1190 T=INT(RND(0)*5)+1:C=CI(1,T)+RND(0
)*5*-1^(INT(RND(0)*2)+1):D=SQR((C-MI(5
,B))^2+(MI(6,B)-1)^2)
1200 MI(3,B)=(C-MI(5,I))/D:MI(4,B)=(MI
(6,B)-1)/D:MI(1,B)=MI(5,B):MI(2,B)=MI(
6,B)
1210 NEXT B:RETURN


1220 GRAPHICS 1+16:POSITION 6,6:PRINT
#6;"PLAYER #1":POSITION 4,9:PRINT #6;I
NT(M);" * THE VALUE"
1230 POSITION 5,20:PRINT #6;"press 'fi
re'"
1240 IF STRIG(0)=1 THEN 1240
1250 RETURN
```



NEW ARRIVALS AT:

MasterCard VISA

TSE-HARDSIDE

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

RAMWARE™

PASCAL-80

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 (test and record oriented), n-dimensional ARRAY (and ARRAY of ARRAY, etc.), global GOTO, ELSE in CASE statements, BCD arithmetic accurate to a full 14 places (including log and trig functions), and 6 digit optional.
S-80 32K Disk #26-141001D $99.95

Strategic Simulations

COMPUTER AIR COMBAT

You'll swear your keyboard has melted into the throttle, rudder, altimeter, and other cockpit instrumentation of a World War II combat plane. In fact all 36 famous fighters or bombers are rated for firepower, speed, maneuverability, damage-tolerance, and climbing and diving ability. Play the computer or a human.
Apple II 48K Disk #47-219001D $59.95

COMPUTER NAPOLEONICS

You are on the battlefields of Waterloo on the fateful day of June 18, 1815. Here, the greatest battle ever fought is about to begin, awaiting only your commands to set the amassed armies in motion.

You and your friend choose your role — either as the military genius, Napoleon, or as the Duke of Wellington, the iron-willed leader of the Anglo-Allied forces.

The Computer, in the solitaire scenario, plays Wellington while you play Napoleon. "Computer Napoleonics" has all the advantages of a traditional wargame — meticulous detail, realism, and playability, and because the computer keeps track of all the rules, neither player can make an illegal move. This makes learning it a cinch. Comes with the game program, two mapboard cards, a rule book, and two player-aid charts.
Apple II 48K Disk #47-219004D $59.95

OPERATION APOCALYPSE

Features a 7 by 18 hexagonal mapboard generated by Hi-Res graphics and color to represent rivers, bridges, hills, towns, and forests. Incorporating four different scenarios, this game has everything wargamers love: paratroopers, pre-plotted artillery bombardment engineers, tanks, commandos, and infantry.
Apple II+ 48K Disk #47-219005D $59.95

COMPUTER QUARTERBACK

You are propelled onto the playing field of the Super Bowl. From its multiple offensive and defensive plays and its real-time playing conditions to the animated video display of the gridiron and the halftime statistics, no strategy football game has ever been more complete in detail or as exciting in realism. Three versions are offered: Semi-Pro, Pro, and Computer-as-Opponent.

"The Robots" is the team most ably coached by your friendly computer. It's ready to play any time you are. It even "learns" your tendencies and patterns through time, and it will make the necessary tactical adjustments to provide you a tough and well-fought battle.
Apple II 48K Disk #47-219003D $39.95

COMPUTER CONFLICT

"Computer Conflict" actually consists of two fast-paced, action-packed wargames played on full-color mapboards of Hi-Res graphics: "Rebel Force" and "Red Attack".

"Rebel Force" puts you in the role of a Soviet commander whose regiment must face a computer-directed guerrilla uprising which has overrun a vital town.

"Red Attack!" simulates an invasion by a mixed Soviet tank and infantry force against a defending battalion. As the defender, your task is to deploy your infantry units effectively to protect three crucial towns that must not fall!
Apple II 48K Disk #47-219002D $39.95

CARTELS & CUTTHROATS

A business game that is challenging, informative, and above all else, fun. Allowing for up to six players (of which a maximum of five can be played by the computer), each player manages a manufacturing firm in direct competition against the other players.
Apple II+ 48K Disk #47-219007D $39.95

Hayden Book Company, Inc.

BASIC COMPUTER PROGRAMS FOR THE HOME

Contains over 75 practical home application programs that will be helpful to the novice or experienced owner in increasing the usefulness of any home computer. Each program is documented with a description of its functions and operation, a listing in BASIC, a symbol table, sample data, and one or more samples. Programs included are: Home Financial Programs; Automobile Related Programs: Kitchen Helpmates; Scheduling Programs for Home Use; and more.

#65-210001B $9.95

HOW TO PROFIT FROM YOUR PERSONAL COMPUTER: PROFESSIONAL, BUSINESS, AND HOME APPLICATIONS

Describes the uses of personal computers in common business applications, such as accounting, managing inventory, sorting mailing lists, and many others. The discussion includes terms, notations, and techniques commonly used by programmers.

#65-210003B $10.75

COMPUTERS IN ACTION: HOW COMPUTERS WORK, Second Edition

The books answers the questions: "What are computers?" "How do computers work?" and "What can a computer do?" New features of this second edition include: microcomputers and microprocessors; new memory devices such as floppy disk; new input/output units such as COM units; and a section on structured programming, the APL language, and the RPG language.

Numerous photographs, diagrams, and cartoons illustrate specific concepts and equipment.

#65-210007B $7.25

THE 6800 MICROPROCESSOR: A SELF-STUDY COURSE WITH APPLICATIONS

Includes introduction to the 6800, simple programming, introduction to Motorola 6800 Input/Output, using the Micro-68 displays, using the Micro-68 keyboard, handling data arrays on the Micro-68, and much more. Includes 15 laboratory experiments.

#65-210008B $9.55

THE FIRST BOOK OF MICROCOMPUTERS: THE HOME COMPUTER OWNER'S BEST FRIEND

The author explains what personal computers are and what you can do with them. You'll learn how to speak the language of computing and how a personal computer is constructed and how it works.

Then, the last chapters detail the many home and business applications of your personal computer, including such things as home protection, keeping track of budgets and bills, game playing, inventory management, and tax calculation.

#65-210002B $5.95

AH QUALITY from AVALON HILL

CONFLICT 2500

The year is 2500 A.D., and earth is threatened by marauding "Planet Pulverizers" looming on the horizon to blow up earth. The computer takes the part of the attacking aliens programmed with an infinite number of attack strategies with which to tease the player, who must defend earth against such attack. Solitaire only.

S-80 Cassette #26-237001C $15.00
Apple II Cassette #47-237001C $15.00
Atari Cassette #36-237001C $15.00

LORDS OF KARMA

The player finds himself in a mythical, magical city to perform as many deeds of kindness and bravery as possible. The fun is in the gradual deciphering of the many secrets while exploring verdant forests, twisting trails, rugged mountains, and labyrinthine caverns...and avoiding lurking monsters. Almost no rules to learn; you tell the computer what you want by typing simple English sentences on the keyboard. The computer responds with many surprises in store. Solitaire only.

S-80 Cassette #26-237003C $20.00
Apple II Cassette #47-237003C $20.00
Atari #36-237003C $20.00

COMPUTER ACQUIRE

Object is to become the wealthiest person in this business game about hotel acquisitions and mergers. When played as a two to six player game, it is a subtle game of strategy. As a solitaire game, one plays against the computer that has been programmed to play "smart". One can even pit the computer against itself.

S-80 Cassette #26-237002C $20.00
Apple II Cassette #47-237002C $20.00
Atari Cassette #36-237002C $20.00

MUSE

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.

Apple 48K Disk #47-214002D $59.95

PROGRAMMA

HIRES-80

"Hires80" is a powerful graphics editor for a 80-GRAFIX equipped S-80. With this editor you can conveniently create, store, recall and edit character sets of your own design.

S-80 16K Cassette #26-TRS43 .... $15.95
S-80 32K Cassette #26-TRS20 .... $19.95

SHAPE BUILDER II

Sick and tired of writing out all those little arrows converting them to hexadecimal, etc.? Cure the pain of messy shape tables — see the shape in Hi-Res as you enter it.
Apple II 48K Disk #47-277008D $19.95

ZAP

You move in a spaceship and try to ZAP the colored objects. If you crash into a star, you are destroyed.
Atari 8K Cassette #36-277024C...................... $6.95

STARTRAK

As captain of a starship, your mission is to rid the warring Stingons from the federation and make the galaxy safe.
Atari 8K Cassette #36-277023C...................... $6.95

REVERSE

The object of "Reverse" is to put a set of randomly placed numbers in order. You do this by taking groups of words and turning them around.
Atari 8K Cassette #36-277022C...................... $6.95

CONNECT 4

In this game you and another player drop tokens into a slot to build up four in a row to win.
Atari 8K Cassette #36-277021C...................... $6.95

LETTERS

Displayed on your screen is a square divided into smaller squares. All but one of the squares has a letter in it. You move the letters around and put them in order.
Atari 8K Cassette #36-277020C...................... $6.95

LANDER

You are captain of a spaceship about to land on the moon. You must keep a cool head and have sharp reflexes to land safely.
Atari 16K Cassette #36-277019C..................... $6.95

ASM/65

"ASM/65" is a powerful 2-pass disk based assembler. It is a compatible subset of FORTRAN cross-assemblers available for the 6500 series family. Apple "PIE" is provided with "ASM/65" to form a comprehensive development tool for the Assembly Language programmer.
Apple II 48K Disk #47-277003D $69.95

MINI LEDGER

"Mini Ledger" is the perfect bookkeeper for a small family business. Options include the following: sales and cash receipts; payments and expenses; business header; sales and cash summary; plus others. Search routines are included as well as a printer routine for a hard copy record. An excellent investment.
Apple II 48K Disk #47-277004D $59.95

REALTY PACKAGE

This program is an invaluable tool for the real estate broker or investor. It will calculate: present value, price to yield, amortization schedules, cash flow analysis, and much more. It comes with a 60-page manual and many examples.
Apple II 48K Disk #47-277005D $59.95

DISK MAGIC

This utility program allows the user to examine and modify diskettes created for the Apple II from the physical sector level and without the limitations imposed by standard DOS commands. It is possible to determine actual remaining disk space, release system space for program use, fix damaged files of all types, and restore some files that have been deleted. A comprehensive manual is included.
Apple II 48K Disk #47-277006D $24.95

AMPERSORT II

"Ampersoft II" is a high-speed Machine Language utility sort subroutine capable of sorting integer, floating point and string arrays. Examples are included.
Apple II 32K Cassette #47-277007C $15.95

HOUSEHOLD INVENTORY

Do you have a tough time keeping track of all your valuable possessions? Now your Apple and "Household Inventory" can help. Items are listed by room and entries contain information such as manufacturer, serial number and value. Values can also be represented graphically.
Apple II 48K Disk #47-277013D $24.95

HOUSEHOLD FINANCE

Now there is a way to keep track of your personal and household finances. This program allows you to see just what you spend and will help you to stretch your budget. Printer routines are included so that you may have a hard copy record.
Apple II 48K Disk #47-277012D $24.95

CHECKBOOK

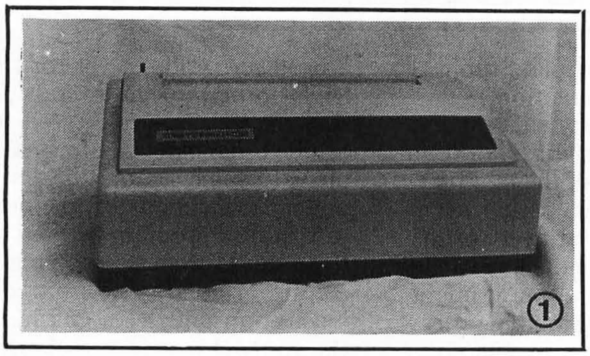
This is a complete checkbook management program that resides in only 12K of memory. This allows the user almost the entire diskette for storage of data. This program allows: posting of checks and deposits, listing checks and deposits, reconciling, balancing, searching on any field, and clearing accumulated balance. Hard copy routines are built into the program to aid the user in record keeping for tax purposes.
Apple II 48K Disk #47-277011D $34.95

PICOMATH-80

"Picomath," the study of computer algebra, has four demonstration programs which include polynomial expansion, rational expansion and simplification, trigonometric expansion and simplification and fourier transformations.

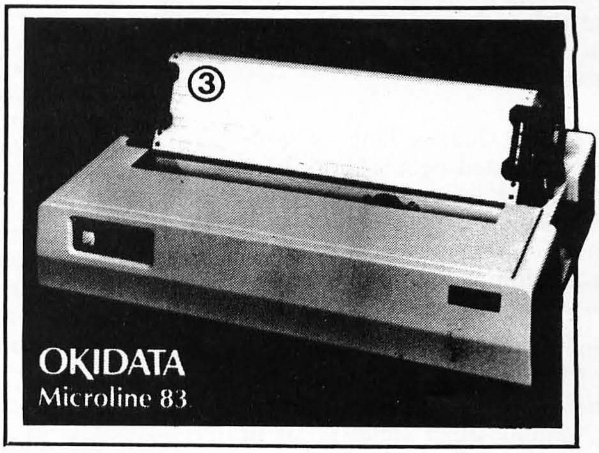
The design goal of extreme compactness necessitated reliance upon built-in approximate floating-point arithmetic and use of indirect synthetic techniques that are quite different from those generally employed manually or by more sophisticated large symbolic math systems.

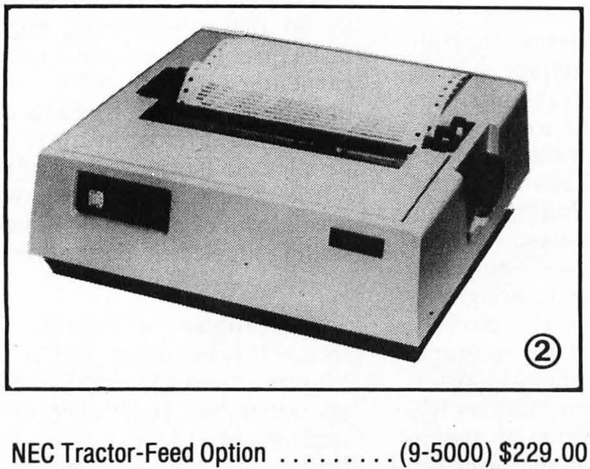
The original source code written in BASIC is easily adapted to other languages. Includes a 50-page manual.
Apple Cassette #47-277010C........................ $19.95
S-80 Cassette #26-277010C $19.95



PICK YOUR PRINTER FROM TSE•HARDSIDE TODAY!

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)	Call for Price
CENTRONICS 737	(9-737)	
OKIDATA Microline-80	(9-80)	$459.00
② OKIDATA Microline-82	(9-82)	$649.00
③ OKIDATA Microline-83	(9-83)	$949.00
OKIDATA SL300 (300 1pm)	(9-SL300)	$3795.00
EPSON MX-70	(9-MX70)	$419.00
EPSON MX-80	(9-MX80)	$519.00
EPSON MX-80FT	(9-MX80 FT)	$619.00





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

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

THE SENSUOUS PROGRAMMER

by J

THE FIRST IN A SEEMINGLY ENDLESS SERIES OF COLUMNS

When people ask me what my job is, and I tell them that I work for **SoftSide** magazine, I am learning to anticipate one of two reactions. The first one comes from women, who stare at me briefly with a mixture of uncertainty and embarrassment, and then walk away without another word. The second reaction comes from men, who stare at me briefly with a mixture of uncertainty and envy, and then ask me if I could send them a sample copy — in a plain brown wrapper.

So I figure that as long as non-computer-oriented people think of a skin magazine when they hear the name **"SoftSide"**, there might as well be something sexy inside, if only the title of a column.

Actually, the title is not quite so totally frivolous as that. (Almost, but not quite.) "Sensuous" is a word that means (among other things) "of or relating to the senses." And what are the senses but the means that people use for interfacing and communicating? And what is computer programming but the means of communication between people and computers? So, you see, it's quite logical that a column having to do with programming should have such a title.

Besides, would you have read this far if the title had been "Multifaceted Considerations of Technical and Stylistic Aspects of BASIC, With Accompanying Relevant Personal Observations and Occasional Irrelevant Digressions"?

No, I didn't think so.

Well, let's get into the socially redeeming part of this column, which is supposed to be an informal discussion of things relating to programming. In this my goal is to follow the Ancient Adage of ATOGADAYA (that's ah-TOE-gah-DIE-ah): Assume The Other Guy's As Dumb As You Are. (This is the timeless word of wisdom which has lately been corrupted by our modern permissive society into the form, "KISS" — Keep It Simple, Stupid. I prefer the classic formulation.)

As dumb as I assume that you are, then, it seems appropriate in this premiere column to ask the question, "Why write computer programs in the first place?" The answer to this question won't provide the key to the meaning of the universe, but it might help to make programming less mysterious and more enjoyable.

At first glance, the question seems to have as many different answers as there are programs. One program is written to display a battle against an attacking alien force bent on destroying your galaxy. Another is written to take you on an imagined adventure through Count Dracula's castle. Others are written to help you with bookkeeping and mailing lists and medical records. Still others are designed to beat you at chess, bridge, and backgammon, and to simulate baseball, basketball, airplane piloting, and nuclear power engineering.

On the outside, it seems that all these types of programs are written for different reasons, ranging from pure entertainment to serious business. But on the inside, there is just one reason WHY any computer program is written, and that fundamental reason controls the whole programming process — from concept, through analysis and design, to coding and debugging. It's entirely possible to write a working program, without ever being consciously aware of this "inside" reason, just as it's possible to play a game of tennis without being aware of the need for mental discipline. A person who knows a few BASIC keywords can write a program, just as a person who knows a few tennis strokes can hit the ball. But to be good at it, you need to be more conscious of the inner processes that are going on as you play the game — or program the computer.

So what is this mysterious "inner reason" that all computer programs have in common? Very simple, and not mysterious at all (though I do admit to leading you on somewhat in the last couple of paragraphs). Every computer program is created to solve a problem. And, except for the most trivial program, the overall problem that the program is created to solve can be broken down into smaller problems, usually many "layers" deeps under the main one. If you plunge into writing a program without keeping this multi-layered problem-solving picture in mind, you are doomed to frustration and teeth-gnashing from the start. (I just love making sweeping generalizations like that!)

Look at it this way. Which of you, given the goal of building a house from the ground up, would take tools in hand and simply start working on whatever part of the house you happened to think of first? (Would the young man in the back row who just raised his hand please leave quietly with the usher.) The general contractor for such a construction project is paid to know exactly how all the subsystems of the house relate to one another, and with that knowledge, can organize the job in a logical, efficient way. The task (or problem, if you will) is to build a house. In order to do that, the various specialized tasks (problems) are subcontracted to carpenters, electricians, masons, plumbers, and so on. Each of these specialized groups, in turn, employs people who have a variety of skills — who know the step-by-step breakdown of a given task such as installing an electric outlet box.

The problem of building a house, then, breaks down, after several "layers", into the problem of wiring an outlet box (x-number of times), nailing a stud in place (x-number of times), laying a concrete block (x-number of times), and so on. If the general contractor loses sight of the multiple layers of problems to be solved, there's trouble. The house might eventually get built, but a lot of unnecessary work is going to be done and undone along the way. And the end result, even if it keeps out the rain and has working switches and sinks, won't look too great inside the walls and under the carpeting. Not only that, but later additions and remodeling projects will be difficult and frustrating, since the original studs might not be on 16-inch centers and the white and black wires might be reversed in any given circuit.

Am I getting too much of a strangle-hold on the obvious by applying this to computer programming? An awful lot (I use the work "awful" intentionally) of programs are written by grabbing one's programming tools (the BASIC language and the computer keyboard) and starting work on whatever part of the program first comes to mind. (Yes, I confess, I do it too.) But more is needed to produce a program that not only has working switches, but also is relatively easy to create step-by-step, looks good inside the walls, and is amenable to later remodeling and additions. A "good" program is one that solves the main problem, and all the subproblems that it breaks down into, in a logical and purposeful way. Which implies that you, the Master Program Builder, have a good picture of all the layers of problems before you enter a single word of BASIC code.

Possibly (just possibly) a simple example will help to pull all the preceding verbiage together:

The Main Problem: To allow a person to type ten test scores into the computer, and receive a printout of the ten scores, their total, and their average value.

Subproblem #1: To design a routine to accept entry of the ten scores.

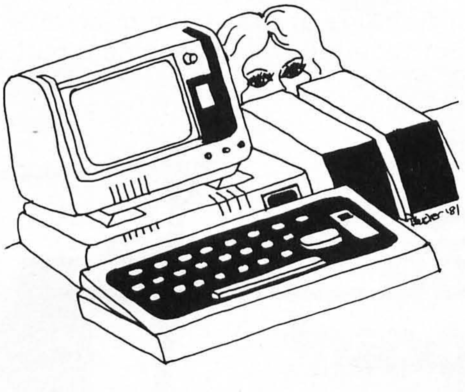
Subproblem #2: To design a routine to calculate the sum and average of the numbers.

Subproblem #3: To design a routine to print out the results.

This is simple enough to be done by what you might call a "home handyman" program (well, that's what I might call it), not really requiring the expertise and organization of a general contractor. But for the purposes of trying to apply all the good stuff I've written above, let's take the more expert approach anyway. The structure of the program will then be in four well-defined sections: One to take care of each of the above subproblems, plus one "general contractor" routine to organize and assign the various aspects of the work.

The general contractor routine is responsible primarily for assigning work to the subcontractors. Usually it will also need to perform simple housekeeping functions, and to set up certain standards so that the subroutines will work effectively together. For this program, there's the matter of the information that the first subroutine picks up being made available for the use of the other subroutines. It seems logical to set up an array where those numbers can be stored, and referred to as needed. With more complex problems, there are usually many more arrangements for this supervising routine to handle, but that's about all that needs to be done in this case.

The functioning of the three subcontractor subroutines will be very straightforward in this example. The first will most logically run through a loop ten times to receive the ten scores and store them in the array for future use. The second will retrieve the numbers, add them together, store their total, and compute and store their average. The third will retrieve the ten original numbers plus the two computed ones, and print them in an orderly format.



For complex programs, all those steps should normally be outlined on paper, including a running list of the chief variable names and their significance. That may sound like work, but it's a whole lot less work in the long run than a haphazard approach. Would you build a house without an architect's drawing? (Would somebody PLEASE get that guy in the back row out of here...)

Having done the right headwork and paperwork up to this point, then, the rest is almost pure fun. Translating the step-by-step problem-solving procedure into an actual computer program is called "coding". This is the step which is often called "programming", even though it is merely one step in the programming process (and NOT the most important one). After a few decisions about variable names, and input and output formatting, the following finished program springs full-blown from the Master Programmer's mind.

```
  9 REM "GENERAL
CONTRACTOR" CONTROL
ROUTINE
  10 DIM S(10)
  20 GOSUB 100
  30 GOSUB 200
  40 GOSUB 300
  50 END
  99 REM "SUBCONTRACTOR"
#1: INPUT SUBROUTINE


  100 FOR I = 1 TO 10
  110 PRINT "SCORE #";I;
  120 INPUT S(I)
  130 NEXT I
  140 RETURN
  199 REM
"SUBCONTRACTOR" #2:
COMPUTATION SUBROUTINE


  200 T = 0
  210 FOR I = 1 TO 10
  220 T = T + S(I)
  230 NEXT I
  240 A = T / 10
  250 RETURN
  299 REM
"SUBCONTRACTOR" #3:
PRINTOUT SUBROUTINE


  300 PRINT "TEST SCORES:" :
PRINT
  310 FOR I = 1 TO 10
  320 PRINT S(I)
  330 NEXT I


  340 PRINT : PRINT
"TOTAL = ";T


  350 PRINT : PRINT
"AVERAGE = ";A
  360 RETURN
```

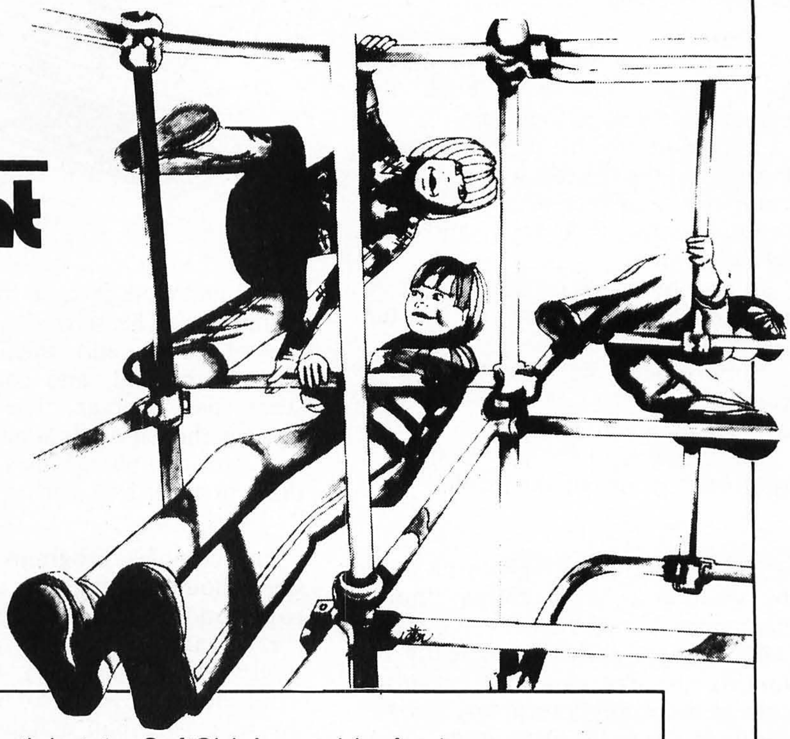
There, you see how logical it all is? And coding IS easy once all the preliminary programming work is done. In future columns I'll be discussing a lot of the fundamental and finer points of coding in BASIC, but I'll try not to let you forget about what's been covered in this premiere installment. It's great to be able to write BASIC code with flawless syntax. But far more important is the ability to dissect a problem *into* smaller and smaller parts until it almost codes itself.

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 __________

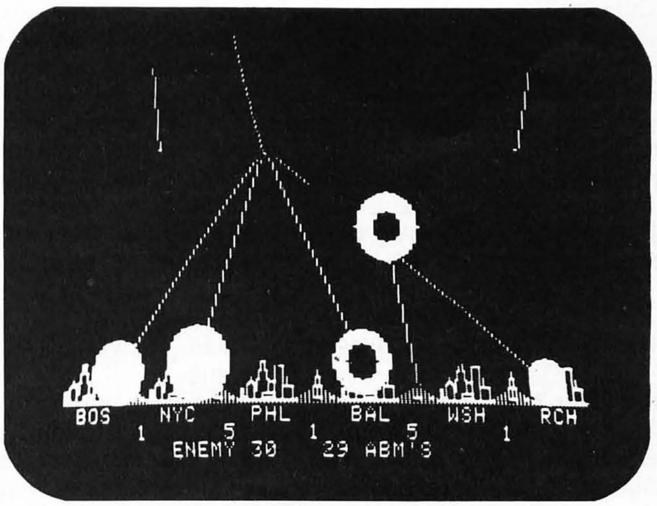
SoftSide Publications 6 South Street Milford, NH 03055

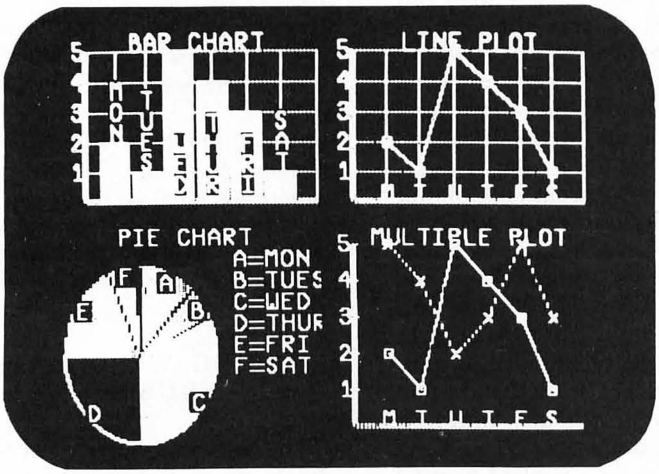
SS51

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 fireball 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

TSE-HARDSIDE

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

MY SIDE OF THE PAGE

by Lance Micklus

Getting a Bit Serious-Part 3

This month, I'm going to switch gears while I still have my topic fresh in my mind. I wanted to include somewhere in this series a critique of how a program began, how it was developed, and finally, how it made out. I have such a program in mind. I have no way of knowing, at this point, if the program will be a huge success, a total flop, or will do fairly well. But I'm game, even if it means failing openly in public.

WHERE THE IDEA CAME FROM

On February 28th, yours truly tied the big knot with his girlfriend of four years. On the day of the wedding, Dianne and I didn't know exactly where we were going to go for our honeymoon except that we wanted to head south. The first night we stayed in Albany and then headed into New Jersey the next day.

One of the places we had considered going to was Atlantic City. Another was Washington, D.C. Both required getting on the Garden State Parkway, which happens to go right by my aunt's house in Clifton. This aunt had just gotten out of the hospital with back problems and didn't make it up to Vermont for the wedding, so she was glad to see us. We discussed our plans with her, my uncle, and one of my cousins and they all said "Go to Atlantic City." Both my uncle and my cousin had been there and told us that we'd have a blast.

So Dianne and I left Clifton committed to going to Atlantic City. I told Dianne I wanted to do this right — I didn't want to tell people we stayed at the Sleeping Bear Motel in Atlantic City. Rather, we were going to stay at one of the big-name hotels. We had the money and this was the only honeymoon we were going to get.

When we got to the boardwalk, we eyed both Caesar's Boardwalk Regency and the newly opened Golden Nugget. We decided on Caesar's and checked in.

On the television set in our room was one channel that had a twenty-minute tape, made at the hotel, which explained all of the casino games. Neither Dianne or I knew anything about these games but we must have watched the tape four times before we left to get something to eat. We quickly got lost trying to find the restaurants and learned a simple secret: To go anywhere in the hotel, you must first go to the casino. From the casino you get to everything else.

After eating, we decided it was time to check out the casino. I still don't believe I saw what I saw. It is an experience and that's all you can say about it. We walked around looking at all of the table games and then decided to blow away $11.00 in the slot machines. That didn't take long. Again, it was back to the tables to watch.

At this point, my interest was focused mostly on craps. It is a game I knew almost nothing about. Two pictures come to mind, though. One is of a dice game in a back alley. The other is of James Bond with some sexy girl in Las Vegas (**Diamonds Are Forever**).

By the following morning, Dianne and I felt we hadn't taken it all in yet. So, we decided to stay for another day, walk the boardwalk down to The Golden Nugget, and see what they had there. What they had, among other things, were nickel slot machines.

So that night we went out of our way to look in our own casino at Caesar's to find nickle slot machines. Sure enough, we found them and together blew away $12.00, but this time it took an hour for both of us to do it. I do recall feeling stupid putting my little nickel in the bandit and pulling the lever so many times. It just seemed so dumb.

Well after that we walked around the casino again and I stopped at one of the craps tables. This game really fascinated me. I noted that the table number was 13, and 13 is Dianne's lucky number. We almost decided to buy some chips and play but for some reason we didn't. Just as well or the story might have ended here.

Instead we figured we'd just buy a craps layout and a toy slot machine to take home with us. So, that's where $100.00 of our gambling money went — in souveniers.

We left the hotel the following day but the experience left several strong impressions on me.

During our stay, I must have watched the gambling TV show about twenty times. It turns out that in all of these casino games, no gambling skill is required by the dealer. The rules of the game predetermine exactly what he will do in any situation. Indeed, no matter what the casino game, the dealer is nothing more than a machine — like a computer.

A few days after we got home we discovered the toy slot machine we bought for $75.00 was also sold by Service Merchandise here in Burlington, Vermont, for $49.95 (list price is $65.00). In Atlantic City we were told that these toy slot machines could only be bought in Atlantic City and Las Vegas. Later, we would also find out that the odds shown on the craps layout we bought were wrong.

The craps game was a big hit with my son, Tony (by my first marriage). Typical of him, he ran the bank broke. But Dianne had a another comment about craps. She knew that it was only a matter of time before I would have a computer version of craps. The girl didn't marry a stranger, you know. She understands how this mind of mine works.

Well, she was right, but I had a lot of other projects to work on so I looked at the Radio Shack " Casino" package thinking I could just buy something already made up. Forget it. That's not like casino craps. I knew how to write the program and so I sat down one evening and began working on the code.

CREATING A PROGRAM FROM SQUARE ONE

The color computer was the perfect choice. It is very portable so I could bring it over to the house anytime we wanted to play at home. Originally, I was also going to make use of the joysticks to select where the bets would be placed. Later, I changed my mind. Finally, the thought occurred to me that I could take the finished game and blow it over to the Model I using "ST80-III" to simulate a lineprinter. The game logic would still be valid. Only the graphics and sound effects would have to be changed. The sound effects would be easy to implement; I'd just rob the routines out of Star Trek III. After getting it running on the Model I, it should also work on the Model III.

And, if there isn't a lot involved, I could then blow the code over the Model II for my game package. Also, with a little more effort, I might be able to talk The Source or MicroNET into putting it on line for their users' enjoyment.

In Part 2 of this series, I had mentioned the advantages of the professional over the amateur. One of those advantages is the fact that I can bring up a product on several machines, whereas the amateur can usually only get it running on one. The above line of thinking makes that point rather well.

Before I write any code, I always like to have a plan of attack. The game of craps has two parts, the first part of which is the betting. The second part consists of rolling the dice and paying off the bets. My thinking on the total programming concept of the game wasn't clear yet. However, the dice roll would be a must. It could be written by itself and then the other parts of the program, no matter how they were done, could just interface to it.

In the real casino craps, the shooter shakes the dice and throws them across the table. The dice must bounce off the opposite side of the table and roll back, landing flat when they stop. I wanted to approximate that action, but not to the extent that it became my whole program. My dice stay stationary on the screen but keep changing face numbers to simulate the shaking. When the ENTER key is pressed, the dice quickly roll toward the top of the screen where they hit a line and start bouncing back. They keep rolling toward the bottom of the screen, stopping at some random point. It looked good.

The next part took a lot of planning. The fun of craps is in all of the options you have to make bets. This was the feature sadly lacking in the Radio Shack "Casino" package. To do this effectively, the entire betting field must be visible to the player showing all of his bets. Using graph paper, I worked out a playing field that would show all of the bets at once for each player. I then implemented the routine by building each line with FOR-NEXT loops. In the final version, I figured I would just put the graphics and text in PRINT @ statements which would save me a lot of memory.

After the playing field was built, the next step was to consider how the computer would handle bets and pay them off. I often tell people who are serious about programming in a high level language like BASIC for FORTRAN that the most important tools they have are arrays. Arrays are your friends and they can do miracles for you. If you can't handle arrays, you will never be able to work with complicated programs.

I decided to use a two-dimensional array. The second array element would indicate which player, while the type of bet would be stored in a logical position indicated by the first element of the array. This would allow the same win/lose logic to work with any bet no matter what it was. At the same time, several people could play the game and the computer would be able to keep each player's bets straight.

Now that I had the array, I began to develop a routine to display the bets. The concept was to display the bet field and then overwrite the screen with the betting information. This routine was tested by simply force-feeding numbers into the array to make sure they were displayed in the correct place.

This now left two unfinished sections — one to take bets and another to pay them. I decided to write enough code for the computer to pay off only three types of bets — ANY 7, ANY 3, and ANY 11. These bets are simple. The dice are rolled and if they come up with 7s, then you pay any 7 five times what he bet. Then I wrote enough code to make the above three bets from the keyboard.

Now I had the program running enough to see how it was working out. I was very happy so I added the rest of the bet and payoff logic to finish the game. Then I began to play it for while.

AN EVEN BETTER IDEA

A day or two later, Dianne and I were talking and she mentioned that she had read about some guy who figured out a way to beat the house. Now, none of the casinos would let him play. Personally, I didn't think it could be done. Even if a computer was doing the betting, it couldn't be done. Or could it? Well, how would the computer know where to make the bets and for how much? There's got to be a way to do it. I was having incredible luck at this game. Maybe I could get the computer to play also. Features like that help to sell the program.

So, the great mind spun its wheels to try to find a way for the computer to make bets also. The trick was to figure the odds and to pick the

continued on next page

continued from previous page

number most likely to come up. The more likely the number, the higher the bet. Since dice produce random numbers, the occurrence of any number should be evenly spread out according to the odds.

I finally decided to go with a history stack. There are only 36 ways dice can land. In theory, in 36 rolls, the dice will land once each way. Thus, the history stack would store each roll in the bottom and push all the other combinations up one. When a combination came close to the top, that meant it was time for that combination to come up again.

The result was a process by which the computer could bet with some kind of logic. The trick was to prime the history stack. Originally, I randomly loaded the history stack with numbers so the computer had some data to work with and bet. I had to do that to check out its betting logic. The problem was this: Suppose the first number in the top of the stack, even randomly mixed, is a two. Then the computer will start betting everything it owns on the two's for no good reason. It should only do that if two has failed to come up for a long time. With no history in the stack, the computer didn't know what to do. So, it wouldn't bet at all for a while. After some experimenting, I found that it worked best if the stack was loaded with only six random numbers. This was enough to work with while keeping the integrity of the system.

One of the beautiful things about BASIC is the fact that you can BREAK the program at any time so you can change key variables to make sure things are working the way you expected them to. For example: Putting a two at the top of the history stack should cause the computer to bet heavily on ANY 2, ANY CRAPS, and DON'T PASS/COME. These are all-powerful tools that make some languages better than others for doing a job.

Nearing completion of the program, I decided it needed a name that was better than just CRAPS. Jokingly, I told Dianne one day that I was going to call it "The Mean Crap Machine". We had a good laugh over that, but some dealers might get tense over that name. So, I added the "s" and called it "The Mean Craps Machine"™ to tie it in with "The Mean Checkers Machine"™. If you build a name, cash in on it.

The final version is done but it is running on a 16K Standard BASIC machine since my Extended Color BASIC chips haven't come in yet. When they do, I'll make whatever changes are needed to get it working in Extended Color BASIC, the language it will actually be sold for.

IS ANYBODY REALLY GOING TO BUY IT?

Let's take a look at where things stand now.

So far, several people have written computer craps programs and nobody seems to have shaken the Earth. I believe this is because these programs did not fully implement all of the betting options in the game. It is the betting options that make the action exciting.

On the plus side is the fact that many casinos have proven that craps is a fascinating game to play. So fascinating, people will spend fortunes to play it. Not very many people are willing to risk $100,000 on "Star Trek" games.

Another consideration is that I talked to a number of people about craps and found that they shared the same interest in the game as I did. Almost nobody knew much about the game but just about everybody seemed to agree that a good craps game was not available for microcomputers. There's no point doing what's already been done and done well.

Something to think about is the fact that while there may not be very many S-80 Color Computers out there right now, there also isn't much competition, either. I'm gambling (no pun intended) that the Color Computer will sell well. I'm backing my bet up with the fact that the Model I and Model III are plentiful so I have at least one other market for the program which is known to be solid.

I spent a lot more time on "The Mean Craps Machine"™ than I had originally intended to. I figure I need to make $2000 to say it paid even money. If it sells for $14.95 and I get a 15% royalty on each sale, then I will need to sell 893 copies of the program. A good selling program on the Model I/III sells about 500 copies. Considering that the Color Computer market is small right now, you can see I'm in trouble. However, the present lack of competition right now might give me a bigger market share to make up for it.

If I add the additional time to bring the program up on the Model I/III, then I guess I would have to see it make $3000 to call it an even payoff. That would mean I'd have to sell 1340 of these craps games. I'm still in trouble.

A more realistic figure would be 250 on the Color Computer and 500 for the Model I/III. If I get a 15% royalty, then working the numbers the other way around, I'd have to sell "The Mean Craps Machine"™ for roughly $26.75. But that price is too high and people will complain. Of course, I might not sell any. Or, I might sell 3,000 of them. The whole thing is a risk.

Now, I think you see a fundamental problem with the pricing in computer software. The reason nobody's written a good craps game is because there probably isn't enough money in it to make it worth the effort. The only way such a program can be brought into existence is for somebody to do it anyway, no matter what, and then try to cash in on it. This is not the way to stimulate product development.

In future articles, I'll keep you posted on what's happening with "The Mean Craps Machine"™ so we can all keep track of it. But, if nothing else, I hope the experience I've shared with you this time will prove to be valuable to both of us.

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

KUDOS AND KLUNKERS

by Wynne Keller

Radio Shack's Model III has many improvements over the Model I. These hardware changes have reeived considerable media attention, but the changes in the DOS have not been discussed in any detail. This article reviews some of the new commands: What they are supposed to do, and whether they function as promised. Of necessity, commands which have problems will receive more attention. This should not be construed to mean the DOS is full of "bugs". On the contrary, it is on the whole a very useful and fine DOS.

DOS COMMANDS:

HELP

Perhaps the most welcome new command is HELP. Typing this, plus the name of the command in question, gives an explanation of correct syntax and use, thereby solving the frustration of looking up minor details in the manual. HELP is available for all DOS library commands.

BUILD, DO, AND PAUSE

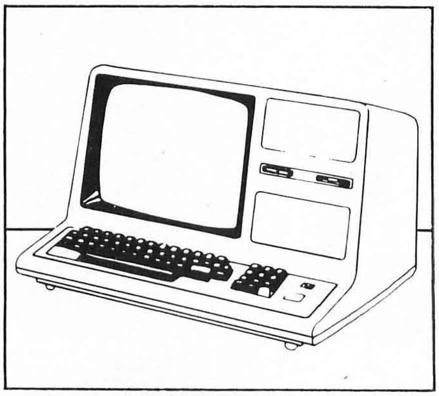
BUILD, DO and PAUSE are used to create turnkey systems such that the inexperienced user need only type in the date at system startup; the rest will be automatic. BUILD creates a group of commands to be executed, such as pausing while disks are inserted, booting BASIC, answering the file number and memory size question, and running the desired program. This BUILD file is given a name, for example, "STARTER". Then the command AUTO DO STARTER alters the disk such that the commands in STARTER will be executed automatically whenever the reset button is pressed. The manual doesn't explain how to get past the two questions which are always asked when BASIC is requested: "How many files?" and "Memory size?". This is done by pressing [ENTER] twice (assuming you wish default values) while BUILDing. For example, in DOS Ready, type BUILD *STARTER*. For the next input type BASIC [ENTER]. Then [ENTER] twice more for the next two prompts.

DUAL

There is a new screen print function, similar to the JKL of NEWDOS. It is implemented with shift + down arrow + asterisk, and may be used in both DOS and BASIC. DUAL is a DOS command which provides printer and screen output simultaneously, each time the user presses enter. Thus DUAL prints the operator interaction as it occurs, a line at a time, rather than a whole screen.

FORMS

The new FORMS control had more promise than it delivered. The manual seemed to imply that it would be possible to set printer line width and paper length with FORMS, thereby avoiding the problem of LLIST printing on the tear lines of pages. Unfortunately, only the width may be controlled in LLIST. The FORMS command will keep track of the number of carriage returns, and may be PEEKed during program operation to see if it is time to advance to the next page, at which point CHR$(12) (Top of Form) may be used. However it is not possible to PEEK at the forms line number during an LLIST operation, so program printouts are still out of control.



TRS-80 Model III

FREE

The FREE command has been enhanced to show a map of free space, rather than just the number of granules. This is useful for those who can never remember how many bytes are in a granule, and simply want an answer to the question "Is the disk almost full?". The FREE map can also be output to the printer.

It is now possible to LIST the contents of a file on screen or printer, with optional slow output and ASCII format. This is extremely useful for examining files whose contents you can no longer remember, or for viewing damaged files.

MASTER

MASTER allows one disk drive to be specified as the Read/Write drive. When a master drive is specified, all attempts to read or write a file start at that drive, bypassing lower numbered drives. There is also a new way to write-protect a drive. Using the WP command, a single drive may be protected from writing. Unfortunately this protection simply stops the attempt to write and produces an error. The DOS does not try to write on another drive. However, write-protect tabs, when placed on a disk to be protected, will force the writing to another disk. This is an improvement over the Model I TRSDOS.

PATCH

PATCH allows minor corrections in the contents of a disk file. This is extremely helpful to Machine Language programmers, and will also be used by Radio Shack to provide changes for bugs in TRSDOS.

PURGE

PURGE runs through the visible user files on the disk, prints their names individually on screen, and asks if the file is to be killed. This is a great improvement over the old system of typing each name one at a time. Purge can also kill system files, but will not present them one at a time — it kills them all.

TAPE

TAPE allows Machine Language programs to be copied from tape to RAM or disk, or from disk to tape. The manual does not mention that this function is for Machine Code programs only. This feature removes the necessity of using special utilities to backup such programs.

TIME

The TIME function has been improved. It can be displayed on screen at all times, if desired, in the upper right-hand corner. It no longer destroys cassette operations, as the clock shuts itself off during I/O.

DIRECTORY

There is a problem with the DIRectory command. It is supposed to show invisible and system files when requested. However, it only shows invisible files if system files are requested, and it never shows system files at all. The rest of the directory works fine; user files are presented clearly with helpful information such as date created, size of file, protection status, etc. One idiosyncracy of the DIR command is that to prevent scrolling, press the @ key only, not shift @ as you would during a BASIC program LIST.

ROUTE

ROUTE is supposed to allow I/O to be sent to various devices, such as printer, display, or RS-232. Unfortunately it freezes the keyboard and does not function. A phone call to the Radio Shack hotline brought the information that there are serious problems with ROUTE. Presumably they will be corrected in future versions of the DOS. This feature is also available in the cassette BASIC of the Model III, and appears to function perfectly there.

BASIC COMMANDS:

There are many new CMD commands used from BASIC which are welcome additions. A few are disappointments. The greatest disappointment was CMD"X" which is a cross-reference utility to help BASIC programmers keep track of variables and program flow. It is necessary to request each variable or function one at a time, unlike NEWDOS where the entire program can be cross-referenced with one command. Also, output is supposed to go to the printer automatically if one is available, but it does not. To solve this, use CMD"Z" which is equivalent to DUAL of DOS mode: It prints all screen interaction as hard copy.

CMD"C"

A packing routine, CMD"C", worked well with one flaw. Using this command, it is possible to delete REM lines or extra spaces, or both, from a BASIC program. Line numbers and line packing are unaffected. The drawback is that if space deletion alone is requested, it will also delete spaces in REM lines, rather than leaving them alone as it should. This ruins the appearance of program comments, titles, and credits. The solution is to add quotation marks to all REM lines to protect them from compression.

CMD"O"

The new sort feature, CMD"O" provides Machine Code sorting speed. The drawback is that only single-dimension arrays can be used. This feature sorted 900 records (average of 5.5 bytes each) in 40 seconds. It is a welcome addition to BASIC but very limited in application as most programs need at least two dimensions for arrays.

CMD"I"

CMD "I" is supposed to allow TRSDOS commands to be executed from BASIC without disturbing BASIC if possible. However, we tested it with many commands and could not find one which left BASIC intact. Therefore the only merit to this command appears to be the ability to write a TRSDOS command as one statement from BASIC. Assume it will destroy your resident program.

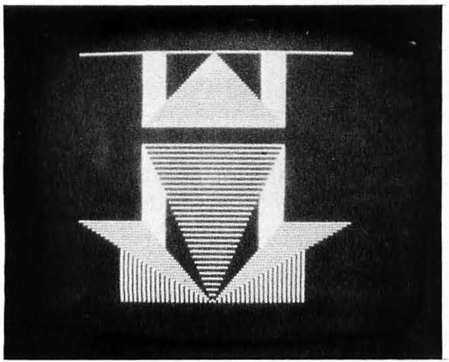
An important new file handling feature allows variable length files of up to 256 bytes per record, for Random Access storage. Testing this feature, we were unable to make it function if we used a variable to specify file length. Hours later the problem was found: When fielding, any variable used to specify the number of bytes for a label must have the % sign (integer sign) after the variable. It is not possible to use DEFINT instead as the DOS does not recognize DEFINT in this situation. This feature can help programmers save a lot of disk space when file size is not uniform.

Although there are many more features which could be mentioned, these are sufficient to show the great versatility of the Model III. The review of fixes for DOS problems may save you some time when your own Model III arrives. Eventually new versions of TRSDOS will undoubtedly be released and many of these problems will disappear.

TRIVIA

by Michael Bendik

"Trivia is an Apple program requiring Applesoft and 16K RAM.



"Trivia" is an excellent Hi-Res graphics art program for a number of reasons. The program is comprised of six segments which together run for approximately 12 minutes. Each segment has continuous motion, and the second and last segments contain random plotting variables. The program itself is an endless loop, restarting itself after the full cycle of displays.

One of the best features of "Trivia" is the internal structure of the program. A quick glance at the listing shows a logically structured program, where the flow of control from one section to another is clear and straightforward. There are six separate subroutines for the six basic parts of the program, and other subroutines could be easily added for future expansion. Multiple-statement lines are used very sparingly, making each line easy to follow. Although this takes up slightly more memory (there are five bytes of "overhead" for each program line), it does help to keep things tidy when memory space is not critical.

The displays themselves are created using just a few simple shapes in creative ways, using the Apple's fast DRAW command and its associated SCALE and ROTation possibilities. The author makes good use of the screen wrap-around characteristic of the DRAW command, to produce symmetrical patterns at opposite edges and corners of the high-resolution screen. His techniques invite others of you imaginative people out there to add your own simple-but-elegant designs.

Although the program does not require user input, it is necessary to be familiar with the color and brightness settings of your TV or monitor. Setting the brightness all the way down and the color to its maximum, for example, gives a nice irridescent appearance. You'll want to do some experimenting for different effects.

SHAPE TABLE

Shape 1: Box without corners.
Shape 2: Square.
Shape 3: Cross.
Shape 4: Line.

VARIABLES

A, BLP, CL, CR, H1C, H2C, LP, LX, RB, STP, WT, XLP, YLP: Misc. loops, color variables, and counters.
DD: Checks when to leave dandelion routine.
EN: Used with the shape table.
F: Used as a flag.
HBC: Randomly generated color variable.
HC: Hcolor.
RR: Random rotation variable.
SC: Scale variable.
SH: Shape number in table.
TB: Shape table data.
TM: Time delays.
X: Misc. loops and horizontal drawing position.
XPT: Random horizontal plotting position.
XX: Horizontal plotting position.
Y: Vertical drawing position.
YPT: Random vertical plotting position.
YY: Vertical plotting position.

```
100  TEXT : HOME
110  FOR X = 1 TO 10: PRINT : NEXT

120  SPEED= 1
130  PRINT "      A P P L E  T
     R I V I A  I"
140  PRINT : PRINT "
      BY M.BENDIK"
150  SPEED= 255
160  FOR X = 1 TO 3000: NEXT
170  FOR X = 1 TO 24: PRINT
180  FOR A = 1 TO 70: NEXT A
190  NEXT X
200  HOME


 A separate subroutine is used for
 each of the six basic drawing
 routines.


210  GOSUB 1000
220  GOSUB 2000
230  GOSUB 3000
240  GOSUB 4000
```

```
250  GOSUB 5000
260  GOSUB 6000
270  GOSUB 7000
280  HGR2 : GOTO 100


 Data for the four shapes of the
 shape table.

1000  DATA  4
1010  DATA 0
1020  DATA 10,0,20,0,23,0,32,0
1030  DATA 18,63,32,100,45,21,54,
     30,7,0
1040  DATA 44,62,0
1050  DATA 58,60,44,44,53,53,55,7
     ,0
1060  DATA 36,36,4,0
1070  DATA 0
1080  POKE 232,0: POKE 233,3
1090  RESTORE
1100  FOR TB = 768 TO 803
1110  READ EN: POKE TB,EN
1120  NEXT TB
1130  RETURN
```

```
 Hurricane or spiral shape
 simulation.


2000 SH = 1
2010 CR = 7
2020  HCOLOR= CR
2030 LP = 35:X = 139:Y = 96
2040  FOR TM = 1 TO 2500: NEXT TM

2050  GOSUB 2130
2060 LP = 70:X = 0:Y = 95
2070  FOR TM = 1 TO 2500: NEXT TM

2080  GOSUB 2130
2090 LP = 60:X = 0:Y = 0
2100  FOR TM = 1 TO 2500: NEXT TM

2110  GOSUB 2130
2120  GOTO 2280
2130  HGR2
2140  FOR R = 1 TO 1000: NEXT R
2150  FOR R = 1 TO LP
2160  ROT= R
2170  SCALE= R
```

```
2180  DRAW SH AT X,Y
2190  NEXT R
2200  FOR R = 1 TO 1000: NEXT R
2210  FOR R = 1 TO LP
2220  ROT= R
2230  SCALE= R
2240  XDRAW SH AT X,Y
2250  NEXT R
2260  FOR R = 1 TO 1000: NEXT R
2270  RETURN
2280  RETURN
```

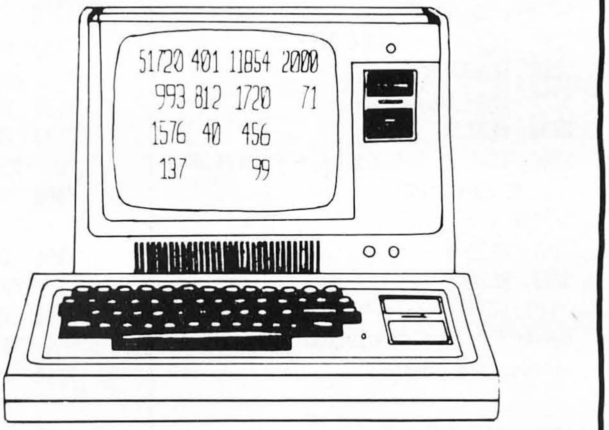
Sunburst or equalizer type drawings.

```
3000  HGR2
3010  FOR STP = 1 TO 3
3020  IF RB = 2 THEN RB = 5:WT =
      7: GOTO 3060
3030  IF RB = 6 THEN RB = 2:WT =
      3: GOTO 3060
3040  IF RB = 5 THEN RB = 6:WT =
      7: GOTO 3060
3050 RB = 5:WT = 7
3060  FOR TM = 1 TO 2000: NEXT TM

3070 CL = 99
3080 X = 0:Y = 0
3090 XX = 279:YY = 191
3100 XPT =  INT (279 *  RND (1))
3110 YPT =  INT (191 *  RND (1))
3120  FOR XLP = 0 TO 279
3130  HPLOT XPT,YPT TO X,Y
3140  HPLOT XPT,YPT TO XX,YY
3150 X = X + 1
3160 XX = XX - 1
3170  IF CL < 28 GOTO 3220
3180  IF HC = WT THEN HC = RB: GOTO
      3200
3190 HC = WT: GOTO 3200
3200  HCOLOR= HC
3210 CL = 0
3220 CL = CL + 1
3230  NEXT XLP
3240 X = 279:Y = 1
3250 XX = 0:YY = 190
3260 CL = 99
3270  FOR YLP = 1 TO 190
3280  HPLOT XPT,YPT TO X,Y
3290  HPLOT XPT,YPT TO XX,YY
3300 Y = Y + 1
3310 YY = YY - 1
3320  IF CL < 19 GOTO 3370
3330  IF HC = WT THEN HC = RB: GOTO
      3350
3340 HC = WT: GOTO 3350
3350  HCOLOR= HC
3360 CL = 0
3370 CL = CL + 1
3380  NEXT YLP
3390  FOR TM = 1 TO 2500: NEXT TM
3400  NEXT STP
3410  RETURN
```

Manipulation of box shape.

```
4000  HGR2
4010  FOR TM = 1 TO 1000: NEXT TM

4020 SH = 2
4030 HC = 1
4040  FOR BLP = 1 TO 5
4050 HC = HC + 1
4060  GOSUB 4100
4070  FOR TM = 1 TO 1500: NEXT TM

4080  NEXT BLP
4090  GOTO 4240
4100  IF HC = 4 THEN HC = 5
4110  HCOLOR= HC
4120  FOR R = 1 TO 87
4130  ROT= R
4140  SCALE= R
4150  DRAW SH AT 140,80
4160  FOR TM = 1 TO 12: NEXT TM
4170  NEXT R
4180  FOR R = 88 TO 174
4190  ROT= R: SCALE= 175 - R
4200  DRAW SH AT 140,80
```

continued on next page



continued from previous page

```
4210  FOR TM = 1 TO 12: NEXT TM
4220  NEXT R
4230  RETURN
4240  RETURN
```

Manipulation of cross shape.

```
5000  HGR2 : ROT= 0
5010  FOR TM = 1 TO 1500: NEXT TM

5020  FOR BLP = 1 TO 6
5030  IF BLP = 1 THEN X = 0:Y = 0
      : GOTO 5070
5040  IF BLP = 2 THEN X = 0:Y = 1
      25: GOTO 5070
5050  IF BLP = 3 THEN X = 140:Y =
      0: GOTO 5070
5060  IF BLP = 4 THEN X = 140:Y =
      125: GOTO 5070
5070  HCOLOR= 5
5080 SH = 3
5090  GOSUB 5120
5100  NEXT BLP
5110  GOTO 5270
5120  FOR R = 1 TO 60 STEP 2
5130  SCALE= R
5140  DRAW SH AT X,Y
5150  NEXT R
5160  IF BLP < > 6 THEN  GOTO 51
      90
5170  FOR TM = 1 TO 1000: NEXT TM
5180  GOTO 5250
5190  HCOLOR= 2
5200  FOR R = 1 TO 60 STEP 2
5210  SCALE= R
5220  DRAW SH AT X,Y
5230  NEXT R
5240  IF F < > 9 THEN F = 9: HCOLOR=
      4: GOTO 5200
5250 F = 0
5260  RETURN
5270  RETURN
```

Dandelion type simulation using the single line shape.

```
6000  HGR2 :SH = 4
6010  FOR TM = 1 TO 1000: NEXT TM
6020 X = 139:Y = 96:H1C = 3:DD =
      1:LX = 29
6030  GOSUB 6140
6040 X = 139:Y = 191:DD = 1:LX =
      27
6050 H1C = 6
6060  GOSUB 6140
6070 X = 0:Y = 0:DD = 2:LX = 30
6080 H1C = 5:H2C = 7
6090  GOSUB 6140
6100 X = 0:Y = 95:DD = 2:LX = 42
6110 H1C = 2:H2C = 3
6120  GOSUB 6140
6130  GOTO 6360
6140  HGR2 : FOR TM = 1 TO 1500: NEXT
      TM
6150  FOR LP = 1 TO LX STEP 2
6160 SC = LP
6170  HCOLOR= H1C
6180  GOSUB 6220
6190  NEXT LP
6200  FOR TM = 1 TO 2000: NEXT TM
6210  RETURN
6220  FOR R = 1 TO 64
6230  ROT= R
6240  SCALE= SC
6250  DRAW SH AT X,Y
6260  NEXT R
6270  IF DD = 1 THEN  GOTO 6350
6280  IF LP = LX - 1 THEN  GOTO 6
      350
6290  HCOLOR= H2C
6300  FOR R = 1 TO 64
6310  ROT= R
6320  SCALE= SC
6330  DRAW SH AT X,Y
6340  NEXT R
6350  RETURN
6360  RETURN
```

Manipulation of various backgrounds.

```
7000  FOR BLP = 1 TO 25
7010  HGR2
7020 HBC =  INT (8 *  RND (1))
7030 HC =  INT (8 *  RND (1))
7040  IF HBC = HC THEN  GOTO 7030
7050  ON HBC GOTO 7070,7070,7080,
      7080,7070,7070
7060  GOTO 7080
7070  ON HC GOTO 7020,7020,7080,7
      080,7020,7020
7080  IF HBC = 0 AND HC = 4 THEN
      7020
7090  IF HBC = 3 AND HC = 7 THEN
      7020
7100  IF HBC = 4 AND HC = 0 THEN
      7020
7110  IF HBC = 7 AND HC = 3 THEN
      7020
7120 X =  INT (279 *  RND (1))
7130 Y =  INT (191 *  RND (1))
7140 RR =  INT (100 *  RND (1))
7150 SC =  INT (4 *  RND (1)) + 1
7160  ON SC GOTO 7170,7190,7210,7
      230
7170  IF RR < 50 THEN RR = RR + 5
      0
7180  GOTO 7250
7190  IF RR < 90 THEN RR = RR + 9
      0
7200  GOTO 7250
7210  IF RR < 50 THEN RR = RR + 5
      0
7220  GOTO 7250
7230  IF RR < 75 THEN RR = RR + 7
      5
7240  GOTO 7250
7250  HCOLOR= HBC: HPLOT 0,0
7260  CALL 62454
7270  HCOLOR= HC
7280  FOR R = 1 TO RR
7290  ROT= R
7300  SCALE= R
7310  DRAW SC AT X,Y
7320  NEXT R
7330  FOR TM = 1 TO 1500: NEXT TM
7340  NEXT BLP
7350  RETURN
```

APPLE ONE LINERS

Applesoft

```
1  HGR2 : FOR I = 0 TO 279:C = C +
     1:C = C - (6 *  INT (C / 6))
     : HCOLOR= C: HPLOT I,0 TO 27
     9 - I,191: NEXT I: FOR I = 0
      TO 191:C = C + 1:C = C - (6
      *  INT (C / 6)): HCOLOR= C:
      HPLOT 0,191 - I TO 279,I: NEXT
     I: END
```

Mark Johnson
Minneapolis, MN

Applesoft

```
1  HGR2 : FOR J = 1 TO 150:S =  PEEK
     ( - 16336): HPLOT 140,96:X =
      INT ( RND (1) * 277):Y =  INT
     ( RND (1) * 151): HPLOT  TO
     X,Y:Z =  INT ( RND (1) * 8):
      HCOLOR= Z: NEXT J: GOTO 1
```

David Sepinski
Verona, NY

Integer BASIC

```
 1 GR : POKE -16302,0: CALL -1998
   : FOR J=1 TO 30000: COLOR= RND
   (16):X= RND (30):Y= RND (37
   ): FOR K=0 TO 10: HLIN X,X+
   10 AT Y+K: NEXT K: NEXT J
40 GR : CALL -1998: POKE -16302
   ,0: FOR A=1 TO 32000:B= RND
   (20):C= RND (24): COLOR=C+ RND
   (2)+B: PLOT B,C: PLOT 39-B,
   C: PLOT B,47-C: PLOT 39-B,47
   -C: NEXT A
```

Paul Mosher
Voorheesville, NY

Integer BASIC

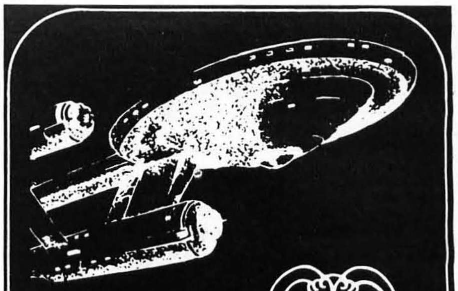
```
1 GR : POKE -16302,0: CALL -1998
  : FOR J=1 TO 30000: COLOR= RND
  (3):X=J MOD 40:Y=J MOD 48: HLIN
  X,X+ RND (40-X) AT Y: VLIN
  Y,Y+ RND (48-Y) AT X: NEXT
  J
```

Paul Mosher
Voorheesville, NY

SOLUTION TO APRIL'S WORD SEARCH PUZZLE

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

A new puzzle can be found on page 13.



DODGE

by Matt Rutter

"Dodge" is a program for the Atari requiring 8K RAM and one joystick.

In this game, your mission is to *pick up all* the gold pellets and diamonds on the board by running your car over them. Sound simple? Not really. There's a computer car whose only mission is to destroy you.

Your car moves around the screen counter-clockwise while the computer car moves clockwise. Your only way to avoid a head-on collision is to move into one of the other lanes or to adjust your speed by pushing the accelerator, which is the fire button on the joystick. You change lanes by moving your joystick in the direction you want to change at one of the four sides of the screen. When you crash you will receive another car until all three of your cars are gone.

Every once in a while, the computer car will change the gold pellets into diamonds when it runs over them. These diamonds are worth more points when collected by you. If you clear the board of all the gold pellets and diamonds, you receive a bonus of 500 points and a fresh board with all new pellets.

VARIABLES

A, B: Position of computer car.
AS, BS: Horizontal and vertical speed of computer car.
C: Random number used in determining whether or not the computer car leaves diamonds.
L: Lane of player car.
L1: Lane of computer car.
P: Joystick value.
S: Cars left.
SC: Score.
T: Number of gold and diamonds left.
X, Y: Position of player car.
XS, YS: Horizontal and vertical speed of computer.
Z1: ASCII value of character under computer car.

DOCUMENTATION

NOTE: The string "**BONUS**" in line 620 should be typed in reverse video.

Lines 20-50: Draw screen.
Line 60: Variable initialization.
Line 70: Draws the two cars.
Lines 90-95: Give player a head start.
Lines 100-110: Branch to player and computer moving routines.
Line 120: If player is at any of the lane-changing positions and the joystick is moved then branch to the lane-changing routine.
Line 130: If computer car is at lane-changing position then branch to computer lane-changing routine.
Line 200: Erases player and turns off sound.
Line 205: If button is pressed, moves player an extra space.
Line 210: Moves player, checks if player is over a period.
Line 220: Checks if player is over a diamond.
Line 225: Prints score, draws player, if all pellets are gone then branches to bonus routine.
Lines 230-245: Check if player is in a corner.
Line 300: Checks for a collision before computer moves.
Line 305: Prints diamond.
Line 320: Moves computer and draws computer car.
Lines 330-345: Check *if computer is* in corner and change direction.
Line 350: Checks for collision after computer car moves.
Line 400: Erases player.
Lines 410-490: Change lanes according to the position on the screen and joystick position.
Line 500: Start of lane-changing routine.
Lines 520-590: Change lanes according to position on board and lane of other player.
Lines 600-640: Bonus routine. Adds 500 points to score and goes back to beginning of program to draw next board.
Lines 650-670: Explosion.
Line 680: Delay, then checks if any cars are left and goes to the beginning of the program.
Lines 690-695: End of game routine. Check for start button being pressed to start another game.

```
1 REM DODGE
2 REM by Matt Rutter
3 REM written Dec,1980
5 SC=0:S=3
10 GRAPHICS 0:POKE 752,1
20 COLOR 124:FOR I=0 TO 8 STEP 2:PLOT
I,I:DRAWTO I,22-I:PLOT 38-I,I:DRAWTO 3
8-I,22-I:NEXT I
25 COLOR 18:FOR I=0 TO 8 STEP 2:PLOT I
,I:DRAWTO 38-I,I:PLOT I,22-I:DRAWTO 38
-I,22-I:NEXT I
30 FOR I=0 TO 8 STEP 2:COLOR 17:PLOT I
,I:COLOR 26:PLOT I,22-I:COLOR 5:PLOT 3
8-I,I:COLOR 3:PLOT 38-I,22-I:NEXT I
35 COLOR 32:FOR I=9 TO 13:PLOT 1,I:DRA
WTO 37,I:PLOT I+8,1:DRAWTO I+8,21:NEXT
 I
40 COLOR 46:FOR X=1 TO 15 STEP 2:FOR Y
=1 TO 7 STEP 2:PLOT X,Y:PLOT 38-X,Y:PL
OT X,22-Y:PLOT 38-X,22-Y
45 NEXT Y:NEXT X:POSITION 17,9:? "DODG
E":POSITION 13,10:? "by Matt Rutter":P
OSITION 13,12:? "SCORE:";SC
50 POSITION 9,13:FOR I=1 TO S:? CHR$(2
0);:NEXT I
60 XS=1:YS=0:AS=-2:BS=0:X=21:Y=21:A=17
:B=21:L=4:L1=4:Z1=32:T=128:C=1
70 POSITION X,Y:? CHR$(20):POSITION A,
B:? "x"
90 FOR I=1 TO INT(23*RND(1)+1):GOSUB 2
00:P=STICK(0):IF Y=10 OR Y=11 AND P=11
 THEN GOSUE 400
95 NEXT I
100 GOSUB 200
110 GOSUB 300
120 P=STICK(0):IF (X=19 OR X=20 OR Y=1
0 OR Y=11) AND (P=11 OR P=7 OR P=14 OR
 P=13) THEN GOSUB 400
130 IF A=19 OR B=11 THEN GOSUB 500
140 GOTO 100
200 POSITION X,Y:? " ":SOUND 0,0,0,0
205 IF STRIG(0)=0 AND X/2<>INT(X/2) AN
D Y/2<>INT(Y/2) THEN X=X+XS:Y=Y+YS
210 X=X+XS:Y=Y+YS:LOCATE X,Y,Z:IF Z=46
 THEN SC=SC+5:T=T-1:SOUND 0,50,10,8
220 IF Z=96 THEN SC=SC+10:T=T-1:SOUND
0,100,10,8
225 POSITION 19,12:? SC:POSITION X,Y:?
 CHR$(20):IF T<1 THEN 600
230 IF X=L*2+29 AND Y=L*2+13 THEN XS=0
:YS=-1
235 IF X=L*2+29 AND Y=9-L*2 THEN XS=-1
:YS=0
240 IF X=9-L*2 AND Y=9-L*2 THEN XS=0:Y
S=1
245 IF X=9-L*2 AND Y=L*2+13 THEN XS=1:
YS=0
290 RETURN
300 IF (X=A AND Y=B) OR (X+XS=A AND Y+
YS=B) THEN 650
305 POSITION A,B:IF Z1=46 AND C<0.3 TH
EN ? CHR$(96):GOTO 320
310 ? CHR$(Z1)
320 A=A+AS:B=B+BS:LOCATE A,B,Z1:POSITI
ON A,B:? "x"
330 IF A=L1*2+29 AND B=L1*2+13 THEN AS
=-2:BS=0
335 IF A=L1*2+29 AND B=9-L1*2 THEN AS=
0:BS=2
```

```
340 IF A=9-L1*2 AND B=9-L1*2 THEN AS=2
:BS=0
345 IF A=9-L1*2 AND B=L1*2+13 THEN AS=
0:BS=-2
350 IF (X=A AND Y=B) OR (X+XS=A AND Y+
YS=B) THEN 650
360 RETURN
400 POSITION X,Y:? " ":IF X<30 THEN 42
0
410 IF P=7 AND L<4 THEN X=X+2:L=L+1:RE
TURN
415 IF P=11 AND L>1 THEN X=X-2:L=L-1:R
ETURN
417 RETURN
420 IF X>8 THEN 440
430 IF P=7 AND L>1 THEN X=X+2:L=L-1:RE
TURN
435 IF P=11 AND L<4 THEN X=X-2:L=L+1:R
ETURN
437 RETURN
440 IF Y>8 THEN 460
450 IF P=14 AND L<4 THEN Y=Y-2:L=L+1:R
ETURN
455 IF P=13 AND L>1 THEN Y=Y+2:L=L-1:R
ETURN
457 RETURN
460 IF P=14 AND L>1 THEN Y=Y-2:L=L-1:R
ETURN
470 IF P=13 AND L<4 THEN Y=Y+2:L=L+1
490 RETURN
500 C=RND(1):IF L=L1 THEN RETURN
510 POSITION A,B:? " ":IF A<30 THEN 53
0
520 IF L>L1 THEN L1=L1+1:A=A+2:GOTO 59
0
525 L1=L1-1:A=A-2:GOTO 590
530 IF A>8 THEN 550
540 IF L>L1 THEN L1=L1+1:A=A-2:GOTO 59
0
545 L1=L1-1:A=A+2:GOTO 590
550 IF B>8 THEN 570
560 IF L>L1 THEN L1=L1+1:B=B-2:GOTO 59
0
565 L1=L1-1:B=B+2:GOTO 590
570 IF L>L1 THEN L1=L1+1:B=B+2:GOTO 59
0
580 L1=L1-1:B=B-2
590 RETURN
600 POP :POSITION 13,10:? "  **BONUS**
   "
610 FOR I=0 TO 500 STEP 20:POSITION 15
,10:? "**BONUS**":SOUND 0,70,10,15
620 IF I/40=INT(I/40) THEN POSITION 15
,10:? "**BONUS**":SOUND 0,110,10,15
630 FOR W=1 TO 20:SC=SC+1:POSITION 19,
12:? SC:NEXT W:NEXT I
640 SOUND 0,0,0,0:FOR I=1 TO 500:NEXT
I:GOTO 10
650 POP :SOUND 0,60,12,15:POSITION X-1
,Y-1:? "***":POSITION X-1,Y:? "***":PO
SITION X-1,Y+1:? "***"
660 FOR W=1 TO 100:NEXT W
670 POSITION X-1,Y-1:? "   ":POSITION
X-1,Y:? "   ":POSITION X-1,Y+1:? "   "
680 SOUND 0,0,0,0:FOR I=1 TO 500:NEXT
I:S=S-1:IF S>0 THEN 10
690 POSITION 15,8:? "GAME OVER":POSITI
ON 0,23:? "Push START to play again";
695 IF PEEK(53279)=6 THEN RUN
696 GOTO 695
```

Atari One Liners

```
10 GRAPHICS 8:FOR I=260 TO 271:READ X
:POKE I,X:NEXT I:Y=USR(260):DATA 173,
0,210,141,88,208,141,90,208,24,144,24
4
```

Wayne Kinzebach
Federal Way, VA

```
1 FORA=0 TO 2300:J=PEEK(A):SETCOLOR 2
,0,J+4:SOUND 0,J,10,8:SETCOLOR 4,0,J+
4:SOUND 0,J,10,8:NEXT A:GOTO 1
```

Dave Field
Bangor, ME

```
10 GRAPHICS 23:FOR A=1 TO 124:COLOR A
:PLOT 0,90:DRAWTO A,0:PLOT 158,90:DRA
WTO A,0:NEXT A:FOR W=1 TO 100000:NEXT
W
```

Robert Hinkle
Hinsdale, IL

The Most Exciting Event of All Time for Apple* Computer Users

For the first time ever, a computer show devoted *exclusively* to the Apple computers. Applefest '81. The largest event in the world for Apple users.

You'll see it all at Applefest. All of the latest Apple software for home, business and education. New peripherals and accessories. Useful publications and support services. Over a hundred exhibits of products and applications from around the country.

Don't miss the Applefest '81 Seminar Program, a two day series with different topics running every hour. Learn about business software, making the Apple work for you at home, programming the Apple, Apples in education and more. Meet representatives from Apple Computer and other leading Apple support manufacturers. All of the seminars are free of charge to Applefest attendees.

Regular adult admission is just $3 per day. You can register at the door, or use the pre-registration form.

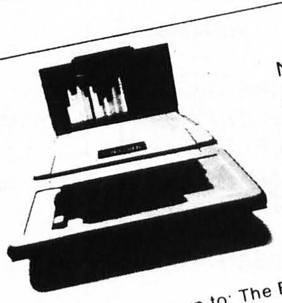
If you use the Apple, or are thinking about buying one, you won't want to miss a minute of Applefest '81.

Saturday, June 6 — Sunday, June 7, 1981

11:00 A.M. - 7:00 P.M. Saturday; 11:00 A.M. - 6:00 P.M. Sunday

The Plaza Castle

Arlington Street at Columbus Avenue, Boston, Massachusetts



Name ____________________

Street ____________________ State ________ Zip ________

City ____________________ tickets. $________ enclosed.

Please send me ________

Detach and return to: The Boston Computer Society, Three Center Plaza, Boston, Massachusetts 02108

applefest '81

Sponsored by Apple/Boston
The Boston Computer Society's Apple User Group
Three Center Plaza, Boston, Mass. 02108
(617) 367-8080

*Apple is a registered trademark of Apple Computer, Inc.

CALENDER OF CONFERENCES, WORKSHOPS, AND FAIRS

May 1-2
Third Annual Computers in Education Conference
Seattle Pacific University, Seattle, WA
Special emphasis on the use of microcomputers in grades 1-12 classrooms of various disciplines.
Contact: Jerry Johnson, Seattle Pacific University, Seattle, WA 98119

May 4-7
National Computer Conference
NCC '81 McCormick Place, Chicago, IL
Over 30 technical sessions will be held. The use of robots and artificial intelligence will be among the sessions. Debate on "Do Machines Think?"
Contact: NCC '81 PCF, c/o AFIPS, 1815 N. Lynn St., Arlington, VA 22209

May 2
National Computer Problem Solving Contest
Washington, DC
For Junior High and High School students.
Contact: Dr. Donald T. Piele, Assoc. Prof. of Mathematics, Univ. of Wisconsin, Parkside, Kenosha, WI 53141

May 11-13
31st Electronic Components Conference
Colony Square Hotel, Atlanta, GA
Papers on semiconductor-processing technology, optoelectronic devices, etc. will be read.
Contact: T.G. Grau, Bell Laboratories, Whippany Rd., Rm.3B-312, Whippany, N.J. 07981

May 11-13
Tenth ASIS Mid-Year Meeting
Fort Lewis College, Durango, CO
Theme is "Using Information". Some topics are user studies, access to information, and designing information systems for use.
Contact: ASIS, 1010 16th St. NW, Washington, D.C. 20036

May 21-23
National TRS-80™ Microcomputer Show
New York Statler Hotel, New York, NY
First computer show designed for owners of the TRS-80™ Model I, II, III, Color Videotex, and Pocket Computers. Forums and user-group seminars on topics of interest to computerists as well as commercial exhibits to browse through.
Contact: Kengore Corp., 3001 Route 27, Franklin Park, N.J. 08823

May 21-23
Expo '81
Philadelphia Civic Center
Exhibit of business and personal computers.
Contact: Philadelphia Produx, 2000 Box 2000 Bala Cynwyd, PA 19004

May 29-31
MACC Show
Franklin University Library Expansion Building, Columbus, OH
The oldest personal computing show. Speakers on microprocessor applications, exhibits, user group meetings, seminars, workshops, computer music contest, and flea market.
Contact: COMPUTERFEST '81 c/o Paul Pittenger, 215 Delhi, Apt. J, Columbus, OH, 43202

Dragonquest!

From

The Programmer's Guild

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

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

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

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

RAMWARE

Sports fans

Enjoy two of your favorites without leaving home!

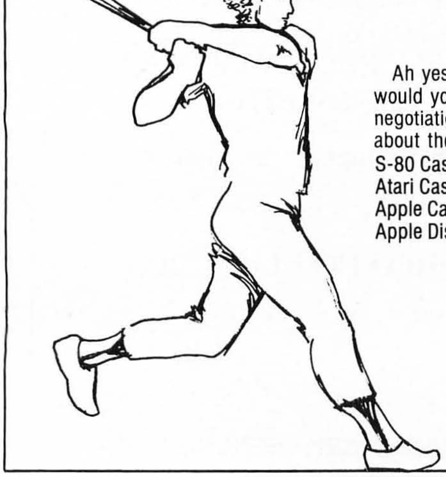
written by
David Bolke

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 Cassette #26-104004C $9.95
Atari Cassette #36-10400C $9.95



WORLD SERIES

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

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

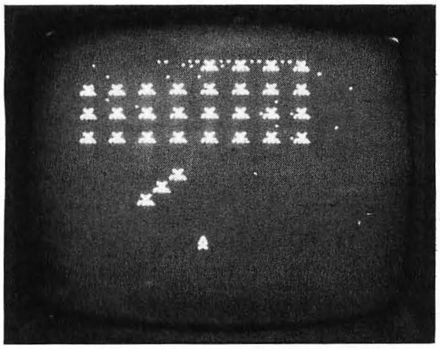
TSE·HARDSIDE
6 South St., Milford, NH 03055 (603) 673-5144
TOLL FREE OUT-OF-STATE 1-800-258-1790

GALAXIA

by Michael Prescott

"Galaxia" is an Apple Arcade game in Applesoft and requires 32K RAM.

Does the title "Galaxia" ring a bell? If it doesn't, you've probably missed out on some of the best alien invasion action to hit the arcades in a long time. The object of the game is to destroy the swarms of constantly advancing Crinoids, and score as many points as possible before losing your fifth laser base (at which point you've lost the entire galaxy to the invaders).



The game uses the Apple's high-resolution graphics screen, with movement on the screen controlled by several Machine Language routines. These execute much faster than mere BASIC commands, so that the action is very lively. Other such routines handle the various bleeps, blips, and zaps that accompany the action.

To control your laser base and to fire, use paddle 0. The knob moves your laser from side to side at the bottom of the screen, and the button fires a blast at the invading aliens. You'll notice that the paddle is more responsive to your command than in most space games; the more you play, the more you'll enjoy this feature.

The introduction of "Space Invaders" to the market initiated the space game trend. "Galaxia", although similar, provides a different type of excitement. The aliens do not fire at you from their "hovering" positions, but instead dispatch several of their number to swoop down on you in an attack formation. The quick movement of the laser base and the various antics of the Crinoids will provide you with hours of fun on your Apple.

VARIABLES

A: Number of non-flying Crinoids.
A (*, *): Flying Crinoid matrix.
A1, A2, A3, A4: Matrix components of B1, B2, B3, B4.
ACK: Number of flying Crinoids.
AP: Pointer to the next flying Crinoid's position in array A.
B: Pointer to phaser turret.
B1, B2: X and Y values of new position of flying Crinoid.
B3, B4: X and Y values of old position of flying Crinoid.
C: Vertical position of a Crinoid.
D: Ship's position in matrix-compatible form.

D1, D2: Loop variables.
DE: Pointer to vertical destination used by move routine.
DM: Pointer to destination margin used by move routine.
DX, DY: Duration and number of firings for phaser routine.
HIT: Address of explosion sound routine.
LEVEL: Level of difficulty.
LFT: Pointer for left margin used by move routine.
M1, M2: Motion commands for move routine.

MOVE: Address of Machine Language move routine.
N1, N2: Matrix pointer corrections for Crinoid.
NL: Pointer to Crinoid farthest left.
NOHIT: Used to check for a hit.
NU: Pointer to highest row of Crinoids.
P, P1: Pointers to new and old ship positions.
PHASER: Address of Machine Language phaser routine.
S: Multiplier to convert screen position to matrix position.

S (*, *): Main space matrix.
SCR: Score.
STAR: Address of star-drawing routine.
T: Multiplier to convert paddle position to screen position.
UP: Pointer to top of display used by move routine.
X, X1, Y, Z1: Loop variables.
XX, YY: General-purpose variables.
Z: Routine counter.
ZX, ZY: Star turn-off and turn-on variables for star routine.

Initialization

```
90  LOMEM: 24577: HIMEM: 31000: GOSUB
     1000
100  DIM S(20,15),A(6,3)
110  TEXT : HOME :S = 19 / 265:T =
     265 / 255
120  POKE 232,99: POKE 233,124: SCALE=
     1: ROT= 0: HCOLOR= 0:S1 = 31
     789:S2 = 31821:STAR = 31776:
     HIT = 31917:DX = 31950:DY =
     31951:PHASER = 31952:M1 = 32
     322:M2 = 32325:UP = 32247:LF
     T = 32255:DE = 32273:DM = 32
     281:MOVE = 32241
130  FOR X = 0 TO 3: FOR X1 = 0 TO
     6:A(X1,X) =  - 1: NEXT X1,X
140  VTAB 12: INPUT "SKILL LEVEL?
      (1-3) ";A$:LE =  VAL (A$): IF
     LE < 1 OR LE > 3 THEN 140
150  HGR2 :SCR = 0:AN = 0:LE = 4 -
     LE
```

Draw crinoids on screen.

```
160  IF LE < 1 THEN LE = 1
170  POKE DY,1: POKE DX,1: CALL P
     HASER: CALL PHASER: CALL PHA
     SER: CALL PHASER: POKE DX,0:
      POKE DY,60: CALL PHASER
180 Z = 0:ACK = 0:N2 = 0:NL = 5:N
     1 = 0:NU = 2:A = 8 * ( INT (
     6 / LE) + 1): FOR Y = 2 TO 8
      STEP LE: FOR X = 2 TO 16 STEP
     2:S(X,Y) = 10: XDRAW 2 AT X *
     14 + 7,Y * 10: NEXT X,Y
190  HCOLOR= 0:Z = Z + 1:P =  PDL
     (0) * T: DRAW 1 AT P1,152:B =
     P + 6:D =  INT (P * S):S(S *
     P1,15) = 0:S(D,15) = 255: XDRAW
     1 AT P,152:P1 = P
```

Make crinoids jump to empty
higher levels.

```
200  IF A = 0 OR Z <  > 7 *  INT
     (Z / 7) THEN 270
210  FOR X1 = 2 TO 16 STEP 2:XX =
     XX + S(X1,2): NEXT : IF XX +
     10 * ACK < 60 THEN 230
220  GOTO 270
230  HCOLOR= 0: FOR Y = LE + 2 TO
     8 STEP LE: FOR X = 16 TO 2 STEP
```

continued on next page

continued from previous page

```
      - 2: IF S(X,Y) = 10 THEN S(
     X,Y) = 0: GOTO 250
240  NEXT X,Y: GOTO 280
250  FOR X1 = 16 TO 2 STEP  - 2: IF
     S(X1,2) = 0 THEN S(X1,2) = 1
     0: DRAW 2 AT 14 * (X + N2) +
     7,10 * (Y + N1): XDRAW 2 AT
     14 * (X1 + N2) + 7,20 + 10 *
     N1:XX = XX + 10:X1 = 2: IF X
     X =  > 60 THEN X = 2:Y = 8
260  NEXT : GOTO 240
```

Check if all crinoids are dead.

```
270  IF ACK = 0 AND A = 0 THEN LE
      = LE - 1: GOTO 160
```

Subroutine to fire phasers.

```
280  IF  PEEK ( - 16287) < 128 THEN
     380
290  POKE DY,50: POKE DX,1: CALL
     PHASER: IF ACK = 0 THEN 330
300 NOHIT = 255: FOR X = 1 TO 5: IF
     A(X,2) = D THEN X1 = X:Y1 =
     A(X,3):X = 5:NOHIT = 0
310  NEXT : IF NOHIT THEN 330
320 C = Y1 * 10: FOR X = 0 TO 3:A
     (X1,X) =  - 1: NEXT :ACK = A
     CK - 1: CALL HIT: HCOLOR= 3:
      HPLOT B,150 TO B,C: SCALE=
     2: XDRAW 2 AT 14 * D + 7,C:
      HCOLOR= 0: XDRAW 2 AT 14 *
      D + 7,C: SCALE=
     1: DRAW 2 AT 14 * D + 7,C: HPLOT
     B,150 TO B,C:SCR = SCR + 10 ^
     (4 - LE): GOTO 380
330 NOHIT = 255: IF D <  > 2 *  INT
     ((D + N2) / 2) - N2 OR N2 >
     D THEN 360
340  FOR X1 = 8 + N1 TO 2 + N1 STEP
      - LE: IF S(D - N2,X1 - N1) <
      > 0 THEN X = X1:C = 10 * X:
     X1 = 2 + N1:NOHIT = 0
350  NEXT
360  IF NOHIT THEN  HCOLOR= 3: HPLOT
     B,150 TO B,0: HCOLOR= 0: HPLOT
     B,150 TO B,0: GOTO 380
370  CALL HIT: HCOLOR= 3: HPLOT B
     ,150 TO B,C: SCALE= 2: XDRAW
     2 AT 14 * D + 7,C: HCOLOR= 0
     : XDRAW 2 AT 14 * D + 7,C: SCALE=
     1: DRAW 2 AT 14 * D + 7,C: HPLOT
     B,150 TO B,C:S(D - N2,X - N1
     ) = 0:A = A - 1:SCR = SCR +
     10 ^ (3 - LE)
```

Subroutine to make crinoids swoop down to attack.

```
380  IF ACK = 0 THEN 550
390  FOR X = 1 TO 5: HCOLOR= 0:A1
      = A(X,0):B1 = 14 * A1 + 7:A
     2 = A(X,1):B2 = 10 * A2:A3 =
     A(X,2):B3 = 14 * A3 + 7:A4 =
     A(X,3):B4 = 10 * A4: IF A2 <
      >  - 1 THEN 410
400  NEXT : GOTO 550
410  IF D > A1 THEN X1 = 1: GOTO
     440
420  IF A1 > D THEN X1 =  - 1: GOTO
     440
430 X1 = 0
440  IF A2 = 16 THEN 510
450  DRAW 2 AT B3,B4: HCOLOR= 3: DRAW
     2 AT B1,B2:A(X,0) = A1 + X1:
     A(X,1) = A2 + 1:A(X,2) = A1:
     A(X,3) = A2: IF  INT ( RND (
     1) * 3 + 1) <  > 3 THEN 400
460  POKE 31951,50: POKE 31950,1:
      CALL 31952: HPLOT 14 * A1 +
     14 + X1,10 * A2 + 10 TO 14 *
     A1 + 14 + X1,160: HCOLOR= 0:
      HPLOT 14 * A1 + 14 + X1,10 *
     A2 + 10 TO 14 * A1 + 14 + X1
     ,160: IF D = A1 THEN  GOSUB
     480
470  GOTO 400
480  SCALE= 2: XDRAW 1 AT P,150: FOR
     Z1 = 1 TO 10: POKE  - 16336,
      PEEK ( - 16336) ^ 0 + 23 *
      PEEK ( - 16336) ^ 0 + 64 *
      PEEK ( - 16336) ^ 0: NEXT :
      XDRAW 1 AT P,150: SCALE= 1:
     AN = AN + 1
490  IF AN < 6 THEN  RETURN
500  GOTO 550
510  IF A1 = D THEN  GOSUB 480: DRAW
     2 AT B3,B4:ACK = ACK - 1: FOR
     D1 = 0 TO 3:A(X,D1) =  - 1: NEXT
     : GOTO 400
520  FOR D1 = 16 TO 2 STEP  - 2: IF
     S(D1,8) = 0 THEN S(D1,8) = 1
     0: XDRAW 2 AT 14 * (D1 + N2)
      + 7,10 * (8 + N1):A = A + 1

530  NEXT : DRAW 2 AT B3,150: FOR
     DL = 0 TO 3:A(X,DL) =  - 1: NEXT
     : FOR D1 = 2 TO 8 - LE STEP
     LE: FOR D2 = 16 TO 2 STEP  -
     2: IF S(D2,D1) = 0 THEN S(D2
     ,D1) = 10: XDRAW 2 AT 14 * (
     D2 + N2) + 7,10 * (D1 + N1):
     D2 = 2:D1 = 8 - LE:A = A + 1

540  NEXT D2,D1:ACK = ACK - 1: GOTO
     400
```

Update score and ask for another game.

```
550  IF AN < 6 THEN 600
560  HOME : TEXT : VTAB 3: PRINT
      TAB( 17)"LEVEL ";4 - LE
570  VTAB 14: PRINT  TAB( 16)"SCO
     RE:";SCR * 10
580  TEXT : VTAB 20: HTAB 14: PRINT
     "PLAY AGAIN?";: GET A$: PRINT
     : IF A$ = "N" THEN  END
590  RUN 100
```

Draw stars.

```
600 ZZ = 5 * (Z -  INT (Z / 3) *
     3): POKE S1,ZZ: POKE S2,ZX: CALL
     STAR:ZX = ZZ
```

Subroutine to make crinoids march.

```
610  IF A = 0 THEN 190
620  IF ACK <  > 0 AND Z <  > 10 *
      INT (Z / 10) THEN 680
630 YY =  SGN ( RND (1) - .5): IF
     NU - YY < 0 OR NU - YY > 3 OR
     YY = 0 THEN 630
640 XX = 2 *  SGN ( RND (1) - .5)
     : IF NL + XX < 1 OR NL + XX >
     10 THEN 640
650  IF  SGN (YY) =  - 1 THEN 670

660  POKE M1,238: POKE M2,238: POKE
     UP,10 * NU: POKE LFT,NL - 1:
      POKE DE,10 * (NU - YY): POKE
     DM,NL + XX: CALL MOVE:NU = N
     U - YY:NL = NL + XX:N1 = N1 -
     YY:N2 = N2 +  SGN (XX): GOTO
     680
670  POKE M1,206: POKE M2,206: POKE
     UP,10 * (NU + 8) - 1: POKE L
     FT,NL - 1: POKE DE,10 * (NU -
     YY + 8) - 1: POKE DM,NL + XX
     : CALL MOVE:NU = NU - YY:NL =
     NL + XX:N1 = N1 - YY:N2 = N2
      +  SGN (XX)
```

Subroutine to create marching sound.

```
680  POKE  - 16336, PEEK ( - 1633
     6): POKE  - 16336, PEEK ( -
     16336): POKE  - 16336, PEEK
     ( - 16336): IF A > 3 * ( INT
     (6 / LE) + 1) AND N1 <  > -
     2 THEN 190
```

Subroutine to determine which crinoids will drop down to attack.

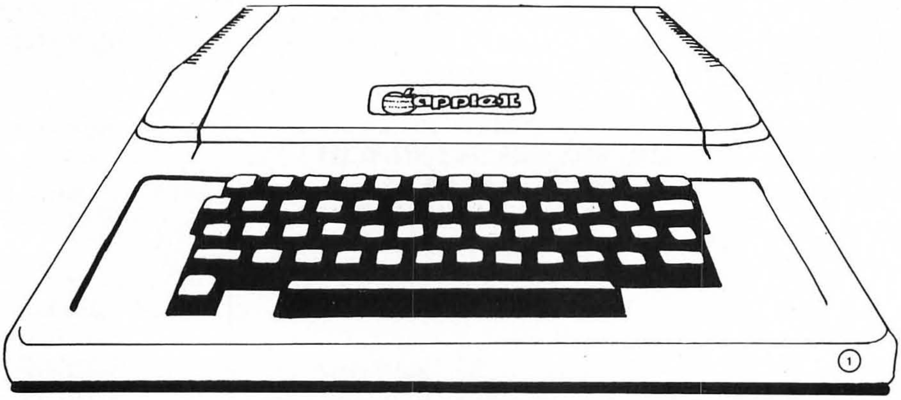
```
690  IF ACK > 4 OR LE * 5 > Z OR
     N1 >  - 2 THEN 190
700  FOR X = 1 TO 5: IF A(X,1) =
      - 1 THEN AP = X:X = 5
710  NEXT : HCOLOR= 0: FOR X = 2 TO
     16 STEP 2: IF S(X,2) <  > 0 THEN
      DRAW 2 AT 14 * (X + N2) + 7
     ,0: XDRAW 2 AT 14 * (X + N2)
      + 7,90:ACK = ACK + 1:A(AP,0
     ) = X + N2:A(AP,1) = 11 + N1
     :A(AP,2) = X + N2:A(AP,3) =
     11 + N1:S(X,2) = 0:A = A - 1
     :X = 16
720  NEXT : GOTO 190
```

Instructions.

```
1000  TEXT : HOME : VTAB 10: PRINT
      TAB( 14)"G A L A X I A": VTAB
     13: PRINT  TAB( 11)"BY MICHA
     EL PRESCOTT"
1010  GOSUB 1110
1020  HOME : PRINT "THE CRINOIDS,
      AN INTELLIGENT SPECIES OF P
     LANT LIFE, ARE NOW TRYING TO
      INVADE    YOUR GALAXY."
1030  PRINT : PRINT "THEIR GOAL I
     S TO DESTROY ALL ANIMAL LIFE
     AND MAKE THE AUTOTROPHIC FOR
     M OF LIFE   DOMINANT."
1040  PRINT : PRINT "YOUR MISSION
      IS TO DESTROY AS MANY OF
     THE INVADING CRINOIDS AS YOU
      CAN."
1050  PRINT : PRINT "WARNING!!!
     SOME OF THE CRINOIDS WILL
     SWOOP DOWN AND FIRE AT YOU.
      IF THEY"
1060  PRINT "GET TO THE BOTTOM, T
     HEY WILL REAPPEAR   WITH REI
     NFORCEMENTS."
1070  PRINT : PRINT "YOUR SHIP CA
     N TAKE ONLY FIVE HITS--THE
     SIXTH WILL DESTROY YOU, AND
     YOU WILL    LOSE THE GALAXY.
      . . ."
1080  VTAB 24: PRINT " >>>  PRESS
      SPACE BAR TO CONTINUE  <<<
     ";
1090  IF  PEEK ( - 16384) < 128 THEN
     1090
1100  POKE  - 16368,0: HOME : RETURN

1110  FOR X = 0 TO 615: READ A: POKE
     X + 31744,A:B = B + A: NEXT
     : IF B <  > 50709 OR X <  >
     616 THEN  PRINT "***DATABASE
      ERR": END
1120  RETURN
```

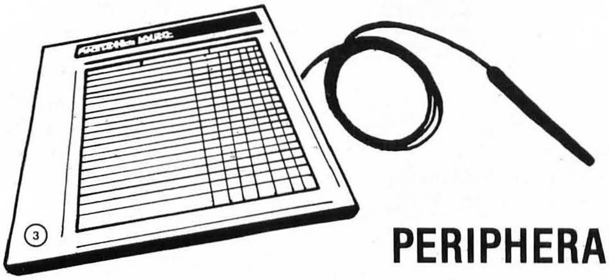
Data for Machine Language routines and shape table.

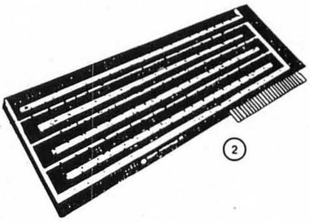
```
1130  DATA 147,211,0,98,33,16,68,
     38,76,4,10,35,51,64,68,65,64
     ,70,64,73,70,64,66,68,66,67,
     68,67,66,67

1140  DATA 67,65,160,0,173,45,124
     ,105,15,141,51,124,169,16,19
     0,0,124,142,57,124,190,25,12
     4,142,58,124,141,65,67,200

1150  DATA 192,6,208,236,160,0,17
     3,77,124,105,14,141,83,124,1
     69,0,190,0,124,142,89,124,19
     0,0,124,142,90,124,141,16

1160  DATA 70,200,192,6,208,236,2
     4,96,0,2,0,6,0,28,0,73,73,62
     ,54,55,62,14,45,54,63,55,77,
     73,60,39

1170  DATA 40,5,56,36,39,60,0,73,
     46,45,45,37,181,27,63,55,45,
     53,63,55,45,45,45,46,46,222,
     27,56,60,63

1180  DATA 62,4,56,63,62,62,0,0,0
     ,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     ,169,0,168,56,46,171,124

1190  DATA 46,172,124,144,3,173,4
     8,192,106,106,106,77,172,124
     ,10,10,10,136,152,208,1,96,7
     6,177,124,0,1,1,172,206

1200  DATA 124,173,48,192,174,207
     ,124,202,208,253,136,208,244
     ,206,207,124,208,236,96,0,0,
     0,0,0,0,0,0,0,0,0

1210  DATA 0,0,0,0,0,0,0,0,0,0,
     0,0,0,0,0,64,68,72,76,80,84,
     88,92,64,68,72,76,80,84

1220  DATA 88,92,65,69,73,77,81,8
     5,89,93,65,69,73,77,81,85,89
     ,93,66,70,74,78,82,86,90,94,
     66,70,74,78

1230  DATA 82,86,90,94,67,71,75,7
     9,83,87,91,95,67,71,75,79,83
     ,87,91,95,64,68,72,76,80,84,
     88,92,64,68

1240  DATA 72,76,80,84,88,92,65,6
     9,73,77,81,85,89,93,65,69,73
     ,77,81,85,89,93,66,70,74,78,
     82,86,90,94

1250  DATA 66,70,74,78,82,86,90,9
     4,67,71,75,79,83,87,91,95,0,
     0,0,0,0,0,0,128,128,128,12
     8,128,128

1260  DATA 128,128,0,0,0,0,0,0,0,
     0,128,128,128,128,128,128,12
     8,128,0,0,0,0,0,0,0,128,12
     8,128,128

1270  DATA 128,128,128,128,0,0,0,
     0,0,0,0,0,128,128,128,128,12
     8,128,128,128,40,40,40,40,40
     ,40,40,40,168,168

1280  DATA 168,168,168,168,168,16
     8,40,40,40,40,40,40,40,40,16
     8,168,168,168,168,168,168,16
     8,40,40,40,40,40,40,40,40

1290  DATA 168,168,168,168,168,16
     8,168,168,40,40,40,40,40,40,
     40,40,0,162,0,142,240,125,16
     0,10,190,0,125,185,120,125

1300  DATA 105,2,144,2,232,24,141
     ,41,126,141,48,126,142,42,12
     6,142,49,126,160,20,190,0,12
     5,185,120,125,105,1,144,2

1310  DATA 232,24,141,44,126,142,
     45,126,162,0,160,0,185,131,7
     2,153,1,81,138,153,131,72,20
     0,192,30,208,241,24,238,240

1320  DATA  125,169,80,205,240,12
     5,240,9,206,247,125,206,17,1
     26,76,246,125,24,96,0,0,0,12
     6,76,246,125,24,96,0,50

1330  DATA  48,48,48,48,54,50,48,
     51,49,55,52,52,90,90,96,0
```



COMPUTERS

①	APPLE II PLUS Computer, 48K RAM (#47-203)	$1299.00





PERIPHERALS

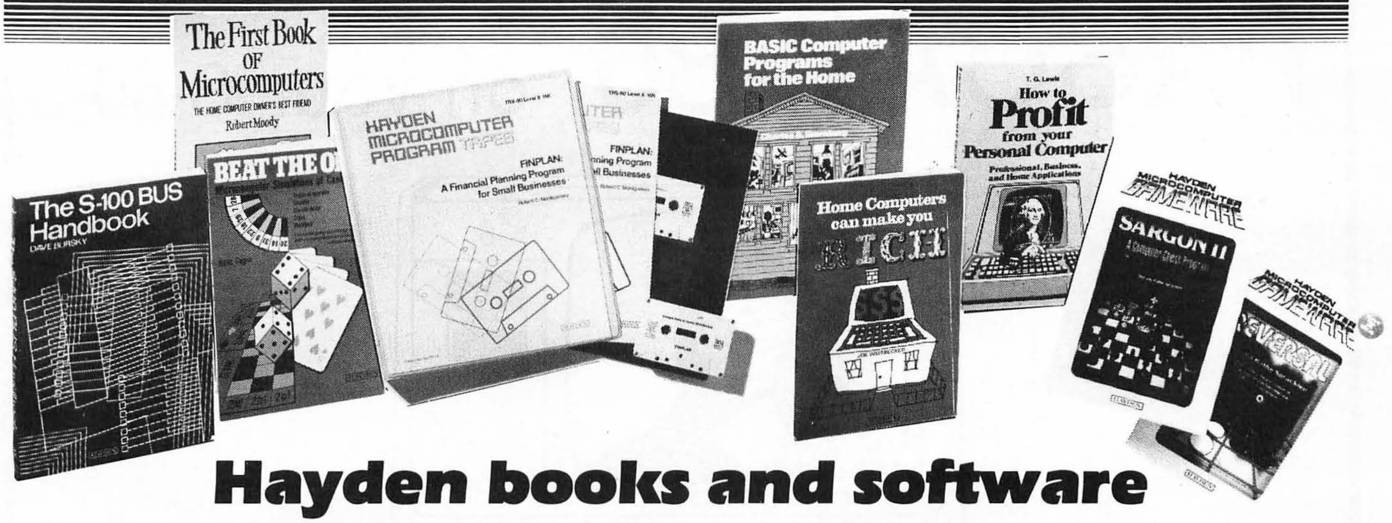
	SUP-R-MOD RF Modulator (#47-100)	$34.95
	APPLE II Disk Controller w/Drive (#47-004)	$579.00
	APPLE II Disk Drive (#47-005)	$439.00
	LOBO Disk Drive (#47-3101)	$329.00
	MICROSOFT Z-80 SoftCard (#47-80)	$279.00
	MICROSOFT RAMCard (#47-81)	$169.00
	AppleSoft BASIC Language Card (#47-ASC)	$195.00
	Integer 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

MasterCard VISA

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 suject 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 tradmark of Tandy Corp.


The First Book OF Microcomputers
THE HOME COMPUTER OWNER'S BEST FRIEND
Robert Moody
The S-100 BUS Handbook
DAVE BURSKY
BEAT THE O
HAYDEN MICROCOMPUTER PROGRAM TAPES
FINPLAN: A Financial Planning Program for Small Businesses
BASIC Computer Programs for the Home
Home Computers can make you RICH
T. G. Lewis
How to Profit from your Personal Computer
Professional, Business, and Home Applications
HAYDEN MICROCOMPUTER SOFTWARE
SARGON II
REVERSAL

Hayden books and software will help you have fun, save time, save energy, and maybe — even get rich!

NEW! HISTO-GRAPH (Boyd) A calendar-based histogram or bar-graph production system. Allows the user to enter numeric data that relates to a date, and reproduces that data as a high-resolution histogram. *09009. Apple II Disk. $29.95*

NEW! **DATA-GRAPH** (Boyd) Aids in the preparation of graphs and charts. Numeric data can be entered into Data-Graph and used to create colorful one-, two-, or four-quadrant graphs. *09109. Apple II Disk. $49.95*

FINPLAN: A Financial Planning Program for Small Businesses (Montgomery) Allows you to enter data from a balance sheet into the program, to make assumptions about the future growth of business, and to have the computer project results for up to a five year period based on those assumptions. And if you change any data, the program revises all resulting data automatically. The disk version can be used only with TRSDOS Version 2.3. *05103. TRS-80 Level II tape. $69.95; 05108. TRS-80 Level II Disk Version. $74.95*

SARGON II (Spracklen) The Champ of champs. "...an excellent program which will provide a true challenge for many players...Save your money and buy SARGON II..." *'80 Software Critque. 03403. TRS-80 Level II; 03404. Apple II; 03410. OSI C1P; 03440. OSI C4P; each tape $29.95. 03408. TRS-80 Level II Disk; 03409. Apple II Disk; 03414. OSI C1P Disk; 03444. OSI C4P Disk; 03484. C8P Disk; each $34.95*

REVERSAL (Spracklen) Winner of the software division of the *First International Man-Machine OTHELLO™ Tournament*, this version of the 200-year old game Reversi, features 27 levels of play and high-resolution color graphics. *07004. APPLE II tape. $29.95; 07009. APPLE II Disk. $34.95*

INTRODUCTION TO VSAM (Atkinson-DeSanctis) Shows the manager and technician working with a COBOL DRIVEN IBM 370 DOS/VS how to convert the files and accessing program of a data base supported by the indexed Sequential Access Method. *#5159-X. $11.95*

HOME COMPUTERS CAN MAKE YOU RICH (Weisbecker) Teaches the home computer owner and non-owner how to make money by consulting, programming, inventing. *5177-8. $6.50*

BASIC COMPUTER PROGRAMS FOR BUSINESS: Vol. 1 (Sternberg) With more than 35 programs ranging from budgets to accounts payable, each program has a description of its functions, a listing in BASIC, a symbol table and samples. *#5162-X. $9.95*

BEAT THE ODDS: Microcomputer Simulations Of Casino Games (Sagan) Provides simulations of Trente-et-Quarante, Roulette, Chemin-de-Fer, Craps, and Blackjack, all written in BASIC. *5181-6. $7.95*

THE S-100 BUS HANDBOOK (Bursky) Discusses CPU, memory, input/output, computer fundamentals and more. *#0897-X. $14.50*

DR. DOBB'S JOURNAL OF COMPUTER CALISTHENICS & ORTHODONTIA: Running Light Without Overbyte Vols. 1, 2, & 3 (The People's Computer Company) Vol. 1 (1976) holds the first ten issues of Dr. Dobb's Journal and reflects the changes that took place in personal computing. Vol. 2 (1977) documents the growth of the small computer as a tool. Vol. 3 (1978) details the new interest in programming languages. *Vol. 1. #5475-0; Vol. 2. #5484-X; Vol 3. #5490-4; each $18.95 each.*

HOW TO PROFIT FROM YOUR PERSONAL COMPUTER: Professional, Business, and Home Applications (Lewis) Describes the uses of personal computers in business applications, such as accounting, inventory, mailing lists and others. *5761-X. $10.75*

PASCAL WITH STYLE: PROGRAMMING PROVERBS (Ledgard, Nagin and Hueras) Helps PASCAL users write more accurate programs with guidelines and samples of programs. *#5124-7. $7.70*

BASIC COMPUTER PROGRAMS FOR THE HOME (Sternberg) Offers more than 75 programs such as finances, kitchen, automotive, recreational and a listing in BASIC, a symbol table and sample data. *#5154-9. $9.95*

THE FIRST BOOK OF MICROCOMPUTERS: The Home Computer Owner's Best Friend (Moody) Chapters discuss language, architecture, machine functions, home applications and much more. *#5121-2. $5.95*

BASIC BASIC: An Introduction to Computer Programming in Basic Language, Second Edition (Coan) Contains more than 100 sample problems complete with results. *#5106-9. $10.50*

SIXTY CHALLENGING PROBLEMS WITH BASIC SOLUTIONS, Second Edition (Spencer) Contains sixty games, puzzles, mathematical recreations and science and business problems. *#5180-8. $7.70*

ADVANCED BASIC: Applications And Problems (Coan) A useful book for the programmer acquiring BASIC as a second or third language or a student seeking more about BASIC. *#5855-1. $10.75*

SMALL COMPUTER SYSTEMS HANDBOOK (Libes) Enables the small computer user to purchase, assemble and interconnect components and program the microcomputer. *#5678-8. $9.95*

COMPUTERS IN ACTION: How Computers Work, Second Edition (Spencer) Features microprocessors, floppy disks, COM units and languages. *#5757-1. $7.25*

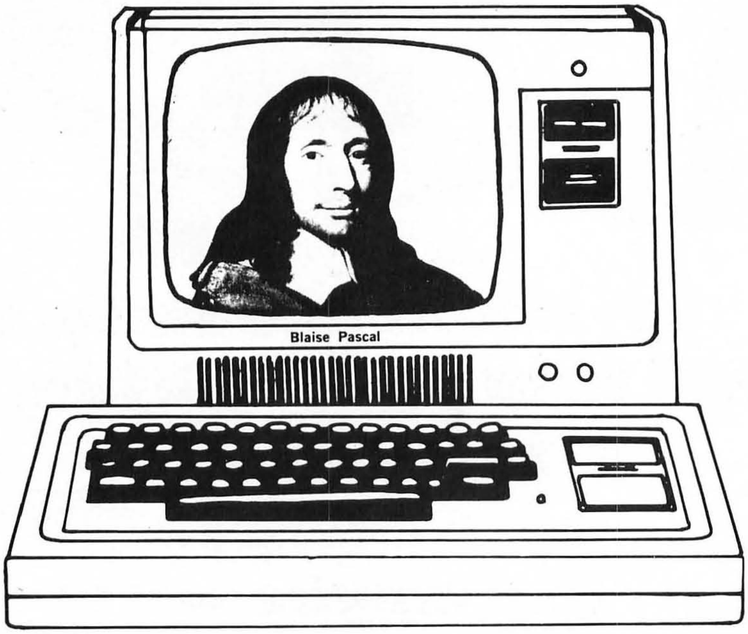
HOW TO BUILD A COMPUTER-CONTROLLED ROBOT (Loofbourrow) Contains an introduction to the robot and construction of the basic framework, power supply, speed control and directional control circuits. *#5681-8. $9.75*

The Software Exchange

6 South Street, Milford, NH 03055
TO ORDER TOLL-FREE: In NH call 673-5144
1-800-258-1790

Hayden Book Company, Inc.

PASCAL-80™ on your S-80™



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

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

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

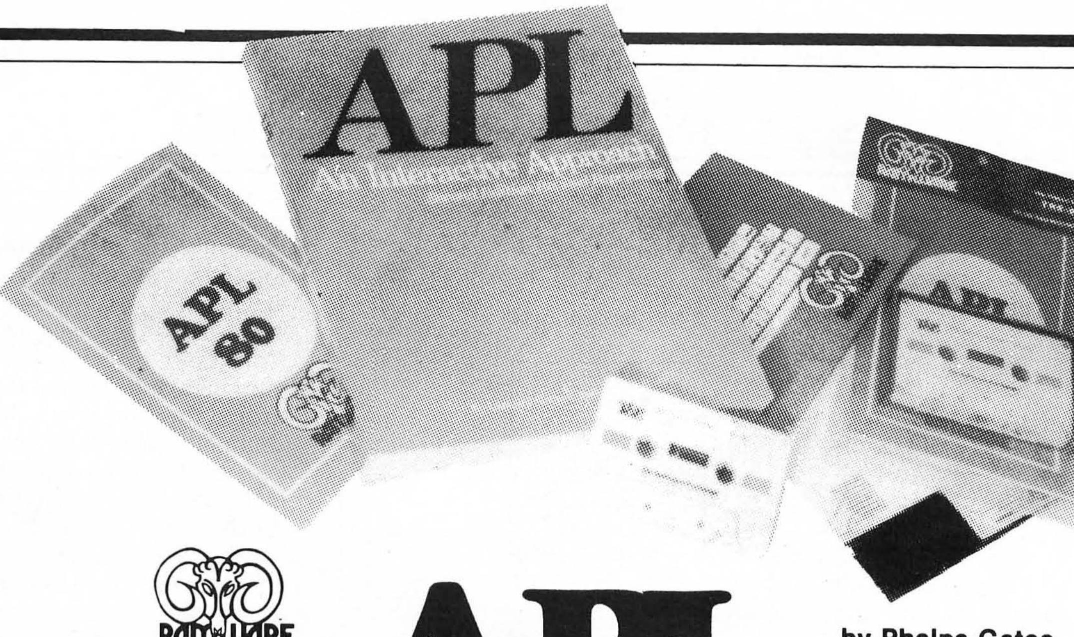
"Pascal-80" does not implement variant records, pointer and window variables, or functions and procedures used as parameters.

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

RAMWARE™

VISA MasterCard

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



RAMWARE

APL

by Phelps Gates

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

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

FEATURES

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

COMMANDS APL-80

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

SPECIFICATIONS

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

Reduced feature: 16K Level II tape version, no lessons.
Transpositions, format, and inner product not implemented. Reduced domain for some functions, 6 digit accuracy.
. . . #26-APLT . . . . $14.95 on cassette

LIMITATIONS

Due to the absence of the special APL character set on the TRS-80 , APL-80 uses shifted letters to represent the various APL characters. In addition to the keyboard limitations, lamination, domino, and matris 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.

MasterCard VISA

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

I DON'T THINK WE'RE IN KANSAS ANYMORE . . .

by Mark Pelczarski

Starting off a little different this month, I'm handwriting this column and boy does it feel good. My computer's tied up in production of a program on which I've spent most of the last three months working full-time. Free at last! I'm tired of friends making fun of my beady eyes as I occasionally wander out of the workroom for a look at the outside world.

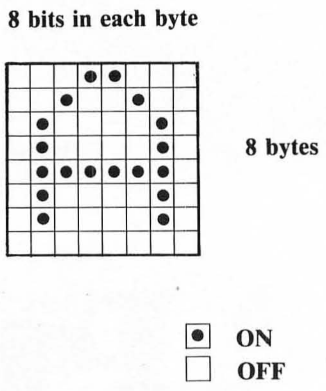
I hope Dave and the gang don't have too much trouble reading and typesetting these scratches. I'm a little out of practice. Oh, yes; I have been a little slow replying to inquiries about the continuation of "Data Base" but never fear, that's the next project.

As I recall, last time I made some ordeal about how a computer stores numbers. You, of course, never really have to worry about hexadecimal or binary unless you start using Machine Language. People have already written nice little Machine Language programs that let the computer translate normal numbers for normal people to use.

Computers also store characters: things like the letter "Y", or the symbol "%", or the numeral "9" (not to be confused with the number 9, if that's okay with you) (even if it's not) (you see, a numeral is only a symbol, a number is an amount) (I use parentheses a lot when I'm handwriting something). You're probably wondering how all these strange symbols can be stored in a contraption that only understands numbers. If so, you forgot that it doesn't even understand numbers, only ONs and OFFs (or Highs and Lows, I's and O's, or whatever you call them).

Let's go back to the unit called a byte. A byte is eight bits, eight ONs and OFFs that can hold 256 different patterns, or in binary the numbers 0 to 255. Characters are stored as a certain pattern in a byte. They are assigned binary numbers with which they are associated. The letter "A" for example, is given the binary equivalent of the number 65. There's nothing magical about associating "A" with 65; it's just a standard that's been adopted, called ASCII (American Standard Code for Information Interchange). It's not the only coding system used in the world; IBM uses their own, called EBCDIC, which stands for something I can't remember. That works out nicely for IBM, since hardware that isn't compatible with EBCDIC can't be used with their systems. Some microcomputer manufacturers have tried the IBM approach, requiring all add-on equipment to be purchased from their own company, and so far it looks like that idea fails in this marketplace. I hope no one company will get strong enough to dictate its own standards entirely; some degree of compatibility is nice to have.

Figure 1



So what happens when you type the letter "A"? A contact on the keyboard is closed, and a character chip sends a bit pattern corresponding to that key to a location in memory. If the character is to be printed, a table in memory is checked for the pattern of dots that should be turned on. You guessed it, that pattern is stored as a set of ONs and OFFs, usually eight bytes worth, one for each row of the finished character (see Figure 1). So you see, it never really understands a character; it just dutifully stores a number and occasionally looks up that numbered entry in a table and throws the information stored there on the screen.

Doesn't it get confused between numbers and letters, if they're both stored the same way? Good question. Actually, any given byte of storage can be interpreted a lot of ways. It depends on the context, much like the words "to", "two", and "too" when spoken. A single pattern within a byte can be interpreted as a character, a hexadecimal number, a Machine Language command, part of a base 10 number (and there it can be interpreted different ways — integer or floating point). It can also be interpreted as part of an address in memory (in a 64K machine, every location is assigned a unique address, a pattern that can be stored in two bytes), or it can be interpreted as a BASIC command. (The word "PRINT" is usually not stored, for example. It would take five bytes. Instead a one-byte pattern is assigned to the PRINT command). Anyway, how each pattern is interpreted depends on what previous instructions told the computer to look for when it got to that location. That's often the reason for programming errors...the computer was looking for something that could be interpreted as a BASIC instruction, for example, and it found a weird pattern that made no sense (remember, there are 256 possible patterns, and nowhere near as many BASIC commands). Such is the world of ONs and OFFs, and now, I'm off...

As Scott would say, "YOHO"...

APPLE ONE LINERS

Integer BASIC

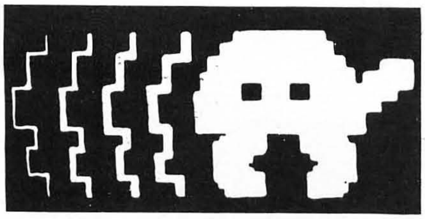
```
1 GR : FOR L=1 TO 10000: FOR
  J=0 TO L MOD 40:C=L MOD (J+
  1): COLOR=C: FOR K=0 TO J: HLIN
  0,J AT K: CALL -1953: HLIN
  39-J,39 AT 39-K: COLOR=C: NEXT
  K,J,L: END
```

T. Jackson
Whitier, CA.

Applesoft

```
0 XC = 140:YC = 80:XL = 0:YL = 0:
   HGR : FOR I = 0 TO 1E + 9:X
   = POL(0) / 255 =  PDL (0) /
  255 * 279:Y =  PDL (1) / 255
   * 159:F = ( PEEK ( - 16287)
   > 127):XC = XC + F * (X = X
  C): HCOLOR= 0:YC = YC + F *
  (Y - YC): HPLOT XC,YC TO XL,
  YL: HCOLOR= 7: HPLOT XC,YC TO
  X,Y:XL = X:YL = Y: NEXT
```

Joe Humphrey
Topeka, KS.

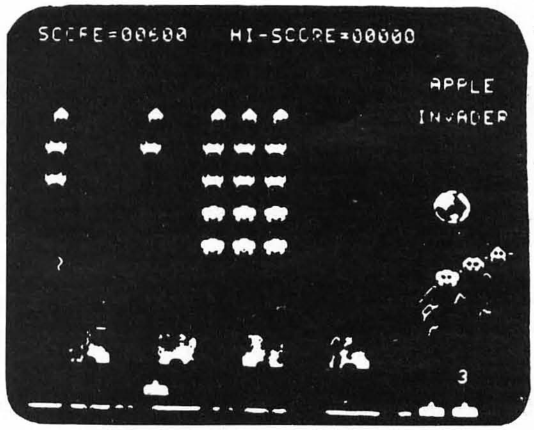


FROM

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.

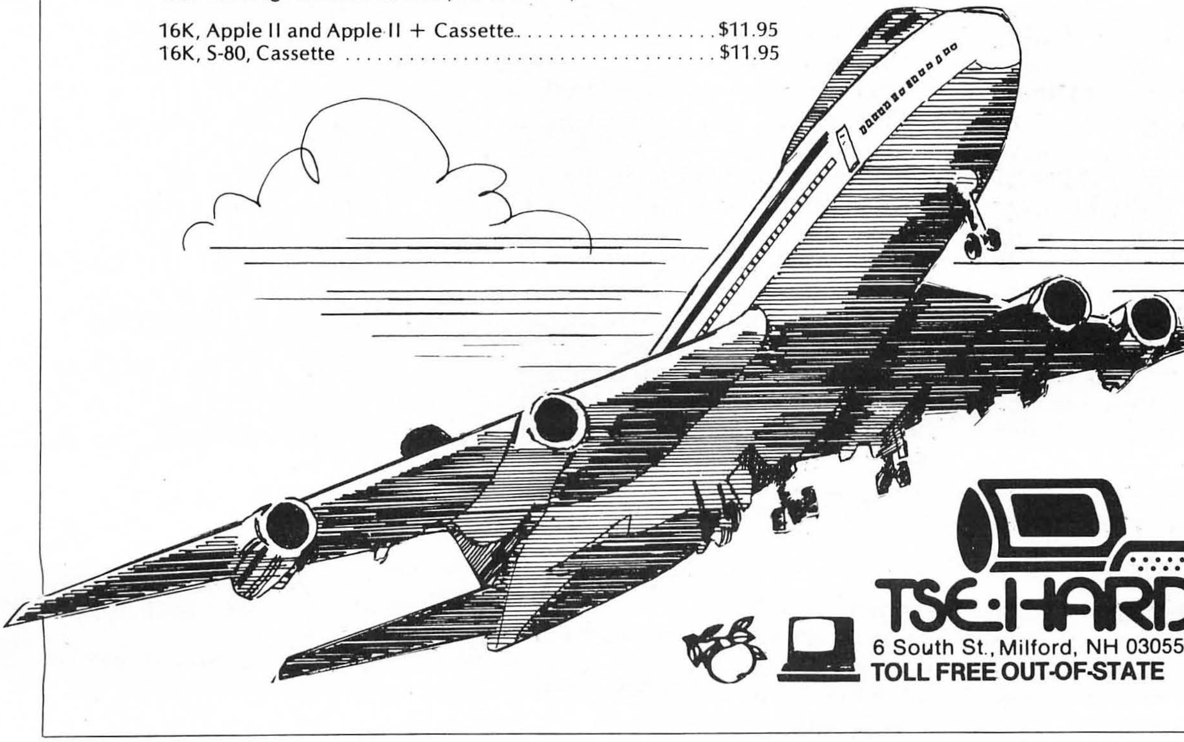
Super Invasion **Only**
32K, Apple II, Cassette $19.95

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

DAIRY FARMING

by Scott Tapley

"Dairy Farming" is an S-80 simulation requiring 16K RAM.

In this simulation you get half a million dollars to buy a farm and get started in your own dairy farming business. The object is to accumulate 10,000 points, based on your total assets of milk, land, cash, cows, etc. After buying one of three farms shown to you by the computer, you go to the auction and buy some cows. You then enter the daily cycle of the game, which is in three phases.

PHASE ONE

In the first phase you have six options: (1) Milk the cows, (2) Buy feed, (3) Buy cows, (4) Feed the cows, (5) Take the cows to the pasture to graze, and (6) Sell milk. When you begin phase one you should buy feed if needed, unless you are going to take the cows to the pasture. Although it costs nothing to let them graze, the grass will eventually get eaten and you will need to let it grow back for a few days. If you don't feed the cows at all, two things will happen. First, the cows will become too weak to produce good milk, and some of the milk will have to be thrown out. And second, one of the cows will die later in the day.

After feeding the cows, you will want to milk them. You'll notice that each type of cow gives a different amount of milk — from Jersey, through Guernsey, Ayrshire, and Brown Swiss, to Holstein. If there isn't enough room in your coolers to hold all the milk, the extra will have to be sold at a cheaper price to get rid of it quickly. When you finish milking, you can then sell some or all of your milk to the local milkman — if you have at least 200 gallons to sell. Each time you sell, the price per gallon will change. With the money you can buy a cow, or wait to buy something else. You can also keep the money as a means to get to your 10,000-point goal.

PHASE TWO

The second phase of each day's cycle involves buying (and possibly selling) capital goods. You can buy or sell some land, buy a large or small cooler, buy another barn, buy another silo (small, medium, or large), or even sell the whole farm (which, of course, ends the simulation). Except for selling land, all these things help you to accumulate points. Just remember not to spend too much, or you'll have to take out a loan.

PHASE THREE

In the third and final phase of the work day, the computer calculates the interest on your loan (if you have one), and asks you if you would like to make a payment on it. Your point total is also calculated, on the following basis —

One point for every:
$10,000 in cash.
5 acres of land.
500 gallons of milk.
Small cooler.
7 Jersey cows.
6 Guernsey cows.
5 Ayrshire cows.
4 Brown Swiss cows.
3 Holstein cows.
2 calling birds.
And a partridge in a pear tree.
(Just Kidding.)
Two points for every large cooler.
Four points for every small silo.
Five points for every barn or medium silo.
Six points for every large silo.
MINUS one point for every $100 borrowed.

If you don't yet have the necessary 10,000 points to win, you start a new day with phase one again. When (and if!) you reach your goal, you are given a rating based on the number of days elapsed, and you can go back to sleeping in past sunrise every morning.

VARIABLES

A: Acres of land owned.
AY: Ayrshire cows owned.
BB: Your bid.
BID: The current bid.
BN: Number of barns owned.
BR: Random increase of BID.
BS: Brown Swiss cows owned.
C: Total number of cows owned.
CE: Indicates whether cows have been fed or not.
CM: Total capacity of your coolers (in gallons).
CO$: Type of cow.
DA: Day.
D$: Type of cow that died.
ES: Total point value of your silos.
F: The farm you picked (1, 2, or 3).
FV: Current value of your farm.
G: Gallons of milk in your cooler.
GU: Guernsey cows owned.
HO: Holstein cows owned.
I: Time delay loop variable.
JE: Jersey cows owned.
K: Gallons of milk that didn't fit in your cooler.
L: Interest rate on your loan.
LA: Large coolers owned.
LO: Loan balance.
MC: Cash on hand.
MI: Gallons of milk produced that day.
OP, OP$: Input from keyboard.
P: Price per acre of land.
PB: Used to determine if bid is high enough to buy cow.
PE: People at the auction.
PER: Percentage of silo capacity actually filled.
PV: Number of days of good grass left in your pasture.
S: Cubic feet of feed in your silos.
SH$: Color of man's shirt.
R: Random number.
VA: Gallons of milk each cow gave that day.
VB: Value of farm at beginning of game.
W: Number of acres bought.
WP: Your total points.
WW: Used to value W at end of game.

```
     DAIRY FARMING  --  PROGRAM LISTING

 Instructions and initialization.

20 CLS:PRINT:PRINT:PRINT:PRINT"   IN THIS SIMULATION OF DAIRY FA
RMING, YOU WILL HAVE 500,000   DOLLARS TO BUY A FARM AND GET STA
RTED IN YOUR DAIRYING CAREER."
30 PRINT:PRINT"   THE OBJECT OF THE GAME IS TO ACCUMULATE 10,000
 POINTS.  YOU  CAN GET POINTS FOR THINGS SUCH AS MILK, CASH, LAN
D, COWS, ETC.":PRINT:PRINT:PRINT
40 GOSUB1670:GOSUB1650
50 PRINT:PRINT"              THE FIRST THING YOU DO IS BUY A FARM.
               EACH FARM CONSISTS OF 1 BARN, 1 SMALL HOUSE, 1
LARGE SILO,"
60 PRINT"1 REFRIGERATED MILK COOLER, AND 1 HAYSHED. THE ONLY DIF
FERENCE      IS THE NUMBER OF ACRES THAT THE FARM HAS FOR GRAZ
ING."
```

```
70 PRINT:PRINTTAB(9)"FARM #","1","2","3"
80 PRINTTAB(9)"ACRES","140","115","95"
90 PRINTTAB(9)"PRICE","$439,000","$399,000","$369,000"
100 CLEAR 275:INPUT"WHICH FARM WOULD YOU LIKE";F
110 IF F<1ORF>3THENGOTO100ELSE120
120 ON F GOTO 130,140,150
130 A=140:MC=61000:FV=439000:PV=-20:GOTO160
140 A=115:MC=101000:FV=399000:PV=-12:GOTO160
150 A=95:MC=131000:FV=369000:PV=-7:GOTO160
160 RANDOM:BN=1:G=0:PER=0:DA=1:CE=0:CM=500:SC=8000:LO=0:L=0:WP=0
:LA=1:SA=0:ES=6:WW=900
```

The auction.

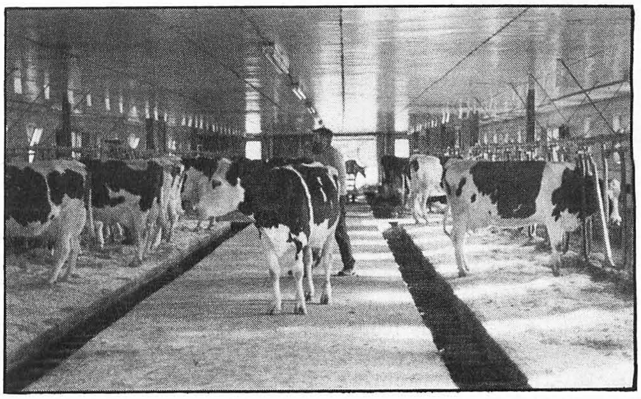
```
170 CLS:PRINT"   NOW IT'S TIME TO GO TO THE AUCTION TO BUY SOME
CATTLE.":FORI=1TO500:NEXTI
180 CLS:PRINT"            THERE ARE FIVE DIFFERENT KINDS OF COWS."
190 PRINT"LISTED IN ORDER OF MILK PRODUCTION THEY GO ..."
200 PRINT"                 # 1. HOLSTEIN"
210 PRINT"                 # 2. BROWN SWISS"
220 PRINT"                 # 3. AYRSHIRE"
230 PRINT"                 # 4. GUERNSEY"
240 PRINT"                 # 5. JERSEY"
250 INPUT"WHICH KIND DO YOU WANT TO BUY (1-5)";OP
260 IF OP <1 OR OP>5THEN GOTO250
270 ON OP GOTO 280,290,300,310,320
280 CO$="HOLSTEIN":BID=1500:PB=1500:GOTO330
290 CO$="BROWN SWISS":BID=1400:PB=1400:GOTO330
300 CO$="AYRSHIRE":BID=1325:PB=1325:GOTO330
310 CO$="GUERNSEY":BID=1250:PB=1250:GOTO330
320 CO$="JERSEY":BID=1185:PB=1185:GOTO330
330 CLS:PE=RND(50)-1:IFPE<10THENGOTO330ELSEPRINT"                THER
E ARE";PE;"PEOPLE AT THE AUCTION TODAY.":PRINT"
  THE NUMBER OF PEOPLE AT THE                                  AUCTION
 MAY DETERMINE HOW HIGH YOU HAVE TO BID"
340 PRINT"   THE BIDDING ON THIS ";CO$;" HAS JUST BEGUN WITH A B
ID OF                         $";BID
350 PRINT"THE CURRENT BID IS NOW $";BID:INPUT"DO YOU WANT TO BID
 AGAIN (Y/N)";OP$:IFOP$="N"THENGOTO520ELSEGOTO360
360 INPUT"WHAT IS YOUR BID";BB
370 IF BB<BID THEN GOTO360
380 IF MC-BB<0THEN PRINT"YOU DON'T HAVE ENOUGH MONEY TO BID THAT
 HIGH!!"ELSEGOTO400
390 FORI=1TO500:NEXTI:GOTO350
400 BID=BB:BB=0:IFBID>PB+(PE*25)+RND(500)THENGOTO430
410 BR=RND(25):IFBR<11THENGOTO410ELSEBID=BID+BR
420 GOTO350
430 IFCO$="HOLSTEIN"THENHO=HO+1ELSEGOTO440
440 IFCO$="BROWN SWISS"THENBS=BS+1ELSEGOTO450
450 IFCO$="AYRSHIRE"THENAY=AY+1ELSEGOTO460
460 IFCO$="GUERNSEY"THENGU=GU+1ELSEGOTO470
470 IFCO$="JERSEY"THENJE=JE+1
480 MC=MC-BID:CLS:PRINTCHR$(23):PRINT@450,"         S O L D !!!!":
PRINT"          FOR $";BID
490 INPUT"DO YOU WANT TO BUY ANY MORE COWS (Y/N)";OP$
500 IF OP$="Y"THENGOTO180
510 IF DA=>1THENGOTO580ELSEGOTO570
520 R=RND(3):ON R GOSUB 540,550,560
530 PRINT"SOLD, TO THE MAN IN THE ";SH$;" SHIRT.":FORI=1TO500:NE
XTI:GOTO490
540 SH$="YELLOW":RETURN
550 SH$="GREEN":RETURN
560 SH$="STRIPED":RETURN
```

Displays everything you own and asks you what transaction you would like to make.

```
570 CLS:PRINT"  NOW THAT YOU ARE DONE BUYING YOUR CATTLE, YOU MU
ST MILK THEM    AND SELL THE MILK FOR A PROFIT. NOW LET'S GO MIL
K THEM COWS.":GOSUB1670
580 GOSUB 1650
590 PRINT"    DAY";DA;"                                  YOU HAVE $";
INT(MC):PRINT"YOUR FARMING EQUIPMENT CONSISTS OF:
600 C=HO+BS+AY+GU+JE
610 PRINT"1 NO";F;"FARM WITH";A;"ACRES OF LAND AND";BN;"BARN(S).
"
620 GOSUB1680
630 PRINT"YOU HAVE";INT(G);"GALLONS OF MILK IN YOUR";CM;"GALLON
COOLER(S)."
640 PER=S/SC:PRINT"YOUR SILO(S) ARE";(PER*100);"% FULL"
650 GOSUB1680
660 PRINTTAB(26)"# OF COWS"
670 PRINT"   HOLSTEIN      BROWN SWISS     AYRSHIRE      GUERNSEY
   JERSEY"
680 R=640:PRINT@R+5,HO;:PRINT@R+20,BS;:PRINT@R+35,AY;:PRINT@R+48
,GU;:PRINT@R+59,JE;
690 PRINT:GOSUB1680
700 PRINT"1. MILK COWS      2. BUY FEED      3. BUY COWS"
710 PRINT"4. FEED COWS      5. TAKE COWS TO PASTURE  6. SELL MILK
"
720 INPUT"ENTER OPTION (0 TO CONTINUE).";OP
730 IF OP<1THENGOTO1700
740 IF OP>6THENGOTO700
750 IF OP=3THENZ=1ELSEGOTO760
760 ON OP GOTO 770,980,180,1130,1280,1450
```

Milking the cows.

```
770 IF PA>0THENPRINT"YOU'VE ALREADY MILKED THEM TODAY!!"ELSEGOTO
800
780 FORI=1TO750:NEXTI
790 GOTO580
800 MI=0:GOSUB1650
810 VA=RND(2)+RND(3)+(RND(9)/10):VA=VA*2
820 IFHO>0THENPRINT"YOUR HOLSTEIN COW(S) GAVE";HO*VA;"GALLONS OF
 MILK"ELSEGOTO830
830 VA=VA-.2:IFBS>0THENPRINT"YOUR BROWN SWISS COW(S) GAVE";BS*VA
;"GALLONS OF MILK"ELSEGOTO840
840 VA=VA-.2:IFAY>0THENPRINT"YOUR AYRSHIRE COW(S) GAVE";AY*VA;"G
ALLONS OF MILK"ELSEGOTO850
850 VA=VA-.2:IFGU>0THENPRINT"YOUR GUERNSEY COW(S) GAVE";GU*VA;"G
ALLONS OF MILK"ELSEGOTO860
860 VA=VA-.2:IFJE>0THENPRINT"YOUR JERSEY COW(S) GAVE";JE*VA;"GAL
LONS OF MILK"
870 PRINT
880 MI=MI+(HO*(VA+.8))+(BS*(VA+.6))+(AY*(VA+.4))+(GU*(VA+.2))+(J
E*VA):PA=1:G=G+MI
890 R=RND(50)-1:IFR>37ORR<10THENGOTO890  ELSEIFCE=0THENPRINT"SIN
CE YOU DIDN'T FEED YOUR COWS,";R;"PERCENT OF THE MILK WASN'T  GO
```



continued on next page

continued from previous page

```
OD ENOUGH TO BE SOLD AND HAS BEEN THROWN OUT."ELSEGOTO910
900 R=R/100:G=G-(MI*R)
910 IF G>CMPRINT
920 IF G>CMTHENPRINT"YOUR COOLER ISN'T BIG ENOUGH TO HOLD ALL OF
 THAT MILK...
YOU WILL HAVE TO SELL THE EXTRA MILK AT $.75 PER GALLON.":ELSEGO
TO970
930 K=G-CM:IFK>0MC=MC+(K*.75):G=CM
940 IF K>0PRINT
950 IF K>0PRINT"YOU GOT $";(K*.75);"FOR THAT EXTRA MILK."
960 K=0
970 GOTO 1660
```

Buying grain.

```
980 GOSUB 1650
990 PRINT"YOUR SILO IS";(PER*100);"% FULL."
1000 PRINT"IT COSTS $2.50 A DAY TO FEED A COW."
1010 PRINT"YOU HAVE";C;"COWS AND $";INT(MC);"."
1020 PRINT"HOW MANY DAYS WORTH OF FOOD WOULD YOU LIKE TO BUY"
1030 INPUTOP
1040 IF OP*C*2.5>MCTHENPRINT"YOU DON'T HAVE ENOUGH MONEY"ELSEGOT
O1060
1050 FORI=1TO750:NEXTI:GOTO1020
1060 IF OP<=0THENGOTO580ELSEGOTO1070
1070 PRINT"THAT COST YOU $";OP*C*2.5;
1080 S=S+OP*C*2.5:PER=S/SC
1090 MC=MC-(OP*C*2.5)
1100 PRINT"YOU HAVE $";INT(MC);"LEFT."
1110 IF PER > 1 THEN PRINT"YOUR SILO IS FULL!!!"ELSEGOTO1120
1120 GOSUB1670:GOTO580
```

Feeding the cows.

```
1130 IF CE>0THENGOTO1250 ELSEGOSUB1650
1140 OP$=""
1150 IF S - (C*2.5) < 0 THEN PRINT@460,"BUT YOU DON'T HAVE ENOUG
H TO FEED ALL
      YOUR COWS. YOU WILL HAVE TO BUY MORE FEED FIRST."ELSEGOTO1
180
1160 FORI=1TO1400:NEXTI
1170 GOTO580
1180 PRINT:PRINT:INPUT"           HIT ENTER TO FEED COWS (D FOR D
ON'T FEED)";OP$
1190 IFOP$="D"GOTO580
1200 S=S-(C*2.5):PER=S/SC:PV=PV-1
1210 PRINTTAB(20)"COWS ARE BEING FED"
1220 FORI=1TO500:NEXTI
1230 CE=1
1240 GOTO580
1250 CLS:PRINT@464,"YOU HAVE ALREADY FED THEM TODAY"
1260 FORI=1TO850:NEXTI
1270 GOTO580
```

Taking the cows to the pasture.

```
1280 IFCE>0THENGOTO1420ELSEGOSUB1650
1290 PRINT
1300 IF PV>9THENPRINT"YOUR COWS HAVE EATEN ALL OF THE GOOD GRASS
"ELSEGOTO1340
1310 PRINT"IN THE PASTURE. IN OTHER WORDS, YOU WILL HAVE TO LET"
1320 PRINT"THE GRASS GROW FOR A FEW DAYS."
1330 PRINT:GOTO1690
1340 IFC>ATHENPRINT@455,"YOU NEED TO BUY MORE LAND!!!"ELSEGOTO13
60
1350 FORI=1TO500:NEXTI:GOTO1700
1360 PRINT:PRINT"                   COWS ARE ENTERING PASTURE":FOR
I=1TO200:NEXTI
1370 PRINT:PRINT"                   COWS ARE EATING GRASS":FORI=
1TO200:NEXTI
1380 PRINT:PRINT"                   COWS ARE DONE EATING":FORI=1
TO200:NEXTI
1390 PRINT:PRINT"                 COWS ARE GOING BACK TO BARN":FO
RI=1TO200:NEXTI
1400 CE=1:PV=PV+1
1410 GOTO580
1420 CLS:PRINT@466,"YOU ALREADY FED THEM TODAY"
1430 FORI=1TO850:NEXTI
1440 GOTO580
```

Selling milk.

```
1450 GOSUB1650
1460 IF G<0THENG=0
1470 IFG<200 THEN PRINT@459,"YOU DON'T HAVE ENOUGH MILK TO SELL
YET!"ELSEGOTO1500
1480 FORI=1TO950:NEXTI
1490 GOTO580
1500 BP=RND(125):IFBP<90THENGOTO1500ELSEGOTO1510
1510 BP=BP/100:PRINT:PRINT"  THE LOCAL MILKMAN WILL BUY YOUR MI
LK FOR $";BP;"PER GALLON.":PRINT"                 YOU HAVE";INT
(G);"GALLONS OF MILK."
1520 PRINT:INPUT"      HOW MANY GALLONS OF MILK DO YOU WISH TO
SELL";OP
1530 IF G-OP<0THENPRINT"YOU DON'T HAVE THAT MUCH MILK!!"ELSEGOTO
1550
1540 FORI=1TO350:NEXTI:GOTO1520
1550 MC=MC+(OP*BP):G=G-OP:GOTO580
```

Taking out a loan.

```
1560 CLS:PRINT@465,"YOU HAVE SPENT TOO MUCH MONEY ...
             YOU WILL HAVE TO TAKE OUT A LOAN."
1570 FORI=1TO500:NEXTI
1580 PRINT"YOU ARE $";ABS(MC);"IN DEBT."
1590 L=RND(19):PRINT"THE CURRENT INTEREST RATE IS";L;"%"
1600 INPUT"HIT ENTER TO ACCEPT THE LOAN.";LI
1610 LO=LO+ABS(INT(MC))
1620 MC=0
1630 L=L/300
1640 RETURN
```

Short subroutines.

```
1650 CLS:PRINT@24,"DAIRY FARMING":RETURN
1660 PRINT:INPUT"HIT ENTER TO CONTINUE";OP:GOTO580
1670 INPUT"HIT ENTER TO CONTINUE";OP:RETURN
1680 PRINT STRING$(63,":"):RETURN
1690 PRINT:INPUT"HIT ENTER TO CONTINUE";OP
```

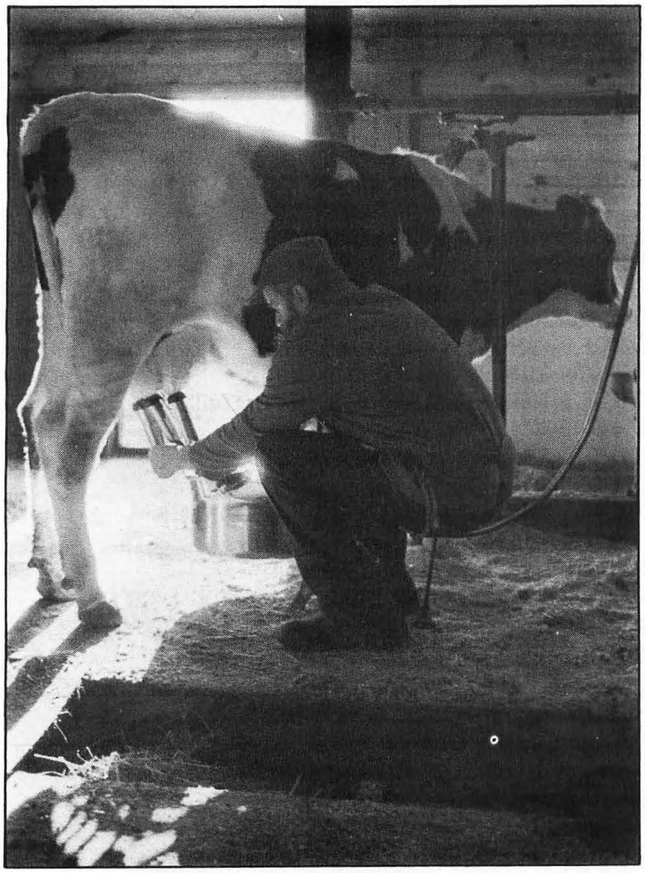
Asks what you want to buy or sell.

```
1700 GOSUB1650
1710 PRINT:PRINT"AT THIS TIME YOU MAY BUY ANY NECESSARY THINGS F
OR YOUR FARM."
1720 IF C>(BN*35) THEN PRINTTAB(20)"(YOU NEED A BIGGER BARN!)"EL
SEPRINT
1730 GOSUB1680
1740 PRINT:PRINT"1. BUY MORE LAND FOR GRAZING"
1750 PRINT"2. SELL LAND FOR QUICK CASH"
1760 PRINT"3. BUY A LARGER OR SMALLER COOLER"
1770 PRINT"4. BUY A BIGGER BARN (EACH BARN HOLDS 25 COWS)"
1780 PRINT"5. BUY A LARGER SILO"
1790 PRINT"6. SELL FARM AND END GAME"
1800 PRINT:INPUT"ENTER OPTION NUMER DESIRED (0 TO CONT.)";OP
1810 IF OP<1THENGOTO2540
1820 IF OP>6THENGOTO1800
```

```
1830 ON OP GOTO 1840,1930,2030,2110,2190,2330
```

Buying land.

```
1840 GOSUB1650
1850 P=RND(1750):IF P<850 THEN GOTO1850 ELSEPRINT:PRINT"AN ACRE
OF LAND WILL COST YOU $";P
1860 INPUT"HOW MANY ACRES DO YOU WISH TO BUY";OP
1870 IFOP*P>MCTHEN PRINT"YOU DON'T HAVE ENOUGH MONEY"ELSEGOTO190
0
1880 FORI=1TO250:NEXTI
1890 GOTO1860
1900 IFOP=0THENGOTO1700 ELSEPRINT"THAT COST YOU $";(OP*P);". Y
OU NOW HAVE $";MC-(P*OP);
```



```
1910 MC=MC-(P*OP):A=A+OP
1920 GOTO1690
```

Selling land.

```
1930 GOSUB1650
1940 P=RND(1750):IF P<950THENGOTO1940ELSEPRINT:PRINT"YOU MAY SEL
L ANY GIVEN AMOUNT OF LAND FOR $";P
1950 INPUT"HOW MANY ACRES DO YOU WISH TO SELL";OP
1960 IFOP<1THENGOTO1700ELSEPRINT"YOU GOT $";(OP*P);"FOR THAT LAN
D."
1970 IF (A-OP)<CTHENGOTO2020
1980 PRINT"YOU NOW HAVE $";MC+(OP*P);
1990 MC=MC+(OP*P):A=A-OP
2000 GOTO1690
2010 MC=MC-2500:CM=CM+200:IF MC<0GOSUB1560
2020 CLS:PRINT@462,"YOU DON'T HAVE ENOUGH ACRES LEFT TO SELL!:";
FORI=1TO875:NEXTI:GOTO1700
```

Buying milk coolers.

```
2030 GOSUB1650
2040 OP$=""
2050 PRINT:PRINT:INPUT"                               DO YOU WANT A
                  <L>ARGE COOLER - $4,500
                  <S>MALL COOLER - $2,500
                     <N>ONE OF ABOVE";OP$
2060 IF OP$="N"THENGOTO1700ELSEIFOP$="L"THENGOTO2070ELSEIFOP$="S
"THENGOTO2090ELSEGOTO2030
2070 MC=MC-4500:CM=CM+500:LA=LA+1:IF MC<0GOSUB1560
2080 GOTO1700
2090 MC=MC-2500:CM=CM+200:SA=SA+1:IF MC<0GOSUB1560
2100 GOTO1700
```

Buying barns.

```
2110 GOSUB1650
2120 OP$=""
2130 PRINT@275,"A BARN COSTS $5,000."
2140 PRINT@520,"(IF YOU DON'T WANT IT, HIT ANY OTHER KEY)"
2150 PRINT@319,"":INPUT"              HIT ENTER TO PURCHASE BARN
.";OP$
2160 IFOP$=""THENGOTO2170ELSEGOTO1700
2170 BN=BN+1:MC=MC-5000:IF MC<0GOSUB1560
2180 GOTO1700
```

Buying silos.

```
2190 GOSUB1650
2200 PRINT:PRINT:PRINT"                    # 1. LARGE SILO -- $7,50
0"
2210 PRINT"              # 2. MEDIUM SILO - $6,000
2220 PRINT"              # 3. SMALL SILO -- $5,000"
2230 PRINT"              # 4. NO SILO"
2240 INPUT"           WHICH SILO WOULD YOU LIKE TO BUY";OP
2250 IFOP<1OROP>4GOTO2240
2260 ON OP GOTO 2270,2290,2310,2320
2270 MC=MC-7500:SC=SC+8000:ES=ES+6:IF MC<0GOSUB1560
2280 GOTO1700
2290 MC=MC-6000:SC=SC+7000:ES=ES+5:IF MC<0GOSUB1560
2300 GOTO1700
2310 MC=MC-5000:SC=SC+6000:ES=ES+4:IF MC<0GOSUB1560
2320 GOTO1700
```

Selling the farm.

```
2330 GOSUB1650
2340 ON F GOTO 2350,2360,2370
2350 VB=439000:W=140:GOTO2380
2360 VB=399000:W=115:GOTO2380
2370 VB=369000:W=95:GOTO2380
2380 PRINT:PRINT"WHEN YOU BOUGHT YOUR FARM IT WAS WORTH $";VB
2390 PRINT"(THIS INCLUDES THE";W;"ACRES THE FARM CAME WITH)"
2400 PRINT"BUT NOW IT'S WORTH $";FV+INT(FV/10);:PRINT
2410 GOSUB2520:IF M<0M=0
2420 IF A-W>0THENPRINT"THE LAND YOU BOUGHT IS NOW WORTH $";MELSE
PRINT"YOU DIDN'T BUY ANY ADDITIONAL LAND"
2430 R=RND(35)-1:PRINT"YOUR COWS ARE WORTH $";C*(1400+(R*35));
2440 PRINT:PRINT"YOU HAVE";INT(WP);"POINTS"
2450 PRINT:INPUT"ARE YOU SURE YOU WANT TO SELL YOUR FARM (Y/N)";
OP$
2460 IF OP$="N"THENGOTO1700ELSEGOTO2480
2470 FV=FV+INT(FV/9):FV=(FV+MM)+C*(1400+(R*35))
2480 PRINT:PRINT"YOU HAVE JUST BEEN PAID, IN CASH, $";FV;
2490 MC=MC+FV:PRINT:PRINT"YOU NOW HAVE $";MC
2500 PRINT:INPUT"DO YOU WANT TO PLAY AGAIN (Y/N)";OP$
2510 IF OP$="Y"THENRUNELSEEND
2520 W=A-W:R=RND(2000):IF R<WWGOTO2520
```

continued on next page

continued from previous page

```
2530 M=(W*R):RETURN
```

Sets up the next day.

```
2540 FV=FV+INT(FV*.01):WW=WW+10:PA=0:DA=DA+1:LO=INT(LO)+INT(LO*L
)
```

Checks to see if you fed your cows; if not, one dies.

```
2550 IFMC>0ANDLO>0THENGOTO2630
2560 IFCE=1THENGOTO2730ELSEIFC=0THENGOTO2730ELSER=RND(5):ON R GO
SUB 2580,2590,2600,2610,2620
2570 GOSUB1650:PRINT@448,"SINCE YOU DIDN'T FEED YOUR COWS, ONE O
F YOUR ";D$;" COWS DIED.":FORI=1TO1000:NEXTI:GOTO2730
2580 D$="HOLSTEIN":IF HO=0THENGOTO2560ELSEHO=HO-1:RETURN
2590 D$="SWISS":IF BS=0THENGOTO2560ELSEBS=BS-1:RETURN
2600 D$="AYRSHIRE":IF AY=0THENGOTO2560ELSEAY=AY-1:RETURN
2610 D$="GUERNSEY":IF GU=0THENGOTO2560ELSEGU=GU-1:RETURN
2620 D$="JERSEY":IF JE=0THENGOTO2560ELSEJE=JE-1:RETURN
```

Making payment on your loan (if you have one).

```
2630 GOSUB1650
2640 PRINT:PRINT:PRINT:INPUT"WOULD YOU LIKE TO MAKE A PAYMENT ON
 YOUR LOAN (Y/N)";OP$
2645 IFOP$="N"THENPRINT"YOU HAVE $";LO;"OUT ON YOUR LOAN":GOTO27
10
2650 PRINT"YOU HAVE $";LO;"OUT ON YOUR LOAN AND $";MC;"."
2660 INPUT"HOW MUCH WOULD YOU LIKE TO PAY";OP
2670 IFOP=0THENGOTO2700ELSEIFMC<OPTHENPRINT"YOU DON'T HAVE ENOUG
H MONEY"ELSEGOTO2690
2680 FORI=1TO450:NEXTI:GOTO2650
2690 LO=LO-OP:MC=MC-OP:IFLO<=0THENPRINT"YOUR LOAN IS PAID OFF!!"
:FORI=1TO750:NEXTI:OP=0:GOTO2560
2700 PRINT"YOU STILL HAVE $";LO;"OUT ON YOUR LOAN."
2710 FORI=1TO1000:NEXTI
2720 GOTO2560
```

Adds up all your points and tells you how many points you have.

```
2730 IFHO=0THENGOTO2740ELSEWP=WP+(HO/3)
2740 IFBS=0THENGOTO2750ELSEWP=WP+(BS/4)
2750 IFAY=0THENGOTO2760ELSEWP=WP+(AY/5)
2760 IFGU=0THENGOTO2770ELSEWP=WP+(GU/6)

2770 IFJE=0THENGOTO2780ELSEWP=WP+(JE/7)
2780 IFBN=0THENGOTO2790ELSEWP=WP+(BN*5)
2790 IFG=0THENGOTO2800ELSEWP=WP+(G/500)
2800 WP=WP+ES
2810 WP=WP+(A/5)
2820 WP=WP+LA*2
2830 WP=WP+SA
2840 IFMC=0THENGOTO2850ELSEWP=WP+(MC/10000)
2850 IFLO<=0THENGOTO2860ELSEWP=WP-(LO/100)
2860 GOSUB1650
2870 PRINT@470,"YOU HAVE";INT(WP);"POINTS
              (";(INT(WP)/10000)*100;"% OF YOUR GOAL)"
2880 CE=0:PA=0:FORI=1TO700:NEXTI
```

If you have the winning 10,000 points, the computer gives you a rating based on the number of days it took to achieve your goal.

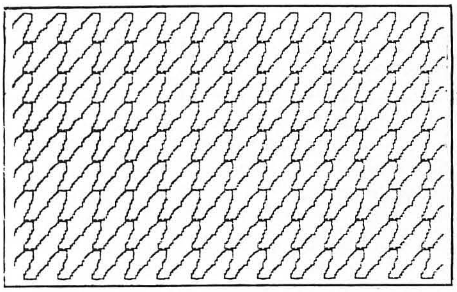
```
2890 IF WP>9999THENGOTO2900ELSEGOTO580
2900 GOSUB1650
2910 OP$="       !!  YOU DID IT  !!  YOU DID IT  !!  YOU DID IT
!!":FORI=1TO13:PRINTOP$:NEXTI
2920 IF DA>125THENOP$="NEEDS IMPROVEMENT - A LOT"ELSEGOTO2940
2930 GOTO 3010
2940 IF D>118THENOP$="COULD BE BETTER"ELSEGOTO2960
2950 GOTO 3010
2960 IF DA>112THENOP$="AVERAGE"ELSEGOTO2980
2970 GOTO 3010
2980 IF DA>105THENOP$="PRETTY GOOD !"ELSEGOTO3000
2990 GOTO 3010
3000 IF DA<=105THENOP$="DAIRY FARMER !!"
3010 PRINT"YOU HAVE 10,000 POINTS ----- RATING : ";OP$
3020 INPUT"DO YOU WISH TO PLAY AGAIN (Y/N)";OP$:IFOP$="Y"THENRUN
ELSEEND
```

COMPUTER GRAPHICS

Computer-aided Pattern Generation

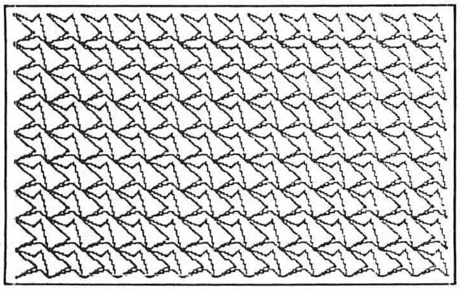
by Joan Truckenbrod

Creating patterns with the aid of a computer is very exciting as a wide variety of designs can be generated. Numerous different shapes, forms and figures can be used to create intricately detailed ornamental patterns. Repeating a figure at regular intervals on a grid generates a pattern. For instance, a pattern can be created on graph paper by drawing the same figure in each square. The figure used to create the pattern is described in terms of X and Y coordinates. The number of points used to describe the figure is included in line 40 in the program in the variable NP. The X and Y coordinates are listed sequentially in the DATA statement in line 80. The size of the figure and the resolution of the graphics screen determine the number of times the figure is repeated horizontally and vertically. For example, on the Apple II Microprocessor system a figure 20 units by 20 units can be repeated 14 times across the screen and nine times down the screen.



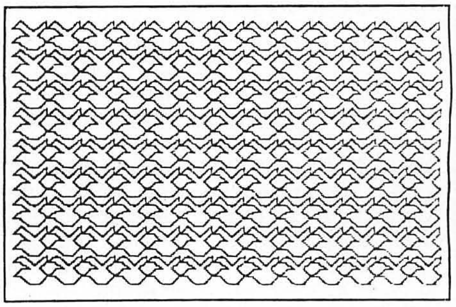
Carefully calculate the size of the figure and the number of times it is repeated in relationship to the resolution of the graphics screen. As the examples of this pattern program illustrate, the character of the pattern is dependent upon the design of the figure used to create the pattern. This figure can be a closed shape, a line, a shaded figure or a figure consisting of varying gray values. Figures can be curved or straight line forms. Pictorial shapes can be used as well as abstract forms in creating patterns.

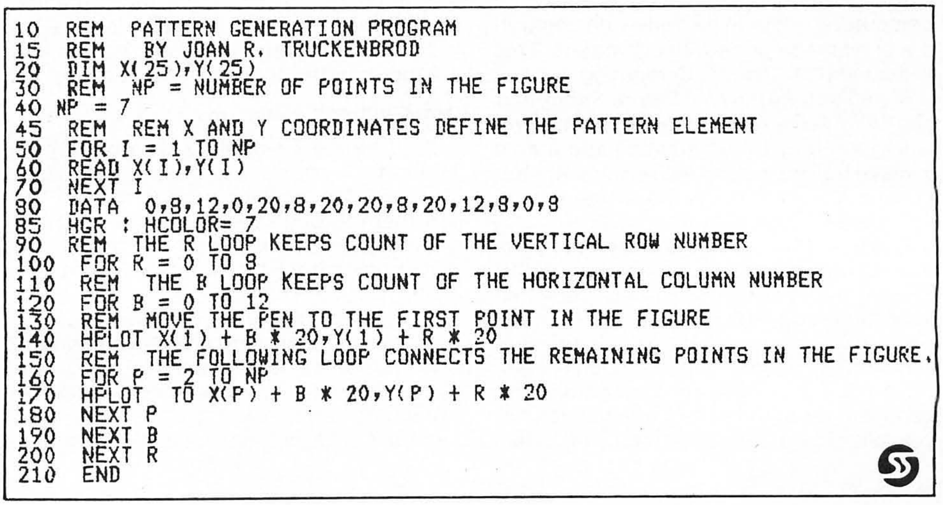
A pattern is created in a series of steps with the aid of a computer. First the figure is constructed in an imaginary box 20 units across and 20 units down with the upper left-hand corner of the box at the origin (0,0), and the X and Y coordinates of each point in the figure are identified. The figure is initially drawn in this position and then is repeated at regular intervals in each box of an imaginary pattern grid. This grid is filled with the pattern elements from left to right, and top to bottom. The figures are repeated in the horizontal row by incrementing all of the X coordinates in the figure by a given value.



Generally the value of this increment is the width of the shape so that the shapes touch one another as they are repeated. In the following pattern generation program the horizontal repetition of the figure takes place in the B FOR/NEXT loop. After the horizontal row of figures is completed the figure must be moved down the height of the figure in order to construct another row of figures. The figure is moved down by adding the height of the figure to all of the Y coordinates defining the figure. After the Y coordinates are incremented, another horizontal row of figures can be drawn from left to right by incrementing the X values again. However, since it is necessary to always begin a row from the left side, the X coordinates in the X array must not be changed, rather this increment value can be added in the HPLOT statement. The Y coordinates do not need to be returned to their original values and thus cumulative Y values can be used. In the pattern program the R FOR/NEXT loop increments the Y coordinates so that a series of horizontal rows will be drawn.

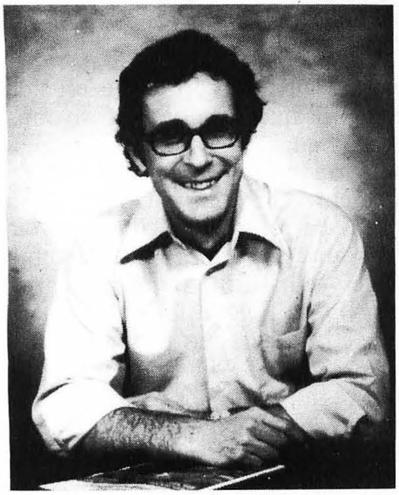
Using the computer to create a pattern provides the designer with a new system to develop a wide range of patterns. A regular pattern is created with the repetition of one shape or figure on a grid. More variation can be created in a pattern by systematically changing the color, shape, and/or size of the pattern element as it is repeated. In this instance the pattern consists of the repetition of a series of related shapes rather than the repetition of a single shape. The potential for creating a wide variety of patterns provides an exciting challenge for the beginning or advanced designer. Variations on this general pattern generation program will be discussed in future issues of **SoftSide.**





```
10   REM  PATTERN GENERATION PROGRAM
15   REM   BY JOAN R. TRUCKENBROD
20   DIM X(25),Y(25)
30   REM  NP = NUMBER OF POINTS IN THE FIGURE
40  NP = 7
45   REM  REM X AND Y COORDINATES DEFINE THE PATTERN ELEMENT
50   FOR I = 1 TO NP
60   READ X(I),Y(I)
70   NEXT I
80   DATA  0,8,12,0,20,8,20,20,8,20,12,8,0,8
85   HGR : HCOLOR= 7
90   REM  THE R LOOP KEEPS COUNT OF THE VERTICAL ROW NUMBER
100   FOR R = 0 TO 8
110   REM  THE B LOOP KEEPS COUNT OF THE HORIZONTAL COLUMN NUMBER
120   FOR B = 0 TO 12
130   REM  MOVE THE PEN TO THE FIRST POINT IN THE FIGURE
140   HPLOT X(1) + B * 20,Y(1) + R * 20
150   REM  THE FOLLOWING LOOP CONNECTS THE REMAINING POINTS IN THE FIGURE.
160   FOR P = 2 TO NP
170   HPLOT  TO X(P) + B * 20,Y(P) + R * 20
180   NEXT P
190   NEXT B
200   NEXT R
210   END
```

A REMARKABLE MAGAZINE



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 lighthearted 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 understand 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 acessible to the beginner, theoretically non-trivial, interesting on more than one level, and perhaps even humorous.

David Gerrold of *Star Trek* fame says, "*Creative Computing* with its unpretentious, down-to-earth lucidity encourages the computer user to have fun. *Creative Computing* makes it possible for me to learn basic programming skills and use the computer better than any other source.

Hard-hitting Evaluations

At *Creative Computing* we obtain new computer systems, peripherals, and software as soon as they are announced. We put them through their paces in our Software Development Center and also in the environment for which they are intended—home, business, laboratory, or school.

Our evaluations are unbiased and accurate. We compared word processing printers and found two losers among highly promoted makes. Conversely, we found one computer had far more than its advertised capability. Of 16 educational packages, only seven offered solid learning value.

When we say unbiased reviews we mean it. More than once, our honesty has cost us an advertiser—temporarily. But we feel that our first obligation is to our readers and that editorial excellence and integrity are our highest goals.

Karl Zinn at the University of Michigan feels we are meeting these goals when he writes. "*Creative Computing* consistently provides value in articles, product reviews and systems comparisons . . . in a magazine that is fun to read."

Order Today

To order your subscription to *Creative Computing*, send $20 for one year (12 issues), $37 for two years (24 issues) or $53 for three years (36 issues). If you prefer, call our toll-free number, **800-631-8112** (in NJ 201-540-0445) to put your subscription on your MasterCard, Visa or American Express card. Canadian and other foreign surface subscriptions are $29 per year, and must be prepaid. We guarantee that you will be completely satisfied or we will refund the entire amount of your subscription.

Join over 80,000 subscribers like Ann Lewin, Director of the Capital Children's Museum who says, "I am very much impressed with *Creative Computing*. It is helping to demystify the computer. Its articles are helpful, humorous and humane. The world needs *Creative Computing*."

creative computing
Attn: Elyse
P.O. Box 789-M
Morristown, NJ 07960
Toll-free **800-631-8112**
(In NJ 201-540-0445)

The story behind the two best selling computer games books in the world.

Computer Games

by David H. Ahl

Everybody likes games. Children like tic tac toe. Gamblers like blackjack. Trekkies like Star Trek. Almost everyone has a favorite game or two.

It Started in 1971

Ten years ago when I was at Digital Equipment Corp. (DEC), we wanted a painless way to show reluctant educators that computers weren't scary or difficult to use. Games and simulations seemed like a good method.

So I put out a call to all our customers to send us their best computer games. The response was overwhelming. I got 21 versions of blackjack, 15 of nim and 12 of battleship.

From this enormous outpouring I selected the 90 best games and added 11 that I had written myself for a total of 101. I edited these into a book called 101 Basic Computer Games which was published by DEC. It still is.

When I left DEC in 1974 I asked for the rights to print the book independently. They agreed as long as the name was changed.

Contents of *Basic Computer Games* (right) and *More Basic Computer Games* (below).

Artillery-3
Baccarat
Bible Quiz
Big 6
Binary
Blackbox
Bobstones
Bocce
Boga II
Bumbrun
Bridge-It
Camel
Chase
Chuck-A-Luck
Close Encounters
Column
Concentration
Condot
Convoy
Corral
Countdown
Cup
Dealer's Choice
Deepspace
Defuse
Dodgem
Doors
Drag
Dr. Z
Eliza
Father
Flip
Four In A Row
Geowar
Grand Prix
Guess-It
ICBM
Inkblot
Joust
Jumping Balls
Keno
L Game

Life Expectancy
Lissajous
Magic Square
Man-Eating Rabbit
Maneuvers
Mastermind
Masterbagels
Matpuzzle
Maze,
Millionaire
Minotaur
Motorcycle Jump
Nomad
Not One
Obstacle
Octrix
Pasart
Pasart 2
Pinball
Rabbit Chase
Roadrace
Rotate
Safe
Scales
Schmoo
Seabattle
Seawar
Shoot
Smash
Strike 9
Tennis
Tickertape
TV Plot
Twonky
Two-to-Ten
UFO
Under & Over
Van Gam
Warfish
Word Search Puzzle
Wumpus 1
Wumpus 2

Introduction
The Basic Language
Conversion to Other Basics
Acey Ducey
Amazing
Animal
Awari
Bagels
Banner
Basketball
Batnum
Battle
Blackjack
Bombardment
Bombs Away
Bounce
Bowling
Boxing
Bug
Bullfight
Bullseye
Bunny
Buzzword
Calendar
Change
Checkers
Chemist
Chief
Chomp
Civil War
Combat
Craps
Cube
Depth Charge
Diamond
Dice
Digits
Even Wins
Flip Flop
Football
Fur Trader
Golf
Gomoko
Guess
Gunner
Hammurabi
Hangman
Hello
Hexapawn

Hi-Lo
High I-Q
Hockey
Horserace
Hurkle
Kinema
King
Letter
Life
Life For Two
Literature Quiz
Love
Lunar LEM Rocket
Master Mind
Math Dice
Mugwump
Name
Nicomachus
Nim
Number
One Check
Orbit
Pizza
Poetry
Poker
Queen
Reverse
Rock, Scissors, Paper
Roulette
Russian Roulette
Salvo
Sine Wave
Slalom
Slots
Splat
Stars
Stock Market
Super Star Trek
Synonym
Target
3-D Plot
3-D Tic-Tac-Toe
Tic Tac toe
Tower
Train
Trap
23 Matches
War
Weekday
Word

Converted to Microsoft Basic

The games in the original book were in many different dialects of Basic. So Steve North and I converted all the games to standard Microsoft Basic, expanded the descriptions and published the book under the new name Basic Computer Games.

Over the next three years, people sent in improved versions of many of the games along with scores of new ones. So in 1979, we totally revised and corrected Basic Computer Games and published a completely new companion volume of 84 additional games called More Basic Computer Games. This edition is available in both Microsoft Basic and TRS-80 Basic for owners of the TRS-80 computer.

Today Basic Computer Games is in its fifth printing and More Basic Computer Games is in its second. Combined sales are over one half million copies making them the best selling pair of books in recreational computing by a wide margin. There are many imitators, but all offer a fraction of the number of games and cost far more.

The games in these books include classic board games like checkers. They include challenging simulation games like Camel (get across the desert on your camel) and Super Star Trek. There are number games like Guess My Number, Stars and Battle of Numbers. You'll find gambling games like blackjack, keno, and poker. All told there are 185 different games in these two books.

Whether you're just getting started with computers or a proficient programmer, you'll find something of interest. You'll find 15-line games and 400-line games and everything in between.

The value offered by these books is outstanding. Every other publisher has raised the price of their books yet these *sell for* the same price as they did in 1974.

Moneyback Guarantee

Examine one or both of these books and key some games into your computer. If you're not completely satisfied we'll refund the full purchase price plus your return postage.

Basic Computer Games costs only $7.50 and More Basic Computer Games just $7.95 for either the Microsoft or TRS-80 edition (please specify your choice on your order). Both books together are $15. Send payment plus $2.00 shipping and handling to Creative Computing Press, Morris Plains, NJ 07950. Visa, MasterCard and American Express orders should include card number and expiration date. Charge card orders may also be called in toll-free to 800-631-8112 (in NJ 201-540-0445).

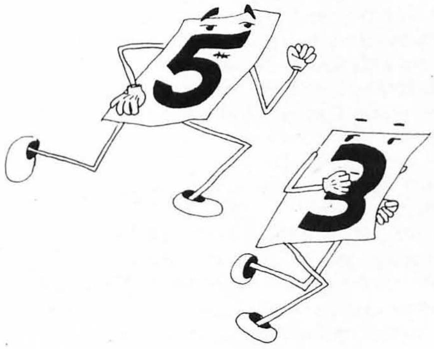
Order today to turn your computer into the best game player on the block.

creative computing

Morris Plains, NJ 07950
Toll-free **800-631-8112**
(In NJ 201-540-0445)

MATH DECATHLON

Part 3
by Jon Voskuil



Once again, welcome to the Olympics. This third installment features (not one! not two! but...) three brand-new events for your programming and gaming pleasure.

If this is the first installment of "Math Decathlon" that you've seen, don't despair, because the first two installments are available (along with many other goodies) in the March and April back issues. Or, if you have an Apple tape or disk subscription, there's no problem at all — you have the complete program (to date) ready to load.

Assuming that you're just a plain plebeian, though, and are doing your own typing, it's that time again. Load the program from tape or disk as you've entered it so far, take a few deep breaths of fresh springtime air (not applicable in certain regions), and get ready to caress the keys for awhile.

A few preliminary lines need to be added or changed. Line 100 has just one slight addition (HTAB 1) to make sure it doesn't mess up the screen, and lines 310 and 320 are filled out with new event titles. Line 330 is also added, to keep 320 from getting overly long.

EVENT #5: LOWEST COMMON DENOMINATOR

Maybe fractions aren't as exciting as starry nights (event #4), but they do have their uses. In this game two or three fractions are displayed (all with 1 as the numerator), and you are challenged to find their lowest common denominator. After printing the *instructions* in lines 5000-5040, two or three random numbers are chosen in lines 5050-5090, their size limited by the skill level. Line 5080 sends control down to 5130 if the skill level is zero — that is, if only two fractions are to be generated and displayed.

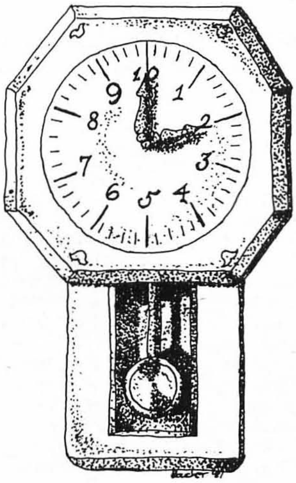
The calculating of the L.C.D. is done in lines 5110-5120 (for three fractions) or 5140-5160 (for two). The value of the variable LCD starts out, in line 5100 or 5130, being equal to the product of the denominators, D. Then, for all integers from D/2 down to 2, a check is made to see if all the denominators might be evenly divisible by that integer. If so, the value of LCD is revised downward accordingly. This checking, which can take a couple of seconds in some cases, is done while the fractions are being displayed, so that there isn't an inappropriate pause in execution. You know how embarrassing that can be! You know, too (if you just look) that the fractions are printed either in line 5100 or 5130, and that the player's input is grabbed and graded in 5170.

EVENT #6: MAGIC SQUARES

What you may not know is what a "magic square" is. If not, let me have the pleasure of introducing these marvelous mathematical creatures to you. They consist of a square grid filled with numbers — numbers such that adding them across any row, down any column, or along either of the corner-to-corner diagonals, will always give the same sum. This sum is called the "magic number". Magic squares can be created with additional interesting properties as well; but in this decathlon event we're sticking with simple, third-order (3 by 3) squares. If you're interested in finding out more about these creatures, your local library should have a book or two on the topic.

Lines 6060-6080 generate the values for the square. Three numbers — A, B, and C — are chosen at random in 6060 and 6070, their limits keyed to the skill level as usual. These are then used in 6080 to compute the nine values for the magic square, SQ(1) through SQ(9). Of course, the point of the event is that the square is displayed with one or more of the numbers missing. So lines 6090-6130 choose one or more locations for the missing values. Note that X, Y, and Z are not the unprinted numbers, but rather the locations (1-9) of those numbers in the square. If there are only two missing numbers, Z remains equal to zero; and if only one, then Y also is zero. Having done all the preliminaries, the square is then printed by lines 6140-6200, with the letters "X", "Y", and "Z" being inserted where appropriate.

Lines 6210 and following ask the player to input the missing number(s), and then they compare the inputs to the correct values. One or more of the variables X$, Y$, and Z$ are used as input variables, after initializing all of them in 6210. Their values are then compared with the corresponding values of the SQ array in line 6240, and if they all match then the player gets ten points. Note that the value of SQ(0) is zero, so that if Z is zero (meaning that there is no third missing number), then SQ(Z) will match the value of Z$ (which also remains at zero), and will pass the scoring test — and the same goes for Y.



EVENT #7: CLOCK MATH

Now, if you think that squares of numbers are magic, how about clocks that have more or less than 60 minutes in an hour? We start becoming accustomed to thinking in 60s fairly early in life; but if an hour were split into 50 minutes, or 85, then we'd have some rethinking to do. Which is precisely the point of this event. The computer sets the number of minutes per hour in this new space-time continuum, picks a time of day and a number of minutes, and asks what time it will be after those minutes have elapsed.

This is actually a thinly-veiled exercise in "modular arithmetic", as you astute mathematicians have undoubtedly deduced. If you've never heard of modular arithmetic — well, this is it! Most of our everyday arithmetic is done with a "modulus" of 10, or "mod 10". Which merely means that after you reach the tenth digit (9), you start over with the first (0), carrying one over into the next place to the left. We use modular arithmetic all the time, not only in counting hours and minutes (mod 60), but also in counting eggs (mod 12), yards (mod 3 or mod 36), and all kinds of other good stuff.

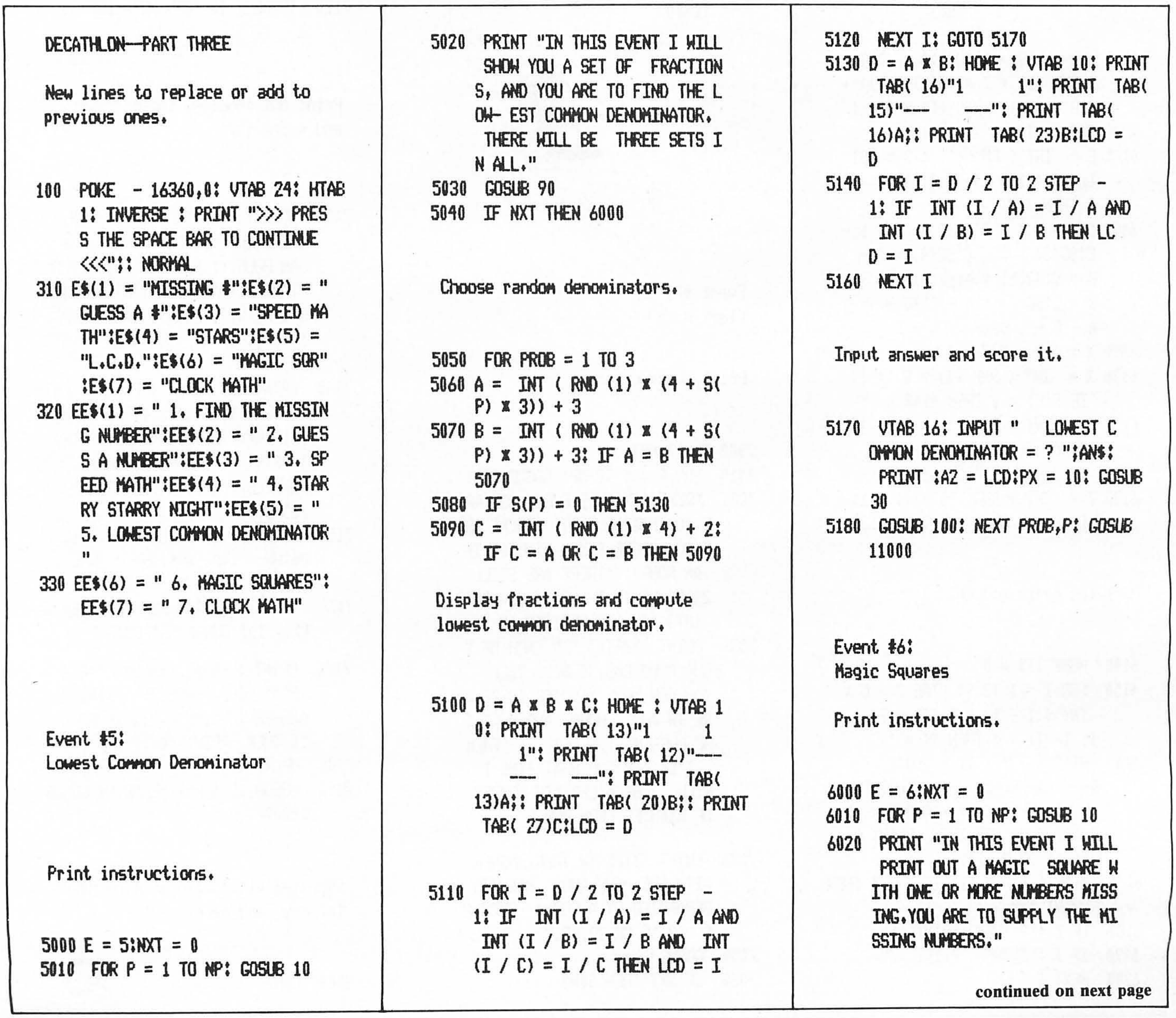
Anyway, this is why there's a variable in this event's program lines called MOD. It is the number of minutes in an hour, and is always set to 50 on skill level 0, or some other value at higher levels (lines 7070-7080). Line 7090 sets the current time as H1:M1 (hour:minutes) and picks a value for M, the number of minutes to elapse. Lines 7100-7120 compute the future time based on these, as H2:M2. If you have trouble following the logic of these three lines, just pick some numbers for MOD, H1, M1, and M, and carry out the specified tests and instructions on paper to see how they work.

Lines 7130 and 7140 print the problem on the screen; and the remaining coding, from line 7150 on, is a straightforward input and scoring routine. The POKE at the end of 7140 merely assures that some stray character left over in the keyboard buffer won't be counted as part of the player's input.

Well, that's "Math Decathlon" for this month. Not a single translation note in the whole muddy puddle! Next month's installment will conclude the series, with the last three exciting events and (at last) the routine for playing the grand winner's full national song. Or maybe you will have already written that routine yourself by then.

VARIABLES (Part 3):

A, B, C: Denominators used in event #5; numbers used to generate magic squares in event #6.
D: Product of denominators.
H: Elapsed hours (computed from elapsed minutes).
H1: Current time (hour).
H2: Future time (hour).
II: Print-position indicator.
LCD: Lowest common denominator.
M: Elapsed minutes.
M1: Current time (minutes).
M2: Future time (minutes).
MOD: Number of minutes per hour.
SQ(i): Value of number at position i of magic square.
X, Y, Z: Positions of missing numbers in magic square.
X$, Y$, Z$: Answer input variables.



```
DECATHLON—PART THREE

New lines to replace or add to
previous ones.


100  POKE  - 16360,0: VTAB 24: HTAB
     1: INVERSE : PRINT ">>> PRES
     S THE SPACE BAR TO CONTINUE
     <<<";: NORMAL
310 E$(1) = "MISSING #":E$(2) = "
     GUESS A #":E$(3) = "SPEED MA
     TH":E$(4) = "STARS":E$(5) =
     "L.C.D.":E$(6) = "MAGIC SQR"
     :E$(7) = "CLOCK MATH"
320 EE$(1) = " 1. FIND THE MISSIN
     G NUMBER":EE$(2) = " 2. GUES
     S A NUMBER":EE$(3) = " 3. SP
     EED MATH":EE$(4) = " 4. STAR
     RY STARRY NIGHT":EE$(5) = "
     5. LOWEST COMMON DENOMINATOR
     "
330 EE$(6) = " 6. MAGIC SQUARES":
     EE$(7) = " 7. CLOCK MATH"



Event #5:
Lowest Common Denominator



Print instructions.


5000 E = 5:NXT = 0
5010  FOR P = 1 TO NP: GOSUB 10
5020  PRINT "IN THIS EVENT I WILL
      SHOW YOU A SET OF  FRACTION
     S, AND YOU ARE TO FIND THE L
     OW- EST COMMON DENOMINATOR.
      THERE WILL BE  THREE SETS I
     N ALL."
5030  GOSUB 90
5040  IF NXT THEN 6000


Choose random denominators.


5050  FOR PROB = 1 TO 3
5060 A =  INT ( RND (1) * (4 + S(
     P) * 3)) + 3
5070 B =  INT ( RND (1) * (4 + S(
     P) * 3)) + 3: IF A = B THEN
     5070
5080  IF S(P) = 0 THEN 5130
5090 C =  INT ( RND (1) * 4) + 2:
      IF C = A OR C = B THEN 5090

Display fractions and compute
lowest common denominator.


5100 D = A * B * C: HOME : VTAB 1
     0: PRINT  TAB( 13)"1      1
        1": PRINT  TAB( 12)"---
       ---    ---": PRINT  TAB(
     13)A;: PRINT  TAB( 20)B;: PRINT
      TAB( 27)C:LCD = D


5110  FOR I = D / 2 TO 2 STEP  -
     1: IF  INT (I / A) = I / A AND
      INT (I / B) = I / B AND  INT
     (I / C) = I / C THEN LCD = I
5120  NEXT I: GOTO 5170
5130 D = A * B: HOME : VTAB 10: PRINT
      TAB( 16)"1      1": PRINT  TAB(
     15)"---    ---": PRINT  TAB(
     16)A;: PRINT  TAB( 23)B:LCD =
     D
5140  FOR I = D / 2 TO 2 STEP  -
     1: IF  INT (I / A) = I / A AND
      INT (I / B) = I / B THEN LC
     D = I
5160  NEXT I


Input answer and score it.


5170  VTAB 16: INPUT "   LOWEST C
     OMMON DENOMINATOR = ? ";AN$:
      PRINT :A2 = LCD:PX = 10: GOSUB
     30
5180  GOSUB 100: NEXT PROB,P: GOSUB
     11000



Event #6:
Magic Squares

Print instructions.


6000 E = 6:NXT = 0
6010  FOR P = 1 TO NP: GOSUB 10
6020  PRINT "IN THIS EVENT I WILL
      PRINT OUT A MAGIC  SQUARE W
     ITH ONE OR MORE NUMBERS MISS
     ING.YOU ARE TO SUPPLY THE MI
     SSING NUMBERS."
```

continued on next page

continued from previous page

```
6030  PRINT : PRINT "A MAGIC SQUA
      RE IS A SQUARE ARRANGEMENT
      OF NUMBERS IN WHICH EVERY CO
      LUMN, EVERY ROW, AND EACH LO
      NG DIAGONAL ADDS UP TO"
6040  PRINT "THE SAME TOTAL.  ONC
      E YOU DETERMINE THIS'MAGIC N
      UMBER' YOU CAN FIGURE OUT TH
      E  MISSING QUANTITIES."
6050  GOSUB 90: IF NXT THEN 7000
```

Generate random magic square.

```
6060  FOR PROB = 1 TO 3:A =  INT
      ( RND (1) * 7 * (S(P) + 1)) +
      1:B =  INT ( RND (1) * 4 * (
      S(P) + 1)) + 1
6070 C =  INT ( RND (1) * 3 * (SI
      LL(P) ^ 2 + 1)) + 1: IF C =
      B THEN 6070
6080 SQ(1) = A - B:SQ(2) = A + B -
      C:SQ(3) = A + C:SQ(4) = A +
      B + C:SQ(5) = A:SQ(6) = A -
      B - C:SQ(7) = A - C:SQ(8) =
      A - B + C:SQ(9) = A + B
6090 X = 0:Y = 0:Z = 0
6100 X =  INT ( RND (1) * 9) + 1:
       IF S(P) = 0 THEN 6140
6110 Y =  INT ( RND (1) * 9) + 1:
       IF Y = X THEN 6110
6120  IF S(P) = 1 THEN 6140
6130 Z =  INT ( RND (1) * 9) + 1:
       IF Z = Y OR Z = X THEN 6130
```

Display magic square.

```
6140  HOME :II = 0
6150  FOR I = 1 TO 9: VTAB 2 + 5 *
       INT ((I - 1) / 3):II = II +
       1: IF II = 4 THEN II = 1
6160  HTAB 6 + 7 * II - (SQ(I) <
      0) - (SQ(I) <  - 9) - (SQ(I)
       > 9): IF I <  > X AND I <  >
       Y AND I <  > Z THEN  PRINT S
      Q(I);: GOTO 6200
6170  HTAB 6 + 7 * II: IF I = X THEN
       PRINT "X"
6180  IF I = Y THEN  PRINT "Y"
6190  IF I = Z THEN  PRINT "Z"
6200  NEXT I
```

Input answer(s) and score them.

```
6210 X$ = "0":Y$ = "0":Z$ = "0": PRINT
      : VTAB 18: INPUT "VALUE OF X
       = ? ";X$: IF Y = 0 THEN 624
      0
6220  INPUT "VALUE OF Y = ? ";Y$:
       IF Z = 0 THEN 6240
6230  INPUT "VALUE OF Z = ? ";Z$
6240  IF SQ(X) =  VAL (X$) AND SQ
      (Y) =  VAL (Y$) AND SQ(Z) =
       VAL (Z$) THEN PX = 10: GOSUB
       50: GOTO 6280

6250  PRINT : PRINT "SORRY, ";PL$
      (P);":": PRINT "  X = ";SQ(X
      );: IF Y = 0 THEN 6280
6260  PRINT "   Y = ";SQ(Y);: IF
      Z = 0 THEN 6280
6270  PRINT "   Z = ";SQ(Z);
6280  GOSUB 100: NEXT PROB,P: GOSUB
      11000
```

Event #7:
Clock Math

Print instructions.

```
7000 E = 7:NXT = 0
7010  FOR P = 1 TO NP: GOSUB 10
7020  PRINT "IN THIS EVENT YOU AR
      E TO IMAGINE THAT  THERE AR
      E NO LONGER SIXTY MINUTES IN
       AN HOUR!  (THERE ARE STILL
      24 HOURS IN THE DAY, DIVIDED
       INTO 12-HOUR PERIODS.)"
7030  PRINT : PRINT "IN EACH OF Y
      OUR PROBLEMS, I WILL TELL
      YOU HOW MANY MINUTES THERE A
      RE IN AN    HOUR, AND WHAT T
      HE PRESENT TIME IS.    THEN
      , I'LL ASK YOU WHAT TIME IT
      WILL    BE A CERTAIN NUMBER
      OF MINUTES FROM NOW."

7040  PRINT "TYPE IN YOUR ANSWER
      WITH THE HOUR AND   MINUTES
      SEPARATED BY A COMMA--SUCH A
      S   2,15 OR 12,08."
7050  GOSUB 90
7060  IF NXT THEN 8000
```

Choose minutes per hour, current time, and elapsed time; calculate future time.

```
7070  FOR PROB = 1 TO 3:MOD = 50:
       IF S(P) = 1 THEN MOD = 10 *
       INT ( RND (1) * 5 + 4)
7080  IF S(P) = 2 THEN MOD = 5 *
       INT ( RND (1) * 20 + 4)
7090 H1 =  INT ( RND (1) * (8 + S
      (P) * 2) + 1):M1 =  INT ( RND
      (1) * MOD):M =  INT ( RND (1
      ) * MOD + MOD - M1) * (S(P) +
      1)
7100 H =  INT ((M1 + M) / MOD):H2
       = H1 + H:M2 = M1 + M - MOD *
      H
7110  IF M2 >  = MOD THEN M2 = M2
       - MOD:H2 = H2 + 1: GOTO 711
      0
7120  IF H2 > 12 THEN H2 = H2 - 1
      2
```

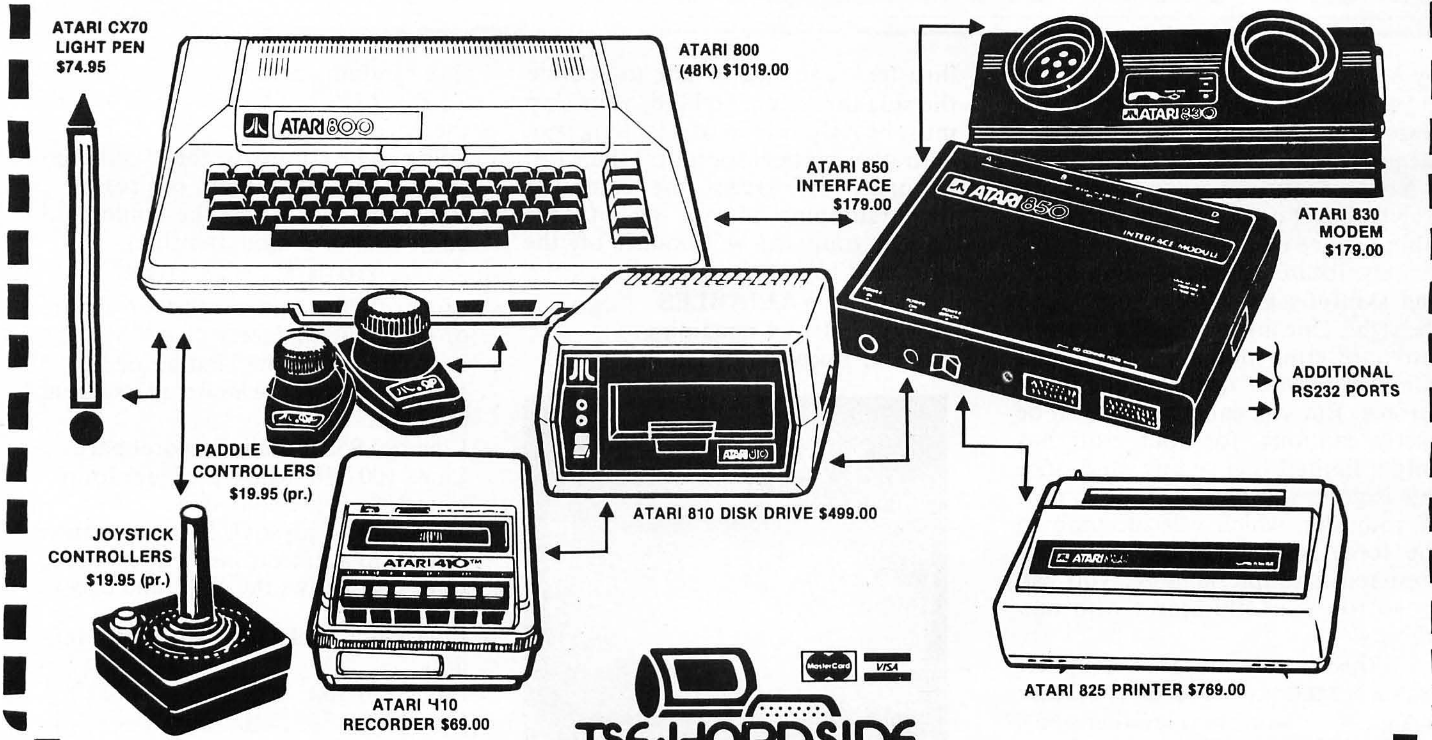
Print out problem, input answer, and score it.

```
7130  HOME : VTAB 5: PRINT "  IF
      THERE ARE ";MOD;" MINUTES IN
       AN HOUR,": PRINT : PRINT : PRINT
       TAB( 11)"AND THE TIME NOW I
      S": PRINT : PRINT  TAB( 18)H
      1;",";: IF M1 < 10 THEN  PRINT
      "0";
7140  PRINT M1: PRINT : PRINT : PRINT
      "  WHAT TIME WILL IT BE IN "
      ;M;" MINUTES?": PRINT : PRINT
      : PRINT "(TYPE HOUR, COMMA,
      MINUTES, 'RETURN'.)": POKE  -
      16368,0
7150  VTAB 20: HTAB 18: INPUT H$,
      M$:HH =  VAL (H$):MM =  VAL
      (M$)
7160  IF HH = H2 AND MM = M2 THEN
      PX = 10: GOSUB 50: GOTO 7190

7170  PRINT : PRINT "SORRY, ";PL$
      (P);", THE CORRECT": PRINT "
      ANSWER IS ";H2;",";: IF M2 <
      10 THEN  PRINT "0";
7180  PRINT M2
7190  GOSUB 100: NEXT PROB,P: GOSUB
      11000
```

Event #8 will begin at line 8000; for now, end here.

```
8000  END
```

The Atari* Connection.



TSE-HARDSIDE
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	$34.95	(#36-BASK)
Chess	$34.95	(#36-CHS)
Editor/Assembler	$49.95	(#36-ASE)
Music Composer	$49.95	(#36-MUSE)
Star Raiders	$49.95	(#36-STRDS)
Super Breakout	$34.95	(#36-SUPB)
Tele-Link	$24.95	(#36-TEL)
3D Tic-Tac-Toe	$34.95	(#36-3TTT)
Video Easel	$34.95	(#36-VIDEO

Software on Disk

VISICALC from Personal Software	$199.95	(#36-VICL)
MAILING LIST	$24.95	(#36-279002D)

Software on Cassette

Star Trek 3.5	$14.95	(#36-200025T)
Deflection/Simon Says	$9.95	(#36-200078T)
Mountain Shoot	$9.95	(#36-200079T)
Angle Worm/Crolon Diversion	$9.95	(#36-200092T)
Fleet	$6.95	(#36-27701BC)
Connect Four	$6.95	(#36-277020C)
Letters	$6.95	(#36-277022C)
Reverse	$6.95	(#36-277023C
Zap	$6.95	(#36-277024C)
Lander	$6.95	(#36-277019C)
Chase	$6.95	(#36-277017C)
Computer Acquire	$20.00	(#36-237002C)
Conflict 2500	$15.00	(#36-237001C)
Lords Of Karma	$20.00	(#36-237001C)
Biorhythms	$14.95	(#36-BIOR)
Blackjack	$14.95	(#36-BLACK)
Hangman	$14.95	(#36-HANG)
Kingdom	$14.95	(#36-KING)
3-Dimensional Graphics Package	$29.95	(#36-3D-G)

TERMS: Prices and specifications are suject 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.

* ATARI® A Warner Communications Company

LUNAR MISSION

by Matt Rutter

"Lunar Mission" is an Atari program requiring a joystick and 8K of memory.

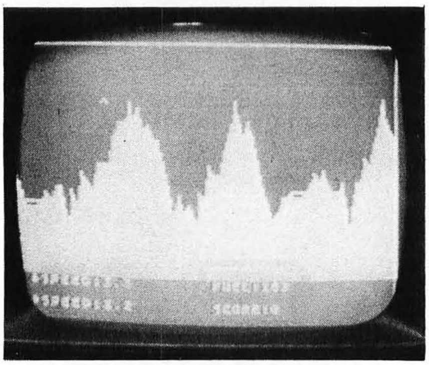
You are attempting to land your fragile rocket on the rough, unpredictable surface of the moon. You must quickly locate a safe landing spot, and skillfully guide your craft onto the sight. One mistake, and you crash into a towering mountain summit, or plummet to your death on the lunar surface. But you cannot afford to be overly cautious, for your craft has only a limited fuel supply, and once this is gone you have lost all control of your ship which will succumb to the force of gravity. After you've mastered a simple landing, you can try to top your old scores with new records.

In this simulation the computer draws a random scene of the lunar surface with two main landing pads, represented by dark lines beneath the surface. Your job is to land as many times as possible, thus earning points, before running out of fuel or crashing. The two main landing pads, however, are not the only places you can land. Landing on any flat surface at least three blocks long will earn you more points.

Push forward and back on the joystick to operate the main thrusters, and side to side to operate the side thrusters. To land, your ship must be right on top of a landing spot with your vertical speed less than 1.5. In order to prevent the computer from stopping if you go off the screen, your ship will bounce off the sides and top of the screen.

VARIABLES

F: Units of fuel remaining.
L1, L2: Location of landing pads.



P: Joystick value from STICK(0) command.
S: Points for last landing.
SC: Score.
V1, V2, V3: Temporary variables used in explosion sound routine.
X, Y: Position of ship.
XS, YS: Horizontal and vertical speed of ship.
Y1: Temporary variable used in plotting random surface.
Z, Z1, Z2: Used to hold the color of the three pixels where the lander is going to be. If any of these variables are not zero, than there is already something else where the lander will be (a wall or a mountainside).

DOCUMENTATION

Lines 20-70: Draw a random picture of the lunar surface.
Line 60: Draws the landing pads.
Line 80: Draws the lander at its initial position.
Lines 90-95: Initializes scoreboard.
Lines 100-180: Main program loop.
Line 100: Erases old ship.
Line 120: If joystick is moved, then branch to a subroutine to alter speed.
Line 130: Draws the ship, and checks for a crash.
Line 140: Checks for a successful landing.
Lines 150-160: Update scoreboard.
Lines 170-175: If the ship hits a wall then reverse the speed.
Lines 200-230: Subroutine to change the speed when joystick is moved.
Line 240: Out of fuel message.
Lines 300-340: Explosion routine.
Lines 350-380: Restart routine.
Lines 400-440: Successful landing routine. Adds points to the score and restarts the game. More points are earned by landing off the landing pad.

```
1 REM LUNAR MISSION
2 REM by Matt Rutter
3 REM written Jan,1981
10 F=200:SC=0
20 GRAPHICS 6:Y=INT(40*RND(1)+20):L1=I
NT(37*RND(1))*2:L2=INT(37*RND(1))*2+80
:COLOR 1:D=1
30 FOR X=0 TO 159 STEP 2:Y1=INT(15*RND
(1)+Y-5*D):PLOT X,80:DRAWTO X,Y:PLOT X
+1,80:DRAWTO X+1,(Y+Y1)/2
40 Y=Y1:IF Y>60 THEN Y=Y-10:D=2
45 IF Y<20 THEN Y=Y+10:D=1
50 IF X<>L1 AND X<>L2 THEN NEXT X:GOTO
 70
60 FOR I=X+2 TO X+7:PLOT I,80:DRAWTO I
,Y1:NEXT I:COLOR 0:PLOT I-5,Y1+1:DRAWT
O I-2,Y1+1:COLOR 1:X=X+6:NEXT X
70 PLOT 159,80:DRAWTO 159,0:DRAWTO 0,0
:DRAWTO 0,80
80 X=4:Y=3:PLOT 4,3:PLOT 3,4:PLOT 5,4:
XS=3:YS=0
90 POKE 752,1:POKE 656,0:? CHR$(27);CH
R$(29);"SPEED:":POKE 656,2:? CHR$(27);
CHR$(31);"SPEED:"
95 POKE 656,0:POKE 657,20:? "FUEL:":PO
KE 656,2:POKE 657,20:? "SCORE:"
100 COLOR 0:PLOT X,Y:PLOT X-1,Y+1:PLOT
 X+1,Y+1:X=X+XS:Y=Y+YS:YS=YS+0.1
110 LOCATE X,Y,Z:LOCATE X-1,Y+1,Z1:LOC
ATE X+1,Y+1,Z2
120 P=STICK(0):IF P<>15 AND F>0 THEN G
OSUB 200:GOTO 130
125 SOUND 0,0,0,0
130 COLOR 1:PLOT X,Y:PLOT X-1,Y+1:PLOT
 X+1,Y+1:IF Z OR Z1 OR Z2 THEN 300
140 LOCATE X-1,Y+2,Z:LOCATE X+1,Y+2,Z1
:IF Z AND Z1 AND YS<1.5 THEN 400
150 POKE 656,0:POKE 657,9:? YS;"   ";:
POKE 657,25:? F;" "
160 POKE 656,2:POKE 657,9:? XS;"   ";:
POKE 657,26:? SC;
170 IF X+XS<5 OR X+XS>156 THEN XS=-XS
175 IF Y<3 THEN YS=ABS(YS)
180 GOTO 100
200 IF P>8 AND P<12 THEN XS=XS-0.2:F=F
-1:SOUND 0,10,8,15
210 IF P>4 AND P<8 THEN XS=XS+0.2:F=F-
1:SOUND 0,10,8,15
220 IF P=6 OR P=10 OR P=14 THEN YS=YS-
0.2:F=F-1:SOUND 0,20,8,15
230 IF P=9 OR P=13 OR P=5 THEN YS=YS+0
.2:F=F-1:SOUND 0,20,8,15
240 IF F<1 THEN F=0:POKE 656,1:POKE 65
7,15:? "*OUT OF FUEL*":SOUND 0,50,10,1
5
250 RETURN
300 SOUND 0,0,0,0:POKE 656,1:POKE 657,
15:? "**YOU'RE DEAD**"
310 SOUND 2,75,8,15:V1=15:V2=15:V3=15
320 SOUND 0,20,8,V1:SOUND 1,40,8,V2:SO
UND 2,70,8,V3:FOR I=1 TO 10:NEXT I
330 V1=V1*0.89:V2=V2*0.94:V3=V3*0.97:I
F V3>1 THEN 320
340 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND
2,0,0,0
350 POKE 656,3:? "Press the fire butto
n to play again";
360 IF STRIG(0)=1 THEN 360
370 IF STRIG(0)=0 THEN 370
380 RUN
400 FOR I=10 TO 100 STEP 10:SOUND 0,I,
10,15:FOR W=1 TO 25:NEXT W:NEXT I:SOUN
D 0,0,0,0
410 POKE 656,1:POKE 657,2:? "The EAGLE
 has landed.  ";:S=100
420 IF (X>L1+2 AND X<L1+8) OR (X>L2+2
AND X<L2+8) THEN S=50
430 ? S;" points":SC=SC+S:POKE 656,2:P
OKE 657,26:? SC
440 FOR I=1 TO 2000:NEXT I:GOTO 20
```

DOMOD

by Rich Bouchard

"Domod" is a short utility program for S-80 Model III disk users.

The DO function in Model III TRSDOS is an extremely useful feature, especially when doing long repetitive tasks. In the short time that I have been using the Model III, the DO feature has become my favorite. It does have a problem, though. After using BUILD to create a DO file, there is no way to edit that file. So if you want to make a change, you are forced to type the entire file over again. This program is a short routine that will allow you to quickly and easily modify your DO files.

First, the program will ask you for the filespec of the DO file. Next, each line of the file will be displayed, with four options: Change, Delete, Insert and Save. [C]hange allows you to retype the current line. [D]elete will delete the current line. [I]nsert allows a line to be added before the current one. Only one line may be inserted at a time. If you wish to insert more than one line, type the first one, then [S]ave it, and then [I]nsert the second line, and so on. [S]ave tells the computer that the current line is acceptable as it is, and allows the computer to go read the next line.

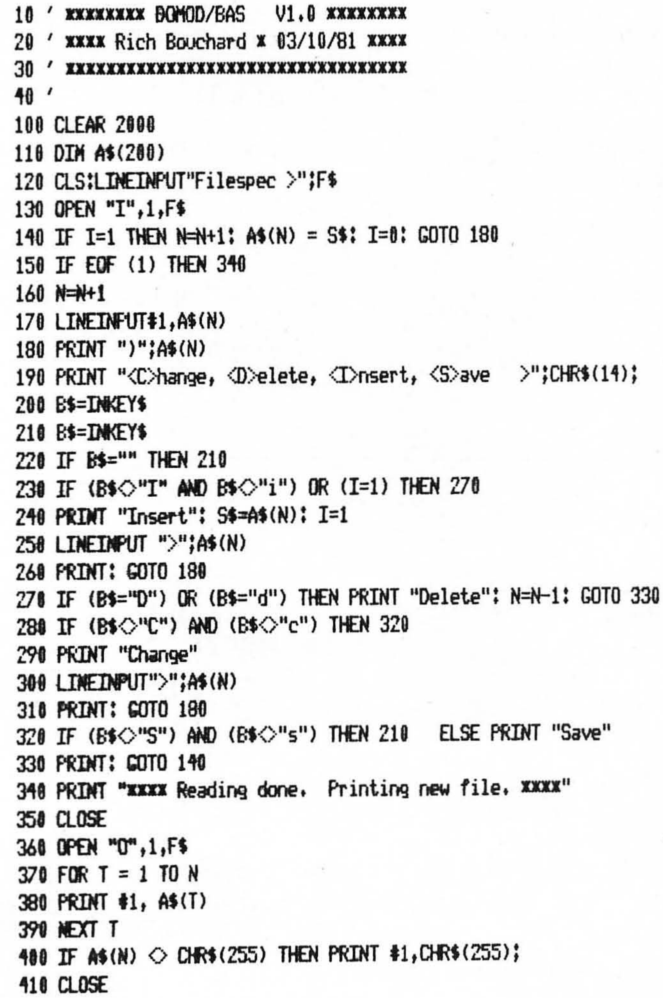
When the program has reached the end on the DO file, the updated version will then be saved under the same filespec, and will be ready for use through the DO command.

Some DO files may contain blank lines. This may look a little strange to have a blank line printed on the screen and to be given a chance to modify the line, but blank lines are sometimes useful.

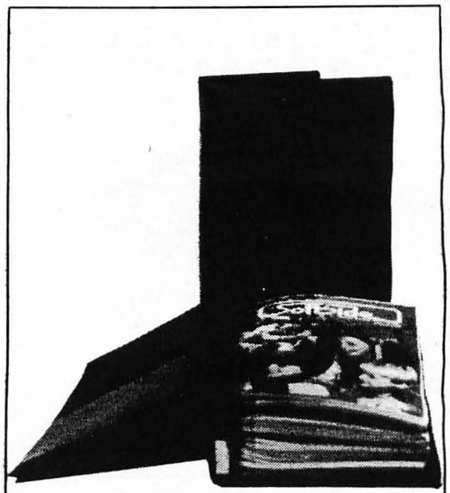
The preliminary Model III disk manual states that a DO file may only contain commands that are normally typed in response to TRSDOS READY. I have found this not to be the case, I have been able to create DO files that enter BASIC, set the NUMBER OF FILES and MEMORY SIZE, and then proceed to load, or even create, a BASIC program.

VARIABLES

A$( ): A string array containing each line in the DO file.
B$: Option selected, recieved via INKEY$.
F$: Filespec of DO file.
I: Flag. 1 if a line has recently been inserted. Used to prevent multiple inserts (line 230), and decide when an old line should be restored (see S$ below).
N: Number of lines currently in memory from the DO file.
S$: Storage for a line when a new line is to be added in front of it. Line 140 restores it after the inserted line has been either saved or deleted.



```
10 ' xxxxxxxx BOMOD/BAS   V1.0 xxxxxxxx
20 ' xxxx Rich Bouchard x 03/10/81 xxxx
30 ' xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
40 '
100 CLEAR 2000
110 DIM A$(200)
120 CLS:LINEINPUT"Filespec >";F$
130 OPEN "I",1,F$
140 IF I=1 THEN N=N+1: A$(N) = S$: I=0: GOTO 180
150 IF EOF (1) THEN 340
160 N=N+1
170 LINEINFUT#1,A$(N)
180 PRINT ")";A$(N)
190 PRINT "<C>hange, <D>elete, <I>nsert, <S>ave   >";CHR$(14);
200 B$=INKEY$
210 B$=INKEY$
220 IF B$="" THEN 210
230 IF (B$<>"I" AND B$<>"i") OR (I=1) THEN 270
240 PRINT "Insert": S$=A$(N): I=1
250 LINEINPUT ">";A$(N)
260 PRINT: GOTO 180
270 IF (B$="D") OR (B$="d") THEN PRINT "Delete": N=N-1: GOTO 330
280 IF (B$<>"C") AND (B$<>"c") THEN 320
290 PRINT "Change"
300 LINEINPUT">";A$(N)
310 PRINT: GOTO 180
320 IF (B$<>"S") AND (B$<>"s") THEN 210   ELSE PRINT "Save"
330 PRINT: GOTO 140
340 PRINT "xxxx Reading done.  Printing new file. xxxx"
350 CLOSE
360 OPEN "O",1,F$
370 FOR T = 1 TO N
380 PRINT #1, A$(T)
390 NEXT T
400 IF A$(N) <> CHR$(255) THEN PRINT #1,CHR$(255);
410 CLOSE
```



SOFTSIDE DIRECTORY

SoftSide is now over 2½ years old! During these years we have produced four different magazines, with the final result being all of them combined into the magazine you now hold in your hands. This directory contains the titles, authors, and the page number where each article or program appeared, up to and including last month's issue. Henceforth we will print an annual directory in the May issue. Look through our back pages and see what you may have missed out on, and what you might be interested in.

October 1978

SOFTSIDE: S-80 EDITION

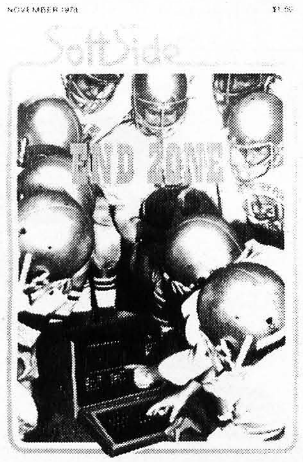
"Cribbage"
Roger W. Robitaille
page 8

"State Capital Quiz"
Roger W. Robitaille
page 28

'Death Star"
Roger W. Robitaille
page 39

"Calculator"
Roger W. Robitaille
page 39

"Pillbox"
Gene Perkins
page 41



November 1978

"Bad Code" (a puzzle)
Lance Micklus
page 14

"Troll's Gold"
George Blank
page 17

"Shopping List"
Roger W. Robitaille
page 27

"End Zone"
Roger W. Robitaille
page 32

"What They Never Told You About Level II"
George Blank
page 47

"Octal to Hex"
Lance Micklus
page 61

December 1978

"Santa Paravia En Fiumaccio"
George Blank
page 8

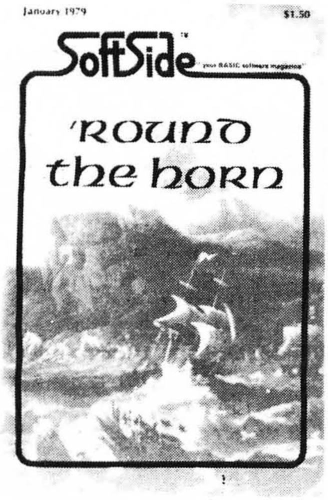
"Biorhythms"
Frank B. Rowlett
page 25

"Mortgage Calculations"
Russell Starkey
page 29

"Spelling Bee"
Lance Micklus
page 33

"Six Million Dollar (10K) Clock"
Peter Ashley
page 37

"Chess Clock"
Roger W. Robitaille
page 39



January 1979

"Round the Horn"
George Blank
page 8

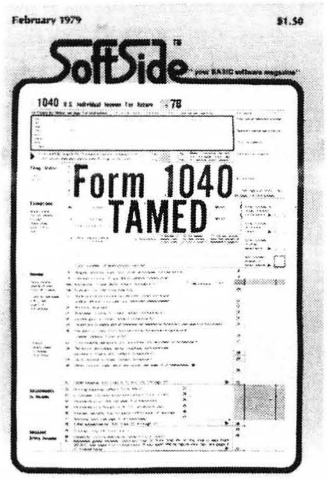
"Writing Good Computer Games
Part 1: Philosophy"
George Blank
page 29

"Kiddy Slot Game"
David Bohlke
page 35

"Ten Pin Bowling"
Frank B. Rowlett
page 40

"High Speed Graphics for the
TRS-80™ Computer"
Frank B. Rowlett
page 53

"Comput-A-Sketch"
Albert C. Blackwell
page 58


February 1979
$1.50
SoftSide
your BASIC software magazine
1040 U.S. Individual Income Tax Return 78
Form 1040
TAMED

February 1979

"Form 1040"
George Clisham
page 10

"Writing Good Computer Games
Part 2: Mechanics"
George Blank
page 27

"Concentration"
Lance Micklus
page 37

"Elements Quiz"
Roger W. Robitaille
page 47

"Read Error for Disk
Programmers"
Staff
page 53

"Cribbage Update" (corrections)
Roger W. Robitaille
page 55

March 1979

"Tarot"
Frank B. Rowlett
page 8

"Metric/English Converter"
Dr. Stephen W. Hebbler
page 17

"Can You See Through This?
(Fog index)"
George Blank
page 26

"Dive Bomb"
George Blank
page 30

"Personal Finance"
(checkbook and checkfinder)
Lance Micklus
page 32

"Jig Saw"
David Bohlke
page 46

continued on next page

continued from previous page

April 1979

"Checkfinder"
Lance Micklus
page 8

"Don't It Make My Brown Eyes Blue?"
James Garon
page 17

"Rabbits and Foxes"
Russell Starkey
page 24

"Spring Flowers"
David Eide
page 31

"Safari"
David Bohlke
page 36

"Series Circuits"
Phillip Brown
page 54

May 1979

"Dog Star Adventure"
Lance Micklus
page 8

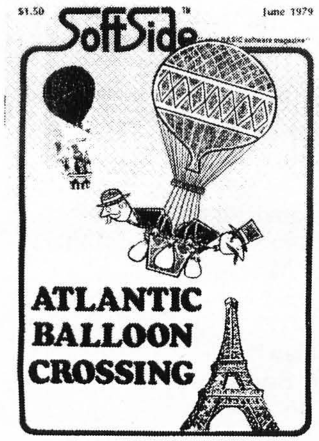
"Awari"
Scott Adams
page 24

"Take A Part Awari"
Scott Adams
page 29

"Letter Crunch"
David Bohlke
page 30

"Math Drill"
D.L. Brown
page 36

"Super Sub"
Michael Winter
page 48



June 1979

"Amazin' Maze"
Michael Kelleher
page 6

"Atlantic Balloon Crossing"
Dean Powell
page 8

"German Word Quiz"
Computer Graphics
page 27

"Nim"
James Garon
page 36

"Home Appliance Record System"
Ray Herald
page 43

"Entrapment"
Robert C. Hall
page 53

July 1979

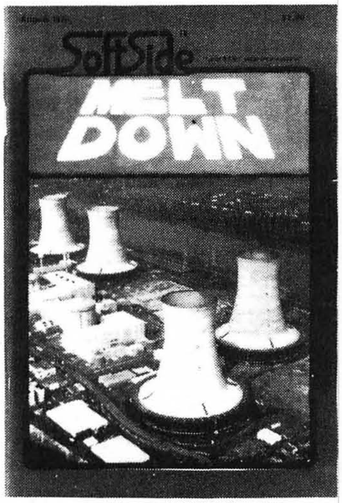
"All Star Baseball"
David Bohlke
page 10

"Yahtzee"
Al Lowe
page 25

"Tank"
James Garon
page 40

"Spelling Bee"
James Garon
page 52

"Collision"
David Bohlke
page 64



August 1979

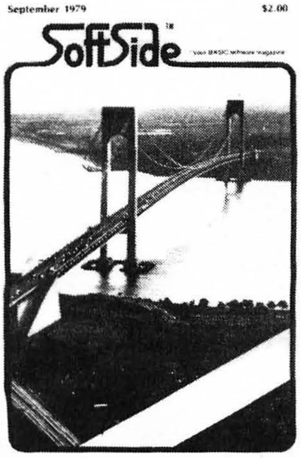
"Meltdown"
Michael Winter
page 8

"Shooting Gallery"
Frank B. Rowlett
page 26

"Storybook"
Otto Hansen
page 36

"Morse Code Tutor"
Donald E. Smith
page 52

"Cryptogram"
Rik Pierce
page 57



September 1979

"Treasure Dungeon" (Tutorial)
David E. White
page 8

"Hangman"
Russell Starkey
page 26

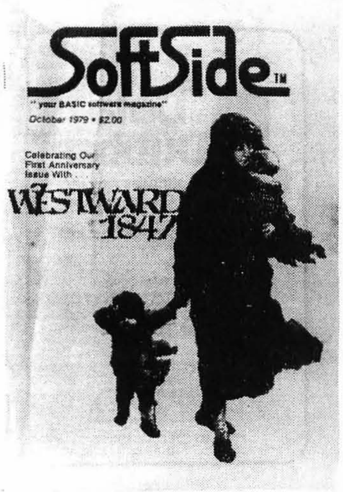
"Pyramids"
Lewis E. Garrison
page 36

"Immediate Mode"
(Programming Hint)
Dean C. Westervelt
page 54

"Engineer"
David Bohlke
page 58

continued on next page

continued from previous page


SoftSide
"your BASIC software magazine"
October 1979 • $2.00
Celebrating Our First Anniversary Issue With...
WESTWARD 1847

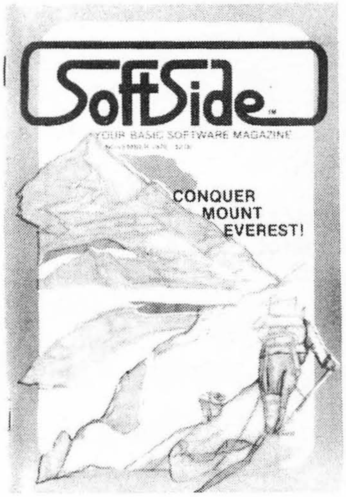
October 1979

"What Kind of Work Do You Enjoy?"
G.P. Cocanour
page 10

"Reaction Time"
Chris Freund
page 20

"Battleship"
John M. Delaney
page 28

"Westward 1847"
Jon C. Sherman
page 36



November 1979

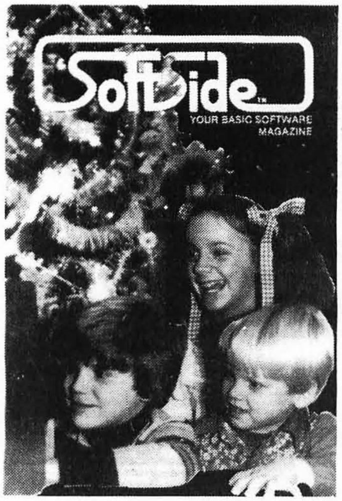
"Energy Cost Comparison Program"
George Blank
page 12

"Isolate"
James Garon
page 18

"Picture Puzzles"
Arnold Gross
page 30

"Everest"
Byron A. Myhre
page 36

"Doodle Pad"
R.B. Nottingham
page 60



December 1979

"Oil Baron"
Paul Knechtel
page 8

"Drag Race"
Chris Freund
page 26

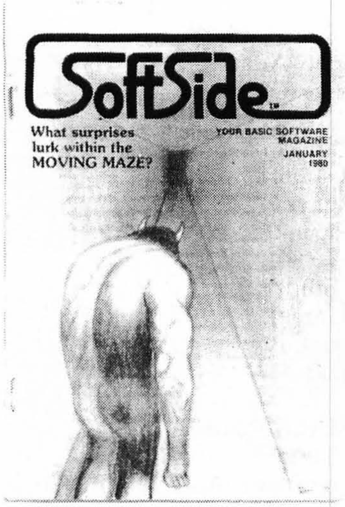
"Christmas Show and Tell"
Randy Hawkins
page 33

"Index to BASIC Reference Manual"
Kemp J. Beaty
page 39

"Towers of Hanoi"
Chet Kacneski
page 44

"Slalom"
Denslo Hamlin
page 50

"Review of Bridge Challenger"
Wynne Keller
page 63


SoftSide
What surprises lurk within the MOVING MAZE?
YOUR BASIC SOFTWARE MAGAZINE
JANUARY 1980

January 1980

"Dante's Inferno"
Gerald Bernor
page 36

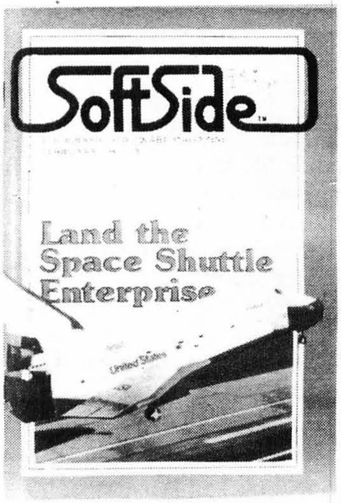
"Basic Pitfalls" (article)
William Mason
page 52

"Duck Hunter"
Don Lubarsky
page 58

"Moving Maze"
Paul Klinger
page 10

"Bomber"
Chris Freund
page 18

"Reflection"
(deductive reasoning game)
Ted Lindstrom
page 23



February 1980

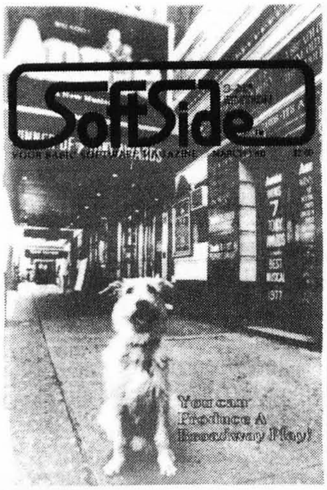
"Scott Adams' Backgammon"
Scott Adams
page 10

"Parachute"
Norman McAllister
page 27

"Deadstick"
Al Ragsdale
page 35

"Play it Again Sam"
Alan Zett
page 54

"Deep Six"
Chris Freund
page 70


SoftSide
YOUR BASIC SOFTWARE MAGAZINE
You can Produce A Broadway Play!

March 1980

"Add" (Children's Program)
Terry Hazelett
page 12

"Line Four" (Tic Tac Toe Strategy)
Robert Speaks
page 18

"System Emulator"
Roy Niederhoffer
page 28

"Broadway"
Robert Saturn
page 40

"A Game Designer's View"
(What makes a game appealing)
Ian Chadwick
page 58

"Sonic Torpedoes"
Phillip Case
page 69

continued on next page

continued from previous page

April 1980

"Solitaire"
James Garon
page 12

"Hope it Doesn't Work"
(Article)
Ron Potkin
page 18

"ICBM"
(Game)
Rod Fitzgerald
page 24

"Programming Made Easy in 580 Bytes"
(Review of T-short)
Sherry M. Taylor
page 34

"Review of Word Challenge"
Scott Adams
page 35

"Slot Machine"
(Game)
James A. Hagani
page 40

"Stud Poker"
(Game)
Dave Bohlke
page 48

"Draw Poker"
(Game)
Lee Blumenthal and Joel Willard
page 55

May 1980

"Livekeys"
(World's smallest word processor)
Douglas Werbeck
page 12

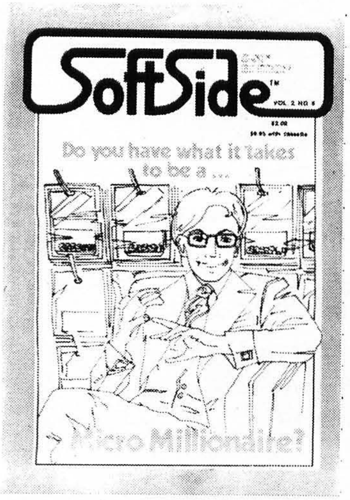
"The $44000 Star Trek program"
Lance Micklus
page 18

"Review of Tycoon"
Brad Cameron
page 37

"Star Trek III.4"
(Game)
Lance Micklus
page 40

"Day of the Week Program"
George Blank
page 69

"Encounter in the Near Tholian Sector"
Barry L. Adams
page 72



June 1980

"Printing Calculator"
D.L. Morrow
page 12

"Numbagels"
(Random number game)
Peter Kirsch
page 20

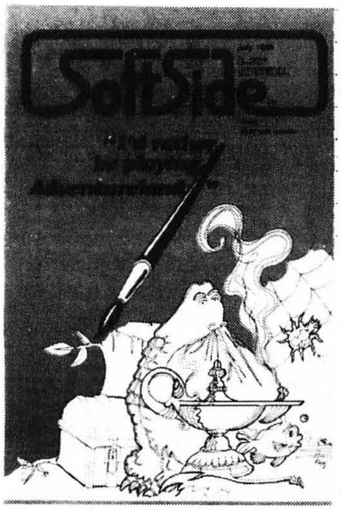
"Fix Keyboard Bounce"
Al and Gary Lowe
page 30

"Simple Sound Interface"
Edward Ting
page 36

"Micro Millionaire"
(Program/article)
George Blank
page 40

"Galaxy Combat"
(Game)
Phil Case
page 56

"Adventures"
(Article on compunovels)
James Garon
page 17



July 1980

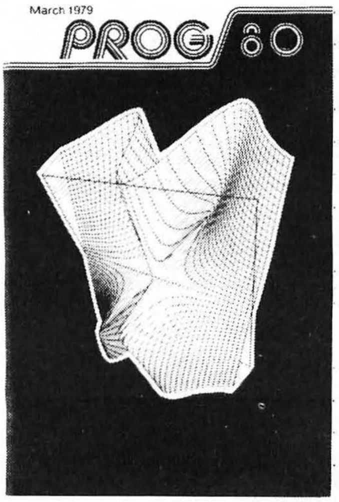
"Micro-80 Pinball Machine"
(Game)
Garth Jensen
page 22

"Adventureland"
(Data tape for Adventureland in BASIC)
Scott Adams
page 36

"Adventure Interpreter"
Scott Adams
page 44

"Lost Dutchman's Gold"
(Game)
Teri Li
page 58

PROG 80



March 1979

"Strings Fever"
(Level II string variables)
Robert A. Wallace
page 6

"Upper Case/Lower Case Modification"
(S-80 routine)
Lance Micklus
page 9

"Variables in Level II S-80 Programs"
(article)
John A. Millspaugh Jr.
page 12

"Getting the Most Out of the Data Statements"
(article)
George Blank
page 17

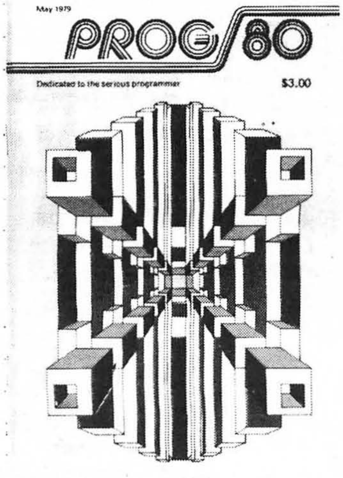
"Simple Simon"
(Experiment with Machine Language and Disassemble your BASIC ROM)
George Blank
page 24

"Inkey Routines" (Level II Basic)
Lance Micklus
page 37

"Disk Drive Clunk?"
(Programming routine)
Lance Micklus
page 39

"Clock Routines"
(Disk BASIC routines to manipulate the clock)
Lance Micklus
page 41

"Super Graphics"
(Graphics on the S-80)
George Blank
page 42



May 1979

"Clock Routines"
(Disk BASIC program to manipulate the clock)
Lance Micklus
page 6

"Rescue"
(Restore crashed Disk BASIC program)
George Blank
page 10

"Scattergram/Correlation Program"
(S-80 Level II Basic)
Gary S. Breschini
page 16

"A Piece of the ROM"
(Three routines get data from the keyboard)
Lance Micklus
page 27

"Super Graphics Without Disk"
(Basic program acts as graphics editor)
James Garon
page 29

continued on next page

continued from previous page

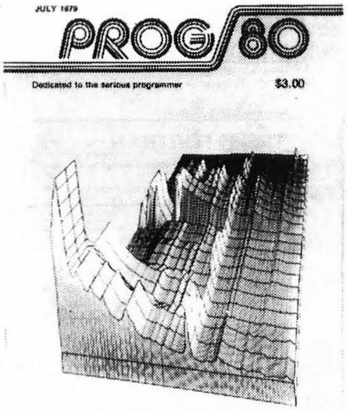
"Boomer Box"
(Level II BASIC sound box)
Frank B. Rowlett, Jr. and
Garry W. Woodruff
page 36

"Upper/Lower Case Mod"
(Schematic and BASIC subroutine in high memory)
Lance Micklus
page 48

"Squish/Bas"
(S-80 program that compresses BASIC programs)
Bill Driscoll
page 62

"Init"
(Initialize program for the S-80)
George Meyer
page 66

"Graphics Aid"
(Translate graphics on screen into data that can be output)
Phillip Brown
page 51



July 1979

"A Sound Subroutine"
James Garon
page 6

"Machine Language Video Output"
Paul Johnson
page 14

"High Density Data Storage"
(Bit packing and unpacking)
George Blank
page 20

"Histogram and BASIC statistics program"
(Level II S-80)
Gary S. Breshini
page 25

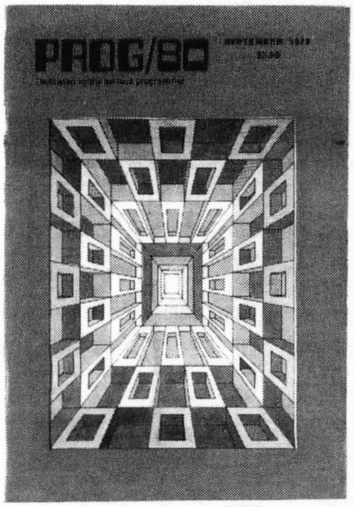
"Talking Banko" (Game)
Lance Micklus
page 38

"Plot" (Machine Language version of set (XY) for the S-80)
James E. Randall
page 52

"Cassette Controller"
(Constructing a control box)
John D. Eaton
page 57

"Book Review of S-80 Assembly Language Programming"
George Blank
page 70

"S-80 Music"
(Harness your computer and radio)
James Garon
page 72



September 1979

"DOS"
(How to stop lost data errors)
Lance Micklus
page 12

"Computers are Kid Stuff"
(How to use computers with children)
Michael Potts
page 16

"Basic Disk Cross Reference Utility"
L.C. Chesbro
page 30

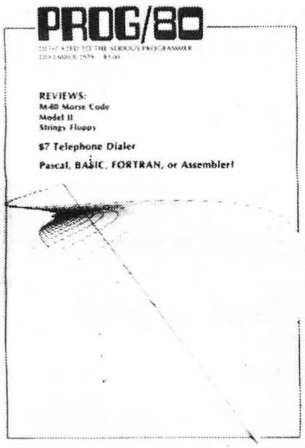
"Renumbering Program for TRSDOS"
Rocky Smolin
page 34

"Squish-2"
Bill Everett
page 41

"Old Business"
(Another modification to upper/lower case modification)
Lance Micklus
page 46

"String Crasher for 32K S-80"
Clayton E. Schneider
page 50

"Date and Time Routine for TRS DOS 2.2"
George W. Blank
page 52



December 1979

"Computer Telephone Dialing"
(Allows S-80 to dial numbers from in-memory directory)
Robert Sours and Steve McKee
page 10

"Choosing the Right Language"
Lance Micklus
page 26

"Devices"
(Use of command for TRSDOS)
Lance Micklus
page 34

"An Alternative to a Disk System"
(Discussion of Exatron stringy floppies)
Wynne Keller
page 46

"Variable Uniqueness and Validity"
(Check intended variable names through name-checker)
Jeff W. Collins
page 50

"Shedding Light on a Dim Area"
(Routine to increase speed)
James Garon
page 54

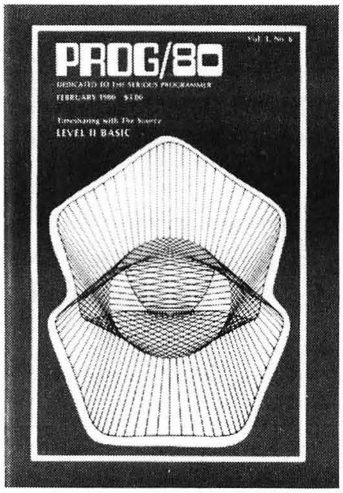
"Preference Poll"
(S-80 program)
George Blank
page 58

"Macrotronics M-80 Review"
Donald W. Dejarnette
page 65

"Hexadecimal to Decimal Converter"
Bill Everett
page 70

"Model II — The First Look"
(Review)
Lance Micklus
page 72

"Review — Blaiseing along in Pascal"
(People's Pascal I and II)
Robert P. Johnson
page 74



February 1980

"May the Source Be With You or Will it be Micronet?"
(Review)
Richard Taylor and Mark Kirmayer
page 10

"The Time Sharer's Paradise — ST80D + Electric pencil + VTOS 3.0"
Richard Taylor
page 13

"Are Your Programs Write Protected?"
(Procedure with just 2 lines)
Frederick B. Hagar
page 21

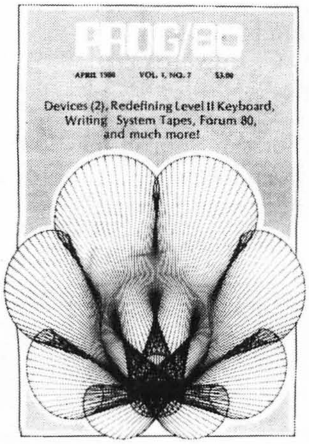
"What's In a Name"
(Give a name a meaning)
James Garon
page 24

"Superzap Review"
Teri Li
page 28

"Hexmem"
(Basic program that duplicates some Machine Language monitors)
John T. Phillip
page 41

"High Speed String Handling in BASIC"
(Discussion and subroutines)
Marty Zwilling
page 52

"CTR-41 Modifications"
Bob Cahill
page 62


PROG/80
APRIL 1980 VOL. 1, NO. 7 $3.00
Devices (2), Redefining Level II Keyboard, Writing System Tapes, Forum 80, and much more!

April 1980

"Redefining Level II Keyboard"
(Intercept the S-80 input and output routines to change value of keys)
Phelps Gates
page 10

"Writing System Tapes"
(Article and program)
George Blank
page 18

"Voice Synthesizer Laboratory"
(Program displays phonemes on screen)
John R. Olsen, Jr.
page 26

"BASIC File Utility"
(Disk program)
Clayton Schneider
page 36

"Devices"
(Article)
Lance Micklus
page 46

"A New User's Guide to the Source"
Richard Taylor
page 56

"Forum 80"
(Review of a computer bulletin board service)
Richard Taylor
page 65

"Stacking Your Subroutines on the Heartbeat Interrupt"
(Article/routine)
George Blank and John Hartford
page 70

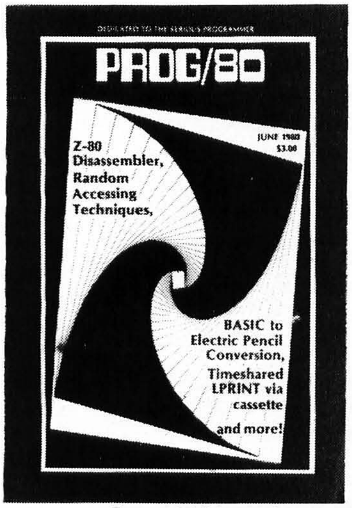
continued on next page

continued from previous page

"Repeatable Randomness"
(Routine can repeat a sequence of random numbers)
James Garon
page 72

"Zero Slash Killer"
(Assembly Language routine to eliminate the zero slash in line printer outputs)
Timothy Smith
page 74

"Proofread"
Frank S. Kasper
page 76


DEDICATED TO THE SERIOUS PROGRAMMER
PROG/80
JUNE 1980
$3.00
Z-80 Disassembler, Random Accessing Techniques,
BASIC to Electric Pencil Conversion, Timeshared LPRINT via cassette
and more!

June 1980

"All Purpose Print Routine"
C.E. Aidlaw
page 69

"Z-80 Disassembler"
(Utility program)
George Blank
page 10

"Printer Terminal"
(Use time-sharing terminal to print hard copy with your S-80)
Dr. Owen Gaede
page 19

"Random Accessing Techniques"
(Disk file structures and programming aids)
Will Hagenbuch
page 56

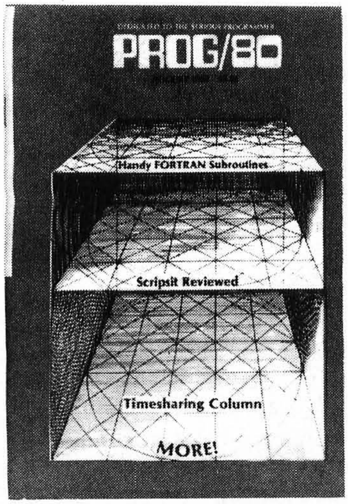
"Form Letter"
(Letters from Electric Pencil to be used in conjunction with mailing lists)
Richard Taylor
page 49

"BASIC to Electric Pencil Conversion "
James Garon
page 40

"New NEWDOS command — LOC(X)"
(Explanation and check routine for Model II)
Carl William Mackey
page 76

"Branching Functions in APL80"
(Article)
Phelps Gates
page 44

"Using Electric Pencil with Radio Shack Lower Case Modification"
Jeff Brown
page 82


DEDICATED TO THE SERIOUS PROGRAMMER
PROG/80
Handy FORTRAN Subroutines
Scripsit Reviewed
Timesharing Column
MORE!

August 1980

"A Second Look at Micronet and Source is Now MNET-80 Too"
Richard Taylor
page 14

"Program Portability"
(Makes programs portable between Level II and Disk BASIC)
James Garon
page 22

"Splat — An Introduction to Tiny Comp"
David Bohlke
page 26

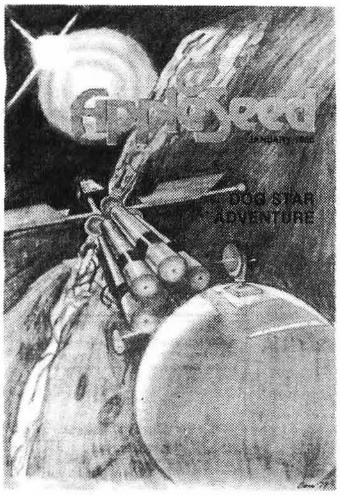
"Scripsit Review"
Richard Taylor and Mark Kirmayer
page 33

"S-80 Program Storage and Some useful Corollaries"
(Article/charts)
Jeff Meyers
page 38

"Varlist"
(Program to list variables)
Charles C. Edwards
page 46

"S-80 Fortran Subroutines"
David E. White
page 54

SOFTSIDE: APPLE EDITION



January 1980

"Troll's Gold"
George Blank
(Conversion by Murray Summers)
page 28

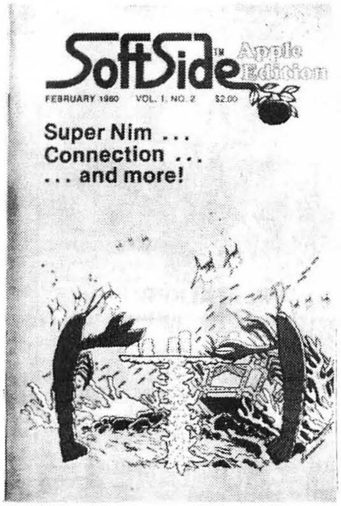
"Dog Star Adventure"
Lance Micklus
page 35

"Editing Features"
(Programming Hints)
staff
page 53

"Loan Amortization"
Richard T. Rodgers
page 12

"Reverse" (Skill game)"
Austin R. Brown
page 20

"Giant Clock"
(Digital computer clock)
Mike Weinstein
page 24



Feburary 1980

"SuperNim"
Dr. Jerry Dubnoff
page 10

"Elementary Math"
staff
page 22

"State Capitals"
Jack Brandon
page 26

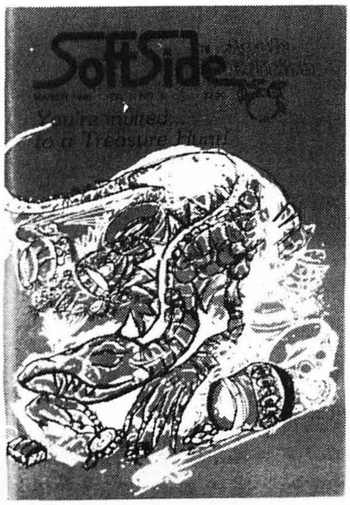
"Connection"
(Game with Lo-Res graphics)
Chip Anderson
page 32

"Musical Scales"
(Educational program)
Roger Wagner
page 39

"Sort"
(Utility to help reorganize words or strings)
Michael Weinstock
page 46

"Random Thoughts on Loading"
Staff
page 20

"More Programming Hints on Editing"
Staff
page 54



March 1980

"The Care and Feeding of Integer Hi-Res"
Gale Porter
page 9

"Renumber and Merge the Easy Way"
Frank D. Chipchase
page 19

"Format Decimals"
(A subroutine to format numbers)
Staff
page 21

"Acey Ducey" (Game)
Mike Weinstock
page 27

"Treasure Hunt" (Game)
Lance Micklus
page 32

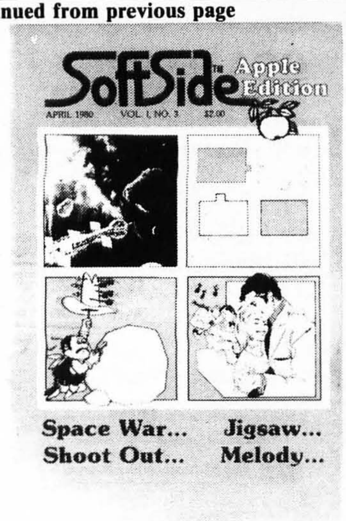
"Bouncing Ball Catcher" (Game)
Mark Cross
page 46

"Switch Puzzle" (Game)
Staff
page 50

"Programming Hint on INT Function"
Staff
page 15

continued on next page

continued from previous page



April 1980

"Applesoft Ampersand"
D. Pachin
page 16

"Shoot Out" (Game)
Alan Hill
page 19

"Jig-Saw"
Bob Sander-Cederlof
page 28

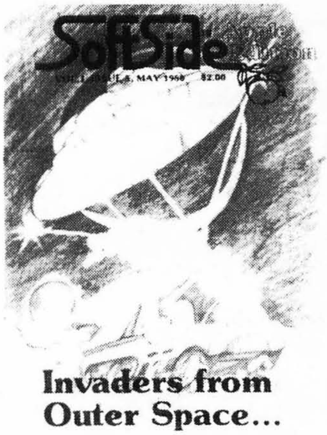
"The Invisible Signature"
(Utility routine)
Craig Crossman
page 32

"Space War" (Game)
Bob Sander-Cederlof
page 35

"Melody"
(Converts your Apple to a piano)
Mitch Kapur
page 40

"Book excerpt — Inimate Instructions in Integer BASIC"
Brian and George Blackwood
page 49

"Display Control Characters"
(Control characters in flashing video)
page 52



May 1980

"Small Marquee" (Game)
Mark J. Wolfson
page 10

"Hyperboloid"
(3-D Hi-Res graphics)
John Mauch
page 16

"Magic Cave" (Game)
Chuck Hartley
page 18

"Fifteen Game"
Mark Cross
page 24

"Review of Android Nim"
Brad Cameron
page 28

"The Utimate Program Protection"
(Programming hint)
Phillip Case
page 29

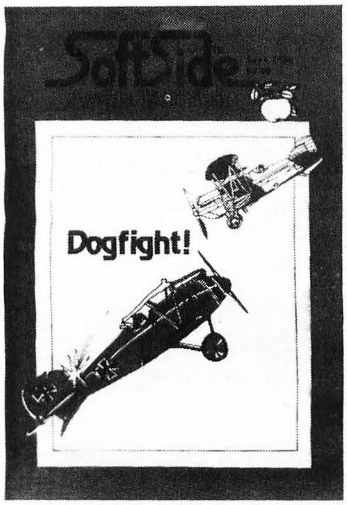
"Black Box" (Game)
Edward Garner
page 33

"Invaders" (Game)
Mark Pelczarski
page 38

"Right/Left"
(Program for toddlers)
Carl Swenson
page 50

"Book excerpt — Intimate Instructions in Integer BASIC"
Brian and George Blackwood
page 53

"Catalog II"
(Boot or hello program)
Lennis Hall
page 56



June 1980

"Apple Graphics" (article)
Joan Truckenbrod
page 12

"ROM the Robot"
Bill Smith
page 18

"Dodger" (Game)
Richard Johnson
page 18

"A Hello Program for Your Disk System"
Mark Pelczarski
page 33

"Dogfight" (Game)
Mark Pelczarski
page 36

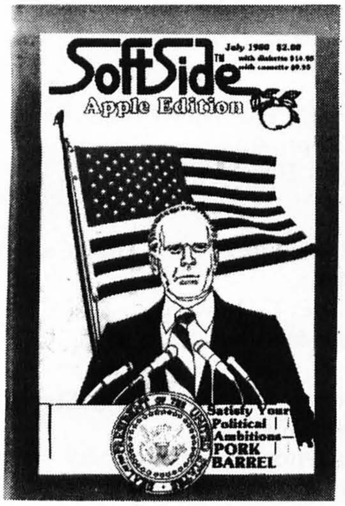
"Magic Paintbrush"
(Super Etch-A-Sketch program)
Mark Pelczarski
page 44

"Book excerpt — Intimate Instructions in Integer BASIC"
(Print statements)
Brian and George Blackwood
page 49

"Reviews of Three Mile Island, Compu-Read, Windfall, and Visicalc"
Staff
page 55

"Double Rainbow"
(Increase number of colors in Hi-Res)
Don Jacobs
page 63

"Programming Hint on How to Add Sound to Integer programs"
Mark Pelczarski
page 65



July 1980

"ROM the Robot — Part 2"
(How to move your robot)
Bill Smith
page 12

"No-Mis" (Repeat the tune game)
Robert Hausman Jr.
page 22

"It's a Small World"
(Subroutine to produce tones through speaker)
Timothy and Ping Patrick
page 28

"Lo-Res Printout"
(Routine to dump the Lo-Res screen onto printer)
Scott Nevin
page 34

"Pork Barrel" (Game)
George Blank
(Translated by Murray Summers)
page 40

"Apple III Review"
Mark Pelczarski
page 66

"Horse Race" (Game)
Steven Kahn
page 75

"Reviews of Global War, Sargon II, and Maze game"
Mark Pelczarski
page 81

"Apple Graphics"
(Simulated motion article)
Joan Truckenbrod
page 85

NEW SOFTSIDE

August 1980

"Fiddling with VARPTR"
(S-80 programming concepts)
James Garon
page 8

"Four-Oh-One-Six"
(The secret to load and go system tapes for S-80 Level II users)
Robert W. Lord and Rajeev Jayavant
page 14

"How Much Computer Do You Need?"
(Article)
George Blank
page 15

"Converting Graphics From One Computer to Another"
(S-80, Apple, and Atari)
Staff
page 22

"Mod-Prog"
(S-80 and Apple routines that allow easy program changes)
Thomas Hanlin III
page 24

"Sailplane"
David J.T. Nunn
page 26

"Caribbean Cruising"
(Apple)
Mark Pelczarski and Jim Klink
page 30

"Pro Tour '80"
(S-80 golf program)
Dave Bohlke
page 34

"Master's Golf"
(Atari)
David Bohlke
page 38

"ROM the Robot III"
Bill Smith
page 42

"Maze Search"
(Atari)
Dave Bohlke
page 48

"Grand Prix"
(S-80 Formula 1 car race)
Max Chauvet
page 50

"Concentration"
(Apple)
Harris Kirk
page 52

"Stratoblaster Outpost"
(Atari)
Dave Bohlke
page 54

"ATARI Keystroke Karnival — Poor man's joystick"
(Program/explanation)
James Garon
page 56

"Save Your Crashed Scripsit"
George Blank
page 64

"You Can Have Sound on Your Computer"
(Apple, Atari, and S-80)
Mark Cross and Rich Bouchard
page 66

continued on next page

continued from previous page

September 1980

"Apple Data Base Reviews"
(What to look for when you buy)
Mark Pelczarski
page 3

"Saving Time and Fingertips"
Shane Causer
page 8

"Computers in Education — Have They Made a Difference?"
Mark Pelczarski
page 13

"What Do You Do With Your Computer? — Is the Expense Justifiable?"
George Blank
page 13

"Goal"
(Apple game of logic)
Fred Pence
page 14

"Computer Graphics"
(How to scale objects)
Joan Truckenbrod
page 15

"Wandering Along Pathways Through the ROM"
Robert F. Nicholas
page 16

"Stereo Generator"
(Apple)
Dandelion Micro Products
page 18

"Developing Data Base II"
(Change and delete are added for S-80, Apple, and Atari)
Mark Pelczarski, James Garon, and Rich Bouchard
page 22

"Concentration II"
(S-80)
Randy Hawkins
page 22

"Super Barricade"
(Atari)
Paul Johnson
page 24

"Dr. Livingston/in search of"
(S-80)
Carl and Karen Russell
Ralph and Becky Fullerton
page 26

"Foosball"
(Apple)
Douglas Johnson
page 30

"Lone Star Corral"
(S-80 game)
Robert C. Hall III
page 34

"Adventurer's Guidebook"
(Shorthand, maps, and mazes)
James Garon
page 36

"Sound For Your Computer"
(S-80 program)
James Hagani
page 41

"Unlimited Level II User Calls"
(S-80)
Charles C. Edwards
page 44

"Tone Generator for the Stereo Generator"
Dandelion Micro Products
page 45

"HPLOT and DRAWTO for the S-80"
James Garon
page 45

"Sleuth"
(Atari word game)
Paul Johnson
page 48

"Ricochet"
(Atari)
Paul Johnson
page 48

"Developing Data Base I"
(Putting in information, saving it, and recalling it)
Mark Pelczarski, James Garon, and Rich Bouchard
page 50

"Video Easel Review"
James Garon
page 52

October 1980

"Garon's Goodies"
(S-80 programming concepts for space compression codes)
James Garon
page 25

"Sound For Your Computer"
(S-80)
James Hagani
page 32

"Computer Graphics"
(3-D for the Apple)
Joan Truckenbrod
page 66

"The Hidden Colors in Graphics 8"
(Atari)
James Garon
page 69

"Checker Challenge"
(Solitary puzzle for the Atari)
James Garon
page 69

"Dr. Livingston explained"
(Documentation)
Carl and Karen Russell
Ralph and Becky Fullerton
page 76

"Inhotep"
(S-80)
Terry Clark
page 18

"World Series"
(Atari)
David Bohlke
page 26

"Escape"
(S-80 Maze with a view)
Joel Mick
page 34

"Monster Maze"
(S-80)
Joel Mick
page 36

"Interna-Maze"
(Apple)
Dennis Ward
page 43

"Moonlanding"
(Apple)
Herb Sandy
page 44

"Earth-Port II"
(S-80)
Phil Case
page 46

"Prepack"
(S-80 utility program)
John R. Olsen
page 48

"Datapoke"
(S-80)
William K. Mason
page 49

"Strings and Things"
(S-80 and Atari graphics)
Mark A. Ohlund
page 60

"Chainer"
Frank R. Neal
page 62

"Mumath Review"
James Garon
page 81

"Double Cannon"
(Atari shootdown for two)
Dave Bohlke
page 50

KRIEGSPIEL
ENGINEER
BOWLING
METEOR STORM
MAD SCIENTIST ADVENTURE
PLUS!
Data Base Sorting Routines,
Graphics Rotations, Game Design,
Programming Hints and more!

November 1980

"Card Draw"
(S-80 computer card shuffler)
Arthur Glecker
page 19

"Developing Data Base III"
(Sorting routine for S-80, Apple, and Atari)
Mark Pelczarski, James Garon, and Rich Bouchard
page 30

"Z-80 Softcard and DOS 3.3"
(Apple reviews)
Mark Pelczarski
page 42

"Computer Aided Drawing and Design"
(Rotation techniques)
Joan Truckenbrod
page 74

"Copycat +"
(S-80 live key input routine)
A. Douglas Werbeck
page 87

"Meteor Storm"
(S-80)
Roy Harper
page 23

"Kriegspeil"
(S-80 wargame for two)
Ron Potkin
page 34

"Collision"
(Apple)
Mark Pelczarski
page 44

"Bowling"
(Apple)
Phil Case
page 47

"Engineer"
(Atari)
James Hagani
page 50

"Mad Scientest"
(S-80 Adventure)
Thomas Hanlin III
page 52

"Trench"
(Atari program to destroy the Death Star)
Tim Hays
page 66

"Boing"
(S-80)
Ron Webster
page 82

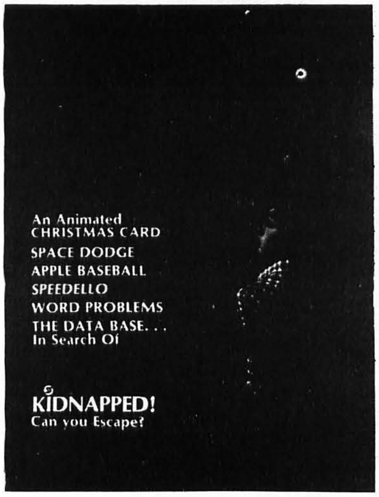
"Cards"
John T. Phillips
page 57

"Keyboard Organ"
(Apple)
Rob Hausman
page 62

"Pocket Tape Dubber"
(S-80 device to make to duplicate cassette tapes)
Edward Ting
page 85

continued on next page

continued from previous page



December 1980

"Say Yoho"
(How to put Adventures on stringy floppy for the S-80)
Scott Adams
page 23

"What to Do While the Program Loads"
Sherry M. Taylor
page 28

"Developing Data Base IV"
(Search routine for the S-80, Atari, and Apple)
Mark Pelczarski, James Garon, and Rich Bouchard
page 30

"Connect-A-Dot"
(Apple Program)
Duane Barts
page 34

"Word Problems"
(S-80 educational math program)
Denslo Hamlin, Jr.
page 38

"Missile Evasion"
(S-80 version of "Head On")
Thomas Harleman
page 43

"Christmas Card"
(Apple program with sound)
Fred Pence
page 48

"Kidnapped"
(S-80 adventure)
Peter Kirsch
page 52

"Whoa"
(How to slow down S-80 listings)
Shane Causer
page 57

"Squish-3"
(Add 2K to your S-80 programs)
Dave Archibald
page 59

"Command"
(S-80 command files)
Denslo Hamlin, Jr.
page 61

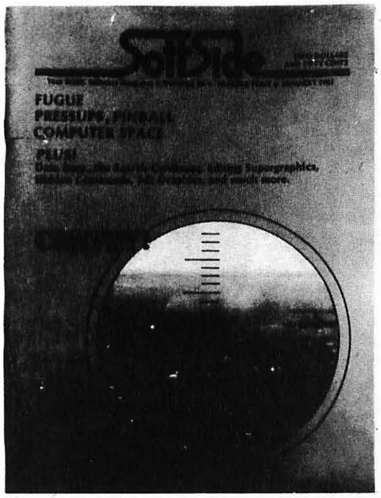
"Baseball"
(Apple Program)
Dave Bohlke and Steve Macleay
page 65

"Space Dodge"
(S-80 and Atari game)
Mike McKenna and Rich Bouchard
page 70

"States and Capitals"
(Atari matching game)
David Bohlke
page 80

"Speedello"
(Atari strategy game)
Dave Bohlke
page 88

"Reviews of Olympic Decathlon and Galaxy Invasion"
Dave Albert and Glen Ohlund
page 95



January 1981

"Developing Data Base V"
(Develops search routine further)
Mark Pelczarski, James Garon, and Rich Bouchard
page 16

"Say Yoho"
(Rules for good compunovels)
Scott Adams
page 26

"Press-ups"
(Match wits with your S-80)
Carl and Ellen Weaver
page 31

"Take-A-Part Ship Destroyer"
Phillip Case
page 39

"Computer Space"
(Apple space war game)
William Edmunds
page 40

"Pinball II"
(Apple game)
Thomas Deaux
page 43

"Convoy"
(S-80 and Apple game)
William Morris and John Cope
page 48

"Ship Destroyer"
(S-80 game)
Chris Freund
page 60

"Angle Cannon"
(Atari program)
David Bohlke
page 62

"Animation"
(Atari 3-D)
Thomas G. Marshall
page 65

"Using Those Hidden Commands"
(Level II Disk commands)
Phillip Case
page 71

"Fugue"
(Atari plays Bach fugue)
William Morris and John Cope
page 74

"Computer Graphics"
(Apple 3-D techniques)
Joan Truckenbrod
page 83

"Reviews of Olympic Decathlon and Flight Simulator"
Dave Albert and Phil Case
page 95

SoftSide
Miner
Can You Succeed Where Others Failed?
MINI GOLF
FAMOUS SAYINGS HANGMAN
TANKS-A-LOT
THE DATA BASE . . . Print Formatting
PLUS! Hiding Your Code, The Doubler Uncovered, Understanding the CLEAR Command, and much more.

February 1981

"Developing "Database VI"
(Capabilities for print formatting)
Mark Pelczarski, Rich Bouchard ,
and Phillip Case
page 16

"Computer Graphics"
(3-D rotation)
Joan Truckenbrod
page 22

"Denizens of Byte Hotel"
(Memory management)"
Roger W. Robitaille
page 36

"Eye/hand coordination"
(Using computers with children)
Judy Neyhart
page 56

"Out of Sorts"
(Sorting techniques)
George Blank
page 66

"The Art of the One Liner"
Phillip Case
page 76

"Reviews of Visicalc, Enhanced Basic, Lunar Lander, and Interactive Fiction"
Phillip Case and Dave Albert
page 80

"Long Distance"
(S-80 program)
Phillip Case
page 33

"Mini Golf"
(S-80 and Atari)
Mitch Voth and Rich Bouchard
page 42

"Miner"
(S-80, Apple, and Atari game)
Phil Case, Steve Justus, and Rich Bouchard
page 48

"Famous Sayings Hangman"
(Apple game)"
Mark Cross
page 59

"Changing Hearts"
(Atari brain twister)
Stan Ockers
page 64

"Tanks-A-Lot"
(S-80 tank duel)
Mark Koenig
page 69

"Darts"
(Apple program)
John Daoust
page 84

"Hiding Your Code (Part 1)"
(How to protect your S-80 program)
Phillip Case
page 70

"Say Yoho"
(Software piracy)
Scott Adams
page 12

SoftSide
TWO DOLLARS AND FIFTY CENTS
Your Basic Software Magazine • VOLUME III • NUMBER SIX • MARCH 1981
Volcano!
Can you escape the mountain of doom!

March 1981

"I Don't Think We're in Kansas Anymore"
Mark Pelczarski
page 12

"Say Yoho"
Scott Adams
page 19

"Math Decathlon"
Jon Voskuil
page 21

"Basic Personified"
Roger W. Robitaille
page 26

"Bugs, Worms & Other Undesirables"
SoftSide Editors
page 26

"My Side of the Page"
Lance Micklus
page 30

"Developing Data Base VII"
Mark Pelczarski and Rich Bouchard
page 34

"Volcano"
(S-80 game)
Victor T. Albino
page 36

continued on next page

continued from previous page

"Flags"
(Atari)
William Morris and John Cope
page 45

"Strategy Strike"
(S-80 and Apple)
David Steenson and Jon Voskuil
page 52

"Computer Graphics"
(3-D rotation)
Joan Truckenbrod
page 66

"Mini Golf"
(Apple)
Mitch Voth and Steve Justus
page 68

"Reviews of ABM, Galaxy Wars, and Monty plays Monopoly"
Dave Albert and Jon Voskuil
page 71

"Number Battle"
(Atari)
Dave Bohlke
page 72

"Score 4"
Stephen Miliken
page 76

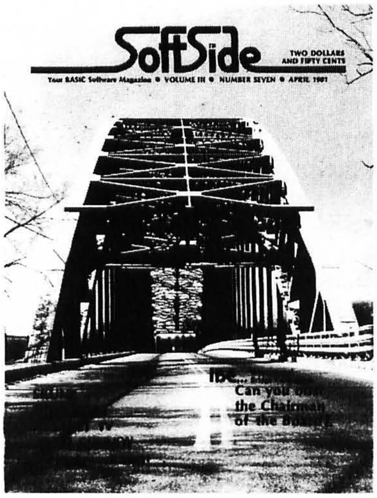
"Big Apple Adventure"
Christopher U. Light and
Chris Light Jr.
page 79

"Hardware Corner"
(Cassette recorders)
Ed Umlor
page 88

"What's New"
(Model III and Super Utility)
Ed Umlor
page 89

"Atari Memory Upgrade"
Paul Johnson
page 92

"The Money Whirlpool"
Judy Neyhart
page 94


SoftSide
TWO DOLLARS AND FIFTY CENTS
Your BASIC Software Magazine • VOLUME III • NUMBER SEVEN • APRIL 1981
Can you beat the Chairman of the Board?

April 1981

"Modifying String Variables "
(S-80)
Benjamin Junge
page 10

"Messiah"
(Atari)
William Morris and John Cope
page 20

"International Bridge Contractors"
(S-80)
Phillip Case
page 22

"Computer Graphics — Perspective"
(Apple)
Joan Truckenbrod
page 26

"Convoy"
(Atari)
William Morris and John Cope
page 34

"Assault IV"
(S-80)
Jerald Uptain
page 42

"Dominoes"
(S-80)
Peter Kirsch
page 48

"Applesoft Chaining"
Murray Summers
page 54

"Skunk"
(Apple)
John Daoust
page 58

"Strategy Strike"
(Atari)
David Steenson and Rich Bouchard
page 60

"Math Decathlon"
(Apple)
Jon Voskuil
page 66

"What To Do After the Program Has Loaded"
Ed Ting
page 69

"Robot Builder"
(S-80)
Arnold Gross
page 70

"Flags"
(Apple)
William Morris and John Cope
page 74

"Battle at Sea"
(Apple)
Jon Voskuil
page 76

"Paddle Sound Effects"
(Apple)
James Dwyer
page 83

"Reviews of Temple of Apshai, Apple Machine Language, and Dragonquest"
Michael Humes, Jon Voskuil, and Dave Albert
page 84

"Auto Modprog"
(S-80 programming perspectives)
Shane Causer
page 93

REVIEWS

HELLFIRE WARRIOR
from Automated Simulations

One of the most popular fantasy games of all times is Dungeons and Dragons™. Riding the crest of that popularity to a peak of its own, the D & D style Dunjonquest series from Automated Simulations has steadily been increasing its audience to the point where each new offering is breathlessly anticipated by its dedicated followers.

"Hellfire Warrior", the newest in this series, is no disappointment. The cover art, which comes on both box and book, displays a fiery demon hovering ominously above an unconscious female warrior. This image is bathed in an explosion of shaded blacks, yellows and reds. I wish they would sell posters of this picture, I'd buy one right now.

The cover art sets the tone for the further adventures of Brian Hammerhand, Fantasy Warrior and Adventurer, whose exploits votees of Dunjonquest have been following since the first ziplock bag. Brian, who can't seem to stay away from intrigue and danger, rescues Shalimar, a female warrior, from a taunting mob of unbelievers in her tale of a successful descent and emergence from an ancient labyrinth which has claimed every mortal who dared test its dangers. Brian has returned here because of a persisting dream which haunts him at night, a dream of an enchanted warrior queen held prisoner for 300 years deep in the demon-haunted bowels of a cavern not far from the site of Apshai. If Brian is to attempt to free her, he needs very potent magic and highly proficient weaponry, which means he will have to penetrate the depths of the Temple to acquire these things and the skill to use them....

If all this seems a little far-fetched, remember, we're still in the DOCUMENTATION! The computer is sitting dark on the desk. Carefully studying the book that comes with the game is a must. In the case of "Hellfire Warrior", it is not a bad idea to get a couple of lesser Dunjonquests under your belt, or at least to have a very good working knowledge of the game structure. "Hellfire Warrior" is brutal.

Our old friend the Innkeeper is back and from the look of it he must have been practicing selling oil to Arabs during his vacation between games. He is just as tough to deal with as he was the first time we encountered each other. "Hail and well met", is a prelude for this electronics Ebenezer to skin every dime you have, while giving up the minimum possible weaponry in return. He is every inch a match for any salesman or bargainer in existence. He's fun, in an exasperating sort of way.

Having survived verbal combat with the Innkeeper, it is now time to enter the Dunjon. Newcomers will find that they are on Level 5, or "The Lower Reaches of Apshai", the first floor of which is specifically a sequel to the "Temple of Apshai" program which leads off the series. This level is described by the author as "probably less dangerous or instantly fatal than the fourth level of TEMPLE". I was killed within five minutes, notwithstanding, and even reincarnation by the Dwarf did little good. I only bit the dust again.

"Hellfire Warrior" requires imagination. The graphics in the S-80 version are faster than the earlier games, but are still block, line and blob, which is as much as the S-80 is capable of while still delivering the screen information and room display data. These games do not pay lip service to those who believe that every rivet must be shown in a warrior's armor before a game can be called "graphic". Automated Simulations has developed its own style, improved it to the maximum and are aiming squarely at those gamers who can draw fantasy pictures in their minds.

"Hellfire Warrior", like its predecessor "Temple of Apshai", and indeed, all of the Dunjonquest series, is not for everyone. It requires dexterity, imagination, and a dedication to task, previously reserved only for Adventure fans. The production is flawless. Even the young computer waifs who periodically invade my lab found it highly entertaining, as did I.

One thing, though. As an avid science fiction and fantasy reader, I would like to see a compiled edition of the Adventures of Brian Hammerhand, in book form. The writing style and prose in the documentation booklet do not satisfy my yearning to read more about the exploits of this adventurer who would penetrate Apshai to become a Hellfire Warrior.

How about it, author? I'm sure I'm not alone.

Bob Lidell

PSEUDODISK: A Disk Simulator for Integer Basic
from Hayden

Don't let the name mislead you. Even if you already have a disk, you can probably make good use of "Pseudodisk". And if you don't have a disk, it might be just the thing to fill in until you can afford one (and to continue using after you get one).

Andy Neuschatz has written a utility which allows as many as 16 Integer BASIC programs to reside in your Apple's memory simultaneously. Programs may be typed in from the keyboard, or loaded from tape or disk, and then SAVed to "Pseudodisk". They may then be LOADed back from "Pseudodisk" and run as usual. You can get a CATalog of the "Pseudodisk" file at any time, which also tells you how much memory you've used and how much is still available. The entire file of programs may be saved to tape or disk as a unit, and loaded back in the same way. Programs are also easily linked to one another, so that one program can automatically run any other one in the file.

To give you a feel for the way it works, here is how you would load two programs from tape, store them in a "Pseudodisk" file, and then run the first program. You begin by LOADing and RUNning

continued on next page

continued from previous page

"Pseudodisk". (Although the program is in Machine Language, it loads just as a normal Integer BASIC program does.) Then you type in the following series of commands:

NEW: NEW Is now a "Pseudodisk" command.
LOAD: Normal load from tape — 1st program.
SAV FIRST: Saves 1st program in P-disk file under the name "FIRST".
NEW: Clears program memory, not P-disk file.
LOAD: Normal load from tape — 2nd program.
SAV SECOND: Saves 2nd program in P-disk file under the name "SECOND".
LOA A: Loads 1st program into program memory from P-disk.
RUN: Runs 1st program as usual.

One of the plusses of "Pseudodisk" is that the saving and loading of programs, once they are in Apple's memory, are instantaneous. If you have a group of programs that you want to use together (games, graphics displays, whatever), storing them together in a "Pseudodisk" file will allow you instant access to any one of them. Not to mention pause-free chaining of one program to the next, which even a "real" disk can't beat.

I should mention that there are a couple of crucial errors in the documentation of "Pseudodisk" which need to be corrected in order to save and retrieve files from disk successfully. The "TO DISK" and "FROM DISK" programs listed on page 8 should read as follows, according to author Neuschatz:

To disk:

```
10 DIM A$(30) : INPUT "NAME?",A$
20 POKE 202, PEEK(22)
30 POKE 203, PEEK(23)
40 POKE 76, PEEK(24)
50 POKE 77, PEEK(25)+1
60 PRINT "[ctrl-D] BSAVE ";A$;"2,A21,L5"
70 PRINT "[ctrl-D] SAVE ";A$
80 END
```

From disk:

```
10 DIM A$(30) : INPUT "NAME?",A$
20 PRINT "[ctrl-D] BLOAD ";A$;2
30 POKE 76, PEEK(24)
40 POKE 77, PEEK(25)+1
50 PRINT "[ctrl-D] LOAD ";A$
60 END
```

There are two fundamental limitations in using "Pseudodisk". It only works with Integer BASIC, not Applesoft, Pascal or Machine Language. And the number of programs that you can save in a single file is limited by the RAM of your Apple. But within those limitations, you can do a lot with "Pseudodisk". A tape user will find horizons expanded considerably, and a disk user will be able to take advantage of the ability to group and chain programs together very conveniently.

John Voskuil

Strip Dice & Concentration
from Adventure International

What!!??!! X-Rated software in a family magazine? Well, not exactly. Despite the promise (?) of the packaging, "Strip Dice and Concentration" is pretty tame stuff. It is an adult party game, whose name describes it perfectly. One program is a "take off your clothes" version of the old classic Concentration game, a grid of 30 boxes which conceal the names of items of men's and women's clothing and the occasional wild card. The other program is virtually the same thing, only it allows for up to a dozen players, and is based on graphics dice rolls on the screen.

The package itself is a little misleading. It bears a warning about "explicit sexual dialoge" (yes, that's how they spelled it) and "humorous sound effects and graphics", neither of which are particularly explicit nor offensive. Perhaps the word "tush" would upset the vehemently squeamish, but such language would hardly earn a PG rating in the movies. The graphics are either mundane or slow or both.

There are some cute moments in the program(s) when certain types of clothing are removed...the computer prints out messages on the screen which could provoke a titter or squeal here and there. But if one is brash enough to get involved in games of this nature, one usually exhibits more spontaneous wit than the program does.

One other feature of the program is the "special activities" section, wherein a bulletin flashes across the screen at random intervals, giving players special instructions of a somewhat tame nature.

"Strip Dice and Concentration" is a straightforward novelty package. There is really very little to the program other than a certain amount of cuteness. The games themselves have been around forever, people have always been able to find devices such as playing cards or dice with which they can use to persuade their friends to disrobe. The only difference with this one is that it is done on a microcomputer. It hardly poses a threat to the print or film cheesecake industries.

Dave Albert

BASKETBALL
by John Allen (Acorn Software)

There are two ways to measure sports-based game software: One is by an absolute standard whereby the game is judged against the real-life sport it attempts to simulate; the other is to judge the program on its own merits, exclusive of any other considerations. "Basketball" from Acorn Software deserves to be judged in the latter fashion.

A Machine Language program designed for one or two players, "Basketball" is an ambitious undertaking for the S-80. To begin with, the limited graphics capability of the S-80 makes it hard to approximate a fast-moving, smooth-flowing game like basketball. The lack of color on the S-80 further complicates matters by making it exceedingly difficult to distinguish which player is which. The basketball itself looks suspiciously like the alien flagship found in virtually all of the Big 5 arcade software on the market.

John Allen, the author of "Basketball", has done as well as could be expected, given the inherent limitations of the idea. The game plays reasonably well. There are five skill levels for the single-player version, in which the computer is always the home team and starts the game with possession of the bail. The two-player version favors the player on the left by making him the home team. There are never more than two players on the court, one being either the computer or the human adversary.

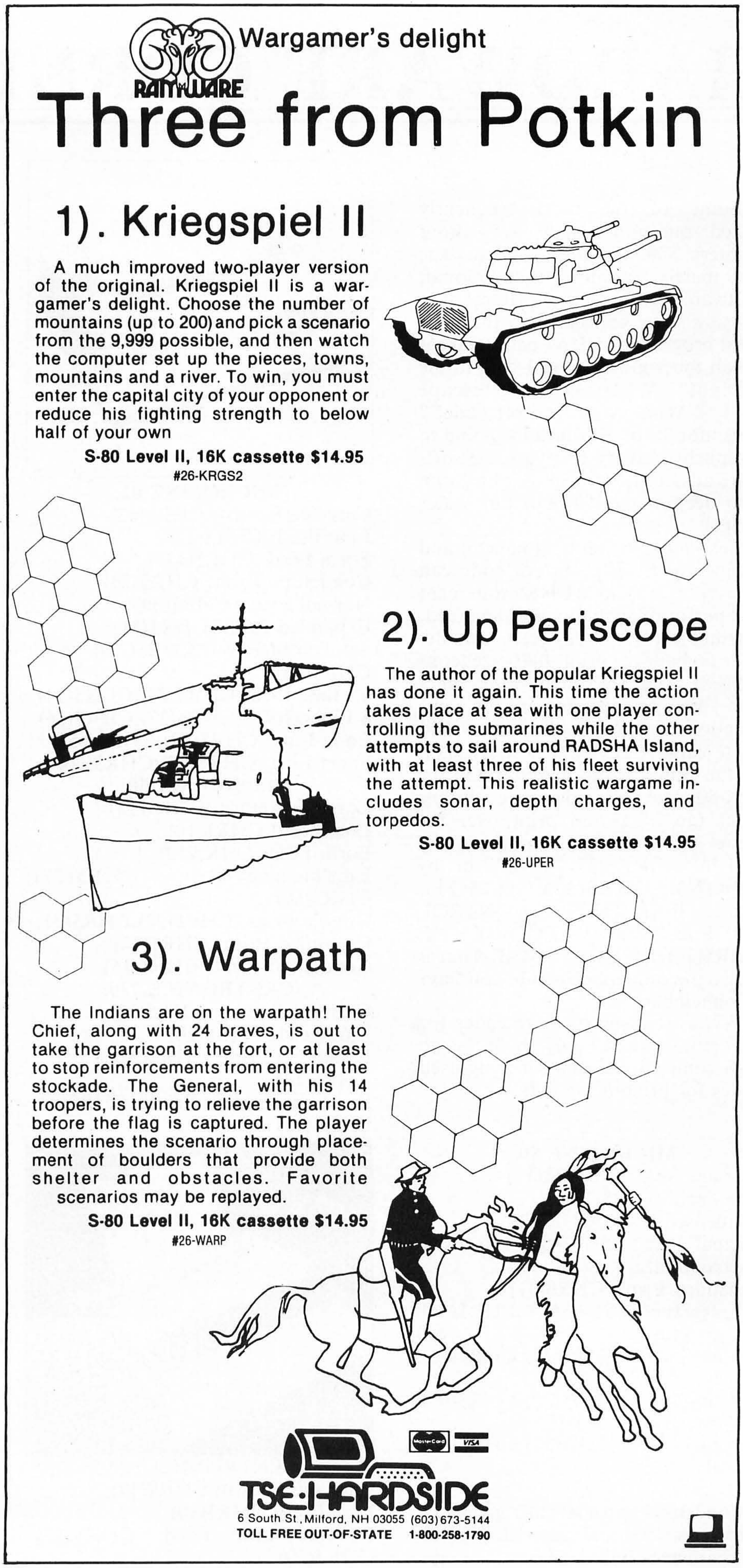
To play, one must manipulate five keys — one each to move in any direction, and one more to shoot. Here is where the first problem arises: The keys are clustered together on the keyboard in such a fashion as to preclude one-handed playing, and to make two-handed manipulation of the keys an exercise in digital contortion. Once one has refined manual dexterity to the point of being a fine art, one can then approach such esoteric matters as stealing the ball and shooting. To steal the ball, one must place his player on top of the opponent's player, at which point the player who attempted the steal is guaranteed success in the endeavor. There are no fouls in this game, so the not-so-subtle approach of a head-on tackle seems to work the best. The larcenous defender has another advantage as well, he is, with little effort, able to overtake his opponent in a footrace.

If, through some quirk of capricious fate, the offensive player reaches his basket in possession of what can only euphemistically be called the sphere, there arises the not unconsiderable matter of shooting. This reviewer has yet to figure out the logic of shooting. My player misses from underneath the basket while the computer's floats in these 25-foot jumpers that would turn Jamal Wilkes green with envy. Nevertheless, this is only the voice of frustration carping in your ear and should be ignored. If there is a lesson to be learned here, it is that always play the home team when having a friend over for a game.

Once one gets over the limitations of S-80 "Basketball", one realizes that it can be a lot of fun to play. It doesn't bear a whole lot of resemblance to the actual game, but it is written in Machine Language and thus the pace of the game is quite satisfying. After a while one gets used to the ball bouncing only when a player is dribbling it. The thrill of competition is still there, just like in the real thing. However, one major flaw in the program is that it only plays one on one. Basketball is primarily a TEAM sport, the best teams are the one that move the ball around the floor. Computerized basketball, at least in the S-80 version, has no provision for this.

All things considered, Acorn's "Basketball" is a satisfying first attempt at graphically capturing the excitement of a fine sport on the S-80. By no stretch of the imagination is it a true representation or a near simulation of the actual game, but once one gets over that stumbling block, hours and hours of enjoyment lie ahead.

Dave Albert



HARDWARE CORNER

by Edward E. Umlor

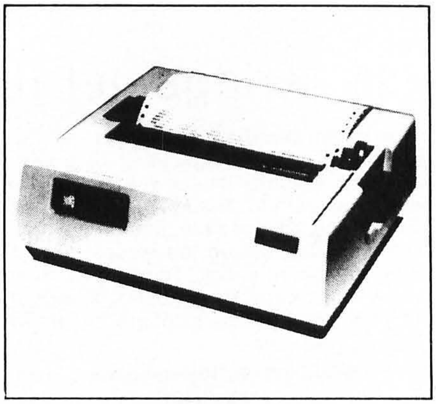
Some of the most frequently asked questions lately are about printers. The new breed of low-cost, dot matrix, graphic, bi-directional, software razzle-dazzle printers are causing some confusion to the hobbyist programmer. How can I make it do all those great things I read in the manual? What is an "escape code"? What is a "control code"? This month the Hardhead is going to go mushy and try to give some software advice. Lord help us who jump into deep waters without our water wings!

Let's start by defining control and escape codes. The control code can be output in a single CHR$ statement and performs such things as line feed, carriage return, form feed, double-wide (expanded) print, half-size (condensed) print, and many reset functions. Escape codes require the combination of two CHR$. This would be (escape) CHR$(27) + CHR$(xx) — the code for the function. On the MX-80, for example, the double print (go back and print over the same line) mode is triggered by LPRINT CHR$(27) +"G" or by LPRINT CHR$(27);CHR$(71). REMEMBER THAT A CONTROL CODE DOES NOT REQUIRE THE CHR$(27) TO DO ITS JOB. That is really the only ground rule you have to remember!

What are some of these codes for different printers? OK, here we go with some of the more heavily-used codes for printers we sell:

MICROLINE 80:
Carriage Return: CHR$(13)
Line Feed: CHR$(10)
Condensed Print: CHR$(29)
Normal Print: CHR$(30) acts as a reset/default
Expanded Print: CHR$(31)
6 Lines/Inch: CHR$(27);CHR$(54) ESC-6 default
8 Lines/Inch: CHR$(27);CHR$(56) ESC-8
80 Chrs/Line: CHR$(27);CHR$(65) ESC-A
64 Chrs/Line: CHR$(27);CHR$(66) ESC-B

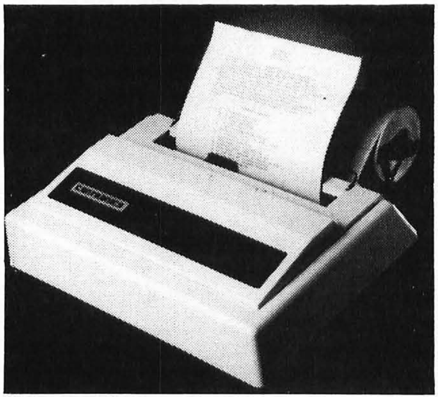
Condensed automatically gives 132 characters/line and expanded gives 40 characters/line.



MICROLINE 82:
Carriage Return: CHR$(13)
Line Feed: CHR$(10)
Form Feed: CHR$(12)
Condensed Print: CHR$(29)
Normal Print: CHR$(30)
Expanded Print: CHR$(31)
Set Top of Form: CHR$(27); CHR$(53)
6 Lines/Inch: CHR$(27);CHR$(54)
8 Lines/Inch: CHR$(27);CHR$(56)
Long Line: CHR$(27);CHR$(65)
Short Line: CHR$(27);CHR$(66)

EPSON MX-80:
Carriage Return: CHR$(13)
Line Feed: CHR$(10)
Form Feed: CHR$(12)
Emphasized Print: CHR$(27); CHR$(69)
Double Print: CHR$(27);CHR$(71)
Expanded Print: CHR$(14)
Condensed Print: CHR$(15)

CENTRONICS 730:
Carriage Return: CHR$(13)
Line Feed: CHR$(10)
Expanded Print: CHR$(27); CHR$(14)
Reset Expanded Print: CHR$(27); CHR$(15)



CENTRONICS 737:
Carriage Return: CHR$(13)
Line Feed: CHR$(10)
Reverse Line Feed: CHR$(27); CHR$(10)
Proportional Print: CHR$(27); CHR$(17)
Expanded Print: CHR$(27); CHR$(14)
Reset Expanded Print: CHR$(27); CHR$(15)
Condensed Print: CHR$(27); CHR$(20)
Reset Condensed Print: CHR$ (27); CHR$(19)

I hope that the above will help you use your printer's capabilities more fully. I have not attempted to put down all the codes, only those that I have used most often. Now, how do we make the easiest use of all these codes when programming?

The area of programming for specific usage has as many "easy" ways of doing it as there are programmers in the world. The method described here is the one that I use in setting up the print routines. I like to set up a string file of all the printer function codes that I am going to be using. This is done at the beginning of the program along with variable definitions and setting values. This allows me to just add a CODE STRING; to any LPRINT line to change the printer's format. This makes life a lot easier and makes much more efficient use of memory. For example:A$ = CHR$(10): A1$ = CHR$(13): A2$ = CHR$(27) + "E":A3$ = CHR$(27) + "F"

Continue on until all your codes are done and keep a copy (crib sheet) by your elbow as you program.

LPRINTA2$;"Now is the time for all good men to come to the aid of their country." ;A3$

This LPRINT will print the line in the emphasized mode and then reset the emphasized mode before going on. It just does not take long to get back the space used to set up the strings.

I hope that this little blurb will help you to use and enjoy your printer more. The printer is one of the most useful tools that can be connected to your computer. The more you know about using its full capabilities, the more useful it becomes. Well it's time for the old Granite Knoggin to sign off again. Please send in your questions about hardware. I'm starting to run out of problems to write about. I sure have plenty of problems but I can't write about most of them.

WHAT'S NEW

by Edward E. Umlor

HARDWARE

This month we are not doing a hardware item. There are several reasons for this.

First: We have not received any items from out there to review. If the manufacturers do not send us new product releases or samples for review, it is very difficult to do a hardware review.

Second: I have not received a Model III with disk to work over and report on as the March article indicated. I have not given up hope yet, and it will be reviewed as soon as I can.

Third: We haven't heard from any of you readers concerning any new products you have bought or heard about. It would be nice to hear something from you (good or bad), then at least I would know there really is a **SoftSide** reader. About all I can say is HELP!!!

SOFTWARE

We have received a little item from Ramware called " Autograf". TSE will soon have this program on the shelf (disk-based). The purpose is to allow you to create your graphics, save them on disk as a data file, and play them at any time. The program is written in BASIC and converts your picture into data statements for reading in a BASIC program. It does stick on a little four-liner at the beginning of the file for direct playback of your picture (one of the first to do this).

The program has complete controls for setting or resetting pixels, storing your X, Y coordinates, converting to program with data statements including line numbers, and storing the converted file on disk. These are amply described in the documentation so I won't bother taking up space here. Of the different programs I have seen of this type, I like this one for the ease of combining a picture into your program. Most are Assembly Language programs and require insertion through USR function or some direct Assembly Language routine. Not all of us are up on these fancy techniques and want to stick to basic BASIC.

How fast is it? It is written in BASIC. We all know that this will slow it down as compared to its Assembly Language counterpart. However, the fastest program in the world is totally useless, if you can't get the results into your program. I did find that once you become familiar with the speed of drawing (using the arrow keys), you could do a very complex pattern in reasonable (acceptable) time. The more you use it, the more accurate your initial moves become and the faster the picture shapes up. Speed is more a function of the operator than the program. It really helps to plot out your picture first, using one of R/S graph sheets.

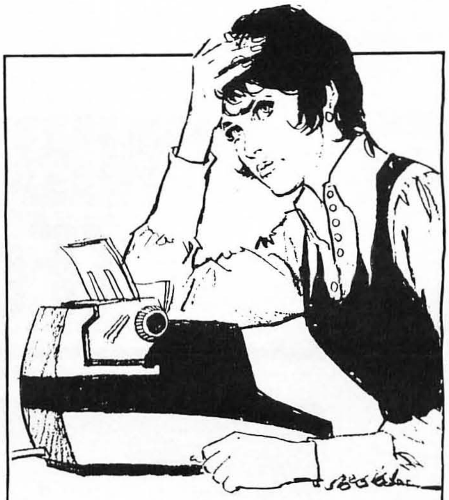
GEE MA!! I drew a picture. Now what do I do with it? This is where I like the routines built into this program. First you convert the picture. This routine allows you to back out before it's too late to add more to it, asks you for a disk filespec/tag, and then does its thing. You end up with a disk file that can be directly run or loaded from BASIC. WHOOPEE!! I have some pictures that I can recall while in BASIC. BIG DEAL!! Actually, yes it is. These files can be combined in a BASIC program fairly easily. Load in your picture file, renumber into a fairly high line number area, and start your program that will use the picture. Append, affix, or string picture files together into a whole series. Let your imagination and ability be your guide.

If you are programming in BASIC and don't like the hassle of typing in long strings to do your drawing, then this little package is for you. We hope to have it in production by the time you read this.

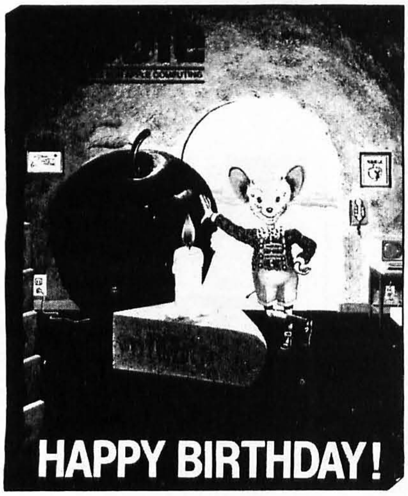
Well that's all for now. Please write in and let us hear from you.

EDITORIAL continued from page 4

postulated on fantasy. My question, then, is where are other true simulations? In the sympathetic magic/wargame version of simulations, the actual components of the exercise were carefully controlled to cover every aspect of the reality under scrutiny. In most microcomputer simulations I have seen, this faithfulness to life is secondary at best; usually it is ignored. To fantasize is nice, to simulate is sublime.



"NIBBLE® IS TERRIFIC" (For Your Apple)



NIBBLE IS: *The Reference for Apple computing!*

NIBBLE IS: One of the Fastest Growing new Magazines in the Personal Computing Field.

NIBBLE IS: Providing Comprehensive, Useful and Instructive Programs for the Home, Small Business, and Entertainment.

NIBBLE IS: A Reference to Graphics, Games, Systems Programming Tips, Product News and Reviews, Hardware Construction Projects, and a host of other features.

NIBBLE IS: A magazine suitable for both the Beginner and the Advanced Programmer.

Each issue of NIBBLE features significant new Programs of Commercial Quality. Here's what some of our Readers say:

– ***"Certainly the best magazine on the Apple II"***
– ***"Programs remarkably easy to enter"***
– ***"Stimulating and Informative; So much so that this is the first computer magazine I've subscribed to!"***
– ***"Impressed with the quality and content."***
– ***"NIBBLE IS TERRIFIC!"***

In coming issues, look for:

□ Stocks and Commodities Charting □ Assembly Language Programming Column
□ Pascal Programming Column □ Data Base Programs for Home and Business
□ Personal Investment Analysis □ Electronic Secretary for Time Management
□ The GIZMO Business Simulation Game

And many many more!

NIBBLE is focused completely on the Apple Computer systems.

Buy NIBBLE through your local Apple Dealer or subscribe now with the coupon below.

Try a NIBBLE!

nibble
Box 325, Lincoln, MA. 01773 (617) 259-9710

I'll try nibble!
Enclosed is my $17.50 (for one year).
(Outside U.S., see special rates on this page.)
□ **check** □ **money order**
Your subscription will begin with the next issue published after receipt of your check/money order.

Name ____________________
Address ____________________
City ____________________
State ____________________ Zip ____________

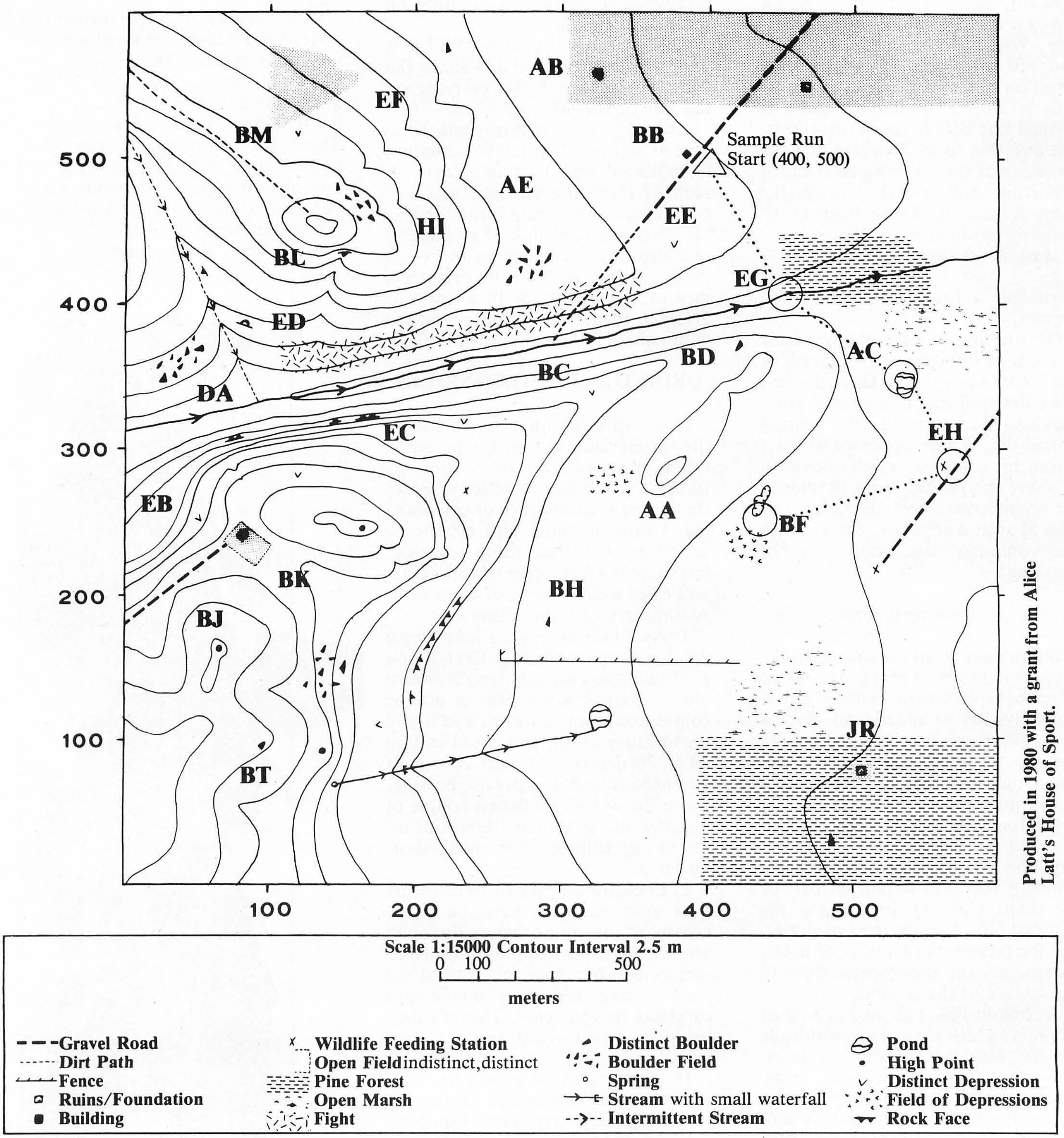
NOTE
First Class or Air Mail is required for all APO, FPO and all foreign addresses with the following additional amounts

Air Mail Postage Rates 12-14 oz. x 8
Europe $32.00
Mexico and Central America $21.00
South America $32.00
Middle East $35.00

Africa: North $32.00
Central $43.00
South $43.00
Far East, Australia $43.00
Canada $18.00
APO FPO $7.50

ORIENTEERING AT JACQUE'S COULEE

by Michael A. O'Keefe



Produced in 1980 with a grant from Alice Latt's House of Sport.

"Orienteering at Jacque's Coulee" is an S-80 simulation written in Level II BASIC and requiring 16K of memory.

If you have *no idea* what the above title means, then you are but one of the great muddy majority. But by the time you've read this article, and particularly by the time you've spent a few evenings with this computer simulation, you will be one of the elite minority who knows where the real action is.

PART I: AN INTRODUCTION TO ORIENTEERING

"Orienteering" is the sport of using a map and compass to find your way through unfamiliar terrain. The skills developed in orienteering can

continued on next page

continued from previous page

be directly applied to backpacking, hiking, mountaineering, birdwatching, and other outdoor activities.

BASIC SKILLS

1. Using a Compass

North, south, east, and west are the four cardinal points of the compass. When you orienteer, however, you will often want to go in some direction other than just these four. This is why a modern compass is divided into 360 different directions, one direction for each degree of a circle. Each of these directions is called a bearing, and is identified by the angle formed clockwise from north to the direction of travel. The bearing of east is 90 degrees; south is 180 degrees; west is 270 degrees; and north has a bearing of 0 (or 360) degrees.

On the map a normal protractor can be used to measure the bearing of any direction of travel. Once this has been determined, a magnetic compass is used to determine this bearing in real life. A special device called a protractor compass, which does both of these things, has been developed for orienteering. You will find a version of such a compass, for use with the computer simulation, on the next page.

2. Reading a Map

If you remove all the spaghetti-like contour lines which tend to clutter up all topographic maps, what is left is generally easy to understand. Such a map is like a photograph taken from high up. If you can read a highway map, you can read these portions of a topographic map.

Contour lines are more difficult to get used to. They do not represent anything you can see, yet they are probably the most important part of the map. Contour lines show the shape of hills, the steepness of slopes, and the presence of all sorts of holes, bumps, gullies, and imperfections in the surface of the ground.

A contour line is defined as a set of all points which are at a given altitude — say 800 feet above sea level. Topographic maps contain many contour lines, each at an altitude a certain interval from the text. A map could have contour lines for 800, 810, 820, 830 and so on; such a map would have a contour interval of 10 feet. The following rules should help in understanding contour lines.

1. A hill summit is circled by contour lines.

2. If you travel across one contour line to another, you are on a slope going either uphill or downhill.

3. The closer together contour lines are, the steeper is the slope they represent.

4. If a contour line has a "v" in it and the "v" points uphill, the "v" represents a depression or gully on the side of the hill.

5. If a contour line has a "u" in it and the "u" points downhill, the "u" represents a bump or ridge on the side of the hill.

Any map is a miniature representation of an area. Just HOW miniature is indicated by the map's scale. A scale of 1:15000 means that one unit of measure on the map is the same as 15,000 of the same unit of measure in real life. Since converting inches to yards or feet is difficult, orienteers measure in metric. With a scale of 1:15000, then, one centimeter on the map equals 150 meters in real life.

ORIENTEERING TECHNIQUES

In orienteering one tries to choose the fastest route between two points. Often the shortest route is not the fastest. The ability to select which is the fastest is the essence of orienteering. Common sense and experience are essential in this route selection, but there is a basic core of techniques and rules which can be of great help. A summary of these follows.

1. Avoid depending on following a precise compass bearing. Even if you go slowly and carefully, you'll almost always wander a few degrees off the course you want to travel; and if you try to hurry at all, you could end up 20 to 30 degrees off your course. If you need to follow a precise bearing, try to do so for less than a couple of hundred meters; a few degrees of error is acceptable over such short distances.

2. Look for an easy-to-find feature near your goal. If there's a pond, crossroad, or some other easily found feature within a couple of hundred meters of your goal, go fast to the easy feature and then slowly and carefully to your goal. This is called using an attack point and is a very useful technique.

If your goal is on a path or road, you can take advantage of this by travelling fast to a portion of the path to one side of your goal, then following the path to the goal. It is important, of course, that you travel so that when you reach the path you're sure which way to follow it to get where you're going. This technique is called aiming off and is also very useful.

3. Recognize when you've made an error. Since you WILL make mistakes, it's valuable to try to detect them as soon as possible. Look for features that will tell you when you've gone too far, or wandered to the right, or gone in the wrong direction.

4. Be aware of how far you've gone. Keep a mental note of how far you've travelled from your last known location. You can simply guess at this by intuition, but it is better to keep a count of your paces. This information will tell you when you've gone past your goal, where you are on a path, or about where you are when you get disoriented. (Orienteers never get "lost"!)

5. Follow along linear features

when possible. If a path, roadway, ridge top, stream, or other linear feature roughly follows the direction you're traveling, follow along with it. Such features are easy to locate and can be followed as fast as you care to travel with no danger of wandering off course. The problem with using such a feature is that it's hard to determine where to leave it. Pace-counting or an identifying feature (trail intersection, building, etc.) is necessary to establish when it's time to change direction.

6. Go as far as you think you should, then go a little further. Unless you have a very good reason to think otherwise, you generally don't reach a goal as fast as you think you should. So if you're travelling along and haven't been pace-counting, and there are no distinctive features in sight, you generally won't reach your goal until after you expect to.

7. Believe your compass. If you think that your compass is wrong, check to be sure that there aren't any large pieces of metal immediately around it. If there aren't, assume that the compass is correct — even if you're sure that it's wrong. Your internal sense of direction is much easier to confuse than is the compass.

8. If you get disoriented, look for a long, distinctive feature and head for it. Typically, such a feature is a road. When you reach it, follow it until you reach a distinctive feature (e.g. a stream crossing) and use that to establish your location.

PART II: THE COMPUTER SIMULATION

Jacque's Coulee is the local name of a small stream in central Minnesota. Along that stream is an orienteering area complete with 24 permanent markers, as detailed on the accompanying map. You are invited to orienteer there; and you don't even have to go to Minnesota to do so. With the Jacque's Coulee orienteering simulation you can "run" in this area from the comfort of your armchair and keyboard. The simulation allows the computer to be your eyes and legs at Jacque's Coulee while you make the orienteering decisions. Do you take the road to travel fast, or go cross-country to save distance? Use an attack feature, or precise bearing? Up over the hill, or around it? The decisions, and consequences, are yours.

HOW TO PLAY THE SIMULATION

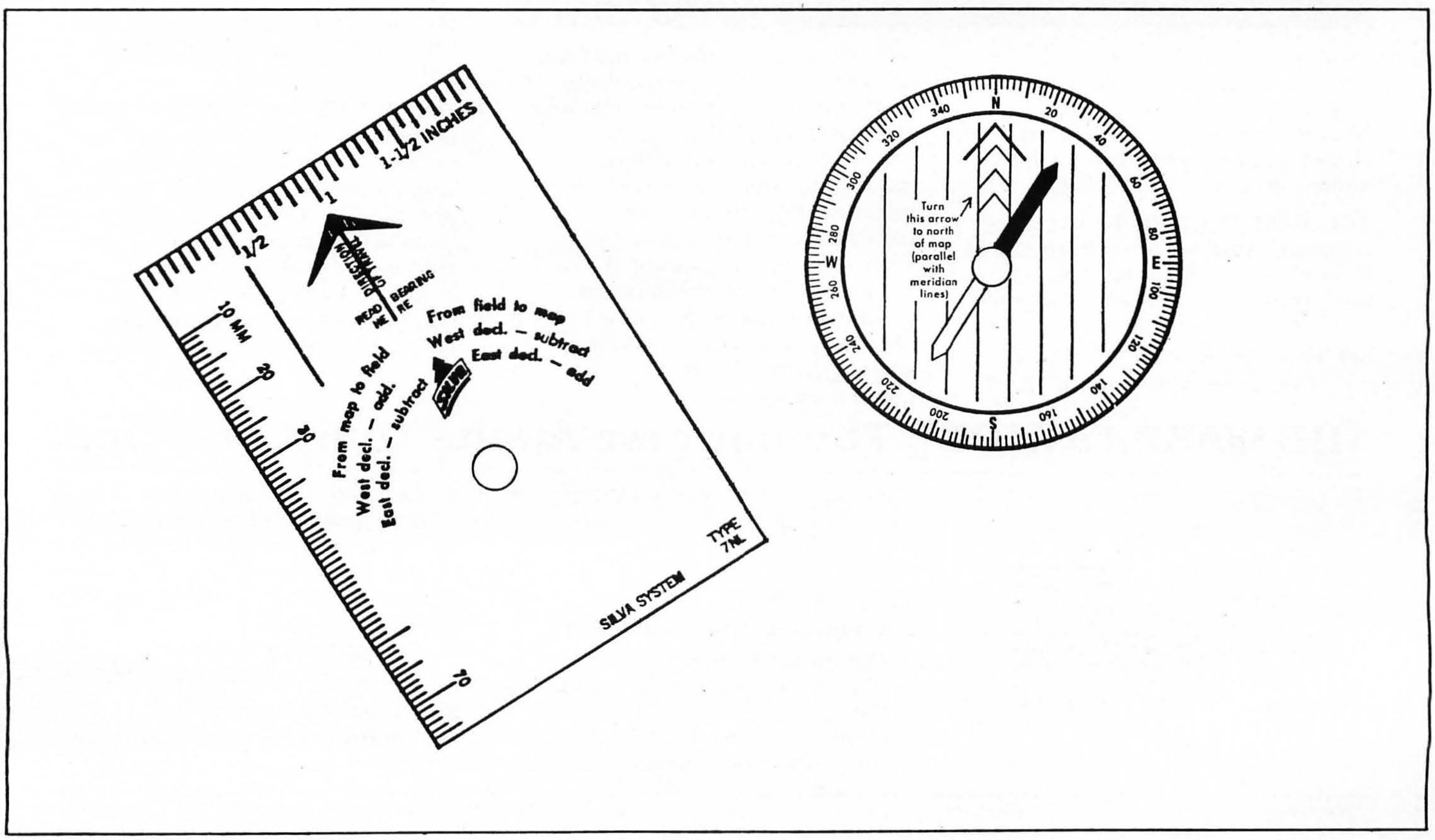
The computer uses a 600-by-600 grid to locate any place in the orienteering area. Whenever the computer gives you coordinates, or you give them to the computer, the column is always the first number and the row is the second. See-through graph paper can be used to help you determine the location of any coordinates.

As the simulation begins the computer will ask you for the coordinates of the location at which you want to start. You can tell the computer this by typing the column, then a comma, then the row, and then ENTER. Your direction of travel is given to the machine by typing a bearing and then pressing ENTER. Any bearing between 0 and 360 may be used (0 and 360 both representing straight north). The vertical grid lines on the map indicate magnetic north, with north to the top.

You may run along the bearing you select at any one of five speeds: 1 is very slow, 2 is slow, 3 is normal, 4 is fast, and 5 is very fast. To select your speed, simply type a number and press ENTER. It is important that you remember that the faster your speed, the greater the error you may realize as you try to follow along your bearing. At the slowest speed the maximum navigational error that you could realize is 2 degrees; at the fastest speed it could be as much as 35 degrees.

After you have set your speed, the computer will move you 25 meters along the bearing you selected, altered by navigational error. At this point you will stop and "look around." Generally the computer

continued on page 91



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

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

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

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

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

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

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

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

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

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

THE WARP FACTOR™. The Universe Awaits Your Command.

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

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

FOR YOUR APPLE®:

□ **Computer Bismarck:** $59.95.

□ **Computer Ambush** (a tactical simulation of man-to-man combat in WWII): $59.95.

□ **Computer Napoleonics**, the Battle of Waterloo: $59.95.

□ **Computer Quarterback** (a real-time strategy football game): $39.95.

□ **Computer Conflict** (two modern-day tactical warfare simulations featuring *REBEL FORCE* and *RED ATTACK!*): $39.95

□ **Computer Air Combat** (a simulation of air combat in WWII): $59.95.

FOR YOUR TRS-80®:

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

Apple is a registered trademark of Apple Computer Inc. *TRS-80 is a registered trademark of Tandy Corporation.*

continued from page 89

will give you a short description of your surroundings (e.g. "You are by a stream," "You are in a boulder field"). Unless the description mentions vegetation, you can assume that you are in an open deciduous forest. If you are on a hillside, the computer will give you a rough idea of the direction of the slope (e.g. "You are on a slope — N is uphill"). This direction is meant only to be a general description of the slope. If north is described as uphill, it is possible that northeast is actually straight uphill and north is uphill at an angle. The description is never wrong, but may not be precise.

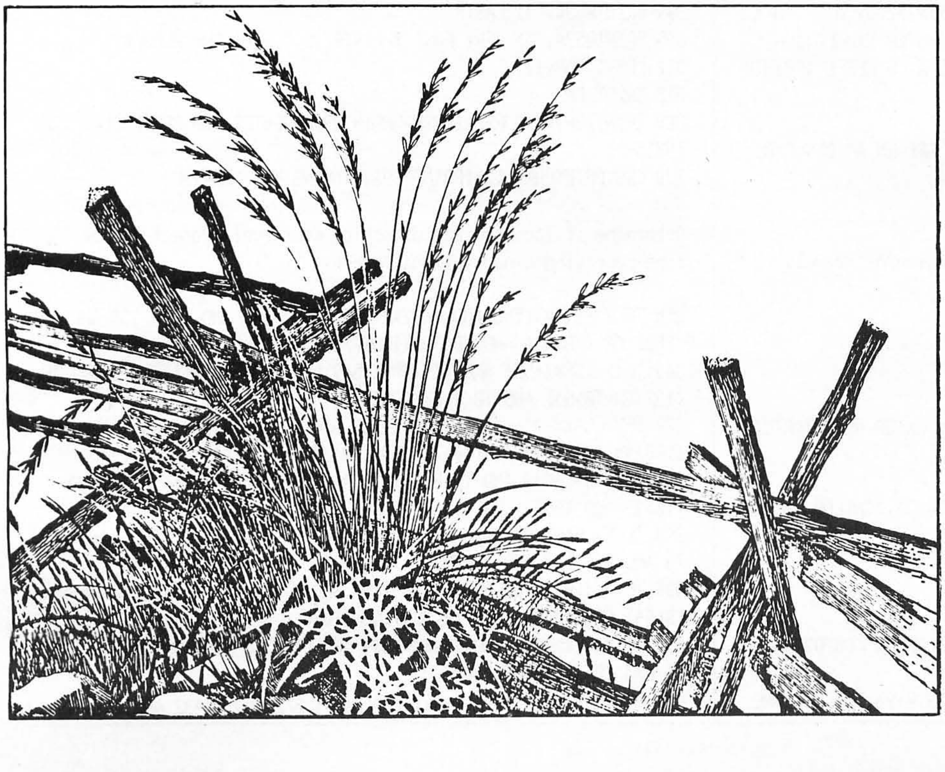
If you are within sight of an attack feature when you stop to look around, the computer will give you the coordinates of your location. This is done only within sight of features that are distinct enough to allow you to orient your exact location. Such features include buildings, isolated distinct boulders, high points, stream crossings on trails, and others.

At the same time that the computer describes your location, it will ask you if you want to continue on the course just set. Pressing "Y" will cause the computer to move you another 25 meters on your bearing (altered by navigational error) at the speed you last selected. Pressing "N" will allow you to change direction and speed.

It is possible to exercise almost all orienteering techniques while "running" at Jacque's Coulee. Attack points, collecting features, roads, corridor features, rough compass, and aiming off all can be used in the same way as in real orienteering events. Following a linear feature (such as a road or stream) to avoid any directional error is possible by simply reaching such a feature and then entering a direction of travel equal to the direction in which the feature runs. It is important that this be accurate within a degree or two, or else the computer will assume that you want to angle away rather than follow along it. In this case you would get a navigational error as usual. The simulation does not allow any way to use contouring as a navigational technique.

When you come within sight of a marker, the computer will tell you that you can see it. It will also ask if you want to go to the marker. If you do, then press "6". This will move you to the marker and allow you to see its control code. Be sure to check it against the code of the marker you are looking for, since there are 24 markers in Jacque's Coulee and some are near each other. If you want to punch in, type "Y" in response to the computer's question, and it will register the marker on your control card. If you don't punch in, no record is kept of your having reached the marker.

The computer constantly updates and displays the time it has taken to travel along your route. The time is determined both by your speed and by the terrain; you can go one-third faster on a gravel road than through flat woods. The time that it takes you to consider your route and respond to the computer's prompts is not accumulated — only the simulated run time is counted.



As you run at Jacque's Coulee you gradually get more and more tired, just as you would in a real competition. The more tired you get, the harder it is for you to concentrate, and therefore the more navigation errors you make. For every "fatigue error point" you accumulate, the maximum possible error you could have at any given speed increases by one degree. Running at "very slow" speed adds fatigue points slowly — only one point in nearly two hours. By contrast, running only five minutes at "very fast" speed will add the same fatigue unit. The computer displays your current fatigue error points throughout the simulation.

If you become hopelessly disoriented, you can either abandon the course by entering a bearing of 999, or pray. For the latter option, type "PRAY" in response to the computer's question about whether you want to continue as you have been.

USES OF THE SIMULATION

1. As a Teaching Aid. Any subject is learned more thoroughly when students discover techniques on their own rather than simply being taught abstract rules. In addition, the simulation can be used to illustrate both how and why various orienteering techniques are used.

2. Point-to-Point Competition. Any number of point-to-point courses may be set using the 24 markers in the area.

3. Score Competition. You can set a start location and then assign appropriate score points to all or some of the available markers. Competitors are than allowed a given accumulated time to put as many markers on their control card as they can. Penalty points are given to those who use more than the allotted time. It is suggested that the finish point be a specified marker so that it can be reached precisely at the end.

4. Lost. An interesting game for two or more competitors can be played using the simulation. One player secretly enters a start location into the computer. Another player then takes over and tries to figure out where he is. He does this by deducing his general location from the description of his surroundings given by the computer, and then navigating to a feature distinctive enough to allow him to orient his exact location. The person who is able to orient himself in the shortest time wins.

continued on next page

continued from previous page

VARIABLES:

A: Slope of travel bearing.
AD %: Distance traveled by runner since last change in speed or direction.
B: The point on the Y axis intersected by the travel bearing.
BG: Runner's intended bearing.
BY$; Used in describing runner's location.
CARD$; String of control codes of markers at which runner has punched in.
CC$: Control code of a marker.
D: Distance traveled by runner since last time/fatigue update (usually 25 m.)
DB: Difference between travel bearing and straight uphill.
DER: Actual directional error.
DR$: Cardinal direction closest to straight uphill.
F: Percent of top speed which defines the other four speeds.
FER: Maximum fatigue-caused directional error.
INC: X-distance between points which are 25 m. apart on the runner's travel bearing.
MER: Maximum possible speed-caused directional error.
MSP: The time (in seconds) needed to cover one meter of a given terrain at maximum speed.
N$, 0$: Used in describing runner's location.
SP: Speed code of runner.
SR: Flag indicating whether the runner is searching for a marker. (Used to avoid "seeing" a marker that the runner is leaving.)
TB: Runner's travel bearing.
TM, TS: Runner's accumulated time in minutes and seconds.
X, Y: Runner's horizontal and vertical coordinates.
XT, YT: Horizontal and vertical coordinates of point which is 1000 m. further along runner's bearing.
ZONE: Flag indicating whether runner's current location will allow exact orientation.

GLOSSARY

AIMING OFF — An orienteering technique in which the bearing followed is to one side of a direct bearing. This is done so that the direction in which to look for the goal is known.
ATTACK POINT — A distinctive, easy-to-find feature near a goal which is used as the start of a final approach to the goal.
BEARING — Also called azimuth, a direction of travel defined by the angle formed clockwise from magnetic north (ex., the bearing of east is 90°).
CATCH FEATURE — A linear feature (ex., stream, path) that is perpendicular to your direction of travel and which can be used to indicate your general location, and especially to indicate when you've gone beyond your goal.
CLUE SHEET — A list of short descriptions of the locations of orienteering markers and the identifying codes written on those markers.
CONTROL CARD — The card which is punched with identifying codes at each orienteering marker to prove which markers were found.
CONTROL CODE — A unique identifying code, either letters or a number, that is written on every orienteering marker and is used to confirm that the marker found is the marker sought.
FIGHT — Thick brush of any sort that a person has to"fight" to get through.
HAND RAIL — A linear feature (ex., stream, path) that parallels your direction of travel and which can be used to guide you to your goal.
RE-ENTRANT — A small valley on a hillside, usually formed by an intermittent stream.
ROCK FACE — A small exposed rock bluff.
SADDLE — The low point on a ridge, normally a low point between the summits of a hill with more than one summit.
SPUR — A small ridge on a hillside, usually formed by two intermittent streams, one on each side of the spur.

```
ORIENTEERING AT JACQUE'S COULEE  --  PROGRAM LISTING

Initialize program and variables.

10 CLS:PRINT@210,"O R I E N T E E R I N G":PRINT@268,"A T   J A
C Q U E ' S   C O U L E E":PRINT@403,"(A COMPUTER SIMULATION)":
PRINT@599,"COPYRIGHT 1980":PRINT@661,"MICHAEL A. O'KEEFE":PRINT@
918,"(PRESS ANY KEY)"
20 IFINKEY$=""GOTO20
90 CLEAR100:RANDOM:O$="YOU ARE ON A":N$="YOU ARE IN A":BY$="YOU
ARE BY A":CARD$="":FER=0:TM=0:TS=0:ZNE=1
110 CLS: PRINT CHR$(23)

Input the start coordinates, speed, and direction of travel.

120 PRINT@262,"WHAT ARE THE COORDINATES"
130 PRINT@338,"OF THE START?"
150 PRINT TAB(10);
160 INPUT X,Y
170 IFX<0 OR Y<0 OR X>600 OR Y>600 PRINT@912,"ENTER NEW START":G
OTO150
190 CLS:PRINT CHR$(23)
200 PRINT@264,"WHAT IS THE BEARING YOU":PRINT@334,"INTEND TO TRA
VEL"
220 PRINT TAB(12):AD%=0
230 INPUT TB
235 IF TB=999 GOTO8000
240 IF TB<0 OR TB>360  PRINT@908,"ENTER NEW BEARING":GOTO220
250 CLS:PRINT CHR$(23)
260 PRINT@ 134,"YOU MAY RUN THIS BEARING":PRINT@196,"AT ANY ONE
OF THESE SPEEDS:"
280 PRINT@338,"1) VERY SLOW"
285 PRINT@402,"2) SLOW"
290 PRINT@466,"3) NORMAL"
300 PRINT@530,"4) FAST"
305 PRINT@594,"5) VERY FAST":PRINT@718,"ENTER THE NUMBER"
310 PRINT TAB(12)
315 INPUT SP
320 IFINT(SP)<>SPORSP<1ORSP>5PRINT@906,"ENTER NEW SPEED CODE":GO
TO310
330 CLS:PRINTCHR$(23):PRINT@396,"TAKING THE BEARING"

Determine if location and direction of travel protect runner
from receiving a navigational error.

350 IF Y>-2.1*X+508 AND Y<-2*X+530 AND Y>336 AND ((TB>150 AND TB
<158) OR (TB>330 AND TB<338))GOTO7000
360 IFY>-.78*X+554 AND Y<-.77*X+569 AND Y>460 AND((TB>123 AND TB
<132)OR(TB>302 AND TB<310))GOTO7000
370 IFY>.84*X+346 AND Y<.99*X+339 AND Y<-1.2*X+807 AND Y>480 AND
((TB>44 AND TB<52)OR(TB>220 AND TB<230))GOTO7000
380 IFY<.8*X+348 AND Y>.81*X+328 AND Y>-X+630 AND Y<-X+770 AND (
(TB>45 AND TB<53) OR (TB>225 AND TB<233))GOTO7000
390 IF Y>.44*X+377 AND Y<.4*X+400 AND Y>-1.67*X+763 AND Y<-2*X+9
70 AND ((TB>62 AND TB<72)OR(TB>242 AND TB<252))GOTO7000
400 IF Y<1.21*X+35 AND Y>1.25*X-5 AND Y>.167+323 AND ((TB>34 AND
TB<44) OR (TB>216 AND TB<224))GOTO7000
410 IF Y<.2*X+320 AND Y>.2*X+304 AND ((TB>75 AND TB<83) OR (TB>2
55 AND TB<263))GOTO7000
420 IF Y<3.5*X-65 AND Y>4*X-178 AND Y>372 AND Y<432 AND ((TB>12
AND TB<20) OR (TB>192 AND TB<200))GOTO7000
440 IFY>.44*X+182 AND Y<.45*X+195 AND X<370 AND X>210 AND ((TB>6
1 AND TB<69)OR(TB>241 AND TB<249))GOTO7000
450 IFY>1.33*X-483ANDY<1.3*X-442ANDY>-.8*X+620AND((TB>34ANDTB<42
)OR(TB>214ANDTB<222))GOTO7000
460 IF Y>-.02*X+160 AND Y<-.02*X+170 AND X>260 AND X<422 AND((T
B>86 AND TB<94) OR (TB>266 AND TB<274))GOTO7000
470 IF Y<.2*X+48 AND Y>.23*X+28 AND X>46 AND X<320 AND ((TB>75 A
ND TB<83) OR (TB>255 AND TB<263))GOTO7000
480 IFY>.74*X+172 AND Y<.72*X+190 AND X<70 AND ((TB>49 AND TB<57
) OR (TB>229 AND TB<237))GOTO7000
485 IFY>335ANDY<354ANDX>528ANDX<544GOTO7000
```

Assign navigational error.

```
500 ONSPGOTO505,510,515,520,525
505 MER=3:GOTO550
510 MER=6:GOTO550
515 MER=15:GOTO550
520 MER=25:GOTO550
525 MER=35
550 MER=MER+FER
560 MER=MER*.8:DER=RND(MER)-RND(MER)+RND(MER)-RND(MER):DER=INT(D
ER+.5):TB=TB+DER:IFTB<0TB=TB+360ELSEIFTB>360TB=TB-360
650 IFTB<4ORTB>374VERT=1ELSEIFTB>176ANDTB<184VERT=-1ELSE660
655 Y=Y+VERT*6.562:GOTO850
660 VERT=0
```

Move runner to new coordinates.

```
670 QD=INT(TB/90)+1:K1=.01745329:K2=262.48:ON QDGOTO680,690,700,
710
680 TX=X+SIN(TB*K1)*K2:TY=Y+COS(TB*K1)*K2:GOTO720
690 TX=X+COS((TB-90)*K1)*K2:TY=Y-SIN((TB-90)*K1)*K2:GOTO720
700 TX=X-SIN((TB-180)*K1)*K2:TY=Y-COS((TB-180)*K1)*K2:GOTO720
710 TX=X-COS((TB-270)*K1)*K2:TY=Y+SIN((TB-270)*K1)*K2
720 A=(Y-TY)/(X-TX):B=Y-A*X:INC=(TX-X)/40
800 X=X+INC:Y=A*X+B
```

Check to see if runner is still on the map; if not, end the run.

```
850 D=25:AD%=AD%+D
860 IFX>0ANDX<600ANDY>0ANDY<600GOTO1140
960 CLS:PRINTCHR$(23):PRINT@260,"YOU HAVE RUN OFF THE MAP":PRINT
@322,"AND BEEN EATEN BY MONSTERS!":PRINT@842,"PRESS C TO CONTINU
E"
970 IFINKEY$="C"THEN8000ELSE970
```

Determine which square inch the runner is in.

```
1140 CLS:PRINTCHR$(23):X%=X/100:X%=X%+1:Y%=Y/100:Y%=Y%+1:IFINKEY
$="R"PRINT@448,X;Y;TB;MSP
1150 ONX%GOTO1160,1170,1180,1190,1200,1210
1160 ONY%GOTO2000,2100,2200,2300,2400,2500
1170 ONY%GOTO2600,2700,2800,2900,3000,3100
1180 ONY%GOTO3200,3300,3400,3500,3600,3700
1190 ONY%GOTO3800,3900,4000,4100,4200,4300
1200 ONY%GOTO4400,4500,4600,4700,4800,4900
1210 ONY%GOTO5000,5100,5200,5300,5400,5500
```

Determine the zone that the runner is in, set the defining characteristics, and go to the appropriate program line.

```
2000 DR$="W":IFY<72ANDY>30ANDX>13ANDX<26GOSUB10500:GOTO6200
2010 IFY>90ANDX>90MX=95:MY=95:CC$="BT":GOSUB10500:GOTO6000
2020 IFY<-.5*X+105ANDY>-.5*X+90ANDY<2.75*X-128GOSUB11100:GOTO620
0
2030 GOSUB11300:GOTO6200
2100 IFX>55ANDX<70ANDY>135ANDY<170MX=65:MY=162:CC$="BJ":GOSUB105
00:GOTO6000
2110 IFY>.74*X+172ANDY<.72*X+190GOSUB10700:GOTO6200
2120 IFX>80ANDY<108GOSUB11450:GOTO6200
2130 IFY<-2*X+190ANDY>133THENDR$="SE":GOSUB11100:GOTO6200
2135 IFX>50ANDX<61ANDY>115ANDY<136GOSUB11250:GOTO6200
2140 IFX>80ANDY<140ANDY>130THENDR$="W":GOSUB11100:GOTO6200
2150 IFY<3.3*X-43THENDR$="W":GOSUB11300:GOTO6200
2160 DR$="SE":GOSUB11300:GOTO6200
2200 IFY<1.2*X+158ANDY>1.7*X+60ANDY>-.7*X+281ANDY<-.7*X+312MX=99
:MY=233:CC$="BK":MSP=.4:GOSUB10500:IFX>90:GOTO6000ELSEGOTO6200
2210 IFX>70ANDX<82ANDY<292ANDY>278GOSUB10500:GOTO6200
2215 IFX>45ANDX<55ANDY<259ANDY>248MX=50:MY=251:CC$="EB":GOSUB105
00:GOTO6000
2220 IFY>.74*X+172ANDY<.72*X+190ANDX<75GOSUB10700:GOTO6200
2230 IFY<1.7*X+90GOSUB11250:GOTO6200
2240 DR$="SE":GOSUB11300:GOTO6200
2300 IFY<390ANDY>372ANDX<90ANDX>75GOSUB10500:MX=82:MY=384:CC$="E
D":GOTO6000
2310 IFX>85ANDY>326ANDY<339:MX=91:MY=332:CC$="DA":GOSUB10500:GOT
O6000
2320 IFX>20ANDX<55ANDY<.5*X+349ANDY>.7*X+321GOSUB10600:GOTO6200
2330 IFX>69ANDX<83ANDY<315ANDY>302GOSUB11200:GOTO6200
2340 IFY<.2*X+320ANDY>.2*X+304GOSUB11500:GOTO6200
2350 IFY>-2.1*X+508ANDY<-2*X+518GOSUB10650:GOTO6200
2360 IFY>=.2*X+320THENDR$="N":GOSUB11300:GOTO6200
2370 DR$="S":GOSUB11400:GOTO6200
2400 IFX>46ANDX<54ANDY<429ANDY>415GOSUB11200:GOTO6200
2410 IFY>-2.1*X+508ANDY<-2*X+530GOSUB10650:GOTO6200
2420 IFY<=-2.1*X+508THENDR$="W":GOSUB11300:GOTO6200
2430 IFY>-.78*X+554ANDY<-.77*X+569GOSUB11650:GOTO6200
2440 IFY>=-.77*X+569THENDR$="SW":GOSUB11300:GOTO6200
2450 DR$="NE":GOSUB11300:GOTO6200
2500 IFY>-.78*X+554ANDY<-.77*X+569GOSUB11650:GOTO6200
2510 IFY<=-.78*X+554THENDR$="NE":GOSUB11300:GOTO6200
2520 DR$="SW":GOSUB11300:GOTO6200
2600 IFX>130ANDX<140ANDY>88GOSUB10500:GOTO6200
2610 IFX>139ANDX<148ANDY>65ANDY<75GOSUB10500:GOTO6200
2620 IFY<.2*X+48ANDY>.23*X+28ANDX>146IFX>187GOSUB10500:GOTO6200:
ELSEGOSUB11500:GOTO6200
2630 IFY>-8*X+1410ANDY<-3*X+640GOSUB10800:GOTO6200
2640 DR$="W":GOSUB11300:GOTO6200
2700 DR$="W":IFX>170ANDY<180ANDY>105ANDY<115GOSUB10500:GOTO6200
2710 IFX>121ANDX<151ANDY>130ANDY<163GOSUB10600:GOTO6200
2730 IF(Y>125ANDY<135ANDX>195)OR(Y>145ANDY<2.3*X-307)GOSUB11200:
GOTO6200
2740 IF(X>163ANDX<190ANDY<160)OR(Y>160ANDY<1.1*X-17)GOSUB10800:G
OTO6200
2750 IFY>125ANDY<145ANDX<128GOSUB11100:GOTO6200
2760 GOSUB11300:GOTO6200
2800 IFX>152ANDX<180ANDY>238ANDY<260GOSUB10500:GOTO6200
2810 IFX<122ANDX>114ANDY<285ANDY>273GOSUB10500:GOTO6200
2820 IFY>-1.6*X+506ANDY<-1.3*X+474ANDY<238THENDR$="NW":GOSUB1110
0:GOTO6200
2830 IFY<-.8*X+343ANDX<150GOSUB11250:GOTO6200
2840 IFY>260ANDY>1.3*X+55THENDR$="S":GOSUB11300:GOTO6200
2850 IFY>-1.7*X+520THENDR$="W":GOSUB11300:GOTO6200
2860 IFY>240THENDR$="E":GOSUB11300:GOTO6200
2870 DR$="N":GOSUB11300:GOTO6200
2900 IFY<.17*X+352ANDY>.17*X+332ANDX>110GOSUB11600:GOTO6200
2910 IFY<3.5*X-65ANDY>4*X-178ANDY>372THENDR$="N":GOSUB11100:GOTO
6200
2920 IFY<.1*X+312ANDY>.14*X+296AND((X>135ANDX<150)OR(X>156ANDX<1
75))MX=165:MY=324:CC$="EC":ZNE=2:GOSUB11200:IFX>155GOTO6000ELSE6
200
2930 IFY<.15*X+303ANDY>.19*X+284THENDR$="S":GOSUB11400:GOTO6200
2940 IFY>=.2*X+320THENDR$="N":GOSUB11300:GOTO6200
2950 IFY<=.2*X+304THENDR$="S":GOSUB11300:GOTO6200
2960 GOSUB11500:GOTO6200
3000 IFX>130ANDX<140ANDY>450ANDY<459GOSUB10500:GOTO6200
3005 IFY>-.78*X+554ANDY<-.77*X+569ANDX<130GOSUB11650:GOTO6200
```

continued on next page

continued from previous page

```
3010 IFY<437ANDY>425ANDX<155ANDX>140MX=152:MY=433:CC$="BL":GOSUB
10500:GOTO6000
3020 IFY>.84*X+346ANDY<.99*X+339ANDY>480THENDR$="SW":GOSUB11100:
GOTO6200
3030 IFY<.8*X+348ANDY>.81*X+328ANDY>-X+630GOSUB11450:GOTO6200
3040 IFY>.44*X+377ANDY<.4*X+400ANDY>-1.67*X+763THENDR$="SW":GOSU
B11100:GOTO6200
3050 IFY<3.5*X-65ANDY>4*X-178ANDY<432THENDR$="N":GOSUB11100:GOTO
6200
3060 IFY<-X+643ANDY>-X+610ANDY<481ANDY>456GOSUB10600:GOTO6200
3070 IFY>-.67*X+540THENDR$="SW":GOSUB11300:GOTO6200
3080 DR$="N":GOSUB11300:GOTO6200
3100 DR$="SW":IFY<.6*X+644ANDY>.67*X+443GOSUB10900:GOTO6200
3110 IFY>510ANDY<520ANDX>115ANDX<124MX=119:MY=514:CC$="BM":GOSUB
10500:GOTO6000
3120 IFY>.84*X+346ANDY<.99*X+339MX=187:MY=511:CC$="EF":ZNE=2:GOS
UB11100:IFX>180ANDX<190GOTO6000ELSE6200
3130 GOSUB11300:GOTO6200
3200 IFY<.2*X+48ANDY>.23*X+28GOSUB11500:GOTO6200
3210 IFY<-2*X+510ANDY>-2*X+450THENDR$="W":GOSUB11300:GOTO6200
3220 GOSUB10800:GOTO6200
3300 IFY>-.02*X+160ANDY<-.02*X+170ANDX>260GOSUB11050:GOTO6200
3305 IFX>252ANDX<261ANDY<165ANDY>151GOSUB10500:GOTO6200
3310 IFX>285ANDY>175ANDY<189MX=290:MY=182:CC$="BH":GOSUB10500:GO
TO6000
3320 IFY<.2*X+48GOSUB11500:GOTO6200
3330 IF(Y<1.5*X-150ANDY>1.75*X-222)OR(Y<135ANDY>123ANDX<205)GOSU
B11200:GOTO6200
3340 GOSUB10800:GOTO6200
3400 IFX>210ANDX<226ANDY>220ANDY<232GOSUB10500:GOTO6200
3410 IFY>.44*X+182ANDY<.45*X+195ANDX>210GOSUB11150:GOTO6200
3420 IFX>230ANDX<240ANDY>266ANDY<279GOSUB10500:GOTO6200
3430 IFX>235ANDX<260ANDY>262GOSUB10800:GOTO6200
3440 DR$="W":GOSUB11300:GOTO6200
3500 IFY<.4*X+296ANDY>.15*X+338ANDX>210GOSUB11600:GOTO6200
3510 IFX>286ANDY<389ANDY>361MX=294:MY=371:CC$="BC":GOSUB10500:GO
TO6000
3520 IFX>229ANDX<246ANDY<322ANDY>311GOSUB10500:GOTO6200
3530 IFY<1.21*X+35ANDY>1.25*X-5ANDY>370GOSUB11650:GOTO6200
3540 IFY>.44*X+182ANDY<.45*X+195GOSUB11150:GOTO6200
3550 IFY>=.2*X+320ORY<=.44*X+182THENDR$="N":GOSUB11300:GUTO6200
3560 IFY<=.2*X+304THENDR$="S":GOSUB11300:GOTO6200
3570 GOSUB11500:GOTO6200
3600 IFY>.81*X+328GOSUB11450:GOTO6200
3610 IFX>260ANDX<293ANDY>413ANDY<441GOSUB10600:GOTO6200
3620 DR$="SW":IFY>.44*X+377ANDY<.4*X+400ANDY<-2*X+970IFX>230MX=2
36:MY=487:CC$="AE":ZNE=2:GOSUB11100:GOTO6000:ELSEIFX<211MX=205:M
Y=475:CC$="HI":ZNE=2:GOSUB11100:GOTO6000:ELSEGOSUB11100:GOTO6200
3630 IFX>260GOSUB10800:GOTO6200
3640 DR$="W":GOSUB11300:GOTO6200
3700 DR$="SW":IFX>215ANDX<225ANDY>569ANDY<580GOSUB10500:GOTO6200
3710 IFY>.84*X+346ANDY<.99*X+339ANDY<-1.2*X+807GOSUB11100:GOTO62
00
3720 IFY<.8*X+348ANDY>.81*X+328ANDY<-X+770GOSUB11450:GOTO6200
3730 IFY<-1.8*X+948GOSUB11300:GOTO6200
3740 GOSUB10800:GOTO6200
3800 IFY>.23*X+28GOSUB11500:GOTO6200
3810 GOSUB10800:GOTO6200
3900 IFY>-.02*X+160ANDY<-.02*X+170GOSUB11050:GOTO6200
3910 IFX>321ANDX<331ANDY<121ANDY>111GOSUB11000:GOTO6200
3920 IFX>310ANDX<340ANDY>101ANDY<131GOSUB10500:GOTO6200
3930 IFX<320ANDY<.2*X+48ANDY>.23*X+28GOSUB11500:GOTO6200
3940 GOSUB10800:GOTO6200
4000 IFY<290ANDY>270ANDX>320ANDX<350GOSUB11800:GOTO6200
4010 IFX<380ANDX>365ANDY>278ANDY<290MX=369:MY=281:CC$="AA":GOSUB
10500:GOTO6000
4020 IFY<3.5*X-945ANDY>.7*XANDY>-.9*X+597GOSUB11350:GOTO6200
4030 IFY>1.5*X-170THENDR$="NW":GOSUB11300:GOTO6200
4040 GOSUB10800:GOTO6200
```

```
4100 IFX>315ANDX<325ANDY<341ANDY>330GOSUB10500:GOTO6200
4110 IFY<1.21*X+35ANDY>1.25*XGOSUB11750:GOTO6200
4120 IFY>.44*X+182ANDY<.45*X+195ANDX<370GOSUB11150:GOTO6200
4130 IFY>.3*X+285GOSUB11600:GOTO6200
4140 IFY>=.2*X+320THENDR$="N":GOSUB11300:GOTO6200
4150 IFY>.2*X+304GOSUB11500:GOTO6200
4160 IFY>=.45*X+196THENDR$="S":GOSUB11300:GOTO6200
4170 IFY>X-20THENDR$="N":GOSUB11300:GOTO6200
4180 IFY<1.2*X-125GOSUB11350:GOTO6200
4190 GOSUB10800:GOTO6200
4200 IFX>370ANDX<383ANDY>432ANDY<448GOSUB10500:GOTO6200
4210 IFY>490ANDX>380GOSUB10500:GOTO6200
4220 IFY<1.21*X+35ANDY>1.25*X-5IFY>460GOSUB10700:GOTO6200:ELSEIF
Y>450MX=356:MY=455:CC$="EE":ZNE=2:GOSUB11650:GOTO6000:ELSEIFY>41
6GOSUB11650:GOTO6200:ELSEGOSUB11750:GOTO6200
4230 IFX<351ANDY<.18*X+358GOSUB11600:GOTO6200
4240 GOSUB10800:GOTO6200
4300 IFY<510ANDX>379CC$="BB":MX=384:MY=506:GOSUB10500:IFY>502GOT
O6000ELSE6200
4310 IFY<1.21*X+35ANDY>1.25*X-5GOSUB10700:GOTO6200
4320 IFX>322ANDX<328ANDY>876-XANDY<559GOSUB11000:GOTO6200
4330 IFX>310ANDX<340ANDY>540ANDY<570MX=325:MY=560:CC$="AB":GOSUB
10500:GOTO6000
4340 IFY>538GOSUB10900:GOTO6200
4350 GOSUB10800:GOTO6200
4400 IFX>480ANDX<490ANDY>25ANDY<39GOSUB10500:GOTO6200
4410 GOSUB10950:GOTO6200
4500 IFY>145ANDY<160ANDX>405ANDX<425GOSUB10500:GOTO6200
4510 IFY>-.02*X+160ANDY<-.02*X+170ANDX<420GOSUB11050:GOTO6200
4520 IFY<X-100ANDY<170ANDX>420GOSUB10850:GOTO6200
4530 GOSUB10800:GOTO6200
4600 IFY<2.2*X-680ANDY>2.5*X-845ANDY>252ANDY<277GOSUB10500:MX=43
2:MY=255:CC$="BF":IFY<260GOTO6000ELSE6200
4610 IFY>223ANDY<248ANDX>410ANDX<441GOSUB11800:GOTO6200
4620 GOSUB10800:GOTO6200
4700 IFX<427ANDX>418ANDY>365ANDY<379MX=422:MY=368:CC$="BD":GOSUB
10500:GOTO6000
4710 IFY<.2*X+320ANDY>.2*X+306GOSUB11500:GOTO6200
4720 IFX>412ANDX<423ANDY>330ANDY<342THENDR$="NW":GOSUB11100:GOTO
6200
4725 IFY>1.4*X-270ANDY<.78*X+30ANDY<-X+822GOSUB11350:GOTO6200
4730 IFY>.25*X+265THENDR$="S":GOSUB11300:GOTO6200
4740 IFX>480ANDY>320THENDR$="W":GOSUB11300:GOTO6200
4750 GOSUB10800:GOTO6200
4800 IFY<.2*X+320ANDY>.2*X+304:IFX>450GOSUB11550:MX=450:MY=405:C
C$="EG":ZNE=2:IFX<460GOTO6000ELSE6200:ELSEGOSUB11500:GOTO6200
4810 IFX>450ANDY<444GOSUB10950:GOTO6200
4820 GOSUB10800:GOTO6200
4900 IFY<1.21*X+35ANDY>1.25*X-5IFY>534GOSUB10750:GOTO6200:ELSEGO
SUB10700:GOTO6200
4910 IFY>534ANDY<565ANDX>450ANDX<485GOSUB10500:GOTO6200
4920 IFY>534GOSUB10900:GOTO6200
4930 GOSUB10800:GOTO6200
5000 IFX>498ANDX<512ANDY>72ANDY<83MX=505:MY=80:CC$="JR":GOSUB105
00:GOTO6000
5010 GOSUB10950:GOTO6200
5100 GOSUB10800:GOTO6200
5200 IFX>510ANDX<522ANDY>211ANDY<222GOSUB10500:GOTO6200
5220 IFX>559ANDX<575ANDY>280ANDY<295MX=565:MY=287:CC$="EH":GOSUB
10500:GOTO6000
5225 IFY>1.33*X-480ANDY<1.3*X-441ANDY>-.8*X+635GOSUB10700:GOTO62
00
5230 GOSUB10800:GOTO6200
5300 IFY>1.33*X-483ANDY<1.3*X-442GOSUB10700:GOTO6200
5320 IFX>531ANDX<540ANDY>338ANDY<351GOSUB11000:GOTO6200
5330 IFX>523ANDX<550ANDY>330ANDY<358MX=530:MY=352:CC$="AC":GOSUB
10500:IFY>345GOTO6000ELSE6200
5335 IFY>-3.2*X+2046ANDY>.3*X+185GOSUB10850:GOTO6200
5340 GOSUB10800:GOTO6200
```

```
5400 IFY<.2*X+320ANDY>.2*X+304IFX<551GOSUB11550:GOTO6200:ELSEGOS
UB11500:GOTO6200
5410 IFY<450ANDX<550GOSUB10950:GOTO6200
5420 GOSUB10800:GOTO6200
5500 IFY>536GOSUB10900:GOTO6200
5510 GOSUB10800:GOTO6200
```

Input runner's reactions to new location and respond appropriately.

```
6000 IFSR=0SR=-1:GOTO6200:ELSESR=-1:PRINT@522,"YOU SIGHT A MARKE
R!!":PRINT@648,"GO TO IT OR NOT? (G/N)"
6030 A$=INKEY$:IFA$="N"THEN6200ELSEIFA$<>"G"THEN6030
6040 GOSUB10200:D=SQR((X=MX)[2+(Y-MY)[2):X=MX:Y=MY:PRINT@390,"TH
E CONTROL CODE IS ";CC$:PRINT@452,"DO YOU WANT TO PUNCH IN?":PRI
NT@536,"(Y/N)"
6080 A$=INKEY$
6090 IFA$="N"THEN6100ELSEIFA$="Y"THENTS=TS+3:CARD$=CARD$+CC$:ELS
E6080
6100 GOSUB10200:IFZNE=1GOSUB10500ELSEGOSUB10000
6130 IFLEN(CARD$)>7PRINT@448,"ENTER A TRAVEL BEARING OF 999":PRI
NT@514,"IF THIS WAS YOUR LAST MARKER"
6140 SR=0:GOTO6510
6200 SR=SR+1:ONSPGOTO6220,6230,6240,6250,6260
6220 A$=" VERY SLOW":GOTO6265
6230 A$=" SLOW":GOTO6265
6240 A$=" NORMALLY":GOTO6265
6250 A$=" FAST":GOTO6265
6260 A$=" VERY FAST"
6265 BG=TB-DER:IFBG<0BG=360+BG
6270 PRINT@514,"YOU'VE BEEN RUNNING";A$:PRINT@586,"ON A BEARING
OF";BG:PRINT@646,"AND HAVE GONE"AD%;"METERS":PRINT@772,"CONTINUE
 LIKE THIS? (Y/N)"
6300 A$=INKEY$:IFA$="N"THEN6500ELSEIFA$="P"THEN8500ELSEIFA$<>"Y"
THEN6300
6330 IFVERT<>0THENY=Y+VERT*6.562:GOTO850
6340 GOTO800
6500 GOSUB10200
6510 PRINT@642,"TYPE AND ENTER A NEW BEARING":GOTO220
```

No directional error assigned because runner is following a linear feature.

```
7000 DER=0:GOTO650
```

Print control card and end the run.

```
8000 CLS:PRINT@144,"CONTROL":PRINT@209,"CARD":PRINT@327,"FINAL";
TAB(20);TM;TAB(28);"MIN";TAB(48);"RUN":PRINT@391,"TIME";TAB(20);
TS;TAB(28);"SEC";TAB(47);"AGAIN?":PRINT@496,"(Y)"
8010 FORR=0TO3
8020 FORC=1TO10STEP2
8030 N=R*10+C:M=516+128*R+3*C:PRINT@M,MID$(CARD$,N,2):IFINKEY$="
Y"THEN90ELSENEXTC:NEXTR
8070 FORN=4TO40:SET(8,N):SET(9,N):SET(70,N):SET(71,N):IFINKEY$="
Y"THEN90ELSENEXTN
8080 FORN=22TO40:SET(21,N):SET(22,N):SET(33,N):SET(34,N):SET(45,
N):SET(46,N):SET(57,N):SET(58,N):IFINKEY$="Y"THEN90ELSENEXTN
8090 FORN=8TO71:SET(N,4):SET(N,13):SET(N,22):SET(N,28):SET(N,34)
:SET(N,40):IFINKEY$="Y"THEN90ELSENEXTN
8120 IFINKEY$="Y"THEN90ELSE8120
```

Divine guidance from Silvus.

```
8500 FORI=1TO3:IFI=1X$="R"ELSEIFI=2X$="A"ELSEX$="Y"
8510 A$=INKEY$:IFA$=""THEN8510ELSEIFA$<>X$THEN6300ELSENEXTI
8550 CLS:FORI=1TO400:NEXTI:PRINT@325,"THERE IS A SUDDEN STILLNES
S IN THE WORLD AROUND YOU...":FORI=1TO650:NEXTI:CLS
8560 PRINT@279,"YOU HEAR A VOICE":PRINT@395,"IT COMES FROM NOWHE
RE, BUT IS EVERYWHERE.":PRINT@526,"'I AM SILVUS - GOD OF ORIENTE
ERING'":FORI=1TO750:NEXTI:PRINT@597,"'YOU ARE AT";INT(X+.5);INT(
Y+.5);"'":FORI=1TO2000:NEXTI:CARD$=CARD$+"**":GOTO190
8570 FORI=1TO2000:NEXTI:PRAY=PRAY+1:CARD$=CARD$+"**":GOTO190
```

Subroutine to update accumulated time and fatigue error.

```
10000 ONSPGOTO10010,10015,10020,10025,10030
10010 F=1.8:GOTO10050
10015 F=1.5:GOTO10050
10020 F=1.29:GOTO10050
10025 F=1.13:GOTO10050
10030 F=1
10050 T%=F*MSP*D:F=SP*SP*T%/7000:TS=TS+T%:M=INT(TS/60):TM=TM+M:T
S=TS-60*M:FER=FER+F:F%=FER:PRINT@0,"ACCUMULATED TIME":PRINT@36,"
FATIGUE ERROR":PRINT@64,TM;"MIN";TS;"SEC":PRINT@112,F%
10130 ZNE=1:RETURN
```

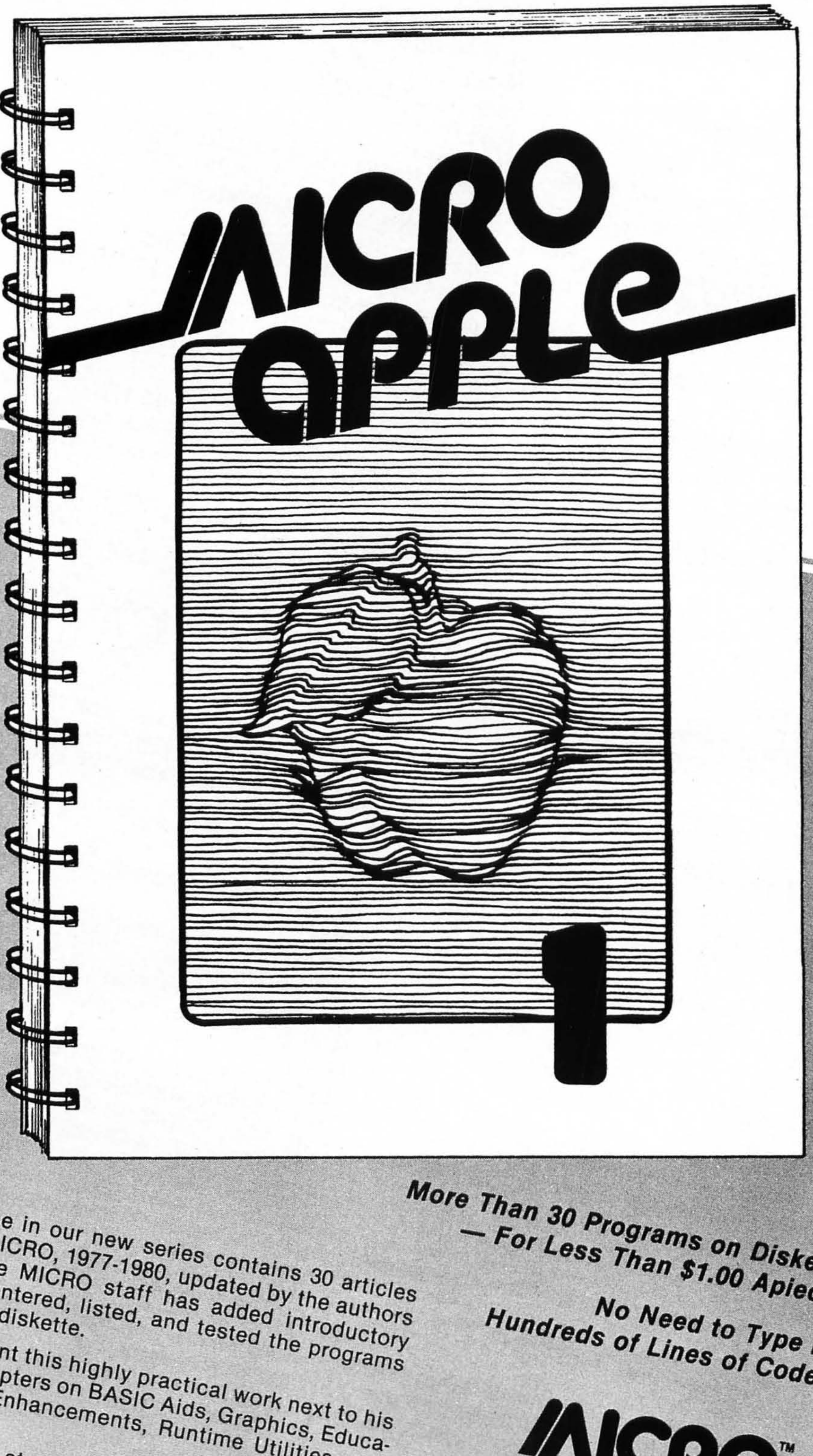
Subroutine to clear screen of all but first two lines.

```
10200 FORI=128TO896STEP64
10210 PRINT@I,"":NEXTI
10230 RETURN
```

Series of one-line subroutines to print description of runner's location.

```
10500 PRINT@260,"YOU HAVE ORIENTED YOURSELF":PRINT@328,"AT COORD
INATES";INT(X+.5);INT(Y+.5):GOSUB10000:RETURN
10600 PRINT@274,N$:PRINT@338,"BOULDER FIELD":MSP=.55:GOTO10000:R
ETURN
10650 PRINT@274,BY$:PRINT@340,"DRY STREAM":MSP=.5:GOSUB10000:RET
URN
10700 PRINT@274,O$:PRINT@340,"GRAVEL ROAD":MSP=.38:GOSUB10000:RE
TURN
10750 PRINT@274,O$:PRINT@326,"GRAVEL ROAD IN A MEADOW":MSP=.38:G
OSUB10000:RETURN
10800 PRINT@274,O$:PRINT@340,"LEVEL AREA":MSP=.45:GOSUB10000:RET
URN
10850 PRINT@274,N$:PRINT@346,"MARSH":MSP=1:GOSUB10000:RETURN
10900 PRINT@274,N$:PRINT@344,"MEADOW":MSP=.4:GOSUB10000:RETURN
10950 PRINT@274,N$:PRINT@340,"PINE WOODS":MSP=.5:GOSUB10000:RETU
RN
11000 GOSUB10500:PRINT@388,"(I SURE HOPE YOU CAN SWIM)":MSP=4:RE
TURN
11050 PRINT@274,BY$:PRINT@340,"RAIL FENCE":MSP=.45:GOSUB10000:RE
TURN
11100 PRINT@274,N$:PRINT@328,"REENTRANT - ";DR$;" IS UPHILL":MSP
=.5:GOSUB10000:RETURN
11150 PRINT@274,O$:PRINT@346,"RIDGE":MSP=.45:GOSUB10000:RETURN
11200 PRINT@274,BY$:PRINT@342,"ROCK FACE":MSP=.8:GOSUB10000:RETU
RN
11250 PRINT@274,N$:PRINT@344,"SADDLE":MSP=.55:GOSUB10000:RETURN
11300 PRINT@274,O$:PRINT@330,"SLOPE - ";DR$;" IS UPHILL"
11310 IFDR$="N"THENUB=0ELSEIFDR$="NE"THENUB=45ELSEIFDR$="E"THENU
B=90ELSEIFDR$="SE"THENUB=125ELSEIFDR$="S"THENUB=180ELSEIFDR$="SW
"THENUB=225ELSEIFDR$="W"THENUB=270ELSEDB=315
11320 DB=ABS(TB-UB):IFDB>180THENDB=360-DB
11330 IFDB<90MSP=.55-DB/45*.05ELSEMSP=.45
11340 GOSUB10000:RETURN
11350 PRINT@274,O$:PRINT@340,"SMALL HILL":MSP=.55:GOSUB10000:RET
URN
11400 PRINT@274,O$:PRINT@324,"STEEP SLOPE - ";DR$;" IS UPHILL":M
SP=.8:GOSUB10000:RETURN
11450 PRINT@274,O$:PRINT@346,"SPUR":MSP=.45:GOSUB10000:RETURN
11500 PRINT@274,BY$:PRINT@344,"STREAM":MSP=.5:GOSUB10000:RETURN
11550 PRINT@274,BY$:PRINT@328,"STREAM IN A PINE WOODS":MSP=.5:GO
SUB10000:RETURN
11600 PRINT@274,N$:PRINT@334,"THICK BRUSH AREA":MSP=2.1:GOSUB100
00:RETURN
11650 PRINT@274,O$:PRINT@346,"TRAIL":MSP=.4:GOSUB10000:RETURN
11750 PRINT@274,O$:PRINT@330,"TRAIL IN THICK BRUSH":MSP=.4:GOSUB
10000:RETURN
11800 PRINT@274,N$:PRINT@324,"FIELD OF SMALL DEPRESSIONS":MSP=.5
:GOSUB10000:RETURN
```

$24.95 With diskette!



MICRO/Apple 1

This first volume in our new series contains 30 articles selected from MICRO, 1977-1980, updated by the authors or our staff. The MICRO staff has added introductory material and re-entered, listed, and tested the programs and put them on diskette.

Every user will want this highly practical work next to his Apple, with its chapters on BASIC Aids, Graphics, Education, Games, I/O Enhancements, Runtime Utilities, and References.

Get MICRO/Apple 1 at your local computer store.

224-page book and diskette $24.95*

More Than 30 Programs on Diskette — For Less Than $1.00 Apiece!

No Need to Type in Hundreds of Lines of Code!

MICRO™

P.O. Box 6502
Chelmsford, MA 01824

** On mail orders, add $2.00 for surface or $5.00 for air shipment.*

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.

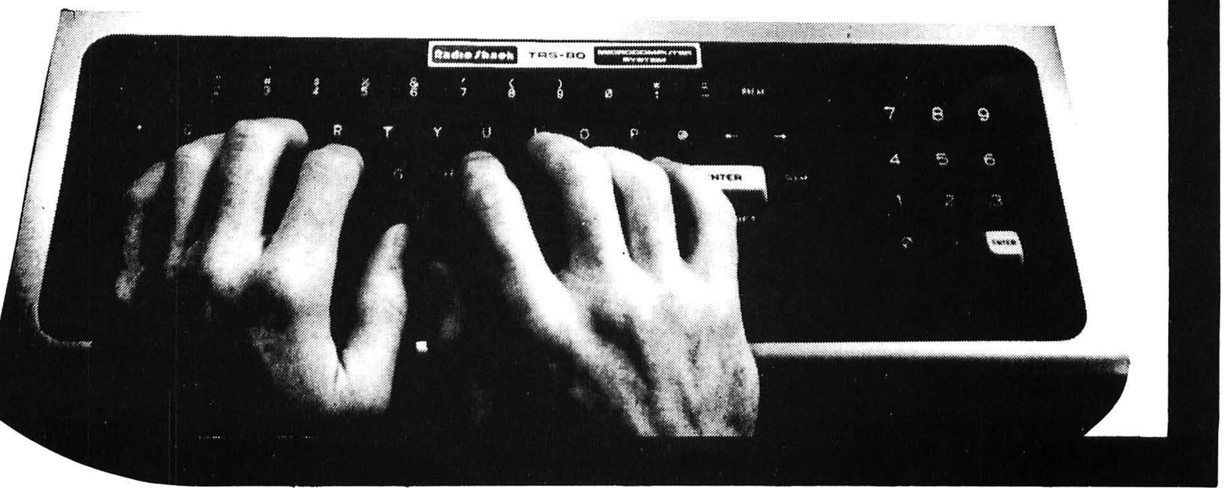
SoftSide: S-80, and APPLE Back Issues (magazine only) $2.50
PROG/80 Back Issues $4.00
Magazine with Cassette programs $9.95
Magazine with Disk programs $14.95

New Super SoftSide Back Issues
(Magazine Only)
August or September, 1980 $3.00
October, 1980 to present $3.50

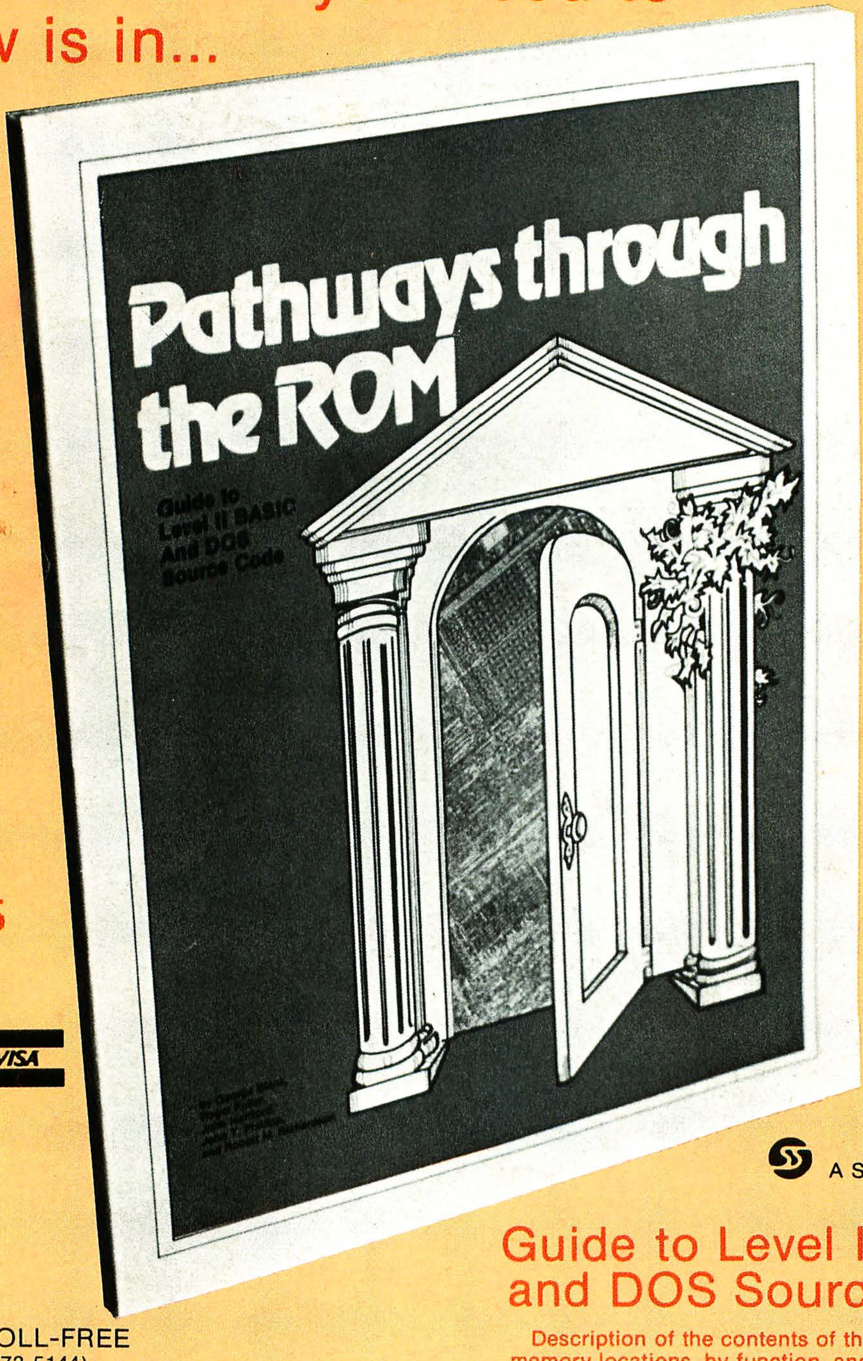
SoftSide™
cassette & diskette

MasterCard VISA

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



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 Phillipp
Monitor written in BASIC

Z-80 DISASSEMBLER
by George Blank

ALL
ONLY
$19.95
plus $1 shipping

MasterCard VISA

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