SoftSide™

Now for the IBM®-PC, too!

#35

THREE DOLLARS

HOPPER

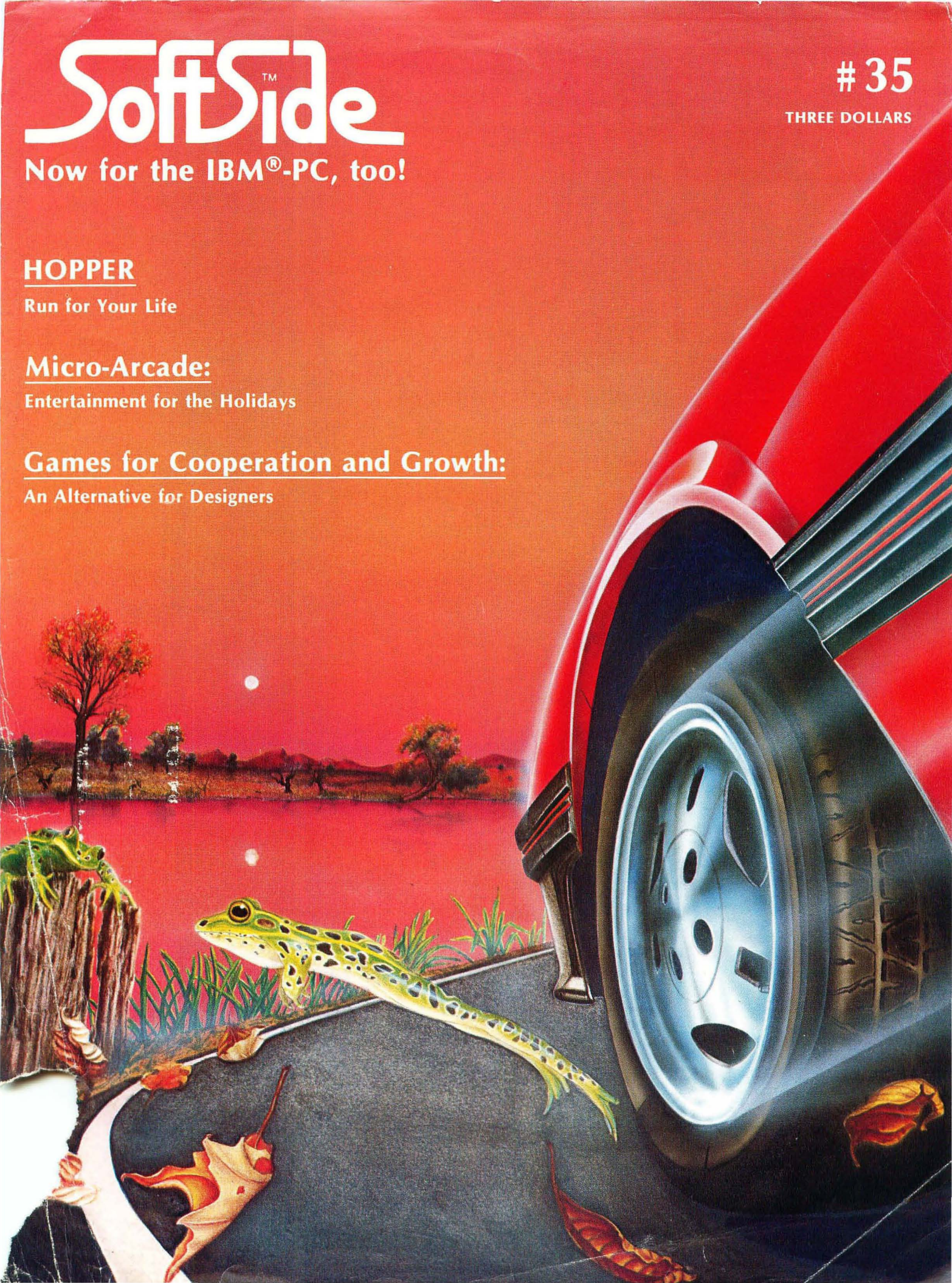
Run for Your Life

Micro-Arcade:

Entertainment for the Holidays

Games for Cooperation and Growth:

An Alternative for Designers



A SPEECH SYNTHESIZER ON A DISK!

That's why Apples and Ataris are saying: "Talk Is Cheap"

IT'S CALLED THE SOFTWARE AUTOMATIC MOUTH, S.A.M. FOR SHORT

It's a high quality speech synthesizer created entirely in software. You use it as a software utility, load it into RAM, and then use your machine as usual, except now you can make your programs talk. It generates the speech sounds on demand, so there is no limit to what it can say.

When you hear S.A.M., you'll probably agree that it sounds better than all the hardware speech synthesizers for Apple or Atari computers. And, it has a truly remarkable price.

YOU CONTROL INFLECTION, PITCH AND SPEED

With its user-variable inflection, S.A.M. can accent words on the right syllable and emphasize the important words in a sentence.

You can also make S.A.M.'s speech higher or lower, and faster or slower, over a wide range of settings.

USE EASY PHONETIC INPUT OR PLAIN ENGLISH TEXT

S.A.M. understands a simple phonetic spelling system, not a mysterious alpha-numeric code. S.A.M. helps you learn phonetic spelling by showing you your mistakes, and the owner's manual gets you started with an English-to-phonetics dictionary of 1500 words. So it's easy to make S.A.M. produce exactly the sounds you want.

But suppose you want to type ordinary English, or you want your machine to read a word processor file aloud. The S.A.M. disk comes with RECITER, an English text-to-speech conversion program that lets S.A.M. speak from plain English text.

ADD SPEECH TO YOUR PROGRAMS WITH EASE

In a BASIC program, you add speech with just a couple of commands. In a machine language program, it's just as easy. S.A.M. comes with four demonstration programs to show off its distinctive features and help to write your own talking programs. Write adventure games with talking characters, educational programs that explain aloud, or utilities with spoken prompts — put your imagination to work.

S.A.M.

S.A.M. programmed by Mark Barton.

You can order S.A.M. directly from DON'T ASK. Add $2.00 for shipping and handling to your check or money order (or order C.O.D.)

S.A.M. for the Apple II/II+
Includes an 8-bit digital-to-analog converter and audio amplifier on a board. Only **$124.95**
Requires 48K, disk. (S.A.M. takes up to 9K; RECITER 6K.) You will also need a speaker.

S.A.M. for the Atari 400/800
S.A.M. talks through your television speaker. No additional hardware is required. Only **$59.95**
Requires 32K, disk. (S.A.M. takes up 9K; RECITER 6K.) Note: to produce the highest quality speech, automatically blanks the scree vocal output; the display is pr S.A.M. can talk with the scree the speech quality is reduced.

DON'T ASK COMPUTER SOFTWARE

2265 Westwood Boulevard, Suite B-15
Los Angeles, California 90064
Telephone: (213) 397-8811

Hear S.A.M. at your favorite computer store today!
Dealer inquiries welcome.

OF ALL THE THINGS YOU BUY, HOW MANY ARE GOOD ENOUGH TO BE WARRANTED 5 YEARS?

Few disks stand the test of time. Because few are built to the precision standards or certified to the critical levels of Omni's complete line.

Each Omni disk is rated for 12 million passes without disk-related errors or significant wear. Each is certified error-free at a minimum of twice the error-threshold of your system. And built to exceed all industry specifications including those of ANSI, ECMA, ISO and virtually every drive manufacturer. So you can count on them for the long haul. We guarantee it.

Call toll-free (800 343-7620) for your nearest dealer. In Mass., call 617 799-0197.

Omni Resources, 4 Oak Pond Ave., Millbury, Mass. 01527

Dealers. Software houses.
Check our prices, services and specifications. We offer duplicating, formatting, private labeling, small minimums, fast delivery and copy protection schemes on disks for virtually any system.

OMNI

THE DISK GOOD ENOUGH TO BE WARRANTED 5 YEARS

SoftSide™

Season's Greetings

EDITOR-IN-CHIEF
Randal L. Kottwitz

SOFTWARE EDITOR
Bill Kubeck

PROGRAMMING STAFF
Rich Bouchard
Alan J. Zett
Fred J. Condo
Kerry Shetline

EDITORIAL ASSISTANT
Joyce Smith

CONTRIBUTING EDITORS
Cary Bradley
Fred D'Ignazio
Sheldon Leemon
Lance Micklus
Mark Pelczarski
Allen L. Wold

ART DIRECTOR/
PRODUCTION MANAGER
Lynn Wood

PRODUCTION STAFF
Lynda Fedas
Denise Chartrand

PUBLISHER
John G. Grow

ASSOCIATE PUBLISHER
Nancy Lapointe

ADVERTISING
Sue Rowland
Bob Mackintosh
Christopher Smith

CUSTOMER SERVICE
Cindy Schalk

DEALER SALES
Kathie Maloof

STAFF
Jeffrey Garrod
Donna Jean
Karen Lawrence
Doris Miller
Cindy Zawacki

FOUNDER
Roger W. Robitaille, Sr.

SoftSide Vol. 6, No. 2

Cover illustration by William A. Giese

SUBSCRIPTION INQUIRIES should be sent to ***SoftSide*** Publications, 100 Pine Street, Holmes, PA 19043. EDITORIAL AND ADVERTISING CORRESPONDENCE should be sent to ***SoftSide*** Publications, 6 South Street, Milford, NH 03055. Telephone (603) 673-0585.

SoftSide (ISSN 0274-8630) is published monthly by ***SoftSide*** Publications, Inc., 6 South Street, Milford, NH 03055. Second class postage paid at Milford, NH, Avon, MA, and at additional mailing offices. Subscription rates: US, and Canada, $30/12 issues. First Class US, APO, FPO, Mexico, $40/12 issues. Other foreign countries, $62/12 issues. Media subscription rates: US Magazine and Cassette, $75/12 issues. US Magazine and Disk $125/12 issues. APO, FPO, Canada and Mexico, add $20/12 issues. Other foreign add $50/12 issues. All remittances must be in U.S. funds. POSTMASTER: Please send form 3579 to ***SoftSide*** Publications, 100 Pine Street, Holmes, PA 19043.

FRONT RUNNER

28
Hopper
by Howard Wolkow
Translations by Rich Bouchard, Kerry Shetline and Alan J. Zett
You poor little frog! You've lost your way and can't find your way home. When you finally see it, a busy highway and river stand in your way.

FEATURES

14
Entertainment Tomorrow
by Allen L. Wold
Have you ever fantasized about driving a race car or exploring the moon's surface? With a *telefactor device* your fantasies may soon come closer to reality.

18
Games for Cooperation and Growth — An Alternative for Designers
by Peter J. Favaro
The pros and cons of computer and video games are a hot topic in all of the media lately. Mr. Favaro offers an alternative to electronic game design which may quell the critics and make the games a positive social force.

22
The Phantom Programmer
If you've ever tried to get your computer to stop doing what you told it, you'll appreciate the Phantom's plea for NEVER MIND.

24
Calc/Side
by David Peters
In this first of an ongoing series for home applications of VisiCalc®, the author covers the spreadsheet calculator's solution for balancing the checkbook.

DEPARTMENTS

CONTENTS

PC/SIDE

TRS-80®/SIDE

ATARI®/SIDE

APPLE™/SIDE

EDITORIAL

The Gift Of A Lifetime

by Randal L. Kottwitz

The snow wafts gently to the ground in New England as the family gathers for the holidays. The glittering ornaments on the tree remind us of Christmases past and we give thanks for all we've been blessed with in our lives. We at ***SoftSide*** have much for which to be thankful — especially the extended family of readers with whom we share our lives.

As we contemplate this season of giving, all of us consider those things of most value to us and determine how they can be shared with others on this planet. The community of computerists has a priceless gift to give — that of time. Knowledge which puts us "ahead of our time" or makes more time available to us is something which can be shared, for it can be communicated to those around us and bring them to the same understanding we hold so dear.

Our gift of time is *computer awareness*. Every citizen of western civilization is affected by computers on a daily basis. And yet, how many of those people have any understanding of this powerful effect on their lives? More importantly, how many of them realize how close computers are to an intimate participation in their home lives or what control of time they'll be offered through these new tools? These are propositions which have become commonplace to those of us who are in frequent contact with computer keyboards. But, they are foreign and frightening predictions to most of the world.

We have had the opportunity to become *computer aware* — to realize that these blinking and whirring machines are nothing more than complex tools to be applied as we see fit. Armed with that realization, we have the opportunity to share it with the larger community of man. Not many of us can consider giving ten microcomputers to the library or even one to the high school. However, there are many other gifts of computer awareness to be had.

Perhaps the library is in need of books on computer science but has spent its budget in that area for the year. Maybe the high school purchased a lab full of computers last year, but neglected to budget the appropriate money for software purchases (regrettably, not an uncommon occurrence). These are golden opportunities for those of us with smaller pocketbooks who wish to contribute something to the computer awareness of our society.

There is a specific gift I would like to suggest to organizations looking for an effective gift for their community. The BBC created a series of programs on computer awareness called *The Computer Programme*. This charming series of ten programs explains each major function of the computer through an appropriate analogy. All of the information is presented in plain English, not computer jargon. My initial reaction to previewing the series was that I'd like to get the entire American public in front of their TV sets for five hours and show them these brilliant programs. Of course, that's not possible, but the series is available on video tape from Films, Inc. in Wilmette, Illinois. The price is not low ($1700 and up), but the benefits to viewers of the programs can carry no price tag.

No gift of computer awareness can carry an accurate price — for its benefits cannot be accurately measured. We are only now discovering the socioeconomic impact of the computer on our lives, negative as well as positive. The more minds we bring to computer awareness, the better our chances will be of accentuating the positive impact and eliminating the negative. As the minds of our young geniuses exploit this tool to its fullest extent, who knows what avenues will be opened to our civilization. Your gift of computer awareness may be the key to one of those minds. Isn't that truly the gift of a lifetime?

A joyous holiday season to you from all of us at ***SoftSide***!

Randal L. Kottwitz

Editor-in-Chief

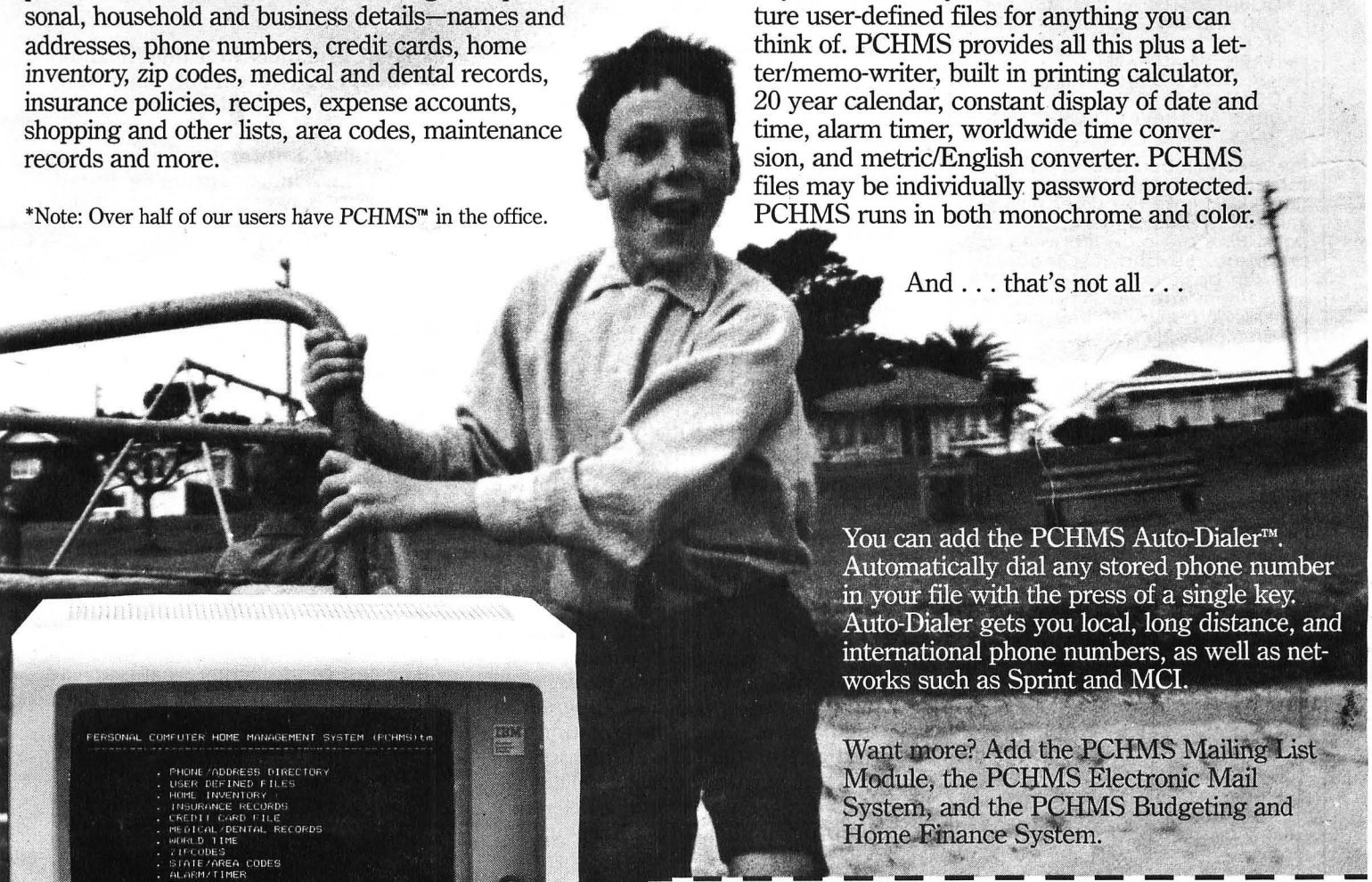
Integrated, Menu-Driven File And Information System for Home or Office* ...with PCHMS™, our Personal Computer Home/Office Management System.

Instantly put your IBM PC to work! PCHMS is a filing and information system that manages all those details that make your office, home and personal life smooth and efficient. Organize personal, household and business details—names and addresses, phone numbers, credit cards, home inventory, zip codes, medical and dental records, insurance policies, recipes, expense accounts, shopping and other lists, area codes, maintenance records and more.

*Note: Over half of our users have PCHMS™ in the office.

PCHMS is an integrated menu-driven system that allows you to add, delete, modify, or print out records in any file by full or partial key search. And you can create 10 full-feature user-defined files for anything you can think of. PCHMS provides all this plus a letter/memo-writer, built in printing calculator, 20 year calendar, constant display of date and time, alarm timer, worldwide time conversion, and metric/English converter. PCHMS files may be individually password protected. PCHMS runs in both monochrome and color.

And . . . that's not all . . .



You can add the PCHMS Auto-Dialer™. Automatically dial any stored phone number in your file with the press of a single key. Auto-Dialer gets you local, long distance, and international phone numbers, as well as networks such as Sprint and MCI.

Want more? Add the PCHMS Mailing List Module, the PCHMS Electronic Mail System, and the PCHMS Budgeting and Home Finance System.

PCHMS™ runs on an IBM Personal Computer with 64K of memory, at least one disk drive (single or double sided). Printer optional. PCHMS Auto-Dialer requires a Hayes Stack Smartmodem and RS232 card.

Ask for PCHMS™ and PCHMS Auto-Dialer™ at your IBM PC dealer or order directly from Arlington Software + Systems. NOW AVAILABLE AT COMPUTERLAND STORES.

Dealer inquires invited.

PCHMS software is supplied on single sided diskettes together with comprehensive user documentation, backup utility and configuration program. Satisfaction Guaranteed!

☐ please send me _____ PCHMS @ $89.95 _____

☐ please send me _____ PCHMS Auto-Dial @ $69.95 _____

☐ please send me _____ PCHMS Mailing List @ $59.95 _____

Shipping and handling $ 3.50 _____

MasterCard and Visa welcome, please add 4% _____

Name: _____

Address: _____

City: _____ State: _____ Zip: _____

Phone: (_____) _____ Card Exp. Date: _____

Charge Card # _____

Signature: _____

97 Bartlett Ave. Arlington, MA 02174
(617) 641-0290

ARLINGTON Software + Systems

INPUT/OUTPUT

From our readers

INPUT

ATARI® GTIA

Dear *SoftSide,*

I recently learned of a new display processor that Atari uses in their 400/800 systems. I am an owner of an Atari 400 and thought other readers of *SoftSide* who own an Atari system would like to know about this new development.

I understand that the older processor is called CTIA and the new one, GTIA. The new GTIA chip gives three more graphic modes — 9, 10 and 11, and is used for 3D graphics. I enjoy working with Atari graphics and would like to know more about this new display processor.

I received issue #33, *Computer Graphics*, and noticed that the CTIA and GTIA chips were briefly discussed in the *Fliker* article and program. I would appreciate any information you could give on the GTIA processor.

Your magazine is excellent. The programs are great and your articles very interesting and helpful. Keep up the good work!

Rich Luce
Orange, CT

Editor's Reply: The GTIA chip has brought many changes to the field of Atari graphics. Machines manufactured after January, 1982, carry the GTIA as standard equipment. If your older 400/800 is still under warranty, the upgrade is free of charge at your nearest authorized Atari Service Center. If not, the modification will cost about $60 and is well worth it. When possible, SoftSide will publish its Atari programs with modifications for the GTIA — eventually supporting only the GTIA in order to fully utilize the three new graphics modes. Watch for further information concerning CTIA/GTIA in an upcoming installment of *Exploring The Atari Frontier.*

Mangled Covers

Dear *SoftSide,*

My latest issue of *SoftSide* just arrived in horrible shape. It was greatly mangled by the Post Office. Please send me another.

I would say your magazine is of such caliber and cost that it justifies sending it in a brown wrapper to protect it. This is not the first time my friends and I have received dog-eared copies of *SoftSide* — but this is by far the worst.

The cost of protecting what is an excellent publication will be more than offset by the happiness of the computerists who recieve it.

Perry C. Goldstein
Westminster, CO

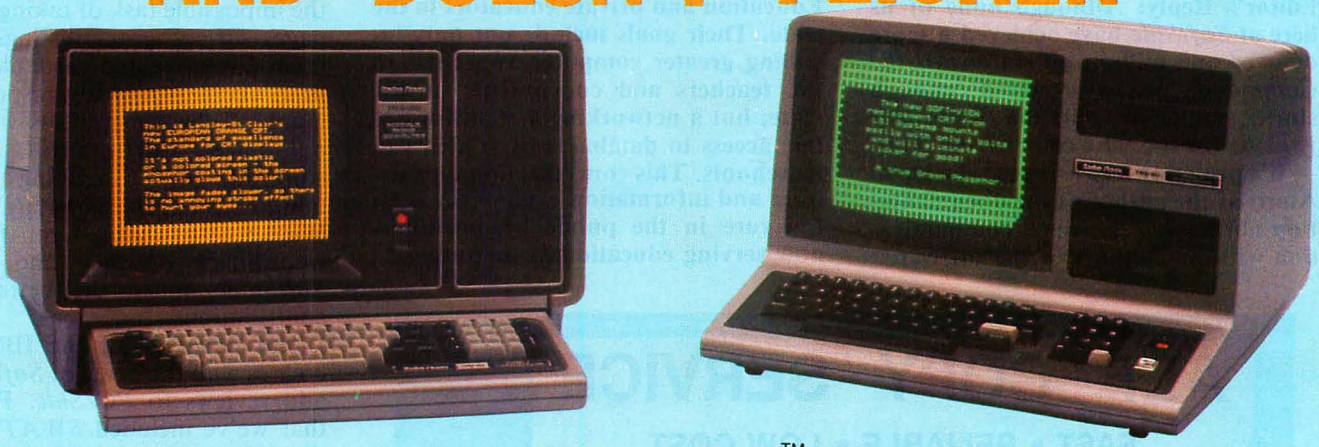
Editor's Reply: We've received many complaints of the post office handling of your magazines over the last year and they've not been ignored. Until very recently, the small circulation of SoftSide has made the cost of wrapping the magazine prohibitive. However, our circulation is growing by leaps and bounds — enough so that you will be receiving protected copies of SoftSide in the very near future.

continued on page 8

Convert your TRS-80* into a World Class Computer that REDUCES EYE FATIGUE AND DOESN'T FLICKER

ALSO.... UNIQUE RED & BLUE CRT'S AVAILABLE NOW!

NOW ALSO AVAILABLE FOR MODEL I AND OTHER 12" MONITORS



...with LSI's new Soft-View™ Replacement CRT...

The black & white "TV Screen" CRT (picture tube) which came with your TRS-80*model II or III is an inexpensive rapid "P4" Phosphor CRT intended for TV use. The display is actually strobing 60 times a second. No amount of "green plastic" will stop this strobing or eliminate the eye fatigue it causes. But a new Soft-View™ CRT display tube with a slower decaying, colored Phosphor will.

- Available in slow-decay green (similar to new IBM* and APPLE III*monitors) or medium decay "European Orange" (easy on the eyes, elegantly beautiful, and the standard for CRT displays in Europe)
- Leaded glass stops X-ray emission
- Optional Anti-Glare Frosted Glass available to reduce eye strain from glare
- Easy installation – tube comes with pre-mounted hardware
- 30-Day Money-Back Guarantee
- Ideal for Word-Processing & Programming, fast enough for Games & Graphics
- Finest quality double-dark glass and phosphor fields make the letters seem to be coming out of black space

Try This Test:

LSI's new Soft-View™ CRT

Turn the brightness control on your TRS-80*all the way up. Wave your hand up and down in front of the screen. See how jerky it seems? Just like in front of a strobe light! That's because the screen actually is strobing at you. A slower-phosphor CRT will reduce that troublesome strobe effect. That's why most of the newer monitors, from IBM* to Apple III* are using the new slow-phosphor CRT's.

LSI SYSTEMS Soft-View™ CRT's: 80M-1182

- ☐ #GN42 Green Phosphor $79.95
- ☐ #GN42G Green Phosphor with anti-glare $89.95
- ☐ #OR34 Orange Phosphor $89.95
- ☐ #OR34G Orange Phosphor with anti-glare $99.95

ADD $7 FOR PACKAGING AND UPS SHIPPING

Langley-St.Clair Instrumentation Systems, Inc. ⌐462

To Order Call: 1-800-221-7070
Dealer Inquiries Invited

132 West 24th Street, New York, N.Y. 10011 212-989-6876

IBM, APPLE* and TRS-80* are trademarks of IBM, APPLE Computer & TANDY Corp.

Frustrated Arcader

Dear *SoftSide,*

I read the *Input* section of ***SoftSide,*** May, 1982, and found sympathy with Buzz Taylor. After reading his letter, I decided one day to try for a million points using the "smart bomb" option on Atari's *Missile Command*. Well, today was the day. Atari's claim that *Missile Command* will display points up to one billion is false. I had a score of 999,930. I hit a "smart bomb" and my score changed to about 1760. How can Atari get away with those false claims? None of my friends believe that I scored over a million points!

Darren Layne
Sewaren, NJ

Editor's Reply: Although none of us here at *SoftSide* have achieved a score of over one million points on *Missile Command,* we've received enough stories similar to yours that *we believe you*! As for the false claims from Atari — it's probably a case (not exclusive to Atari) of the author of the documentation not having thorough communication with the authors of the software.

Educational Networking

Dear *SoftSide,*

I've been running a network in Stowe, Vermont, that connects the elementary and secondary schools in the area. Vermont has recently founded a non-profit organization called VECTOR for information exchange, teacher in-service continuing education, and development of legislative action. Software will be forthcoming from Vermont's teachers.

David Gibson
The Stowe School
Stowe, VT 05672

Editor's Reply: In a brief phone conversation with Mr. Gibson, we got a little more background on VECTOR. The organization is a cooperative effort of the Vermont Department of Education and private educators in the state. Their goals include not only instilling greater computer awareness in the teachers and constituents of the state, but a networking system for online access to databases by students in its schools. This concentration on software and information availability is all too rare in the public organizations now serving educational computing.

OUTPUT

by Randal L. Kottwitz

"Do you think the frog should make a *boink* or a *ribut* sound when he jumps?"

"Oh, I don't know. How about *boink* when he jumps and *ribut* if he gets run over by a car?"

This conversation actually took place as ***SoftSide*** 35 was being constructed. I walked away with my head in my hands, pondering the name of the alien planet to which I had been beamed when I wasn't looking. *Hopper* is the result. We hope the game will provide you hours of enjoyment during the holiday season.

We've received quite a few letters and phone calls lately, complimenting us on the new "look" of ***SoftSide***. Much of ***SoftSide***'s facelift has been the responsibility of our art director, Lynn Wood. Monthly, Lynn confronts the impossible task of taking pages and pages of program listings, making them attractive *and* readable. In addition, she is personally responsible for many of the illustrations which grace our pages. All of this, while the editorial staff is rushing down last minute changes and late material. We'd like to thank Lynn for her extraordinary patience and talent — ***SoftSide***'s much the better for her contributions.

Our coverage of the IBM® PC increases with this issue of ***SoftSide*** as we add reviews to *PC/Side.* Please note that we've included *SWAT* tables for the PC programs in ***SoftSide*** 34 immediately following the line listing of *SWAT*. *DV* for the PC is now available on a single copy or subscription basis. The disks currently carry only the programs printed in ***SoftSide***, but will soon have a bonus program as well as we swing our PC software production into full gear.

We've gotten many software submissions lately which do not include the author's phone number. In many cases, we must contact the author of a program for a modification or repair of a bug. Without a phone number, this process can take vital weeks as letters go back and forth. Please be sure to include your phone number on any correspondence with our editorial offices.

Michel Stuart, producer of the Broadway musical *Nine*, gave nine wishes to the American public when he accepted the Best Musical Tony award for the show. We pass them on as our wishes for you in 1983 — Health, Happiness, Humility, Love, Passion, Devotion, Perseverance, Courage and Serenity. Happy Holidays!

Heartland Software
Holiday Specials

Crush, Crumble And Chomp!!! 32K $21.95

Choplifter 48K Disk $23.95

Bandits 48K Disk $24.95

We Stock

Broderbund	APX
Synapse	PDI
Avalon Hill	JV Software
On-Line	Strategic Simulations
Datasoft	Automated Simulations
Big-Five	Adventure International
Sirius	

Happy Holidays from The Heartland

apple® ATARI® TRS-80®

Title	List	Our Price	Title	List	Our Price	Title	List	Our Price
Action Quest	29.95	23.95	Frogger	34.95	25.95	Ricochet	19.95	15.95
Adventure 12-Pack	129.95	100.00	Galaxy Invasion	15.95	12.95	S.A.G.A. 1-12	39.95	29.95 ea.
Andromeda Conquest	23.00	18.95	Genetic Drift	29.95	23.95	Sammy The Sea Serpent	16.95	13.95
Apple Panic	29.95	23.95	Graphics Composer	39.95	29.95	Scarfman	19.95	15.95
B-1 Nuclear Bomber	16.00	12.95	Ghost Encounter	29.95	23.95	Serpentine	34.95	27.95
Battle of Shiloh	39.95	29.95	Invasion Orion	24.95	18.95	Sea Fox	29.95	23.95
Claim Jumper	34.95	27.95	K-Razy Shootout	49.95	37.95	Shamus	34.95	27.95
Commbat	24.95	19.95	Labyrinth	29.95	23.95	Shooting Arcade	29.95	23.95
Convoy Raider	16.00	12.95	Legionnaire	35.00	27.95	Slime	34.95	27.95
Cosmic Balance	39.95	29.95	Micropainter	34.95	27.95	Softporn Adventure	29.95	23.95
Crossfire	29.95	23.95	Mission: Asteroid	24.95	18.95	Space Eggs	29.95	23.95
Curse of Ra	19.95	15.95	Moon Base IO	29.95	23.95	Star Blazer	31.95	24.95
Cyclod	29.95	23.95	Morloc's Tower	19.95	15.95	Star Warrior	39.95	29.95
Datestones of Ryn	19.95	15.95	Mousekattack	34.95	27.95	Stellar Escort	15.95	12.95
Deadly Secrets	34.95	27.95	Midway Campaign	16.00	12.95	Stellar Shuttle	29.95	23.95
Deadline	49.95	37.95	Nautilus	34.95	27.95	Super Nova	15.95	12.95
Defense Command	15.95	12.95	OK Galaxy	20.00	14.95	Tanktics	24.00	18.95
Dnieper River Line	30.00	24.95	Pacific Coast Hwy	29.95	23.95	Temple Of Apshai	39.95	29.95
Dodge Racer	34.95	27.95	Protector	34.95	27.95	Text Wizard	99.95	75.95
Dog Daze	22.95	19.95	Preppie	29.95	23.95	Tigers In The Snow	39.95	29.95
Eastern Front	29.95	23.95	Reptilian	34.95	27.95	Track Attack	29.95	23.95
Fort Apocalypse	34.95	27.95	Rescue At Rigel	29.95	23.95	Visicalc	250.00	199.95

641-5055

(216)

Indicate type of computer, disk or cassette.
For fast delivery, send Certified Check or Money Order.
Personal checks require two weeks to process.
C.O.D. orders add $1.50.
Master Charge or Visa orders add 3%
(Include all embossed information on card)
Add $2.00 shipping and handling.
Prices subject to change.

Make Checks and Money Orders payable to:
HEARTLAND SOFTWARE DISTR.
P.O. Box 25517
Cleveland, Ohio 44125
Ohio Residents add 6.5% Tax
(216) 641-5055
Order Lines Open 10 a.m. - 9:00 p.m. Mon.-Fri.
Noon - 6:00 p.m. Sat.

Send $1.00 for complete catalog

From our readers

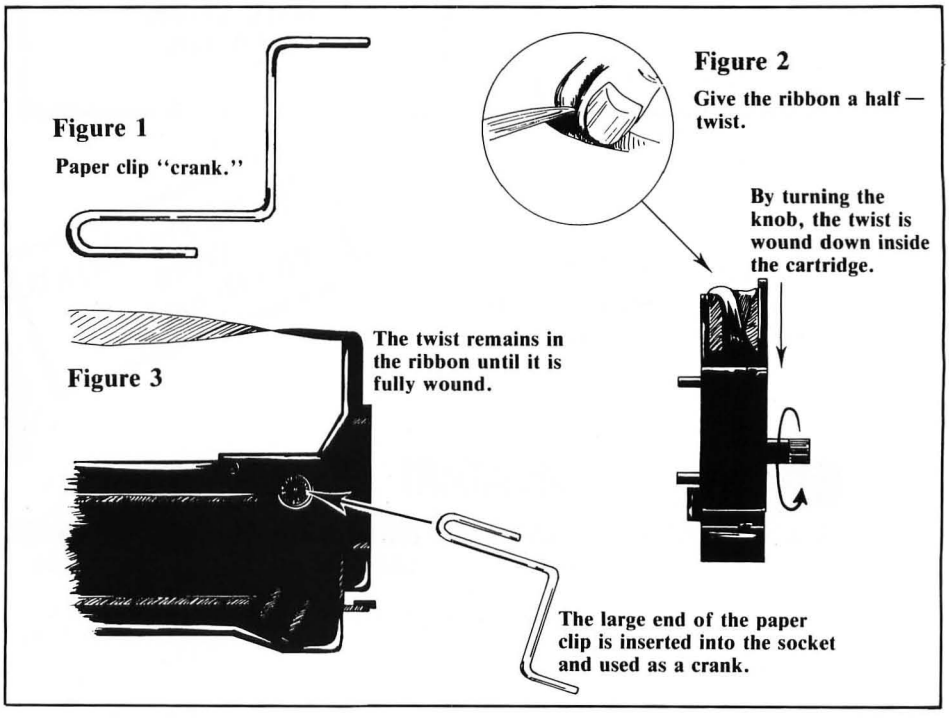
More Mileage from Epson Ribbons

It is possible to extend the life of Epson MX-70 and MX-80 ribbon cartridges by flipping the ribbon over in the cartridge. Since the print head strikes off-center on the ribbon, flipping the ribbon exposes a fresh surface. Here is a simple technique for doing so.

Take an ordinary paper clip and bend it as shown in Figure 1. Lay this aside, within easy reach. Now take the ribbon cartridge and refer to Figure 2. Lift the ribbon at the end near the ribbon advance knob and give the ribbon a half-twist. Turn the advance knob a few turns to feed the twisted ribbon into the cartridge through the advance mechanism. At this point, the ribbon cartridge should look as it did before, except for a half-twist in the exposed portion.

Take the paper clip crank you made earlier and insert the loop end into the ribbon advance socket on the cartridge. This is on the bottom of the cartridge, opposite the advance knob as in Figure 3. Now use the crank to advance the ribbon until the half-twist inside the cartridge comes out the other end. That's all there is to it. The ribbon is now flipped over.

The thing that makes this idea practical is the paper clip crank. Without the crank, you would have to advance the ribbon with the knob. Since there are twenty yards of ribbon in the cartridge, this could take a long time and the hard little knob would wear your fingers off. With the crank, however, it takes no more than two or three minutes to wind the entire ribbon.



Apple *Defense* Enhancement

Here are some enhancements for Apple *Defense* (***SoftSide*** Issue 33).

Swap lines 4123 and 7007. This makes the music more appropriate for the situation.

Change line 11000 as shown. This improves the opening banner page.

```
11000  TEXT : HOME : FLASH : HTAB
     16: PRINT "DEFENSE!": NORMAL
     : PRINT : PRINT : PRINT "
     EVIL ALIENS FROM BEYOND EAR
     TH HAVE INVADED THE MOON.":
     PRINT : PRINT
```

Add the following lines to display the keyboard controls at the start of each game.

```
11035  HTAB 10: INVERSE : PRINT "
     KEYBOARD CONTROLS: ": NORMAL
     : PRINT
11037  HTAB 12: PRINT "A,Z = UP,
     DOWN"
11038  HTAB 6: PRINT "<-,-> (REPE
     ATED) = GND SPEED"
11045  HTAB 10: PRINT "SPACEBAR F
     IRES LASER"
```

Scott C. Seely, D.V.M.
Portsmouth, NH 03801

Death Star Joystick Modifications

After playing *Death Star* for a short while, I felt that the game could use a few enhancements. The way that the program was set up made it easy to make the change from keyboard input to joystick input.

```
403 P=PEEK(764):IF P=10 AND D=1 AND X=
R AND Y=1 AND SD=0 THEN 1100
405 STR=STRIG(0):IF STR=0 THEN 800
406 Z=STICK(0):IF Z=15 THEN 402
420 IF Z<>11 THEN 430
430 IF Z<>7 THEN 440
440 IF Z<>14 THEN 480
480 IF Z<>13 THEN 490
488 P=PEEK(764)
495 P=PEEK(764):IF P=10 AND D=1 AND X=
R AND Y=1 AND SD=0 THEN 1100
815 ? D$(D*5-4,D*5):? "PRESS FIRE":IF
SD=1 THEN POKE 656,3:POKE 657,2:? "COU
NTDOWN:";
824 FOR AAA=1 TO 70:NEXT AAA
825 IF STRIG(0)=0 THEN 300
830 GOTO 820
```

I feel that the changes make the game seem faster. For a game written in BASIC, it runs fast, indeed. I really enjoy your magazine and I am looking forward to submitting a few programs.

Gary Smith
Cape May, NJ 08204

WIN $5,000 Plus Royalties!*

For the best Talking Game

For the Atari® or Apple®II+

Using the VOICE BOX™ Speech Synthesizer

Now you can make your Atari® 400/800 or Apple® II games and other programs come alive with the VOICE BOX by the Alien Group — the first low-cost, smart speech synthesizer with unlimited vocabulary.

Add jokes to your programs. Insults. Compliments. Help messages. Stories. Alien voices. Animal roars. Have your computer talk to the fire department or police in emergencies. To kids. Or blind people. Teach touch typing with immediate spoken feedback. Or just about any other subject — the fun way. Or help a speech-impaired friend communicate . . . the possibilities are limitless.

The VOICE BOX plugs into your Atari's serial port. And talks directly through your TV set. Or into any Apple II slot. No power supply or special interfaces needed.

Just select from its simple screen menu. A dictionary with thousands of common words (on diskette or cassette) automatically translates your text into speech. It's that easy.

But don't let its friendliness fool you. The VOICE BOX has all 64 phonemes (basic sounds, like "ah") built in. So you can precisely create any word or sound you can imagine. And store it all on diskette or tape. Names or foreign language words, for example. Or wierd non-human languages.

Let me entertain you — The VOICE BOX is creative too. It will crack you and your friends up with non-stop random, grammatically correct sentences, using words you specify. It also has an amusing talking face with lip-sync animation — a real crowd-stopper. Best of all, you can call the VOICE BOX from any BASIC program and make your program really hum —literally!

Singing Apples? — Apple owners get all these capabilities too — as a plug-in card plus diskette. Or there's a deluxe version with the dictionary in ROM (no diskettes to bother with), speaker, and ability to "sing" (hey we're not making this up folks) in any key. (Both Apple versions require 32K or more. Applesoft and DOS 3.3).

Don't confuse the VOICE BOX with "dumb" speech synthesizers that can't learn new words. Or software-based ones with lower speech quality — and an annoying tendency to blank out the display when they talk. The VOICE BOX is a true breakthrough in speech synthesis. Small wonder thousands of Atari and Apple owners have already bought the VOICE BOX.

The VOICE BOX is available now at leading computer stores throughout the world. Or direct from the Alien Group, with 10-day money back guarantee if you're not completely satisfied.

VOICE BOX For Atari. $169.00
16K and 32K versions included
(Specify diskette or cassette).

VOICE BOX for Apple II.+ $139.00.
(Requires speaker.)

VOICE BOX for Apple II+. $215.00
(Includes dictionary in ROM and singing capability. Comes with speaker.)

Enclose check or money order.

CHECK YOUR LOCAL COMPUTER STORE FOR TALKING "VOICE BOX" VERSIONS OF YOUR FAVORITE GAMES FROM LEADING GAME COMPANIES

*Win a $5,000 prize — plus royalties — for the best Atari 400/800 or Apple II+ game using the VOICE BOX. Deadline: May 30, 1983. Write for contest details.

Please mail to: The Alien Group, Department , 27 W. 23 St., N.Y., N.Y. 10010
Or call in order to (212) 741-1770

Bugs, Worms, and other Undesirables

ATARI® *Neatlist*

To make *Neatlist* work with cassette, delete line 250. Change the following:

```
230 POSITION 16,12:INPUT FILE$
270 POSITION PP,10:PRINT "HAS ";FILE$;
" BEEN LISTED TO TAPE?";:POSITION 16,1
2:INPUT RESPON$
290 GRAPHICS 0:POSITION PP+2,10:? "YOU
 MUST LIST ";FILE$;" TO TAPE"
400 POKE 764,12:OPEN #1,4,0,"C:":? CHR
$(125);
```

Apple™ *Defense*

The instructions for the Apple translation of *Defense* (*SoftSide,* issue 30) did not properly explain the game controls.

With a joystick, move the lever right to speed up your ship, and left to slow it down. Move the lever up and down to move the ship up and down. Press button 0 to fire the laser.

With keyboard control, press A to move the ship up, Z to move it down. Use the left arrow to slow the ship and the right arrow to speed it up. Press the spacebar to fire.

SHRCG Fix

There is a problem in the Apple *Hi-res Character Generator* program (*SoftSide DV,* issue 33) when the STR$ function is used. The character set pointer is contained in locations 254 and 255, which, according to the *Apple Reference Manual*, are not used by Applesoft. However, it seems that STR$ uses location 255 as a temporary register. This will cause all sorts of bizarre garbage to be printed instead of the proper characters. The problem may be solved by re-POKEing location 255 to the proper value after the use of a STR$. The better solution, however, is to modify the code to use a different pointer. Make a backup of your SHRCG file, then use the following program to fix it.

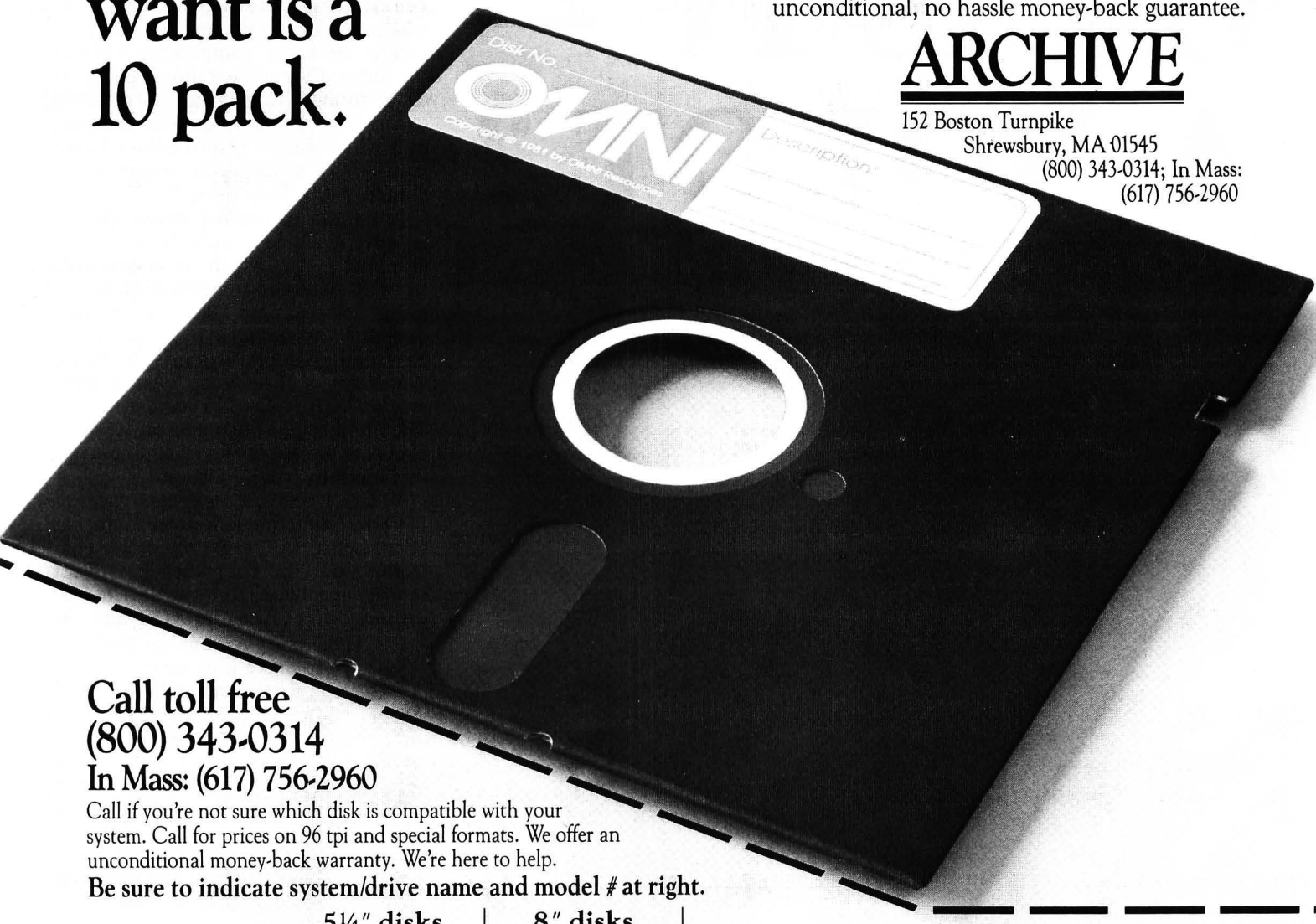
```
10 D$ = CHR$ (4): PRINT D$"BLOAD
    SHRCG,A$2000": IF PEEK (82
    17) = 255 THEN END
20 FOR X = 8219 TO 8248: GOSUB 3
    0: NEXT X: FOR X = 8364 TO 8
    422: GOSUB 30: NEXT X: PRINT
    D$"BSAVE SHRCG,A$2000,L$F4":
    END
30 V = PEEK (X): IF V < 252 OR X
    = 8381 THEN RETURN
40 IF V > 253 THEN POKE X,V - 2
    : RETURN
50 POKE X,V + 2: RETURN
```

Now the character set pointer will be set using POKE 252 instead of POKE 254, and POKE 253 instead of POKE 255. These POKEs should be changed in the FONT EDIT program in lines 110,460,580, and 590 (there are two POKEs to be changed in line 590).

Get Omni quality for as little as $1.99...

even if all you want is a 10 pack.

Call toll-free for great savings on Omni's complete line of 5¼" and 8" premium disks. Each is certified error-free at a *minimum of* twice the error threshold of your system. Each is rated for more than 12 million passes without disk-related errors or significant wear. And each is precision fabricated to exceed all ANSI specifications with such standard features as reinforced hub rings and Tyvec sleeves. Get same day shipment and an unconditional, no hassle money-back guarantee.



ARCHIVE

152 Boston Turnpike
Shrewsbury, MA 01545
(800) 343-0314; In Mass:
(617) 756-2960

Call toll free
(800) 343-0314
In Mass: (617) 756-2960

Call if you're not sure which disk is compatible with your system. Call for prices on 96 tpi and special formats. We offer an unconditional money-back warranty. We're here to help.

Be sure to indicate system/drive name and model # at right.

	5¼" disks		8" disks		
	Cost per 10 pack	Quantity	Cost per 10 pack	Quantity	Total Cost
Single side/single density	$19.90	_____	$24.90	_____	$_____
Single side/double density	$23.90	_____	$31.90	_____	$_____
Double side/single density	----	_____	$34.90	_____	$_____
Double side/double density	$37.50	_____	$37.50	_____	$_____
Flip/Floppy reversible	$39.90	_____	$39.90	_____	$_____
Plastic library case (in lieu of soft storage box)	$ 2.99	_____	$ 3.49	_____	$_____
Shipping and handling ($2.00 first 10 pack, 40¢ additional 10 packs. Continental U.S. only.)					$_____
5% sales tax (Mass only)					$_____
				Total	$_____

☐ Check ☐ Master Card
☐ COD ☐ VISA

Card # _____ Exp._____

System/drive model # _____

Name _____

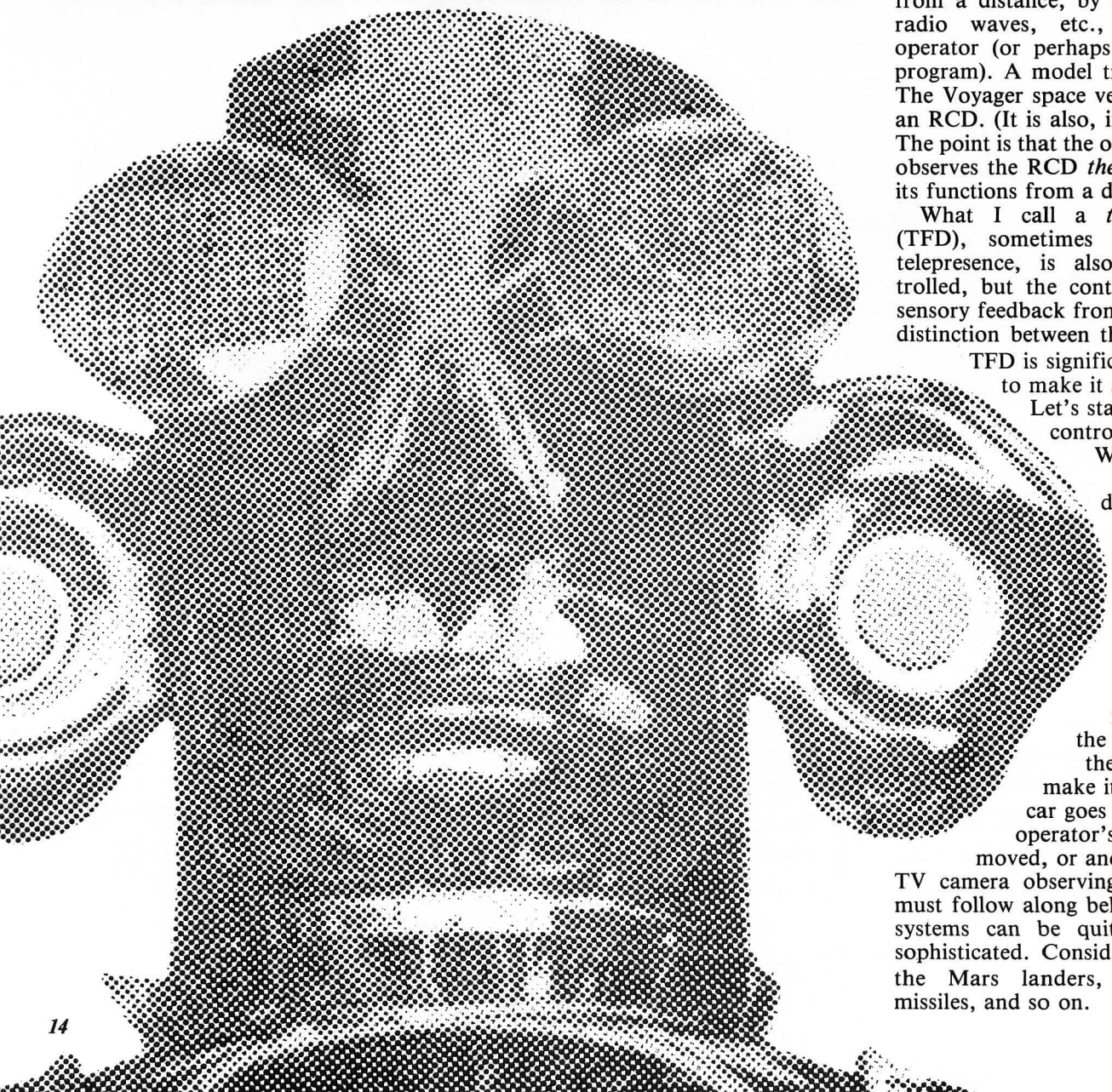
Address _____

_____

Tel. _____

IS IT REAL OR IS IT — A ROBOT ?

by Allen L. Wold



In the world of robotics, the techno-language can get very confusing. Let me start with a few definitions: Robot, Remote Controlled, and Telefactor Device (a term invented by Roger Zelazny).

A *robot* is any machine which is "programmed," by means of cams, gears, computer programs, or whatever, to do a specific task, of greater or lesser complexity, without the supervision of a human being or other form of intelligence. A wind-up toy is a robot. An automatic stamping mill which makes record player based plates is a robot — it works unattended.

A *remote controlled device* (RCD) is any machine which is not programmed, but which is controlled, from a distance, by means of wires, radio waves, etc., by a human operator (or perhaps by a computer program). A model train is an RCD. The Voyager space vehicle is, in part, an RCD. (It is also, in part, a robot.) The point is that the operator sits *here,* observes the RCD *there,* and controls its functions from a distance.

What I call a *telefactor device* (TFD), sometimes referred to as telepresence, is also remotely controlled, but the controller has direct sensory feedback from the device. The distinction between the RCD and the TFD is significant, so I will try to make it as clear as I can. Let's start with a remote controlled automobile. We have a regular car, to which devices have been added allowing the operator to control speed and direction from a distance. The operator sits at a station, the car is on the road. As long as the operator can see the car, he/she can make it perform. If the car goes out of sight, the operator's station must be moved, or another car, with a TV camera observing the RCD car, must follow along behind. Such RCD systems can be quite complex and sophisticated. Consider, for example, the Mars landers, wire-controlled missiles, and so on.

Now, add a TV camera to the car, mounted where the driver would normally sit. It can look straight ahead or to either side. The operator is equipped with a helmet which "remotely" controls the camera. As the operator's head turns, so does the camera. Surrounding the operator's field of vision is a TV screen which shows what the camera sees. The operator's control station also has a speed indicator, fuel gauge, and so on. A steering wheel and foot pedals, remotely controlling those devices, are in the car itself. In fact, the control station looks just like the driver's seat of the car.

Sophisticated flight trainers, used by the military and commercial airlines, include the ability to make the pilot-trainee feel the simulated motions of the aircraft — if the plane banks, the cockpit simulator tips, and so on. We will have the same capability in our TFD control station. If the car hits a bump, the operator, perhaps miles away, can feel it, as the sensors in the car transmit the data to the station.

Thus, when the operator is in the station, it is as if he/she were actually sitting inside the car, though in fact the TFD car may be anywhere. This is why this concept is sometimes called "telepresence," because you seem to be present somewhere other than your actual physical location. (I have heard, but I do not know, that some military aircraft are controlled in this way.)

The military advantages are obvious. A directly controlled fighter plane, for example, puts the pilot's life in jeopardy every time it is taken out. The TFD plane, however, risks only the equipment, not the pilot. If it is shot down, the pilot lives to fly another.

The significant difference between RCD and TFD is sensory feedback from the telefactor device. Computers, of course, play a large part in this idea, providing accurate simulation, communication, control, remote feedback, and so on.

I do not wish to discuss military or industrial applications. In fact, to even begin the list of possibilities would take hundreds of pages. Instead, I would like to concentrate on just a few recreational possibilities, without getting too involved in complicated TFDs, such as humanoid constructs. I will restrict myself primarily to the kinds of devices available today, but with the addition of telefactor control.

So you want to be a race car driver, but haven't the time or energy to make racing your career. You can drag on the streets and get yourself arrested or killed, or you can rent a TFD racer. You and the others participating have been rated for skill and "crash index," so that you won't be competing unfairly. The cars themselves may be regular cars, racing cars, or special cars designed to withstand accidents. You sit in your control seat, you see the track and the other cars, and can race to your heart's content, knowing that nothing can happen to you.

Let's take this a step further. Lots of people attend demolition derbies but, without considerable skill and a suitable car, dare not drive in one. You could get killed that way. Suppose our TFD car is designed to be extraordinarily rugged, so as not to suffer any real, permanent damage, but at the same time be temporarily disabled, according to how it hits or is hit by another car. Such an experience might not be cheap, but what fun! You and a dozen other lunatics would compete on a concrete field, with your weird vehicles, crashing into each other, the last one moving declared the winner — all at no personal risk.

"With telefactor control, one could explore environments otherwise highly hazardous to one's health, such as the interior of a volcano, the depths of the sea, the surface of the moon..."

With telefactor control, one could explore environments otherwise highly hazardous to one's health, such as the interior of a volcano, the depths of the sea, the surface of the moon — now, there's an idea.

Let's install, on the moon, a number of sturdy, well-designed crawlers. Instead of hiring professional astronauts to physically travel to the moon to explore, we sell tickets to amateurs who will never leave the Earth. Everything they see is recorded, and they could conduct a number of experiments. There would be, of course, the problem of the several second time-lag for the signals to go from the earth-based controls to the TFD crawlers and back. Nonetheless, there is no other way most of us would get to see the moon first hand.

There are great possibilities, even if we are restricted to something as prosaic as an ocean reef. In order to see such a reef in person, one must learn the art of scuba diving, purchase or rent equipment, and get there by plane or boat. Our TFD diver will not interest those who would rather do it themselves, and who have the time and money. For the rest of us, however, what an opportunity. Our device is like a miniature submarine, possibly no bigger than a go-cart. It swims through the water, at any depth, for any length of time. There is no need to worry about "the bends," when coming up from the ocean depths. There would be no danger from fire coral, sharks, or sea urchin spines.

Learning to fly a plane is time consuming and expensive. But consider this. Most commercial airliners practically fly themselves. The pilot still needs considerable training and experience, but the plane is so computer-assisted that most of what the pilot has to concern himself with is judgement, not the mechanics of flying a plane. The computer controls the take-off, landing, flight, and so on. Without the pilot, it couldn't get started, but without the computer, it would go nowhere.

A small airplane could be built in which most of the work of flying is taken care of by the computer. Push a button — the plane takes off. In time, this is probably what will happen with real planes anyway, just as, at some time in the future, all highways will be computer controlled, leaving the driver with nothing to do but punch in his/her destination, then sit back and enjoy the view. So, it's easy to imagine a small plane that requires only a minimum amount of skill to fly.

Even so, unskilled pilots would probably kill themselves. But with telefactoring, you or I could climb into a simulation cockpit, and not have to worry about falling out of the sky. The plane could be made very small, too small to carry a person. Thus, with your telecontrol, you could fly a relatively inexpensive plane using proportionately little fuel.

You could do acrobatics, or race, knowing, first, that the computer would help you out of trouble, and,

secondly that even if that didn't work, you were in no physical danger (though you would be billed for a crashed plane). Right now, we dare not let amateurs fly acrobatic planes. But with telefactoring, it would be possible.

In California, certain dry lakes, covering several square miles, have been set aside for model rocketeers. Planes are not allowed to fly over them. Motorcycles cannot race there. (They have their own dry lakes to mess up however they want.) The same could be done for TFD planes. Ground control radar would keep the planes from straying beyond the boundaries of the lake, and communications between planes could keep them from colliding in mid-air.

There, for a few bills, you could take one up, and fly to your heart's content in perfect safety. With a properly designed control-cabin, which reproduced the attitude and feel of the plane itself, it would be almost as good as being there. Planes could be designed for special recreational purposes, such as acrobatics, super-slow flight, speed for racing, high altitude with telescopic observation, and so on. I'm ready for it.

A very popular board game by Metagaming Concepts is *Ogre,* with a number of sequels, additions, and a host of articles in the gaming magazines on special variations. In this game, a group of tanks, ground effect vehicles, self-propelled howitzers, and infantry take on a kind of super-mega-tank, called an Ogre.

Imagine this. Instead of little pieces of cardboard, or even CRT images in a computer, you could have telefactored miniature tanks and vehicles — or an Ogre. No real weapons would be used, but, instead, laser beams which trigger photo-receptors. (See *Light Sabers and Lasers, SoftSide,* February, 1982.) You could climb into your telecontrol cabin, and it would be just like being there. You could direct your tank, communicate with the others on your side, patrol the bizarre terrain, but let the computer do the actual work of driving the tank. You would spot your enemy, and engage it in combat. The possibilities of such a game are endless, subject only to the whim and ingenuity of the designer. Let's examine it a little further.

The game's objective is usually to capture the enemy's stronghold, eliminate the enemy's forces, control a certain amount of territory, or get from point A to point B.

In our telefactored implementation, we will not limit ourselves to extant weapons systems, but model them after those found in *OGRE.* There is a tank, with caterpillar treads, a turret mounted gun and machine gun (simulated by weak lasers). It is slow, but heavily armored. There is a self-propelled gun, with less armor, less mobility, but capable of heavier damage and longer range. There is a ground effect vehicle (GEV, also the the name of a sequel to *OGRE*), which is lightly armored, lightly armed, very fast, and capable of skimming over water. There is the chopper, (a helicopter) without armor, but with machine gun or missile weapons (which explode with lots of smoke but no damage). It is very fast and capable of long range observation. And there are Ogres, combining the power of several self propelled guns and the mobility of a tank with super heavy armor. One Ogre is usually equivalent to five or as many as ten tanks, and frequently fights alone on its side, opposed by a raft of other, smaller enemy vehicles.

These need not, of course, be full sized. In fact, vehicles the size of a go-cart would be suitable, as well as cheaper to build. They would incorporate the durability of our demolition derby cars, and the general computer assisted control of our airplanes. Plus, of course, as you operated a vehicle, it would be as if you were inside looking out.

The terrain could be anything — a desolate wasteland, a forest, a mock-up city, simulation of the craters of the moon, or bizarre and distorted landscapes from the covers of science fiction books.

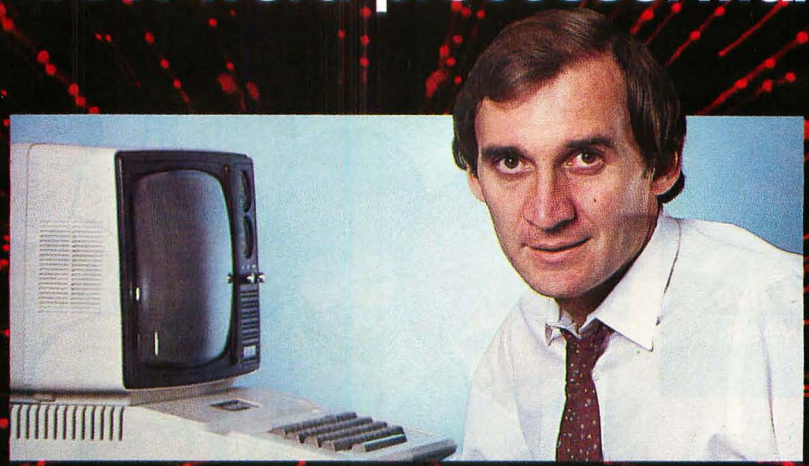
At the present, the technology to provide the appropriate vehicles for any of these "sports," and the ability to provide the telecontrol, has not been developed though it is by no means impossible. We have considered only the simplest of telefactoring devices, avoiding the complex machines which could explore caves, simulate humanoid boxers or football players, and so on. The mere possibility of such devices would generate new ideas for new sports, never conceived of before. If you have any ideas along this line, don't keep them to yourself, share your imagination with me and the world of *Entertainment Tomorrow.*

"Simply the best word processor...anywhere"

For the Apple™

No hardware additions needed — What you see is what you get. All functions are displayed on the screen exactly as they appear in print including:

- Underlining
- Bold
- Superscript
- Even/normal justification
- Lower and upper case
- Block movement
- Global replace
- Plus many more features



"Word Handler is simply the best word processing software I could find anywhere by far."
William R. Moroney
President
Electronic Funds Transfer Association

Now Available!
List Handler
A mailing list program to work with the Word Handler (interfaces with VisiCalc™ and DB Master™) stores up to 3000 records per disk, unlimited sorting fields.

Word Handler

Once you buy a Word Handler we don't forget you; our customer service department is available daily. We support our products.

When upgrades are introduced you are supplied with a replacement disk FREE! Contact your local Apple™ dealer for a demonstration. You'll be glad you did.

Silicon Valley Systems, Inc. 1625 El Camino Real #4 Belmont, CA 94002 (415) 593-4344

Silicon Valley Systems

GET SERIOUS . . .

Uncompromised design delivers superior quality and reliability. Todays latest technology allows your Atari 400 to run up to 50% cooler and provide truer video clarity. We guarantee it.

So let's get down to business.

48K RAM for the ATARI 400

Tara Computer Products INC.®

Send certified check or money order. Visa and Mastercard welcome. N.Y. residents please include sales tax. Dealer inquiries invited.
Atari is a registered trademark

$199 INTRODUCTORY OFFER

ALSO AVAILABLE

Atari	32K RAM Expansion	$109.00
Atari	16K RAM Expansion	89.00
Apple	16K RAM Expansion	99.00

Now formally announcing the industry's first internal direct replacement full travel keyboard for the Atari 400. Direct ribbon cable replacement provides for easy no solder end user installation. Another first from the innovative leader in Atari expansion products all for less than $200.

USA: STATLER BUILDING 107 DELAWARE AVE
SUITE 752 BUFFALO, N.Y. 14202 TEL: (716) 832-0661
CAN: 2 ROBERT SPECK PARKWAY, SUITE 1540
MISSISSIAUGA, ONT. L4Z 1H8 TEL: (416) 273-6820

"The game design philosophy outlined here is intended to aid programmers in developing games which facilitate positive social contact between the players."

Games for Cooperation and Growth:

An Alternative for Designers

by Peter J. Favaro

During this holiday season, parents will spend millions of dollars so their children can happily blast away aliens, munch on goblins, conquer planets and otherwise defend themselves and others in video game fantasyland. As one advertisement dramatically puts it, "The battle for the galaxy has reached the homefront..." Some of these parents will plop down 25 to 50 of their hard-earned dollars on game cartridges or computer software with hopes of rolling up their sleeves, grabbing their favorite joysticks and sharing in the vicarious thrills that delight their children. Others, however, will spend their money with ambivalence and concern, yielding not to a vociferous and perhaps somewhat overzealous group of concerned parents and social scientists who have staunchly criticized video game play, but to the whines and howls of their children, who want what "everyone else" has.

The video game controversy began about two years ago, long after the arrival of the first coin-operated arcade game, *Pong*. A group of angry parents and community leaders in Mesquite, Texas sought, and subsequently succeeded, to ban video game arcades because of their supposed disruptive influence on the behavior of the town children. Other towns and even whole countries (the Philippines and more recently, Singapore followed suit, claiming that the games cause aggression, truancy, "psychological addictions" akin to gambling, and encourage stealing money from parents and others to support children's video game habits. The concerns spread from the arcade games to the home games. Social scientists, i.e. psychologists, psychiatrists and anyone else who had an opinion, jumped on the video game bandwagon and engaged in heated debates, on talk shows across the nation, over the relative consequences and/or benefits of video game play. Most of the debates seem like outlandish parodies of the notions we hold about people in the mental health professions, with the most bizarre comment alluding to the fact that many of the games serve as substitutes for "adolescent masturbatory activity." To date, none of these self-proclaimed experts have presented even a shred of replicable scientific evidence to support their speculations. The media eats all of this up, of course, hyping the controversy which today is as strong, if not stronger, than ever.

I am a school psychologist, a doctoral candidate in clinical psychology and an educational computing consultant. I began doing video game research almost three years ago, long before it was fashionable for social scientists to be concerned about such things. My doctoral dissertation is a major research effort designed to test whether video games cause changes (either positive or negative) in either mood or ability to do certain tasks which require concentration, memory, etc. It will also observe whether the games cause increases in physiological arousal (as measured by heart rate and blood pressure) that may be potentially dangerous, as some have suggested. After three years of work, I refer to the studies I have completed as "exploratory investigations," choosing to save my opinions and speculations for discussions over wine and cheese. The reason for this is that in science, things must be proven over and over again for them to be accepted phenomena. Even the so-called hard sciences, like physics, rely on replicability to prove theories so that there can be certainty that the results are not due to chance factors. The point that I will make

here, and stress throughout this article, is that much research must be done before anyone can say something definitive — good or bad — about video game play.

I will briefly present both sides of the existing video game controversy and the shortcomings of both the pro and con side of the debate. I will discuss some of the work I have done and then present an alternative philosophy to video game design. I will definitely not resolve the issue of whether the current video games are harmful or not, as this remains to be seen.

The alternative design philosophy presented here is called *GCG* design philosophy, *Games for Cooperation and Growth.* This month, I will present the basics of *GCG* design. In a later issue, I will demonstrate the concept by providing an example of a game which employs it.

Video Game Criticisms

People who oppose video game play generally do so on two grounds: (1) that they are addictive, and (2) that they cause inappropriate social behavior. Aside from the fact that there is no evidence to support either of these comments, from a technical point of view they are both illogical and incorrect. Technically, no behavior can be addictive, because the word, "addiction," technically describes a physical dependence on an ingested substance. In general , the words "addiction" and "additive" are confusing and vague—they are not even used in the current clinical diagnostic criteria for mental disorders.

The real concern should be whether or not the games can become targets for compulsive behavior (behavior which is unstoppable because of an uncontrollable urge to perform it). The answer to this question is "yes," and the reason has nothing to do with the games themselves. Individuals who develop compulsions will always find a target for their habits. Food is frequently a target for certain compulsive people, as are work and many other behaviors. In view of this, however, we do not seek to close down supermarkets, ban cakes and chocolate bars, etc. This would be an absurd thing to do. Not everyone who eats chocolate bars and ice cream does so to the extent that it makes them dangerously obese. Some children, I believe, may be honestly and clinically obsessed with *Pac-Man,* and may have a compulsive need to play the game. These children, however, were bound to have had emotional problems before the game was ever invented. The bottom line is that each parent bears the responsibilities of regulating and directing their child's behavior. This includes all kinds of recreational activity, not just video game play.

That the games cause aggression in children also remains to be seen. Parallels have been drawn which compare video game play to watching television. It has been well known that watching violence on television may increase violent behavior in children, so people believe that because some video games have aggressive themes, the same is true. This may or may not be so. A major distinction between video game play and television watching is that T.V. watching is a passive activity, while video game play is an active activity. What difference does that make? Maybe none. Hopefully, my doctoral research, which compares people who play video games to people who watch television, will give us a clue. In the meantime, I would like to mention that a fundamental principle of psychotherapy with children assumes that aggression, acted out in socially appropriate ways during play, is helpful in teaching children to deal with their aggression and their conflicts. Again, we must wait until the research is in to prove such a notion.

I would like to make one general observation regarding children's play activities. Violence and aggression have been major themes in child's play for decades. Think for a moment about the themes, behaviors and roles involved in childhood games such as Cowboys and Indians, Cops and Robbers, even Tag and Hide and Go Seek, all of which involve either stalking, catching, shooting, killing or fighting.

Pro Video Game Arguments

Supporters of video game play have made similar outlandish assumptions. Many have claimed that the games increase eye-hand coordination. This is simply not true, and won't be until there is a good deal of evidence to support it. The first research I conducted on video game play studied whether video game play increased eye-hand coordination in children. The results were not significant. This does not mean that they *couldn't* increase eye-hand coordination if used in a different way than the way I used them. It just means that in the population I studied, I failed to find this effect. One possible reason for this is that, in order to show that transfer from one task to another exists, the task you train people to do must be similar to the task in which you hope to see transfer occur. Here is an example:

Let's say I am going to develop a sports training experiment. I want to train two groups to play different sports activities and then see which group transfers best to the criterion I have set up — racquetball skills. I will train group A at basketball, and group B at tennis. Which group would you predict does better at the criterion task? Group B, of course, because tennis involves skills which are similar to racquetball. Now, could I then say that tennis training improves "sports skills?" Not really, because the generalization would be too broad. There are many sports which tennis training wouldn't improve. The point here is that "eye-hand coordination" is very complex, and that the criteria which we use to measure eye-hand coordination may not involve the same kinds of skills needed to play video games.

"Using GCG design philosophy in video games does not require any drastic change in programming techniques. As a matter of fact, any intermediate level programmer with a sensitivity for cooperation and togetherness could easily develop these games."

GCG — An Alternative

Besides attacking the logic used by my colleagues to either bolster or banish video game play, I have not done much in the way of constructive criticism. This is a bad habit which most scientists learn very early in their careers. Let me mend my ways by offering an alternative video game design philosophy called *Games For Cooperation And Growth. GCG* design philosophy was spawned by my recent experiences with thirty learning disabled and emotionally disturbed children with whom I worked as a school psychologist. My task was to teach these children appropriate social behavior — concepts such as sharing, helping, good sportsmanship, etc. I achieved my goals using video games, specifically the Atari® VCS and several of the cartridges. I did not let the children play in the usual way. I modified both the game tasks and the

Lyco Computer Marketing & Consultants

TO ORDER CALL US

TOLL FREE 800-233-8760

In PA 1-717-398-4079

December ATARI SPECIALS

810 Disk Drive ... $429.00
32K RAM ..... $ 79.00
400 32K RAM ... $349.00

800 48K... $609.00

ATARI®

A Warner Communications Company

PERCOM : In Stock

Single Drive	CALL
Dual Drive	CALL
(Read all Atari Disks)	

PRINTERS : In Stock

Epson Mx 80	$449.00
Epson Mx 80 FT III	$499.00
Okidata 82A	$479.00
Okidata 83A	$719.00
Okidata 84	$1089.00
Citoh	CALL
Prowriter I	$499.00
Prowriter II	CALL
SMITH CORONA TP-1	$625.00
NEC	CALL
(Interfacing Available)	

JOYSTICKS : In Stock

Atari CX-40	$18.00
LeStick	$34.00
Wico Command Control	$24.00
WICO RED BALL	$27.95
STICK STAND	$ 6.75

Computer Covers

800	$6.99
400	$6.99
810	$6.99

DISKETTES : In Stock

Maxell MD1 . . .(10)	$34.00
Maxell MD2 . . .(10)	$44.00
Elephant . . .(10)	$21.00

THIRD PARTY SOFTWARE ATARI PROGRAM EXCHANGE

Eastern Front 1941	$25.50
Avalanche	$15.50
Outlaw/Howitzer	$15.50
Dog Daze	$15.50
Wizard of War	$31.00
Gorf	$31.00
Frogger	$26.00

BUSINESS SOFTWARE : In Stock

Atari Word Processing	$109.00
Letter Perfect	$129.00
Test Wizzard	$ 89.00
Datasam/65	$125.00
Interlisp	$125.00
Monkey Wrench	$ 42.00
Utility Disk	$ 36.50
Ultimate Renumber	$ 15.50

ATARI HARDWARE

410 Cassette Recorder	$75.00
825 Printer	$585.00
830 Phone Modem	$149.00
850 Interface	$164.00

PACKAGES

CX481 Entertainer	$69.00
CX482 Educator	$125.00
CX483 Programmer	$49.00
CX494 Communicator	$325.00

SOFTWARE

CXL4012	MISSILE COMMAND	$28.75
CXL4013	ASTEROID	$28.75
CXL4020	CENTIPEDE	$32.75
CXL4022	PACMAN	$32.75
CXL4011	STAR RAIDER	$34.75
CXL4004	BASKETBALL	$26.75
CXL4006	SUPER BREAKOUT	$28.75
CXL4008	SPACE INVADER	$28.75
CX8130	CAVERNS OF MARS	$31.75
CX4108	HANGMAN	$12.75
CX4102	KINGDOM	$12.75
CX4112	STATES & CAPITALS	$12.75
CX4114	EUROPEAN COUNTRIES	$12.75
CX4109	GRAPHIT	$16.75
CX4121	ENERGY CZAR	$12.75
CX4123	SCRAM	$19.75
CX4101	PROGRAMMING I	$19.75
CX4106	PROGRAMMING II	$22.75
CX4117	PROGRAMMING III	$22.75
CXL4015	TELELINK	$21.75
CX4119	FRENCH	$39.75
CX4118	GERMAN	$39.75
CX4120	SPANISH	$39.75
CX4120	SPANISH	$39.75
CXL4007	MUSIC COMPOSER	$33.75
CXL4002	ATARI BASIC	$45.75
CX8126	MICROSOFT BASIC	$65.75
CXL4003	ASSEMBLER EDITOR	$45.75
CX8126	MACROASSEMBLER	$69.75
CXL4018	PILOT HOME	$65.75
CX405	PILOT EDUCATOR	$99.75
CX415	HOME FILING MANAGER	$41.75
CX414	BOOKEEPER	$119.75

NEW RELEASES

CHOP LIFTER	$27.75
APPLE PANIC	$23.75
PREPPIE	$19.95

THIRD PARTY SOFTWARE for atari 800 or 400

K-BYTE

KRAZY SHOOTOUT	$35.00
K-DOS	$65.00
K-STAR PATROL	$37.75
K-RAZY ANTICS	$37.75
K-RAZY KRITTERS	$37.75
Q-BALL JOYSTICK KIT	$6.75

AUTOMATIED SIMULATIONS

Star Warrior	$28.00
Crush, Crumble & Chomp	$23.00

WE CARRY MANY OTHER THIRD PARTY PRODUCTS YOU CAN CALL FOR PRICES ON AND ASK FOR YOUR FREE ATARI PRODUCT CATALOG.

commodore

VIC-20 ..... $189.00

VIC1530 DATASSETTE	$67.00
VIC1540 DISK DRIVE	$499.00
VIC1515 PRINTER	$355.00
VIC1210 3K RAM	$35.00
VIC1110 8K RAM	$52.00
VIC1211A SUPER EXPANDER	$53.00

VIC-20 SOFTWARE

VIC1212 PROGRAMMER AID	$45.00
VIC1213 VICMON	$45.00
VIC1906 SUPER ALIEN	$23.00
VIC1914 ADVENTURE LAND ADVENTURE	$35.00
VIC1915 PRIVATE COVE ADVENTURE	$35.00
VIC1916 MISSION IMPOSSIBLE	$35.00
VIC1917 THE COUNT ADVENTURE	$35.00
VIC1919 SARGON II CHESS	$35.00

THIRD PARTY SOFTWARE

ALIEN BLITZ	$21.00
Omega Race	$35.00
Gorf	$32.00
16K RAM/ROM	$99.00
AMOK	$21.00
SUPER HANGMAN	$16.00
SPIDERS OF MARS	$45.00

MasterCard **POLICY** VISA

In-Stock items shipped within 24 hours of order Personal checks require four weeks clearance before shipping. PA residents add sales tax. All products subject to availability and price change. Add 4 % for Mastercard and Visa.

TO ORDER CALL TOLL FREE

800-233-8760

In PA 1-717-398-4079 or send order to

Lyco Computer
P.O. Box 5088
Jersey Shore, PA 17740

surroundings. For instance, two players would play *Asteroids,* but not one against the other. One player would fly the ship, while the other player pushed the trigger to shoot the missiles. The children played in pairs for a cumulative score. When all the children reached a grand group total of 1,000,000 points, we had a party. We were all one big team focused on a goal which stressed mutual support, cooperation and social reinforcement. No child was allowed to compare his score to another child's, or make a discouraging comment. Players could only offer positive support to one another. In just a short period of time, these children showed remarkable and demonstrable gains in all aspects of their social behavior. Children who were not used to expressing any other emotion but anger were patting each other on the back and offering one another praise. Months after I finished working with the children, I received many letters, one of which came from a previously withdrawn and isolated child:

Dear Mr. Favaro:

We are sad that you are not here with us anymore. We learned lots of manners. You were very nice to us. Thank you for letting us play Atari.

Your friend,
Michael

Soon after I finished my work with these children, I began experimenting with the Atari 800, creating games using the experiences I gained from my work with them. The game design philosophy outlined here is intended to aid programmers in developing games which facilitate positive social contact between the players. The *GCG* design philosophy suggests modification of current game tasks, roles, themes and values.

Setting the Game Goal: The goal of any game is the very reason why players play it. Common game goals include scoring points, having fun, winning the treasure, saving a planet, etc. Game designers should structure the goals of the games with partnership and cooperation in mind. Make the goal impossible to achieve without a high level of cooperation. Failures should occur when one player tries to overstep his boundaries, show off, etc.

Structure the Game Theme: Life is valuable, even in video games, and not enough has been done to teach this concept. Games should be constructed so that the game theme of "killing the bad guys" becomes "capturing the bad guys." Focus on rescue missions, good will missions and intergalactic peace making. Packaging and advertising art should be carefully planned. One of the most astonishing things about video games, and one which has definitely contributed to their bad image, is that while the actual graphics themselves are often stick-like schematic representations, the packaging is filled with scenes of graphic violence.

Alternate the Game Tasks: Video games typically focus on only one kind of skill — shooting the invaders, completing the maze, etc. Difficulty is increased by increasing the number of targets, the speed of the targets, or both. Game tasks can be made more variable, so that the player who is poor at shooting the invaders might excel in something else, like landing the space pod. In other words, create games that have multiple tasks with different situations or frames requiring different skills. Split up a task so that two players have to develop "togetherness" to do it well, something akin to what must be done in a three-legged race. Let STICK(0) control the horizontal movement of player 1 and STICK(1) control the vertical movement of player 1.

Alternate and Equalize the Roles of the Players: Let both members of the team be co-captains with equal responsibilities. When the game theme calls for roles with unequal power, vary the situation or environment so that player 1's role characteristics are more powerful in situation one, but player 2's characteristics are more powerful in situation two. Make the players *dependent* on each other.

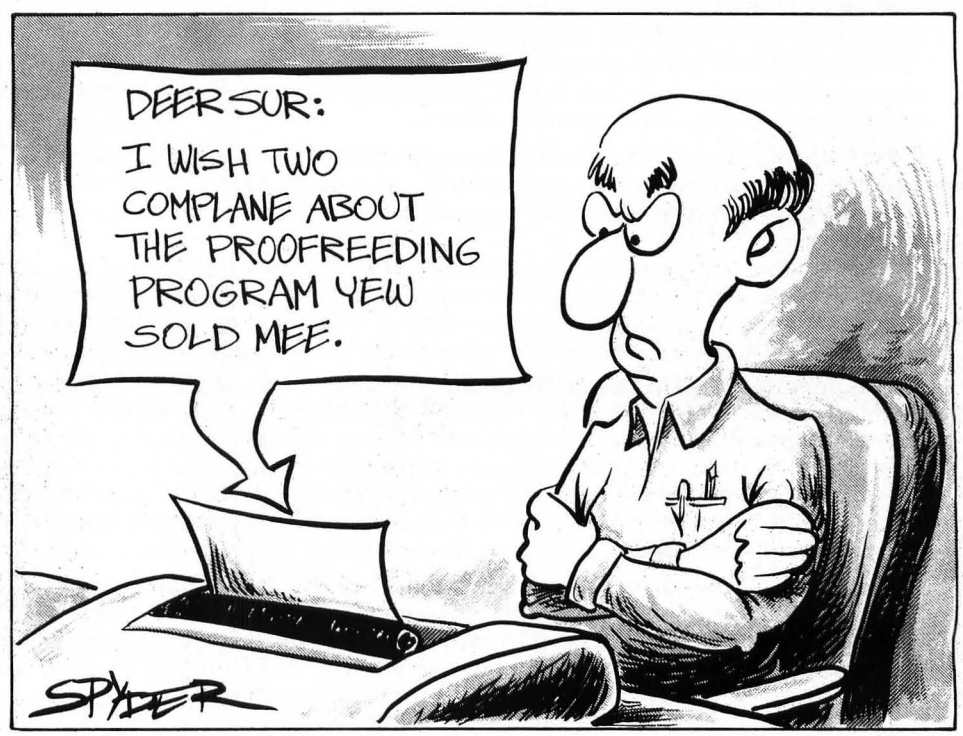
Emphasize Higher Level Thinking Tasks: Stress problem-solving and strategy planning. Create pauses in the game so that players can confer and plan their strategy together. Allow several means to achieve the same goal.

Let the Computer Be a Model: Program the computer to demonstrate the appropriate social reinforcements, such as printed statements like "Nice work." Always make the compliments team-oriented. Have the computer direct cooperation by signalling the need for conferences or strategy planning sessions. If the score of the game reflects the level of cooperation of the players, you could have the computer print a message when the score drops below a preset level.

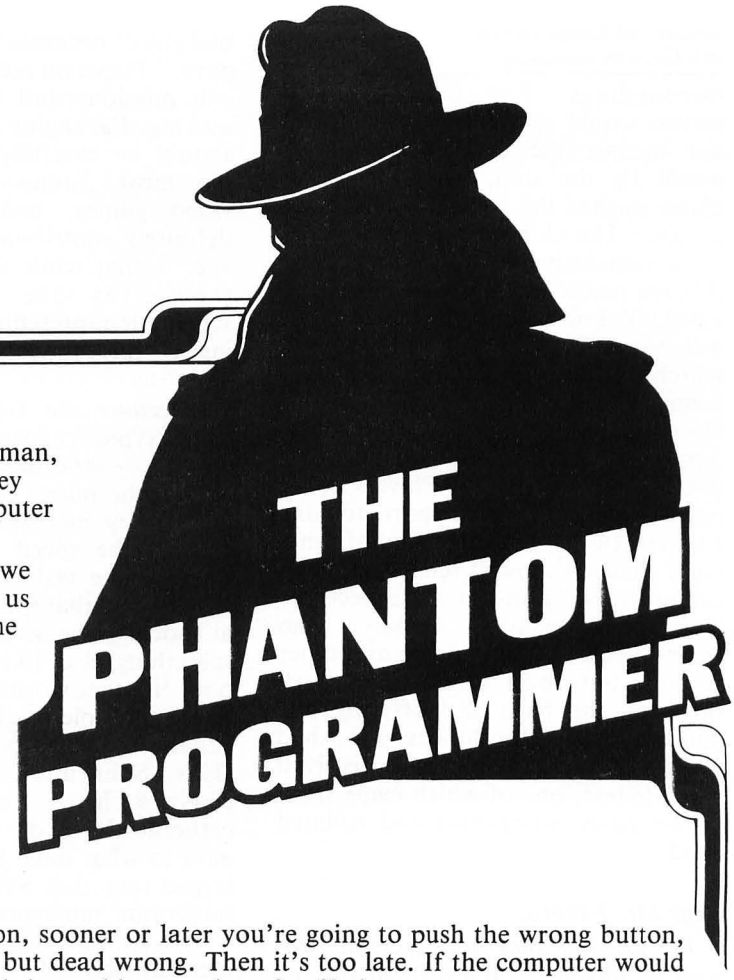
Using *GCG* design philosophy in video games does not require any drastic change in programming techniques. As a matter of fact, any intermediate level programmer with a sensitivity for cooperation and togetherness could easily develop these games.

In a future article, I will feature an example of a game using *GCG* design philosophy. In the meantime, I would be interested in hearing from programmers out there already working on projects incorporating these concepts.

SoftTakes



A FEW WORDS FROM...



To err is human, to forgive, divine.

By the above measurement, computers are probably human, since the hardware and software foul up occasionally. They are definitely not divine, at least not yet. It is a rare computer that shows any tendency to forgive.

This irascibility is quite sad to witness. It is a sign that we have not fully grasped the opportunity computers present us to deal constructively with human error. We who make the hardware and the software are human, too, and tend to underestimate the fallibility of ourselves and those we serve.

Step back and look at the things that go wrong when you use your computer. There is a consistency to the blunders, isn't there? Most of them happen when you do something you really didn't want to do. Maybe you're overtired...not really comfortable with the computer yet..under a deadline and working too fast — maybe you're just due for arrest and persecution under Murphy's Law. Whatever the reason, sooner or later you're going to push the wrong button, enter the wrong command, or do something else innocent but dead wrong. Then it's too late. If the computer would just let you change your mind about the last thing you did, it would save a lot of suffering.

There is a way to build this sort of forgiving nature into computers. I call it the NEVER MIND function. All it does is reverse your last action. That's all. NEVER MIND directly addresses the key element of most mistakes: the fact that people tend to *do* things they haven't *thought* of yet. No matter what hasty, irrevocable thing you have done, just tell the computer, "NEVER MIND", and you are saved.

As I envision it, NEVER MIND will be able to undo *anything* you can do. It should not be too difficult, for instance, to restore a BASIC program deleted by NEW or SCRATCH. Likewise, it should not take much to restore a disk file that has been DELETEd or KILLed. I realize that some belligerent systems zero out such de-allocated sectors, but this is hostile programming and contrary to the spirit of NEVER MIND.

NEVER MIND will allow you to re-enter a program or a language and make a soft landing. This will save you from those moments when you exit BASIC without saving your program, or leave your word processor without saving the last installment of your bi-monthly column.

NEVER MIND will even abort diskette formatting, although this will not be easy to do. It is plain that the previous contents of the diskette will have to be buffered somewhere so that they may be replaced if you suddenly realize that the disk being formatted is your only copy of the data files for the *VisiCalc* model you are showing at a board meeting tomorrow.

I can see that some applications will demand a measure of discretion. For example, if you press NEVER MIND during a PURGE operation, the system must understand whether you want to restore the last file purged, or all files purged. We must carefully define the scope of what we mean by "last action taken." It is conceivable that NEVER MIND might unravel hours or days of work by not knowing where to stop.

It is important also to define what will occur if you press NEVER MIND twice or more in succession. Two interpretations are possible. One is that the second NEVER MIND cancels the first. The other is that each NEVER MIND cancels one more preceding action. It is probably safer to use the first. Using our PURGE example of before, the worst that could happen in the first instance is that you would run in a tight little circle, toggling your file in and out of existence. If we adopt the second method, you might find whole episodes of your life NEVER MINDed by keyboard bounce. In either case, we must be careful how we define what happens when we NEVER MIND a NEVER MIND. Computers do not take kindly to paradox and many become sullen and withdrawn when stumped.

We must work toward the day when NEVER MIND is a prominent button, clearly labeled, on the console of every computer made, supported by every operating system. That's the direct, aggressive approach. Too much time is lost and too much ill feeling generated toward computers by their clumsy handling of human error. Don't trust the applications programmer to properly trap out every error. Build forgiveness right into the machine. Make it a feature that must be used. Never give anyone a chance to paint himself into a corner. We can build computers that can cope gracefully with the mistakes that people always make. We can build computers that forgive — that would be divine indeed.

Franklin's ACE 1000 Runs With The Best!

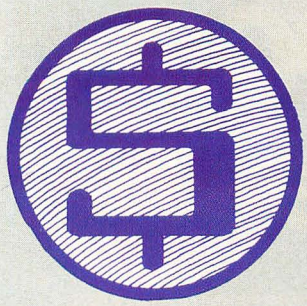
VisiCalc®, DB Master®, Desktop Plan®—they are all running on the Franklin ACE 1000. Cash flow, budgets, word processing or data base management, business or pleasure, the ACE 1000 runs with the best.

The Franklin ACE 1000 is hardware and software compatible with the Apple® II. Franklin users can choose from an enormous selection of programs —programs that run better on the ACE because it includes 64K of RAM, upper and lower case, VisiCalc keys, a numeric pad and an alpha lock key.

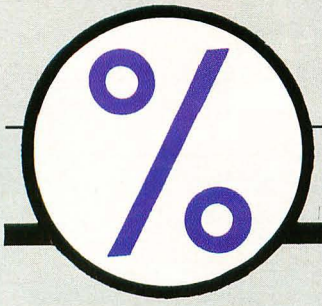
Run with the best. Call or write today for the name of your local authorized Franklin dealer.

FRANKLIN
COMPUTER CORPORATION

7030 Colonial Highway
Pennsauken, NJ 08109
609-488-1700



CALC SIDE



by David Peters

The VisiCalc® Spreadsheet Comes Home

Editor's Note: This month, we welcome a new writer to SoftSide. David Peters will be a regular columnist, and every other month he will cover applications, ideas and tips on using VisiCalc. David is on the staff of Intercalc, the International Electronic Spreadsheet Users Group, and writes regularly for their newsletter, SpreadSheet. (Membership in this group costs $25.00 per year, and includes a subscription to Spreadsheet.) Intercalc started out two years ago as Visigroup, specializing in VisiCalc. The group has broadened its interest to cover similar programs on the market.

For the last five years, there has been a program used in offices all over the country. It has been credited with selling more micro-computers than all other business software applications combined. *VisiCalc*, distributed by VisiCorp, Inc. in California, was written by two geniuses in a (then) small company in Massachusetts called Software Arts. Bob Franklin and Dan Bricklin were students looking for an easier way. The way they found was instantly recognized by all who saw it as having the ultimate appeal in computer programs — a "blank" program that users could "write" exactly as they wanted.

For those of you who don't know the program, a very brief word about what it does. (How it helps will become evident when we look at our first practical application.) *VisiCalc* sets up the computer (many computers, including Apple™, Atari® and TRS-80®) as a blank sheet of accounting paper. The familiar columns are there (62 of them), and 254 lines across the page to keep things straight. Yes, it is a *very large* sheet of paper and it is blank. The CRT screen acts as a window (Imagine a TV camera.) that you can move about the sheet, looking at its segments.

On this blank piece of electronic paper you can write essentially three kinds of things. You can write alpha characters (called *LABELS* in VisiTalk), you can write digits (*VALUES*), or you can write *FORMULAE*. A formula can contain the usual mathematical operators like plus and divide, and also logic operators like greater than, less than, and so on. The formula can, of course, use *VALUES*, but the power of the program is that they can reference *COORDINATES*, the other locations within the matrix. The formula (A1+B1-C1)/D1 tells *VisiCalc* to "add the contents of coordinate location A1 (column A, row 1) to the contents of B1, subtract C1, and then divide the result by D1."

There must be close to a million micros in offices around the country using spreadsheet matrixing programs, and *VisiCalc* started it all! Now the program is coming home, and we are going to concentrate on the more personal uses of *VisiCalc* — ways in which it can help a family. The first application is that monthly chore, balancing the checkbook. In future articles we will take the model we have created

and broaden it, adding an expense management feature, then a budgeting feature, and so on, building each month until we have a complete system for home finances — all in one matrix.

VisiCalc users range from neophytes to afficionados. For the purpose of this column, I will assume that everyone can benefit from knowing what is going on. That way, the "cool" people can just skip over the boring part, while the newcomers will not be left in the dark. The only assumption I will make is that you can boot the program and are aware of the *VisiCalc* commands. Boot up now and away we go!

Maintaining The Checkbook

Take a look at Figure 1. In the top left hand corner of the spreadsheet, there is an activity going on that looks very much like what you always do, either on the check stub or in a little book the bank provides. There's a column for the check number, two more for the dollar amount of the checks and the deposits, and a balance column that tells you the good or bad news.

We have inserted some transactions. As you can see, most of the things in there are "hard" values. The column for check number is either LABELS or VALUES, so you can annotate the entries. Notice the word "MACHINE." That was a withdrawal made from one of those friendly stainless steel things at the shopping center. In the same way, CHARGES might appear, to enter the bank's charges, or DEBIT MEMO for a bounced check, etc.

Over on the right, in BALANCE, there is a simple formula: +E7+(D8-C8). This formula takes the preceding balance, adds anything that appears in the deposit column and subtracts anything in the checks column, thus producing the current state of affairs. This running total formula is /Replicated down the column, and will up-date every time you enter a new transaction.

This formula could handle checks and deposits on the same line. We have chosen to put each transaction on a separate line, however, so that the "notes" can be included for both checks and deposits, and also so that we can introduce some other features that we will build in future articles.

You maintain this part of the model, perhaps weekly, or even daily, if you write a lot of checks. Some prefer to do it once at balancing time, which brings us to an explanation of that cryptic column heading in Figure 1, '1' = OK.

Look now at Figure 2, the complete matrix. When the statement comes, you boot up the model and check off the transactions that have cleared the bank by inserting a 1 in this column. This is just like that narrow column the bank gives you in the little book headed with a checkoff mark. We leave this column blank (We do not insert a 0.) if the item is still open. In Columns F to H in Figure 2 is a work area, a place where *VisiCalc* works by itself.

The formula in Column G (checks not cleared) is @IF(A7=1,0,C7). One of the most powerful features of *VisiCalc* is conditional logic and here is a very useful application of it. It is saying to *VisiCalc*: "Look at Column A. **continued on page 27**

Figure 1: The Daily/Weekly Transactions.

	A	B	C	D	E
1	CHECK BOOK MAINTENANCE & BALANCING.				
2	==========				
3					
4	'1'= OK	CHK #/DEP	CHECKS	DEPOSITS	BALANCE
5	----------				
6		OPENING BALANCE-->			300.00
7				650.53	950.53
8		97	160.23		790.30
9		98	69.86		720.44
10		99	70.00		650.44
11		PAYCHECK		325.00	975.44
12		100	50.00		925.44
13		TRANSFER		267.16	1192.60
14		REFUND		44.10	1236.70
15		MACHINE	40.00		1196.70
16		101	50.00		1146.70
17					1146.70
18					1146.70
19					1146.70
20					1146.70
21					1146.70
22					1146.70
23					1146.70
24					1146.70
25					1146.70
26					1146.70
27	----------				
28	TOTALS		440.09	1286.79	846.70
29	----------				

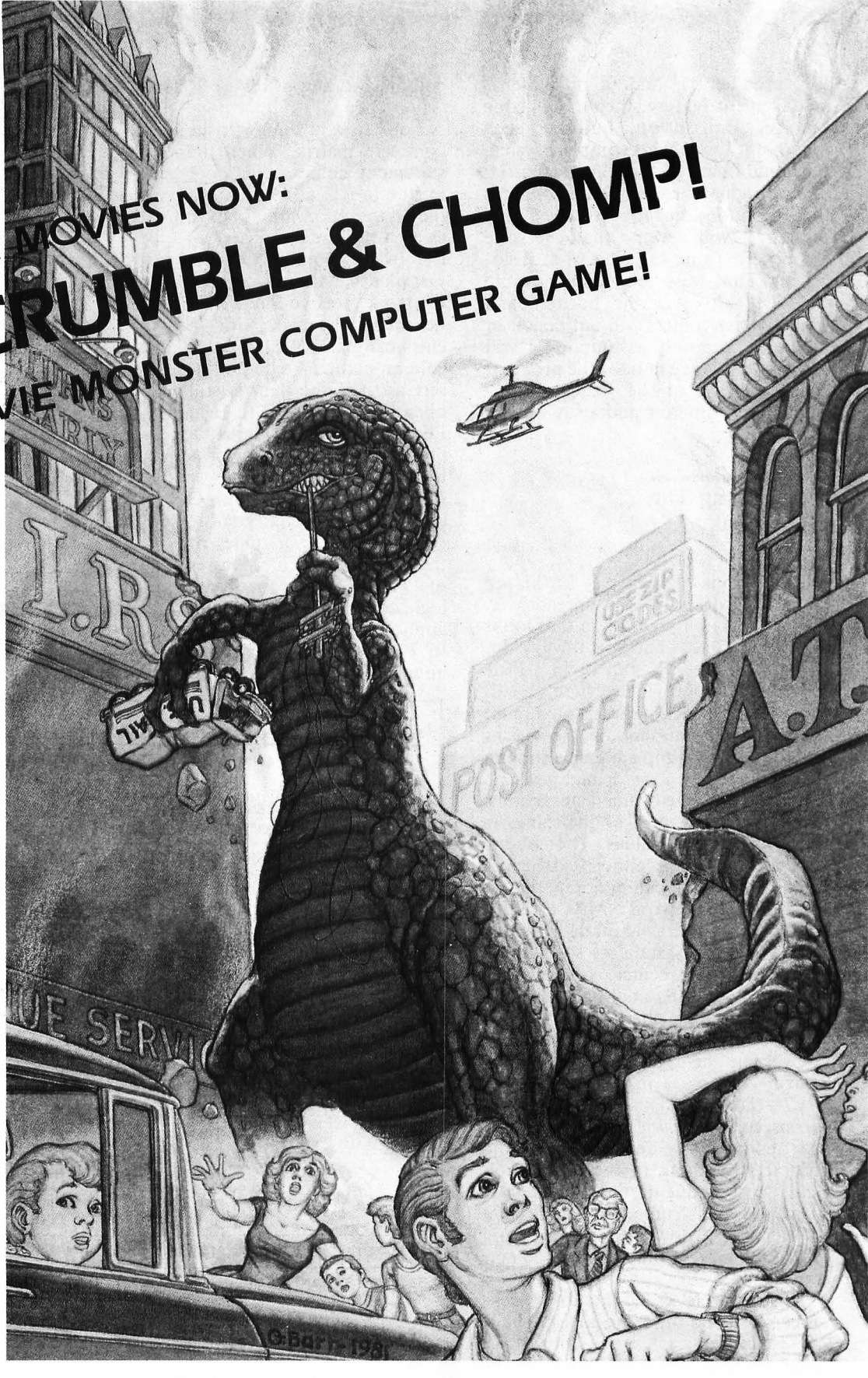
Figure 2: The Complete Matrix, at Statement Balancing.

	A	B	C	D	E	F	G	H
1	CHECK BOOK MAINTENANCE & BALANCING.							
2	==========					VISICALC WORK AREA		
3						NOT CLEARED BANK		
4	'1'= OK	CHK #/DEP	CHECKS	DEPOSITS	BALANCE	CHECKS	DEPOSITS	
5	----------					----------		
6		OPENING BALANCE-->			300.00			
7	1			650.53	950.53	0.00	0.00	
8	1	97	160.23		790.30	0.00	0.00	
9	1	98	69.86		720.44	0.00	0.00	
10	1	99	70.00		650.44	0.00	0.00	
11	1	PAYCHECK		325.00	975.44	0.00	0.00	
12	1	100	50.00		925.44	0.00	0.00	
13	1	TRANSFER		267.16	1192.60	0.00	0.00	
14	1	REFUND		44.10	1236.70	0.00	0.00	
15	1	MACHINE	40.00		1196.70	0.00	0.00	
16		101	50.00		1146.70	50.00	0.00	
17					1146.70	0.00	0.00	
18					1146.70	0.00	0.00	
19					1146.70	0.00	0.00	
20					1146.70	0.00	0.00	
21					1146.70	0.00	0.00	
22					1146.70	0.00	0.00	
23					1146.70	0.00	0.00	
24					1146.70	0.00	0.00	
25					1146.70	0.00	0.00	
26					1146.70	0.00	0.00	
27	----------					----------		
28	TOTALS		440.09	1286.79	846.70	50.00	0.00	
29	----------					----------		
30	ENTER STATEMENT BALANCE--->			1196.70				
31								
32	ENTER CHECKBOOK BALANCE--->			1146.70				
33								
34		DEPS NOT RECORDED		0.00				
35								
36		CHECKS OUTSTANDING		50.00				
37								
38		TRUE BALANCE		1146.70				
39								
40								
41		RECONCILIATION		0.00				
42								
43		YOUR ACCOUNT OKAY IF ZERO						

FIRST: MONSTER MOVIES NOW:

CRUSH, CRUMBLE & CHOMP!

THE GREAT MOVIE MONSTER COMPUTER GAME!



And guess who stars as the movie monster. You! As any of six different monsters. More if you have the disk version.

You can terrorize and destroy four of the world's largest and most densely populated cities in over 100 possible scenarios. From Tokyo to the Golden Gate, you are the deadliest creature in the air, on the land, or in the sea.

You can be the deadly amphibian who simultaneously smashes street cars, lunches on helpless humans and radiates a ray of death.

If you were a giant winged creature, think of the aerial attacks you could make on the terrified but tasty tidbits beneath you.

But as in all the best monster movies, you're up against everything the human race can throw at you—even nuclear warheads and a strange concoction developed by a team of mad scientists.

For only $29.95 you get 6 stupendous monsters, each with its own monstrous summary card, 4 teeming metropoli displayed in graphic detail on your computer display and mapped in the accompanying 48-page illustrated book, the awesome sounds of monsterly mayhem, and spine-tingling, real-time, edge-of-your-seat excitement.

GET CRUSH, CRUMBLE & CHOMP
now at your local dealer for your APPLE, ATARI, or TRS-80 . . **before it's too late.**

COMPUTER GAMES THINKERS PLAY

EPYX

> *"There must be close to a million micros in offices around the country using spreadsheet matrixing programs, and VisiCalc started it all! Now the program is coming home, and we are going to concentrate on the more personal uses of VisiCalc — ways in which it can help a family."*

If the value there equals 1 (cleared the bank), then insert the value of 0. Otherwise, insert the amount found in Column C."

Similarly, in Column H (deposits not recorded by the bank), the formula is: @IF(A7 = 1,0,D7). This time, bring over the amount in Column D.

Some versions of *VisiCalc,* i.e. the 13 sector version for Apple, do not have the logic function. For those of you that have not yet acquired this version, or for whom it is not yet available, there is an alternative to this procedure. You must do the opposite to the above — place a 1 against those items *not* on the statement. Your formula in Columns G and H will respectively be as follows: (C7*A7) and (D7*A7). *VisiCalc* will not multiply by either one or zero and thus carry over the right amounts for you.

These two columns are totalled at the bottom so that *VisiCalc* can perform the completion of the balancing act — it needs only one more figure from you. Insert in Row 30 the actual balance as it appears on your bank statement. Beneath it, *VisiCalc* will bring down the balance it figures you have from D26.

We should say here that, while it doesn't look elegant, this is a reason for the running total column repeating that last balance down the page — we want it in that last position. During the daily or weekly activity you will be /Inserting rows as you need more space, and *VisiCalc* will be pushing the bottom part of the matrix down for you, adjusting the formulae as it goes.

With the entry of this last figure, *VisiCalc* politely informs you if your account balances by putting zero in C41, or an amount indicative of an imbalance, which should help you find the problem.

The formulae in the various spaces are probably obvious. If not, take a look at the printed "aid to balancing" that the bank puts on the back of the statement. *VisiCalc* brings over the two figures from the work area, adds the deposits to the bank's figure and deducts the checks outstanding. It then takes the balance it thinks you have from the resulting adjusted figure.

Next time, we will take this simple, but useful, application and make it even more useful. Feel free to write me with questions or problems you have with *VisiCalc*. I promise to reply if you enclose a stamped, self-addressed envelope. Your letters will give me a clue to the kinds of applications you would like me to cover. If there is something you would like to have *VisiCalc* do, I will devise a model and share it with everyone. Members of *InterCalc* have sent us ideas for literally hundreds of terrific applications, which we publish in *SpreadSheet*, and you can use them too.

HOPPER

***Hopper* is an arcade-style game for a TRS-80® Model I/III with 16K RAM, an Apple with 48K RAM and a joystick (optional), or an Atari with 32K RAM (24K RAM without disk) and a joystick.**

TRS-80® version by Howard Wolkow
Atari® version by Rich Bouchard and Alan J. Zett
Apple™ version by Kerry Shetline
Documentation by Fred J. Condo

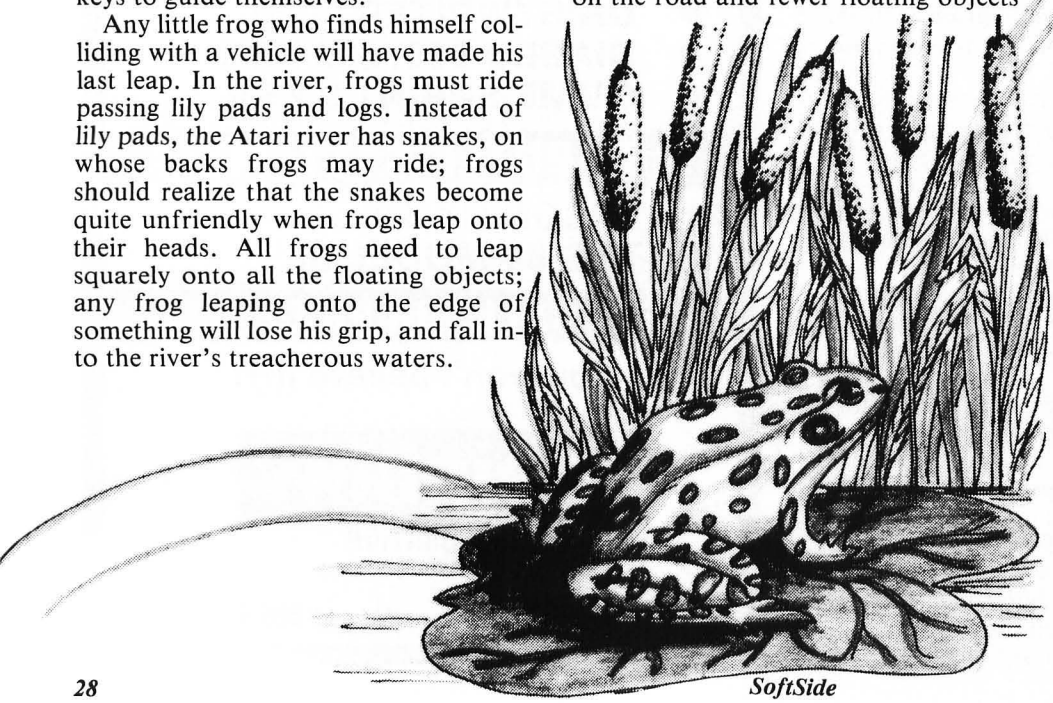
You are a lost little frog. You've been wandering everywhere in search of your swampy home. At long last, you catch sight of it. With a croak of relief, you dash forward, only to find that a superhighway, teeming with speeding vehicles, blocks your way. The question is, can you cross the road without colliding with several tons of high-velocity metal? If you can, will you then be able to negotiate the debris-ridden river beyond?

You will have to be most dexterous to accomplish these tasks. Atari frogs must use their joysticks plugged into port 1. Apple frogs have a choice of using a joystick (a self-centering one is recommended) or the keyboard. If they choose the keyboard, the A and Z keys will move them up and down, and the arrow keys will move them left and right. TRS-80 frogs must use the arrow keys to guide themselves.

Any little frog who finds himself colliding with a vehicle will have made his last leap. In the river, frogs must ride passing lily pads and logs. Instead of lily pads, the Atari river has snakes, on whose backs frogs may ride; frogs should realize that the snakes become quite unfriendly when frogs leap onto their heads. All frogs need to leap squarely onto all the floating objects; any frog leaping onto the edge of something will lose his grip, and fall into the river's treacherous waters.

On the far riverbank are five swampy havens — the only safe places around. To reach home safely, frogs must land squarely in these little niches; in this unfriendly world, inaccuracy's reward is an untimely demise. Also, frogs can survive outside their havens for only a short while. If time runs out, again, the result is demise.

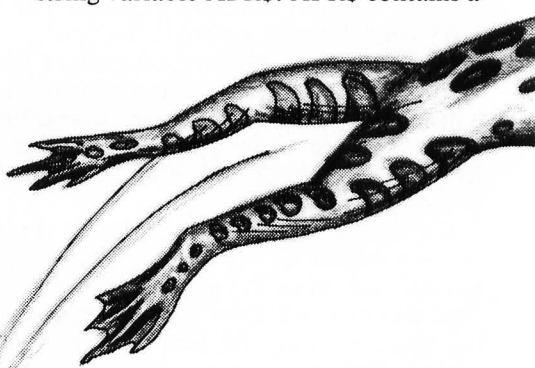
On the bright side of things, however, are points. Each time a frog hops, he earns 20 points for Frogdom, except that Atari frogs get these points only for hopping *forward.* In addition, Atari frogs receive bonus points in proportion to the amount of time left on the clock when they reach their havens. Reaching a haven is worth 250 points for all three kinds of frog. Reaching all five havens is worth 2000 points, and causes the world to move to the next, harder level, where there are more cars on the road and fewer floating objects in the river. In addition, unlike poor humans, who have but one life, frogs have five lives. Naturally, the game ends when all five lives have been exhausted.



Notes on the Apple *Hopper:* The listing shown is for Apples with a disk drive. If you have a cassette system, first type the program *exactly* as it appears in the listing. Then *SWAT* the program to check for errors. When your *SWAT* table matches the published table, delete lines 150 through 180, and 570 through 600.

Notes on the Atari *Hopper: Hopper* contains many lines of graphics and of data, some of which are very difficult to type in. Thus, an extended SWAT table has been printed for the Atari program. It is highly recommended that you use SWAT on *Hopper* before running it, because a small mistake can cause disastrous results. It is also suggested that you save the program to tape or disk every time you RUN it. When working correctly, there is a long initialization phase at the start of the program, so be patient. Also, be sure to hit "System Reset" immediately after exiting from *Hopper* in any manner.

In line 30010 in the Atari listing is a string variable XFR$. XFR$ contains a Machine Language routine that quickly sets up a redefined character set for the Atari. It is very difficult to represent this string in the listing of the program, so here are explicit instructions on how to type it.



In these instructions, each key is represented by something between brackets. Thus [h] means to type a lower-case "h"; likewise, [Atari] means to press the Atari logo key, which is found next to the right-hand shift key, and [CTRL-N] means to press the "N" key while holding down the "CTRL" key. Special note: [1] is the numeral one — it is *not* the lower-case version of "L"!

Type the following sequence exactly to produce XFR$:

[h] [Atari] [)] [Atari] [CTRL-,]
[Atari] [CTRL-E] [K] [CTRL-E] [M]

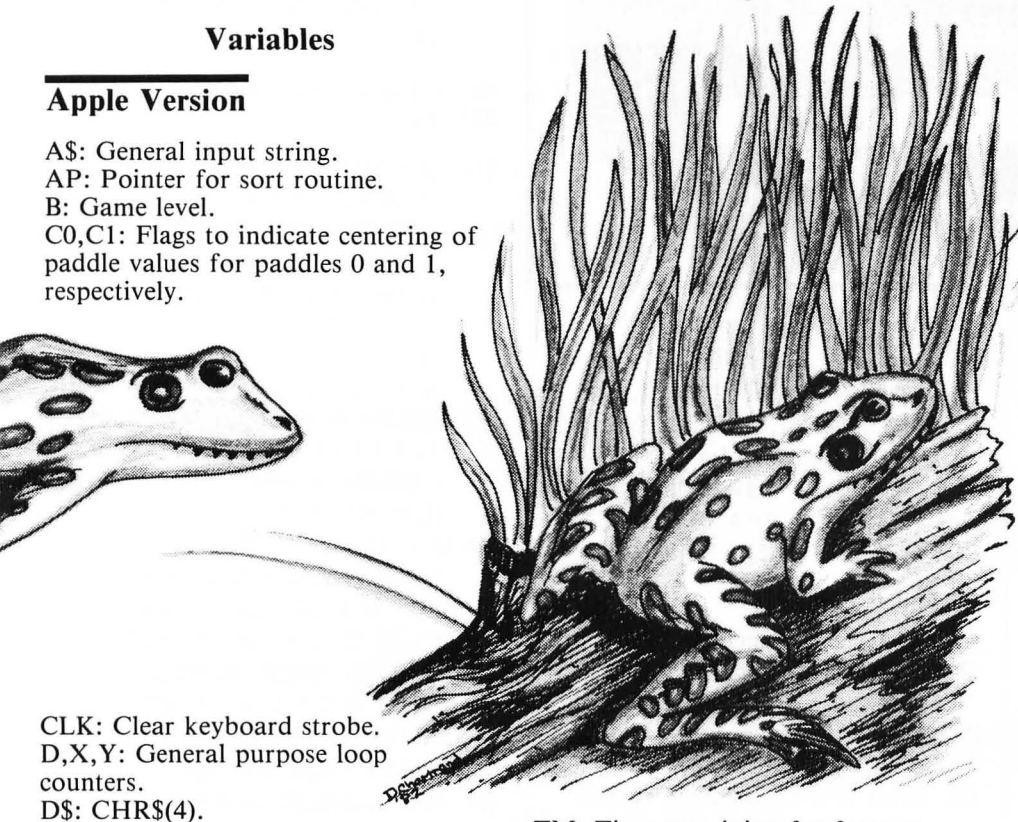
[)] [CTRL-.] [CTRL-E] [N] [%] [Atari] [j] [CTRL-X] [i] [CTRL-A] [Atari] [CTRL-E] [L] [spacebar] [Atari] [CTRL-,] [Atari] [1] [M] [CTRL-Q] [K] [H] [P] [y] [f] [L] [f] [N] [%] [N] [I] [d] [P] [m] [Atari] [CTRL-.]

Atari line listing: The *Hopper* printed in the magazine is for Atari owners who have the GTIA graphics chip. A listing of changes for CTIA owners follows. Due to the wide variety of color displays in use by Atari owners, you may need to change the four color values in lines 11150 and 11160 to make the colors look right. These four bytes are the colors of each of the four areas of the screen. They consist of a color number times sixteen plus a luminance number.

Variables

Apple Version

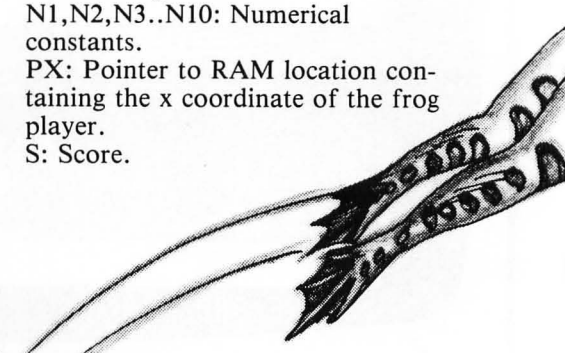
A$: General input string.
AP: Pointer for sort routine.
B: Game level.
C0,C1: Flags to indicate centering of paddle values for paddles 0 and 1, respectively.



CLK: Clear keyboard strobe.
D,X,Y: General purpose loop counters.
D$: CHR$(4).
DA(*): Rate of drift for logs and lily pads.
FA: Number of frogs that have gotten across.
FR: Number of frogs remaining.
FX: Absolute X coordinate of frog.
FY: Y coordinate of frog, as a row number.
H2: Address of the start of hi-res page 2.
HB,LV,SL,SS: Variables used in generation of hi-res look-up table.
HN,HN$: Hold variables for sort routine.
ID: Index from side of screen for placement of logs.
JF: Joystick mode flag.
KBD: Keyboard input address.
LG: Number of logs per row.
LP: Number of lily pads per row.
LT: Hi-res look-up table address.
M: Players move, from keyboard or joystick.
MB: Memory location at start of line of hi-res screen.
MU: Location of sound routine.
N: Number to be displayed on screen.
N$: String equivalent of N.
NX: New X coordinate.
NY: New Y coordinate.
P0,P1: Paddle values 0 and 1, respectively.
P2: Screen switch to display Page 2.
RD: Random displacement of cars from the side of the screen.
SC: Score.
SI: Score increment.
SP: Spacing between logs or lily pads.
TL: Table look-up value.
TM: Time remaining for frog.
TN$(*): Names of top ten players.
TS(*): Scores of top ten players.
XP: Position to start plotting numbers on screen.
YA: Absolute Y coordinate of frog.
YN$: Name of player who made top ten.

Atari Version

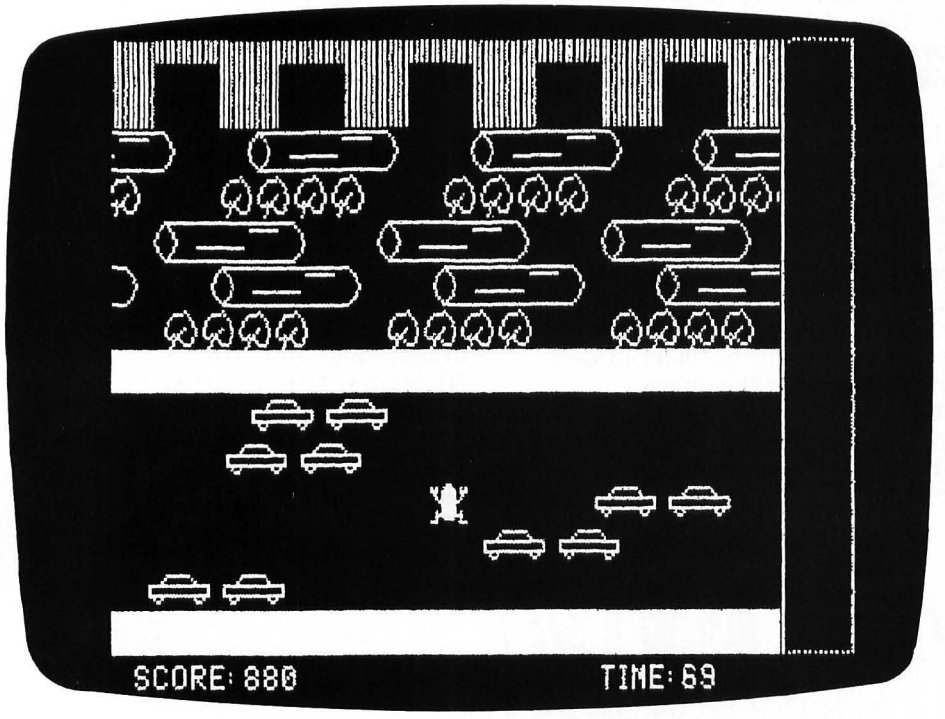
A,A1,A2,A3,A$,A1$: Misc. uses.
A9: Used to hold position of joystick.
BYTE$: Two byte hexidecimal number.
DEATH$: Graphics for drowning frog.
DOWN$: Machine Language routine to move frog down.
F$: One graphics frog.
FP: Furthest progress the frog has made across the screen.
FR: Number of frogs remaining.
HS(*): Top five scores.
HS$: Names of top five scorers.
J,J1: Loop variables.
PM: Base for Player/Missile graphics.
M,M1,M2: Used to keep track of remaining time.
MB: Number of different levels in the game.
MOVE$: Machine Language routine to move logs and cars.
N1,N2,N3..N10: Numerical constants.
PX: Pointer to RAM location containing the x coordinate of the frog player.
S: Score.



S(*),S1(*): Values for the background sounds to be played when the frog is moved onto different screen lines.
SC: Top of screen memory.
SP(*): Direction and speed the frog is moved on each screen line (when on logs).
UP$: Machine Language routine to move frog up.
UP(*): Number of screen lines the frog moves up when on a given screen line.
XP,YP: X and Y coordinates of frog on screen.

TRS-80 Version

A: General variable.
AM: Score.
B: Board number (1-3).
CA, CB, CL, CP, CS, CT, G: Used to construct screen graphics.
D: Number of frogs left.
F$: Graphics character for frog.
FL, FR, R, L: Counters for screen graphics.
H: Number of frogs safely home.
HS(i): High scores.
HS$(i): Names of high scorers.
I, N: Loop variables.
M: Time left.
P: Position of frog (1-1023).
SA, SC, SL, T$: Screen graphics strings.
VM: Start of video memory (equals 15360).
X: General variable.
XP: X position.




```
  SS SS SS SS SS SS SS SS SS SS SS
  SS                             SS
  SS       APPLESOFT BASIC       SS
  SS            'HOPPER'         SS
  SS    AUTHOR: HOWARD WOLKOW    SS
  SS    TRANSL: KERRY SHETLINE   SS
  SS      COPYRIGHT (C) 1982     SS
  SS SOFTSIDE PUBLICATIONS, INC SS
  SS                             SS
  SS SS SS SS SS SS SS SS SS SS SS
```

Program Initialization.

```
100  LOMEM: 24576: TEXT : HOME : VTAB
     11: PRINT "JOYSTICK OR KEYBO
     ARD (J/K)? ";:JF = 0
110  GET A$: IF A$ <  > "J" AND A
     $ <  > "K" THEN 110
120  IF A$ = "J" THEN JF = 1
130  HOME : PRINT "INITIALIZING..
     .": DIM DA(6),TS(9),TN$(9):M
     U = 768: FOR X = MU TO 788: READ
     N: POKE X,N: NEXT X:HS = 122
     88: FOR X = HS TO 12432: READ
     N: POKE X,N: NEXT X: FOR X =
     13056 TO 13553: READ N: POKE
     X,N: NEXT X: POKE 232,0: POKE
     233,51:SG = 49232:P2 = 49237
     :KBD = 49152:CLK = 49168
140  H2 = 16384:LT = 12544: FOR Y =
     0 TO 23:SS =  INT (Y / 8):SL
      = Y - SS * 8: FOR D = 0 TO
     7:LV = H2 + SS * 40 + SL * 1
     28 + D * 1024:HB =  INT (LV /
     256): POKE LT,LV - HB * 256:
      POKE LT + 1,HB:LT = LT + 2:
      NEXT D,Y:LT = 12544:DA(1) =
     7:DA(2) =  - 7:DA(3) = 14:DA
     (4) =  - 7:DA(5) = 14:DA(6) =
     0
```

Read high scores from disk (disk systems only).

```
150  D$ =  CHR$ (13) +  CHR$ (4): ONERR
      GOTO 170
160  PRINT D$"OPEN HOPSCORES": PRINT
     D$"READ HOPSCORES": FOR X =
     0 TO 9: INPUT TS(X): INPUT T
     N$(X): NEXT X
170  PRINT D$"CLOSE": POKE 216,0
```

Create hi-res game display.

```
180  HGR2 : TEXT : ROT= 0: SCALE=
     1: HCOLOR= 6: FOR Y = 0 TO 6
     : HPLOT 0,Y TO 250,Y: NEXT Y
     : FOR X = 0 TO 14 STEP 2: HPLOT
     X,7 TO X,25: NEXT X: FOR X =
     40 TO 232 STEP 48: FOR D = 0
      TO 22 - 4 * (X = 232) STEP
     2: HPLOT X + D,7 TO X + D,25
     : NEXT D,X
190  HCOLOR= 1: HPLOT 253,0 TO 25
     3,181 TO 279,181 TO 279,0 TO
     253,0: HCOLOR= 3: FOR Y = 91
      TO 103: HPLOT 0,Y TO 251,Y:
      HPLOT 0,Y + 78 TO 251,Y + 7
     8: NEXT Y: DRAW 15 AT 10,185
     : DRAW 16 AT 184,185
```

Set up for new game.

```
200  POKE SG,0: POKE P2,0:SC = 0:
     FR = 5:B = 1: HCOLOR= 0: FOR
     Y = 185 TO 191: HPLOT 50,Y TO
     120,Y: NEXT Y: HCOLOR= 3: FOR
     Y = 25 TO 85 STEP 20: DRAW 1
     1 AT 260,Y: NEXT Y:SI = 0: GOSUB
     640
```

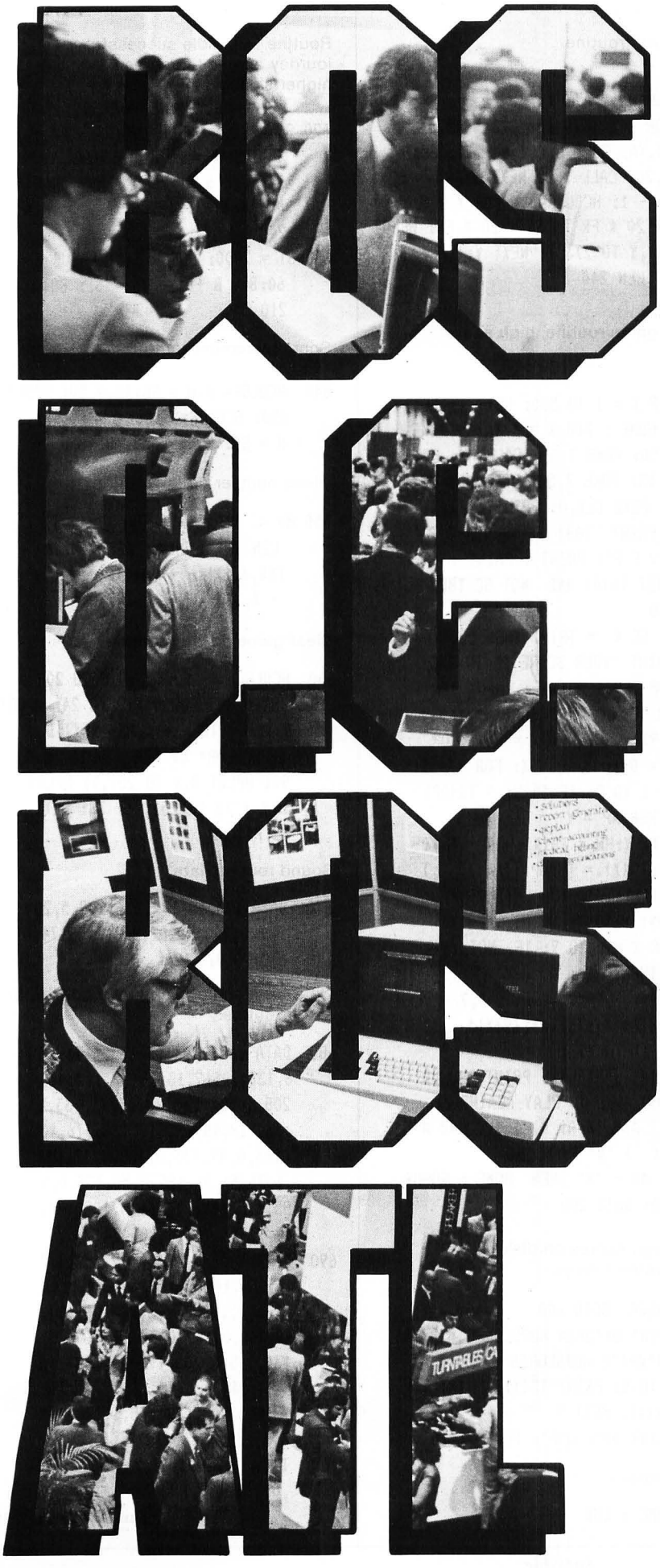
Set up display for present level of difficulty.

```
210  CALL 54915:FA = 0:LG = 4 - B
      + (B = 3):LP = 4 - B: FOR Y
      = 27 TO 79 STEP 13: HCOLOR=
     3: IF Y = 27 OR Y = 53 OR Y =
     66 THEN SP =  INT ( INT (252
      / LG) / 7 + .5) * 7:ID = 42
      * (LG = 2) + 7: FOR X = ID TO
     ID + (LG - 1) * SP STEP SP:
     HCOLOR= 4: HPLOT X,Y - 1
     TO X + 41,Y - 1
     : HCOLOR= 3: DRAW 13 AT X +
     23,Y: NEXT X: GOTO 230
220  SP =  INT ( INT (252 / LP) /
     7 + .5) * 7:ID = 42 * (LP =
     2) + 98 * (LP = 1) + 7: FOR
     X = ID TO ID + (LP - 1) * SP
      STEP SP: HCOLOR= 4: HPLOT X
     ,Y - 1 TO X + 41,Y - 1: HCOLOR=
     3: FOR D = 6 TO 48 STEP 14: DRAW
     12 AT X + D,Y: NEXT D,X
230  NEXT Y: HCOLOR= 3: FOR Y = 1
     06 TO 158 STEP 13:RD =  INT
     ( RND (1) * (120 - (B + 1) *
     14)) + 10: FOR X = 0 TO 28 *
     B STEP 28: DRAW 14 AT RD + X
     ,Y: NEXT X,Y
```

Start off a new frog.

```
240  FOR D = 1 TO 100: NEXT D: POKE
     CLK,0: HCOLOR= 0: FOR Y = 18
     5 TO 191: HPLOT 215,Y TO 228
     ,Y: NEXT Y:FX = 120:FY = 12:
     TM = 80
```

continued on page 32



SHOWS FOR YOU IN '82

THE NATIONAL COMPUTER SHOWS are the largest public computer expositions in the country. They feature hardware and software for business, industry, government, education, home, and personal use.

Under one roof you'll be able to test the new computers, desk top computers, data and word processing equipment, and a huge array of computer peripherals, computer services and computer supplies.

At each show you'll see all the major brands, all the major manufacturers, the big distributors and the local dealers and retailers. Everything together and for sale at super show prices.

Don't miss the National Computer Show coming to a city near you. Admission is $5 per person per day.

TICKET INFORMATION:
Send $5 (payable to National Computer Shows) for each day-long ticket along with the name of the Show you plan to attend to the address below. Tickets can also be purchased at the show.

The National Computer Shows,
822 Boylston St., Chestnut Hill, MA 02167
Telephone: 617-739-2000

BOSTON
THE NORTHEAST PERSONAL COMPUTER SHOW
Friday-Monday October 8-11 (Columbus Day Weekend)
Hynes Auditorium/Prudential Center
Show Hours: Friday 11 am to 9 pm,
Saturday & Sunday 12 noon to 9 pm,
Monday 11 am to 6 pm

WASHINGTON, D.C.
THE MID-ATLANTIC COMPUTER SHOW
Thursday-Sunday October 28-31
D.C. Armory/Starplex
Show Hours: 11 am to 6 pm daily

BOSTON
THE NORTHEAST BUSINESS COMPUTER SHOW
Thursday-Sunday November 11-14
Hynes Auditorium/Prudential Center
Show Hours: 11 am to 6 pm daily

ATLANTA
THE SOUTHEAST COMPUTER SHOW
Thursday-Sunday December 9-12
Atlanta Civic Center
Show Hours: 11 am to 6 pm daily

THE NATIONAL COMPUTER SHOWS

822 Boylston St., Chestnut Hill, MA 02167

Hopper *continued*

Move frog and determine results, keep track of time.

```
250 YA = 14 + FY * 13: XDRAW 11 AT
    FX,YA: IF FY > 6 AND FY < 12
     THEN  IF  PEEK (234) <  > 8
    8 THEN 480
260  IF FY < 6 THEN TL = LT + 2 *
    (YA - 1):MB =  PEEK (TL) + 2
    56 *  PEEK (TL + 1): IF  PEEK
    (MB +  INT (FX / 7)) < 128 THEN
    480
270  HCOLOR= 0:XP = 215:N = TM +
    1: GOSUB 650: HCOLOR= 3:N =
    TM: GOSUB 650: POKE 6,255: POKE
    7,3: CALL MU:M =  PEEK (KBD)
    : POKE CLK,0
280  IF M <  > 155 THEN 310
290  IF  PEEK (KBD) < 128 THEN 29
    0
300  GOTO 270
310  IF  NOT JF THEN 340
320 M = 0:P0 =  PDL (0):P1 =  PDL
    (1):C0 = 30 < P0 AND P0 < 22
    5:C1 = 30 < P1 AND P1 < 225:
     IF C1 AND  NOT C0 THEN M =
    136 + 13 * (P0 > 127): GOTO 340
330  IF C0 AND  NOT C1 THEN M = 1
    93 + 25 * (P1 > 127): GOTO 340
340  IF M = 193 OR M = 136 OR M =
    149 OR (M = 218 AND FY <  >
    12) THEN  POKE 6,50: POKE 7,
    20: CALL MU: POKE 6,40: POKE
    7,30: CALL MU:SI = 20: GOSUB
    640: GOTO 360
350  FOR D = 1 TO 50: NEXT D
360 NX = FX:NY = FY: IF M = 193 THEN
    NY = FY - 1
370  IF M = 136 AND FX - 14 > =
    0 THEN NX = FX - 14
380  IF M = 149 AND FX + 14 < 238
     THEN NX = FX + 14
390  IF M = 218 AND FY <  > 12 THEN
    NY = FY + 1
400  IF NY THEN 430
410  XDRAW 11 AT FX,27:YA = 14: XDRAW
    11 AT FX,YA: IF  PEEK (234) <
     > 88 THEN 480
420  GOTO 620
430  IF FY > 6 THEN  XDRAW 11 AT
    FX,YA: CALL HS: XDRAW 11 AT
    FX,YA: GOTO 460
440 NX = NX + DA(FY): IF NX < 0 OR
    NX > 237 THEN 480
450  CALL HS:FX = FX + DA(FY)
460  XDRAW 11 AT FX,YA:FX = NX:FY
     = NY:TM = TM - 1: IF TM THEN 250
```

Dead frog routine.

```
470  XDRAW 11 AT FX,YA
480  FOR X = 1 TO 15: XDRAW 11 AT
    FX,YA: POKE 6,X * 6 + 10: POKE
    7,20: CALL 768: NEXT X:FR =
    FR - 1: HCOLOR= 0: FOR Y = 5
     + 20 * FR TO 36 + 20 * FR: HPLOT
    260,Y TO 273,Y: NEXT Y: IF F
    R THEN 240
```

End of game routine, high score sorting.

```
490  FOR X = 1 TO 500: NEXT X: TEXT
    : HOME : FOR X = 1 TO 4: POKE
    6,50: POKE 7,50: CALL MU: POKE
    6,15: POKE 7,50: CALL MU: NEXT
    X: POKE CLK,0: PRINT : PRINT
    : PRINT  TAB( 12)"G A M E
    O V E R": PRINT : PRINT : IF
     NOT TS(0) AND  NOT SC THEN
    550
500  IF SC <  = TS(9) THEN 530
510  PRINT "YOUR SCORE IS IN THE
    TOP TEN": PRINT : INPUT "ENT
    ER YOUR NAME PLEASE: ";YN$:T
    S(9) = SC:TN$(9) = YN$: FOR
    X = 0 TO 8:AP = X: FOR Y = X
     + 1 TO 9: IF TS(Y) > TS(AP)
     THEN AP = Y
520  NEXT Y:HN = TS(X):HN$ = TN$(
    X):TS(X) = TS(AP):TN$(X) = T
    N$(AP):TS(AP) = HN:TN$(AP) =
    HN$: NEXT X: PRINT : PRINT
530  FOR X = 0 TO 9: IF  NOT TS(X
    ) THEN 550
540  PRINT  LEFT$ ("      ",7 - LEN
    ( STR$ (TS(X))));TS(X)"   "T
    N$(X): NEXT X
550  PRINT : PRINT : PRINT "WOULD
     YOU LIKE TO PLAY AGAIN? ";
560  GET A$: IF A$ <  > "Y" AND A
    $ <  > "N" THEN 560
570  IF A$ = "Y" THEN  HOME : GOSUB
    660: GOTO 200
```

Write high scores on disk (disk systems only).

```
580  ONERR  GOTO 600
590  PRINT D$"OPEN HOPSCORES": PRINT
    D$"WRITE HOPSCORES": FOR X =
    0 TO 9: PRINT TS(X): PRINT T
    N$(X): NEXT X
600  PRINT D$"CLOSE": POKE 216,0
```

End program.

```
610  HOME : END
```

Routine to handle successful journey across the screen. Will set higher skill level if necessary.

```
620 SI = 250: GOSUB 640:FA = FA +
    1: FOR X = 25 TO 13 STEP  -
    1: POKE 6,X: POKE 7,30: CALL
    MU: NEXT X: IF FA <  > 5 THEN
    240
630 SI = 2000: GOSUB 640: GOSUB 6
    60:B = B + 1 - (B = 3): GOTO
    210
```

Score increment routine.

```
640  HCOLOR= 0:N = SC:XP = 50: GOSUB
    650: HCOLOR= 3:SC = SC + SI:
    N = SC: GOSUB 650: RETURN
```

Hi-res number diplay routine.

```
650 N$ =  STR$ (N): FOR X = 1 TO
     LEN (N$): DRAW  ASC ( MID$
    (N$,X,1)) - 47 AT XP + 7 * X
     - 7,185: NEXT X: RETURN
```

Clear game display routine.

```
660  HCOLOR= 4: FOR X = 15 TO 209
     STEP 48: FOR D = 0 TO 24: HPLOT
    X + D,7 TO X + D,25: NEXT D,
    X: HCOLOR= 0: FOR Y = 26 TO
    90: HPLOT 0,Y TO 251,Y: HPLOT
    0,Y + 78 TO 251,Y + 78: NEXT
    Y: RETURN
```

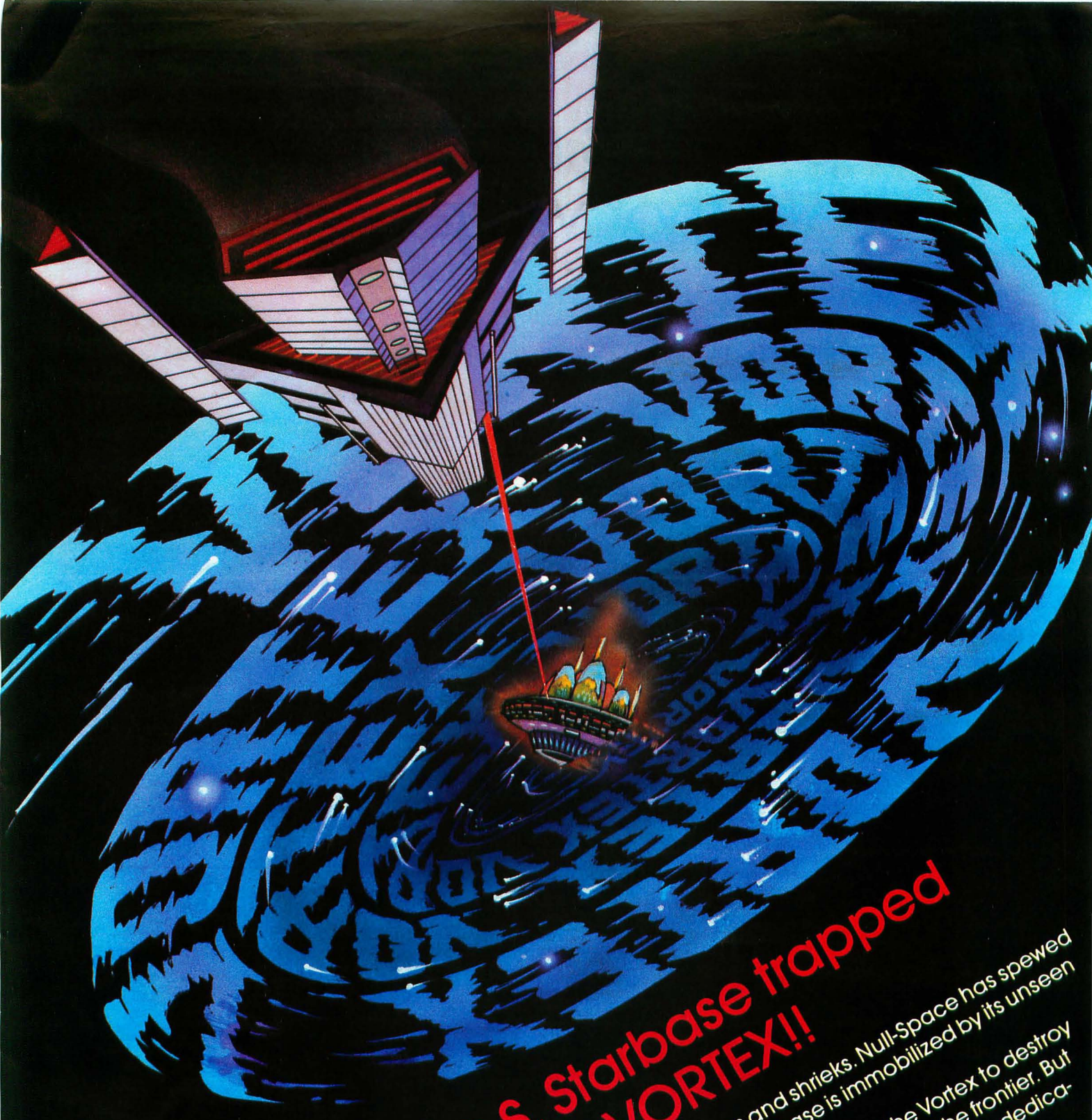
Sound routine data.

```
670  DATA 173,48,192,136,208,5,20
    6,7,0,240,9,202,208,245,174,
    6,0,76,0,3,96
```

Data for Machine Language horizontal scroll routine.

```
680  DATA 169,0,133,3,133,4,169,2
    6,133,2,162,13,164,2,192,91,
    208,10,24,165,2,105,13,133,2
    ,76,119,48,152,10,168,176,13
    ,185,0,49,133,0,185,1,49,133
    ,1,76,56,48,185,0,50,133,0,1
    85,1,50,133,1,164,4,185,134,
    48,133,5,165,3,208,24
690  DATA 160,35,177,0,72,136,177
    ,0,200,145,0,136,208,247,104
    ,145,0,198,5,208,235,76,114,
    48,160,0,177,0,72,200,177,0,
    136,145,0,200,192,35,208,245
    ,104,145,0,198,5,208,233,230
    ,2,202,208,149,165,3,73,1,13
    3,3,230,4,165,4,201
700  DATA 11,208,133,96,1,1,2,1,2
    ,0,1,1,2,2,1
```

continued on page 35



S.O.S. S.O.S. Starbase trapped in the VORTEX!!

A blast of black light! The universe shakes and trembles and shrieks. Null-Space has spewed forth the Vortex, a deadly gravitational whirlpool!! Your Starbase is immobilized by its unseen grip just as the Valtron fleet attacks.

These methane breathing aliens spiral into the coils of the Vortex to destroy Starbase, Earth's farthest outpost. You must hold them on the frontier. But they're lightfast, suicidal, fanatical. Superhuman skill and dedication are called for . . . and you alone can answer.

A new, fantastic fighting game for the Apple II*.

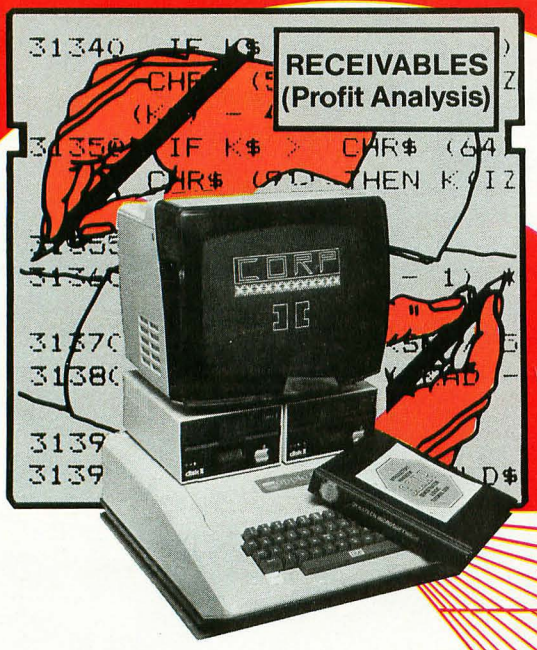
Only $29.95 at computer stores, or:

DATAMOST

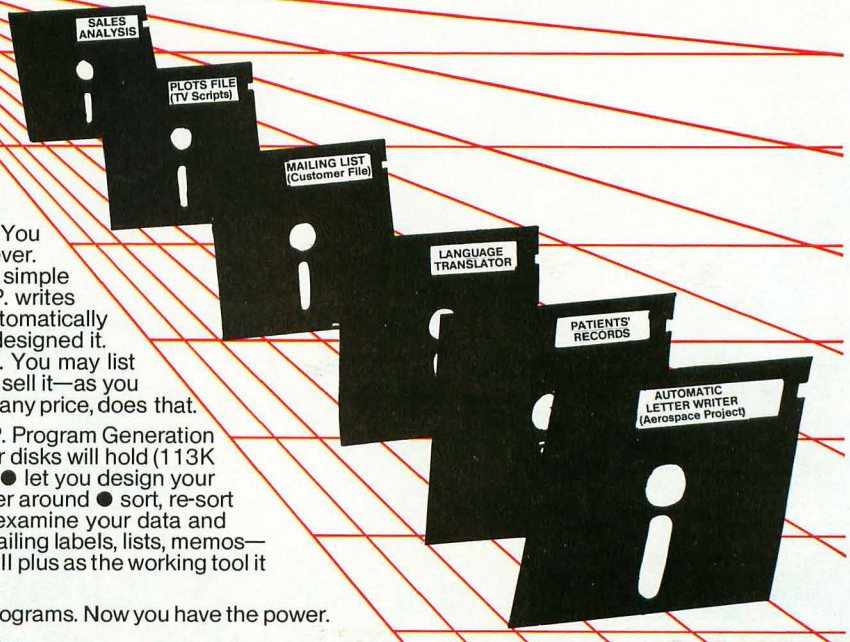
9748 Cozycroft Avenue, Chatsworth CA. 91311 (213) 709-1202.

VISA/MASTERCARD accepted.
$2.00 shipping/handling charge.
(California residents add 6½% tax.)
*Apple II is a trademark of Apple Computer, Inc.

your power is ready...
software that writes software
for Apple II™



NEEDED: Apple II Plus with Auto Start, 2 drives, Applesoft in Rom, 48K.



C.O.R.P.™ writes software. Perhaps you should read that again. Not "simplifies programming" or "makes debugging easier." C.O.R.P. writes APPLESOFT BASIC—complete, stand-alone programs which run, bug-free, the very first time. You need not type a single character of basic code—ever. C.O.R.P. does that. Your answers to questions in simple English "design" the program. In minutes, C.O.R.P. writes all the program code, scrolls it to the screen and automatically saves it to your disk. Your program because you designed it. Once written, your program runs without C.O.R.P. You may list your program, examine it, modify it further or even sell it—as you wish, royalty free. No mere "data base manager," at any price, does that.

The applications are almost limitless. Your C.O.R.P. Program Generation System will: ● handle nearly as much data as your disks will hold (113K per disk!) ● find any record in less than a second ● let you design your own screens as quickly as you can move the curser around ● sort, re-sort and update your data for you ● examine and re-examine your data and then print reports, checks, invoices, statements, mailing labels, lists, memos—you name it. C.O.R.P. allows you to use your Apple II plus as the working tool it really is. Isn't that why you bought one?

Computers should write code. You sould design programs. Now you have the power.

Complete C.O.R.P. System includes: Master Disk ● Utility 1 ● Utility 2 ..... $250

Diagnostic Disc $30

User Demo Disk $20

Manual (Deductible from later order of complete system)

CORTECHS CORPORATION

7847 N. Caldwell, Chicago, Illinois 60648

See your dealer today or call toll-free 1-800-621-4109

(In Illinois, call (312) 470-0700

Dealer inquiries welcome.

C.O.R.P.™

C.O.D. welcome.

C.O.R.P. is a trademark of Maromaty Scotto Software Corporation.

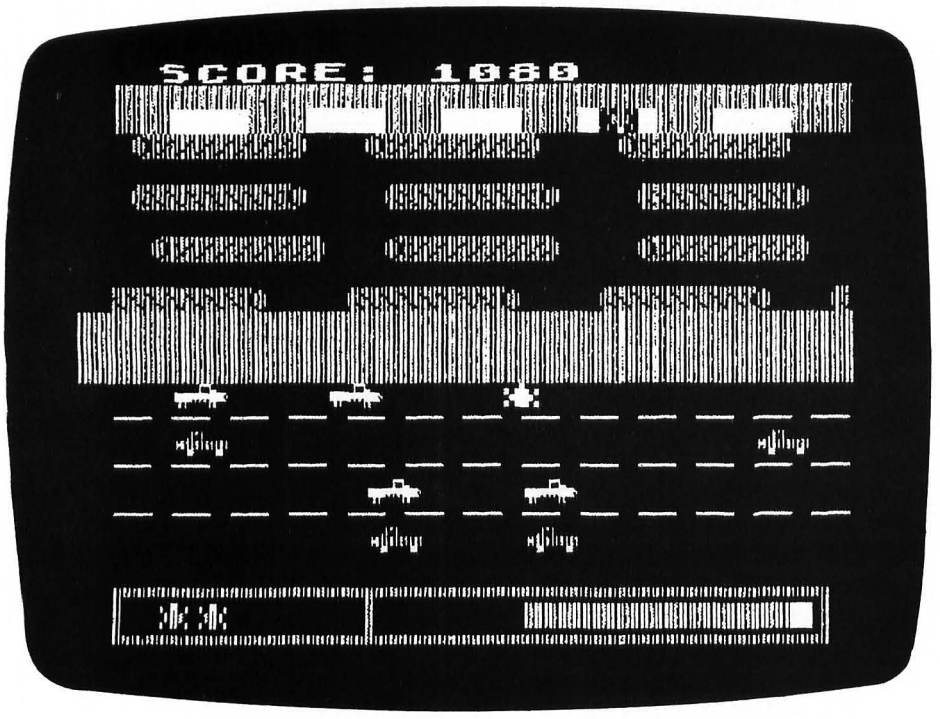
Hopper *continued*

Shape table data.

```
710 DATA 16,0,36,0,50,0,58,0,69,
    0,79,0,87,0,98,0,113,0,121,0
    ,135,0,149,0,205,0,234,0,86,
    1,131,1,195,1,242,1,41,173,5
    4,54,30,63,7,32,101,12,220,5
    1,34,0,9,62,14,54,54,253,7,0
    ,42,40,173,246,30,30,23,45,4
    5,5,0,42,40,173,246
720 DATA 14,246,63,7,40,0,54,46,
    45,37,180,50,54,0,45,45,222,
    27,54,45,173,246,63,63,0,73,
    17,28,63,23,54,54,14,45,5,32
    ,28,63,7,0,45,45,54,30,30,30
    ,46,0,41,173,246,63,23,118,4
    5,5,32,220,27,32,4,0,146,18,
    14,45,5,32,36,228,63
730 DATA 23,118,45,5,0,73,73,173
    ,63,255,251,46,245,14,14,37,
    36,45,45,109,48,45,244,242,3
    0,39,60,63,55,45,53,63,247,4
    5,45,53,63,63,255,106,45,45,
    109,54,46,253,3,8,24,63,63,6
    3,63,54,63,77,9,40,45,5,0,21
    ,21,21,118,54,30,246
740 DATA 63,7,224,100,12,223,146
    ,23,246,4,8,24,8,23,63,32,10
    0,100,12,45,45,0,45,45,45,45
    ,45,45,45,45,45,45,45,45,21,
    21,14,118,246,246,30,30,63,6
    3,63,63,63,63,63,63,63,63,63
    ,63,63,63,63,63,63,63,63,63,
    63,63,63,63,7,56,56
750 DATA 32,28,100,100,12,12,14,
    14,14,118,246,246,30,77,73,7
    3,9,8,24,40,45,45,45,45,45,4
    5,45,45,77,73,73,73,1,8,24,8
    ,24,56,63,63,63,63,255,219,2
    19,3,8,24,63,63,63,63,63,63,
    63,63,63,63,63,7,0,45,45,21,
    21,21,45,45,54,62,55
760 DATA 30,63,96,57,63,63,63,63
    ,63,63,55,30,63,96,57,63,39,
    36,45,45,37,5,40,40,45,150,2
    19,43,45,45,45,45,45,5,0,45,
    21,223,51,14,45,21,246,63,7,
    104,73,9,36,36,12,45,21,150,
    242,63,68,73,9,118,45,5,32,3
    6,228,63,23,54,77,73
770 DATA 33,36,45,173,246,63,30,
    54,77,225,28,77,9,36,36,45,4
    5,150,59,183,51,45,45,77,8,2
    4,4,32,0,45,45,222,54,54,110
    ,73,45,28,36,36,60,13,77,54,
    54,54,5,8,24,8,24,168,46,8,2
    4,37,22,54,54,77,36,36,36,45
    ,45,150,59,183,42,45
780 DATA 77,8,24,4,32,0
```

APPLE™ SWAT TABLE FOR:
HOPPER (Modified Parameters: NU = 3, B = 500)

LINES	SWAT CODE	LENGTH
100 - 120	MB	103
130 - 150	GX	401
160 - 180	AE	204
190 - 210	XB	374
220 - 240	TE	276
250 - 270	FY	194
280 - 300	LI	45
310 - 330	JY	135
340 - 360	AB	132
370 - 390	GX	85
400 - 420	RC	67
430 - 450	XI	94
460 - 480	NX	157
490 - 510	SC	267
520 - 540	WP	158
550 - 570	OD	96
580 - 600	NX	102
610 - 630	VC	112
640 - 660	YF	197
670 - 690	QN	504
700 - 720	KE	474
730 - 750	SN	650
760 - 780	XI	454




```
SS SS SS SS SS SS SS SS SS SS SS
SS                            SS
SS         ATARI BASIC        SS
SS           'HOPPER'         SS
SS    AUTHOR: HOWARD WOLKOW    SS
SS    TRANSL: RICH BOUCHARD    SS
SS       AND ALAN J. ZETT      SS
SS      COPYRIGHT (C) 1982     SS
SS SOFTSIDE PUBLICATIONS, INC SS
SS                            SS
SS SS SS SS SS SS SS SS SS SS SS
```

Program Initialization.

```
10 GOSUB 10000:GOSUB 11000:GOSUB 12000
:GOSUB 13000:GOSUB 30000:GOSUB 15000
```

Draw background.

```
100 GOSUB 2300
```

Draw logs and cars, prepare for next frog.

```
110 GOSUB 2400
```

Check for joystick movement.

```
120 A9=STICK(N0):A1=N0:A2=N0
130 U=USR(ADR(MOVE$)):ON A9 GOTO 200,2
00,200,200,200,200,140,200,200,200,150
,200,160,170,200
140 A1=N2:GOTO 200
150 A1=-N2:GOTO 200
160 A2=UP(YP+N1):GOTO 200
170 A2=-UP(YP)
```

Move frog, and check for frog death.

```
200 XP=XP+A1+SP(YP)*(A2=N0):YP=YP+A2:I
F XP<N2 OR XP>37 THEN 1200
203 IF A1 OR A2 THEN SOUND N0,210,10,1
0
205 A=PEEK(SC+YP*40+XP):A3=PEEK(SC+YP*
40+XP+N1)
210 IF A2<N0 THEN U=USR(ADR(UP$),PM+52
8):IF A2=-N2 THEN U=USR(ADR(UP$),PM+52
8)
220 IF A2>N0 THEN U=USR(ADR(DOWN$),PM+
528):IF A2=N2 THEN U=USR(ADR(DOWN$),PM
+528)
225 IF A1 OR A2 THEN SOUND N0,120,N10,
10
230 IF YP>=13 AND YP<=19 AND (A<>90 OR
 A3<>90) THEN SOUND N0,N0,N0,N0:GOTO 1
200
240 SOUND N0,N0,N0,N0:IF YP>=N3 AND YP
<=9 AND (A=90 OR A3=90 OR A=80 OR A3=8
0 OR A=83 OR A3=83) THEN 1200
250 IF YP<FP THEN FP=YP:S=S+20:POSITIO
N N9,N0:? S;:IF YP=N2 THEN 1000
260 POKE PX,47+XP*N4:SOUND N3,S(YP),S1
(YP),N2
```

Decrease time remaining.

```
270 M1=M1-N1:IF M1<>N0 THEN 120
280 SOUND N2,200-150*(M<5),12,N8+N6*(M
<N5):M1=M2:POKE SC+917-M,N0:M=M-N1:SOU
ND N2,N0,N0,N0:IF M<>N0 THEN 120
285 SOUND N2,200,12,14:FOR J=N1 TO 250
:NEXT J:SOUND N2,N0,N0,N0
290 GOTO 1400
```

Frog safely in bunker.

```
1000 SOUND N3,N0,N0,N0:IF A<>113 OR A3
<>113 THEN 1200
1010 POKE PX,N0:XP=INT((XP+N3)/N7)*N7-
N1:POSITION XP,YP:? "uv";
1020 S=S+250+M*N10:GOSUB 2480
1030 H=H+N1
1040 FOR T=40 TO N10 STEP -10:SOUND N3
,T,N10,N10:SOUND N3,N0,N0,N0:FOR I=N1
TO 15:NEXT I:NEXT T
1050 IF H<>N5 THEN FOR I=N1 TO 100:NEX
T I:GOTO 110
```

All five frogs safely home.

```
1100 POSITION N2,N0:? "congratulations
!":FOR J=N1 TO 400:NEXT J
1110 FOR J=N1 TO N5:POSITION J*7-N1,N2
:? "qq";:S=S+400:GOSUB 2480
1120 FOR J1=N0 TO N8 STEP 1.2:SOUND N0
,20-J1,N10,N8-J1:SOUND N0,80-J1,N10,N8
-J1:NEXT J1:SOUND N0,N0,N0,N0
1130 FOR J1=N0 TO 20:NEXT J1:SOUND N0,
20,N2,N4:SOUND N1,80,N10,N4:FOR J1=N1
TO 20:NEXT J1
1140 SOUND N0,N0,N0,N0:SOUND N1,N0,N0,
N0:NEXT J:FOR J=N1 TO 200:NEXT J:H=N0:
IF B<MB THEN B=B+N1:GOSUB 2500
1150 GOTO 110
```

Frog is dead. Determine which death routine to use.

```
1200 IF YP>N9 AND (XP<N2 OR XP>37) THE
N SOUND N3,200,14,N10:XP=XP-A1:SOUND N
3,N0,N0,N0:GOTO 205
1202 IF YP=N2 THEN 1400
1205 IF YP>11 THEN 1300
```

Splashed in river or onto snake's mouth.

```
1210 A1=N1:IF XP>=N2 AND XP<=36 AND A<
>80 AND A3<>80 AND A<>83 AND A3<>83 TH
EN A1=N0
1220 POKE PX,N0:FOR J=N1 TO N4
1225 SOUND N3,N3,N8,12-J*N2
1230 IF XP<N2 THEN XP=37
1240 IF XP>37 THEN XP=N2
```

```
1250 A=PEEK(SC+YP*40+XP):A3=PEEK(SC+YP
*40+XP+N1):POSITION XP,YP:? DEATH$(J*N
2-N1,J*N2)
1260 FOR I=N1 TO 30:NEXT I
1270 SOUND N3,N0,N0,N0:POKE SC+YP*40+X
P,A:POKE SC+YP*40+XP+N1,A3:U=USR(ADR(M
OVE$))
1280 XP=XP+SP(YP)*A1:NEXT J
1290 GOTO 1500
```

Run over by car or truck.

```
1300 FOR J=N0 TO N4:SOUND N3,200,N4,N1
0-J*N2:IF J<>N2 AND J<>N4 THEN POKE 53
256,J:POKE PX,47+XP*N4-J*N4
1310 FOR J1=N0 TO 23:IF J1/N8=INT(J1/N
8) THEN U=USR(ADR(MOVE$))
1320 NEXT J1:NEXT J:POKE PX,N0:POKE 53
256,N0:SOUND N3,N0,N0,N0:GOTO 1500
```

Out of time, or missed bunker.

```
1400 FOR J=40 TO 245 STEP N5:FOR J1=N0
 TO N8 STEP N2:SOUND N3,J+J1,N10,N10:P
OKE 704,J+J1:NEXT J1:NEXT J
1410 SOUND N3,N0,N0,N0:POKE 704,24:FOR
 J=N1 TO 200:NEXT J:POKE PX,N0
```

Subtract one life from player, and continue game unless there are no more lives remaining.

```
1500 FR=FR-N1:POSITION N4+FR*N2,22:? "
  ";
```

Get high scoring player's name.

```
1510 IF FR>N0 THEN 110
1520 GOSUB 2200:POSITION N1,N0:? SP$(N
1,20);
1525 POSITION 5-INT(LOG(S+N1)/LOG(N10)
/N2),N0:? "SCORE: ";S;SP$(N1,N10);
1530 ? :? :FOR A=N1 TO N5:IF S<=HS(A)
THEN NEXT A:GOTO 1600
1540 ? "CONGRATULATIONS!  YOUR SCORE I
S THE   ";RA$(A*N3-N2,A*N3);" HIGHEST
SCORE TODAY.":? :TRAP 1520
1550 ? "PLEASE ENTER YOUR NAME,":? "(U
P TO EIGHT CHARACTERS LONG).":? "   >"
;:INPUT A$
1560 TRAP N0:A1$=SP$
1570 IF A=N1 THEN A1$(N1,N8)=A$:A1$(N9
,40)=HS$(N1,32):GOTO 1590
1580 A1$(N1,A*N8-N8)=HS$(N1,A*N8-N8):A
1$(A*N8-N7,A*N8)=A$:IF A<>N5 THEN A1$(
A*N8+N1,40)=HS$(A*N8-N7,32)
1590 HS$=A1$:FOR J=N5 TO A+N1 STEP -N1
:HS(J)=HS(J-N1):NEXT J:HS(A)=S
```

Display high scores and allow another game to be played.

```
1600 GOSUB 2200:POSITION 14,N2:? "HIGH
 SCORES"
1610 FOR J=N1 TO N5:IF HS(J)>N0 THEN P
OSITION 11,N3+J:? J;". ";HS(J);:POSITI
ON 21,N3+J:? HS$(J*N8-N7,J*N8)
1620 NEXT J
1630 POSITION N6,14:? "MOVE JOYSTICK T
O EXIT GAME,":? "   OR PRESS BUTTON TO
 PLAY AGAIN."
1640 IF STICK(N0)<>15 THEN 1700
1650 IF STRIG(N0)=N1 THEN 1640
1660 POSITION N2,N0:? SP$(N1,20);:GOSU
B 2200:GOTO 100
1700 TRAP 1600
1710 POSITION N4,17:? "ARE YOU SURE YO
U WISH TO EXIT";:INPUT A$
1720 TRAP N0:IF A$(N1,N1)<>"Y" THEN 16
00
1730 GRAPHICS 0:STOP
```

Subroutine to read BYTE$ from data, and convert it into a one byte value in A$.

```
2000 READ BYTE$:A1=ASC(BYTE$)-48:IF A1
>N10 THEN A1=A1-N7
2010 A2=ASC(BYTE$(N2,N2))-48:IF A2>10
THEN A2=A2-N7
2020 A$=CHR$(A1*16+A2):RETURN
```

Subroutine used to initialize pointer for Machine Language scroll screen routine.

```
2100 FOR J=N0 TO N3:A2=INT(A1/256):A3=
A1-A2*256:POKE A,A2:POKE A+N1,A3
2120 A=A+N3:A1=A1+80:NEXT J:RETURN
```

Subroutine to clear video screen.

```
2200 COLOR 32:FOR J=N1 TO 23:PLOT N1,J
:DRAWTO 39,J:NEXT J:POSITION N0,N0:RET
URN
```

Subroutine called at the beginning of every game to set up unchanging screen display and to initialize variables.

```
2300 POSITION N2,N1:? "rrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrqqqqrrrqqqqr
rrqqqqrrrqqqqrrrqqqqrrr"
2310 FOR J=14 TO 18 STEP N2:POSITION N
2,J:? "oo oo oo oo oo oo oo oo oo oo o
o oo oo";:NEXT J
2320 POSITION N2,21:? "abbbbbbbbbbbbab
bbbbbbbbbbbbbbbbbbbbc"
2330 ? "s (12 spaces) s (22 spaces)
      t":? "deeeeeeeeeeeedeeeeeeeeeeee
eeeeeeeeef";
2340 POSITION N4,22:? F$;F$;F$;F$;F$;F
R=5
2350 B=N1:S=N0:H=N0
2360 GOSUB 2500
2390 RETURN
```

Subroutine called at the beginning of every frog, to draw logs and cars and to initialize some variables.

```
2400 XP=14+INT(RND(N0)*N10):YP=20:FP=Y
P:GOSUB 12030
2410 POSITION 17,22:? "rrrrrrrrrrrrrr
rrrrrq";:M=20:M2=INT(3.5+B/N2):M1=M2
2430 RESTORE 14000+B*100
2440 COLOR 26:FOR J=N3 TO 19 STEP N2
2445 IF J<>11 THEN PLOT N0,J:DRAWTO 39
,J:READ A$:A=37-LEN(A$):POSITION N2+IN
T(RND(N0)*A),J:? A$;
2450 POKE SC+J*40+39,90:NEXT J
2460 SOUND N3,50,N10,N10:FOR J=N1 TO N
5:NEXT J:SOUND N3,N0,N0,N0
2480 POSITION N2,N0:PRINT "SCORE: ";S;
SP$(N1,20)
2490 POKE 77,N0:RETURN
```

Subroutine to read in data about the speed of each line in each new board.

```
2500 RESTORE 14900+B*N10
2520 FOR J=N1 TO N8:READ A:POKE 1663+J
*N3,A:NEXT J
2530 IF B=N1 THEN SP(N3)=N1:SP(N9)=-N1
2540 IF B=N4 THEN SP(N9)=-N2
2550 IF B=N5 THEN SP(N3)=N2
2590 RETURN
```

Subroutine to initialize constants, some variables, and arrays.

```
10000 N0=0:N1=1:N2=2:N3=3:N4=4:N5=5:N6
=6:N7=7:N8=8:N9=9:N10=10
10010 DIM BYTE$(2),MOVE$(200),A$(40),U
P$(28),DOWN$(28),SP$(40),F$(N2),DEATH$
(8),HS$(40),RA$(15),A1$(40)
10020 DIM SP(24),UP(24),HS(5),S(24),S1
(24)
10030 SP$="
            ":HS$=SP$:A1$=SP$
10040 RESTORE
10050 F$=CHR$(10):F$(N2,N2)=CHR$(11):D
EATH$="ghijklmn":RA$="1ST2ND3RD4TH5TH"
10060 PX=53248:MB=N5
10070 FOR J=N1 TO 24:S(J)=N0:S1(J)=N1:
NEXT J:FOR J=N3 TO N9 STEP N2:S(J)=N3+
INT(J/N2):S1(J)=N8:NEXT J
10080 FOR J=13 TO 19 STEP N2:S(J)=93+I
NT(J/N2):S1(J)=N2:NEXT J
10100 FOR J=N1 TO 24:SP(J)=N0:NEXT J
10110 SP(N3)=N1:SP(N5)=-N1:SP(7)=N1:SP
(9)=-N1
10120 FOR J=N4 TO 19:UP(J)=N2:NEXT J:U
P(20)=N1:UP(21)=N0:UP(N3)=N1
10130 FOR J=N1 TO N5:HS(J)=N0:NEXT J
10990 RETURN
```

Subroutine to set up display list.

```
11000 GRAPHICS 21:POKE 752,N1:C=N0:POK
E 87,N0
11010 DL=PEEK(560)+PEEK(561)*256+N4
11020 MEM=PEEK(DL)+PEEK(DL+1)*256+40
11030 MH=INT(MEM/256):ML=MEM-MH*256
11040 POKE 559,N0:POKE DL-N1,198
11050 POKE DL+N2,66:POKE DL+N3,ML
11060 POKE DL+N4,MH
11070 FOR J=N5 TO 26:POKE DL+J,N2:NEXT
 J
11080 POKE DL+12,130:POKE DL+15,130
11090 POKE DL+22,130:POKE DL+27,65
11100 POKE DL+28,PEEK(560)
11110 POKE DL+29,PEEK(561)
11120 READ A:IF A=999 THEN 11140
11130 POKE 1744+C,A:C=C+N1:GOTO 11120
11140 POKE 512,208:POKE 513,N6
11150 POKE 1774,176:POKE 1775,180
11160 POKE 1776,N0:POKE 1777,144
11170 POKE 1778,N0:POKE 54286,192
11180 POKE 559,34
11190 DATA 72,138,72,174,242,6,189,238
,6,141,10,212,141,24,208,232
11200 DATA 224,4,144,2,162,0,138,141
11210 DATA 242,6,104,170,104,64,999
11220 POKE 710,0:POSITION N2,N0:? "INI
TIALIZING..."
11990 RETURN
```

Subroutine to set up player graphic.

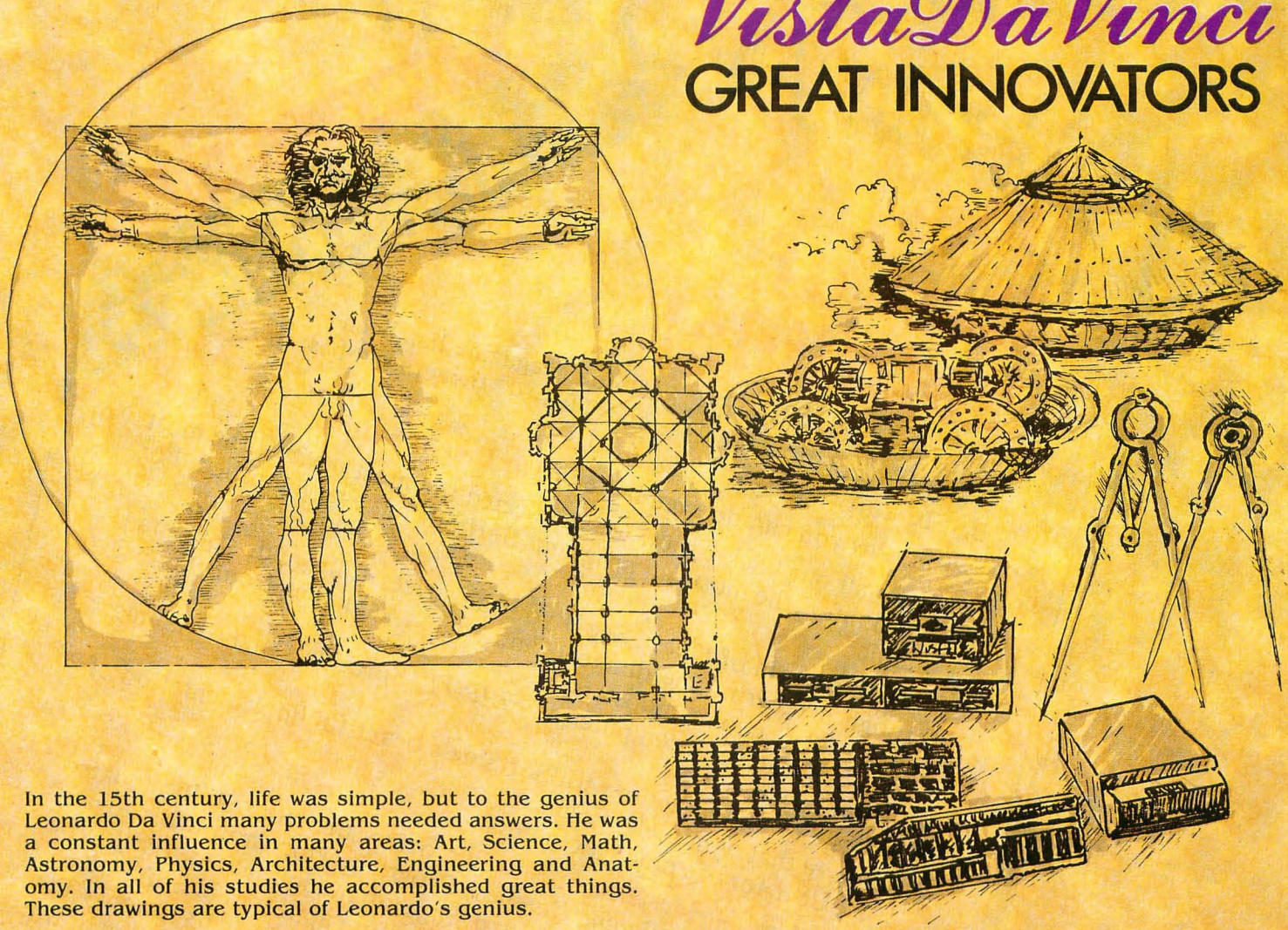
```
12000 A=PEEK(106)-N8:POKE 54279,A:PM=2
56*A:POKE 106,A:XP=-N10:POKE 623,0
12020 POKE 559,46:POKE 53277,N3
12025 POKE 704,24
12030 FOR I=PM+512 TO PM+640:POKE I,N0
:NEXT I
12040 RESTORE 12050:FOR J=PM+608 TO PM
+611:READ A:POKE J,A:NEXT J
12050 DATA 8,93,62,93
12060 POKE PX,47+XP*N4
12070 RETURN
```

Subroutine to initialize Machine Language routine in MOVE$, UP$ and DOWN$.

```
13000 FOR J=N1 TO 171:GOSUB 2000:MOVE$
(J,J)=A$:NEXT J
13010 SC=PEEK(88)+PEEK(89)*256:A=1664
13020 A1=SC+N2+40*N3:GOSUB 2100
13030 A1=SC+N2+40*13:GOSUB 2100
13040 POKE A,N0
13045 DATA D8,A9,80,85,CD,A9,06,85,CE
13050 DATA 38,B0,3D,60,A0,00,B1,CB
13055 DATA 48,A2,25,C8,B1,CB,88,91
13060 DATA CB,C8,CA,D0,F6,68,A0,25
13065 DATA 91,CB,38,B0,27,A0,00,B1
13070 DATA CB,48,C8,B1,CB,48,A2,24
13075 DATA C8,B1,CB,88,88,91,CB,C8
13080 DATA C8,CA,D0,F4,68,A0,25,91
13085 DATA CB,68,88,91,CB,38,B0,04
13090 DATA D8,68,C6,CD,E6,CD,A0,00
13095 DATA B1,CD,F0,B7,85,CC,E6,CD
13100 DATA B1,CD,85,CB,E6,CD,B1,CD,F0,
0C
13105 DATA C9,01,EA,F0,20,C9,02,EA
13110 DATA F0,A0,D0,B7,A0,25,B1,CB
13115 DATA 48,A2,25,88,B1,CB,C8,91
13120 DATA CB,88,CA,D0,F6,68,A0,00
13125 DATA 91,CB,38,B0,C5,A0,25,B1
13130 DATA CB,48,88,B1,CB,48,A2,24
13140 DATA 88,B1,CB,C8,C8,91,CB,88,88,
CA
13145 DATA D0,F4,68,A0,00,91,CB,68
13150 DATA C8,91,CB,38,B0,A2
13200 FOR J=N1 TO 28:GOSUB 2000:UP$(J,
J)=A$:NEXT J
13210 DATA 68,68,85,CC,68,85,CB,A2
13220 DATA 60,A0,00,C8,C8,C8,C8,B1
13230 DATA CB,88,88,88,88,91,CB,C8
13240 DATA CA,D0,F0,60
13300 DOWN$=UP$
```

continued on page 41

Vista Da Vinci
GREAT INNOVATORS



In the 15th century, life was simple, but to the genius of Leonardo Da Vinci many problems needed answers. He was a constant influence in many areas: Art, Science, Math, Astronomy, Physics, Architecture, Engineering and Anatomy. In all of his studies he accomplished great things. These drawings are typical of Leonardo's genius.

Vista Computer strives to answer the complex problems of today's computer high technology. Vista excells in many areas also: Apple® Disk Storage, Apple® Drive Subsystems, Vision Series Video Cards, IBM™ Memories and IBM™ I/O.

Vista sees the continuing need for advancement in all these areas and are constantly expanding the product line to meet tomorrow's computer needs.

Vista Family

® Apple II is a registered trademark of Apple Computer Co.

Listed below are products currently manufactured by Vista: • V1000 Dual 8" Disk Drives • V1100 Dual Thinline 8" Disk Drives • V1200 6MB Cartridge Drive • Solo 5¼" 35 Track Disk Drive single sided full height • Duet 5¼" 40 Track Disk Drive double sided half-height • Quartet 5¼" Dual double sided Disk Drive in side-by-side chassis • A800 DMA Double Density Disk Controller for 8" Disk Drives • Apple III™ Time Card • Keyboard Buffer • GB75 Interface • Vision 80 80 x 24 Card • Vision 40 Softscreen Interface • Vision 20 Character Generator • IBM® 576K Maxicard • PC Master • PC Extender • PC Extender Plus • PC Expander • PC Expander Plus • IBM® Multicard.

Continue to watch Vista Computer Co., Inc., as their line expands to meet all your future computer needs.

Contact Your Local Vista Dealer or Call our Vista Hotlines.

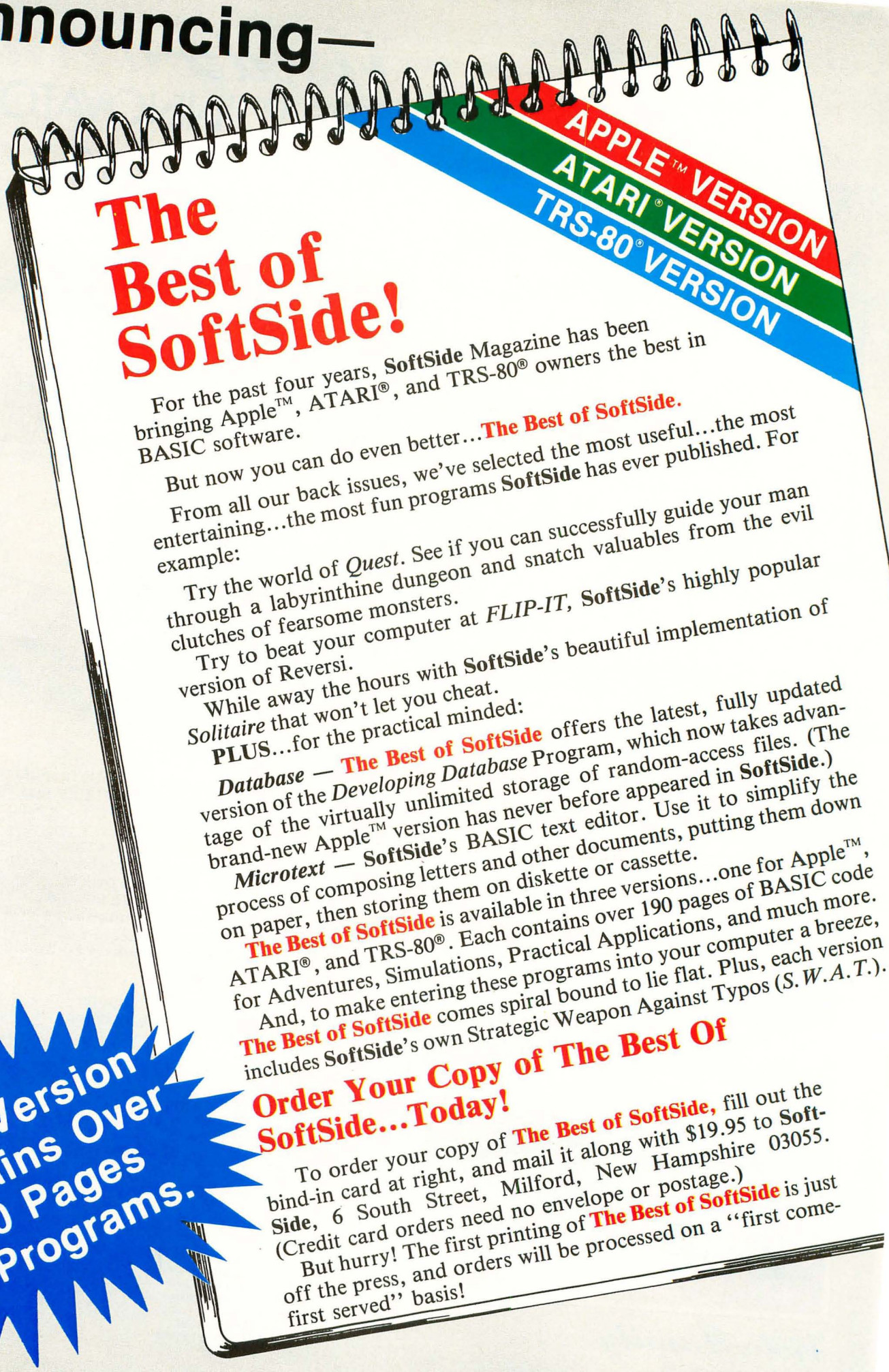
Vista COMPUTER COMPANY, INC.

1317 East Edinger / Santa Ana, CA 92705
(714) 953-0523 / (800) 854-8017

DISTRIBUTORS / REPRESENTATIVES

Western - Group 3 Wholesale
(213) 973-7844 (408) 732-1307
South Central - M.P. Systems
(214) 385-8885
UK - Hal Computers Ltd.
(0252) 517175 / TWX: 858404
Northeast - Computers & Peripherals Inc.
(315) 476-6664
Florida - Audio Marketing (REP)
(305) 322-8327
Central - Wyatt & Associates
(317) 773-4791

Announcing—



APPLE™ VERSION
ATARI® VERSION
TRS-80® VERSION

The Best of SoftSide!

For the past four years, **SoftSide** Magazine has been bringing Apple™, ATARI®, and TRS-80® owners the best in BASIC software.

But now you can do even better…**The Best of SoftSide.**

From all our back issues, we've selected the most useful…the most entertaining…the most fun programs **SoftSide** has ever published. For example:

Try the world of *Quest*. See if you can successfully guide your man through a labyrinthine dungeon and snatch valuables from the evil clutches of fearsome monsters.

Try to beat your computer at *FLIP-IT*, **SoftSide**'s highly popular version of Reversi.

While away the hours with **SoftSide**'s beautiful implementation of *Solitaire* that won't let you cheat.

PLUS…for the practical minded:

***Database* — The Best of SoftSide** offers the latest, fully updated version of the *Developing Database* Program, which now takes advantage of the virtually unlimited storage of random-access files. (The brand-new Apple™ version has never before appeared in **SoftSide**.)

***Microtext* — SoftSide**'s BASIC text editor. Use it to simplify the process of composing letters and other documents, putting them down on paper, then storing them on diskette or cassette.

The Best of SoftSide is available in three versions…one for Apple™, ATARI®, and TRS-80®. Each contains over 190 pages of BASIC code for Adventures, Simulations, Practical Applications, and much more.

And, to make entering these programs into your computer a breeze, **The Best of SoftSide** comes spiral bound to lie flat. Plus, each version includes **SoftSide**'s own Strategic Weapon Against Typos (*S.W.A.T.*).

Order Your Copy of The Best Of SoftSide…Today!

To order your copy of **The Best of SoftSide,** fill out the bind-in card at right, and mail it along with $19.95 to **SoftSide**, 6 South Street, Milford, New Hampshire 03055. (Credit card orders need no envelope or postage.)

But hurry! The first printing of **The Best of SoftSide** is just off the press, and orders will be processed on a "first come-first served" basis!

Each Version Contains Over 190 Pages Of Programs.

Hopper *continued*

```
13310 DOWN$(11,11)=CHR$(94):FOR J=12 T
O 15:DOWN$(J,J)=CHR$(136):NEXT J
13320 FOR J=18 TO 21:DOWN$(J,J)=CHR$(2
00):NEXT J:DOWN$(24,24)=CHR$(136)
13990 RETURN
```

Data for graphics display of each board. Note that the lower case characters in these lines are CONTROL characters. Actually, every character in this block is a control character.

```
14100 DATA fghhhhhhizzzzfghhhhhhizzzfg
hhhhhhi
14110 DATA lmmmmmmnozzzzlmmmmmmnozzzlm
mmmmno
14120 DATA fghhhhhhizzzzfghhhhhhizzzfg
hhhhhhi
14130 DATA lmmmmmmnozzzzlmmmmmmnozzzlm
mmmmno
14140 DATA ,abzzzzz,ab
14150 DATA cdezzzzzcde
14160 DATA ,abzzzzz,ab
14170 DATA cdezzzzzcde
14200 DATA fghhhhhizzzzzfghhhhhizzzzfg
hhhhhi
14210 DATA pqqqqqqzzzzzzlmmmmmmnozzzlm
mmmno
14220 DATA fghhhhhhhizzzzzzzzzzzzzzfgh
hhhhhhi
14230 DATA lmmmmnozzzzzzlmmmmmmnozzzlm
mmmno
14240 DATA ,abzzzzz,abzzzzz,ab
14250 DATA wxyzzzzzzzzzzwxy
14260 DATA tuvzzzzzzzzzz,ab
14270 DATA cdezzzzzcdezzzzzcde
14300 DATA fghhhhizzzzzrrrrrszzzzfghhh
hi
14310 DATA lmnnnnnnnozzzzzlhmno
14320 DATA fghhhhhhizzzzzzzzzzzzzzfghh
hhhhi
14330 DATA pqqqqqzpqqqqqzzzzzzzzpqqqqq
14340 DATA ,abzzzz,abzzzz,abzzzz,ab
14350 DATA zwxyzzzzzzzzzzwxy
14360 DATA tuvzzztuvzzzzzzzzzz,ab
14370 DATA zcdezzzzcdezzzzcdezzzzcde
14400 DATA fhhhhizzzzzrrrrrszzzzfghhhi
14410 DATA lmnnnozzzzlmnnnozzzzlmnnno
14420 DATA lmnnnnnnnnno
14430 DATA pqqqqqzpqqqqqzzzzpqqqqq
14440 DATA ,abzzz,abzzzzzz,abzzz,abzzz
z,ab
14450 DATA zwxyzzzwxyzzzwxy
14460 DATA tuvzzztuvzzzzzzzzzzz,abzzz,
ab
14470 DATA zwxyzzcdezzwxyzzzzzzzcdezzz
wxyz
14500 DATA fhhhhizzzzzrrrrrszzzzfghhhi
14510 DATA lmnnozzzzlmnnozzzzlmnno
14520 DATA lmnnnnnnnno
14530 DATA pqqqqqzpqqqqqzzzzpqqqqq
14540 DATA ,abzzz,abzzzzzz,abzzz,abzzz
z,ab
14550 DATA zwxyzzzwxyzzzwxy
14560 DATA tuvzzztuvzzzzzzzzzzz,ab
14570 DATA zwxyzzcdezzwxyzzzzzzzcdezzz
wxyz
```

Data for the speed on each line on each board.

```
14910 DATA 0,2,0,2,0,2,0,2
14920 DATA 0,2,0,2,0,2,0,2
14930 DATA 0,2,0,2,0,3,0,2
14940 DATA 0,2,0,3,0,3,0,2
14950 DATA 1,2,0,3,0,3,1,2
```

Subroutine to display instructions. Note that the lower case letters in lines 15110 through 15140 are CONTROL characters.

```
15000 POSITION N2,N0:? "     HOPPER";S
P$(N1,9);
15010 POSITION 12,N2:? "A GAME OF SKIL
L.":POSITION N5,N5:? "MOVE JOYSTICK FO
R INSTRUCTIONS"
15020 ? "   OR PRESS BUTTON TO START G
AME."
15030 POSITION N1,21:? "WRITTEN BY RIC
H BOUCHARD AND ALAN ZETT"
```

SOFTSIDE ORDERING INFORMATION

FORM OF PAYMENT

USA

VISA, MasterCard, certified checks, money orders and personal checks are accepted.

Canada/Mexico

The preferred method of payment is by VISA or MasterCard. A bank check is acceptable if it has been preprinted for payment in U.S. dollars. No personal or company checks accepted.

Other Foreign Orders

Payment must either be by a bank check drawn on a U.S. bank payable in U.S. dollars or by affiliated bank credit cards of VISA or MasterCard.

GUARANTEE

All software is guaranteed to load and run. If you experience difficulties with the product within 30 days, it may be returned for replacement. Send your properly protected tape or disk to the attention of the Customer Service Representative and include your name, address, and the reason it is being returned.

LIABILITY

All software is sold on an as-is basis. **SoftSide** assumes no liability for loss or damage caused or alleged to be caused directly or indirectly by products sold or exchanged by them or their distributors, including, but not limited to, any interruption in service, loss of business or anticipatory profits or consequential damages resulting from use or operation of such software.

PRICES

Prices are subject to change without notice. We are not responsible for typographical errors.

Unless otherwise noted in a published advertisement, the following prices are in effect as of this issue:

	USA/Canada APO/FPO	USA/Canada FIRST CLASS Mexico	Other Foreign
SoftSide Magazine (yr)	$30	$40	$62
SoftSide Magazine (6 mo.)	$15	$20	$31

	USA APO/FPO	Mexico Canada	Other Foreign
CV (year) & magazine	$75	$95	$125
(6 mo.)	$40	$50	$70
DV (year) & magazine	$125	$145	$175
(6 mo.)	$70	$80	$90

Adventure of the Month	Disks	Cassettes
3-Mo. Trial	$29	$19
6-Mo.	$49	$29
12-Mo.	$89	$49
24-Mo. Charter	$169	$89

BACK ISSUES

Minimum order for magazines only — 3 issues. There is no minimum order for magazine/media combinations.
Price includes shipping to the 48 states only. Alaska, Hawaii, Puerto Rico, APO/FPO, and ALL foreign orders — postage is additional.

ALL Foreign orders and all magazine/media combination orders — Order directly from **SoftSide, 6 South St., Milford, NH 03055.**

```
15040 POSITION N5,23:? "(C) 1982  SOFT
SIDE PUBLICATIONS";
15050 IF STRIG(N0)=N0 THEN GOSUB 2200:
RETURN
15060 IF STICK(N0)=15 THEN 15050
15070 GOSUB 2200
15100 POSITION N2,N3:? "YOUR MISSION:
MOVE FROGS INTO BUNKERS":? :?
15110 ? "fghhhhhhhi  MOVE ONTO LOGS lm
nnnnnnno"
15120 ? :? "rrrrrs  AVOID SNAKE HEADS,
 USE BODIES"
15130 POSITION N10,15:? ",ab   AVOID C
ARS   cde"
15140 ? :? " tuv   WATCH OUT FOR FASTE
R CARS   wxy"
15150 POSITION N8,21:? "MOVE JOYSTICK
TO CONTINUE"
15160 IF STICK(N0)=15 THEN 15160
15170 GOSUB 2200
15180 POSITION N2,N5:? "THE  JOYSTICK
IS  USED TO  MOVE  THE  FROGS.  A TIME
  BAR IS  DISPLAYED AT"
15190 ? "THE BOTTOM OF THE SCREEN.  WH
EN THIS      RUNS OUT, YOUR FROG EXPIR
ES."
15200 POSITION 16,13:? "SCORING":? :?
"  EACH FORWARD HOP:              20":?
"  REACHING A BUNKER:            250"
15210 ? "       + BONUS FOR REMAINING
TIME":? "  5 FROGS REACHING BUNKERS:
2000"
15220 POSITION N7,21:? "PRESS BUTTON T
O START GAME"
15230 IF STRIG(N0)=N1 THEN 15230
15240 GOSUB 2200:RETURN
```

Subroutine to create redefined character set.

```
30000 POKE 106,PEEK(106)-5:START=(PEEK
(106)+1)*256:POKE 756,START/256:POKE 7
52,1
30010 DIM XFR$(38):XFR$="h),KM)'NZj
xiaL ,1MKHPyfLfNZNIdP@'"
30020 Z=USR(ADR(XFR$)):RESTORE 30100
30030 READ X:IF X=-1 THEN RESTORE :RET
URN
30040 FOR Y=0 TO 7:READ Z:POKE X+Y+STA
RT,Z:NEXT Y:GOTO 30030
30100 DATA 512,0,0,0,127,127,127,20,20
30101 DATA 520,31,17,17,255,255,85,1,1
30102 DATA 528,0,0,192,252,250,252,64,64
30103 DATA 536,0,0,2,42,58,42,3,3
30104 DATA 544,168,136,136,170,170,170
,128,128
30105 DATA 552,0,0,0,170,170,170,56,56
30106 DATA 560,10,34,168,168,168,168,3
4,10
30107 DATA 568,170,170,138,168,170,162
,138,170
30108 DATA 576,170,170,42,168,170,162,
138,170
30109 DATA 584,168,168,42,170,170,138,
40,168
30110 DATA 592,1,17,17,5,5,5,17,17
30111 DATA 600,192,196,68,80,80,80,68,
68
30112 DATA 608,42,42,168,170,170,162,4
0,42
30113 DATA 616,170,170,168,42,170,138,
162,170
30114 DATA 624,170,170,162,42,170,138,
162,170
30115 DATA 632,128,136,42,42,42,42,136
,160
30116 DATA 640,248,220,252,28,24,48,10
3,254
30117 DATA 648,0,0,0,0,0,248,155,31
30118 DATA 656,0,0,0,0,0,31,217,248
30119 DATA 664,31,59,63,56,24,12,230,1
27
30120 DATA 672,0,0,0,85,170,170,10,10
30121 DATA 680,0,8,2,84,170,170,2,2
30122 DATA 688,0,0,0,160,168,168,128,1
28
30123 DATA 696,0,0,0,21,85,85,5,5
30124 DATA 704,0,16,64,42,85,85,0,0
30125 DATA 712,0,0,0,170,85,85,80,80
30126 DATA 720,0,0,0,0,0,0,0,0
30127 DATA 776,0,0,0,170,170,170,160,1
60
30128 DATA 784,0,0,0,170,170,170,0,0
30129 DATA 792,0,0,0,170,170,170,10,10
30130 DATA 800,160,160,170,170,170,0,0,0
30131 DATA 808,0,0,170,170,170,0,0,0
30132 DATA 816,10,10,170,170,170,0,0,0
30133 DATA 824,0,0,0,1,1,0,0,0
30134 DATA 832,0,0,0,128,128,0,0,0
30135 DATA 840,0,0,48,5,5,48,0,0
30136 DATA 848,0,0,48,64,64,48,0,0
30137 DATA 856,25,13,0,229,5,0,13,25
30138 DATA 864,140,152,0,92,64,0,152,1
40
30139 DATA 872,25,13,0,224,0,0,13,25
30140 DATA 880,140,152,0,28,0,0,152,14
0
30141 DATA 888,0,0,0,255,0,0,0,0
30142 DATA 896,170,170,170,170,170,170
,170,170
30143 DATA 904,255,255,255,255,255,255
,255,255
30144 DATA 912,85,85,85,85,85,85,85,85
30145 DATA 920,160,160,160,160,160,160
,160,160
30146 DATA 928,10,10,10,10,10,10,10,10
30147 DATA 936,17,17,5,5,5,17,17,1
30148 DATA 944,68,68,80,80,80,68,196,1
92
30149 DATA -1
```

ATARI® SWAT TABLE FOR: HOPPER - ATARI VERSION

(Modified Parameters: NU = 6, B = 500)

LINES	SWAT CODE	LENGTH
10 - 140	MF	271
150 - 205	KA	195
210 - 250	PK	370
260 - 1000	QI	282
1010 - 1100	CK	237
1110 - 1200	EA	366
1202 - 1230	GO	172
1240 - 1290	GD	206
1300 - 1500	SL	323
1510 - 1550	TC	322
1560 - 1610	AZ	325
1620 - 1700	XH	196
1710 - 2020	XR	206
2100 - 2320	RA	374
2330 - 2400	HR	220
2410 - 2460	SE	290
2480 - 2540	GI	153
2550 - 10030	RH	401
10040 - 10100	OE	270
10110 - 11010	UK	234
11020 - 11070	LN	171
11080 - 11130	RV	200
11140 - 11190	KT	220
11200 - 12020	JG	230
12025 - 12070	KP	150
13000 - 13045	MC	200
13050 - 13075	CW	174
13080 - 13105	JG	180
13110 - 13140	LN	180
13145 - 13230	TF	181
13240 - 14100	ID	230
14110 - 14160	RM	171
14170 - 14240	HC	197
14250 - 14320	PV	168
14330 - 14400	UV	178
14410 - 14460	XO	173
14470 - 14540	SK	182
14550 - 14930	CI	152
14940 - 15030	TE	257
15040 - 15110	ZO	218
15120 - 15170	AH	223
15180 - 15230	IC	423
15240 - 30040	RY	282
30100 - 30105	MW	203
30106 - 30111	DD	221
30112 - 30117	CQ	222
30118 - 30123	RN	190
30124 - 30129	FT	185
30130 - 30135	VP	180
30136 - 30141	OG	187
30142 - 30147	CI	218
30148 - 30149	VQ	43

```
 SS SS SS SS SS SS SS SS SS SS SS
 SS                            SS
 SS        ATARI BASIC         SS
 SS  'HOPPER - CTIA CHANGES'   SS
 SS   AUTHOR: HOWARD WOLKOW    SS
 SS   TRANSL: RICH BOUCHARD    SS
 SS      AND ALAN J. ZETT      SS
 SS    COPYRIGHT (C) 1982      SS
 SS SOFTSIDE PUBLICATIONS, INC SS
 SS                            SS
 SS SS SS SS SS SS SS SS SS SS SS
```

```
1410 SOUND N3,NO,NO,NO:POKE 704,216:FO
R J=N1 TO 200:NEXT J:POKE PX,NO
11150 POKE 1774,208:POKE 1775,178
11160 POKE 1776,144:POKE 1777,128
12025 POKE 704,216
30106 DATA 560,5,17,84,84,84,84,17,5
30107 DATA 568,85,85,69,84,85,81,69,85
30108 DATA 576,85,85,21,84,85,81,69,85
30109 DATA 584,84,84,21,85,85,69,20,84
30110 DATA 592,3,35,34,10,10,10,34,34
30111 DATA 600,128,136,136,160,160,160
,136,136
30112 DATA 608,42,42,168,170,170,162,4
0,42
30113 DATA 616,85,85,84,21,85,69,81,85
30114 DATA 624,85,85,81,21,85,69,81,85
30115 DATA 632,64,68,21,21,21,21,68,80
30127 DATA 776,0,0,0,85,85,85,80,80
30128 DATA 784,0,0,0,85,85,85,0,0
30129 DATA 792,0,0,0,85,85,85,5,5
30130 DATA 800,80,80,85,85,85,0,0,0
30131 DATA 808,0,0,85,85,85,0,0,0
30132 DATA 816,5,5,85,85,85,0,0,0
30135 DATA 840,0,0,96,10,10,96,0,0
30136 DATA 848,0,0,48,128,128,48,0,0
30137 DATA 856,25,13,0,202,10,0,13,25
30138 DATA 864,134,140,0,167,160,0,140
,134
30139 DATA 872,25,13,0,192,0,0,13,25
30140 DATA 880,134,140,0,7,0,0,140,134
30142 DATA 896,85,85,85,85,85,85,85,85
30144 DATA 912,170,170,170,170,170,170
,170,170
30145 DATA 920,80,80,80,80,80,80,80,80
30146 DATA 928,5,5,5,5,5,5,5,5
30147 DATA 936,34,34,10,10,10,34,35,3
30148 DATA 944,136,136,160,160,160,136
,136,128
```

Translation of the Month Contest

You could win a FREE *SoftSide* DV or CV Subscription!

SoftSide's* Translation of the Month** has been so well received by our readers, we're offering a greater author incentive than ever before. No, we can't give you a job at the U.N., but we will award a one-year subscription to ***SoftSide DV or an 18-month subscription to ***SoftSide CV*** for a high-quality translation of one of our past programs. That's a value of $125 for the ***Disk Version*** or $112.50 for the ***Cassette Version*** — you'll be rewarded every month for your translation efforts!

Here are some of the most important qualifications we look for in a translation winner.

Your entry must be a translation of one of the featured programs from a past issue of ***SoftSide*** (We're particularly interested in Apple™, ATARI®, and IBM® PC translations of some of our older TRS-80® only issues. Write for a list of suggested candidates.) In general, we're looking for translations of programs which are a CHALLENGE to translate. Some of the programs we publish are written in more or less "generic" BASIC, which can be typed into another computer with very few changes. Although these programs require the least effort to translate, they are also the least likely candidates for contest winners.

Your translation should be thoroughly tested and completely bug-free. Just converting program lines doesn't automatically ensure a workable translation. Be sure to use-test your translation as carefully as you would test a program you had written entirely from scratch.

Your translation should fully utilize the unique features of the computer for which it is written. The objective of a translation is to "fit" the capability and convention of its host computer, not simply mechanically duplicate the operation of the original program. This is especially true of programs which use graphics, and should be kept in mind for such minor features as keyboard layout (use of such special keys as arrows, ESC, CTRL, CLEAR, etc.). Also be careful with screen formatting; a word that spills over into the next line because of a PRINT statement that wasn't properly rewritten betrays such carelessness that we'll probably reject your translation automatically.

Your entry should incorporate any improvements and enhancements you can add to the original program. Don't feel that you have to limit yourself to the boundaries of the original. (On the other hand, don't go overboard and destroy the character of the original by completely rewriting it!) An enhanced translation is much more likely to catch our attention than a line-for-line duplicate, and it will have more value to our readers.

It's not necessary to include extensive documentation with your translation, only that which is different from the original. If most of the originally published documentation applies to your translation, simply say so. You should, however, include descriptions and explanations of any changes or enhancements you've made.

All **Translation Contest** entries must be submitted on disk, with documentation in printed or typed form. Disks will be returned only if accompanied by a self-addressed, stamped envelope. Send your entries to:

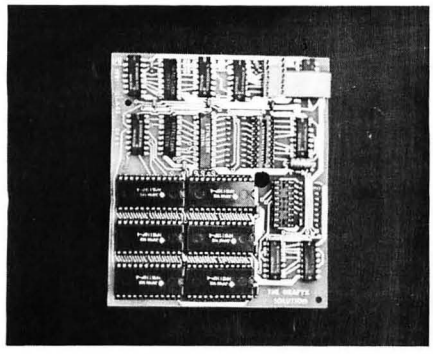
SoftSide

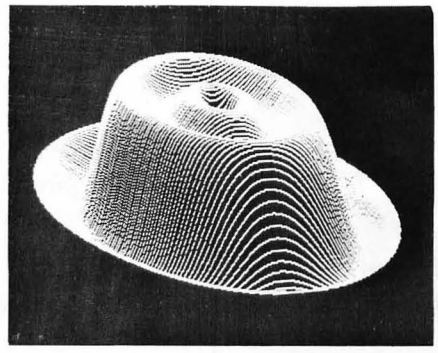
Translation Contest
6 South Street, Milford, NH 03055

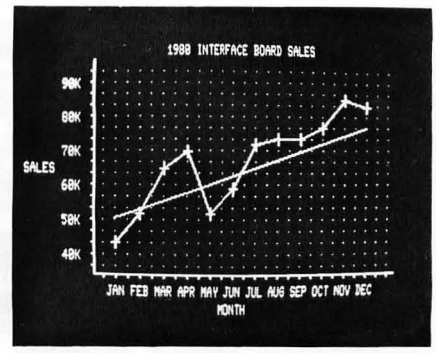
ATARI® SWAT TABLE FOR:
HOPPER - CTIA ATARI
(Modified Parameters: NU = 6, B = 500)

LINES	SWAT CODE	LENGTH
10 - 140	MF	271
150 - 205	KA	195
210 - 250	PK	370
260 - 1000	QI	282
1010 - 1100	CK	237
1110 - 1200	EA	366
1202 - 1230	GO	172
1240 - 1290	GD	206
1300 - 1500	SA	323
1510 - 1550	TC	322
1560 - 1610	AZ	325
1620 - 1700	XH	196
1710 - 2020	XR	206
2100 - 2320	RA	374
2330 - 2400	HR	220
2410 - 2460	SE	290
2480 - 2540	GI	153
2550 - 10030	RH	401
10040 - 10100	OE	270
10110 - 11010	UK	234
11020 - 11070	LN	171
11080 - 11130	RV	200
11140 - 11190	GP	226
11200 - 12020	JG	230
12025 - 12070	KE	150
13000 - 13045	MC	200
13050 - 13075	CW	174
13080 - 13105	JG	180
13110 - 13140	LN	180
13145 - 13230	TF	181
13240 - 14100	ID	230
14110 - 14160	RM	171
14170 - 14240	HC	197
14250 - 14320	PV	168
14330 - 14400	UV	178
14410 - 14460	XO	173
14470 - 14540	SK	182
14550 - 14930	CI	152
14940 - 15030	TE	257
15040 - 15110	ZO	218
15120 - 15170	AH	223
15180 - 15230	IC	423
15240 - 30040	RY	282
30100 - 30105	MW	203
30106 - 30111	UP	203
30112 - 30117	VH	204
30118 - 30123	RN	190
30124 - 30129	IW	172
30130 - 30135	CM	169
30136 - 30141	VO	191
30142 - 30147	FA	205
30148 - 30149	GI	49

continued on page 46



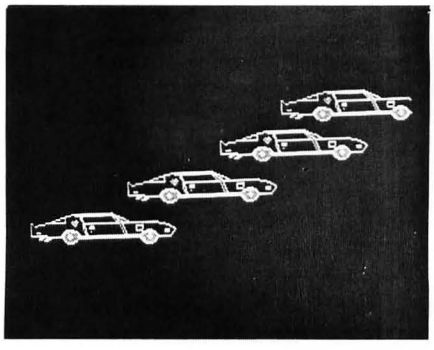




Mod III **GRAFYX SOLUTION**™ $299.95 ↙464

- 512 × 192 Dot Graphics increases resolution 16× (better than Apple)
- Extensive Business, Personal, Educational, and Scientific applications
- 14 BASIC commands set and reset Points, Lines, Circles, Boxes, etc.
- Allows Display and Printout of detailed Line, Bar, and Pie graphs
- 80 Character/line display compatible with BASIC programs and DOS
- Price includes 98K bits memory, 30 programs, 52 page manual
- Plug-in, clip-on board eliminates soldering for easy installation

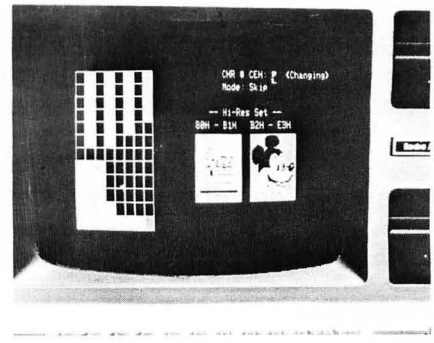
---



REBATE **$40** REBATE

VALID 11-1 TO **12-31-82**

After purchasing Grafyx Soln. or 80-Grafix, return this coupon with registration form to receive rebate.



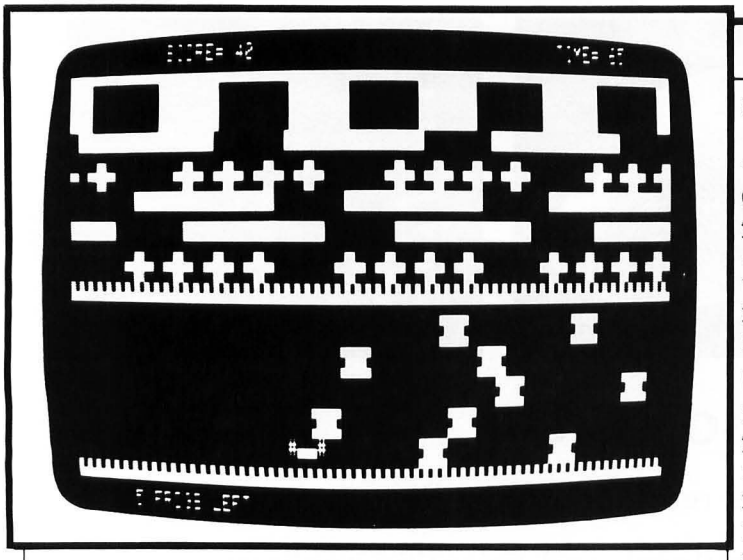
Mod I, III **80-GRAFIX**™ $169.95 ↙465

- Create and Save 128 programmable hi-resolution characters/shapes
- Characters are generated within an 8 × 12 matrix (Mod I: 6 × 12)
- Great for creating game figures, symbols, and alternate character sets
- Excellent software support (20 programs) & extensive documentation
- Module easily installs within the computer with no soldering

Manual only $15
Products guaranteed
Dealers welcome

MICRO-LABS, INC. 214-235-0915
902 Pinecrest, Richardson, Texas 75080

FREE shipping
Prepaid or COD
Tx. res add 5%



```
SS SS SS SS SS SS SS SS SS SS SS SS
SS                                SS
SS          TRS-80 BASIC          SS
SS            'HOPPER'            SS
SS      AUTHOR: HOWARD WOLKOW     SS
SS       COPYRIGHT  (c) 1982      SS
SS  SOFTSIDE PUBLICATIONS, INC.   SS
SS                                SS
SS SS SS SS SS SS SS SS SS SS SS SS
```

Initialization.

```
10 CLS:CLEAR3000:GOSUB1000:GOTO60
15 GOSUB490:GOTO60
```

Move on log or lily.

```
20 P=P+XP(XP):RETURN
```

Check for car crash.

```
30 IFXP<>15ANDXP>9ORXP=3THENFORX=-2TO1STEP3:K=PEEK(X+VM+P):IFK>1
28THENGOSUB240RETURN:ELSENEXT:RETURN
40 IFXP=15ORXP=9THENRETURN
50 FORX=-2TO1STEP3:IFPEEK(X+VM+P)=32THENGOSUB240ELSENEXT:RETURN
```

Adjust positions of logs, lily pads, and frog.

```
60 FR=FR-3:FL=FL+3:R=R-2:L=L+2:M=M-1:PRINT@10,"SCORE=";AM;:PRINT
@50,"TIME=";M;
70 IFFR<1THENFR=63
80 IFFL>63THENFL=1
90 IFR<1THENR=63
100 IFL>63THENL=1
110 IFM=0THENGOTO240
120 PRINT@P-2,F$;
```

Print updated board.

```
130 PRINT@192,MID$(SL(B),R,63);:PRINT@256,MID$(ST(B),L,63);:PRIN
T@320,MID$(SL(B),FR,63);:PRINT@384,MID$(SL(B),L,63);:PRINT@448,M
ID$(ST(B),FR,63);:PRINT@512,G$;:PRINT@896,G$;
140 PRINT@576,MID$(SA(B),R,63);:PRINT@640,MID$(SA(B),L,63);:PRIN
T@704,MID$(SA(B),FR,63);:PRINT@768,MID$(SA(B),FL,63);:PRINT@832,
MID$(SA(B),R,63);:PRINT@896,G$;:GOSUB30:GOSUB20:PRINT@P-2,F$;
```

Read keyboard for player's move.

```
150 A=PEEK(14400):IFA=0THENGOTO220
160 P=P+M(A):XP=XP+XM(A):YP=YP+YP(A)
210 AM=AM+20:PRINT@960,"";:SOUND35,3:SOUND100,3:GOSUB30:IFXP=3TH
ENAM=AM+500:M=80:AM=AM+(M*5):PRINT@P-2,F$:H=H+1:U(H)=P-2:XP=15:P
=920:IFH=5THENAM=AM+2000:D=D+1:GOSUB460:GOTO60
215 IFA>=32THENAM=AM-20
220 IFYP<2ORYP>62THEN240
230 GOTO60
```

TRS-80®

```
240 PRINT@965,D-1;"FROGS LEFT";
245 A=INT((P-2)/64):X=P-2-A*64:X=X+(X-59)*(X>59):PRINT@A*64+X,"#
###";:FORX=1TO35:SOUNDX,10:NEXT:YP=24:PRINT@960,"";
```

Check if game is over.

```
250 FORX=1TO1000:NEXT:D=D-1:IFD=0THENGOTO270ELSEP=920:M=80:XP=15
:IFH=0THENGOTO60ELSECLS:PRINT@64,T$:FORX=1TOH:PRINT@U(X),F$;:NEX
T:GOTO60
260 FORX=1TOH:PRINT@U(H);F$:NEXT
270 PRINT@538,"GAME OVER":FORX=1TO9000:NEXT
```

High score routine.

```
280 M=0:FORI=1TO10:D=0
290 IFAM>HS(I)THENM=I
300 NEXT:IFM=0THEN380
310 CLS:PRINT@153,"CONGRADULATIONS;";
320 PRINT@208,"YOUR SCORE IS IN THE TOP TEN.";
330 PRINT@278,"PLEASE ENTER YOUR NAME";
340 SC="":PRINT@479,;:INPUTSC:IFSC=""THEN340
350 CLS:SC=LEFT$(SC,15):D=5
360 FORI=1TOM:HS(I-1)=HS(I):HS$(I-1)=HS$(I):NEXT
370 HS(M)=AM:HS$(M)=SC
380 CLS:PRINTCHR$(23);"TOP TEN SCORES":PRINT
390 FORI=10TO1STEP-1
400 IFHS$(I)>""THENPRINTHS$(I),HS(I)ELSEPRINT
410 NEXT
420 D=5:PRINT:PRINT"PRESS <ENTER> TO PLAY AGAIN"
430 IFINKEY$<>CHR$(13)THEN430
440 AM=0
450 B=0:GOSUB460:GOTO60
```

Set up new board.

```
460 CLS:XP=15:B=B+1:P=920:VM=15360:PRINT@10,"SCORE=";AM;:PRINT@5
0,"TIME=";M;:PRINT@64,T$;:PRINT@512,G$;:PRINT@896,G$;:R=RND(32)+
10:L=RND(32)+10:FR=RND(32)+10:FL=RND(32)+10:M=80:YP=24
470 FORN=1TO5:U(N)=0:NEXT:H=0:IFB=4THENB=3:RETURN
480 PRINT@965,D;"FROGS LEFT";:RETURN
```

Initialization.

```
490 DEFSTRC,S-T:DEFINTB,D-G,I-R,U-Z:DIMXP(15),M(255),XM(255),YP(
255):D=5:VM=15360:CL=STRING$(14,143):CT=CHR$(140)+CHR$(191)+CHR$
(140)+CHR$(128):CT=CT+CT+CT+CT:CA=CHR$(162)+CHR$(191)+CHR$(191)+
CHR$(145)
500 CB=STRING$(64,32):SA(1)=LEFT$(CB,18)+CA+LEFT$(CB,10)+CA+LEFT
$(CB,27):ST(1)="  "+CT+LEFT$(CB,5)+CT+LEFT$(CB,5)+CT+"    ":SL(1)
="   "+CL+LEFT$(CB,7)+CL+LEFT$(CB,7)+CL+"     "
510 SA(3)=" "+CA+"    "+CA+"    "+CA+"        "+"    "+CA+"    "
+CA+"    "+"        "+CA+"  ":ST(3)=LEFT$(CB,47)+CT:SL(3)=LEFT$(
CB,49)+CL
520 SL(2)=SL(3):ST(2)=CT+LEFT$(CB,15)+CT+LEFT$(CB,16):SA(2)=LEFT
$(SA(3),32)+STRING$(32,32)
530 F$="#"+CHR$(140)+CHR$(140)+"#":T$=CHR$(151)+STRING$(6,131)+C
HR$(171)+CHR$(26)+STRING$(8,24)+CHR$(149)+"      "+CHR$(170)+CHR
$(27):D$=T$
540 CS=STRING$(5,191)+CHR$(26)+STRING$(5,24)+STRING$(5,191)+CHR$
(27):CP=CHR$(191)+CHR$(24)+CHR$(26)+CHR$(191)+CHR$(27):T$=CP+CP+
T$+CS+T$+CS+T$+CS+T$+CS+T$+CP:;G$=STRING$(63,141)
550 FORX=1TO3:SA(X)=SA(X)+SA(X):ST(X)=ST(X)+ST(X):SL(X)=SL(X)+SL
(X):NEXTX
555 XP(4)=2:XP(5)=-2:XP(6)=3:XP(7)=-2:XP(8)=3:M(32)=-5:M(8)=-64:
M(16)=64:M(64)=5:XM(16)=1:XM(8)=-1:YP(32)=-5:YP(64)=5
560 CLS:PRINTTAB(13)H$:PRINT
565 PRINTTAB(24);"A GAME OF SKILL":PRINT:PRINTTAB(26);"BY H. WOL
KOW":PRINT:PRINT:PRINTTAB(21);"PRESS <ENTER> TO START":PRINT
570 A$=INKEY$:IFA$=""THEN570ELSEGOSUB460:GOTO60
```

Data for "HOPPER" logo.

```
1000 DATA 191,149,32,170,191,32,32,191,151,131,171,191,32,32
1010 DATA 191,151,131,171,191,32,32,191,151,131,171,191,32,32
1020 DATA 191,151,131,131,131,32,32,191,151,131,171,191,-1
1030 DATA 191,157,140,174,191,32,32,191,149,32,170,191,32,32
1040 DATA 191,151,131,131,131,32,32,191,151,131,131,131,32,32
1050 DATA 191,157,140,140,32,32,32,191,151,139,147,131,-1
1060 DATA 191,149,32,170,191,32,32,191,181,176,186,191,32,32
1070 DATA 191,149,32,32,32,32,32,191,149,32,32,32,32,32
1080 DATA 191,181,176,176,176,32,32,191,149,32,130,164,-1
```

Display logo.

```
1090 PRINTTAB(12);" ";
1100 N=0:FORX=1TO3
1110 READA:IFA=-1THEN1120:ELSEH$=H$+CHR$(A):PRINTCHR$(A);:N=N+1:
GOTO1110
1120 H$=H$+CHR$(26)+STRING$(N,24):PRINTCHR$(26)+STRING$(N,24);:N
=0
1130 NEXT
1140 Y=0:FORX=1TO500:NEXT
1150 Y=Y+2:IFY=14THEN1180ELSECLS:PRINT@Y*64+13,H$
1160 FORX=1TO500:NEXT
1170 GOTO1150
1180 Y=Y-2:IFY=4THEN60000ELSEPRINT:PRINT:FORX=1TO500:NEXT:GOTO11
80
```

Machine Language sound routine.

```
60000 Z=0:FORX=1TO158:READY:Z=Z+Y:NEXT:IFZ<>15204THENCLS:PRINT"D
ATA BASE ERROR IN LINES 60060-60160, CHECK LISTING.":PRINT:LIST6
0060-60160ELSEY=86:X=255:POKE-1,0:IFPEEK(-1)<>0THENX=191:POKE-16
385,0:IFPEEK(-16385)<>0THENX=127
60010 POKE16562,X:POKE16561,Y:A1=PEEK(16561)+2:A2=PEEK(16562):A=
A1+A2*256:Z=A-1:RESTORE:FORX=1TO123:READZ9:NEXT:FORX=1TO158:Z=Z+
1:Z=Z+65536*(Z>32767)
60020 READY:IFY<0THENY=A1+ABS(Y):POKEZ,Y+256*(Y>255):Z=Z+1:POKEZ
,A2-(Y>255):NEXTELSEPOKEZ,Y:NEXT
60030 IFPEEK(16396)=201POKE16526,A1:POKE16527,A2ELSECMD"T":DEFUS
R=A1+(A2+256*(A2>127))*256:POKE14308,0
60040 IFPEEK(16807)+PEEK(16808)*256<>A+24THENA=USR(0)
60050 GOTO490
60060 DATA58,166,65,50,-164,42,167,65,34,-165,62,195,50
60070 DATA166,65,33,-24,34,167,65,201,245,123,254,2,40,4,254
60080 DATA16,32,79,229,213,42,230,64,126,183,32,4,35,35,35,35
60090 DATA215,6,5,17,-156,26,190,32,104,19,35,16,248,43,215
60100 DATA43,34,230,64,241,241,241,241,197,213,215,205,55,35
60110 DATA229,205,127,10,42,33,65,34,-167,225,215,43,34,230,64
60120 DATA35,205,55,35,43
```

TRS-80® SWAT TABLE FOR: HOPPER

LINES	SWAT CODE	LENGTH
10 - 110	QF	317
120 - 215	MS	514
220 - 320	PB	435
330 - 440	PR	310
450 - 500	SZ	534
510 - 550	FW	508
555 - 1040	WS	537
1050 - 1160	HH	418
1170 - 60040	ZG	535
60050 - 60140	LN	514
60150 - 60160	UN	84

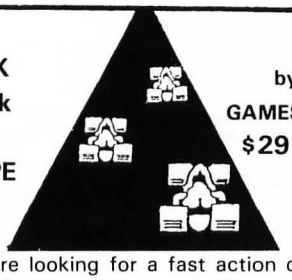
Royal Software

WE DEAL EXCLUSIVELY IN PRODUCTS
FOR THE ATARI (THE BEST) COMPUTER

YOUR MARKETPLACE FOR:

* ATARI

BAJA BUGGY



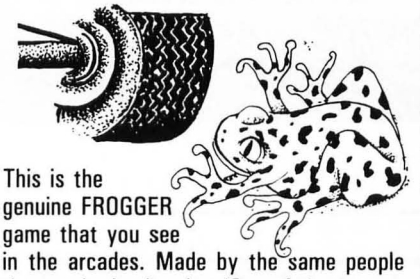
16K Disk OR TAPE

by GAMESTAR

$29.60

If you're looking for a fast action driving game then we recommend this one. Good graphics and sound.

FROGGER



This is the genuine FROGGER game that you see in the arcades. Made by the same people that made Jawbreaker (One of the top ten sellers.)

by ON-LINE

32K Disk 16K Tape $31.40

GORF

★ASTRO BATTLES
★LAZER ATTACK
★SPACE WARP
★FLAG SHIP

by ROKLAN MIDWAY

Just like the arcade game that has been so popular. This is one of our BEST sellers for all ages.

$35.10 24K Disk $39.90 16K Cart.

HARDWARE

Prices listed are the CASH discount price, charge prices are slightly higher.

800 Computer 16K	$575
800 Computer 48K	$678
400 Computer (used)	$199
810 Disk Drive	$439
850 Interface	$178
410 Recorder	$ 78
16K Ram module	$ 69
32K Ram module	$ 89
Percom Disk Drive	$648
48K Ram module for 400	$178
WICO JOYSTICK	$ 26
Trac Ball controller	$ 63
800/400 Dust cover	$ 9
10 Blank Disks	$ 28
Mark II Modern	$ 99
Alien Group Synthesizer	$153
Epson MX-80FT/Plus	$588
Amdek Color 1 monitor	$387
12' Joystick Extension Cord	$ 10
Computer Paper 1000 8½ x11	$ 17

Trade in your 400 for a NEW 800 Computer. Call for details.

SOFTWARE

ZAXXON

Now you can get that powerful 3-D game for the Atari. Made by one of the BEST game makers, Datasoft. Order early to get yours. D $35.10

WIZARD & THE PRINCESS

A terrific HI-RES graphics adventure that is one of the best for the Atari 40K Disk $29.90

EASTERN FRONT

The MOST popular of all the APX programs and truly one of the best war games for the Atari. D,T $27.90

SAMMY THE SEA SERPENT

A must for the little computerists ages 4-7. The kiddies LOVE this one and "Sammy" too. Tape $15.90

WAR

A colorful war simulation for either one or two players. If you enjoy wargaming, then this great game is certain to please. 32K Disk $22.50

THRESHOLD

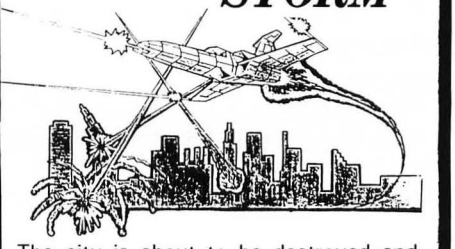
Superb graphics makes this one of the BEST space games for the Atari. Fast paced and arcade quality make this a must. 40K Disk $35.10

DE RE ATARI

This manual is a favorite among the serious programmers. Learn the many secrets of the Atari. $17.90

FREE
ATARI PRODUCTS CATALOG
(includes program descriptions)
with any order or
send $1.00 refundable with order
Send your name to get our
FREE monthly flyers

METEOR STORM



The city is about to be destroyed and ONLY you can save its people from destruction. FAST, arcade type action game.

24K Disk
16K Tape $29.95

O'RILEY'S MINE

16K Disk or Tape $31.50

by DATASOFT

A SUPER graphic game in which you are the miner and must watch out for mine flooding and monsters of the deep.

CROSSFIRE

The aliens have landed and it's your job to save the city. (If you can!) TOP RATE GAME, ARCADE QUALITY, HI-RES Graphics & Sound.

**16K TAPE
32K DISK
$26.90**

VISA master charge

TO ORDER CALL (503) 683-5361

HOW TO ORDER: Send check or money order or phone your order using your charge card. Ask about our 48 hour delivery service. Shipping on programs is $2.00 per order in USA or $3.90 for the 48 hour AIR service. Call for Hardware shipping costs. Prices subject to price change without notice.

Store Hours
8 am – 6 pm
Mon. – Sat.

Royal Software
2160 West 11th Ave., Eugene, Oregon 97402

GENERAL INFORMATION

Concerning SoftSide line listings, SWAT & Magnetic Media

Follow these procedures unless otherwise instructed by the documentation in the magazine. Back issues may differ in some details.

SWAT TABLES

At the conclusion of each line listing of a *SoftSide* program, we include a *SWAT* (*Strategic Weapon Against Typos*) Table. *SWAT* was published in issue #30 of *SoftSide* and is available as a free reprint. Please send a self-addressed, stamped envelope to *SoftSide* Publications, Inc., Dept. *SWAT*, 6 South Street, Milford, NH 03055.

APPLE™

Disks are in 16-sector format, created under DOS 3.3. To use, just boot the disk. A cover/menu program will run automatically.

Tapes LOAD in the normal manner. Advance the tape to the beginning of the lead-in tone; stop the tape; insert the plug into the EAR jack; type LOAD; start the tape; and press RETURN. Side two of the tape is a duplicate of side one, unless one or more Integer BASIC programs are included, in which case side two contains the Integer programs.

ATARI®

Line Listings use the following conventions in representing unprintable characters, unless otherwise noted:

Characters (including blank spaces) which are underlined should be typed in inverse video.

When graphics or control characters are to be included in a string (between quotation marks), it will be noted in a nearby REMark. In such cases, graphics characters are represented by the corresponding lower-case letter, and control characters are represented by the corresponding unshifted key symbol. For example: The lower-case letter s represents a graphics cross, entered by holding down the CTRL key and then pressing the S key. The symbol = represents a control-down-arrow, entered by first pressing and releasing the ESC key, then holding down the CTRL key and pressing the = key. (See Appendix F, and the back cover, of the *ATARI® BASIC Reference Manual.)*

The one exception to the above practice is that a clear-screen character (ESC CTRL-↰) is represented in listings by a right-hand brace, which looks like this: }

A shifted = is represented in the listings by a vertical line with a small gap in it:

SWAT — Before appending *SWAT* to a program in memory, the program to be SWATed must first be LISTed to disk or cassette (using LIST "D:FILENAME" for disk or LIST "C:" for tape). Next, turn the computer off, then on again, to clear the system and ENTER the program back into memory (using ENTER "D:filename" for disk or ENTER "C:" for tape). Because of the unique method in which *ATARI® BASIC* stores variables in a program, the variable table must always be in the same order to produce accurate *SWAT* codes. LISTing and ENTERing the program is the only known way to rebuild the variable table in a specific order so that *SWAT* codes can match.

Disks do not contain DOS.SYS files, and are therefore not bootable by themselves. First boot a disk which contains any version of DOS, then insert the *SoftSide* disk and RUN "D:COVER" (*Adventure of the Month* — RUN "D:INTRO").

Tapes CLOAD in the normal manner. If you have difficulty, try this procedure:
(1) Type POKE 54018,54 and press RETURN.
(2) Turn up the volume on your TV.
(3) Type CLOAD and press RETURN once.
(4) Press the PLAY button and listen.
(5) When you hear a steady lead-in tone, press RETURN again.
Side two of the tape is a duplicate of side one.

IBM® PC

DV is available on individual order. There is no CV at this time.

TRS-80®

Disks are available in Model I or Model III format. They contain the DOS PLUS operating system, and a cover program which automatically runs upon booting. Back issues prior to May, 1982, are available only in Model I format, and may be converted using the TRSDOS CONVERT utility on a two-drive Model III. Older back issues (with Model I TRSDOS) require you to enter BASIC and then type RUN "COVER".

Tapes CLOAD in the normal manner on Model I's, and at low speed (500 baud) on Model III's. The first program is a cover/menu program. Side two of the tape is a duplicate of side one.

NOTES ABOUT MAGNETIC MEDIA

SoftSide disks and tapes are duplicated by reliable, professional duplication services; bad copies are very rare. However, the trip through the mail occasionally wreaks havoc with sensitive magnetic media. If, after a reasonable number of tries and a careful check and cleaning of your equipment, you are not able to load a program from a tape or disk, please return it to us with an exact description of the problem. If we cannot duplicate the problem on our systems, we will advise you when we send the replacement copy.

We use no copy-protection on our media. We urge you to make a backup copy of every disk or tape as soon as you receive it (and at the same time resist the urge to give copies to friends). Our replacement policy does not extend beyond 30 days.

PC/SIDE page 51

TRS-80®/SIDE page 70

ATARI®/SIDE page 83

APPLE™/SIDE page 107

Kelly's Computing

New Toll Free Number
800-572-9215

$2.00 OFF COUPON

ATARI IBM

ATARI

☐ 400 COMPUTER	399.00	Now	299.00
☐ 800 COMPUTER	899.00	Now	660.00
☐ 810 DISK DRIVE	599.00	Now	440.00
☐ 410 CASSETTE RECORDER	99.00	Now	87.50
☐ 850 INTERFACE MOD	219.00	Now	180.00
☐ PAC MAN (CART)	44.95	Now	38.50
☐ CENTIPEDE (CART)	44.95	Now	38.50
☐ CAVERNS OF MARS (DISK)	39.95	Now	33.50
☐ STAR RAIDERS (CART)	49.95	Now	33.50
☐ SPACE INVADERS (CART)	44.95	Now	27.50
☐ ASTEROIDS (CART)	44.95	Now	27.50
☐ MISSILE COMMAND (CART)	44.95	Now	27.50
☐ JOYSTICKS (PAIR) (ACCS)	21.95	Now	19.50
☐ PADDLES (PAIR) (ACCS)	21.95	Now	19.50
☐ GOLD EDITION 1-12 (DISK & CASS) (WHILE THEY LAST)	100.00	Now	87.50
☐ ADVENTURES 1-12 (CASS) 24.95 ea.	24.95	Now	19.50
☐ LUNAR LANDER (CASS)	14.95	Now	13.50
☐ STAR FLITE (DISK)	19.95	Now	16.50
☐ REAR GUARD (DISK)	19.95	Now	16.50
☐ GALACTIC EMPIRE (CASS)	19.95	Now	16.50
☐ GALACTIC TRADER (CASS)	19.95	Now	16.50
☐ PREPPIE (DISK)	29.95	Now	25.50
☐ PREPPIE (CASS)	29.95	Now	25.50
☐ TEMPLE OF APSHAI (DISK & CASS)	39.95	Now	33.50
☐ STAR WARRIOR (DISK & CASS)	39.95	Now	33.50
☐ RESCUE AT RIGEL (DISK & CASS)	29.95	Now	25.50
☐ DATESTONE OF RYN (DISK & CASS)	19.95	Now	16.50
☐ CRUSH, CRUMBLE & CHOMP (DISK)	29.95	Now	25.50
☐ INVASION ORION (DISK & CASS)	24.95	Now	19.50
☐ UPPER REACHES OF APSHAI	19.95	Now	16.50
☐ CURSE OF RA	19.95	Now	16.50
☐ CRYPTS OF TERROR (DISK)	34.95	Now	29.50
☐ CRYPTS OF TERROR (CASS)	29.95	Now	25.50
☐ EMPIRE OF THE OVERMIND (DISK)	35.00	Now	29.50
☐ TANKTICS (CASS)	24.00	Now	19.50
☐ CONTROLLER (DISK)	30.00	Now	25.50
☐ EMPIRE OF THE OVERMIND (DISK)	35.00	Now	29.50
☐ EMPIRE OF THE OVERMIND (CASS)	29.50	Now	25.50
☐ SHAMUS (DISK & CASS)	34.95	Now	29.50
☐ MASTER TYPE (CASS)	39.95	Now	33.50
☐ BASKETBALL (CART)	34.95	Now	29.50
☐ SUPER BREAKOUT (CART)	44.95	Now	33.50
☐ SCRAM (CASS)	24.95	Now	19.50
☐ TOUCH TYPING (CASS)	24.95	Now	19.50
☐ MAILING LIST (CASS)	24.95	Now	19.50
☐ MACRO ASSEMBLER & TEXT EDITOR	89.95	Now	78.50
☐ ASSEMBLER	59.95	Now	50.50
☐ FILE MANAGER 800 (DISK)	99.95	Now	87.50
☐ DISK MANAGER (DISK)	29.95	Now	25.50
☐ MICRO PAINTER (DISK)	34.95	Now	29.50
☐ GRAPHIS MASTER (DISK)	39.95	Now	33.50
☐ SAMMY THE SEA SERPENT (DISK & CASS)	24.95	Now	19.50
☐ PAGE 6 (DISK)	34.95	Now	29.50
☐ DISK DETECTIVE (DISK)	29.95	Now	25.50
☐ PATH FINDER (DISK)	34.95	Now	29.50
☐ SPEED READ PLUS (DISK)	59.95	Now	50.50
☐ SPACE EGGS (DISK)	29.95	Now	25.50
☐ GALAXY (DISK)	25.00	Now	19.50
☐ B-1 NUCLEAR BOMBER (CASS)	16.00	Now	13.50
☐ MIDWAY CAMPAIGN (CASS)	16.00	Now	13.50
☐ NORTH ATLANTIC CONVOY RAIDER (CASS)	16.00	Now	13.50
☐ NUKEWAR (CASS)	16.00	Now	13.50
☐ CONFLICT 2500 (CASS)	16.00	Now	13.50
☐ LORDS OF KARMA (CASS)	20.00	Now	16.50
☐ APPLE PANIC (DISK)	29.95	Now	25.50
☐ RASTER BLASTER (DISK)	29.95	Now	25.50
☐ BUG ATTACK (DISK)	29.95	Now	25.50
☐ CANYON CLIMBER (DISK & CASS)	29.95	Now	25.50
☐ LE STICK (ACCS)	39.95	Now	33.50
☐ PACIFIC COAST HIGHWAY (DISK & CASS)	29.95	Now	25.50
☐ INTRUDER (DISK)	34.95	Now	29.50
☐ INTRUDER (CASS)	29.95	Now	25.50
☐ SHOOTING ARCADE (DISK & CASS)	29.95	Now	25.50
☐ TRACK ATTACK (DISK)	29.95	Now	25.50
☐ CLOWNS & BALLOONS (DISK & CASS)	29.95	Now	25.50
☐ MEGALEGS (DISK & CASS) (WHILE THEY LAST)	34.95	Now	29.50
☐ COMPU-READ (DISK)	29.95	Now	25.50
☐ COMPU-MATH/FRACTIONS (DISK)	39.95	Now	33.50
☐ COMPU-MATH/DECIMALS (DISK)	39.95	Now	33.50
☐ COMPU-MATH/FRACTIONS (CASS)	29.95	Now	25.50
☐ COMPU-MATH/DECIMALS (CASS)	29.95	Now	25.50
☐ MATCH RACERS (DISK)	29.95	Now	25.50
☐ PATHFINDER (AVAIL. FALL) (DISK)	34.95	Now	29.50
☐ DEADLINE (DISK)	49.95	Now	42.50
☐ ZORK I (DISK)	39.95	Now	33.50
☐ ZORK II (DISK)	39.95	Now	33.50
☐ POOL 400 (CART)	39.95	Now	33.50
☐ ACTION QUEST (DISK & CASS)	29.95	Now	25.50
☐ K-RAZY SHOOTOUT (CART)	49.95	Now	42.50
☐ SPEED READ PLUS (DISK)	59.95	Now	50.50
☐ HI-RES ADV. #0- MISSION: ASTEROID (DISK)	24.95	Now	19.50
☐ HI-RES ADV. #2- WIZ & PRINCESS (DISK)	32.95	Now	29.50
☐ CROSSFIRE (DISK)	29.95	Now	25.50
☐ JAWBREAKER (DISK & CASS)	29.95	Now	25.50
☐ THRESHOLD (DISK)	39.95	Now	33.50
☐ THE NEXT STEP (DISK)	39.95	Now	33.50
☐ ALI BABA AND THE FORTY THIEVES (DISK)	32.95	Now	29.50
☐ DELUXE INVADERS (DISK)	34.95	Now	29.50
☐ GALACTIC CHASE (DISK)	29.95	Now	25.50
☐ GALACTIC CHASE (CASS)	24.95	Now	19.50
☐ THE SHATTERED ALLIANCE (DISK)	39.95	Now	33.50
☐ PROTECTOR (DISK & CASS)	34.95	Now	29.50
☐ CHICKEN (DISK & CASS)	34.95	Now	29.50
☐ DODGE RACER (DISK & CASS)	24.95	Now	19.50
☐ WARLOCK'S REVENGE (DISK)	35.00	Now	29.50
☐ WORDRACE (DISK)	24.95	Now	19.50
☐ ALIEN SWARM (DISK)	34.95	Now	29.50
☐ ALIEN SWARM (CASS)	29.95	Now	25.50
☐ SNAKE BYTE (DISK)	29.95	Now	25.50
☐ CYCLOD (DISK)	29.95	Now	25.50
☐ JOYSTICK	29.95	Now	25.50
☐ JOYSTICK DELUXE	39.95	Now	34.50
☐ FAMOUS RED BALL	34.95	Now	29.95
☐ TRACKBALL	69.95	Now	63.50

IBM

☐ LOST COLONY (DISK)	29.95	Now	25.50
☐ TEMPLE OF APSHAI (DISK)	39.95	Now	33.50
☐ GALAXY (DISK)	25.00	Now	20.50
☐ MIDWAY CAMPAIGN (DISK)	21.00	Now	19.50
☐ COMPUTER STOCKS AND BONDS (DISK)	25.00	Now	20.50
☐ VOYAGER (DISK)	25.00	Now	20.50
☐ DRAW POKER (DISK)	21.00	Now	19.50
☐ CHAMPIONSHIP BLACKJACK (AVAIL JULY) (DISK)	39.95	Now	33.50
☐ THE HOME ACCOUNTANT PLUS (DISK)	150.00	Now	125.00
☐ WRITE-ON (DISK)	129.95	Now	110.00
☐ EASy (EXECUTIVE ACCOUNTING SYSTEM) (DISK)	725.00	Now	575.00
☐ DEADLINE (DISK)	49.95	Now	42.50
☐ ZORK I (DISK)	39.95	Now	33.50
☐ ZORK II (DISK)	39.95	Now	33.50
☐ T.I.M. III (DISK) ("Total Information Management")	495.00	Now	399.00
☐ MATHEMAGIC (DISK)	89.95	Now	75.00
☐ EASY SPELLER (DISK)	175.00	Now	149.00
☐ EASY FILER (DISK)	400.00	Now	335.00
☐ EASYWRITER II (DISK)	350.00	Now	280.00
☐ THE TAX MANAGER (DISK)	250.00	Now	199.00
☐ WORDSTAR (DISK)	495.00	Now	399.00
☐ MAILMERGE (DISK)	150.00	Now	99.00
☐ CONQUEST (AVAIL JULY) (DISK)	29.95	Now	25.50
☐ SUPERCALC (DISK)	295.00	Now	225.00
☐ SUPERWRITER (AVAIL JULY) (DISK)	395.00	Now	299.00
☐ IBM JOYSTICKS (ACCS)	64.95	Now	55.00
☐ GRAPHICS HARDCOPY SYSTEM (DISK)	24.95	Now	19.50
☐ VERSAWRITER GRAPHICS TABLET (ACCS)	299.00	Now	250.00
☐ DESKTOP PLAN I (DISK)	300.00	Now	250.00
☐ VISITREND/PLOT (AVAIL. JULY) (DISK)	300.00	Now	250.00

RAM FOR ATARI 400 + 800

48K (Reg. $299.00) *NOW* **$149.00**

32K (Reg. $199.00) *NOW* **$75.00**

COUPON
$2.00 OFF ORDER

To redeem, place a mail order for $75.00 or more and include this coupon. OFFER EXPIRES MARCH 31, 1983.

★ **TO ORDER CALL TOLL FREE 1-800-572-9215. CALIFORNIA RESIDENTS CALL 714-369-8113 OR WRITE TO 3515 BRUCE WAY, RIVERSIDE, CA 92506.** ★

WE PAY SHIPPING on software orders OVER $75 in the Continental U.S. (Foreign & Air extra). ADD $2.50 SHIPPING & HANDLING on orders under $75. **CALIF. RESIDENTS ADD 6% SALES TAX.** We accept **MASTER CARD** and **VISA. C.O.D.'s ADD $5.00.**

NAME _____

STREET _____

CITY _____ **STATE & ZIP** _____

CARD # _____ EXP. DATE _____

SIGNATURE _____

CHECK HERE FOR FREE CATALOG _____

CIRCLE 187 ON READER SERVICE CARD

by Alan J. Zett and Fred J. Condo

***PC Blues Box Part II* is a music editor program for an IBM® PC with 64K RAM, Advanced BASIC, color graphics adapter, and at least one disk drive.**

Editor's Note: The program listing that follows the instructions contains only new or modified lines, and requires Part I from issue 34. SWAT tables for both parts follow the listing, first for Part I, second for Parts I and II combined. To use this month's listing, simply load Part I, then type in the lines listed while Part 1 is still in memory. This will give you the finished program. Be sure to save the complete program!

Note Entry

The arrow keys on the numeric keypad and the adjacent RETURN key are the main input controllers. Try pressing one of the arrow keys. If you see the message "Invalid command ...," then press the Num Lock key, which is located just above the numeric keypad. The up and down arrows control the pitch of the note. The left and right arrows control the length of the note. Striking the left arrow key *increases* the length of the note or rest. Striking the space bar toggles between notes and rests. Whenever you strike a valid key, the tone currently displayed on the screen is played very briefly.

The pertinent information concerning what these keys control is displayed in various places on the screen. In the upper right hand corner is the information square. It tells you the tone (A through G) of the note, its position in the music memory buffer (next to the word "Note"), its tempo (32-255), its octave (1-5), its time length (1 means whole note or rest, 64 means 64th note or rest, etc.), and the number of blocks active. For now, you need only pay attention to Tone, Note, and Length.

The musical staff at the left side of the screen shows you the exact musical notation of the note you have selected. Between the staff and the information square is an octave-wide piano keyboard, with the key that corresponds to the current note, highlighted in blue. There is also a time indicator, located just below the staff. It shows you all the notes and rests, with the one currently in use highlighted in inverse video.

When the screen is showing you the note you desire, press the RETURN key (this is the large key, labelled with a bent arrow, to the left of the numeric keypad). This enters the note into the note buffer. The computer will play the note, and you will see the number next to the word "Note" increase by one. This means that you are now at the next empty location in the note buffer; the note you just entered by pressing RETURN resides in the immediately previous location.

Stepping through all six of the available octaves would be slow and tedious were it not for function keys one and two. In the large rectangular area below the piano keys and the information square is a summary of many of the commands for *PC Blues Box*. There, you can see that the F1 and F2 keys are, respectively, the Octave Up and Octave Down controllers. Striking these keys moves your note one octave in the indicated direction, leaving the note's other characteristics unchanged.

Block Functions

If your music has repetitive sections, or blocks, you will want to use the block functions. These are associated with function keys F3 through F8. When you use these functions, remember this basic principle: Block functions affect the note to which the number next to the word "Note" in the information square points. For example, if your repetitive block begins at the tenth note and ends at the fifteenth, then you must press F3 (Start Block) while the number next to "Note" is 10; by the same principle, you must press F4 (End Block) while the number next to "Note" is 15. You will remember that when you enter a note, the number next to "Note" increases by one, so be sure to press the appropriate block function key *before* you press the RETURN key to enter the note. Also, see the discussion of editing, below.

If you want to delete or unmark a block, you must be at the first note of the block you wish to change. The easiest way to do this is with the "Previous Block" and "Next Block" keys — F7 and F8. These keys always place you at the first note of a block. The block you currently occupy is shown below the musical staff, along with the numbers of the first and last notes in the block. Once you are at the first note of a block, you may press F6, the Block Delete key. The program will then ask whether you wish to unmark or delete the block. Unmarking a block simply removes its definition; deleting it removes it and all the notes it contains, *forever.* Press the U or D key, as appropriate. After you've answered the question, the computer will ask you, "Delete Repeats?" This is asking you if you wish to delete references to the block you are deleting or unmarking. If your music contains blocks that completely or partially overlap other blocks, you should be aware of the manner in which *deletion* of a block affects the blocks. If a block *marker* (starting or ending) points to a note that has been deleted because of its (the note's) location within an overlapping block you have deleted, then the block associated with that *marker* is effectively

unmarked. Furthermore, if you have responded affirmatively to the question regarding deletion of repeat references, the program removes all references both to the block you have explicitly deleted *and* to any overlapping blocks which have been automatically unmarked. For example, say a block extends from note one to note five, and another extends from note six to note ten. If yet another block, extending from note four to note seven, is deleted, then all the blocks must perforce be unmarked when notes four to seven are removed from the note buffer. It is very important to note that this applies only to block *deletion* — not to block *unmarking*.

You now need to know how to refer to a block. Instead of entering a note as described above, press F5, the Block Repeat key. When you do, the computer will say, "Repeat Block?" Here, you must type the number of a block, a number between 1 and 30, inclusive. If you type something outside of this range, your command will have no effect. Moreover, you must specify a block that *has already been defined* using the F3 and F4 keys; otherwise, the message "Invalid command ..." will flash on the screen. There is one very important thing to remember: repeats may not be nested. This means that a repeated block may not contain another repeat command. If this condition exists, a repeat error will occur during play, causing a message to appear, and the music to stop playing.

Editing

Once you've entered some notes into the *PC Blues Box*, you may wish to alter some of them. To do this, you must position yourself at the position in the note buffer of the note you wish to change. The simplest case of this would be the instance in which you have mistakenly entered a note. In order to change this note, simply press the F9 (Previous Note) key. This key moves you one note backwards in the note buffer. Above the staff will appear the warning "Editing..."; this message will remain there so long as you are not positioned at the first empty location after the last note in the note buffer. Once you are in the editing mode, you may use the arrow keys and RETURN just as you did before, only now, when you press RETURN, you will replace whatever note, rest, or repeat, previously in the note buffer, with whatever is displayed on the screen when you press RETURN. *PC Blues Box* does not require you to remember what was at the location you want to change; it displays its contents.

Use of the F9 and F10 keys is not a convenient method for single-stepping more than a few note positions. You may wish to move from note 40 all the way back to note 25, for instance. If you know the note's number, simply use the Edit Note key, N. The computer will ask you for the number of the note to which you wish to move. If you can't remember the number of the note you want to change, use the N key to get to the general vicinity of the note, then use the F9 and F10 keys to find its exact location.

The N key is invaluable when you want to go back into the note buffer and define blocks. Simply move to the first note in the block, press F3, then move to the last note in the block, and press F4.

The next editing command is Insert. You may insert only when the word "Editing" appears above the musical staff. To enter the Insert mode, press the I key. The word "Editing" will change to "Inserting." Now enter the notes you wish to insert. You should be *very* cautious while in the insert mode, as you will not be able to move back and forth in the note buffer. When you have entered all the notes you want to insert, press the I key again. You will leave the Insert mode and return to the Editing mode. The message "Adjusting Buffer ..." will appear briefly at the bottom of the large rectangular area on the screen. If you have inserted notes in front of any block start or block end markers, they will move forward, along with all the notes that your insertion may have displaced. If you make an erroneous entry while inserting, you may correct it with other editing functions once you have left the Inserting mode.

The converse of inserting notes is deleting spurious ones. Press the D key to delete notes. The computer will ask how many notes to delete. Respond with the number of notes you wish to remove from the note buffer, beginning with the note displayed on the screen when you hit D. If you decide not to delete any notes, type "0" in response to "How many?" or simply hit RETURN.

The last editing command is T, or tempo edit. When you press the T key, the computer will ask you for a tempo. Legal tempos range from 32 (slow) to 255 (fast). The computer will then ask, "Global (Y/N)?" Pressing Y tells the *PC Blues Box* that you wish to change the tempo of some or all of the music in the buffer. If you choose the global tempo change, you will see two more prompts — "Start?" and "End?" To these, you should respond with the numbers of the first and last notes whose tempo you wish to alter. Pressing RETURN in response to them enters the defaults of the first note for the Start and the last note for the End. If you reply negatively to the "Global?" prompt, then all that is changed is the tempo shown in the information square. The note buffer remains unchanged until you begin to enter notes with the RETURN key.

Saving and Loading

The first thing you'll probably want to do after you've entered some music is hear it. The first thing you *ought* to do is save it to disk. To do this, press the S key. The computer will ask you for a file name, which must, of course, be a standard PC DOS file name, and may include a drive specification. Thus "FUGUE.MUS" and "B:WATER" are both valid names, while "SOME NICETUNESIWROTE" and "3:MUSIC.THX" are not.

In order to load a music file from disk, use the L command. If the note buffer is empty, the music in the file will simply be loaded into the note buffer. If there is music in the note buffer, you will need to tell the computer whether or not to append the file to the music already in memory. If you reply to the "Append?" prompt by pressing the N key, then the music in memory *will be erased*, and the music file from the disk will replace it in the note buffer.

Playing Music

OK, now let's play some music. Press the P key. If you want to play the whole tune in memory, then press RETURN in response to the "Start?" and "End?" prompts. Otherwise, type the numbers of the first and last notes of the section you would like to hear. If you decide you don't want to hear the music once it starts, press the escape key. It is the one labelled "Esc" located at the upper left-hand corner of your keyboard, immediately to the right of the ten function keys. Because of the way the PC plays notes, you will hear as many as sixteen notes after you press "Esc".

Clear

Pressing the C key clears the note buffer. You must confirm this irrevocable command by pressing the Y key in response to the "Sure?" prompt.

There is but one more command for you to know about: the Quit command. To issue this command, you must press Control-C. This means you must hold down the "Ctrl" key located at the left side of the keyboard, and, while holding it down, press C. Again, you must confirm this irrevocable command. Before you confirm either this command or the Clear command, make sure your music has been saved on disk, as it will be irretrievable from memory once you have cleared or quit.

Future Enhancements

Look for the following *PC Blues Box* enhancements in a future issue of ***SoftSide***: nested repeats, a utility to transpose music from one key to another, and output of music as DATA statements so that you can incorporate *Blues Box* music in your own programs.

NOTE: The line by line documentation listed is for both parts 1 & 2 combined.

Variables

(Additions to last issue's variable list.)

AZ: Miscellaneous loop variable. Also used in live-key routines to specify the number of characters to INPUT.
AZ$: Contains INPUTted text. Used in live-key routines.
BS: Block Start flag. Set to TRUE when a block start marker is placed in the note buffer. Set to FALSE when a corresponding block end marker is added.
BLOCK(i,j): Block pointer array. Set to an array of 31 x 2 elements. Blocks are stored in two bytes, the start and end position of a block in the note buffer. Element 31 is reserved by the system for the Delete command. There are 30 usable blocks.
IB$(i): Insert buffer array. When Inserting, all notes are placed here and later transferred to the note buffer.
R: Miscellaneous loop variable. Also used in the Repeat option of the Play command.
ZA: Miscellaneous loop and storage variable.
ZA$: Miscellaneous string variable. Used in live-key routines.

```
SS SS SS SS SS SS SS SS SS SS SS
SS                             SS
SS   IBM PC Advanced BASIC     SS
SS   'PC BLUES BOX PART 2'     SS
SS   Author:  Alan J. Zett     SS
SS     Copyright (C) 1982      SS
SS SoftSide Publications, Inc  SS
SS                             SS
SS SS SS SS SS SS SS SS SS SS SS
```

Lines 10-140: Initialize and set up start defaults.

```
10 ON ERROR GOTO 30000:CLS:OPTION BASE 1
:KEY OFF:DEFINT A-Z:MAX=&H29A
50 DEF FNL2!(X)=LOG(X)/LOG(2):DEF FNS$(N
$)=RIGHT$(N$,LEN(N$)-1):TRUE=-1:FALSE=0:
FLAT=FALSE:SHARP=FALSE:OC=3:CN$="C":SCAL
E$="CDEFGAB":NN=7*OC-6
70 STAFFX=6:STAFFY=55:PTN$="":X$="C C#D
E-E F F#G A-A B-B ":FOR X=1 TO 5:PTN$=PT
N$+X$:NEXT:PSCALE$="C.D.EF.G.A.B":PTNPTR
=(12*(OC-1)+INSTR(PSCALE$,CN$)-SHARP+FLA
T)*2-1
90 KBX=123:KBY=6:NX=6:NY=160:RX=6:RY=180
:MIDC=30:NDX=STAFFX+49:RDX=NDX-2:RDY=STA
FFY:IBL=0:R=0:BS=0
100 TEMPO=100:DOTTED=FALSE:NOTE=TRUE:PRE
VNOTE=NOTE:TIME=4:PREVTIME=TIME:PREVDOT=
DOTTED:PREVPTR=PTNPTR:BCOUNT=0:BPOS=0:NC
OUNT=1:NPOS=1:INSERTING=FALSE
110 DIM NCURS(38),SHARP(38),FLAT(38),STA
FF(1884),NOTE1(38),NOTE2(38),NOTE4(38),N
OTE8(38),NOTE16(38),NOTE32(38),NOTE64(38
),BLOCK(31,2),M$(MAX),IB$(99),PCL(104),P
CC(104),PCR(104)
120 DIM REST1(38),REST2(38),REST4(38),RE
ST8(38),REST16(38),REST32(38),REST64(38)
,PIANOL(104),PIANOR(104),PIANOC(104),DOT
(38),NDY(35),BK(5):DEF FNU$(A$)=CHR$(ASC
(A$+CHR$(13))+32*(LEFT$(A$,1)>"Z"))
140 GOSUB 13000:CLS:X$="Just a moment...
":LOCATE 12,20-LEN(X$)/2:PRINT X$;:X=FRE(
A$):GOSUB 16170:GOSUB 2000:GOSUB 15000:G
OSUB 14000
```

Lines 1000-1170: Main input loop. Accept valid characters; reject invalid ones; branch to appropriate subroutines; sound current note.

```
1000 GOSUB 24000:IN$=AZ$
1010 IF IN$="S" AND INSERTING=0 THEN GOS
UB 18000:GOTO 1150
1020 IF IN$="L" AND INSERTING=0 THEN GOS
UB 17000:GOTO 1150
1025 IF IN$="I" THEN GOSUB 25000:GOSUB 2
7300:GOTO 1150
1030 IF IN$="P" AND INSERTING=0 THEN GOS
UB 19000:GOTO 1150
1035 IF IN$="D" AND INSERTING=0 THEN GOS
UB 26000:GOSUB 27300:GOTO 1150
1040 IF IN$="T" THEN GOSUB 20000:GOTO 11
50
1045 IF IN$=" " THEN GOSUB 11000:NOTE=NO
T NOTE:GOTO 1150
1050 IF IN$="N" AND INSERTING=0 THEN GOS
UB 21000:GOSUB 27300:GOTO 1150
1055 IF IN$="C" AND INSERTING=0 THEN GOS
UB 22000:GOTO 1150
1065 IF IN$=CHR$(3) AND INSERTING=0 THEN
 GOSUB 23000:GOTO 1150
1070 IF ASC(LEFT$(IN$,1))<>0 THEN GOSUB
50000:GOTO 1000 ELSE IN=ASC(RIGHT$(IN$,1
))
1080 IF IN<59 OR (IN>68 AND IN<72) OR IN
=73 OR IN=79 OR IN>80 THEN GOSUB 50000:G
OTO 1000
1090 IF IN=72 THEN ON NOTE+2 GOSUB 3000,
50000:GOTO 1150
1120 IF IN=80 THEN ON NOTE+2 GOSUB 4000,
50000:GOTO 1150
1130 IF IN>60 AND IN<>63 AND (INSERTING
OR NCOUNT=1) THEN GOSUB 50000:GOTO 1000
1140 ON IN-58 GOSUB 8000,7000,27000,2710
0,27200,27700,27500,27600,9000,10000
1160 GOSUB 12000:IF IN$<>"" THEN IF ASC(
LEFT$(IN$,1))=0 AND NOTE AND TEMPO>31 TH
EN PLAY "MB T255 L64 O=OC;"+N$
1170 GOTO 1000
```

Lines 9000-9060: Move to the previous note in the buffer.

```
9000 GOSUB 11000
9010 IF NPOS=1 THEN GOSUB 50000:RETURN
9020 NPOS=NPOS-1
9060 GOTO 27300
```

Lines 10000-10060: Move to the next note in the buffer.

```
10000 GOSUB 11000
10010 IF NPOS=NCOUNT THEN GOSUB 50000:RE
TURN
10020 NPOS=NPOS+1
10060 GOTO 27300
```

Lines 12000-12590: Display routine. Re-create the data of the previous note, using the information preserved in line 11000.

Lines 12010-12045: Adjust display for repeats, rests, editing and inserting prompts, and block positions.

```
12010 LOCATE 2,2:IF NPOS<NCOUNT THEN IF
INSERTING THEN PRINT "Inserting ..." ELS
E PRINT "Editing ...  " ELSE PRINT SPC(1
3)
12020 IF MID$(M$(NPOS),10,1)="P" OR NOT
NOTE THEN N2$="RST"
12030 IF NPOS=NCOUNT AND TEMPO<32 THEN T
EMPO=100
12040 FOR X=BCOUNT TO 1 STEP -1:IF NPOS=
>BLOCK(X,1) AND NPOS<=BLOCK(X,2) THEN LO
CATE 19,2:PRINT SPC(13):LOCATE 19,2:PRIN
T "#";FNS$(STR$(X));" [";FNS$(STR$(BLOCK
(X,1)));":";FNS$(STR$(BLOCK(X,2)));"]":X
=0
12045 NEXT X:IF X<>-1 THEN LOCATE 19,2:P
RINT SPC(13)
```

Lines 12050-12060: Update the information square.

```
12050 LOCATE 2,31:IF DOTTED OR LEFT$(N2$
,1)="R" THEN PRINT USING "Tone  \ \";N2$
+"." ELSE PRINT USING "Tone  \\";N2$
12060 LOCATE 5,31:PRINT USING "Octave  #
";OC:LOCATE 3,31:PRINT USING "Note  ###"
;NPOS:LOCATE 4,31:IF TEMPO>31 THEN PRINT
 USING "Tempo ###";TEMPO ELSE PRINT USIN
G "REPEAT ##";TEMPO
12070 LOCATE 7,31:PRINT USING "Blocks ##
";BCOUNT:LOCATE 6,31:PRINT USING "Length
 ##";TIME
```

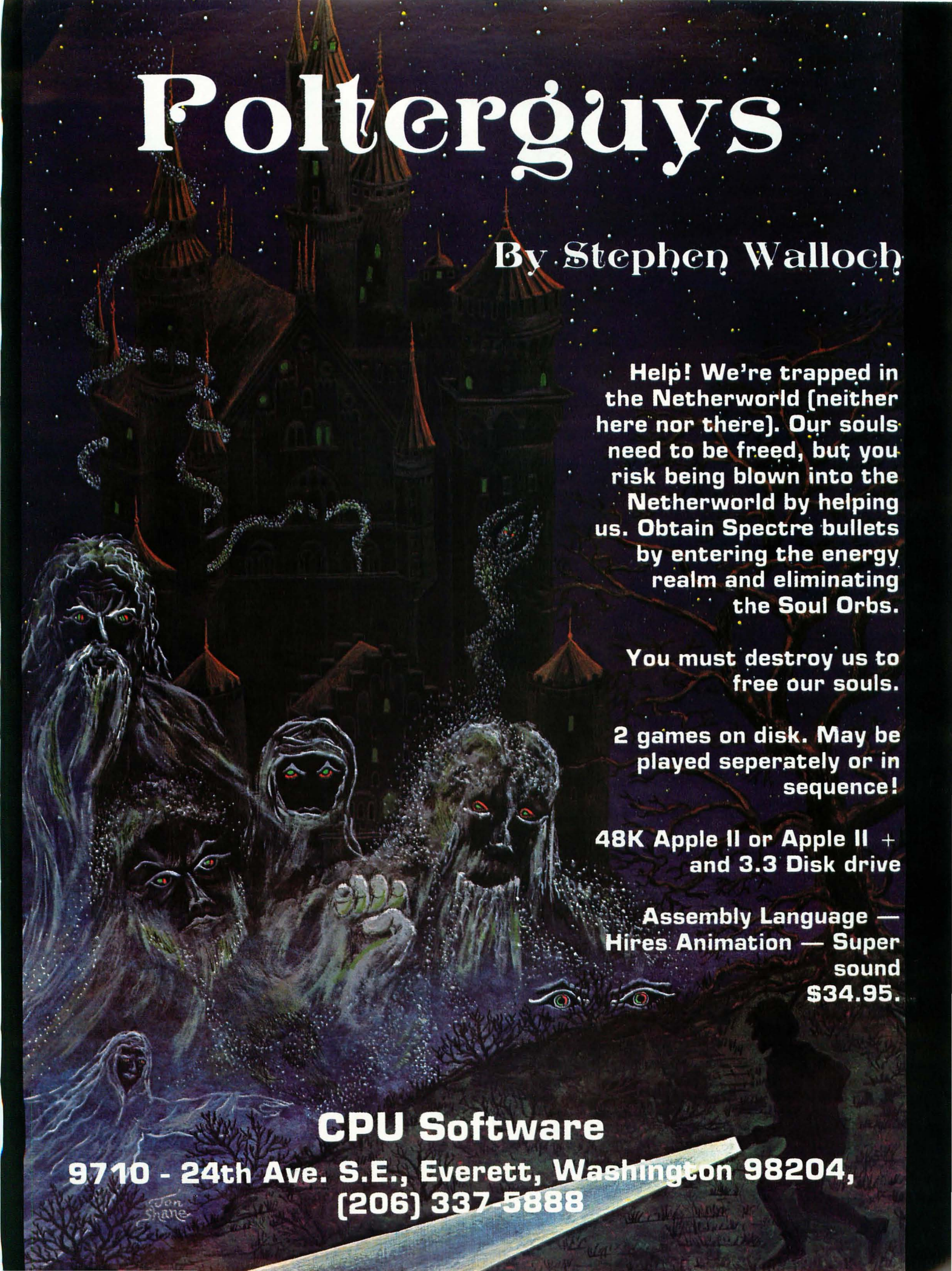
Lines 12160-12190: Update the piano keys.

```
12190 KLUGE=TRUE
```

Lines 13000-13140: Set up shape tables.

```
13000 SCREEN 1:CLS:COLOR 0,1:STAFF$="S10
 A000 BM000,000 C2 D8 R1 U8 L1 R1 BR40 D
8 R1 U8 L1 L40 D2 R40 D2 L40 D2 R40 D2 L
40":STAFF$=STAFF$+" L1 D16 R1 U8 L1 R1 B
R40 D8 R1 U8 L1 L40 D2 R40 D2 L40 D2 R40
 D2 L40":DRAW STAFF$:STAFF$=""
13050 CIRCLE (4,13),3,3,,,.55:GET (0,0)-
(13,17),NOTE1:LINE (7,12)-(6,1),3,B:GET
(0,0)-(13,17),NOTE2:PAINT (4,13),3,3:GET
 (0,0)-(13,17),NOTE4:LINE (7,1)-(12,3),3
:GET (0,0)-(13,17),NOTE8:LINE (7,3)-(12,
5),3:GET (0,0)-(13,17),NOTE16
13060 LINE (7,5)-(12,7),3:GET (0,0)-(13,
17),NOTE32:LINE (7,7)-(12,9),3:GET (0,0)
-(13,17),NOTE64:CLS
13080 LINE(2,0)-(2,6),3:LINE (4,0)-(4,6)
,3:LINE (1,2)-(5,2),3:LINE (1,4)-(5,4),3
:GET (0,0)-(13,17),SHARP:CLS:LINE (2,0)-
(2,6),3:LINE (2,6)-(6,4),3:LINE (6,4)-(2
,2),3:GET (0,0)-(13,17),FLAT
13100 CLS:LINE (3,11)-(9,13),3,BF:GET (0
,0)-(13,17),REST1:CLS:LINE (3,9)-(9,7),3
,BF:GET (0,0)-(13,17),REST2:CLS:LINE (6,
3)-(8,5),3:LINE (8,5)-(6,8),3:LINE (6,8)
-(8,10),3:LINE (8,10)-(5,13),3:LINE (5,1
```

continued on page 57


Polterguys
By Stephen Walloch
Help! We're trapped in the Netherworld (neither here nor there). Our souls need to be freed, but you risk being blown into the Netherworld by helping us. Obtain Spectre bullets by entering the energy realm and eliminating the Soul Orbs.
You must destroy us to free our souls.
2 games on disk. May be played seperately or in sequence!
48K Apple II or Apple II + and 3.3 Disk drive
Assembly Language — Hires Animation — Super sound
$34.95.
CPU Software
9710 - 24th Ave. S.E., Everett, Washington 98204,
(206) 337-5888

SoftSide DV, the magazine of the future, is here!

If your computer could pick a magazine, wouldn't it prefer one in its own language? Now there's one available. **SoftSide DV** is an enhancement of the **SoftSide** you have in your hands.

SoftSide DV contains not only the complete programs listed in every month's issue of **SoftSide,** but additional programs of every conceivable type, as well — multiple and Machine Language programs, modified languages, ongoing modular programs and software so extensive, it would take an entire issue of **SoftSide** just to print the code. Only the documentation for these programs will appear in **SoftSide** Magazine, **NOT** the code.

Feel as though you're missing something? You are! But, you needn't miss out on another issue. **SoftSide DV** is now available for Apple™, ATARI® and the TRS-80® . The cost to you — $125 for 12 magazines and 12 disks, packed with some of the best software available, all delivered to your home in the next year. For orders outside the USA, please add $36. For your convenience, we offer an installment payment plan for VISA and MasterCard holders: You pay only $32.50 per month for four months (a total of $130, which includes a $5 billing charge). Please use the special DV-CV bind-in card in this issue to order.

Computerists are offered the rare opportunity of marching into a new frontier. Advance to the front of the parade by subscribing to **SoftSide DV**, the magazine of the future, available today!

```
3)-(6,15),3:GET (0,0)-(13,17),REST4
13110 PUT (1,0),REST4:GET (0,0)-(13,17),
REST4:CLS:LINE (6,14)-(9,3),3:LINE (7,14
)-(10,3),3:LINE (9,4)-(3,6),3:GET (0,0)-
(13,17),REST8:LINE (9,6)-(3,8),3:GET (0,
0)-(13,17),REST16:LINE (7,8)-(3,10),3:GE
T (0,0)-(13,17),REST32
13130 LINE (7,10)-(3,12),3:GET (0,0)-(13
,17),REST64:CLS:LINE (0,0)-(9,30),3,BF:L
INE (0,30)-(12,50),3,BF:GET (0,0)-(12,50
),PIANOL
13140 CLS:LINE (0,0)-(13,17),3,BF:GET (0
,0)-(13,17),NCURS:CLS:LINE (10,14)-(11,1
5),3,B:GET (0,0)-(13,17),DOT:RETURN
```

Lines 14000-14265: Draw the initial screen. Erase obsolete arrays and play intro music.

```
14000 CLS:LINE (0,0)-(319,199),3,B:LINE
(1,1)-(318,198),3,B:LINE (114,0)-(115,19
9),3,B:LINE (115,62)-(319,63),3,B:LINE (
234,0)-(235,63),3,B
14070 LINE (0,NY-1)-(115,NY-2),3,B:PUT (
NX,NY),NOTE1:PUT (NX+15,NY),NOTE2:PUT (N
X+30,NY),NOTE4:PUT (NX+45,NY),NOTE8:PUT
(NX+60,NY),NOTE16:PUT (NX+75,NY),NOTE32:
PUT (NX+90,NY),NOTE64:TX=RX+15*FNL2!(TIM
E):PUT (TX,NY),NCURS
14090 LINE (0,RY-1)-(115,RY-2),3,B:PUT (
RX,RY),REST1:PUT (RX+15,RY),REST2:PUT (R
X+30,RY),REST4:PUT (RX+45,RY),REST8:PUT
(RX+60,RY),REST16:PUT (RX+75,RY),REST32:
PUT (RX+90,RY),REST64:PUT (STAFFX,STAFFY
),STAFF
14120 FOR X=1 TO 5:LINE (STAFFX+46,STAFF
Y-X*5)-(STAFFX+60,STAFFY-X*5),2:LINE (ST
AFFX+46,STAFFY+X*5+45)-(STAFFX+60,STAFFY
+X*5+45),2:NEXT X:LINE (STAFFX+46,STAFFY
+MIDC)-(STAFFX+60,STAFFY+MIDC),2:PUT (ND
X,NDY(NN)),NOTE4
14160 GOSUB 12000:LOCATE 13,21:DEF SEG:P
OKE 78,1:PRINT"PC Blues Box 2":LOCATE 15
,21:POKE 78,2:PRINT "A Music Editor":LOC
ATE 16,21:PRINT "For The IBM PC"
14162 POKE 78,3:LOCATE 18,20:PRINT "by F
red J. Condo":LOCATE 19,20:PRINT "and Al
an J. Zett":ERASE STAFF,PIANOL,PIANOC,PI
ANOR
14165 PLAY "MBT10502L16C#FA-B-BB-A-FC#FA
-B-BB-A-FF#B-03C#E-EE-C#02B-C#FA-B-BB-A-
FA-01A-02A-GF#01F#02F#EC#.03P8P32.C#":FO
R X=19 TO 13 STEP -1:LOCATE X,16:PRINT S
PC(24):NEXT
14200 LOCATE 12,16:PRINT "F5 REPT.  Bloc
k   DEL F6"
14230 LOCATE 15,16:PRINT "I  INSRT  Note
    DEL  D"
14240 LOCATE 16,16:PRINT "N  NOTE  Edit
  TEMPO  T"
14250 LOCATE 17,16:PRINT "S  SAVE  File
   LOAD  L"
14260 LOCATE 18,16:PRINT "C  CLEAR  Musi
c  PLAY  P"
14265 LOCATE 19,16:PRINT "[Esc] Stop / Q
uit Ctrl-C"
```

Lines 16000-16160: Accept a note into the buffer.

```
16000 GOSUB 11000:IF NCOUNT=MAX THEN GOS
UB 50000:RETURN ELSE M$="T000O0LOONa ":L
Z=-((TEMPO<10)+(TEMPO<100)):MID$(M$,2+LZ
)=FNS$(STR$(TEMPO)):MID$(M$,6)=FNS$(STR$
(OC)):LZ=-(TIME<10):MID$(M$,8+LZ)=FNS$(S
TR$(TIME))
16070 IF NOTE AND TEMPO>31 THEN PLAY M$
16080 IF INSERTING THEN IBL=IBL+1:IB$(IB
L)=M$:RETURN ELSE M$(NPOS)=M$
16090 NPOS=NPOS+1
16110 IF NPOS>NCOUNT THEN NCOUNT=NCOUNT+
1
16120 IF NPOS=NCOUNT THEN IF TEMPO<32 TH
EN TEMPO=100:RETURN ELSE RETURN
```

```
16160 GOTO 27310
```

Lines 17000-17120: Load and append note buffer routines.

```
17000 GOSUB 11000:LOCATE 22,16:PRINT "Fi
le? ";:AZ=14:GOSUB 24010:F$=AZ$:LOCATE 2
2,16:PRINT SPC(24):IF F$<"A" THEN RETURN
 ELSE OPEN F$ FOR INPUT AS #1:INPUT #1,X
,X:IF X=0 THEN 17120 ELSE CLOSE #1:OPEN
F$ FOR INPUT AS #1
17010 IF NCOUNT>1 THEN LOCATE 22,16:PRIN
T "Append? (Y/N)";:GOSUB 24000:X$=AZ$:IF
 X$="Y" THEN 17050
17020 INPUT #1,BCOUNT:INPUT #1,NCOUNT:NP
OS=NCOUNT:IF BCOUNT<>0 THEN FOR X=1 TO B
COUNT:INPUT #1,BLOCK(X,1):INPUT #1,BLOCK
(X,2):NEXT X
17030 BPOS=BCOUNT:FOR X=1 TO NCOUNT-1:IN
PUT #1,M$(X):NEXT X:GOTO 17100
17050 INPUT #1,AZ:INPUT #1,ZA:NPOS=NCOUN
T+ZA-1:IF BCOUNT+AZ>29 OR NCOUNT+ZA>MAX-
1 THEN GOSUB 50000:GOTO 17110 ELSE IF AZ
<>0 THEN FOR X=1 TO AZ:INPUT #1,BLOCK(BC
OUNT+X,1):INPUT #1,BLOCK(BCOUNT+X,2):NEX
T X
17060 FOR X=1 TO AZ:BLOCK(BCOUNT+X,1)=BL
OCK(BCOUNT+X,1)+NCOUNT-1:BLOCK(BCOUNT+X,
2)=BLOCK(BCOUNT+X,2)+NCOUNT-1:NEXT X:BPO
S=BCOUNT+AZ:FOR X=0 TO ZA-2:INPUT #1,M$(
NCOUNT+X):NEXT X
17070 FOR X=NCOUNT TO NPOS-1:R=VAL(MID$(
M$(X),2,3)):IF R<32 THEN R=R+BCOUNT
17080 MID$(M$(X),2,3)=RIGHT$("00"+FNS$(S
TR$(R)),3):NEXT X
17100 LOCATE 22,16:PRINT "Adjusting buff
er ...":X=FRE(A$):BCOUNT=BPOS:NCOUNT=NPO
S:TIME=4:DOTTED=FALSE:NOTE=TRUE:PTNPTR=4
9
17110 LOCATE 22,16:PRINT SPC(24):CLOSE #
1:RETURN
17120 LOCATE 22,16:BEEP:PRINT "NOT A MUS
IC FILE":FOR X=1 TO 1000:NEXT X:GOTO 171
10
```

Lines 18000-18010: Save note buffer routine.

```
18000 IF NCOUNT<2 THEN GOSUB 50000:RETUR
N ELSE GOSUB 11000:LOCATE 22,16:PRINT "F
ile? ";:AZ=14:GOSUB 24010:F$=AZ$:LOCATE
22,16:PRINT SPC(24):IF F$<"A" THEN RETUR
N ELSE OPEN F$ FOR OUTPUT AS #1
18010 PRINT #1,BCOUNT:PRINT #1,NCOUNT:IF
 BCOUNT<>0 THEN FOR X=1 TO BCOUNT:PRINT
#1,BLOCK(X,1):PRINT #1,BLOCK(X,2):NEXT X
```

Lines 19000-19100: Play music and check for valid repeats.

```
19000 GOSUB 11000:R=0:IF NCOUNT=1 THEN G
OSUB 50000:RETURN ELSE PLAY "MB":LOCATE
22,16:PRINT "Start? ";:AZ=4:GOSUB 24010:
X$=AZ$:IF VAL(X$)<1 OR VAL(X$)>NCOUNT-1
THEN X$="1"
19010 LOCATE 22,16:PRINT SPC(24):LOCATE
22,16:PRINT "End? ";:AZ=3:GOSUB 24010:IF
 VAL(AZ$)<VAL(X$) OR VAL(AZ$)>NCOUNT-1 T
HEN AZ$=STR$(NCOUNT-1)
19020 LOCATE 22,16:PRINT "Now playing ..
.":FOR X=VAL(X$) TO VAL(AZ$):IF INKEY$=C
HR$(27) THEN 25100
19030 IF VAL(MID$(M$(X),2,3))<32 THEN 19
050 ELSE PLAY M$(X)
19040 NEXT:GOTO 25100
19050 R=VAL(MID$(M$(X),2,3)):IF R<1 OR R
>BCOUNT THEN 19100
19060 FOR AZ=BLOCK(R,1) TO BLOCK(R,2):IF
 INKEY$=CHR$(27) THEN 25100
19070 IF VAL(MID$(M$(AZ),2,3))<32 THEN 1
9100 ELSE PLAY M$(AZ)
19080 NEXT:GOTO 19040
19100 GOSUB 11000:LOCATE 22,16:PRINT "Re
peat Error !!":FOR X=1 TO 1000:NEXT X:GO
TO 25100
```

Lines 20000-20050: Set the tempo; local or global.

```
20000 GOSUB 11000:LOCATE 22,16:PRINT "Te
mpo (32-255)? ";:AZ=3:GOSUB 24010:X$=AZ$
:IF VAL(X$)<32 OR VAL(X$)>255 THEN 20050
 ELSE TEMPO=VAL(X$):LOCATE 22,16:PRINT S
PC(24):IF INSERTING THEN RETURN
20010 LOCATE 22,16:PRINT "Global? (Y/N)"
:GOSUB 24000:X$=AZ$:IF X$<>"Y" THEN 2005
0 ELSE LOCATE 22,16:PRINT SPC(24):LOCATE
 22,16:PRINT "Start? ";:AZ=4:GOSUB 24010
:X$=AZ$:IF VAL(X$)<1 OR VAL(X$)>NCOUNT-1
 THEN X$="1"
20020 LOCATE 22,16:PRINT SPC(24):LOCATE
22,16:PRINT "End? ";:AZ=4:GOSUB 24010:IF
 VAL(AZ$)<VAL(X$) OR VAL(AZ$)>NCOUNT-1 T
HEN AZ$=STR$(NCOUNT-1)
20030 LOCATE 22,16:PRINT SPC(24):LOCATE
22,16:PRINT "Adjusting Tempo ...":FOR X=
VAL(X$) TO VAL(AZ$):IF MID$(M$(X),2,3)>"
031" THEN MID$(M$(X),2)=RIGHT$("00"+FNS$
(STR$(TEMPO)),3)
20040 NEXT X
20050 LOCATE 22,16:PRINT SPC(24):RETURN
```

Lines 21000-21060: Move to a note to edit.

```
21000 GOSUB 11000:IF NCOUNT=1 THEN GOSUB
 50000:RETURN ELSE LOCATE 22,16:PRINT "E
dit Note? ";:AZ=3:GOSUB 24010:X$=AZ$:LOC
ATE 22,16:PRINT SPC(24):IF VAL(X$)<1 THE
N RETURN ELSE IF VAL(X$)>NCOUNT-1 THEN N
POS=NCOUNT:RETURN
21060 TEMPO=VAL(MID$(M$(NPOS),2,3)):SH7=
(RIGHT$(C7$,1)="#"):FL7=(RIGHT$(C7$,1)="
-"):IF NOTE THEN PTNPTR=(12*(OC7-1)+INST
R(PSCALE$,N7$)-SH7+FL7)*2-1:RETURN ELSE
PTNPTR=49:RETURN
```

Lines 22000-22030: Clear the note buffer and pointers.

```
22000 GOSUB 11000:LOCATE 22,16:PRINT "Su
re? (Y/N) ";:GOSUB 24000:X$=AZ$:LOCATE 2
2,16:PRINT SPC(24)
22010 IF X$=CHR$(13) THEN RETURN
22020 IF X$<>"Y" THEN RETURN
22030 BCOUNT=0:NCOUNT=1:NPOS=1:TIME=4:NO
TE=TRUE:DOTTED=FALSE:PTNPTR=49:TIME=4:LO
CATE 22,16:PRINT "Clearing ...":ERASE M$
:DIM M$(MAX):X=FRE(A$):LOCATE 22,16:PRIN
T SPC(24):RETURN
```

Lines 23000-23010: Quit.

```
23000 LOCATE 22,16:PRINT "Sure? (Y/N)";:
GOSUB 24000:X$=AZ$:LOCATE 22,16:PRINT ST
RING$(24,32):IF X$=CHR$(13) THEN RETURN
23010 IF X$<>"Y" THEN RETURN ELSE SCREEN
 0,0,0:WIDTH 40:COLOR 11,0:CLS:END
```

Line 24000: Livekey routine for one character. Returns a CHR$(13) for a null.

```
24000 AZ$=INKEY$:IF AZ$="" THEN 24000 EL
SE IF LEFT$(AZ$,1)=CHR$(0) THEN RETURN E
LSE AZ$=FNU$(AZ$):RETURN
```

Lines 24010-24070: Livekey routine for a text line. Traps invalid characters, provides key click, and allows INPUT of a specific amount of characters contained in AZ.

```
24010 AZ$="":ZA$="":PRINT CHR$(95+63*(ZA
>10));:IF AZ=0 THEN AZ=255
24020 ZA$=INKEY$:IF ZA$>CHR$(31) AND ZA$
<CHR$(123) AND AZ>0 THEN PRINT CHR$(29);
ZA$;CHR$(95+63*(ZA>10));:AZ$=AZ$+ZA$:AZ=
AZ-1
24030 IF ZA$<>"" AND ZA$<>CHR$(8) THEN S
OUND 1100,.1
24040 IF ZA$=CHR$(13) THEN PRINT CHR$(29
);" ";:RETURN
24050 IF ZA$=CHR$(8) THEN SOUND 1600,.1:
IF LEN(AZ$)>0 THEN PRINT CHR$(29);CHR$(2
9);CHR$(95+63*(ZA>10));" ";CHR$(29);:AZ=
AZ+1:IF LEN(AZ$)>1 THEN AZ$=LEFT$(AZ$,LE
N(AZ$)-1) ELSE PRINT CHR$(29);:GOTO 2401
0
```

continued on page 60

YOU'VE JUST FOUND

THE MISSING LINK!

Computer Shopper is *your link* to individuals who buy, sell and trade computer equipment and software among themselves nationwide. No other magazine fills this void in the marketplace chain.

Thousands of cost-conscious computer enthusiasts save by shopping in Computer Shopper every month through hundreds of classified ads. And new equipment advertisers offer some of the lowest prices in the nation.

Computer Shopper's unbiased articles make for some unique reading among magazines and there's a "help" column to answer difficult problems you may have with interfacing, etc.

For a *limited time* you can subscribe to Computer Shopper with a 6 month trial for $6 or 12 months for $10.

You risk nothing, because if you're not 100% satisfied after receiving the first issue, you may cancel and receive a full refund. No questions asked!

If you want faster service, call in your subscription and charge on your MasterCard or VISA account.

Our business office is open from 8 a.m. to 5 p.m. Monday through Friday. Call for charge card orders or send your check or money order TODAY while the offer is still good.

6 month trial, $6 or 12 months for $10

COMPUTER SHOPPER

P.O. Box **F-704** ● Titusville, FL 32780
305-269-3211

PC Blues Box *continued*

```
24060 ZA=ZA+4:PRINT CHR$(29);CHR$(95+63*
(ZA>10));:IF ZA>20 THEN ZA=0
24070 GOTO 24020
```

Lines 25000-25100: Insert notes. Toggle between insert mode and normal note input. Take all notes entered into the insert buffer and put them into M$ (the note buffer). Also adjust all affected block markers.

```
25000 GOSUB 11000:IF NPOS=NCOUNT THEN GO
SUB 50000:RETURN ELSE IF INSERTING=0 THE
N INSERTING=NPOS:IBL=0:RETURN ELSE IF IB
L=0 THEN INSERTING=0:RETURN
25010 LOCATE 22,16:PRINT "Adjusting buff
er ...":FOR X=NCOUNT-1 TO INSERTING STEP
 -1:M$(X+IBL)=M$(X):NEXT X:NCOUNT=NCOUNT
+IBL:NPOS=NPOS+IBL:FOR X=0 TO IBL-1:M$(X
+INSERTING)=IB$(X+1):NEXT X
25020 IF BCOUNT=0 THEN 25100 ELSE FOR X=
1 TO BCOUNT:IF BLOCK(X,1)=>INSERTING THE
N BLOCK(X,1)=BLOCK(X,1)+IBL
25030 IF BLOCK(X,2)=>INSERTING THEN BLOC
K(X,2)=BLOCK(X,2)+IBL
25040 NEXT X:X=FRE(A$)
25100 LOCATE 22,16:PRINT SPC(24):INSERTI
NG=0:RETURN
```

Lines 26000-26020: Delete notes. Remove notes from the note buffer and block markers that are invalid after the removal. To delete notes, set up a fake block marker and call the block delete routine.

```
26000 GOSUB 11000:IF NPOS=NCOUNT THEN GO
SUB 50000:RETURN ELSE LOCATE 22,16:PRINT
 "How many? ";:AZ=3:GOSUB 24010:X$=AZ$
26010 IF VAL(X$)=0 THEN 25100 ELSE IF NP
OS+VAL(X$)=>NCOUNT THEN NCOUNT=NPOS:IF N
POS=1 THEN 22030 ELSE 25100
26020 IF BCOUNT=0 THEN 25100 ELSE AZ=31:
BLOCK(AZ,1)=NPOS:BLOCK(AZ,2)=NPOS+VAL(X$
)-1:X$="D":GOTO 27715
```

Lines 27000-27040: Block start. If allowed, set up the start position of a block in the marker array. Prompt for optional adjusting of all note buffer repeat block references.

```
27000 GOSUB 11000:IF BS OR BCOUNT=30 THE
N GOSUB 50000:RETURN ELSE FOR X=1 TO BCO
UNT:IF NPOS=BLOCK(X,1) THEN GOSUB 50000:
RETURN ELSE NEXT X:BCOUNT=BCOUNT+1:BS=-1
:IF BCOUNT=1 THEN BLOCK(1,1)=NPOS:BLOCK(
1,2)=0:BPOS=1:RETURN
27010 LOCATE 22,16:PRINT "Adjust Repeats
? (Y/N)":GOSUB 24000:X$=AZ$:LOCATE 22,16
:PRINT "Adjusting buffer ...";SPC(4):BPO
S=0:FOR X=1 TO BCOUNT-1:IF NPOS<BLOCK(X,
1) THEN BPOS=X
27020 NEXT X:IF BPOS=0 THEN BLOCK(BCOUNT
,1)=NPOS:BLOCK(BCOUNT,2)=0:BPOS=BCOUNT:G
OTO 25100
27030 FOR X=BCOUNT-1 TO BPOS STEP-1:BLOC
K(X+1,1)=BLOCK(X,1):BLOCK(X+1,2)=BLOCK(X
,2):NEXT:BLOCK(BPOS,1)=NPOS:BLOCK(BPOS,2
)=0:IF X$="N" THEN 25100
27040 FOR X=1 TO NCOUNT-1:R=VAL(MID$(M$(
X),2,3)):IF R<32 AND R=>BPOS THEN R=R+1
27050 MID$(M$(X),2,3)=RIGHT$("00"+FNS$(S
TR$(R)),3):NEXT X:GOTO 25100
```

Line 27100: Block end. Set the ending block marker.

```
27100 GOSUB 11000:IF BCOUNT=0 THEN GOSUB
 50000:RETURN ELSE IF BS=0 OR BLOCK(BPOS
,1)=>NPOS THEN GOSUB 50000:RETURN ELSE B
LOCK(BPOS,2)=NPOS:BS=0:RETURN
```

Lines 27200-27230: Block repeat. Set up a marker in the note buffer to repeat a specified block. Note that repeats cannot be nested within a block.

```
27200 GOSUB 11000:IF BCOUNT=0 THEN GOSUB
 50000:RETURN
27210 LOCATE 22,16:PRINT "Repeat Block?
";:AZ=2:GOSUB 24010:X$=AZ$:LOCATE 22,16:
PRINT SPC(24):IF VAL(X$)<1 OR VAL(X$)>31
 THEN RETURN ELSE IF VAL(X$)>BCOUNT THEN
 GOSUB 50000:RETURN
27220 AZ$="T"+RIGHT$("00"+X$,3)+"03L04C
":IF INSERTING THEN IB$(NPOS)=AZ$:IBL=I
BL+1 ELSE M$(NPOS)=AZ$
27230 GOTO 16090
```

Lines 27300-27320: Restore next note when editing else set defaults.

```
27300 IF NPOS>=NCOUNT THEN NPOS=NCOUNT:P
TNPTR=49:TIME=4:DOTTED=FALSE:NOTE=TRUE:R
ETURN
27310 C7$=MID$(M$(NPOS),10,2):N7$=LEFT$(
C7$,1):OC7=VAL(MID$(M$(NPOS),6,1)):DOTTE
D=(RIGHT$(M$(NPOS),1)="."):TIME=VAL(MID$
(M$(NPOS),8,2)):NOTE=NOT (ASC(N7$)=80)
27320 TEMPO=VAL(MID$(M$(NPOS),2,3)):SH7=
(RIGHT$(C7$,1)="#"):FL7=(RIGHT$(C7$,1)="
-"):IF NOTE THEN PTNPTR=(12*(OC7-1)+INST
R(PSCALE$,N7$)-SH7+FL7)*2-1:RETURN ELSE
PTNPTR=49:RETURN
```

Lines 27500-27510: Previous block. Move to note position of last block start marker if valid.

```
27500 GOSUB 11000:IF BCOUNT=0 THEN GOSUB
 50000:RETURN ELSE AZ=0:FOR X=1 TO BCOUN
T:IF BLOCK(X,1)<NPOS THEN AZ=X
27510 NEXT X:IF AZ=0 THEN GOSUB 50000:RE
TURN ELSE NPOS=BLOCK(AZ,1):GOTO 27300
```

Lines 27600-27610: Next block. Move to note position of next block start marker if valid.

```
27600 GOSUB 11000:IF BCOUNT=0 THEN GOSUB
 50000:RETURN ELSE AZ=0:FOR X=BCOUNT TO
1 STEP -1:IF BLOCK(X,1)>NPOS THEN AZ=X
27610 NEXT X:IF AZ=0 THEN GOSUB 50000:RE
TURN ELSE NPOS=BLOCK(AZ,1):GOTO 27300
```

Lines 27700-27810: Block delete. Remove block markers and delete notes if applicable. Prompt for removal of repeat references in the note buffer.

```
27700 GOSUB 11000:IF BCOUNT=0 THEN GOSUB
 50000:RETURN ELSE AZ=0:FOR X=1 TO BCOUN
T:IF BLOCK(X,1)=NPOS THEN AZ=X
27710 NEXT X:IF AZ=0 THEN GOSUB 50000:RE
TURN ELSE LOCATE 22,16:PRINT "Unmark or
Delete? (U/D)":GOSUB 24000:X$=AZ$:LOCATE
 22,16:PRINT SPC(24):IF X$<>"U" AND X$<>
"D" THEN RETURN
27715 LOCATE 22,16:PRINT "Delete Repeats
? (Y/N)":GOSUB 24000
27720 LOCATE 22,16:PRINT "Adjusting buff
er ...";SPC(4):IF X$="D" THEN FOR X=1 TO
 NCOUNT-BLOCK(AZ,2)-1:M$(BLOCK(AZ,1)+X-1
)=M$(BLOCK(AZ,2)+X):NEXT X:R=BLOCK(AZ,2)
-BLOCK(AZ,1)+1:NCOUNT=NCOUNT-R
27730 FOR X=1 TO BCOUNT:IF ((BLOCK(X,1)=
>BLOCK(AZ,1) AND BLOCK(X,1)<=BLOCK(AZ,2)
) OR (BLOCK(X,2)=>BLOCK(AZ,1) AND BLOCK(
X,2)<=BLOCK(AZ,2))) AND X$="D" AND X<>AZ
 THEN BLOCK(X,1)=-1:BLOCK(X,2)=-1
27740 NEXT X:FOR X=1 TO BCOUNT:IF X$="D"
 AND BLOCK(X,1)>BLOCK(AZ,1) AND X<>AZ TH
EN BLOCK(X,1)=BLOCK(X,1)-R:BLOCK(X,2)=BL
OCK(X,2)-R
27745 NEXT X:BLOCK(AZ,1)=-1:BLOCK(AZ,2)=
-1:AZ=0
27750 AZ=AZ+1:IF AZ>BCOUNT THEN X=FRE(A$
):GOTO 25100 ELSE IF BLOCK(AZ,1)>0 THEN
27750
27760 FOR X=AZ TO BCOUNT:BLOCK(X,1)=BLOC
K(X+1,1):BLOCK(X,2)=BLOCK(X+1,2):NEXT X:
BCOUNT=BCOUNT-1:IF AZ$="N" THEN AZ=0:GOT
O 27750 ELSE FOR X=1 TO NCOUNT-1:R=VAL(M
ID$(M$(X),2,3))
27770 IF R=AZ THEN M$(X)="X" ELSE IF R<3
2 AND R>AZ THEN MID$(M$(X),2,3)=RIGHT$("
00"+FNS$(STR$(R-1)),3)
27780 NEXT X:FOR X=1 TO NCOUNT-1:IF M$(X
```

IBM®-PC

```
I<>"X" THEN 27810 ELSE FOR R=X TO NCOUNT
-1:M$(R)=M$(R+1):NEXT R:NCOUNT=NCOUNT-1:
FOR R=1 TO BCOUNT:IF AZ=>BLOCK(R,1) AND
X<=BLOCK(R,2) THEN BLOCK(R,2)=BLOCK(R,2)
-1
27790 NEXT R:FOR R=1 TO BCOUNT:IF BLOCK(
R,1)>=BLOCK(R,2) THEN BLOCK(R,1)=-1:BLOC
K(R,2)=-1
27800 NEXT R:X=X-1
27810 NEXT X:AZ=0:GOTO 27750
```

Lines 30000-30030: Error trap routines. (Idiot proofing).

```
30000 BEEP:X=ERR:LOCATE 22,16:IF X=53 TH
EN PRINT "FILE NOT FOUND":RESUME 30020 E
LSE IF X=57 OR X>64 THEN PRINT "DISK ERR
OR":RESUME 30020 ELSE IF X=61 THEN PRINT
 "DISK FULL":RESUME 30020
30010 SCREEN 0,0,0:WIDTH 40:CLS:LOCATE 1
0,1:IF X>30 THEN PRINT "UNKNOWN ERROR";:
RESUME 30030 ELSE PRINT "PROGRAM ERROR";
:RESUME 30030
30020 FOR X=1 TO 2000:NEXT X:LOCATE 22,1
6:PRINT SPC(24):GOTO 1000
30030 PRINT X;"IN";ERL;"... BUG?":PRINT:
PRINT "TO SAVE ANY CURRENT WORK, GOSUB 1
9000":PRINT:ON ERROR GOTO 0:END
```

Line 50000: Invalid message subroutine.

```
50000 GOSUB 11000:LOCATE 22,16:PRINT "In
valid command ...":FOR X=1 TO 1000:NEXT
X:LOCATE 22,16:PRINT SPC(24):RETURN
```

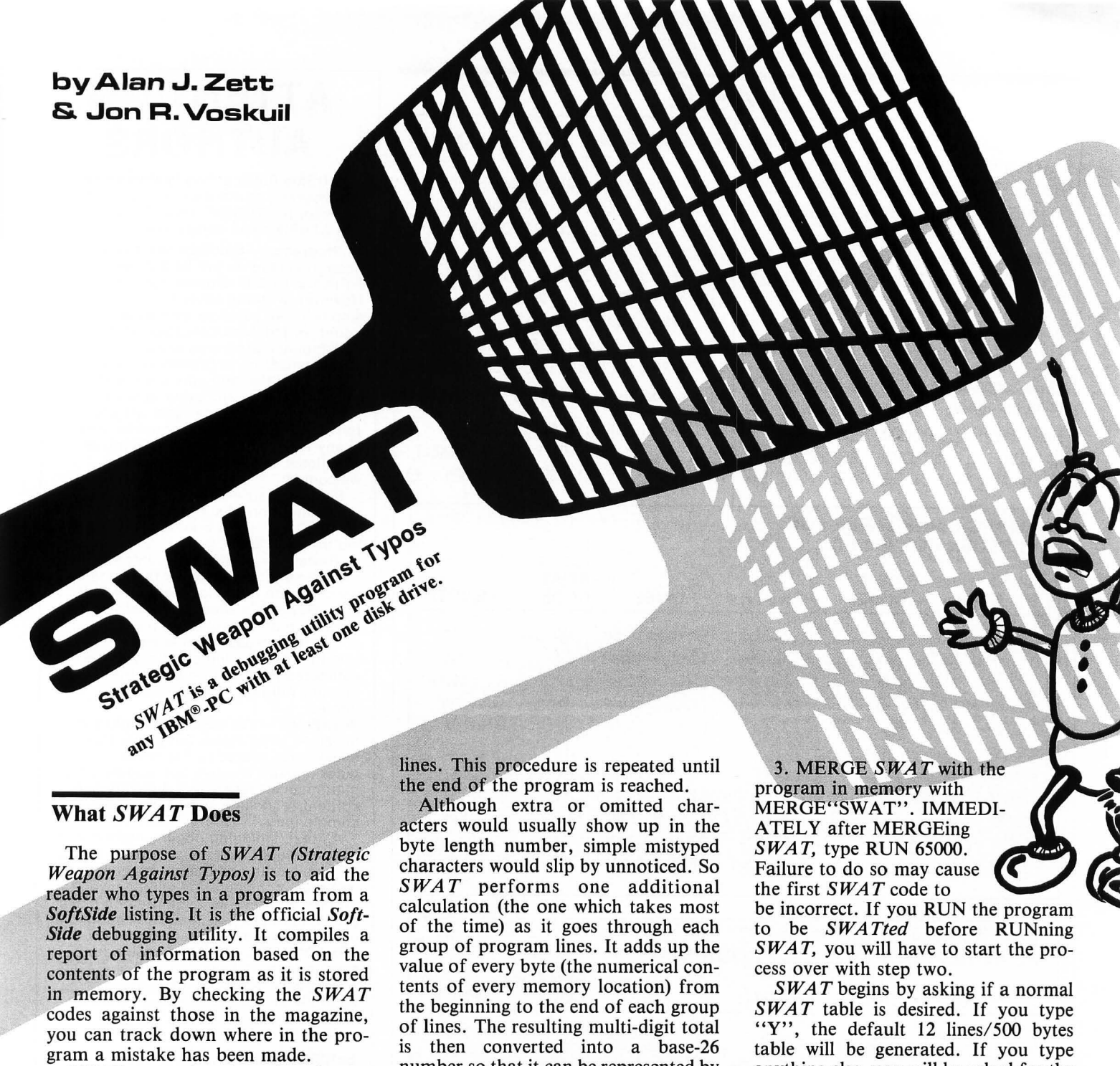
IBM® PC SWAT TABLE FOR: PC BLUES BOX (Part I)

LINES	SWAT CODE	LENGTH	LINES	SWAT CODE	LENGTH
10 - 90	UP	652	12570 - 13050	WW	530
100 - 1045	RX	531	13060 - 13110	EJ	638
1050 - 1150	ME	410	13130 - 14000	LV	541
1160 - 5030	VT	423	14030 - 14070	DK	570
5050 - 8000	LE	362	14090 - 14160	NM	505
8010 - 10020	NC	624	14162 - 14260	IR	536
10060 - 12060	RJ	574	14265 - 16070	HR	511
12100 - 12180	IA	610	16090 - 17000	WT	511
12190 - 12310	LN	494	17060 - 20000	KK	534
12320 - 12440	UT	490	20010 - 22030	SD	565
12450 - 12560	CL	355	23000 - 50000	QQ	187

IBM® PC SWAT TABLE FOR: PC BLUES BOX PARTS 1 & 2 COMBINED

LINES	SWAT CODE	LENGTH	LINES	SWAT CODE	LENGTH
10 - 100	DT	578	14250 - 16000	IV	504
110 - 1020	ME	542	16060 - 17010	DA	522
1025 - 1090	JG	521	17020 - 17070	AQ	540
1100 - 2060	SP	541	17080 - 18060	WE	561
3000 - 6010	NW	324	19000 - 19080	AQ	510
6020 - 9060	OY	280	19100 - 20030	WI	587
10000 - 12040	RN	644	20040 - 22000	SF	532
12045 - 12160	OT	710	22010 - 24020	WU	521
12180 - 12280	OM	519	24030 - 25010	HA	559
12290 - 12400	LI	500	25020 - 27000	NA	607
12410 - 12530	SJ	439	27010 - 27100	BK	561
12540 - 13050	YN	607	27200 - 27320	SM	611
13060 - 13110	RW	653	27500 - 27715	TB	504
13130 - 14000	LM	582	27720 - 27745	NK	501
14030 - 14070	HP	572	27750 - 27790	XR	547
14090 - 14160	DE	503	27800 - 50000	AX	445
14162 - 14240	EH	532			

by Alan J. Zett
& Jon R. Voskuil



SWAT

Strategic Weapon Against Typos

SWAT is a debugging utility program for any IBM®-PC with at least one disk drive.

What *SWAT* Does

The purpose of *SWAT (Strategic Weapon Against Typos)* is to aid the reader who types in a program from a ***SoftSide*** listing. It is the official ***SoftSide*** debugging utility. It compiles a report of information based on the contents of the program as it is stored in memory. By checking the *SWAT* codes against those in the magazine, you can track down where in the program a mistake has been made.

SWAT operation is quite simple. Starting at the first memory location where program lines are stored, it begins examining the computer's memory. It finds the line number of the first line of the program, and the pointers which give the location (address) of the beginning of the next line. Jumping to that address, it finds the address of the following line. Continuing in this manner, it scans twelve program lines or 500 bytes of memory, whichever comes first. This information is incorporated into the three-column *SWAT* Table: The first item in column one lists a range of line numbers, and the corresponding item in column three gives the number of bytes of memory occupied by those lines. This procedure is repeated until the end of the program is reached.

Although extra or omitted characters would usually show up in the byte length number, simple mistyped characters would slip by unnoticed. So *SWAT* performs one additional calculation (the one which takes most of the time) as it goes through each group of program lines. It adds up the value of every byte (the numerical contents of every memory location) from the beginning to the end of each group of lines. The resulting multi-digit total is then converted into a base-26 number so that it can be represented by letters of the alphabet, and the rightmost two "digits" (letters) of this number become the *SWAT* code in column two of the table.

How To Use *SWAT*

The first step, of course, is to type in the *SWAT* program and SAVE it in the ASCII format to disk (SAVE"file name",A). Then, to use *SWAT* on another program, LOAD the program you want to check and MERGE *SWAT*. For example:

1. SAVE the *SWAT* program with SAVE"SWAT",A.
2. LOAD the program to *SWAT* with LOAD"filename".
3. MERGE *SWAT* with the program in memory with MERGE"SWAT". IMMEDIATELY after MERGEing *SWAT,* type RUN 65000. Failure to do so may cause the first *SWAT* code to be incorrect. If you RUN the program to be *SWATted* before RUNning *SWAT,* you will have to start the process over with step two.

SWAT begins by asking if a normal *SWAT* table is desired. If you type "Y", the default 12 lines/500 bytes table will be generated. If you type anything else, you will be asked for the modified parameters. These will be listed with the *SWAT* table when they are needed. After this, you will be prompted to choose the mode of output. *SWAT* will produce a table on your video screen or printer. Once the *SWAT* table has been generated, simply compare it to the one published with the program in the magazine. If it is identical, you can pat yourself heartily on the back and start using your program.

What To Do If The *SWAT* Tables Don't Match

1. First, examine the listed line

numbers in the first column. If they don't match, it probably means that you have inserted, omitted, or changed one or more line numbers. (Although a large error in the length of a particular line could throw the item in column 1 off as well.) An extra or missing line will affect all column one entries from that point on. Search the lines indicated by the earliest item which is fouled up for the problem line number, and correct it. Then try running *SWAT* again to see if the entries match.

2. If, after verifying all the line numbers, there are still differences in column two or three, more detailed debugging is necessary. A wrong item in column three will almost always be accompanied by a wrong item in column two, although the reverse may not be true.

(a) If the length in column three is wrong, but the *SWAT* code in column two is OK, the most likely cause is a mistyped character somewhere in the indicated lines. A variable named N0, for example, may have been typed as NO; a comma and a period may have been exchanged; a number such as 32767 may have been mistyped as 32757; or even the words in a PRINT statement misspelled. Keep in mind that any BASIC keyword (PRINT, INPUT, FOR, etc., etc.) occupies only one byte of memory when stored in a program line, so typing GOSUB instead of GOTO would not change the byte count in column three, but would change the *SWAT* code in column two.

(b) If the length in column three is wrong, it's possible to deduce the reason for the error by comparing the size of the number to what it should be. A number which is too large usually indicates extra characters, while a smaller number usually indicates missing characters. (Because of the way BASIC stores keywords, this can be misleading in some cases.) The only way to find a typing error is to look for them, line by line and character by character. The advantage of using *SWAT* is that it narrows the number of lines to search to no more than twelve, or about 500-700 bytes of code. The "resolution" of *SWAT* can easily be changed to a larger or smaller number of lines or bytes. There will be occasions when we will publish a modified *SWAT* table. This will be for listings which are more difficult to type because of the nature of their content (i.e. a scrambled *Adventure* or a program with many DATA statements).

Errors That Might Not Be

SWAT is very picky. That is both an advantage and a disadvantage. If your typed-in program differs in *any* detail from that on our master disk, the tables won't match, even if those details are of no significance to the program when it is RUN. In particular, here are three types of picky details which you are likely to encounter.

(1) Differences in REM lines are treated by *SWAT* in exactly the same way as differences in other program lines. The majority of the programs we list contain few or no REMs; but if they do, they *must* be entered exactly as listed. If you want to add REMs for your own benefit, wait until after you've *SWATed* the program to do so.

(2) Differences in DATA lines may or may not be significant. On one hand, such lines are the cause of many programming problems and are especially difficult to debug. On the other, extra spaces immediately before or after commas make no difference to the computer in reading data, (unless they are enclosed in quotation marks) but they *do make a difference to SWAT*. If problems show up in lines which contain DATA statements, check the DATA items and their spacing.

(3) Spacing, spelling and punctuation within quotation marks are also important. In program instructions, the strings to be printed must be entered exactly as shown in the listing in order for *SWAT to work* correctly. Also, all spaces between key words, variables, etc., which are typed in, become a part of the program in memory. This means that *all* spacing is critical, not just that included within quotation marks and in data items.

Using *SWAT* To Check Itself

With two simple modifications you can use *SWAT* to verify itself once you've typed it into your computer. First, add one line to the end of the program:

```
65500 REM
```

Second, change the value of LN (in line 65000) to 65500. If you now run *SWAT* by itself (without another program in memory) it should generate the table shown following the program listing for *SWAT*.

Squish

At the end of the *SWAT* table for *SWAT*, we've included the *SWAT* table for *Squish*, published in issue 34 of ***SoftSide***. We hope this will aid our readers who have had any difficulty entering this program.

Variables

A: Memory address.
A1: Beginning memory address for a group of program lines.
AA: Beginning memory address for a single program line.
B: Byte limit for memory scan.
C$: Two-letter *SWAT* Code.
C: Color to print next *SWAT* code on screen.
D: Used in converting S to a base-26 number.
D1,D2: Values of first and second digits of *SWAT* Code.
I,J: Loop counters.
L: Line number.
L1: Lowest line number of a group of lines.
L2: Highest line number of a group of lines.
LN: Beginning line number of the *SWAT* program.
MP: Modified parameters flag.
N: Screen line counter for *SWAT* Table display.
NU: Line limit for memory scan.
S: Sum of the contents of a range of memory locations.
X$: User input string.

IBM®-PC

```
SS SS SS SS SS SS SS SS SS SS SS
SS                             SS
SS        IBM PC BASIC         SS
SS           'SWAT'            SS
SS    Author: Alan J. Zett     SS
SS     Copyright (c) 1982      SS
SS SoftSide Publications, Inc  SS
SS                             SS
SS SS SS SS SS SS SS SS SS SS SS
```

Initialize variables. Find starting address of the BASIC program and clear the screen.

```
65000 CLEAR:LN=65000!:NU=12:B=500
```

Prompt for normal or modified parameters.

```
65010 SCREEN 0:COLOR 15:CLS:DEF SEG:C=10
:MP=0:LOCATE 7,1:PRINT"NORMAL PARAMETERS
?":GOSUB 65120:IF X$()"Y" THEN PRINT:INP
UT"INPUT NUMBER OF LINES, NUMBER OF BYTE
S ";NU,B:MP=-1
```

If printout is desired, OPEN a file to the printer. Otherwise, OPEN a file to the screen.

```
65020 KEY OFF:A=PEEK(48)+PEEK(49)*256:N=
0:CLS:LOCATE 7,1:PRINT"DO YOU WANT A PRI
NTOUT? (Y/N) ";:GOSUB 65120:CLS
65030 IF X$="Y" THEN OPEN"LPT1:" FOR OUT
PUT AS #1:PRINT:LINE INPUT"PROGRAM TITLE
: ";T$:PRINT #1,CHR$(15);SPC(20);"IBM PC
 SWAT TABLE FOR: ";T$:PRINT #1," " ELSE
OPEN"SCRN:" FOR OUTPUT AS #1
65040 IF MP=-1 THEN PRINT #1,SPC(20);"MO
DIFIED PARAMETERS: NU =";NU;", B =";B:PR
INT #1," "
```

Set up the PRINT USING string and PRINT the SWAT table headings.

```
65050 A$="##### - #####"+SPACE$(7)+"\\ "
+SPACE$(6)+"####":PRINT #1,SPC(24);"LINE
S";SPC(7);"SWAT CODE   LENGTH":PRINT #1,
SPC(20);STRING$(13,"-");SPC(3);STRING$(9
,"-");SPC(3);STRING$(6,"-")
```

Look in memory for the next program line number. IF it is the first line of the SWAT program, THEN we are done.

```
65060 COLOR C:L1=PEEK(A+2)+PEEK(A+3)*256
:A1=A:S=0:L=L1:IF L1=LN THEN 65110
```

Begin a loop to search through NU program lines in memory. Store the beginning address of the present line, and look for the beginning address of the next line. Get the next program line number and calculate the current checksum of the program lines being searched. Check again for the begining of SWAT.

```
65070 FOR I=1 TO NU:AA=A:A=PEEK(A)+PEEK(
A+1)*256:L2=L:L=PEEK(A+2)+PEEK(A+3)*256:
FOR J=AA+2 TO A-1:S=S+PEEK(J):NEXT J:IF
L=LN OR A-A1>B THEN I=NU
```

Convert the checksum in S into a two letter code.

```
65080 NEXT I:D=INT(S/676):S=S-D*676:D1=1
NT(S/26):D2=S-D1*26:C$=CHR$(D1+65)+CHR$(
D2+65)
```

Print the three table items for this group of lines. If the screen display mode is being used, pause if the SWAT codes are about to scroll off the screen.

```
65090 PRINT #1,SPC(20);USING A$;L1;L2;C$
;A-A1:IF X$()"Y" THEN N=N+1:IF N=23 THEN
 N=0:LOCATE 24,26:COLOR 30:PRINT"PRESS R
ETURN TO CONTINUE";:GOSUB 65120:LOCATE 2
5,26:PRINT SPC(25);:LOCATE 24,1
```

Go back to search the next group of lines.

```
65100 C=C+1:C=C+2*(C>11):GOTO 65060
```

All done.

```
65110 PRINT #1," ":CLOSE:COLOR 11:KEY ON
:LOCATE 23,1:END
```

INKEY$ subroutine.

```
65120 X$=INKEY$:IF X$="" THEN 65120 ELSE
 RETURN
```

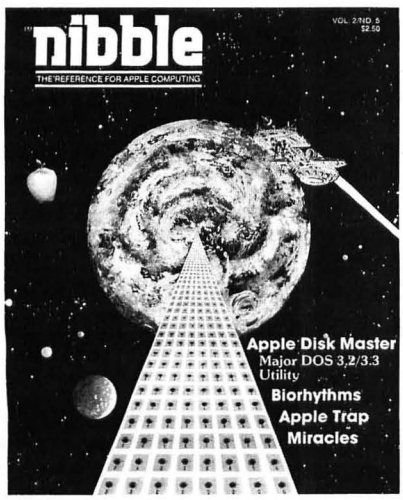
IBM® PC SWAT TABLE FOR: SWAT

LINES	SWAT CODE	LENGTH
65000 - 65050	AS	601
65060 - 65120	OE	479

IBM® PC SWAT TABLE FOR: SQUISH

LINES	SWAT CODE	LENGTH
10 - 30	HB	565
40 - 120	LS	581
130 - 240	WO	540
250 - 300	XG	516
310 - 410	TF	647
420 - 510	VR	610
520 - 560	OU	128

"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

We accept Master Charge & Visa

Box 325, Lincoln, MA. 01773 (617) 259-9710
I'll try nibble!
Enclosed is my $19.95 (for 8 issues)
(Outside U.S., see special note on this page.)
☐ **check** ☐ **money order**

Your subscription will begin with the next issue published after receipt of your check/money order.

Card # _____ Expires _____

Signature _____

Name _____

Address _____

City _____

State _____ Zip _____

NOTE:

— Domestic U.S. First Class subscription rate is $36.50
— Canada Air Mail subscription rate is $42.50
— Outside the U.S. and Canada Air mail subscription rate is $47.50

All payments must be in U.S. funds drawn on a U.S. bank.

Las Vegas Blackjack

Reviewed by Peter Brajer

by Charles J. Montante (Quala™, 1014 Griswold Avenue, San Fernando, CA 91340). System requirements: IBM® PC with at least one disk drive, 64K memory, PC DOS and BASIC A, 80 column display and either the monochrome or color graphics adapter. Suggested retail price is $39.95.

Having just returned from Atlantic City, it was eminently appropriate that I was asked to review this piece of software. Quala's *Las Vegas Blackjack* is a faithful reproduction of the card game, Blackjack, as played in the Las Vegas Strip Casinos. Perhaps, had I had a copy before my visit, I'd have won at the tables. With that in mind, I eagerly unwrapped the package and began to examine the 39 page manual. The manual has an attractive cover and is of the standard 5 1/2 x 8 1/2 IBM manual size (You'll have to punch holes yourself if you want to store it in one of your IBM binders.) I found it easily readable and well organized. A table of contents outlines what is to be presented and becomes a handy guide for information review as you learn the game.

In the introduction, you are told to expect a game which is "both fun and a real challenge." This is certainly true. Just like in a casino, you're asked to place bets and you make decisions which help determine your winning or losing. Your skill at making those choices, plus Lady Luck, determine the outcome. Here is where the secondary purpose of this program comes in. The manual teaches you how to turn the odds in your favor.

A description of a winning strategy, together with a well laid-out table of action-to-take descriptions, quickly help you decide what action to take. When you practice this strategy, your chances of beating the house increase significantly.

The manual also presents a simple card counting technique. You can practice it and confirm how well you are doing by checking with the system, which keeps count as the game runs. The method is based on the realization that composition of the deck changes as you play. As the composition varies, the odds change from favoring the house to favoring the player. Therefore, betting more when the odds favor the player and betting less when they favor the house turns the probability of making a profit in your favor.

Starting up is a two step process. The first time you start, you must copy the PCDOS and BASIC A software onto your *Las Vegas Blackjack* disk. Full, simple directions give you step by step instructions on how to do this. (There are typos in the original manual, but these have been corrected in the form of an errata sheet which comes in the package.) We had no problem making our program perform as described. From then on, startup is automatic and the game will autoboot.

An extensive play example is presented in the manual. This shows you each move, and what happens in a typical game. The graphics are described and the mechanics of betting are explored, including what happens to the representation of the pile of chips (the bankroll) you may gain or lose. The order of the game is presented, i.e. who is dealt first and what responses you are able to make. Messages are displayed in the center of the screen which instruct you to press the function key of your choice, depending on the action you wish to take. Alternatively, you may type in the letter of your choice, such as H for Hit, meaning take another card, or S for Stand, meaning no more cards. At the end of each round, the system scores the value of your cards and either collects or pays the appropriate chips.

Our only criticism of this game comes at this point. After each play, the system prompts you with "press any key to continue." This message is extremely difficult to read because of the colors used by the program. Blue does not contrast adequately on a green field; the message color should be changed for better contrast.

To sum it all up, *Las Vegas Blackjack* performs as expected and provides competent education for playing the game. The strategy appears to improve skill and I believe that serious practice would increase the chances of winning in the real world. Future versions should have cleaner color coordination for player message clarity, and include the possibility of multi-player formats.

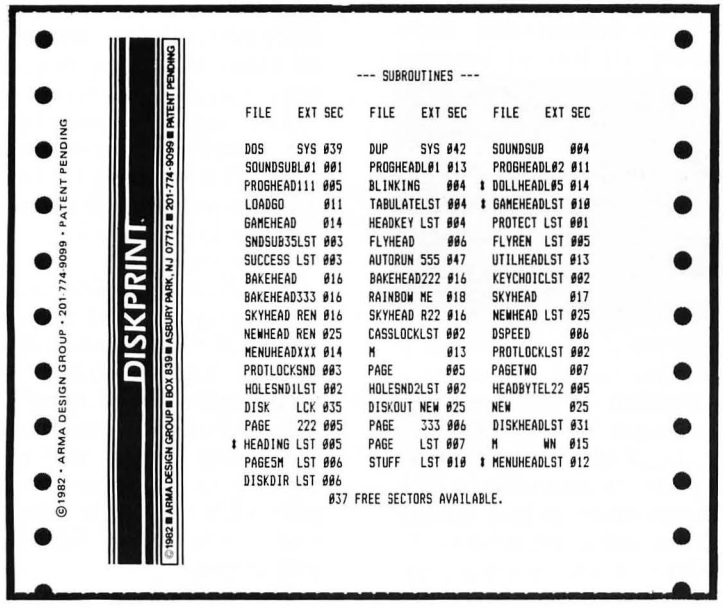
The Arma Design Group Proudly Announces

Diskprint®

If you're using a computer with one or more disk drives, have we got a product for you!

DISKPRINT is beyond a doubt the most useful new package to become available since the computer revolution began.

DISKPRINT will forever free you from having to search for or through that looseleaf binder with all your disk directories listed (if you have one!), or having to spend valuable minutes or hours scanning through each disk directory on-screen looking for that one particular program. DISKPRINT does for you what your computer was supposed to do for you, it eliminates the drudgery of paper shuffling and wasted time.



HERE'S HOW DISKPRINT WORKS...

Simply insert DISKPRINT into one of your disk drives and power up the system. DISKPRINT is self-booting and will load and run automatically. Now load our DISKPRINT custom forms into your printer and switch it on. Insert any disk in your library into your drive and answer the two prompts (which drive and disk title). If you do not choose to title your disk, DISKPRINT will default to its own. Press (RETURN) and DISKPRINT will print out a listing of every program on the disk along with extensions and sectors used as well as the number of free sectors available. This whole process will take about 20 or 30 seconds of your valuable time, so you can run through a one-hundred disk library in about 45 minutes or so. After your form has been printed, advance the next form to the print bar, insert another disc and go again. When you have run through your library, simply remove the tractor strips from the forms and insert them in the jackets with the disks. ***Every disk in your library will have a printed directory in the jacket with it at all times!***

DISKPRINT is delivered with our program on disk, 50 custom DISKPRINT forms and complete instructions. Refill packs of 50 forms each are available at your local dealer or directly from us by mail. (An order form is enclosed).

DISKPRINT is currently availble for:
• ATARI 400/800 & ATARI 825 or EPSON MX Series Printers • IBM-PC & IBM or EPSON MX Series Printers • APPLE II/APPLE II+ & EPSON MX Series Printers • TRS 80 Models I/III & TRS Series Printers • CMB 4032/8032 & CMB/EPSON MX Series Printers

DISKPRINT is now being converted to run on many more makes and models of computers. Call or write for further information.

COMPLETE DISKPRINT KIT FOR ALL MACHINES $24.95
Refill Packs of 50 Custom Forms $8.95
Please add $1.50 per order for P&H

ARMA DESIGN GROUP

"We make the computer revolution a little less revolting." ©

ARMA DESIGN GROUP • P.O. BOX 839 • DEPT SS1182 • ASBURY PARK, N.J. 07712 • 201-774-9099

NAME _____

ADDRESS _____

CITY STATE ZIP _____

TELEPHONE_____ **COMPUTER SYSTEM:**_____

☐ **MASTERCARD** ☐ **VISA** ☐ **MONEY ORDER** ☐ **CHECK** • **INTERBANK# (M.C.)** _____

CREDIT CARD # _____

*For a review of another product available for the IBM®-PC see page 71.

The Millionaire

Reviewed by Howard A. Karten

from Micro-Z Applications, 22704 Ventura Blvd., Suite 141, Woodland Hills, CA 91364. System requirements: IBM® PC with 64K RAM, one disk drive and either a monochrome or color adapter. Retail price is $49.85. It is also available for the Apple™.

If there's one thing more satisfying than setting a new record playing *Pac Man*, it would have to be making money. Few things are sweeter than the sound of money growing and compounding. *The Millionaire* successfully captures the interest and excitement — as well as the inherent risk — of playing the stock market.

As a kid, I had a zillion questions about the stock market. I directed them to my father, who had his own problems with the market. Invariably, he'd give me a book that was being distributed by Merrill, Lynch, Pierce, Fenner and Beane, (That's how long ago it was.) and suggest that I invest an imaginary $10,000 and follow it. That usually worked for three days, after which the requirements of detailed record keeping would force me back to the basement for another round of rocket and robot building — at least until the next go-round with the market.

Modern day parents — those who've wisely chosen to invest their funds in a computer, rather than in the market, can tell their kids to go play with the computer. For those kids (and adults as well) who want to try their hand at playing the market, this will be an educational, perhaps eye-opening, experience.

At the start of the game, each player is given $10,000 to invest in one of 15 companies, such as IBM, Exxon, Kmart, and so forth. Players buy and sell stock as often as they wish. When all the trades for a particular week are finished, the market is advanced to the next week by entering the "WEEK" command. At that point, the game gives players realistic news about business and the economy, as well as a summary of market activity and closing prices. The movement of individual stocks, and of the market as a whole, shows no particular pattern (at least, commendably, none discernable without some detailed investigation).

When the time comes to trade stocks, a highlighted area at the top of the screen displays the positions of the players with information such as net worth, cash position, loans outstanding, and so forth. This menu-driven game allows players to display graphs of market movement, buy and sell stocks, display individual holdings and net gain or loss in each stock, and so forth. When players wish to quit, they can either SAVE the current status of the game for future sessions or QUIT and start afresh.

Players can also trade in *puts* and *calls,* low-cost options that allow stock market traders to leverage their investments. (A *put* or *call* is an option, purchased like a stock but at a fraction of the cost, that allows the buyer to buy or sell a stock at a specified ("strike") price. *Puts* and *calls* are also used to lock up a particular profit or loss, but are distinct from share ownership. In the real-life stock market, individuals wishing to trade in options must meet certain financial criteria; in *The Millionaire*, traders must have a minimum net worth of $18,000 to buy *call* options or to purchase stocks on margin, and $40,000 to buy *put* options.

"The Millionaire successfully captures the interest and excitement — as well as the inherent risk — of playing the stock market."

Unfortunately, it would take several hours of very lucky play to achieve these minima, and therein lies an interesting suggestion for future enhancements: in future editions of *The Millionaire,* players might be allowed to select their own starting net worth to provide for wider latitude of play. Another enhancement, one that should be achievable with a reasonable amount of effort, might permit traders to sell short, as in the real stock market.

Similarly, providing some split-screen capabilities, such as viewing the news or market trends on the same screen as your portfolio, would enhance the game, as would a printer toggle switch to keep a running record of trades.

Overall, however, the game seems well-designed and well-thought-out. The weekly business and economic news in the Micro-Z Journal has a "real" feel to it, and in this game, the market sometimes drops following ostensibly good news, just as in real life. *The Millionaire* is easy to use, contains prompts, and will not succumb to obvious attempts to fool or crash the game (e.g. selling a stock you do not own).

The accompanying instructions are somewhat sparse, though adequate. (Incidentally, the instructions are also a decent, if necessarily abbreviated, introduction to the real-world market.)

The Millionaire does not have the flashy graphics and exciting sound effects of "*Intergalactic Hyperspace Cosmic Laser Blaster,*" so those used to that sort of thing will probably be bored. It effectively mimics the action of the real market — which is to say, it moves at a stately pace with very little direct (or obvious) connection between input and output.

The Millionaire is not geared to an adult market. Those who already play the market (Oops! — Wall Street prefers the terms "invest" and "speculate.") will not find it exciting. Those who do not, or who let others do their investing, will probably not find their interest piqued. For kids, however, — especially those precocious ones interested in business and how the world works — this might well be an educational game. But watch out — *The Millionaire* might only whet their appetite for some real-life action.

appleFest®

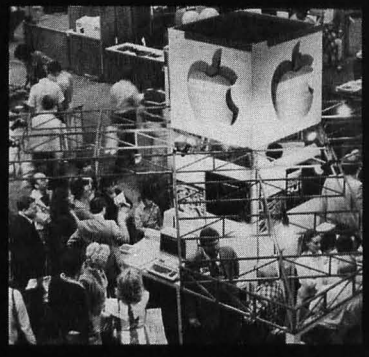
APPLE EUPHORIA

"Absolutely the best show I've ever been to in my life."

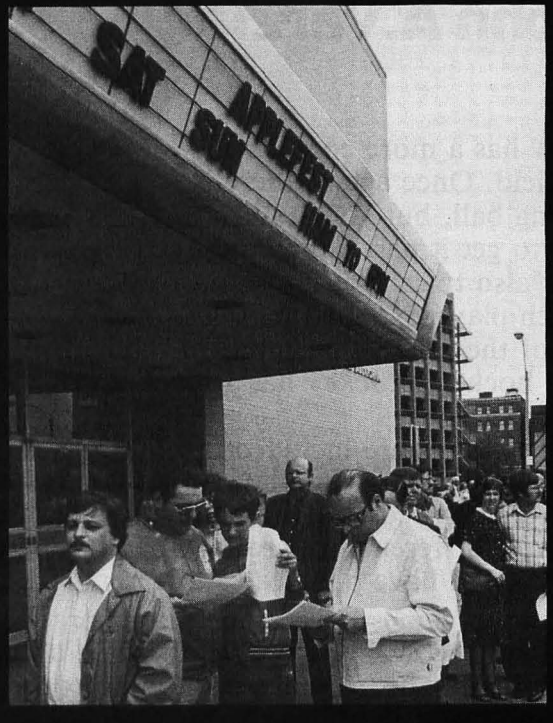
"It was incredible—I never knew there were so many products available for the Apple."

These are some of the things people are saying about Applefest, the world's largest exposition of Apples and Apple compatible products.

At Applefest, you can see and try out everything new for your Apple. Software for every conceivable application from arcade games to investment programs, music to machine language, teaching systems to accounting packages, word processors to graphics processors. Hundreds of peripherals, including printers, hard disks, modems, memory cards, video displays and synthesizers, plus accessories, publications, support services and more! Over 5,000 products are displayed and available for purchase at super show prices.



Get a taste of Apple Euphoria. Don't miss Applefest when it comes to a city near you.



Applefest/Minneapolis
Thursday-Sunday
September 16-19, 1982
Minneapolis Auditorium
Admission: Adult $5 per day, Child $3 per day
Show Hours: Thurs. 11 AM to 7 PM, Fri. and Sat. 11 AM to 7 PM, Sun. 12 NOON to 6 PM

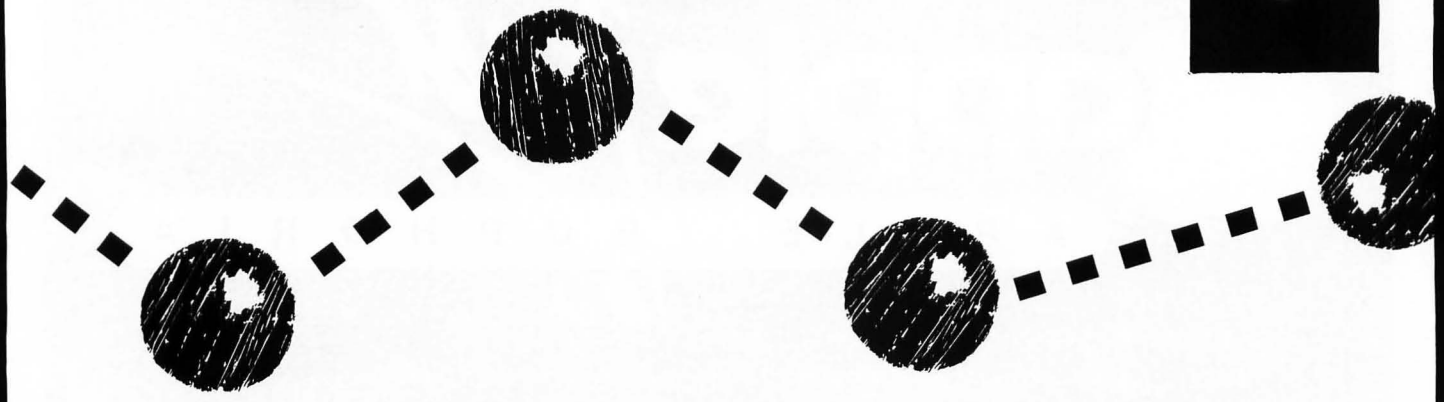
Applefest/Houston
Thursday-Sunday
October 28-31, 1982
Houston Civic Center
Admission: Adult $5 per day, Child $3 per day
Show Hours: Thurs. 11 AM to 7 PM, Fri. and Sat. 11 AM to 7 PM, Sun. 12 NOON to 6 PM

Applefest/San Francisco
Thursday-Sunday
November 18-21, 1982
Brooks Hall
Admission: Adult $6 per day, Child $3 per day
Show Hours: Thurs. 11 AM to 7 PM, Fri. and Sat. 11 AM to 9 PM, Sun. 12 NOON to 6 PM

For information on exhibit space or tickets, call or write:
Northeast Expositions, 824 Boylston Street, Suite 202, Chestnut Hill, MA 02167 (617-739-2000)

Ping Pong

TRS-80® DV SoftSide



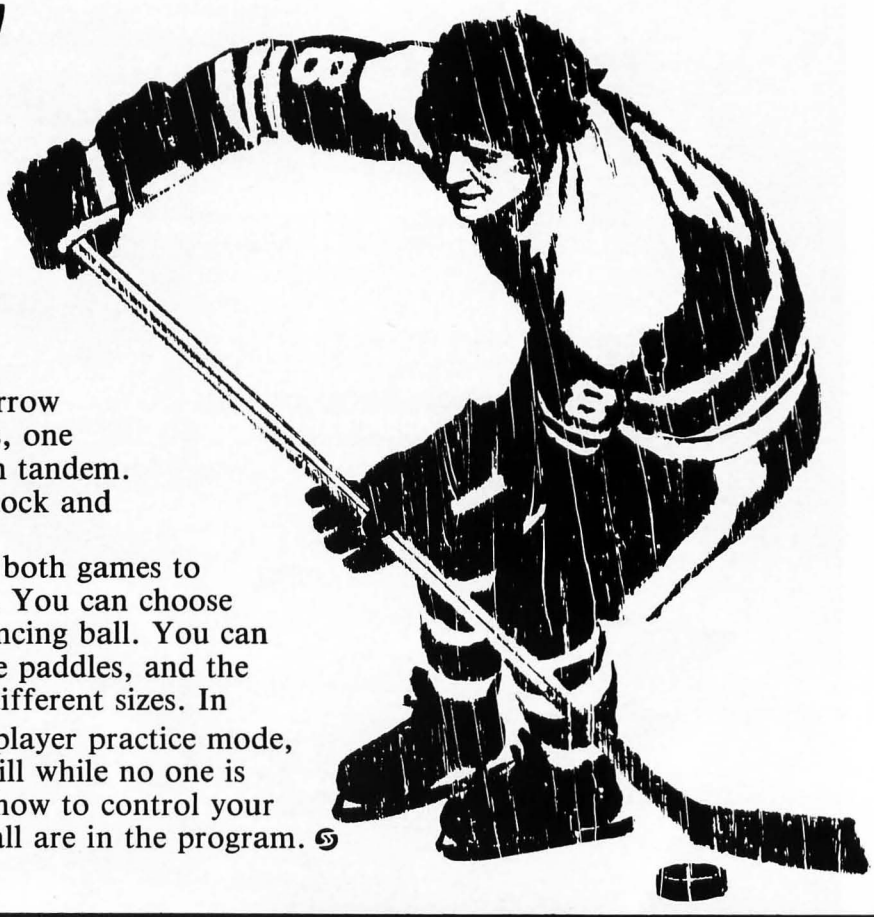
&

Hockey

by John A. Varela

***Ping Pong & Hockey* is an arcade-style game duo with sound, for one or two players, using a TRS-80® Model I or III with 32K RAM and a disk drive. It is included as the bonus program on issue #35 DV.**

Here are challenging versions of two classic arcade games. *Ping Pong* arms you and your opponent with small graphics paddles. Your object is to get the bouncing ball past your opponent. Each player serves five times in succession before the service changes hands. The first player to reach 21 points is the winner.



Hockey has a more complex playing field. Once again, there is a bouncing ball, but your task is not only to get it past your opponent, but also to send it into a narrow goal. Each player has two paddles, one in front of the other, that move in tandem. Like real hockey, *Hockey* has a clock and four timed periods.

Special features are included in both games to tailor the challenge to the players. You can choose three different speeds for the bouncing ball. You can choose three different sizes for the paddles, and the two players can have paddles of different sizes. In addition, both games have a one-player practice mode, which allows you to hone your skill while no one is watching. Simple instructions on how to control your paddles and put English on the ball are in the program.

Asylum

Reviewed by Allen L. Wold

by Frank Corr, Jr. and William F. Denman, Jr. (Med Systems Software, P.O. Box 2674, Chapel Hill, NC 27514. System requirements: 16K TRS-80® Model I or III. Cassette version also contains 32K version with extended messages and larger vocabulary. Suggested retail prices: cassette — $19.95, disk — $22.95. *Asylum* is also available for the IBM® PC with 48K and monochrome adapter and display.

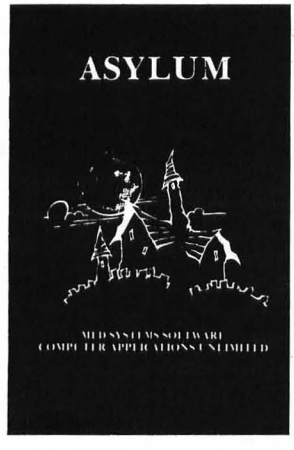
You begin in a room with only a bed, a box, and a locked door. You are wearing a jacket, with a few interesting and unexpected things in the pockets, but no keys. The clock in the upper right hand corner of the screen reads 9:30 PM. You have until 5:30 AM, when the day shift returns, to get out of the asylum. There is no score for anything other than escape.

There are some things in the box which can help you escape from the room, though it may take you a few tries to solve this first puzzle. Once you determine how to get out of your room, you find yourself in the corridors of the asylum, which recede into the distance until you near a corner.

On either side of the corridor are doors, most of which are locked. Even if you have a key, it might not be the right one for that door. Meanwhile, time is ticking away....(Actually, each minute of *Asylum* time is equal to 40 seconds of real time, so you have five hours, 20 minutes instead of eight hours in which to escape.) You will occasionally meet inmates who seem to be friendly and helpful, but be careful about turning your back to them.

You may ask for an inventory of what you are carrying. The list replaces the image on the screen, which returns when you hit ENTER. Boxes are used to represent all objects in the asylum, except for guards, inmates, and beds. To find out what is in a box, or what the box represents, you must examine it. You may take the contents, leave them there, or get only some of the contents if you wish.

The vocabulary of commands is large, and the computer understands a rather sophisticated syntax, such as "get all on the table except the matches." It is a good idea to review the vocabulary after playing a few times. The terms listed will help you understand the kinds of things you will be expected to do, and give you some clues to solutions of some of the puzzles. Some of the most commonly used words, such as "door," "open," etc., can be



replaced by single letters, such as "d," "o," etc. Articles, i.e. "a," "the," need not be used. The vocabulary list replaces the image, and is several pages long.

You may ask for help or a hint. In certain difficult spots, "help" or "hint" will give you a specific hint for the problem at hand. There isn't always help or a hint to be given, however, in which case it will simply tell you to examine things or ask for the vocabulary.

Inmates will talk to you sometimes, and when they do, their dialogue appears at the top of the screen. If you are recaptured (or killed), the invitation to try again appears at the bottom of the screen.

The eight page instruction manual tells you how to move (using arrow keys), syntax rules, how to manipulate objects, open doors, and how to get objects from characters. There is a very general section on hints. However, for an extra dollar, you can get a hint sheet. As in all Med Systems games, the hint sheet is in a simple substitution code, with the key, so that you cannot inadvertently read a hint. As some of the puzzles are extremely tricky, this hint sheet is almost a must.

The game on disk is self loading. The tape version has special loading instructions, and comes in two sizes, 16K and 32K, one on each side of the tape. All puzzles are identical in the long and short versions. The 32K version just has longer messages and more vocabulary words.

The game is highly visual, though the graphics are low resolution. Every room and corridor is drawn in outline perspective, and your point of view is within the asylum and its mazes. Movement takes place in discrete jumps, rather than a smooth flow. This is less than realistic, but it gives you time to think — and you'll need it. The clock ticks on, whether you're standing or moving, so you can't think too long.

The asylum is divided into four areas of unequal size: the asylum itself, with its cells and inmates (and wandering guards); the administrative wing; and two large mazes. You cannot exit the complex without going through all four areas.

Every door opens onto a new puzzle. Even the apparently empty rooms serve some function, such as storage of your loot. You may drop things anywhere, but in some places a mad custodian will appear, stealing everything lying on the floor. Then you'll have to find his hiding place to get it back.

You'll need graph paper to navigate the mazes, since there are some parts which resemble each other, and cannot be distinguished

from each other except by context of the other parts nearby. Each maze is two-dimensionally logical, as in any ordinary maze drawn on paper, but there are tricky places with mirrors, spirals, and teleport squares which will send you into other parts of the maze without warning. Completely mapping even a portion of a maze can take many, many hours.

A few items serve only to confuse, but most have a function in the solution of a later puzzle. Indeed, the whole asylum is an interlocked series of puzzles. The final solution can only be achieved by solving all the little ones along the way.

Encountering a guard can return you to your cell to start over if you're not careful. Failure to placate certain inmates will do the same. In certain areas of the asylum there are alarms which will ring with the same result, unless you have figured out how to circumvent them.

Death does not come often in the asylum. The most frequent cause of failure is recapture, though there are a few things, such as a falling piano, which will actually kill you. If you enter certain portions of a maze without knowing the special and tricky means of getting out again, you will remain there forever.

There seems to be very little randomness in this game. You need cigarettes to bribe one of the inmates, and the number you have is randomly determined. But for the most part, you are not competing against an irrational number generator, but against the diabolical ingenuity of the game's designers.

If you have to stop in the middle of a game, you may save your present position. The save feature is extremely helpful. After you have figured out certain parts of the puzzle, you can save the game at that point. If you make a mistake later, you won't have to start over from scratch. There is room for ten games to be saved, so you can try different strategies.

This type of game is difficult to review comprehensively. I spent a great deal of time simply exploring it. One of the co-authors claims he can get out in 45 minutes real time. Think about that for a minute — he designed the mazes, designed the puzzles and traps, wrote the code, and it still takes him 45 minutes to play *Asylum*. Those of us who must start from scratch have a lot to learn. In terms of playing time before you solve the puzzle, you'll more than get your money's worth. You'll have to play the game continually for weeks before being able to get out in the time allowed. Most of your clock time will be spent thinking about the puzzles and trying to find solutions.

In case you figure out how to escape the asylum, don't despair. There is a sequel, *Asylum II,* which makes the first seem like finding your way out of your own living room.

THE LEMON™ SOURS SURGES

Our crop-The Lemon™, The Lime™, and The Orange™ are designed to eliminate undetected submicrosecond overvoltage transients from electrical circuits. Commonly referred to as "spikes", or "glitches", these transients can cause hardware and software damage to unprotected circuits.

Today's electronic products are often microprocessor controlled - mini and micro computers, televisions, video cassette recorders - to name a few. Each of these products is sensitive to fluctuations in electrical power lines. Power switching devices such as refrigerators coming on and off or air conditioners starting up can be responsible for a momentary surge or spike of electricity in a circuit. Even your local utility stepping-up transformers to add power at peak load times or an electrical storm passing through can trigger surges. Such surges can cause equipment to falter at times, not to work at peak performance or fail completely. An entire data base can be lost.

Now you can prevent this from happening to you with an AC Surge Protector from Electronic Protection Devices. Each Protector is a solid state clamping device with 6 outlets utilizing modern high speed semiconductor technology. Using our Protectors is as simple as plugging it into any standard three wire duplex outlet then plugging what needs protection into it. Each Protector exceeds the IEEE 587-1980 Guide for Surge Voltages in Low Voltage AC Power Circuits.

When you compare the cost of computer hardware, software and your time with the price of a Protector (from $59.95 to $139.95), you'll want to sour your surges with one of the AC Surge Protectors from EPD, which are available through your local dealer.

Electronic Protection Devices
5 Central Avenue
Waltham, Massachusetts 02154

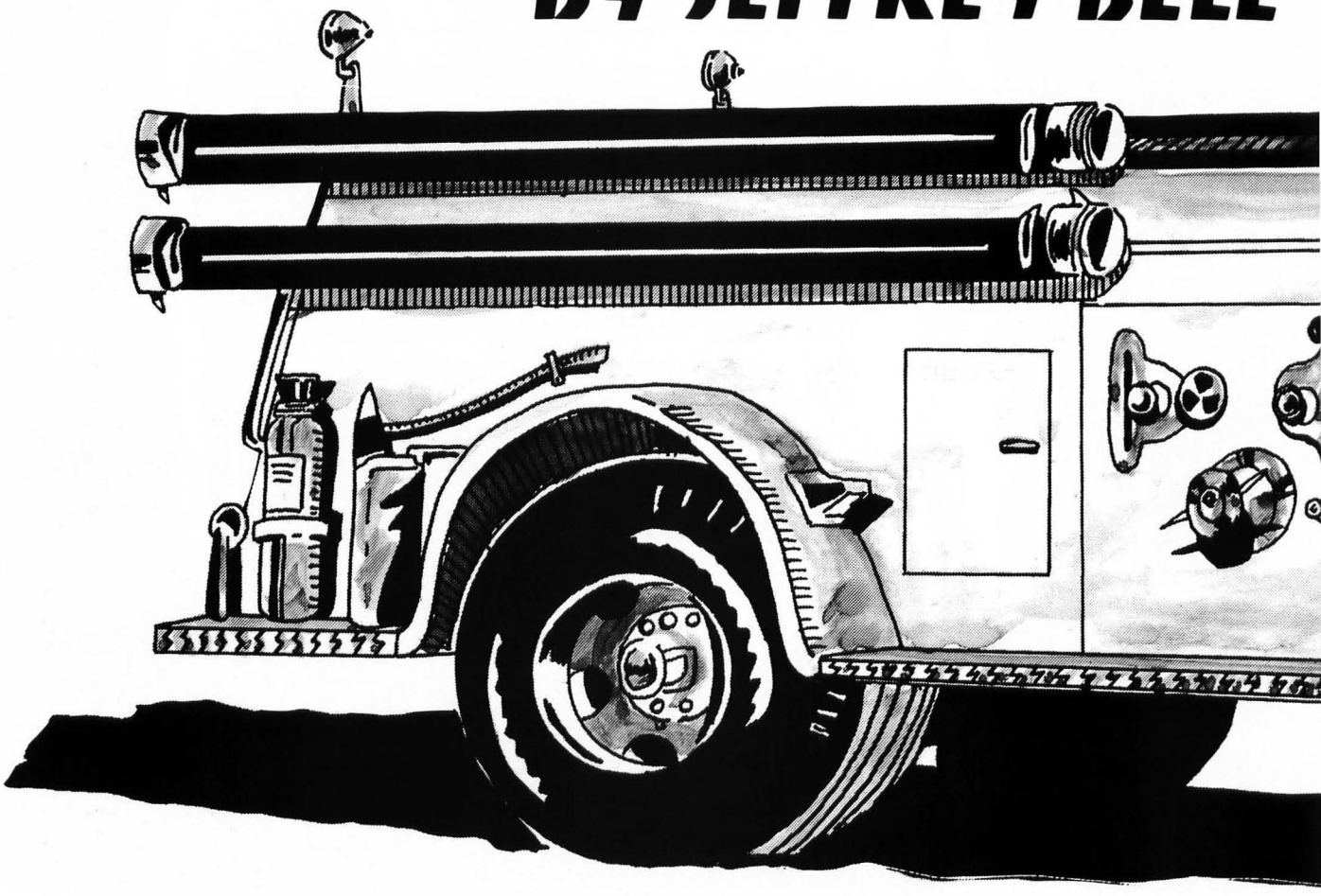
In Massachusetts Call:
(617) 891-6602

Outside Massachusetts Call:
1-800-343-1813

Dealer Inquiries Invited

FIREMAN

By Jeffrey Bell



***Fireman* is an arcade-style game for the TRS-80® Model I or III with 16K RAM.**

You are a heroic *Fireman,* singlehandedly attempting to rescue the helpless victims of a raging blaze in a high-rise building. You are armed only with a net, whose motion you control with your TRS-80's keyboard. With it, you must catch the leaping, terrified people. Once you let three of these hapless folks drop to their demise, the game ends.

You will encounter some opponents to your altruistic goals. There are the "Balloonheads," whose main characteristic is an enlarged noggin, and whom the wind tends to blow about, making their rescue rather difficult. Also, a piece of burning debris may explode on your net if you catch it in error.

Each time you catch twenty people, you move one level closer to the top of the building. Naturally, at the higher levels you have less time in which to react to falling objects and persons. You have the option of starting at a higher level, so that you don't have to work through all the easier levels unless you so desire.

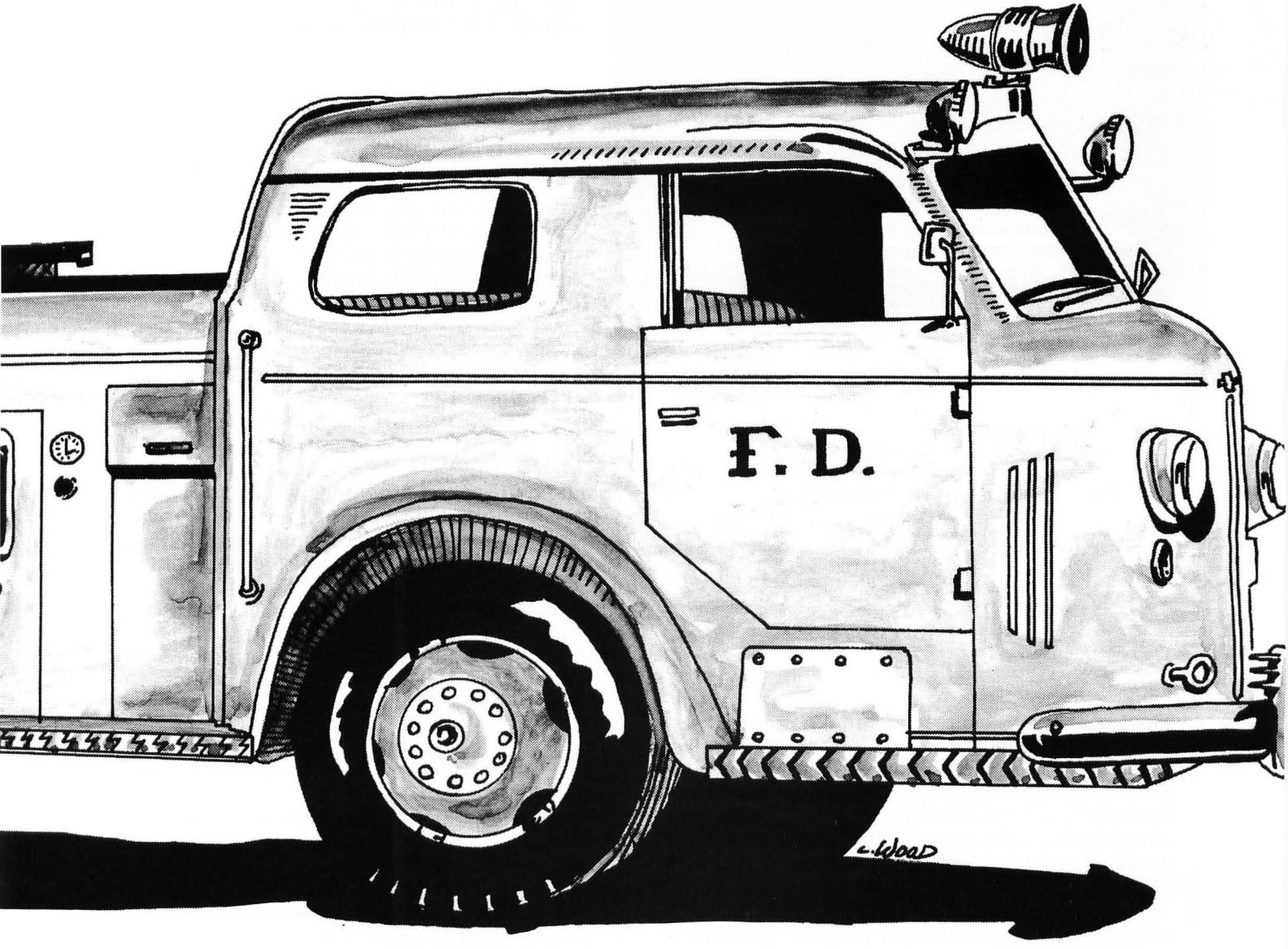
As there are but six windows in the building, objects can fall to only six places. There are two ways to control your net. Pressing the D, F, G, H, J, or K key immediately places your net in the corresponding location. You may alternatively use the arrow keys to move the net in the corresponding directions.

Scoring is as follows: catching a victim — 10 points; catching a "Balloonhead" — 20 points; catching flaming debris — *lose* 100 points.

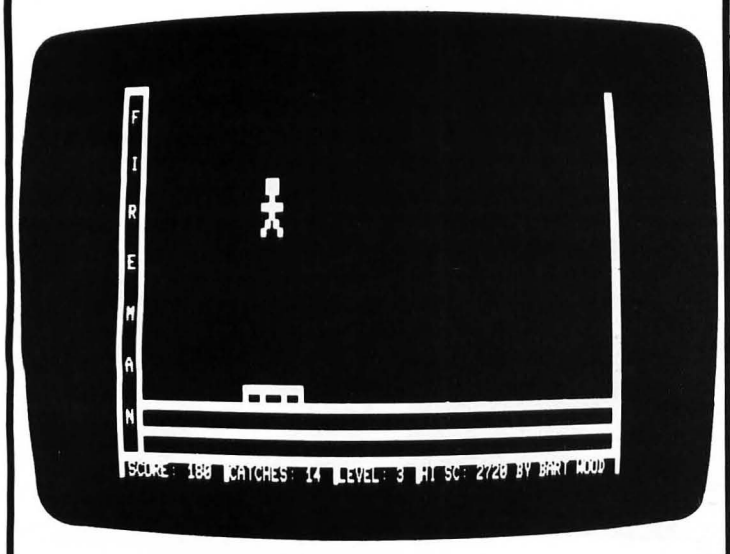
Fireman has sound, so be sure to connect an amplifier to the AUX jack of your computer.

Variables

A$(i,j): Graphics strings for the people, balloonheads, and debris.



A: If zero, use arrow controls; if one, use six-key controls.
BK$: Consists one line feed and three backspaces; used to reposition cursor in A$(i,j) and other strings.
CT: Number of catches.
DM$: Dummy string used to implement the sound routine.
ER$: Blank spaces to erase A$(i,j).
EX$: Graphics for explosion.
F: Position (1 to 6) in which object is falling.
FL: Flag to determine if "Balloonhead" has shifted position.
FP: Position (used in a PRINT @ statement) for a falling object.
F1: Randomly selected new position for a shifted "Balloonhead."
F2: New screen position for a shifted "Balloonhead."
F1$: Flower at left of gravestone.
F2$: Flower at right of gravestone.
GL: Number of catches needed to reach the next level.
HS: High score.
HS$: Name of high-score holder.
I,J,K,L,M: General loop variables.
IN$: Dummy variable for INKEY$.
LP: Screen position for current level. Starting value is 896.
LV: Level.
MD: Which word of MS$ is being printed.
MS$(i): Each element is a word of the phrase "Would you like instructions?"
M1$: Top half of FIREMAN block letters.
M2$: Bottom half of the block tables.
N$: The net.
PO: Position (1 to 6) of the net.
PS: Screen position of the net.
PX: Value to POKE into location 32766.
RIP$: The gravestone.
RC: Number of deceased victims.
SC: Score.
TY: Type of object falling.
X: Work variable for READing DATA; also the value of addresses for keyboard scan.
XX: XX = USR(0).
YN$: Yes/no response regarding the user's need for instructions.




```
SS SS SS SS SS SS SS SS SS SS SS
SS                           SS
SS  TRS-80 Model I/III BASIC  SS
SS          "FIREMAN"        SS
SS    Author: Jeffrey Bell   SS
SS     Copyright (c) 1982    SS
SS SoftSide Publications, Inc SS
SS                           SS
SS SS SS SS SS SS SS SS SS SS SS
```

Most of the variables, especially the graphics strings, are initialized here. Lines 20-60 use five loops to create the graphics strings for the people, Balloonheads, and debris. Line 130 POKEs the sound routine (by John L. Brandolini) into memory. The value POKEd into location 16783 is the note duration; the value at location 16784 is the pitch.

```
10 CLEAR1000:DIMA$(2,2):BK$=CHR$(26)+STRING$(3,24):GOTO4000
20 CLS:PRINT@394,"PLEASE WAIT ONE MOMENT WHILE I LIGHT THE MATCH
.";:FORI=0TO2:FORJ=0TO2:FORK=1TO2
30 FORL=1TO3:READX:A$(I,J)=A$(I,J)+CHR$(X):NEXTL
40 A$(I,J)=A$(I,J)+BK$:NEXTK
50 FORM=1TO3:READX:A$(I,J)=A$(I,J)+CHR$(X):NEXTM
60 NEXTJ:NEXTI
70 HS=2720:HS$="BART WOOD"
80 EX$=CHR$(137)+CHR$(146)+CHR$(150)+CHR$(188)+CHR$(169)+CHR$(16
1)+CHR$(134)+CHR$(26)+STRING$(7,24)+CHR$(176)+CHR$(178)+CHR$(182
)+CHR$(191)+CHR$(185)+CHR$(177)+CHR$(176)
90 N$=CHR$(183)+CHR$(179)+CHR$(183)+CHR$(179)+CHR$(187)+CHR$(179
)+CHR$(187)
100 F1$=CHR$(136)+CHR$(184)+CHR$(152)+BK$+CHR$(176)+CHR$(186):F2
$=CHR$(164)+CHR$(180)+CHR$(132)+BK$+CHR$(176)+CHR$(181)
110 RIP$=CHR$(152)+STRING$(3,131)+CHR$(164)+BK$+CHR$(24)+CHR$(24
)+CHR$(149)+"R"+"I"+"P"+CHR$(170)+BK$+CHR$(24)+CHR$(24)+CHR$(181
)+STRING$(3,176)+CHR$(186)
120 ER$="   "+BK$+"   "+BK$+"   "
```

```
130 DM$="":FORI=1TO29:READJ:DM$=DM$+CHR$(J):NEXTI:POKE16782,201:
IFPEEK(16396)=201THENPOKE16526,PEEK(VARPTR(DM$)+1):POKE16527,PEE
K(VARPTR(DM$)+2)ELSEDEFUSR=PEEK(VARPTR(DM$)+1)+PEEK(VARPTR(DM$)+
2)*256+65536*(PEEK(VARPTR(DM$)+2)>127):CMD"T"
```

Initialization and screen set-up. Draw the borders, using the SET command, and print necessary information.

```
200 LP=896-(LV-1)*64:GL=20:POKE16783,60
210 CLS:PRINT@66,"F":PRINT@194,"I":PRINT@322,"R":PRINT@450,"E":P
RINT@578,"M":PRINT@706,"A":PRINT@834,"N";
220 PRINT@897,STRING$(62,176);:FORI=0TO43:SET(2,I):SET(7,I):SET(
124,I):SET(125,I):NEXTI:FORI=2TO7:SET(I,0):NEXT:PRINT@962,"SCORE
:";:PRINT@973,CHR$(138)"CATCHES:";:PRINT@986,CHR$(138)"LEVEL:";L
V;:PRINT@996,CHR$(138)"HI SC:";HS;"BY ";HS$;
230 FORI=900TO900-(LV-1)*64STEP-64:PRINT@I,STRING$(58,176);:NEXT
I:PO=RND(6):GOSUB1140
240 PRINT@961,CHR$(133);:PRINT@1022,CHR$(138);
```

Main body of the program. Line 300 chooses which type of object will fall and the screen location from which it will fall. Line 310 actually makes it fall, by printing it at progressively lower locations. Lines 320-400 handle the eventuality of an object's reaching the ground. If a victim is caught, then beep, erase victim, and award points. If a victim dies, then print a memorial marker (lines 330-350). If a burning object is caught, then explode.

```
300 FL=0:F=RND(6):FP=F*9:TY=RND(20):IFTY<12THENTY=1ELSEIFTY<17TH
ENTY=0ELSETY=2
310 FORI=0TO2:GOSUB1000:PRINT@FP,A$(TY,I);:NEXTI:FP=FP+64:IFFP<L
P-128THENPRINT@FP-64,"   ";:GOTO310
320 IFF=POTHEN360
330 PRINT@FP-64,ER$;:PRINT@FP,A$(TY,2);:FORI=1TO30:NEXT:PRINT@FP
,ER$;:PRINT@FP+128,STRING$(3,176);:IFTY=2THEN300ELSERC=RC+1:PRIN
T@FP,RIP$;:PRINT@PS,STRING$(7,176);:IFRC>1THENPRINT@FP+61,F1$;
340 IFRC=3THENPRINT@FP+69,F2$;:GOTO3000
350 GOSUB5000:PRINT@FP,"     ";:PRINT@FP+61,"          ";:PRINT
@FP+126,STRING$(10,176);:PRINT@PS,N$;:GOTO300
360 IFTY=0SC=SC+20:PX=200:GOTO400
370 IFTY=1SC=SC+10:PX=100:GOTO400
380 PRINT@FP-64,ER$;:PRINT@FP-1,"BOOM!";:PRINT@FP+62,EX$;:GOSUB6
000:PRINT@FP-1,"      ";:PRINT@FP+62,"      ";:PRINT@FP+126,STR
ING$(6,176);:PRINT@PS,N$;:SC=SC-100:IFSC<0THENSC=0
390 PRINT@968,SC;:PRINT@973,CHR$(138);:GOTO300
400 PRINT@FP-64,ER$;:POKE16784,PX:XX=USR(0):PRINT@968,SC;:PRINT@
973,CHR$(138);:CT=CT+1:PRINT@982,CT;:PRINT@986,CHR$(138);:IFCT=6
LTHEN2000ELSEGOTO300
```

Subroutine to shift a windswept Balloonhead or to move the net if an appropriate key has been hit. If a key has been hit, check various keyboard addresses to see which one was hit.

```
1000 IFTY<>0ORFL=2THEN1020
1010 IFFP<(LP-256)ANDFP>(LP-448)ANDRND(5)=1THENF1=RND(6):F2=FP-F
#9+F1#9:PRINT@FP,ER$;:FP=F2:F=F1:FL=FL+1
1020 IFATHEN1060
1030 IFPEEK(14400)THEN1040ELSERETURN
1040 IF(PEEK(14400)AND32)AND(PO>1)THENPO=PO-1:GOTO1130
1050 IF(PEEK(14400)AND64)AND(PO<6)THENPO=PO+1:GOTO1130
1060 IFPEEK(14337)ORPEEK(14338)THEN1070ELSERETURN
1070 X=PEEK(14337):IFXAND16PO=1:GOTO1130
1080 IFXAND64PO=2:GOTO1130
1090 IFXAND128PO=3:GOTO1130
1100 X=PEEK(14338):IFXAND1PO=4:GOTO1130
1110 IFXAND4PO=5:GOTO1130
1120 IFXAND8PO=6
1130 PRINT@PS,STRING$(7,176);
1140 PS=LP+PO#9-2:PRINT@PS,N$;:RETURN
```

Routine to handle moving up a level.

```
2000 GL=GL+20:PRINT@24,"CONGRATULATIONS!!";:PRINT@76,"YOU HAVE S
UCCESSFULLY COMPLETED LEVEL #";LV;:PRINT@148,"(ENTER TO CONTINUE
 GAME)";:LV=LV+1:GOSUB7000:POKE16783,60:IFLV=13THENLV=12
2010 IFINKEY$<>CHR$(13)THEN2010ELSEPRINT@148,STRING$(25,32);:PRI
NT@24,STRING$(17,32);:PRINT@76,STRING$(42,32);:PRINT@PS,STRING$(
7,176);:PRINT@993,LV;:PRINT@996,CHR$(138);:IFLV<12THENLP=LP-64
2020 PRINT@LP+4,STRING$(58,176);:PS=LP+PO#9-2:PRINT@PS,N$;:GOTO3
00
```

End of game. Lines 3020-3050 are executed only if the high score is set. Line 3090 erases all the necessary variables to run the game over again, then branches to line 4500 to find out what type of control is to be used, and restarts the game.

```
3000 GOSUB5000:FORI=1TO1000:NEXT:CLS
3010 PRINTCHR$(23);:IFSC<HSTHEN3060
3020 PRINT:PRINT:PRINT:PRINT"THE GAME IS OVER, BUT .";:FORI=1TO5
:FORJ=1TO300:NEXTJ:PRINT" .";:NEXTI:CLS
3030 HS=SC:CLS:PRINT@324,CHR$(23);"YOU HAVE THE HIGH SCORE!!!!":
POKE16783,50:FORI=255TO5STEP-5:POKE16784,I:XX=USR(0):NEXT
3040 PRINT:PRINT:PRINTTAB(5)"PLEASE ENTER YOUR NAME":PRINTTAB(7)
"(MAX. OF 10 CHAR.)":PRINTTAB(10)STRING$(10,"-"):PRINTTAB(8);:IN
PUTHS$
3050 IFLEN(HS$)>10THENCLS:PRINT@324,CHR$(23);"YOU HAVE THE HIGH
SCORE!!!":GOTO3040ELSECLS:PRINTCHR$(23):GOTO3070
3060 PRINT@336,"### GAME OVER ###";
3070 PRINT@454,"DO YOU WANT TO PLAY AGAIN?";
3080 YN$=INKEY$:IFYN$="N"THENCLS:ENDELSEIFYN$<>"Y"THEN3080
3090 RC=0:CT=0:SC=0:GOSUB4500:GOTO200
```

The main part of this subroutine comprises the instructions. Lines 4500 on make up a sub-subroutine.

```
4000 CLS:FORI=1TO27:READX:M1$=M1$+CHR$(X):NEXT:FORI=1TO27:READX:
M2$=M2$+CHR$(X):NEXT:PRINT@212,M1$;:PRINT@276,M2$;
4010 PRINT@64,STRING$(64,153);:PRINT@960,STRING$(64,153);:FORI=6
4TO896STEP64:PRINT@I,CHR$(153)+CHR$(153);:PRINT@I+62,CHR$(153)+C
HR$(153);:NEXTI
4020 MS$(1)="WOULD":MS$(2)="YOU":MS$(3)="LIKE":MS$(4)="INSTRUCTI
ONS?"
4030 MD=0:PRINT@466,;
4040 MD=MD+1:IFMD=5THENPRINT@467,STRING$(40,32);:GOTO4030ELSEPR
INT" "+MS$(MD);
4050 FORI=1TO30:IN$=INKEY$:IFIN$="Y"ORIN$="N"THEN4060ELSENEXT:GO
TO4040
4060 I=0:IFIN$="N"THEN4180
4070 CLS:PRINTTAB(27);"FIREMAN":PRINT:PRINT"OBJECT:  TO CATCH PE
OPLE FALLING FROM A BURNING BUILDING":PRINTTAB(9);"BEFORE MISSIN
G AND ALLOWING 3 PEOPLE TO CRASH."
4080 PRINT:PRINT"DIFFICULTIES:  1) BALLOONHEADS--THESE PEOPLE TE
ND TO BE BLOWN":PRINTTAB(18);"ABOUT BY THE WIND, MAKING THEM HAR
D TO CATCH.":PRINTTAB(15);"2) BURNING DEBRIS MAY FALL AND LAND O
N YOU,":PRINTTAB(18);"DAMAGING YOUR NET."
4090 PRINT:PRINT"SCORING:  10 PTS. - CATCHING A NORMAL PERSON":P
RINTTAB(10);"20 PTS. - CATCHING A BALLOONHEAD":PRINTTAB(10);"-10
0 PTS. - HAVING A PIECE OF DEBRIS HIT YOU"
4100 PRINT:PRINTTAB(20);"PRESS ENTER TO CONTINUE"
4110 IFINKEY$<>CHR$(13)THEN4110ELSECLS
4120 PRINT:PRINT"FOR EVERY 20 SUCCESSFUL CATCHES, YOU MOVE UP ON
E LEVEL CLOSER":PRINT"  TO THE TOP OF THE BUILDING, MAKING IT HA
RDER TO CATCH THE":PRINT"  POOR VICTIMS.  AFTER YOU REACH THE 12
TH LEVEL, YOU GO UP NO"
4130 PRINT"  HIGHER.  NORMALLY THE GAME BEGINS ON THE FIRST LEVE
L, BUT YOU":PRINT"  HAVE THE OPTION OF STARTING AT A HIGHER LEVE
L."
4140 PRINT:PRINT"YOUR NET IS CONTROLLED BY EITHER OF THESE TWO M
ETHODS:":PRINT"    1) THE ARROW KEYS (";CHR$(93);CHR$(94);")--EA
SY":PRINT"    2) THE KEYS D,F,G,H,J, AND K--HARD"
4150 PRINTTAB(7);"THERE ARE ONLY SIX DIFFERENT POSITIONS THAT YO
UR NET CAN":PRINTTAB(7);"BE IN--PRESSING 'D' INSTANTLY PUTS YOU
INTO POSITION 1,":PRINTTAB(7);"PRESSING 'F' PUTS YOU IN POSITION
 2, ETC..."
4160 PRINT:PRINT:PRINTTAB(17);"PRESS ENTER TO BEGIN GAME";
4170 IFINKEY$<>CHR$(13)THEN4170
4180 GOSUB4500:GOTO20
4500 CLS:PRINT@335,"WHICH CONTROLS DO YOU WANT TO USE--";PRINT@4
70,"1--ARROW KEYS           OR";PRINT@534,"2--D,F,G,H,J, AND K?"
4510 IN$=INKEY$:IFVAL(IN$)=0ORVAL(IN$)>2THEN4510
4520 IFVAL(IN$)=1THENA=0ELSEA=1
4530 CLS:PRINT@330,"WHICH LEVEL DO YOU WANT TO START AT (1-12)";
:INPUTLV:IFLV<1ORLV>12ORLV<>INT(LV)THEN4530ELSERETURN
```

The funeral dirge for a poor victim who reaches the ground.

```
5000 POKE16783,255:POKE16784,200:XX=USR(0):FORI=1TO10:NEXT:XX=US
R(0):POKE16783,100:XX=USR(0):FORI=1TO5:NEXTI:POKE16783,250:XX=US
R(0)
5010 POKE16783,250:POKE16784,165:XX=USR(0)
5020 POKE16783,120:POKE16784,175:XX=USR(0):POKE16783,200:XX=USR(
0)
5030 POKE16784,200:POKE16783,100:XX=USR(0):POKE16783,200:XX=USR(
0)
```

```
5040 POKE16784,210:POKE16783,120:XX=USR(0):POKE16784,200:POKE167
83,255:XX=USR(0):POKE16783,60:RETURN
```

Taunting sound that results when player catches debris.

```
6000 POKE16783,100:POKE16784,150:FORI=1TO2:XX=USR(0):FORJ=1TO5:N
EXTJ,I:POKE16784,175:XX=USR(0):POKE16784,130:XX=USR(0)
6010 FORI=1TO5:NEXTI:POKE16783,150:POKE16784,150:XX=USR(0):FORI=
1TO3:NEXT:POKE16783,100:POKE16784,175:XX=USR(0):POKE16783,60:RET
URN
```

"Charge!" tune for the player who reaches a new level.

```
7000 POKE16783,80:POKE16784,220:XX=USR(0)
7010 POKE16784,165:XX=USR(0)
7020 POKE16783,110:POKE16784,130:XX=USR(0)
7030 POKE16783,220:POKE16784,110:XX=USR(0)
7040 POKE16783,100:POKE16784,130:XX=USR(0)
7050 POKE16783,200:POKE16784,110:XX=USR(0):RETURN

10000 DATA183,179,131,128,149,170,179,179,132,170,179,179,129,17
0
10001 DATA137,144,152,171,128,150,131,131,148,170,164,128,149
10002 DATA149,128,128,128,149,170,128,137,144,170,176,176,144,17
0
10003 DATA128,138,128,170,128,151,131,131,149,170,128,137,149
10004 DATA143,191,143,166,143,153,130,128,129
10005 DATA188,188,188,140,191,140,138,128,133
10006 DATA176,176,176,167,191,155,168,131,148
10007 DATA138,159,128,163,167,129,129,128,129
10008 DATA168,188,128,140,157,132,131,130,129
10009 DATA160,176,128,178,183,144,137,137,129
10010 DATA162,132,153,191,191,191,131,131,131
10011 DATA168,128,148,190,188,189,143,143,143
10012 DATA160,128,144,185,178,180,191,191,191
10013 DATA221,33,143,65,221,78,0,221,70,1,62,1,211,255,16,254,22
1,70,1,62,2,211,255,16,254,13,32,235,201
```

TRS-80® SWAT TABLE FOR:
FIREMAN

LINES	SWAT CODE	LENGTH
10 - 100	FK	534
110 - 220	NU	602
230 - 360	XY	521
370 - 1050	HP	532
1060 - 2020	JJ	543
3000 - 3090	SU	520
4000 - 4070	MK	525
4080 - 4120	ZX	611
4130 - 4160	CC	503
4170 - 5030	TP	536
5040 - 10000	LD	517
10001 - 10011	JY	503
10012 - 10013	ZF	141

EVERYONE NEEDS A . . .

UNINTERRUPTABLE POWER SOURCE

ADVANCED DESIGN
NEW RELIABILITY
GUARDIAN PROTECTS
ELECTRONIC SAFETY
LIFESAVER FOR DATA

$595

SAVE YOUR DATA FROM POWER OUTAGES!

BACKUP FOR YOUR COMPUTER, MONITOR, PRINTER AND 5¼″ FLOPPY AND HARD DISC DRIVE

• Automatically stops annoying problems from power line interruptions and brown outs • You need standby power to save data • Maintenance free backup power available in 115 volt or 220 volt • 50 or 60 HZ • 150 watts • Complete versatility — operate your system from a 12 volt source, i.e., automobile cigarette lighter, boat or airplane • Rugged self contained gel cell battery • No voiding warranty — no cutting wires • Automatic audio alarm warning tone during commercial power failure or interrupt • UL listed • FCC approved • Transient voltage suppressor gives added insurance from line voltage spikes, utilizing Zener Ray™ • Green/red LED power status indicator • Green — normal AC line power • Slow blinking red — at least 6 minutes of remaining standby power • Fast blinking red — approximately 2 minutes of remaining battery power • Solid state technology unexcelled by any UPS power unit in its class.

RH ELECTRONICS, INC.

566 IRELAN, BUELLTON, CA 93427

(805) 688-2047

SEE YOUR RH ELECTRONICS PRODUCTS DEALER

FOR YOUR APPLE II*:

SUPER FAN II™	$ 74.95
SUPER FAN II™/ZENER RAY™	$109.00
SUPER RAM II™	$125.00
RH 12 VOLT TRANSVERTER	$149.00

FOR MICRO COMPUTERS:

GUARDIAN ANGEL™	$595.00

Galaxy Invasion Plus

Reviewed by Mark E. Renne

by Bill Hogue and Jeff Konyu (Big Five Software, P.O. Box 9078-185, Van Nuys, CA 91409). System requirements: TRS-80® Model I or III with 32K (disk) and 16K (cassette). Suggested retail price for the disk version is $19.95. The tape version retails for $15.95.

It's more trouble from space! Columns of attacking aliens are traveling great distances to eliminate arcaders everywhere. Big Five has unleashed the wrath of Bill Hogue and Jeff Konyu once again.

Based loosely on a popular arcade game, the original *Galaxy Invasion* showed us how challenging computer arcade games could be. The new, improved *Galaxy Invasion Plus* has some additional features, but I'll cover those later. Right now, I'll give you a brief overview of the original game.

In this game, one or two players protect the universe and joystick support is provided. Four rows of ten alien ships are lined up in battle formation with flagships and body guards located at the top of the screen. Aliens break off, usually in groups of three, and attack your laser cannon which moves right and left at the bottom of the screen. At the beginning of play, three cannons are supplied by the federation, with bonus cannons awarded at each additional 10,000 points.

Occasionally, the message "flagship attack" will flash on the viewscreen. When this happens, a flagship must be destroyed as soon as possible or your cannon will be destroyed instead. The disk version features saving of the top ten scores and sound is output through the cassette port.

The additional features added to *Galaxy Invasion Plus* make a good game *great*. By pressing P, a player may temporarily pause execution of the game. This could be vital in case an important communique from starfleet, or your spouse, takes your attention away from the game. Talking prompts in this game are as clear and audible as any I've ever heard. When your ship is destroyed, a clear "Ouch!" or "Oh, No!" makes the loss seem more painful.

A Cosmic Storm, which totally inverts the screen, occurs if you score 50,000 points. You're then at the top of the screen and the invaders are at the bottom. No other game I've ever played has challenged my idea of reality as this one does. Just as I got used to aliens on the bottom, another storm would occur and position them back at the top. From 50,000 on, the screen is inverted at every 10,000 points. This effect presents a challenge that is not easily mastered after only a few plays.

The game's sound is excellent. A nice little tune opens the game, along with the usual assortment of beeps, bomps, and gorks. Aside from the voices I've already mentioned, effects are used to indicate extra ships, flagship attacks, player prompts, and cosmic storms. Cosmic storms also include nice visual effects.

Galaxy Invasion Plus is unique and challenging. It should be added to the repertoire of any serious alien fighter. Big Five always seems to answer the call for a game that's not only different, but more exciting than any before.

Sea Dragon

Reviewed by Dean F. Hayden-Macy

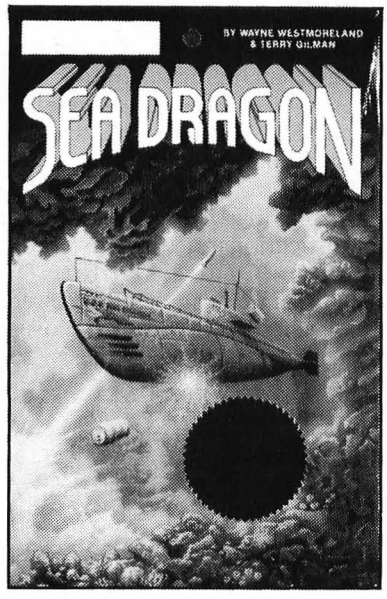
by Wayne Westmoreland and Terry Gilman (Adventure International, Box 3435, Longwood, FL 32750) System requirements: TRS-80® Model I or III, 16K RAM (tape), 32K RAM (disk). Suggested retail price: tape — $19.95, disk — $24.95.

You are the commander of the USS *Sea Dragon* Fleet. *Sea Dragon One* is the Flagship of the newest five ship nuclear submarine fleet built by the Navy. All submarines in the fleet have a complement of the most modern weapons devised by the government. The sea-to-sea and sea-to-air missiles are one-tenth the size of past missiles and carry a smaller, deadlier warhead. The forward torpedoes are heat-sensing devices about the size of basketballs. They are thruster propelled and absolutely reliable.

The Department of Intelligence has been monitoring coded transmissions from the enemy for several months. The decoded messages indicate that the enemy has been building automated nuclear reactor plants, deep under the ocean's surface, with the intent of using them to subdue the world.

Your adventure, as Commander on the *Sea Dragon*, will be a trip through Hell and back. You start with ship number five on the surface of the ocean. You will move at a set speed (difficulty level) and at no time prior to reaching the reactor will you be able to sit dead still in the water. You may move faster by pressing the right arrow and slower by holding the left arrow. To dive, you hold the down arrow. To move toward the surface for air or warfare, you press the up arrow. The air above the ocean surface is indicated by a dashed white line. (You cannot take on air unless you are fully surfaced.) You fire forward torpedoes as fast as you can by hitting the space bar. (There are no aft torpedoes.) Missiles are launched by pressing the ENTER key.

Quickly and quietly, you will navigate your submarine past underwater mountains, through narrow passageways, around and over deadly, rising, explosive mines, and under explosive stalactites and automated laser cannons. If you so much as lightly touch one of the weapons aimed at your sub, or touch a rock or the ocean floor, you will be immediately annihilated. The



weapons are triggered by your proximity to them and are very accurate.

Your mission is to destroy as many of the automated nuclear reactors as possible before you are discovered and annihilated. You will accomplish this by making the reactor destruction appear as an accident, forcing the enemy to abandon the project. The only way to fool the enemy into thinking that their plants are unstable is to fire a nuclear torpedo into the reactor core. This maneuver is not easy.

The reactor chamber is guarded by an upper and a lower laser cannon which fire heat-sensing missiles. The nuclear reactor is housed behind a double neutronium shield. The outer shield is extremely thick at the center, tapering off at each end. The inner shield continually rotates and is of uniform thickness.

To destroy the outer shield takes many, many direct torpedo hits. Once you've pierced this shield, you must fire a torpedo directly through a small hole in the rotating inner shield into the heart of the nuclear reactor core. Your submarine will be constantly on the move because you are under attack by automatic weapons which fire sensing missiles from below and above. You must dodge these because you cannot destroy the embedded guns. While dodging, you must refill your air tanks many times before you are able to complete your mission. If you are able to destroy this reactor, you will receive a 5000 point bonus and move on to another target.

Sea Dragon is an all-consuming, extremely difficult and engrossing graphic adventure. You may play it on either a novice or expert level, alone or with a friend. The Machine Language program is fast and has good sound effects. You may use the keyboard, joysticks or both during play. At the end of each game, you may save your high score permanently on a disk. (The tape version will not save scores.) If you select the two player game, each player may choose his own level of difficulty. Lastly, if you are called away from this absorbing game, you can freeze the action by pressing SHIFT S and then ENTER to continue.

The authors of *Sea Dragon* saved scores of over 3000 points on the commercial disk copy. I have managed over 131,000 points before collapsing into a nervous wreck. I have no idea how many points can actually be scored in this game.

One final warning. The game is habit forming. You will play until you give up in disgust, and then return again, time after time, unless your wife/mother hides your copy or until you destroy it. The frustration can be absolute!

SoftSide

ATARI® DV

DEADSTICK LANDING

by Al Ragsdale; Atari® version by Al Johnston

***Deadstick Landing* is an arcade/simulation program for an Atari with 48K RAM and a disk drive. It is included as the bonus program on issue 35 DV.**

The Space Shuttle Orbiter is really the world's fastest glider. Thrust into orbit by rocket engines, it must return to Earth in an unpowered glide known to pilots as a *Deadstick Landing*.

For most of the re-entry glide, the Orbiter is out of communication with the ground. This "blackout" is a result of the intense heat caused by air friction. At about Mach 10 (10 times the speed of sound), the blackout ends. The pilot then has sufficient communications and navigation data to locate and glide to the landing site. Ground controllers, including a Guidance Officer (GUIDO) and a Flight Dynamics Officer (FIDO), may assist him.

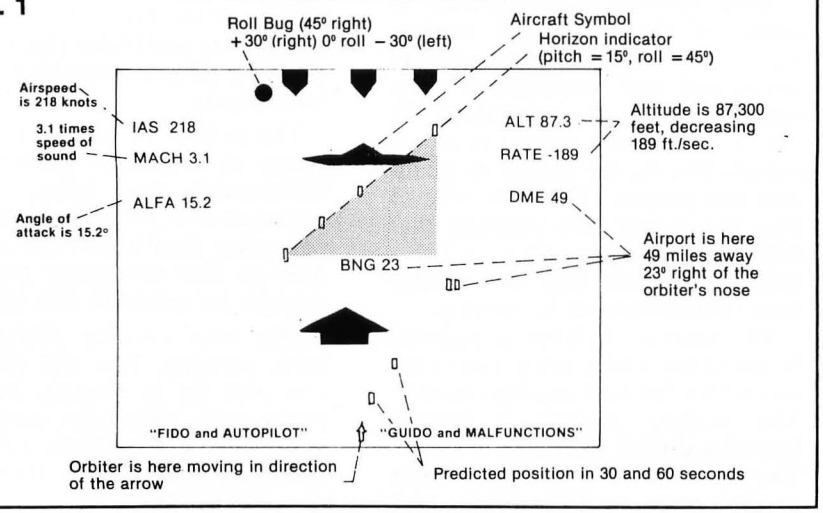
The data displays you will see while playing *Deadstick Landing* are simplifications of those available to a real Orbiter pilot. The aerodynamic and guidance equations were developed by the original author as part of a thesis for a Master's degree in Avionics Instrumentation.

Displays

Figure 1 is a typical *Deadstick Landing* display. The displayed quantities are described below.

FIG. 1

TYPICAL DEADSTICK DISPLAY



IAS — Indicated Airspeed (In Knots)

This is determined by the pressure of the air flowing about the Orbiter, and is about equal to the true airspeed at sea level. Most of a re-entry is flown at an IAS of around 200; this rises to about 300 just before the landing. If the IAS is too high, the Orbiter may be damaged; if it is too low, the Orbiter may stall, as the wings will not provide enough lift to keep flying.

Mach

The Mach number is the ratio of the Orbiter's speed to the speed of sound. This gives an estimate of true airspeed, regardless of altitude. Mach 1 is approximately 700 mph. The Mach number is the major factor determining the Orbiter's flight characteristics. Re-entry begins at Mach 25; before landing, the speed is about Mach 0.5. The simulation starts at Mach 10 to save time.

ALFA — Angle of Attack (in degrees)

This is the angle between the direction the Orbiter is moving (flight path angle) and the direction it is pointing (pitch). See Figure 2. ALFA determines the lift and drag forces. A large ALFA is necessary to generate enough lift in the thin air at high altitudes. ALFA starts as high as 40 degrees, and decreases slowly until it is less than 5 degrees before landing. See Figure 2.

ALT — Altitude (in 1000-foot increments)

This is your height above the runway. For example, an ALT of 41.5 means that you are 41,500 feet above the runway. Below 10,000 feet, the actual altitude is shown.

RATE — Altitude Rate (feet per second)

This is the change in altitude per second. Positive values of RATE indicate an increase in altitude; negative values indicate descent. A descent rate of up to 200 feet per second is typical for the Orbiter.

DME — Distance Measuring Equipment (nautical miles)

This is the range from the Orbiter to the start of the runway.

BNG — Relative Bearing (in degrees)

This is the direction to the runway relative to the current heading (direction the Orbiter is moving). A BNG of zero means the runway is straight ahead; positive BNG means the runway is to the right; negative BNG means it is to the left.

Altitude Display

The display at the top center of the screen is called the "roll bug" or "sky pointer." When it is in the center as in Figure 3, the Orbiter is in a wings-level attitude. Otherwise, the roll bug points towards the sky, opposite the bank angle. There are reference markers at 30 degrees left and right bank. In order to roll back to wings level, always roll towards the roll bug.

The horizon indicator consists of an airplane symbol and a moving horizon. The horizon tilts left or right according to the bank, and the airplane symbol moves up and down according to the pitch. Note that the horizon moves opposite the Orbiter's wings. In Figure 3, it is making a wings-level dive. Don't confuse pitch with ALFA. It is possible for pitch to be down 15 degrees while ALFA is 6 degrees. In fact, this is normal during the landing approach. Usually, pitching up increases the angle of attack and decreases the IAS, but does not necessarily result in a climb. See Figure 3.

Navigation Display

The arrow at the bottom center of the screen indicates the present position of the Orbiter (at the arrow's tail). The indicators near the arrow are position predictors. They predict where the Orbiter will be in 30 and 60 seconds at its present speed and bank angle. Figure 1 shows the Orbiter turning right; in Figure 3, it is flying straight ahead.

The location of the runway, relative to the Orbiter, is indicated by an arrow symbol on the lower half of the screen. If the runway is behind the Orbiter, then the arrow symbol will indicate only whether the airport is to the left or right. The scale of the navigation display changes in proportion to range until the landing site is 20 miles away.

Flight Controls

The pilot lands the Orbiter by controlling its pitch (up/down) and its roll (left/right). There is an autopilot that can fly all the landing maneuvers.

There are three levels, or modes, in *Deadstick Landing*: Rookie, Pilot, and Ace. In Rookie mode, the pilot has the following controls:

Joystick Controls

Moving the stick forward pitches down 2 degrees (1 when ALFA is less than 9); moving it back pitches up in the same way. Moving the stick to the right or left gives a 15-degree roll in the appropriate direction. A diagonal move results in simultaneous pitch and roll commands.

Keyboard Controls

! — pitch to the auto-pitch (no automatic repeat).
= — roll to the auto-roll (no automatic repeat).
> — roll to 45-degree right bank.
< — roll to 45-degree left bank.
Space Bar — roll to wings-level attitude.
A — turn on the *autopilot*.

Note: A beep confirms both keyboard and joystick commands, and the two may be entered simultaneously.

If the pilot's commands are out of limits, GUIDO and FIDO will suggest ways to bring the Orbiter back on course. If the pilot is making a drastic, potentially dangerous error, a warning message will advise the pilot to correct the situation.

In Pilot mode, the pilot has the same controls, and has the assistance of GUIDO and FIDO; however, random malfunctions and turbulence may occur, and the pilot must make corrections. One possible malfunction is the loss of communication with GUIDO and FIDO for several seconds.

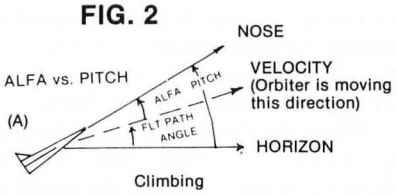
In Ace mode, both the autopilot and ground assistance are unavailable. Also, control failures and turbulence will occur.

Flight Techniques

There are three major phases to a landing: Entry (above Mach 3), Energy Management (below Mach 3), and Final Approach (when the Orbiter is nearly lined up with the runway at less than Mach 1). See Figure 4 on next page.

During Entry, the Orbiter follows a slow pitch-down profile with ALFA decreasing 3 degrees for each Mach number, beginning with ALFA = 30 at Mach 8 and above. For example, at Mach 7, ALFA is

FIG. 2



PILOT'S CHECKLIST

MACH	ALFA	DME (x) BNG = 0
>8	30	>256
7	27	196
6	24	144
5	21	100
4	18	64

EXCEPT IF DME<x, BNG = -30

3 to 1 15 to 9 IF ALT>2DME + 10 BNG = -90
<1 9 to 3 ALT = 2 DME 200<1 AS<300
FLARE: RATE = 10%ALT TD RATE<10

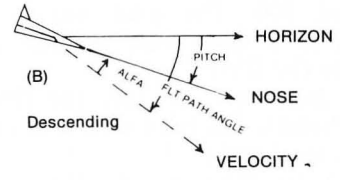


FIG. 3

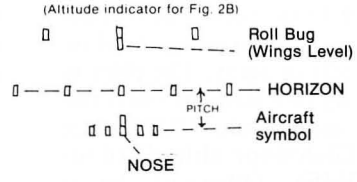
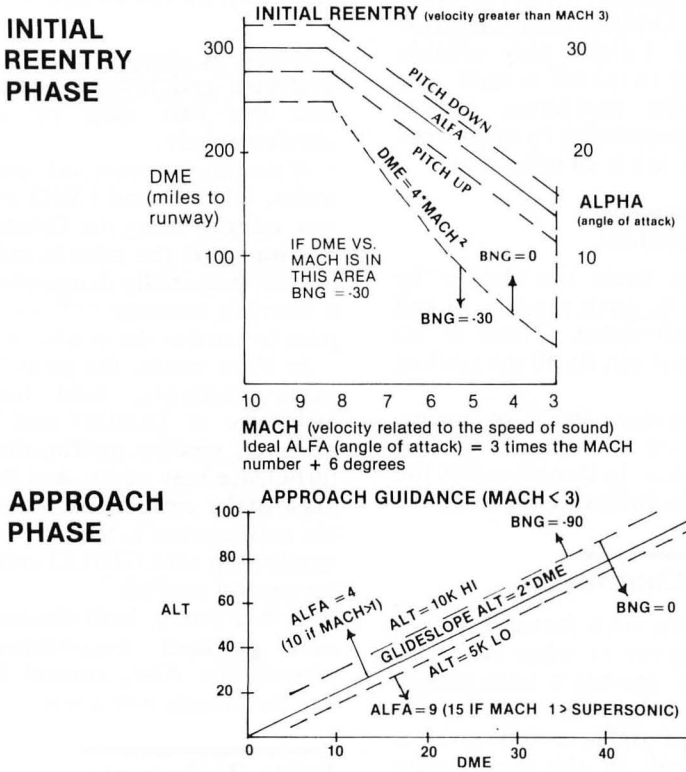


FIG. 4



Explanation: Ideal approach path in approach phase is to keep the altitude in thousands of feet equal to twice the distance to the runway in miles. If you are too high, reduce your angle of attack. If you are too low, increase angle (ALFA). Notice that you need a greater ALFA at supersonic speeds.

Abbreviations:

I.A.S.	Indicated air speed (air pressure, not velocity)
MACH	Velocity (MACH 1 is the speed of sound)
ALFA	Angle between direction shuttle is really traveling and the direction it is pointed
ALT	Altitude in thousands of feet
RATE	Change in altitude in feet per second
DME	Distance measuring equipment (miles to runway)
BNG	Bearing to runway (-left +right of shuttle)
FIDO	Flight dynamics officer
GUIDO	Guidance officer

27; at Mach 6, ALFA is 24. The roll attitude is typically either wings-level or rolled 45 degrees left or right. GUIDO directs the pilot to turn away from the airfield to a BNG of -30 if the Orbiter is approaching too fast. Otherwise, GUIDO will direct the pilot to fly to a BNG of zero (straight to the runway).

During Energy Management, FIDO directs the pilot to achieve an altitude of at least double the DME. For instance, if DME is 20, then ALT ought to be at least 40. The ALFA commands are limited to between 9 and 15 degrees. The roll attitude is usually either wings-level or 45 degrees left or right. The pilot is directed to fly to a BNG of -90 if the altitude is more than 10,000 feet above the Glideslope altitude (double the DME). Otherwise he is directed to turn straight toward the airfield and to maintain the Glideslope with ALT equal to double the DME.

When the Orbiter slows to below Mach 1, there is a large pitch down command, because the aerodynamics change below the speed of sound. At these speeds, ALFA commands are limited to between 3 and 10 degrees until the final flare.

When the Orbiter is approximately lined up with the runway (BNG = 0) and is below the speed of sound, the Final Approach begins. ALFA will drop to around 4 degrees, and IAS will increase to around 300. The pilot ought to maintain a Glideslope altitude of double the DME.

At 2000 feet altitude, the final flare begins. The pilot should roll to wings-level and pitch up (sometimes down) to maintain a sink rate (RATE) of approximately 10% of the altitude. For example, at 200 feet ALT, the RATE should be -20. The landing gear cannot safely handle a touchdown unless the sink rate is smaller than 10. During the final flare, the flight controls are designed to maintain a constant sink rate until new commands are given. Flaring too much may slow the Orbiter to the point that it stalls high above the runway and plummets to the ground.

GUIDO and FIDO will assist the pilot throughout these maneuvers except in Ace mode. If the pilot loses control, he can engage the autopilot (except in Ace mode), and then take over again as soon as the situation improves. The autopilot cannot recover from all possible emergencies.

Scoring

At touchdown, an evaluation of the flight is made. A successful landing must occur (1) on the runway, which is 1 mile wide and 5 miles long; (2) at a sink rate of less than 10 feet per second; (3) with the wings approximately level, so that they will not be torn off; and (4) at an airspeed less than 220 knots, to prevent tire damage. If any of these conditions is violated, then the Orbiter has crashed.

The pilot's score is calculated by starting with 100 points, then adding or subtracting according to the table below:

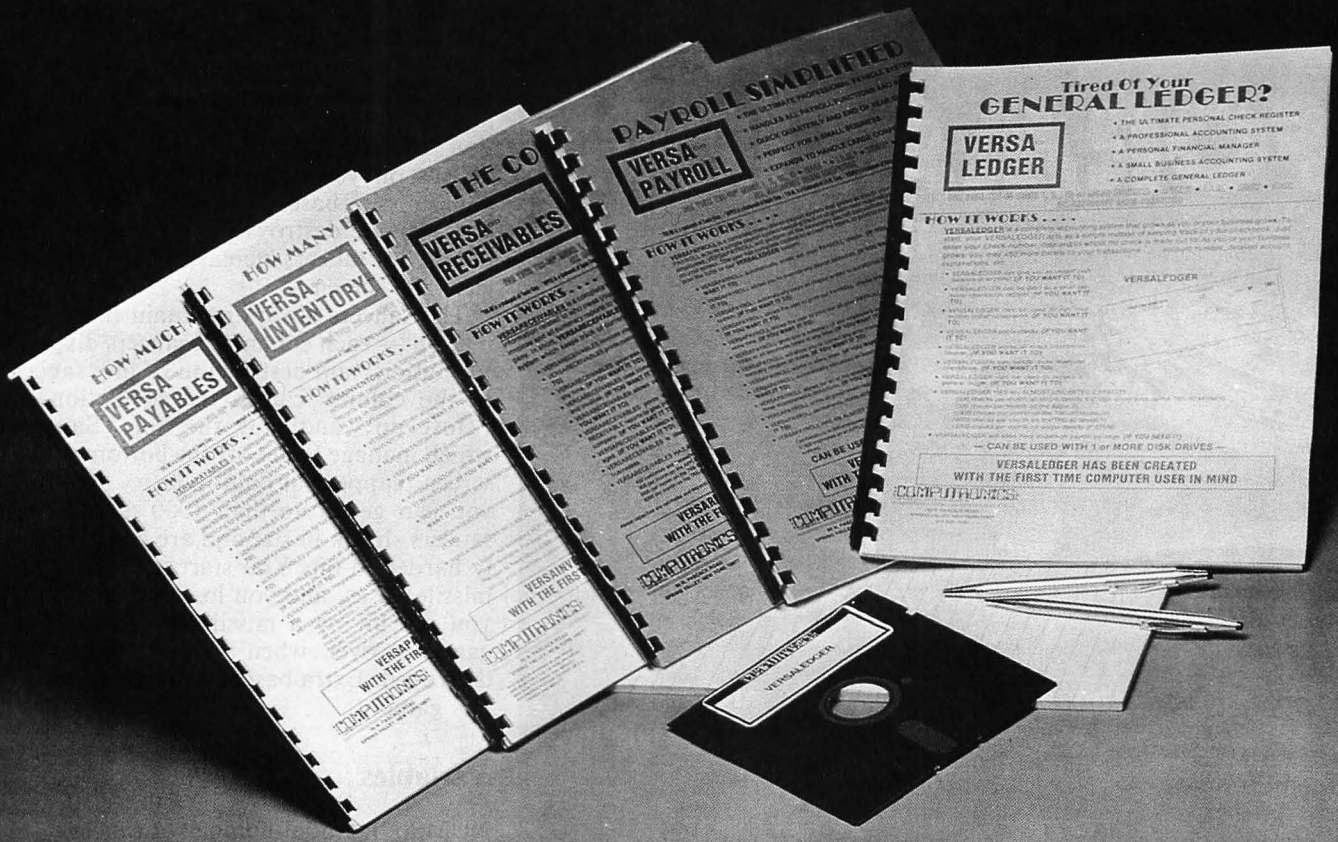
+10 for flying in Rookie mode
+20 for flying in Pilot mode
+30 for flying in Ace mode
−20 for starting with the Final Approach
−10 for starting with Energy Management
- 0 for starting with Entry
-50 for violating the crash criteria
- 1 for each call from GUIDO or FIDO
- 1 for each use of the autopilot

In Ace mode, scoring is identical, but the pilot will be unable to hear GUIDO and FIDO's messages.

If the pilot ignores GUIDO and FIDO, and flies directly over the runway at high altitude, he will crash. There is but one opportunity to land the Orbiter. If the airspeed is too high, the Orbiter will burn up, and the screen will say "BAIL OUT".

Introducing the Most Powerful Business Software Ever!

TRS-80™ (Model I, II, III, or 16) • APPLE™ • IBM™ • OSBORNE™ • CP/M™ • XEROX™



The VERSABUSINESS™ Series

Each **VERSABUSINESS** module can be purchased and used independently, or can be linked in any combination to form a complete, coordinated business system.

VERSARECEIVABLES™ $99.95

VERSARECEIVABLES™ is a complete menu-driven accounts receivable, invoicing, and monthly statement-generating system. It keeps track of all information related to who owes you or your company money, and can provide automatic billing for past due accounts. VERSARECEIVABLES™ prints all necessary statements, invoices, and summary reports and can be linked with VERSALEDGER II™ and VERSAINVENTORY™.

VERSAPAYABLES™ $99.95

VERSAPAYABLES™ is designed to keep track of current and aged payables, keeping you in touch with all information regarding how much money your company owes, and to whom. VERSAPAYABLES™ maintains a complete record on each vendor, prints checks, check registers, vouchers, transaction reports, aged payables reports, vendor reports, and more. With VERSAPAYABLES™, you can even let your computer automatically select which vouchers are to be paid.

VERSAPAYROLL™ $99.95

VERSAPAYROLL™ is a powerful and sophisticated, but easy to use payroll system that keeps track of all government-required payroll information. Complete employee records are maintained, and all necessary payroll calculations are performed automatically, with totals displayed on screen for operator approval. A payroll can be run totally, automatically, or the operator can intervene to prevent a check from being printed, or to alter information on it. If desired, totals may be posted to the VERSALEDGER II™ system.

VERSAINVENTORY™ $99.95

VERSAINVENTORY™ is a complete inventory control system that gives you instant access to data on any item. VERSAINVENTORY™ keeps track of all information related to what items are in stock, out of stock, on backorder, etc., stores sales and pricing data, alerts you when an item falls below a preset reorder point, and allows you to enter and print invoices directly or to link with the VERSARECEIVABLES™ system. VERSAINVENTORY™ prints all needed inventory listings, reports of items below reorder point, inventory value reports, period and year-to-date sales reports, price lists, inventory checklists, etc.

VERSALEDGER II™ $149.95

VERSALEDGER II™ is a complete accounting system that grows as your business grows. VERSALEDGER II™ can be used as a simple personal checkbook register, expanded to a small business bookkeeping system or developed into a large corporate general ledger system **without any additional software.**

- VERSALEDGER II™ gives you almost unlimited storage capacity (300 to 10,000 entries per month, depending on the system),
- stores all check and general ledger information forever,
- prints tractor-feed checks,
- handles multiple checkbooks and general ledgers,
- prints 17 customized accounting reports including check registers, balance sheets, income statements, transaction reports, account listings, etc.

VERSALEDGER II™ comes with a professionally-written 160 page manual designed for first-time users. The VERSALEDGER II™ manual will help you become quickly familiar with VERSALEDGER II™, using complete sample data files supplied on diskette and more than 50 pages of sample printouts.

SATISFACTION GUARANTEED!

Every **VERSABUSINESS**™ module is guaranteed to outperform all other competitive systems, and at a fraction of their cost. If you are not satisfied with any **VERSABUSINESS**™ module, you may return it within 30 days for a refund. Manuals for any **VERSABUSINESS**™ module may be purchased for $25 each, credited toward a later purchase of that module.

To Order:

Write or call Toll-free (800) 431-2818
(N.Y.S. residents call 914-425-1535)

* add $3 for shipping in UPS areas
* add $4 for C.O.D. or non-UPS areas
* add $5 to CANADA or MEXICO
* add proper postage elsewhere

VISA AMERICAN EXPRESS MasterCard

DEALER INQUIRIES WELCOME

All prices and specifications subject to change / Delivery subject to availability.

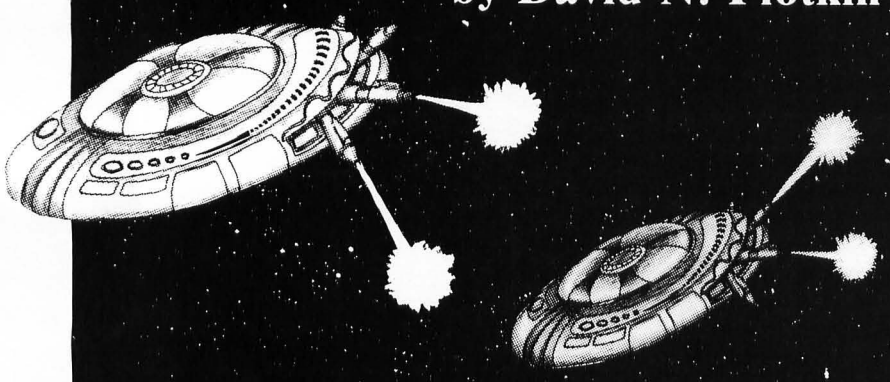
COMPUTRONICS INC.

50 N. PASCACK ROAD, SPRING VALLEY, N.Y. 10977

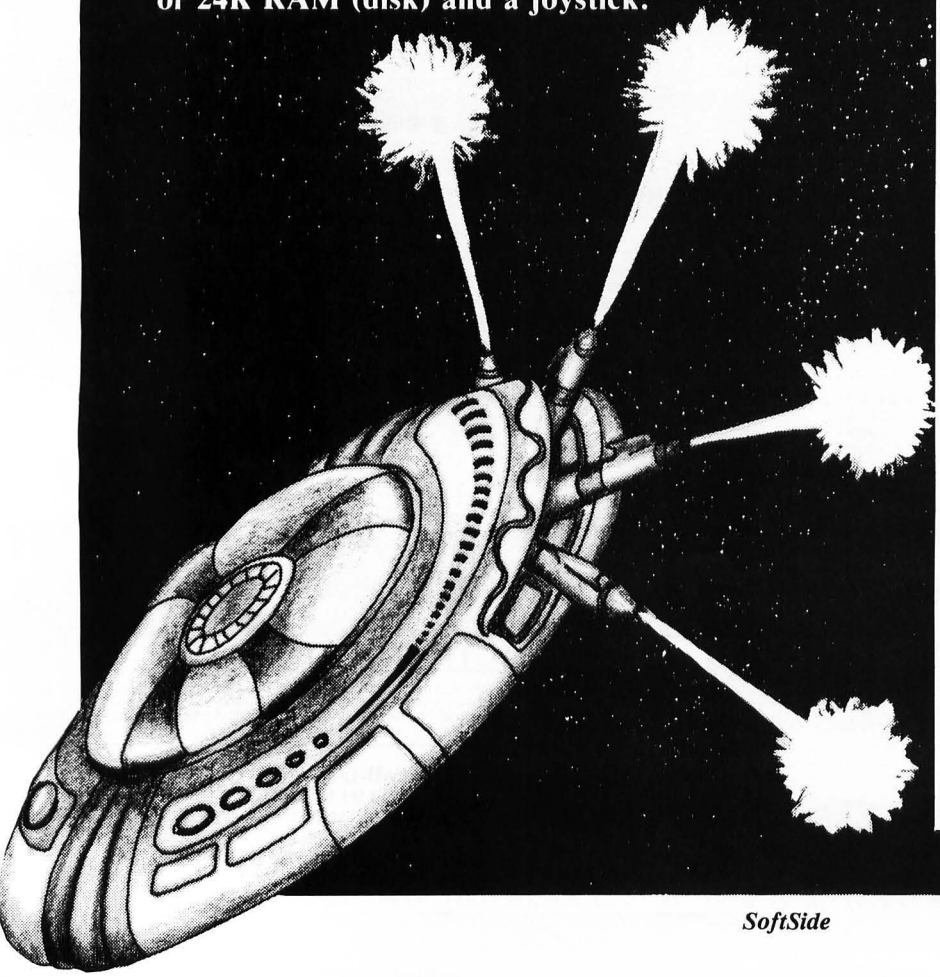
* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp. - *APPLE is a trademark of Apple Corp. - *IBM is a trademark of IBM Corp. - *OSBORNE is a trademark of Osborne Corp. *CP/M is a trademark of Digital Research - *XEROX is a trademark of Xerox Corp.

SAUCER FORMATION

by David N. Plotkin



Saucer Formation **is a real time arcade-style game for an Atari® 400/800 with 16K RAM (tape) or 24K RAM (disk) and a joystick.**

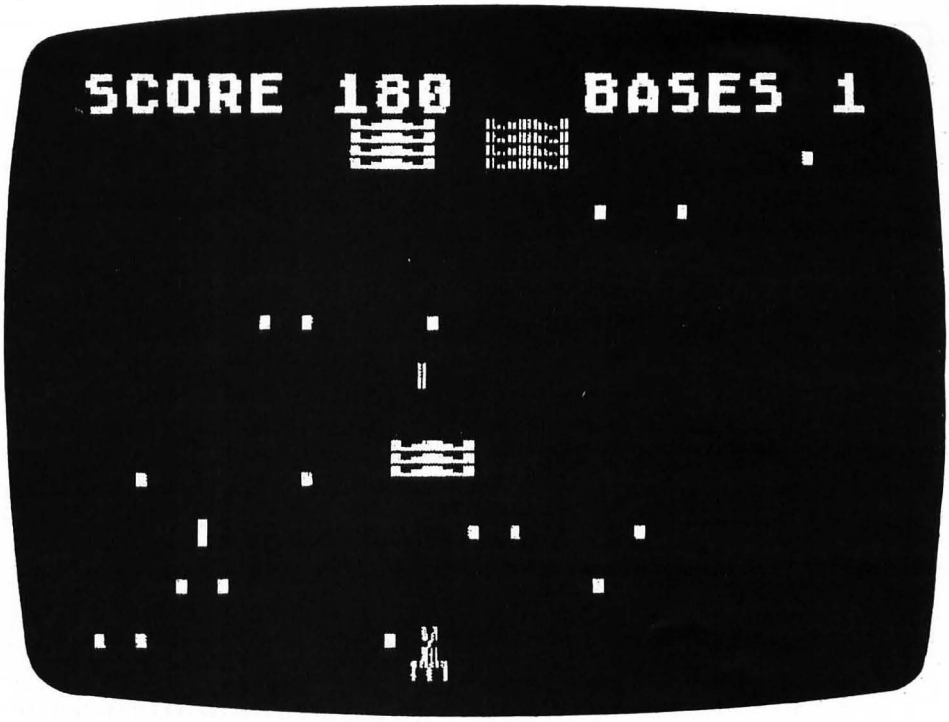


Your missile base moves back and forth across the bottom of the screen, controlled by the joystick. You fire your missile upward by pressing the firing button. Pressing the firing button always launches a new missile; if one has already been fired and is still on the screen, it is aborted (erased). Overhead is the formation of attacking saucers. They drop bombs on you and periodically dive on your missile base. Hitting a saucer with your missile destroys the saucer, but the number of saucers increases each time you clear the screen.

The saucers always maintain their formation — each column of saucers dives in formation, and destruction of any saucer in the column causes the formation to "close up" and plug the hole. Saucers which manage to get off the bottom of the screen reappear at the top to try again. If you can get more than 1000 points, the saucers shrink in size and are consequently harder to hit. You start with three (3) missile bases, and you lose one each time you are hit by a missile or saucer. The game is over when your last base is destroyed. Extra bases are not awarded in this game.

Variables

NUMS(N): Initial number of saucers.
NUMT(N): Current number of saucers in each column.
ST: Value of the joystick.
SC: Score.
NS: Number of missile bases left.
XO, YO: X and Y coordinates of missile base.
X(N), Y(N): X and Y coordinates of saucers.
MISX, MISY: X and Y coordinates of your missile.
MISX(N), MISY(N): X and Y coordinates of bombs.
MIS: =0 if your missile not launched; =1 if your missile launched.
MIS(N): Same as MIS except for bombs.
COL: Collision variable of your missile.
P: -1 if saucers moving left, 1 if saucers moving right.
FL(N): Diving flag; 1 when saucers diving.
PB: Address of PM Graphics in bytes.
XT(N): X coordinate of diving saucer.
A: Address of PM graphics in pages.
JW, JZ, W, N: Loop variables.




```
SS SS SS SS SS SS SS SS SS SS SS
SS                             SS
SS        ATARI  BASIC         SS
SS     'Saucer Formation'      SS
SS   Author:  David Plotkin    SS
SS     Copyright (c) 1982      SS
SS SoftSide Publications, Inc  SS
SS                             SS
SS SS SS SS SS SS SS SS SS SS SS
```

```
1 GOTO 10
2 N=((COL=2) OR (COL=6) OR (COL=10))+2
*((COL=4) OR (COL=12))+3*(COL=8):MIS=0
:D=USR(1600,PB+768+MISY,PB+356)
3 POKE 53278,1:NUMT(N)=NUMT(N)-1
4 D=USR(1536,PB+1024+N*256+Y(N),PB+356
-4*NUMT(N)):FOR JW=15 TO 0 STEP -1:FOR
 JZ=0 TO 9:SOUND 0,30+JZ,10,JW
5 NEXT JZ:NEXT JW:FOR JW=1 TO 5:NEXT J
W:SOUND 0,0,0,0:SC=SC+10:POSITION 6,0:
? #6;SC:SOUND 1,256-SC/10,10,4
6 SOUND 2,246-SC/10,10,4:IF NUMT(1)=0
AND NUMT(2)=0 AND NUMT(3)=0 THEN POP :
GOTO 700
7 RETURN
10 GOSUB 950
```

Read joystick and move missile base.

```
100 ST=STICK(0):X0=X0+4*(ST=7)-4*(ST=1
1):IF X0<48 THEN X0=48
110 IF X0>200 THEN X0=200
120 POKE 53248,X0:IF STRIG(0)=1 THEN G
OTO 140
```

Move your missile or launch a new one. Erase old missile if off the screen.

```
130 D=USR(1600,PB+768+MISY,PB+356):MIS
X=X0+3:MISY=Y0-8:POKE 53252,MISX:MIS=1
:GOTO 160
140 IF MIS=0 THEN GOTO 190
150 D=USR(1600,PB+768+MISY,PB+356):MIS
Y=MISY-8:IF MISY<40 THEN MIS=0:GOTO 19
0
```

Jump to routine to test for hit of your missile on a saucer.

```
160 D=USR(1600,PB+768+MISY,PB+488):COL
=PEEK(53256):IF COL=0 THEN GOTO 190
170 GOSUB 2
```

Calculate new position of saucers which are not diving.

```
190 P=P+2*(X(1)<48)-2*(X(3)>200):FOR N
=1 TO 3
200 X(N)=X(N)+4*(P=1)-4*(P=-1):IF NUMT
(N)=0 THEN GOTO 280
210 IF FL(N)=1 THEN GOTO 240
```

Move saucers which are not diving and test to see if they should dive.

```
220 POKE 53248+N,X(N):XT(N)=X(N):R=INT
(RND(0)*10):IF R=0 THEN FL(N)=1
230 GOTO 280
```

Erase diving saucer at last position and update Y coordinate. Test to see if saucer is off the bottom of the screen.

```
240 D=USR(1536,PB+1024+N*256+Y(N),PB+3
56):Y(N)=Y(N)+8:IF Y(N)<208 THEN GOTO
260
```

Redraw saucer at the top of the screen.

```
250 Y(N)=48:XT(N)=X(N):FL(N)=0:POKE 53
248+N,X(N):D=USR(1536,PB+1024+N*256+Y(
N),PB+356-4*NUMT(N)):GOTO 280
```

Update X coordinate and redraw saucer.

```
260 IF Y(N)+4*NUMT(N)<208 THEN XT(N)=X
T(N)+2*(X0>XT(N)+3)-2*(X0<XT(N)+3):POK
E 53248+N,XT(N)
```

```
270 D=USR(1536,PB+1024+N*256+Y(N),PB+3
56-NUMT(N)*4)
```

Redraw your missile. Necessary because erasing and redrawing the other missiles (bombs) may erase your missile.

```
280 NEXT N:IF MIS=1 THEN D=USR(1600,PB
+768+MISY,PB+488)
```

Test for hit of your missile on a saucer.

```
285 COL=PEEK(53256):IF COL<>0 THEN GOS
UB 2
```

Test for hit on missile base by a saucer.

```
290 IF PEEK(53260)<>0 THEN NS=NS-1:GOT
O 400
```

Launch new bombs from saucers, advance existing bombs, test for hit on missile base, draw and erase bombs.

```
300 FOR N=1 TO 3:IF NUMT(N)=0 OR MIS(N
)=1 OR (Y(N)+NUMT(N)*4>200) THEN GOTO
330
310 R=INT(RND(0)*10):IF R<>0 THEN GOTO
360
```

```
320 MISX(N)=XT(N)+3:MISY(N)=Y(N)+NUMT(
N)*4+8:MIS(N)=1:POKE 53252+N,XT(N)+3:D
=USR(1600,PB+768+MISY(N),PB+488+N*8)
330 IF MIS(N)=0 THEN GOTO 360
340 D=USR(1600,PB+768+MISY(N),PB+356):
MISY(N)=MISY(N)+12:IF MISY(N)>208 THEN
 MIS(N)=0:GOTO 360
350 D=USR(1600,PB+768+MISY(N),PB+488+N
*8):IF PEEK(53256+N)=1 THEN NS=NS-1:GO
TO 400
```

Sounds and explosions for hit on missile base, reinitialize variables.

```
360 NEXT N
370 POKE 53278,1:GOTO 100
400 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND
2,0,0,0:FOR N=10 TO 150:POKE 704,N:SOU
ND 0,N,8,8:NEXT N
410 D=USR(1570,PB+1024+Y0,PB+356):SOUN
D 0,0,0,0:D=USR(1600,PB+768+MISY,PB+35
6):MIS=0
420 FOR N=1 TO 3:D=USR(1536,PB+1024+N*
256+Y(N),PB+356):D=USR(1600,PB+768+MIS
Y(N),PB+356):MIS(N)=0:NEXT N
430 FOR N=1 TO 100:NEXT N:POKE 53278,1
440 IF NS=0 THEN GOTO 500
450 X0=120:X(1)=48:X(2)=74:X(3)=100:Y0
=200:POKE 704,20:D=USR(1570,PB+1024+Y0
,PB+472):POKE 53248,X0
460 FOR N=1 TO 3:Y(N)=48:POKE 53248+N,
X(N):XT(N)=X(N):FL(N)=0
470 D=USR(1536,PB+1024+N*256+Y(N),PB+3
56-NUMT(N)*4)
480 NEXT N:POKE 18,0:PRINT #6;NS
490 GOTO 100
```

End of game routine.

```
500 ? #6;CHR$(125):POSITION 6,4:? #6;"
GAME OVER":? #6;" FINAL SCORE  ";SC
510 POSITION 6,6:? #6;"POINTS":? #6;"
  TO PLAY AGAIN":? #6;"  PRESS fire BU
TTON"
520 FOR W=150 TO 50 STEP -1:IF STRIG(0
)=0 THEN GOTO 550
530 SOUND 0,W,10,6:SOUND 1,W+10,10,6:S
OUND 2,W+50,10,6:SOUND 3,W+100,10,6:NE
XT W
540 GOTO 520
550 FOR N=0 TO 3:SOUND N,0,0,0:NEXT N:
GOSUB 1140:GOTO 100
```

Add one to each column of saucers when the screen is cleared. Redraw saucers at the top of the screen. If score is greater than 1000, make saucers smaller.

```
700 FOR N=1 TO 3:D=USR(1536,PB+1024+N*
256+Y(N),PB+356):D=USR(1600,PB+768+MIS
Y(N),PB+356):MIS(N)=0:NEXT N
710 X(1)=48:X(2)=74:X(3)=100:FOR N=1 T
O 3:NUMS(N)=NUMS(N)+1:IF NUMS(N)>16 TH
EN NUMS(N)=16
720 NUMT(N)=NUMS(N):Y(N)=48:POKE 53248
+N,X(N):XT(N)=X(N):FL(N)=0:NEXT N
730 IF SC>1000 THEN POKE 53257,0:POKE
53258,0:POKE 53259,0
740 FOR N=1 TO 3:D=USR(1536,PB+1024+N*
256+Y(N),PB+356-NUMT(N)*4):NEXT N
750 GOTO 100
800 END
```

Title page and sounds.

```
950 GRAPHICS 2+16:A=PEEK(106)-16:POKE
54279,A:PB=256*A:POKE 559,62:POKE 5327
7,3:POKE 623,1
960 POSITION 7,4:? #6;"SAUCER":POSITIO
N 5,5:? #6;"formation"
970 FOR W=50 TO 150:SOUND 0,W,10,6:SOU
ND 1,W+10,10,6:SOUND 2,W+50,10,6:SOUND
 3,W+100,10,6:NEXT W
980 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND
2,0,0,0:SOUND 3,0,0,0
```

Read data into memory, initialize PM graphics, initialize variables.

```
1000 FOR A=1536 TO 1560:READ I:POKE A,
I:NEXT A:DIM NUMS(3),NUMT(3),FL(3),MIS
(3),X(3),Y(3),XT(3),YT(3)
1005 DIM MISX(3),MISY(3)
1010 DATA 104,104,133,204,104,133,203,
104,133,207,104,133,206,160,0,177,206,
145,203,200,192,64,208,247,96
1020 FOR N=0 TO 23:RESTORE 1030:FOR A=
PB+260+N*4 TO PB+263+N*4:READ I:POKE A
,I:NEXT A:NEXT N
1030 DATA 153,189,255,0
1040 FOR A=PB+356 TO PB+471:POKE A,0:N
EXT A
1050 FOR A=PB+472 TO PB+487:READ I:POK
E A,I:NEXT A
1060 DATA 40,40,56,56,16,16,56,56,56,5
6,254,254,146,146,146,146
1070 FOR A=1570 TO 1594:READ I:POKE A,
I:NEXT A
1080 DATA 104,104,133,204,104,133,203,
104,133,207,104,133,206,160,0,177,206,
145,203,200,192,16,208,247,96
1090 FOR A=1600 TO 1624:READ I:POKE A,
I:NEXT A
1100 DATA 104,104,133,204,104,133,203,
104,133,207,104,133,206,160,0,177,206,
145,203,200,192,8,208,247,96
1110 FOR A=PB+488 TO PB+519:READ I:POK
E A,I:NEXT A
1120 DATA 3,3,3,3,3,3,3,3,12,12,12,12,
12,12,12,12,48,48,48,48,48,48,48,48,19
2,192,192,192,192,192,192,192
1130 FOR I=PB+768 TO PB+2048:POKE I,0:
NEXT I
1140 X0=120:X(1)=48:X(2)=74:X(3)=100:P
OKE 53248,X0:POKE 53249,X(1)
1150 POKE 53250,X(2):POKE 53251,X(3):Y
0=200:Y(1)=48:Y(2)=48:Y(3)=48
1160 POKE 53257,1:POKE 53258,1:POKE 53
259,1:POKE 704,20:POKE 705,214:POKE 70
6,100:POKE 707,166
1170 D=USR(1570,PB+1024+Y0,PB+472):D=U
SR(1536,PB+1280+Y(1),PB+352)
1180 D=USR(1536,PB+1536+Y(2),PB+352):D
=USR(1536,PB+1792+Y(3),PB+352)
1190 FOR N=1 TO 3:NUMS(N)=1:NUMT(N)=1:
FL(N)=0:MIS(N)=0:NEXT N:MIS=0:P=1:SC=0
:NS=3:MISX=0:MISY=0
```

Clear screen, print stars, score and numbers of bases left.

```
1200 PRINT #6;CHR$(125):FOR N=1 TO 20:
POSITION INT(RND(0)*19),INT(RND(0)*11)
:PRINT #6;".":NEXT N
1210 POSITION 0,0:? #6;"score ";SC:POS
ITION 12,0:? #6;"bases ";NS:RETURN
```

ATARI® SWAT TABLE FOR: SAUCER FORMATION

LINES	SWAT CODE	LENGTH
1 - 5	MV	505
6 - 150	GR	514
160 - 250	OK	617
260 - 320	NH	606
330 - 410	CI	561
420 - 470	QB	507
480 - 550	VE	545
700 - 740	PF	505
750 - 980	DK	527
1000 - 1060	WG	512
1070 - 1140	DU	571
1150 - 1190	KC	589
1200 - 1210	RP	196

The Adventure is Waiting for You...

How would you like to go back in time to 19th century London to match wits with Jack the Ripper? Out into space to brave the swirling vortex of a black hole? Into the depths of the ocean, or on a quest to rescue a beautiful princess from the clutches of evil monsters?

You never know where **SoftSide Magazine's Adventure of the Month Club** might take you. But you can be sure that each month you will experience new delights and new challenges as you receive an original adventure on tape or disk, ready to load into your computer.

And now it's even easier for you to join **Adventure of the Month.** A Trial Membership (3 months, 3 different Adventures), costs only $29 for Disks, $19 for Cassettes.

Or choose a Charter Membership, available on disk or tape, for 6 months, 12 months, or 24 months. Your choice of a 24 month Charter Membership will bring you a savings of almost half on the individual price of the Adventures you'll receive.

Super Disks, each containing three Adventures, are also available for $26 each.

Adventure #18 — The Dalton Gang

This adventure takes place in the old west in the late 1800's, a west which is terrorized by the notorious outlaw gang, the Daltons. There were 4 of them: Emmett, Bob, Bill, and Grat. Your job, as sheriff, is to capture, jail, kill these boys. There are also other things you'll have to do in order to gain a perfect score, and above all, don't let the Daltons escape from jail.

Adventures Available:

June 1981
Arabian Adventure

July 1981
Alien Adventure

August 1981
Treasure Island

September 1981
Jack The Ripper

October 1981
Crime Adventure

November 1981
Around the World in Eighty Days

December 1981
Black Hole

January 1982
Windsloe Mansion

February 1982
Klondike Adventure

March 1982
James Brand

April 1982
Witches' Brew

May 1982
Titanic

June #13
Arrow One

July #14
Robin Hood

August #15
The Mouse That Ate Chicago

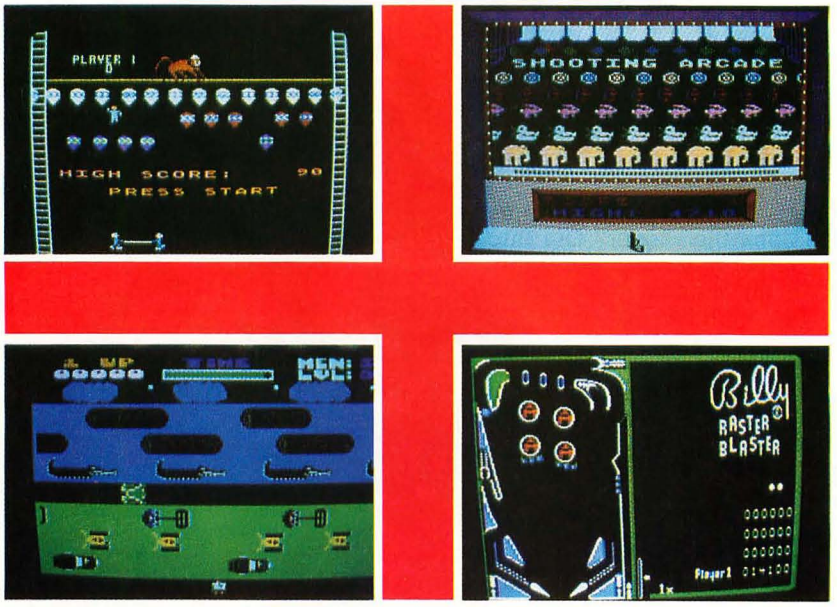
September #16
Menagerie

October #17
The Deadly Game

For more detailed information on prices, and to enter your membership, see the handy postage-free bind-in card found facing page 88 of this issue.

*For reviews of other products available for the ATARI® see page 112.

AN ATARI® CHRISTMAS



Reviewed by David Plotkin

Clowns And Balloons **by Frank Cohen (DATASOFT, 9421 Winnetka Dr., Chatsworth, CA 91311). System requirements: Atari 400/800, 16K RAM, cassette or disk. Suggested retail price: $29.95. Display shown top left above.**

Shooting Arcade **by Mark Riley (DATASOFT, 9421 Winnetka Dr., Chatsworth, CA 91311). System requirements: Atari 400/800, 16K RAM, cassette or disk. Suggested retail price: $29.95. Display shown top right above.**

Preppie **by Russ Wetmore (ADVENTURE INTERNATIONAL, P.O. Box 3435, Longwood, FL 32750). System requirements: Atari 400/800, 16K RAM tape, 32K RAM disk. Suggested retail price: $29.95. Display shown lower left above.**

Raster Blaster **by Bill Budge (BUDGECO, 482 Pala Avenue, Piedmont, CA 94611). System requirements: Atari 400/800, 32K RAM and disk drive. Suggested retail price: $29.95. Also available for the Apple II™ with 48K RAM and disk drive. Suggested retail price: $29.95. Display shown lower right above.**

This has been a good year for sales of the Atari 400/800 home computers, and what better present for the computer enthusiast than some game software. There is now (finally) a huge assortment of games available for the Atari, and the confusion which has faced Apple owners and their friends in past years can now be yours! With this review and those that follow, we hope to guide you to the very best of recently released software for the Atari.

Clowns And Balloons

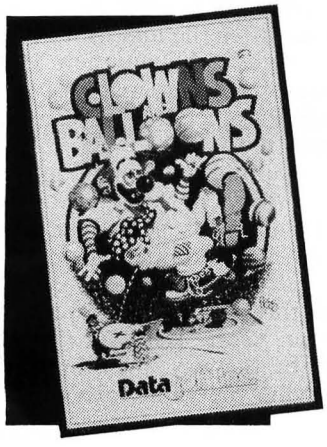
Clowns And Balloons presents the player with the following dilemma: After a performance under the Big Top, layers of balloons are left circling near the ceiling of the circus tent. The player's job is to pop all the balloons and clear the tent for the next performance. To accomplish this, a clown climbs up the ladder located at the side of the tent and leaps from the ladder. The player must control two additional clowns who dash back and forth across the floor of the tent, carrying a trampoline between them. Successfully positioning the trampoline under the clown causes him to rebound back into the air, flying higher and faster with each rebound, and popping all the balloons he encounters. The balloons circulate across the top of the tent, which makes them harder to hit. Each row of balloons must be completely popped before the next row is popped or the monkey, who sits on the high wire overlooking the

clean-up operation, will replace the whole row of balloons popped "out of order."

Playing *Clowns And Balloons* is easy. *Mastering* it requires quick reflexes and considerable skill. The flying clown can be aimed by having him hit a particular spot on the trampoline (much like the ball and paddle in *Breakout*). When the clown hits the ladders located at each side of the screen (*ouch*!), he also rebounds from them, which can cause some pretty frantic scampering by his trampoline-bearing friends to keep him from landing messily on his head. You get three clowns to begin, and an extra clown at 10,000 points. Each time you clear all the balloons in the tent, the clown does a little dance with the monkey on the high wire that is a delight to behold. The balloons then reappear, and everything starts over — faster this time. The player has three options to control the trampoline clowns — joystick, paddle, or trac-ball. The joystick is not going to get you very high scores, as you just can't move fast enough from one position to another. The paddle, however, works very well, since the trampoline can move from one side of the tent to the other at the flick of a wrist. I didn't try the trac-ball option, but Frank Cohen assures me that the game was designed with the trac-ball in mind, so it should work well.

The programmers at DataSoft understand the Atari's features very well. Multiple Display List Interrupts give lots and lots of high resolution color. The balloons have vivid color patterns on them. (It seems a shame to pop them!) The flying clown is done in three colors, created by overlaying two Players. The music is superb. There is an introductory theme, an ending theme, and different tunes for popping each row of balloons and for the bonus clown. Each tune is done in all four voices, with harmony and bass. Composing music on the Atari is not easy, but Mr. Cohen has done an excellent job. (By the time you read this, DataSoft should have released a program for composing music that will allow you to make music and put it into your own programs.)

Clowns And Balloons can be played at four skill levels, but even Level 1 is still a challenge after three months. One word of warning — the arms of the clowns carrying the trampoline can easily be mistaken for the edges of the trampoline because of color similarity. You have to be careful not to catch the clown *too* close to the edge of the trampoline or he'll land on his head. The color of the arms should probably be changed in a future version.



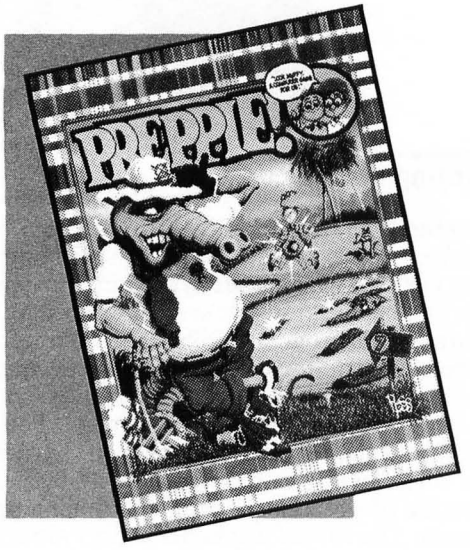
Shooting Arcade

Another DataSoft entry worthy of note is *Shooting Arcade* by Mark Riley. It is perhaps misnamed, because this is not really an arcade type game. Instead, it is the finest implementation I've seen of a shooting gallery for the Atari. This program exhibits the DataSoft hallmarks of fine color and music, as well as excellent animation. The player controls a pistol which moves along the bottom of the screen under joystick control. The balance of the screen is the shooting gallery itself, bordered by curtains, and filled with rows of moving targets. It is the animation of the targets that is the shining point of this program. The row of elephants lumber along; the ducks quack and move their wings; the rabbits hop along; and the smiling faces rotate to frowning faces and back again. As in a real shooting gallery, the directions in which the rows of targets move alternate from row to row. The targets move off one edge of the screen and reappear from the other edge.

Shooting Arcade is unusual in that 16-year-old reflexes are not required to obtain high scores. Instead, a careful aim and a good sense of timing are rewarded. There is no penalty for taking your time on each shot; in fact, this game encourages it, since you get only a limited number of bullets and shouldn't waste them. The bullets take an appreciable time to travel to the back rows of targets, so you have to "lead" those targets more, making them harder to hit. The back three rows of targets add more complexity to the game. The smiling/frowning face targets will disappear and score points regardless of which face is turned toward the player. However, if you hit the face while it is frowning, another rabbit appears in the rabbit row, and you have to use a bullet on the extra rabbit. Careful timing is necessary to hit the face while it is smiling. The next row back is made up of bull's-eye type targets. Each time you hit one of them, *all* the rows reverse direction, forcing you to rethink your timing. To obtain more bullets, you have to hit one of the back rows of diamonds. Thus, if you're getting low on bullets, you have to try to hit the back row *through* the intervening rows of targets, a difficult task.

Once you've cleared all the targets from the shooting gallery, a large brown bear marches out. Each time you hit the bear, it reverses direction and speeds up, and you are awarded 50 points. If you are unable to hit the bear, it moves off the screen and the next round starts.

Typical of all DataSoft material, *Shooting Arcade* makes good use of the Atari's features. The curtains bordering the arcade are Players, so the characters comprising the targets move smoothly "behind" the curtains. The bullets are Missiles, and you can have four in the air at one time, although, as I mentioned earlier, this quick shooting is *not* necessary to enjoy the game. Multiple Display List Interrupts allow for many colors on the screen at once. This game can be played by two players, taking turns, and keeps track of the high score.



Preppie

Adventure International has long been known for its Scott Adams Adventure series and excellent fast action games for the TRS-80® and Apple™. Now, with the introduction of Russ Wetmore's *Preppie*, AI has leapt wholeheartedly into the Atari market. Based loosely on the popular arcade game, *Frogger*, this colorful, high speed game with good sound effects challenges the player to guide the Preppie character from one edge of a golf course to the other, retrieving golf balls. Hazards are quite varied; there are vehicles, a river to cross, and a giant frog. In addition to all that, the player must beat the clock, which relentlessly ticks away the seconds.

The first question, of course, is why is a Preppie retrieving golf balls? The humorously written manual explains how Wadsworth Overcash lost an annual freshman contest at his university. The details of this contest I will not enumerate. The punishment for losing the contest is retrieving golf balls for one day at "Nasty Nine," the world's most dangerous golf course.

The "Nasty Nine" is chock full of dangers. Preppie starts at the bottom of the screen and must proceed to the top, retrieving golf balls one at a time and returning them to the bottom. You begin the game with three Preppies, and get a bonus Preppie at 8000 points. The first obstacle to cross is the fairway, a strip of grass which runs all the way across the screen. Lawn mowing maintenance men and, at the higher levels, bulldozers and speeding golf carts, are constantly crossing the fairway in both directions. Your Preppie must dodge between the vehicles, taking care not to touch any, or he will be flattened out and lost.

If you successfully negotiate the fairway, you end up at the river bank, a narrow strip of land bordering on the next obstacle — the river. Starting with Level 6, a giant frog hops from left to right across the river bank. Be sure not to let him eat you! The river is wide and rows of moving objects fill it — objects which your Preppie can jump to and from in order to get across the river. Falling in the water loses the Preppie. At the lower levels, these objects consist of slow-moving boats and logs, but the speed increases at higher levels and one row of boats is replaced with alligators. *Look out for the alligators*! They periodically open their mouths — causing the Preppie to lose his balance and fall into the water. The Preppie can move left and right on the boats and logs, but the backs of alligators are slippery, and any attempt to turn sidewise on them dumps Preppie in the water. The alligators' mouths are especially difficult to avoid. The strategy that works best for me is waiting until they've just closed their mouths and then jumping onto their backs — and off again as soon as possible. But don't wait too long — remember the giant frog and the ticking timer. If time runs out, you lose your current Preppie.

Finally, after crossing the river, there is the "rough." It looks much like the fairway, except for three waterholes. (Don't fall in!) There are no moving obstacles here. As on a real golf course, most of the balls end up in the rough. At higher levels, more golf balls appear, necessitating more trips for poor Preppie, who can only carry one ball at a time.

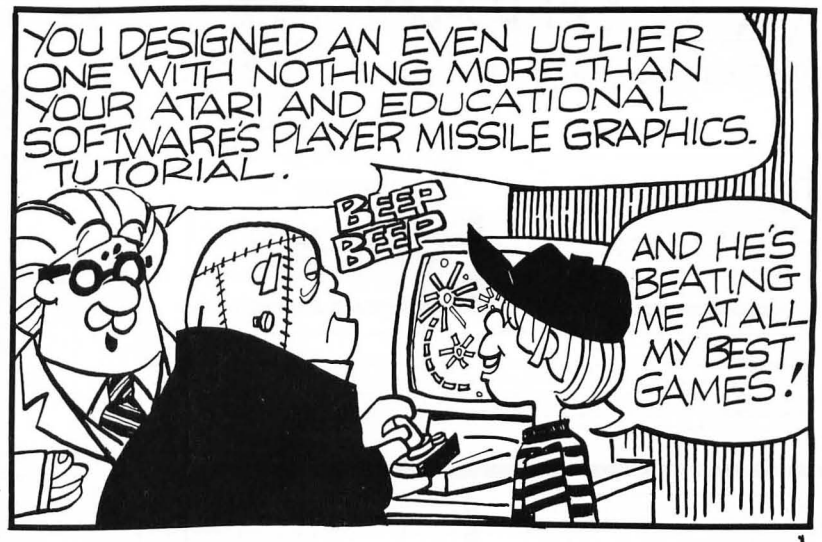
Preppie is played with a joystick. It can be played by one or two players, and two players have the option of using one or two joysticks. Pushing the joystick in one of four directions causes Preppie to turn and move in that direction. Once at the river, Preppie can jump. Push the joystick forward or back and press the fire button. There is a noticeable delay while Preppie flexes his knees (or whatever) and then jumps.

You have to take this delay into account when preparing to jump. Preppie *must* land with both feet squarely on the log, boat, or the alligator, or he loses his balance and **splash**! Scoring is based on two things: 1) successfully crossing an obstacle zone (river, fairway, etc.) and 2) the amount of time left when all the golf balls have been recovered. There are ten skill levels, and the player can choose which level (from 1-9) at which to start. You can only get to level 10, where some surprises await the player, by successfully completing level 9. This ability to skip over the slower, easier levels is an excellent feature. Also very nice is the ability to pause the game in the middle, or to abort and restart a game. The restart sequence after the loss of a Preppie is also very nice — you begin with your new Preppie (provided you've got one left) at the same level you were on when you lost the old Preppie. *Preppie* also sports some snappy four-channel music.

That same snappy four-channel music can get pretty wearing after a while, however, as it plays continuously in the background. Well, never fear. Mr. Scott Adams, president of AI, has permitted me to tell you something that is not in the instructions. Hit SHIFT/CONTROL/M (all three keys together) and *the music turns off*! Thank you, Scott. Tell your friends you read it in ***SoftSide***. Oh, by the way, the other sounds, which are helpful in playing the game, are not bothered by this procedure.

continued on page 94

The Adventures of PROFESSOR VON CHIP & ORBIE



TRICKY TUTORIALS (tm)

There are many things that the ATARI computers can do either better, or easier than other small computers. The following series of programs is designed for anyone who is at least familiar with BASIC programming. What each tutorial offers is similar to an extensive magazine article with all discussion in as simple language as possible, plus you get MANY examples already typed in and running. The instruction manuals range from 10 to 50 pages, and some tutorials fill up a complete tape or disk. There is little overlap in what is taught, so anyone wanting to know all they can should buy them all (my banker thanks you). ATARI buys these from us to use in training their own people! Rave reviews have been published in ANTIC, ANALOG, CREATIVE COMPUTING, and even INFOWORLD. You trust INFOWORLD, don't you?

TT #1: DISPLAY LISTS—This program teaches you how to alter the program in the ATARI that controls the format of the screen. Normally, when you say "Graphics 8", the machine responds with a large Graphics 8 area at the top of the screen and a small text area at the bottom. Now, you will be able to mix various Graphics modes on the screen at the same time. The program does all of the difficult things (like counting scan lines). You will quickly be able to use the subroutines included in your own programs.
16K Tape or 24K Disk. **$19.95**

TT #2: HORIZONTAL/VERTICAL SCROLLING—The information you put on the screen, either GRAPHICS or TEXT, can be moved up, down, sideways, or diagonally. We provide the basic methods and leave the rest up to your skill and imagination. Includes 18 examples to get you started, with several using a small machine language subroutine for smoothness.
16K Tape or 24K Disk. **$19.95**

TT #3: PAGE FLIPPING—Now you don't have to redraw the screen every time you change the picture or text. You will learn how to have the computer draw the next screen you want to see while you are still looking at the previous screen, then flip to it instantly. You won't see it being drawn, so a complicated picture can seem to just appear. Depending on your memory size and which graphics or text modes you are using, you can instantly look at up to 50 pages. The basic method takes only 9 lines and the usefulness is infinite.
16K Tape or 24K Disk. **$19.95**

TT #4: BASICS OF ANIMATION—This program shows you how to animate simple shapes (with some sound) using the PRINT and PLOT commands, and it also has a nice little PLAYER/MISSILE GRAPHICS game you can learn from. The P/M example is explained and will get you started on this complicated subject (more fully explained in TT #5). This would be an excellent way to start making your programs come alive on the screen with movement! Recommended for beginning users.
16K Tape or 24K Disk. **$19.95**

TT #5: PLAYER/MISSILE GRAPHICS—Learn to write your own games and other animated applications! The tutorial begins with many small examples that compliment the 50 page manual, then gradually builds up to a complete game where everything you need to know is fully explained. Also included are two machine language utilities that you can use to animate Players with from BASIC. Next we include two of the best editors currently available; one for editing playfield shapes (backgrounds); and one to edit your players, and all in glorious Technicolor!! Everything except the two editors run in 16K Tape or 32K Disk. **$29.95**

TT #6: SOUND AND MUSIC—Unless you have spent many years experimenting with the four voice channels, you will learn a lot from this one! Learn to play standard notes, chords, and whole songs using some simple "tricks". One of the nicest parts are the examples of special sound effects that you can refer to whenever you need a sound for a program or to impress a friend. This program will be of interest to all ages and levels of experience!
16K Tape or 24K Disk. **$19.95**

SPECIAL DISCOUNT

Order the first six tutorials in a 3-ring binder for $99.95, a $30.00 savings!

TT #7: DOS UTILITIES—We at Educational Software have been shocked by some of the prices others are charging to offer you small utilities to help in the use of your Disk Drive. We now offer you all of the following plus explanation as to how each was written, and how to use them: A UNIQUE MENU PROGRAM, AN AUTORUN.SYS BUILDER, DISK INSPECTOR (LOOK AT SECTORS), DISK JACKET PRINTER, AUTOMATIC FORMATTER, RECORD SAVE AND LOAD UTILITY.
32K Disk Only. **$29.95**

MASTER MEMORY MAP (tm)

This book is the most valuable source of information for your ATARI you can buy. It starts out by explaining how to PEEK and POKE values into memory, so that even new computer owners can use many of these "Tricks". Then you are given 32 pages of the memory locations that are the most useful, along with hints on how to use many of the locations. Finally, it includes hints on problems you may be having with the computer and discusses the new Graphics modes 9 to 11. Even ATARI buys this book from us! **$6.95**

CONTACT YOUR LOCAL DEALER or ORDER BELOW

We have other fine programs for ATARI computers.
Write for a catalog.
Send us your programs to sell too!

USER SUBMITTED PROGRAMS

MINI-DATABASE/DIALER—stores and edits up to 8 lines of information such as names & addresses, phone numbers, messages, inventories, or anything you want. It has the usual sort, search, and print options, but it also has an unusual feature: If your file includes phone numbers and your phone company allows touch-tone phone signals, the program will DIAL THE PHONE NUMBER FOR YOU! 16K Tape or 24K Disk. **$24.95**

THE GRAPHICS MACHINE—allows the ATARI to act like more expensive graphics computers using simple commands like line, box, circle, polygon, fill, and savescreen to get a high resolution picture you can save on disk in only five seconds! Many more features! 48K Disk Only. **$19.95**

BOB'S BUSINESS—14 small business type programs accessed from a common menu. 16K Tape or 32K disk. **$14.95**

MINI WORDPROCESSOR—A simple text editor to write, save, and print several pages at a time. 32K Tape or Disk. **$19.95**

KID'S #1—Includes a MATH QUIZ, a children's TREASURE HUNT, and a DIALOGUE program. 16K Tape or 24K Disk. 3 for . . . **$14.95**

KID'S #2—SPELLING BEE, WORD SCRAMBLE, and TOUCH. 16K Tape or 24K Disk. 3 Educational games for . . . **$14.95**

PLAYER PIANO—Turns your keyboard into a mini-piano and more. 24K Tape or 32K Disk. **$14.95**

DOG DAZE—Two cute little doggies race for the fire hydrants, shoot their bones, and just have a lot of fun! A fast action program for all ages. 8K Tape or 16K Disk, in machine language. **$16.95**

GRAPHIC SYMBOL LABELS—for your keyboard to remind you of the built-in Graphics symbols. 2 complete sets for . . . **$2.95**

OUR NEWEST PROGRAMS

*** OUR BEST GAME ***

SPACE GAMES—Our family is being attacked by ALIENS, and only you can save us. A comic book manual will guide you through three games that test your ability in space skills. Includes ALIENS, SURVIVE, and ROBOT ATTACK, and is for all ages. The first two games require 16K for Tape. The last game and all Disk users need 32K. **$24.95**

MATHS FOR FUN—Another ENGLISH import teaching basic math skills. Very colorful and enjoyable to use. For ages 5 to 16. 16K Tape or 24K Disk. **$19.95**

TT #10: SOUND EFFECTS—From laser blasts to ringing phones, this tutorial will show you how to make unique sound effects in all of your programs! 16K Tape or 24K Disk. **$19.95**

MARATHON—This is a unique math quiz for one or two players. You are in a race to move your runner across the screen first! There are four levels of play with five modes of operation for each. The game uses joysticks for all input, so play is easy for young children. This wonderful learning tool is imported from ENGLAND for your learning pleasure. Your kids will never even notice they are playing an EDUCATIONAL program. 16K Tape or 24K Disk. **$19.95**

TT #8: CHARACTER GRAPHICS—Character Graphics is the best way to animate your ATARI! Make letters look like space monsters, gunfighters, or a myriad of other shapes. Use our editor to create these multicolor shapes and then we'll show you how to move them around the screen. This tutorial even shows how our Space Games were written! 16K Tape or 24K Disk. **$19.95**

To order COD, VISA or MasterCard call — **(408) 476-4901**

By mail include $3.00 postage (USA only) or $1.50 Memory Map only—
California residents add 6.5% TAX.—Specify Tape or disk.

Educational Software inc.
4565 Cherryvale Avenue
Dept. SS
Soquel, Ca. 95073
(408) 476-4901

An ATARI® Christmas *continued*

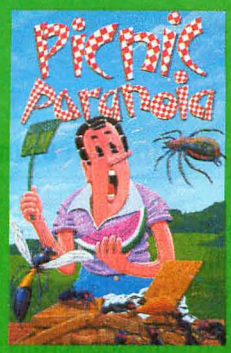
Raster Blaster

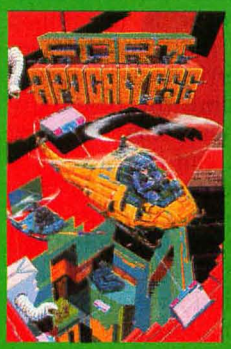
Raster Blaster is a pinball simulation in Machine Language. Although I'm not a pinball enthusiast, I have enjoyed many hours of play on this game. The program is done in very high resolution graphics and four colors. The screen is divided roughly in half, the left half displaying the pinball machine with bumpers, and the right half the scoring, score multipliers and bonus. The blinking lights and sounds are excellent imitations of real pinball. The ball's movement mimics that of a ball in a real pinball game. When it contacts a bumper, your score goes up, the bumper recoils, and the ball moves faster. There are numerous special features.

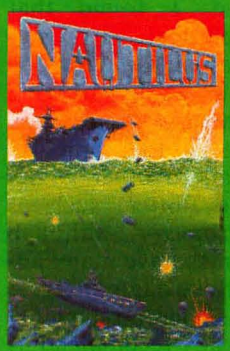
1) Lane Lights. In the alleys at the top of the pinball machine are lane lights, which are rotated by the right flipper and turned on by the ball going through the alley. Turning on all the lane lights adds 10,000 points to your score, lights the "R" raster, and advances the score multiplier. You can cycle the lane lights over and over, getting up to 5X points on your multiplier.

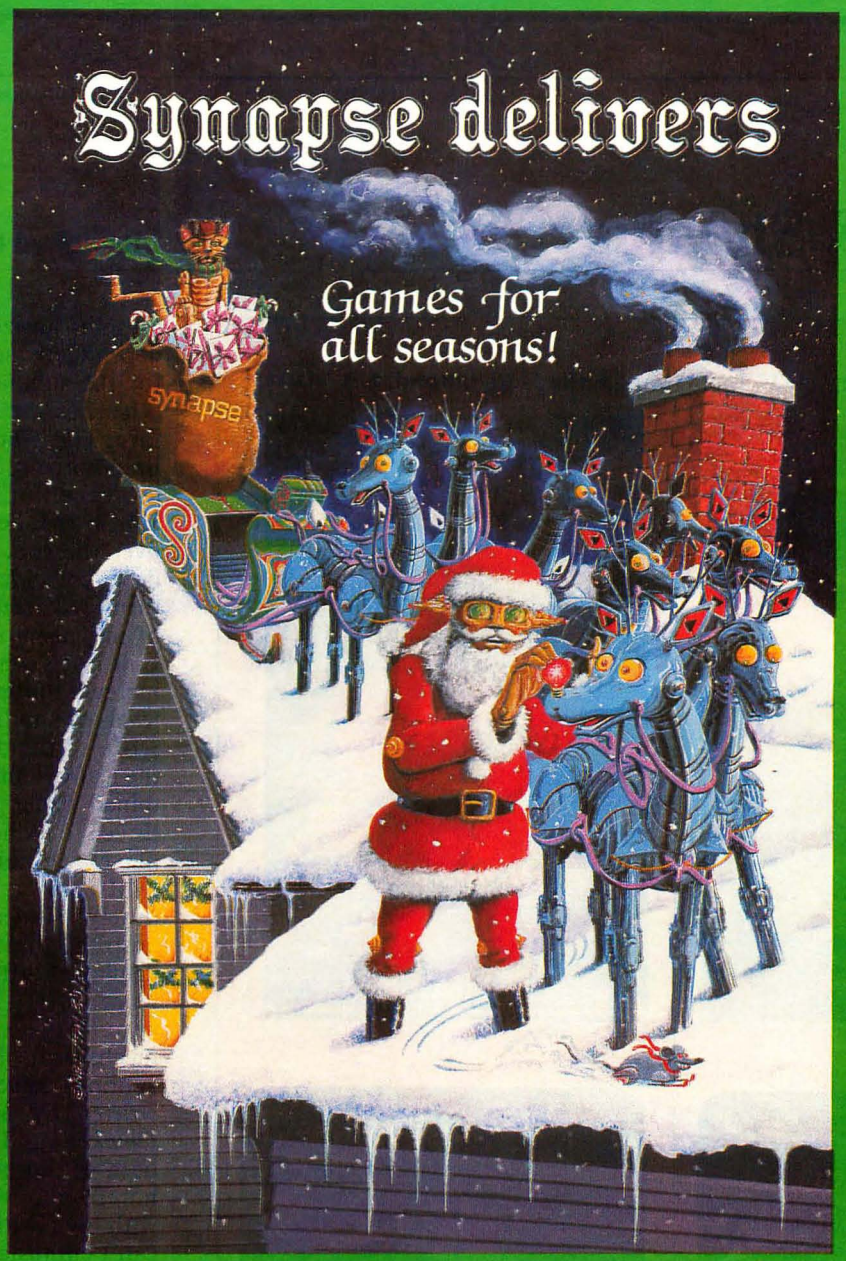
2) Hitting all six of the center targets adds 5000 points to your score and enables three claws. If you shoot a ball into the claws when they are enabled, the ball is held and a new ball is given to the player. Catching three balls (one in each claw) adds 15,000 points to your score, 10,000 points to your bonus and releases all the balls for multiple ball play.

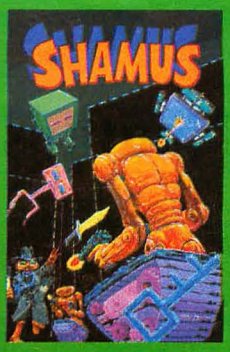
3) Completing all three right side targets adds 10,000 points to your score and lights the blaster "B." Lighting both the "R" and the "B" adds 10,000 points to your bonus. The game also has a realistic spinner worth 600 points per rotation, a force gauge to set the force of each ball launched, and two skill levels. It can be played by one to four persons taking turns. The machine records the score of each person and the high score. *Raster Blaster* uses two joysticks and the fire button on each joystick controls one of the two flippers. The joystick itself is used only to set the force gauge for launching balls. The only real complaint I've heard is that there is no way to "tilt" the machine. *Raster Blaster* is an exceptionally well done pinball simulation, with the excitement of an arcade game.

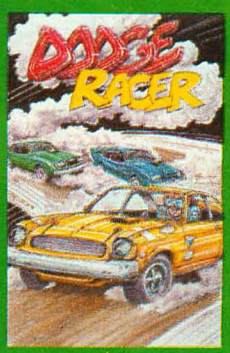


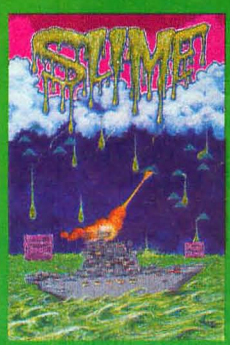


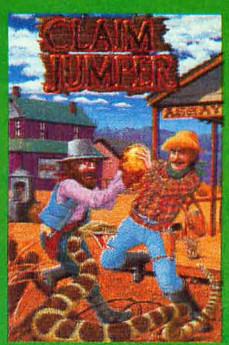












Most games available in disk, cassette, and cartridge for Atari 400/800 computers.
Many titles soon available for the VIC 20, VIC 64, and IBM-PC.

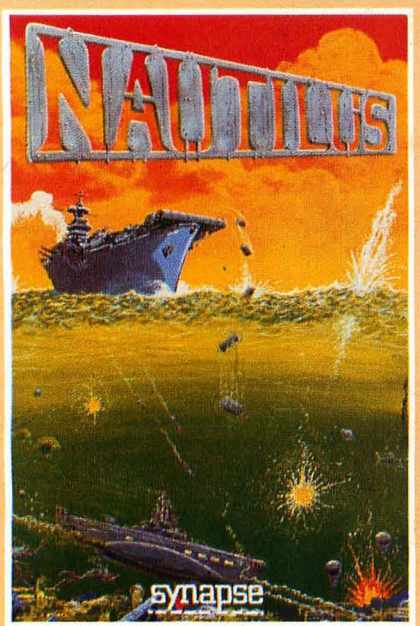
synapse
SOFTWARE

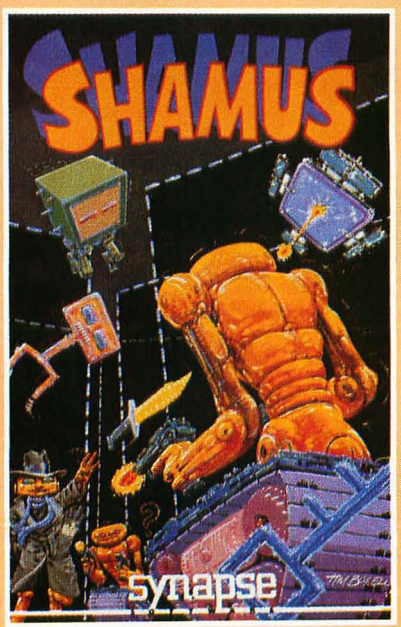
5327 Jacuzzi St., Suite I, Richmond, CA 94804

Nautilus & Shamus

Reviewed by Sheldon Leemon

from Synapse Software, 5327 Jacuzzi St., Suite I, Richmond, CA 94804.





Nautilus: **32K Atari® 400/800 with cassette or disk and one or two joysticks. The suggested retail price is $34.95.**

Shamus: **16K Atari 400/800 with cassette or disk and joystick. The suggested retail price is $34.95.**

Nautilus and *Shamus,* two new games from Synapse Software, prove the old adage about appearances being deceiving. While *Nautilus* is by far the more visually appealing, *Shamus* has a certain intangible quality which makes you want to keep playing long after the novelty has worn off.

Nautilus is a submarine-hunt game for one or two players, in which one player commands a submarine and tries to destroy underwater cities, while the other player (or the computer) controls a destroyer ferrying repair crews from the home base to the repair tunnels from which they can rebuild the cities. The game display makes use of an innovative technique which exploits the advanced graphics capabilities of the Atari; the screen is split horizontally into two independently scrolling windows, one for the submarine display, and the other for the destroyer on the surface. The window for the sub scrolls both horizontally and vertically over the beautiful underwater seascape. When the destroyer is positioned over the sub, so that the two windows coincide vertically, the border turns red, signalling the possibility of an enemy attack. There are nine skill levels which can be selected to increase the speed of play, and eight levels of handicapping to control the score of the destroyer. In addition, the game continues for a predetermined time period, ranging from three to nine minutes, which is player-selectable.

My only criticism of the game is purely subjective. I found that despite the technical virtuosity displayed, after a while the novelty wore off. The game became repetitive, and somewhat mechanical. This is the same reaction I have had to other games written by Mike Potter, such as *Chicken* and *Protector,* although both of those games have many admirers who would strongly disagree with my position. Therefore, while I can't enthusiastically recommend it, I can say that it is technically inventive, extremely well executed and visually appealing. For those reasons alone, it is worth considering.

Shamus does not have the outstanding graphics of *Nautilus.* In *Shamus,* you play the title character, a stick-figure detective who has to solve several puzzles. At the same time, you are fighting off hordes of shoot-em-up bad guys. There is a minimum of color, and the resolution of the graphics is not the greatest. Compensating factors are the sound effects and music, which are appropriate, helpful, and never irritate. The biggest compensating factor, however, is that the game is great fun to play!

Shamus is set in the Lair, a maze-like series of rooms. There are four

continued on page 98

A feast of computing ideas.

If you work with a 6502/6809-based system, you're probably hungry for the facts and ideas that will help you understand the inner workings of your computer. You want to go beyond canned software—use your computer for more than games—learn the advanced programming techniques that enable you to get the most out of your 6502/6809 system.

MICRO, The 6502/6809 Journal, gives you page after page, month after month, of solid information to sink your teeth into. **MICRO** is the premier how-to magazine for serious users of the Apple, PET/CBM, OSI, Atari, AIM, SYM, KIM, and all 6809 based systems including the TRS-80 Color Computer. It's a resource journal internationally respected by professionals in business, industry, and education. Every issue of **MICRO** keeps you informed with up-to-the-minute data on new products and publications:

- **hardware catalog** with organized, concise description
- **software catalog** in an easy-to-use format
- **new publications** listed and annotated
- **reviews and evaluations** of significant products

And there's much more:

- **In-depth hardware tutorials** bring expert advice into your home or office.
- **Detailed discussions of programming languages** deepen and broaden your programming ability.
- **Complete program listings** enable you to increase your machine's capabilities.
- **Bibliography of 6502/6809 information** helps you to find pertinent articles in a timely manner.
- **Special monthly features** with in-depth treatment of one subject or system increase your knowledge of the field.
- **Balanced mix of machine-specific and general articles** for your everyday use as well as long-range reference needs.
- **Informative advertising** focused specifically on 6502/6809 machines keeps you abreast of latest developments.
- **Reader feedback** puts you in touch with other micro-computerists.

MICRO is the magazine you need to get the most from your own 6502/6809 system!

You'll love every byte.

To order, send your check or international money order (payable to MICRO) and the order form at left, to:

Subscription Fulfillment
MICRO, Dept. MI
34 Chelmsford Street
P.O. Box 6502
Chelmsford, MA 01824

Or, for your convenience, call our toll-free number:

1-800-345-8112

(In California, 800-772-3545, Ext. 564)

and charge your subscription to your MasterCard or VISA. (All orders must be prepaid in U.S. dollars or charged to your MasterCard or VISA.)

YES! I want to get more from my microcomputer. Please send me

__ year(s) of MICRO at $_____/year. (Outside U.S. and Canada, please indicate via ☐ surface or ☐ air mail.)

Name

Company

Street

City State Zip Code

☐ Check enclosed for $ _____
☐ Charge my credit card account
☐ VISA ☐ MasterCard

Signature

Card number

Expiration date

SUBSCRIPTION RATES (U.S. dollars)
Yearly subscription (ISSN 027-9002) saves 20% off the single-issue price.

U.S. $24*

Canada $27

Europe $27 ($42 by air mail)

Mexico, Central America, Mideast, North and Central Africa $27 ($48 air)

South America, Far East, South Africa, Australasia $27 ($72 air)

* SPECIAL OFFER—U.S. ONLY:
Save even more—30% off single-issue price: 2 years, $42

Dept. S S

Nautilus & Shamus *continued*

levels, each with 32 rooms. As you enter each room, the room number appears on the screen. These rooms also appear in the same order, so that the left doorway from room 2 leads to room 1, and the right doorway leads to room 3. Some doorways can only be opened by keys, which you must pick up in other rooms. Though usually in the same general area of the maze, these keys are not always in the same room each game. Since the keys for some locks can only be obtained after another lock has been opened, it becomes necessary to memorize or map the various locations of keys and locks. But all of this adventure-type puzzling occurs in the midst of a fast action game. Each room is guarded by several of three types of adversaries, Drones, Droids, and Snap-Jumpers, each type with its own method of attack. You must try to clear the room of enemies with your Ion weapon. While you can leave a room before it is cleared, if you come back again you will find twice as many enemies! To further complicate matters, there is a limit on the amount of time you can spend in any one room. If you stay too long, your arch-rival, Shadow, appears. He cannot be destroyed by your weapons, only briefly stunned, and you must flee if he enters a room. Only after you finish the whole puzzle, consisting of four levels of 32 rooms each, do you have your final showdown with Shadow.

"The combination of arcade action and adventure type problem solving is a great one. Shamus is enjoyable to play alone, but is even more fun with a group of people, each taking turns alternately playing and mapping the Lair."

Many random elements add to the fun. There are pulsing question marks which you may pass over. These randomly distribute bonuses (such as extra points, or extra lives) or penalties (such as taking away one of your hard-earned keys). Extra lives may be obtained from bubbling bottles located randomly within the Lair. There are Pod Rooms, which exist in another dimension, and which can only be entered during a narrow window of time. There are many little puzzles to solve along the way. If the puzzles themselves are not enough of a challenge, there are four levels of play, each of which substantially increases the difficulty of the arcade shoot-em-up action, making the final objective harder to reach.

The combination of arcade action and adventure type problem solving is a great one. *Shamus* is enjoyable to play alone, but is even more fun with a group of people, each taking turns alternately playing and mapping the Lair. The need for a navigator makes it just as much fun to watch someone else play, and offer suggestions, as it is to play the game yourself. You don't need to be a detective to figure out that *Shamus* is a standout in the current field of game software.

A.S.C., INC
Authorized Service Center

"Call us for discount prices, availability, or advice. Remember, service is our middle name." Joe Guzzo, President, A.S.C. Inc.

ATARI 800

800 16K $654.95
800 32K $745.95
800 48K $815.95

ATARI 400

400–16K $264.95
400-32K $364.95
400-48K $430.95

410 Recorder $76.00
810 Disk Drive $444.95

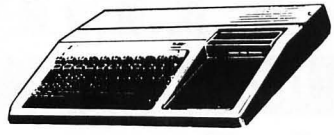
Kits

481 Entertainer $83.00
482 Educator $129.95
483 Programmer $56.95
484 Communicator . . . $339.95

Software

We carry the full line of Atari hardware and software products backed by our Authorized Service Department.

Please call for special prices on all Atari software and APX software.



TEXAS INSTRUMENTS

All models . . . . . Call for prices

MONITORS

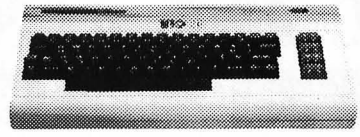
Amdex 12-in. B&W . . . $129.00
Amdex 12-in. Green . . $139.00
Amdex 13-in. Color . . . $349.00
BMC--BM 12A $114.95
NEC 12-in. B&W $169.95
NEC 12-in. Color $339.00
TI 10 . . . . . $344.95
Corvus . . . . . Call for prices
Zenith Z19 12-in. $115.00

DISKS

Elephant . . . . . 10/$21.95
Elephant DD . . . . . 10/$24.95
Dysan 5¼-10/16 . . . . . 10/$38
Verbatim 8-in.SD--SS . . . 10/$32
Verbatim . . . . . 100/$245
Scotch 3M . . . . . 10/$25.95

DISK DRIVES

Corvus . . . . . Call
Percom . . . . . Call
Paradynamics . . . . . Call



VIC-20

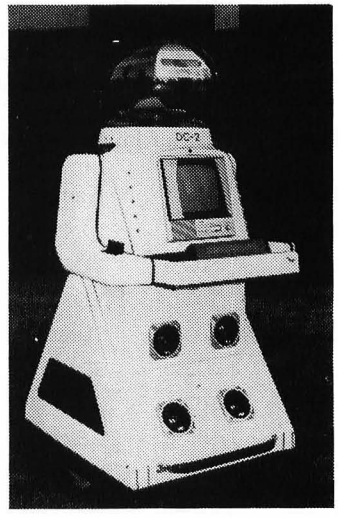
VIC-20 Personal Computer $175

GAMES/ PROGRAMS

Unlimited Availability . . . . . Call

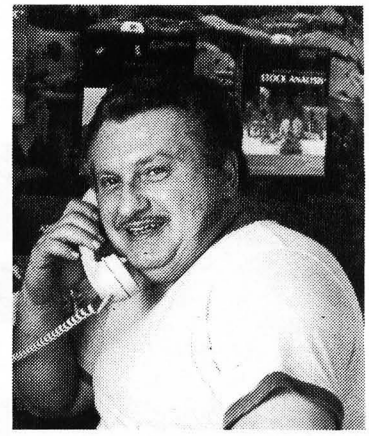
COMPUTER DESKS

Send for catalog and photos.



DC-2 ROBOT

Operates on any home computer.
Stands 4 feet high.
Call for details and prices.



TERMINALS

Televideo $569.00
910 . . . . . $569.00
912C $694.95
920C $744.95
950 . . . . . $934.95

Call for Computers

MODEMS

Lexicon $149.00
U.D.S. 1030ALP $169.00
Racal Vadic VA 3212 . . $749.00
Prentice Star $179.95

PRINTERS

Brother HR-1 Parallel . . $829.95
Brother HR-1 Serial . . . $880.00
Smith-Corona TP-1 Parallel $665
Smith-Corona TP-1 Serial . $665
C. Itoh Starwriter $1,350
Diablo 630 . . . . . $1,999.95
Epson printers . . . . . Call
Centronics printers . . . . . Call
All name brands . . . . . Call

NEW•NEW

Coleco-vision, Mattel Intellivision, and Atari VCS game machines and game cartridges. Call us for information on the latest, improved models and best prices.

Limited space prevents a full listing of the lines we carry. If you don't see it listed, we can get it for you, for less.

TO ORDER: Phone orders accepted via Mastercard, Visa, or bank wire transfer. Visa and MC service charge of 2%. Mail orders may send charge card number (include expiration date), cashiers check, money order, or personal check (allow 10 business days for personal or company checks to clear). Please add 3% ($5.00 minimum) for UPS shipping, handling, and insurance. APO and FPO include 5% ($7.00 minimum) for postage. California residents add 6% sales tax. Please include phone number on all orders. FOREIGN ORDERS include 1% handling. Credit cards not accepted on *foreign* orders. All equipment is in factory cartons with manufacturer warranty. Opened products not returnable. Restocking fee for returned merchandise. Equipment subject to price change and availability. Retail prices differ from mail order prices. WE SHIP THE SAME DAY ON MOST ORDERS.

Call Collect (714) 284-5615 7 Days/Week

Monday through Saturday 8am–8pm
Sundays and Holidays 12n–6pm

After 11/6/82 , Call (619) 284-5615

A.S.C. Inc., 7436 University Ave.
La Mesa, CA 92041

EXPLORING THE ATARI® FRONTIER

Display List Interrupts

by Alan J. Zett

Editor's Note: *This is the fourth part of a series. To avoid confusion, we advise that you read the first three installments (**SoftSide,** May 1982, issue 31, and issue 34) before reading this one. A solid background in BASIC, as well as the rudiments of assembly language, are also recommended.*

A Different Tune

Now that we have covered the subject of modified Display Lists (DL's) in the last three installments, we can approach underlying concepts that make some of the most spectacular visual effects. If the capabilities presented for custom display lists have appeared interesting, the options available while using a modified display list will astound you! As the months unravel, we'll see that once you've unlocked the power stored in the ANTIC chip, programs can be written which *far* exceed the limited ability of BASIC. When short and simple Machine Language routines are added to BASIC, the quality of the software produced improves substantially.

Figure 1: A list of important memory locations

Decimal	Hex	Name	Description
00087	$0057	DINDEX	Current GRAPHICS mode of screen.
00512	$0200	VDSLST	LSB of Display List Interrupt routine.
00513	$0201	VDSLST	MSB of Display List Interrupt routine.
00559	$022F	SDMCTL	Control of video display and PMG.
00560	$0230	SDLSTL	LSB of Display List.
00561	$0231	SDLSTH	MSB of Display List.
01664	$0680	USRAM	128 unused bytes of user RAM.
54282	$D40A	WSYNC	Wait for horizontal sync register.
54286	$D40E	NMIEN	Non maskable interrupt enable.

New Business

Our topic, this time, is Display List Interrupts (DLIs). The DLI is an option that is available in any of the ANTIC Character, Graphic, or Blank mode line instructions. When ANTIC is processing the display, it sends information to CTIA (GTIA in newer machines) for every line on the screen. If you remember, each line is known as an ANTIC mode line. A mode line in turn is a group of television scan lines that, when combined, represent a visual line of display text or graphics. After the video display is composed, it is then sent to the TV or monitor where each mode line is traced on the screen, one scan line at a time. To the human eye, all the scan lines appear to be on the screen at once.

To the computer, however, there is a time delay before each mode line is drawn on the screen. There is actually enough time to execute a small number of Machine Language instructions before the the next mode line is drawn. The length of this code is limited to the number of cycles each instruction takes of the CPU's time. On the average, there is enough time for a few minor, non-repetitive operations. All that we need is some way to check the normal flow of CPU processing in order to divert some of its precious time to a simple task. This is the function of the DLI: to interrupt the normal processing of the DL.

Once we have the DL's attention, what can we do? We can change the color registers in between individual mode lines on the screen. This is the technique used in this month's Atari version of *Hopper*. In fact, we will be dissecting the DL routines from that program later. A form of animation could be accomplished by flipping the DL's pointer to screen memory data between different areas of memory. Player/Missile Graphics (PMG) could be moved or have their colors change. The list is endless and limited only by the amount of time it takes to execute the instructions. Sound effects could also be generated, although a complicated task such as music or graphic animation would require more time. The Vertical Blank Interrupt (VBI) is provided for this purpose. The main difference between DLIs and VBIs is that the DLI can execute between mode *lines* whereas the VBI executes between each complete *screen*. We'll be covering the VBI in a later installment.

Yet Another DL

Figure 1 contains a list of some important memory locations when using DLIs. We can use it in conjunction with Listing 1 to discover how to create a working DL with interrupts. Listing 1 is the routine to create the DL from Atari *Hopper*. We'll go over that in detail in just a moment. Figure 2 is a representation of the DL generated by Listing 1. It justifies a little discussion because of the use of the DLI option.

The DL for *Hopper* is only 34 bytes long. Unlike the Load Memory Scan (LMS) option, which requires two extra bytes for a 16 bit screen memory data address, the DLI is a stand alone instruction. ANTIC is smart. It knows that whenever it processes a character or graphics mode line, only the lower four bits (or nibble) are used. The higher four bits should be reset (equal to zero) on a normal instruction. The exception is that a blank mode line instruction *requires* that the low nibble be zero; that is how it is distinguished from a character or graphics mode line. If, however, ANTIC detects a non-zero high nibble, it will recognize that a mode line option was specified, as long as the corresponding low nibble *is not* zero. Blank mode lines use the lower three bits of the high nibble (bits four, five, and six) to determine how many blank scan lines to skip. On a character or graphics mode line, if bit six is set (equal to one), the LMS option is executed. If bit seven is set, ANTIC interprets this as the DLI option. This means that although the blank mode line can use the DLI option, it *cannot* use the LMS option. The LMS option would be interpreted as part of the number of scan lines to skip. In a character or graphics mode instruction, you can set both bits six and seven at one time. This was done in the first visible mode line of the *Hopper* DL.

If you look at Figure 2, you will see that the first three bytes of the DL are blank mode lines. This is a standard technique used to bring the start of the video display down 24 scan lines (Each blank mode line instruction is set for skipping eight scan lines.) from the top of the video signal's trace pattern to make sure all of it can be seen. Bytes 4 through 6 accomplish two tasks. They set the pointer to the top of screen memory data for the first visible mode line, and also call the first DLI to modify the background color of mode line two. Bytes 7 through 9 adjust the pointer to screen memory data for mode line two. The reason for this will be explained later. Bytes 10 through 31, with the exception of bytes 18, 20, and 27, are all standard GRAPHICS 0 mode lines. Bytes 18, 20, and 27 are also GRAPHICS 0, but each of these has bit seven set (128 added to the normal value) to generate a DLI. Byte 32 is the standard Jump instruction with bit 6 set (64 added to the base value). Jump instructions are totally separate from the other three kinds. Setting bit 6 will cause the Wait for Vertical Blank (WVB) option to execute. This will cause the DL to start processing again at byte number one, but only after the video display has had a chance to bring the scanning beam up to the top of the display. Now that we have walked through the DL itself, we can show how it can be generated from BASIC.

Figure 2: A custom DL with interrupts generated by listing 1 (from Atari *Hopper*).

DL Byte #	Byte Value	Mode Type
01	112	Blank
02	112	Blank
03	112	Blank
04	198	GRAPHICS 1 w/LMS and DLI options.
05	nnn	LSB of screen memory data.
06	nnn	MSB of screen memory data.
07	066	GRAPHICS 0 w/LMS option.
08	nnn	LSB of next line of screen data.
09	nnn	MSB of next line of string data.
10	002	GRAPHICS 0
11	002	GRAPHICS 0
12	002	GRAPHICS 0
13	002	GRAPHICS 0
14	002	GRAPHICS 0
15	002	GRAPHICS 0
16	002	GRAPHICS 0
17	130	GRAPHICS 0 w/DLI option.
18	002	GRAPHICS 0
19	002	GRAPHICS 0
20	130	GRAPHICS 0 w/DLI option.
21	002	GRAPHICS 0
22	002	GRAPHICS 0
23	002	GRAPHICS 0
24	002	GRAPHICS 0
25	002	GRAPHICS 0
26	002	GRAPHICS 0
27	130	GRAPHICS 0 w/DLI option.
28	002	GRAPHICS 0
29	002	GRAPHICS 0
30	002	GRAPHICS 0
31	002	GRAPHICS 0
32	065	Jump w/WVB option.
33	nnn	LSB of Display List.
34	nnn	MSB of Display List.

Take Apart Time

Line 10000: Sets numeric constants. Variables are used to speed up execution time and cut down on program size. Every time you enter a number into your program, it is stored in memory as a six byte floating-point number. That means that the repetitive use of numbers such as 0 and 1 in a BASIC program takes up *six* bytes. After a while, this can really eat up storage space. On the other hand, if you assign a variable the value of say, zero, and use it instead, every number will take up only the number of characters required to specify a variable. How many bytes does it take, you ask? Only *one*! Each variable, no matter how many characters long, is stored as a relative byte position in a lookup table.

Line 11000: Set up a full screen GRAPHICS 5 display. Here we have violated one of the first rules I set in past installments. When creating a modified DL, we *should* choose the GRAPHICS mode in the modified DL that takes up the most memory. If you look at Figure 2, you

will see that nowhere in the DL are there any GRAPHICS 5 mode lines. By rights, we should have used a GRAPHICS 0 DL to modify. The explanation is a little complicated. In the case of *Hopper*, which uses a GRAPHICS 1 mode line at the top followed by 23 mode lines of GRAPHICS 0, the normal problems associated with having a GRAPHICS 1 mode line by itself had to be overcome. Normally, when you add a mode line such as GRAPHICS 1 which uses 20 screen bytes to a mode line such as GRAPHICS 0, which uses 40, you must use two of them so that the visible GRAPHICS 0 lines again line up with the left edge of the screen. (See ***SoftSide*** issue 31.) This could not be done in *Hopper* because all 23 GRAPHICS 0 mode lines were needed. To get around this, the DL was *forced* to ignore the missing 20 bytes by resetting the pointer to screen memory data 20 bytes further down the screen. By moving the actual start of the next mode line's screen memory data down twenty bytes, the wrap around that usually occurs as a side-effect was defeated. This is done with the LMS option, and requires more than one DL byte, thereby changing the length of DL into which it is inserted. Padding the screen with an extra GRAPHICS 1 mode line does not change the length of the DL. Using a GRAPHICS 0 screen with a DL that was longer than the original would be a complicated procedure. Enter GRAPHICS 21! This version of the GRAPHICS 5 mode uses the same amount of screen memory data as GRAPHICS 0, but has more mode line bytes in the DL with which to work. With extra space provided, we could POKE in a GRAPHICS 0 DL with enough room left to use the second LMS option. The result was a display that cannot be duplicated with normal procedures. The POKE 87 tells the Operating System (OS) that we are no longer in GRAPHICS 21.

Listing 1

```
10000 N0=0:N1=1:N2=2:N3=3:N4=4:N5=5:N6
=6:N7=7:N8=8:N9=9:N10=10
11000 GRAPHICS 21:POKE 752,N1:C=N0:POK
E 87,N0
11010 DL=PEEK(560)+PEEK(561)*256+N4
11020 MEM=PEEK(DL)+PEEK(DL+1)*256+40
11030 MH=INT(MEM/256):ML=MEM-MH*256
11040 POKE 559,N0:POKE DL-N1,198
11050 POKE DL+N2,66:POKE DL+N3,ML
11060 POKE DL+N4,MH
11070 FOR J=N5 TO 26:POKE DL+J,N2:NEXT
 J
11080 POKE DL+12,130:POKE DL+15,130
11090 POKE DL+22,130:POKE DL+27,65
11100 POKE DL+28,PEEK(560)
11110 POKE DL+29,PEEK(561)
11120 READ A:IF A=999 THEN 11140
11130 POKE 1744+C,A:C=C+N1:GOTO 11120
11140 POKE 512,208:POKE 513,N6
11150 POKE 1774,176:POKE 1775,180
11160 POKE 1776,N0:POKE 1777,144
11170 POKE 1778,N0:POKE 54286,192
11180 POKE 559,34
11190 DATA 72,138,72,174,242,6,180,238
,6,141,10,212,141,24,208,232
11200 DATA 224,4,144,2,162,0,138,141
11210 DATA 242,6,104,170,104,64,999
```

Line 11010: The variable DL is set to the start of the DL plus four.

Line 11020: MEM is set equal to the start position of the screen memory data of the second mode line. This is used with the LMS option to skip the unused 20 bytes from the GRAPHICS 5 mode line which will be POKEd in later.

Line 11030: MH equals the MSB of MEM while ML equals the LSB of MEM.

Line 11040: POKE 559,0 shuts off the screen to speed up Atari BASIC and to hide the changing display from the user. POKEing DL-1 sets the top visible mode line of the DL to GRAPHICS 5 with the mandatory LMS option for the start of the screen display, as well as a DLI option. The LMS option is specified by adding 64 (set bit 6) to the DL mode byte.

Line 11050: POKEing DL+2 with 66 sets the second visible line of the display to be a GRAPHICS 0 line with the LMS option. ML is POKEd into the low screen byte position of the LMS option.

Line 11060: POKEs in the high screen byte of the LMS option.

Line 11070: POKEs in all of the remaining GRAPHICS 0 lines that will be used. Note that some of these locations will be modified.

Line 11080: POKEs in two GRAPHICS 0 mode lines with the DLI option. When ANTIC encounters a DLI option, it responds by generating a Non Maskable Interrupt (NMI). Every time an NMI is encountered, the CPU jumps to the OS NMI handler routine. OS then checks the status of the Non Maskable Interrupt ENable register (NMIEN) which is preset to select what types of NMIs will be processed. It is actually possible to force the Atari to ignore NMIs. If the interrupt originated in the area of memory occupied by the DL, and the DLI enable bit of NMIEN is set (bit seven), a jump to the routine specified by the DLI jump vector at memory locations 512 and 513 is executed. This pointer must be set to the start of the programmer's DLI routine. It is interesting to note that the DLI option must be specified in the mode line above the visible mode line you wish to interrupt. The interrupt routine will not start to execute until half of the mode line has been drawn. This is caused by delays in the OS for processing. By starting one mode line above, the interrupt will begin processing just before the visible screen line we wish to modify. This means that in order to modify the top visible line of the screen, the DLI must occur in one of the Blank mode line instructions. There is a problem, however. The interrupt will *always* start near the *middle* of the mode line in which the DLI option was specified. This means that if you change the background color in your DLI routine, it will not start at the leading left-hand side of the screen, but rather, somewhere in the middle. In addition, the break in the visible screen line will zigzag back and forth because of timing variations between each processing of the display. The only way to overcome this is to use a special memory location known as WSYNC. This location, when written to, will tie up the CPU until the start of the horizontal sync retrace scan. This will assure you of starting your routine at the beginning of a visible screen line. Simply writing *anything* to this location will cause it to function. This method was used in the assembly language DLI routine used in *Hopper*.

Line 11090: POKEs in the remaining GRAPHICS 0 DLI mode line and sets up a Jump instruction with the WVB option.

Lines 11100-11110: Provides the address needed for the Jump instruction to process.

Lines 11120-11130: POKE the DLI handler routine into locations 1744 ($06D0) on up. When the end of data is found (999), jump to line 11180.

Line 11140: POKEs the address of the DLI handler routine into the DLI jump vector VDSLST at locations 512 ($0200) and 513 ($0201).

Lines 11150-11160: Set up the color lookup table used in the DLI handler routine.

Line 11170: Sets the color table index to the start and enables the DLI interrupt vector by POKEing 192 into NMIEN.

Line 11180: Turns the screen back on after all the work is finished.

Lines 11190-11210: Contain the decimal values of the assembly language DLI handler routine.

And Now... The 6502

That's all there is to setting up the DLI display, but without a DLI handler routine, it won't work. Even though the source code provided in Listing 2 is commented, there are some techniques used in it that require explanation. After all, a DLI is only the means of making a fancy display. If you want to do something useful with a DLI, you have to do it yourself. That is the major advantage of the DLI — the ability to add your own DL processing commands to those already provided by ANTIC.

Timing considerations in the processing of a DLI routine will allow only 15 machine cycles before the start of the horizontal sync and about 11 cycles between the start of the horizontal sync and the appearance of the first electrons of the scan line. (Each 6502 instruction, or OP-CODE, takes a certain number of clock cycles to execute. For more information on the cycle times of Machine Language OP-CODES, look in any good quality 6502 programming book.) That means that you can use about 15 cycles before using the WSYNC location, and *about* 11 more after the CPU restarts. In realistic terms, this means that changing more than three color registers *at one time* is impossible. However, more than one DLI can be used. This can be a problem, however, because there is only one DLI jump vector. If more than one DLI is used, the DLI handler must be able to figure out what interrupt it is currently processing. Luckily, there are many ways to accomplish this task.

If the DLI routine must do the same thing with different parameters or values each time it executes, a simple table and pointer can be used to tell it what value to use. This is the method used in *Hopper*. There are four DLI's with which the DLI handler must cope. The process of changing the color register is common to each DLI — only the color used must be changed at each mode line. All the *Hopper* DLI routine does is hold a table of colors in memory that will constitute the desired display. By keeping track of a pointer within the table, the routine knows exactly which color to use each time it is called.

But take, for example, a display with three DLI's. The first must change the color of the screen; the second must move the position of a PMG; and the third modifies a DL pointer to screen memory data. Each routine is *distinctly* different, but there is only *one* DLI jump vector. How can

Listing 2

```
                0100        *=  $06D0     ;SET START OF ASSEMBLY.
                0110 ;
                0120 ; DEFINE SYSTEM EQUATES
                0130 ;
                0140 WSYNC  =   $D40A     ;WAIT FOR HORIZONTAL SYNC REGISTER.
                0150 BGRND  =   $D018     ;BACKGROUND COLOR HARDWARE REGISTER.
                0160 ;
                0170 ; START OF DLI HANDLER ROUTINE
                0180 ;
06D0: 48        0190 START  PHA           ;SAVE THE A REGISTER ONTO THE STACK.
06D1: 8A        0200        TXA           ;PUT X REGISTER INTO A AND THEN
06D2: 48        0210        PHA           ;SAVE IT ONTO THE STACK ALSO.
06D3: AE F2 06  0220        LDX STORE     ;GET THE CONTENTS OF STORE INTO X.
06D6: BD EE 06  0230        LDA TABLE,X   ;GET COLOR POINTED TO BY X INTO A.
06D9: 8D 0A D4  0240        STA WSYNC     ;WAIT FOR TIMING, THEN PROCEED.
06DC: 8D 18 D0  0250        STA BGRND     ;CHANGE COLORS IN HARDWARE REGISTER.
06DF: E8        0260        INX           ;POINT X TO NEXT COLOR FOR DLI USE.
06E0: E0 04     0270        CPX #$04      ;IF WE'RE NOT PAST THE TABLE'S END,
06E2: 90 02     0280        BCC OUT       ;THEN GET READY TO LEAVE,
06E4: A2 00     0290        LDX #$00      ;ELSE POINT X TO THE TABLE'S START.
                0300 ;
                0310 ; END DLI, SET UP FOR NEXT
                0320 ;
06E6: 8A        0330 OUT    TXA           ;PUT X BACK INTO THE A REGISTER.
06E7: 8D F2 06  0340        STA STORE     ;SAVE UPDATED COLOR TABLE POINTER.
06EA: 68        0350        PLA           ;REMOVE OLD X REGISTER.
06EB: AA        0360        TAX           ;PUT IT BACK IN PLACE.
06EC: 68        0370        PLA           ;RESTORE OLD A REGISTER.
06ED: 40        0380        RTI           ;EXECUTE RETURN FROM INTERRPUT.
                0390 ;
                0400 ; DLI COLOR TABLE AND POINTER
                0410 ;
06EF: 00 00     0420 TABLE  .WORD $00     ;RESERVE 4 BYTES FOR COLOR TABLE.
06F1: 00 00     0430        .WORD $00
06F3: 00        0440 STORE  .BYTE $00     ;RESERVE 1 BYTE FOR TABLE POINTER.
                0450        .END
```

the other two routines be used? There are two ways. If the routines are *very* short, a general menu DLI routine can be set up with a table of all the DLI routines. By keeping a pointer to specify which routine is currently required, the menu can branch to the location of the routine as shown in the table.

This method, however, eats up a lot of the CPU time in computing and branching to the proper DLI routine. Another method is to simply set up the address of the first DLI from BASIC, and after the DLIs are enabled, make each DLI routine *modify the DLI jump vector to point to the next routine.* This requires that each DLI routine knows the memory location of the next routine to be executed, and puts this address into the DLI jump vector at 512 and 513, just before finishing. This method is very effective and simplifies the processing of separate DLIs.

Basic Pitfalls

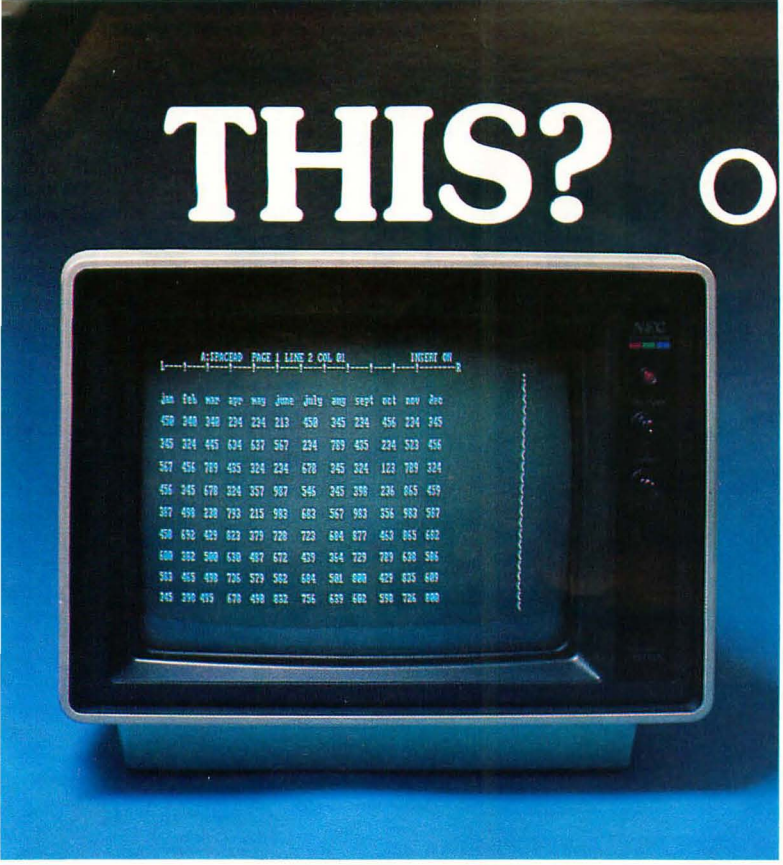
The first thing any DLI routine *must* do is save all of the registers it will use on the stack or in some other memory location (the stack is fastest). After it is finished processing, it must also restore the registers to their original condition by removing them from the stack. This is because the OS does not save any registers other than the Program Counter (PC). In effect, the OS *never sees* the interrupt occur, and as such, has no reason to suspect that any of the registers it was using may have been modified. Needless to say, this could cause grave problems in the OS, hence the need to save the registers.

Second, the programmer must be aware that the shadow registers in RAM (such as 710 for the background color) are written to the hardware registers in GTIA only *during* the VBI. For example, the *Hopper* DLI routine has to change a color register. Since the VBI occurs between the end of one screen display and before the next one is drawn, all of the DLIs will change the color register *before* the VBI is executed. If the DLI used shadow registers, only the *last* color change would be written to GTIA during the VBI because the other color changes would be overwritten before the VBI could look at them. For this reason, whenever you wish to modify a parameter during the DLI that is shadowed into RAM, you *must* write directly to the hardware register in GTIA. This also has the benefit (or disadvantage) of having the normal shadow register restore the modified parameter to its original state at the start of each screen. The OS and hardware technical manuals from Atari cover all of the shadow and hardware register locations in detail.

The only noticeable side-effect of using DLIs is that there is a small problem with the OS keyboard click routine. Whenever a keyboard character is pressed, OS causes the Atari's internal speaker to click. The OS times the clicking by using the WSYNC location. Since we also use this location and process our DLI routines with the expectation of being somewhere in the middle of a scan line when the DLI occurs, typing a key can sometimes cause the screen to jump or flicker. This is because when OS uses the WSYNC location, and we also use it in our routine, the DLI starts executing one line too late. Also, ANTIC and CTIA are confused because the CPU is tied up for a critical segment of time. The jumping is not terrrible, although it can be annoying. The only direct way around this is to not accept any keyboard input. The keyboard can be disabled with a few POKEs but the problem is not really bad enough to warrant the loss of keyboard entry. Take a look at the *Pokey Player Editor* (***SoftSide***, issue 34). It jumps a little, but not enough to distract you.

DLI Fini

These are most of the fundamentals of the DLI. There will always be certain situations that require special solutions, but the material presented so far should cover almost any practical application. We can now sit back and relax as the information slowly disappears from our minds. I would suggest re-reading this article again at a later date to help refresh the large amount of information presented. I am sure that this article will generate some questions, and I am currently compiling a list for a future installment. That's it for this time. Now is a good time to sit back at your computer and experiment with the concepts presented. Learning should be a fun experience, not a frustrating one. Take the time to play with your computer. It's the only way to find out what it truly means to *Explore the Atari Frontier.*



The Graphics Generator

Business and Technical Graphics for the IBM Personal Computer

Now you can have the power, the control, the visual clarity you need for fast and efficient data analysis! Presenting **The Graphics Generator**—the *only* software currently available that gives you complete access to the PC's vast color graphics capacity. You'll create BAR GRAPHS • PIE CHARTS • LINE GRAPHS • even FUNCTION GRAPHS faster and more easily than you've ever imagined! What's more, it features a unique presentation mode, hard copy option, easy edit and repair, retain and recall capability, and more. *Imagine—it even generates regression lines for trend forecasting!*

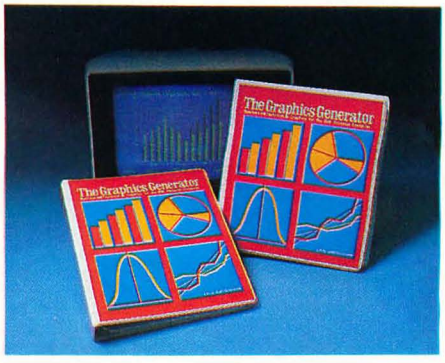
IBM Personal Computer, IBM PC are registered trademarks of International Business Machines, Inc.

The Graphics Generator can take data *directly* from the VisiCalc® worksheet! You'll be able to quickly and easily transform columns and columns of figures into clear, easy-to-read graphs!

System Requirements

- ∧ DOS System—at least one disk drive
- ∧ Advanced BASIC language
- ∧ 64k memory
- ∧ Color/Graphics interface board
- ∧ Color/Monochrome monitor that plugs into the color/graphics board
- ∧ (Optional): IBM® cps dot matrix printer with Epson Graftrax-80® or Epson MX-80® matrix printer with Graphtrax-80®
- ∧ List Price: $195.00

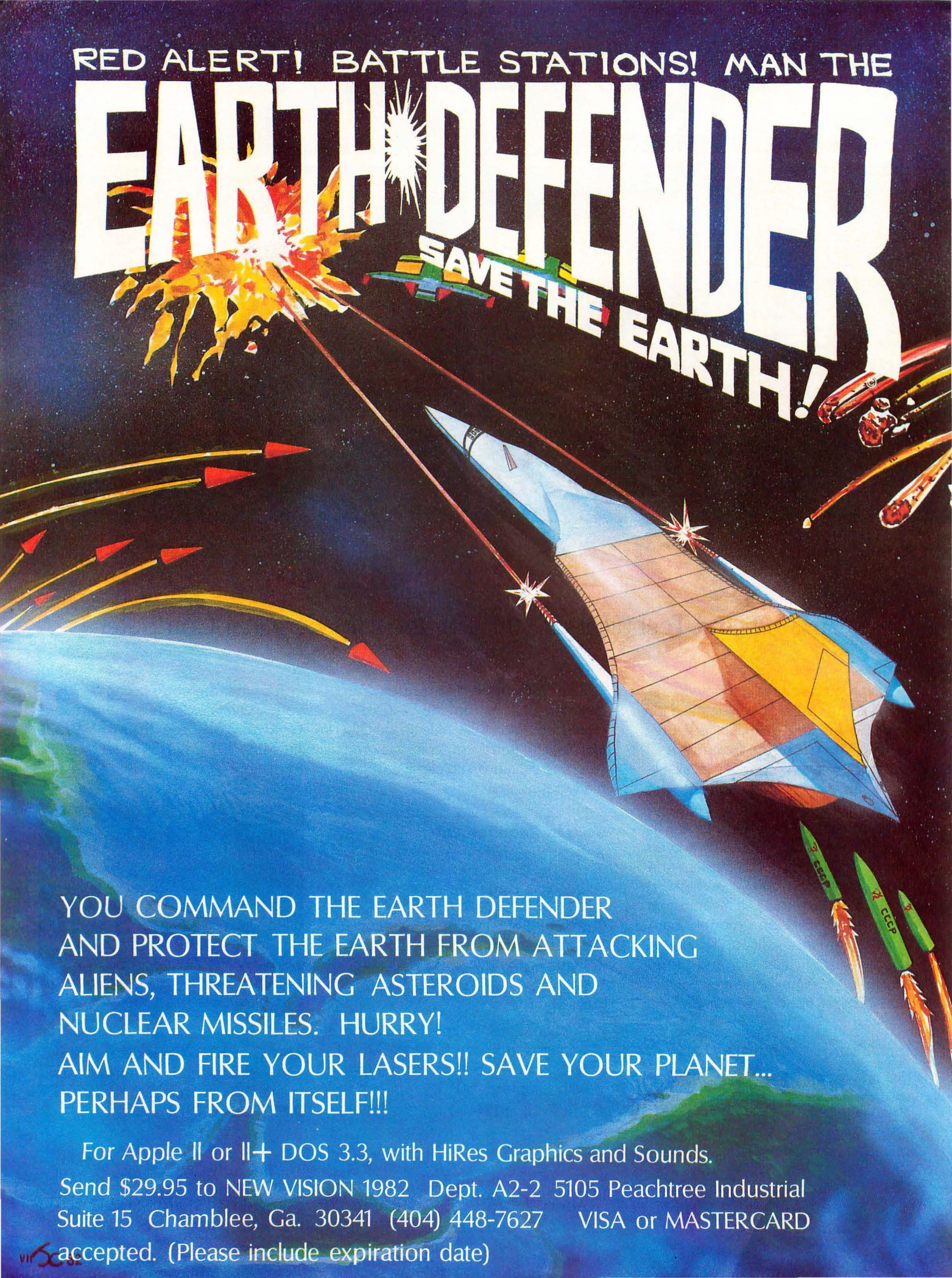
So what are you waiting for? See your PC dealer for a demonstration. Or **Call Toll-Free 800-638-0220** (in MD, 301-262-6300) for *free* information!

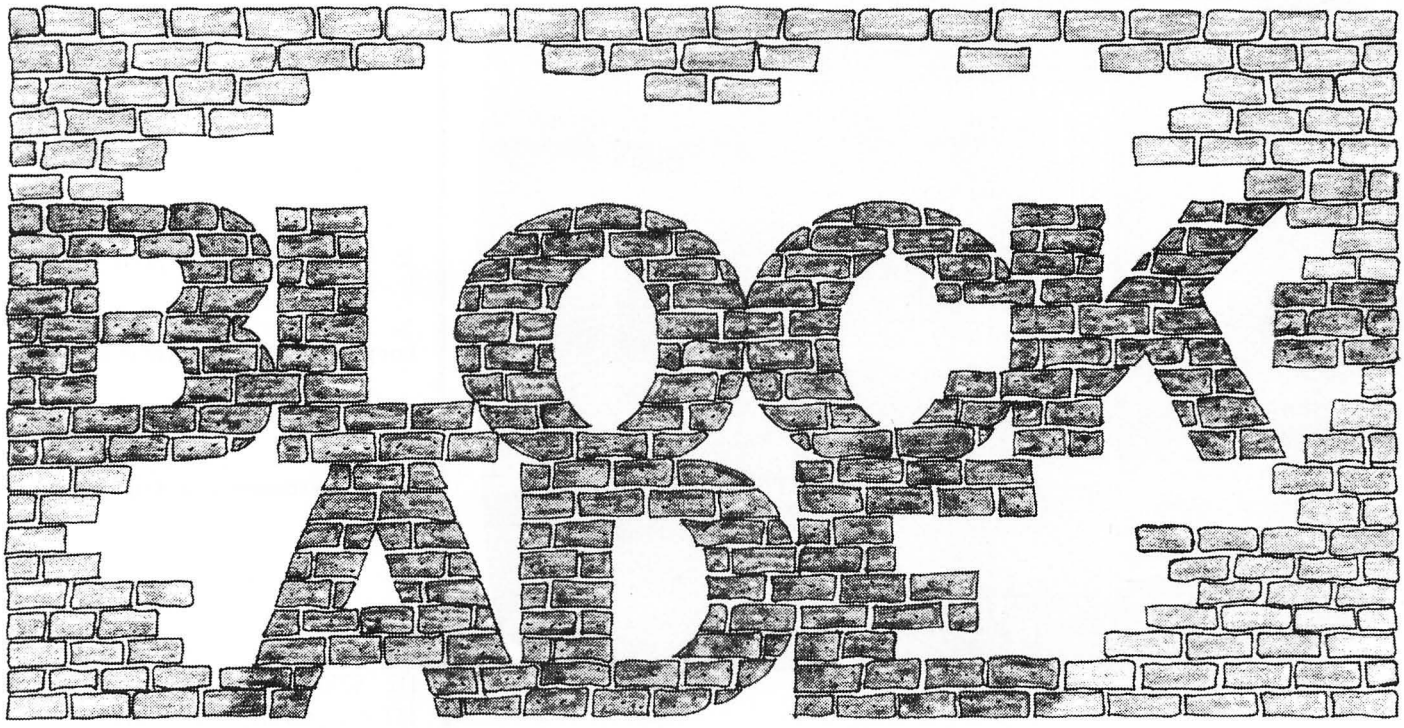


The Graphics Generator includes two diskettes and 160 page documentation manual. (1982/ISBN 0-89303-266-2/D2662-7)

Brady

Robert J. Brady Co. • Bowie, Maryland 20715 • (301) 262-6300
A PRENTICE-HALL PUBLISHING AND COMMUNICATIONS COMPANY


RED ALERT! BATTLE STATIONS! MAN THE
EARTH DEFENDER
SAVE THE EARTH!
YOU COMMAND THE EARTH DEFENDER
AND PROTECT THE EARTH FROM ATTACKING
ALIENS, THREATENING ASTEROIDS AND
NUCLEAR MISSILES. HURRY!
AIM AND FIRE YOUR LASERS!! SAVE YOUR PLANET...
PERHAPS FROM ITSELF!!!
For Apple II or II+ DOS 3.3, with HiRes Graphics and Sounds.
Send $29.95 to NEW VISION 1982 Dept. A2-2 5105 Peachtree Industrial
Suite 15 Chamblee, Ga. 30341 (404) 448-7627 VISA or MASTERCARD
accepted. (Please include expiration date)



by Louis Roy

***Blockade* is a two player game for a 48K Apple II™ with Applesoft and game paddles.**

The object of *Blockade* is to build a wall which traps your opponent, forcing him to run into the wall. In other words, you must surround him. The first player to run into a wall gives a point to his adversary. If the players collide, each gets a point. The first player to get nine points wins the game.

You can play with or without a border surrounding the playfield. Without a border, the players can exit from one side and wrap around to the other side. With a border around the screen, no wrap around is permitted.

To change the direction in which you are moving, position your paddle left or right of the center position and press the button. This makes you turn 90 degrees in the indicated direction. Each time you turn, the button must be released and pressed again.

When the game is over, the screen will begin to flash. Press ESC to end the program, or any other key to play another game.

Programming Notes

1) In the drawing of the words "Blockade" and "by," I have compressed the data in lines 3200 and 3210. First, the data is read in as a string, then the Apple scans the string, character by character. The first character indicates the number of squares to draw on the hi-res screen, the next character indicates the number to leave blank. The Apple reads these numbers by pairs until a "0" is found. This tells the Apple to increment the Y-coordinate by 1 and reset the X-coordinate to its initial value (line 1150). As for the hi-res display of the author's name, I'll leave the unraveling of my logic to you.

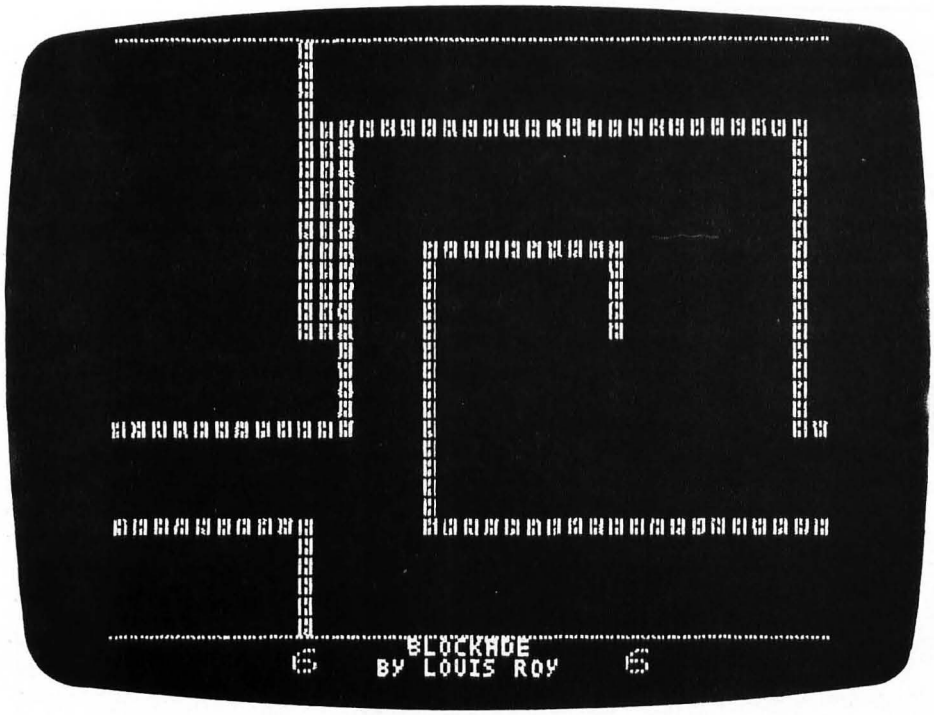
2) In lines 10 and 30, the POKE 230,32 tells the Apple which hi-res page to erase, (Here, it is page 1: $2000.) and the CALL 62450 erases that page. Why not use HGR? Because the HGR command displays the hi-res page, then erases it. We would see the process being done. I think it's a small defect in BASIC. The command should erase the screen, then enter the graphics mode.

3) The CALL 768 executes the Machine Language sound routine.

4) At the end of the game (line 530) we see a CALL 804. This Machine Language subroutine inverts all the bits in hi-res page 1. The inverted picture is stored in hi-res page 2. The "end sequence" alternates between page 1 (normal picture) and page 2 (inverted picture), resulting in a stunning effect that is a snap to create.

Variables

A(*): Flag to signal a possible collision between a player and a wall.
C(*): Players' colors.
D(*): Direction of player movement (1 to 4).
S(*): Players' scores.
X(*), Y(*): x- and y-coordinates in playfield matrix.
XY(*,*): Playfield matrix.
Z(*): Flags to indicate if a player's button was pressed (equals 1 if true).
A, B, I: Various uses.
K: Constant used in title page.
X, Y: Used to initialize matrix XY.
A$: General purpose string.
M$: Answer to "Border?" prompt.




```
SS SS SS SS SS SS SS SS SS SS SS
SS                             SS
SS      APPLESOFT BASIC        SS
SS         'BLOCKADE'          SS
SS      AUTHOR: LOUIS ROY      SS
SS      COPYRIGHT (C) 1982     SS
SS SOFTSIDE PUBLICATIONS, INC SS
SS                             SS
SS SS SS SS SS SS SS SS SS SS SS
```

Initialization.

```
5  LOMEM: 16384: HIMEM: 24575
10  TEXT : HOME : POKE 230,32: CALL
    62450
15  GOSUB 3000
20  GOSUB 1000
25  GOSUB 2000
```

Initialization of hi-res screen.

```
30  POKE  - 16368,0: POKE  - 1630
    2,0: POKE  - 16304,0: CALL 6
    2450
35  HCOLOR= 2: HPLOT 0,0 TO 278,0
    : HPLOT 0,180 TO 278,180
40  HCOLOR= C(0): DRAW S(0) + 2 AT
    71,184: HCOLOR= C(1): DRAW S
    (1) + 2 AT 199,184
45  HCOLOR= 3: DRAW 12 AT 116,181
    : DRAW 13 AT 104,187
```

Initialization of players' variables (re-done after each collision).

```
50 X(0) = 10:X(1) = 24:Y(0) = 15:
    Y(1) = 15:D(0) = 1:D(1) = 1:
    A(0) = 0:A(1) = 0
55  HCOLOR= C(0): DRAW 1 AT X(0) *
    8 + 1,(Y(0) - 1) * 6 + 1: HCOLOR=
    C(1): DRAW 1 AT X(1) * 8 + 1
    ,(Y(1) - 1) * 6 + 1: HCOLOR=
    C(1)
```

Draw border if border option in effect.

```
60  IF M$ = "N" THEN 75
65  HCOLOR= 2: FOR X = 0 TO 34:XY
    (X,0) = 1:XY(X,29) = 1: DRAW
    1 AT X * 8,1: DRAW 1 AT 272 -
    X * 8,29 * 6 + 1: NEXT
70  FOR Y = 1 TO 28:XY(0,Y) = 1:X
    Y(34,Y) = 1: DRAW 1 AT 2,Y *
    6 + 1: DRAW 1 AT 34 * 8,175 -
    Y * 6: NEXT
```

Display countdown before play begins.

```
80  FOR I = 5 TO 2 STEP  - 1: HCOLOR=
    6: DRAW I AT 134,82: FOR A =
    40 TO 2 STEP  - 2: POKE 0,A:
     POKE 1,3: CALL 768: NEXT : FOR
    A = 0 TO 290: NEXT : HCOLOR=
    0: DRAW I AT 134,82: FOR A =
    0 TO 90: NEXT : NEXT
85  POKE 0,255: POKE 1,110: CALL
    768
```

Loop to move each player in succession.

```
100  FOR I = 0 TO 1
```

Update coordinates according to direction of movement.

```
110  ON D(I) GOTO 111,113,115,117
111 Y(I) = Y(I) - 1: IF Y(I) =  -
    1 THEN Y(I) = 29
112  GOTO 120
113 X(I) = X(I) + 1: IF X(I) = 35
     THEN X(I) = 0
114  GOTO 120
115 Y(I) = Y(I) + 1: IF Y(I) = 30
     THEN Y(I) = 0
116  GOTO 120
117 X(I) = X(I) - 1: IF X(I) =  -
    1 THEN X(I) = 34
```

Draw player at new position.

```
120  HCOLOR= C(I): DRAW 1 AT X(I)
     * 8 + 1,Y(I) * 6 + 1:A(I) =
    XY(X(I),Y(I)):XY(X(I),Y(I)) =
    1
```

Change player's direction if paddle button pressed.

```
125  IF  PEEK ( - 16287 + I) < 12
    8 THEN Z(I) = 0: GOTO 145
130  IF Z(I) = 1 THEN 145
135 Z(I) = 1:A =  PDL (I): IF A <
    128 THEN D(I) = D(I) - 1: IF
    D(I) = 0 THEN D(I) = 4
140  IF A > 127 THEN D(I) = D(I) +
    1: IF D(I) = 5 THEN D(I) = 1
```

If no collision, game continues.

```
145  NEXT : IF A(0) = 0 AND A(1) =
    0 THEN  POKE 0,200: POKE 1,4
    : CALL 768: GOTO 100
```

Set collision flags for both players if they collide with each other.

```
200  IF X(0) = X(1) AND Y(0) = Y(
     1) THEN A(0) = 1:A(1) = 1
```

Flash players and scores if collision has occurred.

```
210  FOR I = 0 TO 13:X = (1 - ( -
     1) ^ I) / 2
220  IF A(0) = 1 THEN  HCOLOR= X *
     C(0): DRAW 1 AT X(0) * 8 + 1
     ,Y(0) * 6 + 1: DRAW S(0) + 2
      AT 71,184
230  IF A(1) = 1 THEN  HCOLOR= X *
     C(1): DRAW 1 AT X(1) * 8 + 1
     ,Y(1) * 6 + 1: DRAW S(1) + 2
      AT 199,184
240  IF X = 0 THEN  FOR A = 0 TO
     10:B =  PEEK ( - 16336) +  PEEK
     ( - 16336) -  PEEK ( - 16336
     ) +  PEEK ( - 16336): NEXT
250  FOR A = 0 TO 25: NEXT
260  NEXT
```

Re-initialize playfield matrix.

```
300  FOR X = 0 TO 34: FOR Y = 0 TO
     29:XY(X,Y) = 0: NEXT : NEXT
```

Increment scores.

```
310  IF A(0) = 1 THEN  HCOLOR= 0:
      DRAW S(1) + 2 AT 199,184:S(
     1) = S(1) + 1: HCOLOR= C(1):
      DRAW S(1) + 2 AT 199,184
320  IF A(1) = 1 THEN  HCOLOR= 0:
      DRAW S(0) + 2 AT 71,184:S(0
     ) = S(0) + 1: HCOLOR= C(0): DRAW
     S(0) + 2 AT 71,184
```

Jump to end sequence if game over.

```
330  IF S(0) = 9 OR S(1) = 9 THEN
     500
```

If game not over, erase screen and go to line 50.

```
340  POKE 0,30: POKE 1,80: CALL 7
     68
350  HCOLOR= 0: FOR I = 0 TO 89: HPLOT
     I,I + 1 TO 277 - I,I TO 277 -
     1,179 - I TO I,179 - I TO I,
     I: NEXT
360  GOTO 50
```

End of game sound effect.

```
500  FOR I = 0 TO 3
510  POKE 0,60: POKE 1,90: CALL 7
     68: POKE 0,130: POKE 1,90: CALL
     768
520  NEXT
```

Clear variables, invert hi-res screen.

```
530  CLEAR : POKE  - 16368,0: CALL
     804
```

Slowly alternate between hi-res pages, displaying normal and inverted screens. Stop when a key is pressed.

```
540  POKE  - 16300,0: GOSUB 600: POKE
      - 16299,0: GOSUB 600
550  POKE 0, PEEK ( - 16384): IF
      PEEK (0) < 128 THEN 540
```

If ESC, exit program, otherwise run program again.

```
560  POKE  - 16368,0: IF  PEEK (0
     ) = 155 THEN  TEXT : HOME : PRINT
     : PRINT "BYE !!"; CHR$ (7); CHR$
     (7): END
570  RUN
```

Delay.

```
600  POKE 0,0
610  POKE 0, PEEK (0) + 1: IF  PEEK
     (0) <  > 180 THEN 610
```

Initialization.

```
1000  DIM XY(34,29): DIM A$(100)
1010  POKE  - 16301,0: POKE  - 16
     300,0: POKE  - 16297,0: POKE
      - 16304,0
1020  POKE 232,0: POKE 233,30: ROT=
     0: SCALE= 1
1030 C(0) = 1:C(1) = 5:S(0) = 0:S
     (1) = 0
```

Create title page.

```
1100 K = 47:Y = 35: HCOLOR= 5: GOSUB
     1130:K = 119:Y = 90: HCOLOR=
     1: GOSUB 1130
1120  GOTO 1200
1130 X = K: READ A$: FOR U = 1 TO
      LEN (A$) STEP 2:A =  VAL ( MID$
     (A$,U,1)):B =  VAL ( MID$ (A
     $,U + 1,1))
1140  FOR I = 1 TO A: HPLOT X,Y TO
      X + 2,Y TO X + 2,Y + 3 TO X,
      Y + 3 TO X,Y:X = X + 4: POKE
     0,X - 25: POKE 1,3: CALL 768
     : NEXT
1150  IF B = 0 THEN Y = Y + 5:X =
     K: GOTO 1170
1160 X = X + 4 * B
1170  NEXT : RETURN
```

Draw author's name.

```
1200 X = 110:Y = 140: HCOLOR= 3: GOSUB
     1250: HPLOT 128,141:X = 147:
     Y = 140: GOSUB 1250
1210  RETURN
1250  READ A$: FOR I = 1 TO  LEN
     (A$) STEP 2:A =  VAL ( MID$
     (A$,I,1)):B =  VAL ( MID$ (A
     $,I + 1,1))
1260  HPLOT X,Y:X = X + (A - 2):Y
      = Y + (B - 2): POKE 0,3: POKE
     1,3: CALL 768: NEXT : RETURN
```

Instructions.

```
2000  VTAB 22: HTAB 1: PRINT "DO
     YOU WANT INSTRUCTIONS (Y/N)?
      ";: GET A$: IF A$ <  > "Y" AND
     A$ <  > "N" THEN 2000
2010  PRINT A$: IF A$ = "N" THEN
     2200
2020  TEXT : HOME : FLASH : HTAB
     12: PRINT "<<<  BLOCKADE  >>
     >": NORMAL : HTAB 15: PRINT
     "BY LOUIS ROY"
2030  PRINT : PRINT "   THIS IS T
     HE FAMOUS 'BLOCKADE' GAME
     FOR 2 PLAYERS."
2040  PRINT : PRINT "   EACH PLAY
     ER MOVES ON THE SCREEN,
     LEAVING A WALL BEHIND HIM."
2050  PRINT : PRINT "   THE OBJEC
     T OF THE GAME IS TO FORCE
     YOUR OPPONENT INTO A WALL, E
     ITHER YOURS OR HIS."
2060  PRINT : PRINT "   YOU CAN O
     NLY TURN 90 DEGREES AT A
     TIME."
2070  PRINT : PRINT "   TO TURN R
     IGHT, POSITION YOUR PADDLE
     TO THE RIGHT AND PRESS THE B
     UTTON."
2080  PRINT : PRINT "   TURNING L
     EFT IS SIMILAR EXCEPT YOU
     POSITION YOUR PADDLE TO THE
     LEFT."
2090  VTAB 24: HTAB 8: INVERSE : PRINT
     "(PRESS A KEY TO CONTINUE)";
     : NORMAL : GET A$
2100  POKE 34,2: HOME : POKE 34,0
2110  VTAB 4: PRINT "   REMEMBER
     THAT LEFT AND RIGHT ARE
     REVERSED WHEN YOU ARE MOVING
      DOWN."
2120  PRINT : PRINT "   YOU SCORE
      A POINT WHEN YOUR OPPONENT
     RUNS INTO A WALL."
2130  PRINT : PRINT "   THE FIRST
      PLAYER TO GET TO 9 POINTS
     WINS !!!"
2140  PRINT : PRINT "   YOU CAN P
     LAY WITH OR WITHOUT A BORDER
     AROUND THE SCREEN. IF THERE
```

```
      IS NO       BORDER, A PLAYER
       WHO EXITS THE PLAYFIELDOFF
      ONE SIDE WILL REAPPEAR ON TH
      E OTHER SIDE."
 2200  VTAB 23: HTAB 1: PRINT "DO
      YOU WANT A BORDER (Y/N)? ";:
       GET M$: IF M$ < > "Y" AND
      M$ < > "N" THEN 2200
 2210  PRINT M$: RETURN
```

Poke in shape tables.

```
3000  FOR I = 7680 TO 7995: READ
     A: POKE I,A: NEXT
```

Poke in sound routine.

```
3010  FOR I = 768 TO 803: READ A:
      POKE I,A: NEXT
```

Poke in hi-res screen inversion routine.

```
3020  FOR I = 804 TO 838: READ A:
      POKE I,A: NEXT
```

Shape data.

```
3040  DATA 13,0,28,0,40,0,56,0,65
     ,0,79,0,93,0,107,0,124,0,139
     ,0,149,0,165,0,181,0,243,0,1
     3,13,254,31,110,9,254,31
3050  DATA 110,13,7,0,50,54,118,1
     05,13,76,36,36,220,31,31,150
     ,98,97,49,0,137,76,54,54,246
     ,107,13,5,0,98,105,13,141,24
     6,251,159
3060  DATA 159,110,13,13,13,4,0,1
     3,13,13,13,254,250,106,141,2
     46,255,31,220,4,0,73,73,54,1
     18,177,27,36,31,31,31,100,97
     ,49,0,13
3070  DATA 13,13,13,214,251,31,31
     ,180,18,85,13,13,76,36,4,0,1
     8,54,118,105,13,76,228,251,3
     1,4,96,105,13,4,0,13,13,13,1
     3,254
3080  DATA 250,250,50,38,0,50,22,
     118,105,13,76,36,32,220,31,3
     1,150,13,13,5,0,18,100,105,1
     3,141,54,254,186,251,31,12,6
     4,13,13,5
3090  DATA 0,54,54,45,12,28,39,40
     ,21,76,49,54,46,45,77,41,12,
     36,28,191,54,77,73,36,12,45,
     222,146,45,77,36,36,77,30,23
     ,21
3100  DATA 21,77,36,36,45,53,254,
     173,110,33,36,44,245,146,101
     ,36,76,41,45,22,63,39,22,46,
     45,4,0,54,54,45,12,28,39,40,
     21,76
3110  DATA 49,150,12,44,12,108,73
     ,73,49,54,46,45,77,41,12,36,
     28,191,54,77,73,36,180,82,10
     1,36,108,105,55,54,23,45,77,
     45,12,28
3120  DATA 231,12,45,77,73,73,54,
     54,77,28,60,4,40,21,77,54,14
     ,101,36,220,109,9,182,98,44,
     12,52,0
```

Machine Language subroutine data.

```
3130  DATA 165,1,133,3,166,0,160,
     1,132,2,173,48,192,136,208,4
     ,198,1,240,7,202,208,246,166
     ,0,208,239,165,3,133,1,198,2
     ,208,241,96
3140  DATA 169,32,133,1,169,64,13
     3,3,169,0,133,0,133,2,168,17
     7,0,73,255,145,2,200,208,247
     ,230,1,230,3,165,1,201,64,20
     8,237,96
```

Title page data.

```
3200  DATA "331624321212423340121
     215121215111312121212033151
     212152442121230121215121215
     1131212121210334324321212121
     23340"
3210  DATA "10103411101214103320"
3220  DATA "333332323231211112132
     323233323332313121211313232
     323232132333323121113332322
     123233332312121233333312121
     2
     333333231312132332313123232
     3
     2313122"
3230  DATA "232323232323131221212
     121212121312131323232332313
     1
     312323332323233312131132333
     3
     231211133323123233332312121
     2
     323232313231313113131323132
     3
     1313222"
```

APPLE™ SWAT TABLE FOR: **BLOCKADE**

LINES	SWAT CODE	LENGTH
5 - 60	HC	374
65 - 115	US	409
116 - 230	NK	469
240 - 510	US	401
520 - 1020	BS	287
1030 - 1260	EW	412
2000 - 2070	LY	529
2080 - 2140	IQ	535
2200 - 3060	FU	512
3070 - 3110	HN	590
3120 - 3220	LR	628
3230 - 3230	YJ	148

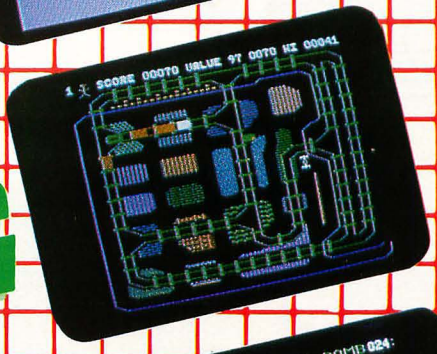
FUNTASTIC!

DATAMOST
9748 Cozycroft Ave.,
Chatsworth, CA 91311
(213) 709-1202

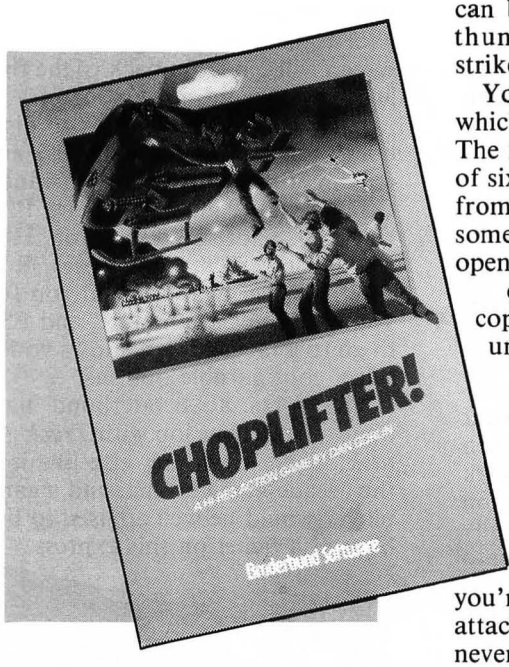
APPLE™ STOCKING STUFFERS

Reviewed by Hartley G. Lesser

Apple owners have gathered around their computers for several Christmases, now. This year, Broderbund Software has three very special offerings, *Choplifter!*, *Track Attack* and *Star Blazer*. These games bring truly unique features to the Apple. When the family picks up their joysticks this year, be prepared for Hi-Res dazzlement and action exciting enough to make the tree wobble.



*For reviews of other products available for the Apple™ see pages 68 and 90.



Choplifter!

by Dan Gorlin (Broderbund Software, Inc., 1938 Fourth Street, San Rafael, CA 94901). System requirements: Apple II™ or Apple II Plus with 48K and DOS 3.2 or 3.3. Requires joystick with two buttons. Suggested retail price: $34.95. Also available for the Atari® 400/800 with 48K and joystick.

WANTED: HEROES...desperately! The Bungeling Empire (no more than an association of cutthroats, kidnappers and thieves) has seized, and is holding for ransom, sixty-four extremely important hostages. They must be rescued, regardless of the cost...in money, equipment, or lives.

No easy chore, this. First and foremost is the task of learning to fly your helicopter from a rescue center, across the barbed wire border and into enemy air space. The joystick admirably assists in this assignment. Superb Hi-Res graphics present your position on the screen. But quickly get used to using Button (1) as your direction command, or you'll find yourself facing in the incorrect direction to handle a multiple tank attack on your ship.

Button (0) on your joystick is the fire button, and must be depressed each time you want to fire a round at the enemy. A machine-gun effect can be attained by limbering one's thumb for rapid and repeated strikes on Button (0).

You are given three sorties within which to retrieve all of the hostages. The first sortie finds the first group of sixteen hostages scrambling away from their barracks, which have, for some unknown reason, been blown open, allowing the prisoners to escape. You must land your helicopter as close to the running figures as possible. Be careful. One slight miscalculation with the joystick and you will have one less hostage to worry about. After you've landed next to the open barracks, you'll see green tanks approaching your position. Without warning, they'll fire on your ship. If you're still grounded when the tanks attack, there's a good chance you'll never lift off again.

Once your chopper has rescued sixteen hostages, no more can make entry. If hostages remain, they will wave goodbye to you as you return to home base to deposit the now free and grateful POW's. They will also wave their thanks to you as you return for more of their kin.

After a successful rescue, and on your return into enemy territory, you'll find all of the other barracks locked tight. What to do? Position your helicopter as close as you can to the barracks and wait for the nearest tank to eliminate the problem. With luck, you can maneuver the tank to fire at your ship, missing you and striking the front of the barracks. The hostages can now run outside. However, get rid of that tank after it has accomplished the barracks opening. If you don't, these sixteen hostages will never make it aboard your helicopter.

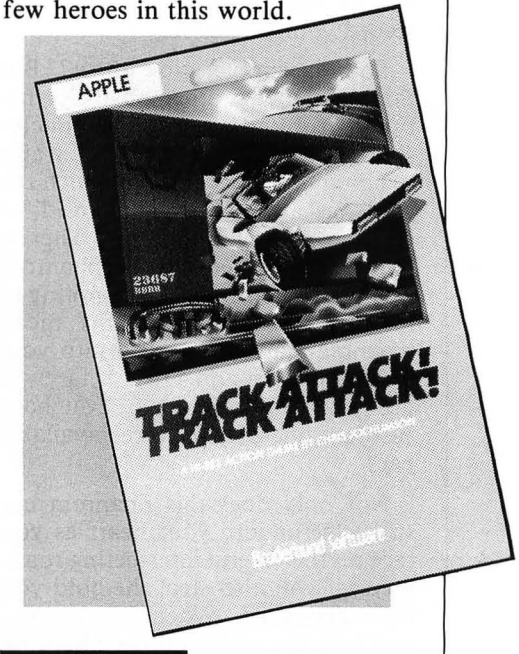
Your count and amount are tallied in the upper portion of the screen. A red lemon on the left side indicates how many hostages have been killed (by either enemy or friendly action), a blue lemon in the middle reveals the number of hostages in your helicopter, and a green lemon on the right side shows how many hostages have been safely returned home.

As you cross the border again on the second sortie, a jet comes screeching out of the sky. Superb three-dimensional graphics capture the jet as it banks and roars in to fire air-to-air missiles. With correct joystick manipulation, you can shoot the jet down and also avoid the missiles, but don't count on being so lucky your first trip out!

The third trip is even worse! Not only do you have the jets, tanks, and those air-to-air missiles, but you must also contend with drone air mines. These follow you, like puppies, no matter where you go. They must be eradicated by gunfire from your ship. They are the only Bungeling weapons that will cross the boundary and attack you at home base. So, once viewed, they had best be destroyed. There's nowhere you can escape from them.

If, in the heat of battle, sweat blinds you momentarily, strike the ESC key for play pause. Should the games's sound only further fray your nerves, a CTRL-S will toggle the noise off. Should you prefer a reverse attitude for your joystick, CTRL-V and CTRL-A reverse up and down, left and right.

There's very little left to add, except that after playing *Choplifter!,* you'll understand why there are so few heroes in this world.



Track Attack

by Chris Jochumson (Broderbund Software, Inc.) System requirements: Apple II™ with 48K memory, Applesoft or Integer. Suggeted retail price: $29.95. Also available, on disk only, for the Atari 400/800 with 32K.

Tired of the same old shoot-em-up arcade games? Discontented with

adventure text games that leave you bored, gored and floored in some never-ending maze? May I suggest a change of pace?

Track Attack could well become your favorite non-typical arcade game. Broderbund, and author Chris Jochumson, have put together an unusual chase and treasure game that revolves around a train.

What could the object of a game be that uses a train? First and foremost, it's enjoyment. The game player operates a car, which must remove gold pieces from the train. Ahh, but the train is moving all over the place — on sidings, on main rails, side rails, and through a train station. All is displayed fantastically with superb high-resolution graphics. You control your car via the joystick or the keyboard, and try to intercept the puffer-belly's freight car as it races through railroad crossings. If you manage to drive through a sidecar, your vehicle's color turns to gold, and you must race headlong to a specific corner of the screen to deposit and record your treasure.

Sounds simple, doesn't it? But there are a few other obstacles between you and a successful conclusion. You could: a) run into a train; b) run into a sidecar from which the gold has already been removed; c) run into the phantom car. There's an irritating green vehicle (phantom car) that is continually attempting to intercept you. Should this mean machine manage to crash into you, a cloud of dust and a hearty farewell are your only epitaphs. You have just lost one of your three available cars.

Not only does this phantom car strike terror into your heart as you race on the several intersecting roadways, it will also steal the gold you have deposited. When this occurs, you turn the tables on the phantom car and chase it. Should you make contact with this vehicle, you retake your gold, and hightail it to the recording area.

The value of the gold you grab is determined by the amount of time necessary for you to accomplish the theft after either starting the game, or after the first piece has already been stolen. Each gold piece has a starting value of 100, which decreases each second after logging your merchandise. A tally display at the top of the screen will remind you of the number of your remaining players, your current score, the value of the next piece of gold captured, the value of all of your captured gold so far, and the high score for the day.

After you have successfully gained at least one piece of gold, you have the opportunity of advancing to Level Two. Level Two is only accessible by positioning your vehicle on the left side of the screen next to the moving train. You must then press the spacebar or the joystick button (depending on the version you're playing). This is actually your jump from the car and onto the train. If you manage this feat, you'll find yourself on top of the train.

Animation is accomplished for Level Two through use of Broderbund Software's *Arcade Machine*, and is truly worth the price of the software. From your position atop a boxcar, you must run forward, which means jumping up or down onto different level freight cars. This is accomplished through use of keys A or Z, or directing the joystick in the appropriate direction.

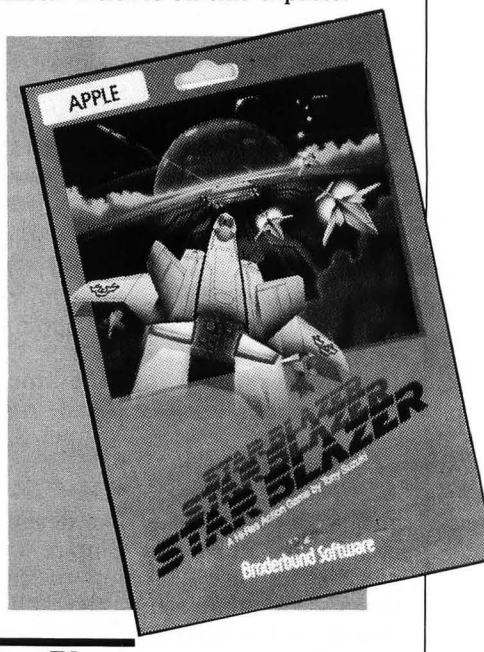
Why jump onto the train? To take control of it, of course. The final jump is a dilly. Should you manage it, the computer takes control and puts you in the driver's seat, just like an old Hertz commercial. Should you fail in this quest, you'll be transported back to Level One, minus one man and all of your accumulated gold.

Once you control the train, you'll arrive at Level Three. A watchman's car will pursue the train, and your goal will be to run over eleven gold pieces scattered about the tracks. These golden rewards are each worth the total amount of gold you collected in Level One. Oh, I forgot to tell you that the watchman's vehicle is the phantom car. Should you run into the phantom car, you'll lose all of your collected gold and all of the points gained in the current level. If you manage to collect all of the gold on the tracks, you will earn an extra man.

Many will prefer to leave Level Three after only the briefest of stays. Pressing the spacebar while your train is on the right side of the screen will send you back to the first level, points and players intact.

High scores can be saved to disk, but this means notching the disk and voiding the Broderbund warranty. Other keys of importance are P to halt play, S for speed selection, H to reveal the current high scores, R to begin all over again should you feel you didn't get a fair start, and ESC to go to a menu which allows you to select configurable options.

The folks at Broderbund have done an excellent job with *Track Attack*. There's nothing else like it in the arcade-game league, and a game player would be well advised to buy himself a ticket on this express.



Star Blazer

by Tony Suzuki (Broderbund Software, Inc.) System requirements: 48K Apple II™ Integer or Plus, DOS 3.3 or 3.2. Joystick optional. Suggested retail price: $31.95. Atari conversion is coded by Solitaire Group and requires 48K Atari 400/800.

O.K. I'm about as ready as I can be — joystick grasped firmly with the right hand and eyes fixed on the title page. With a tap of the spacebar for keyboard play mode, or a push of my joystick button for joystick mode, the game begins.

Quality hi-resolution graphics are presented on the screen. I'm flying a jet, which at low altitude drops

continued on page 116

C&H Best Sellers: Programs That Work!

Higher Text Extended

$22.50

(+$2.00 shipping)

The best just got better! Synergistic software produces an excellent program called "Higher Text II". It's our favorite program for adding text to a hi-res picture, or for producing textual display pic's when the standard Apple text just isn't sufficient. The problem is that Higher Text by itself is not particularly easy to use...necessity being the mother of invention, we set out to enhance the program. We produced some lightning-fast machine language routines to allow any apple user to easily and quickly take advantage of what the program has to offer. What we ended up with is the best and most flexible program of its kind at any price! Higher Text Extended, in conjunction with Higher Text II, allows you to add text to an existing pic or design textual display pics. Use all of the features of the original Higher Text II but in a very easy "user-friendly" environment. Provides instantaneous center, left, or right justification of text on a line-by-line basis. Prevents you from accidentally scrolling or destroying your pic. Presents you with several "menus" which list all of the available options...including: (1)List of all key commands, (2)List of background modes, (3)List of background colors, (4)List of character colors and (5)Selection of font styles...These menus are available at any time in the development of a pic. Provides temporary storage of your pic as it is being developed with instantaneous recall should you make an error. Incorporates extensive error-trapping to prevent the needless loss of time associated with pressing the wrong key or entering the wrong value. Allows you to change fonts at any time during the creation of a pic. Largely self-prompting with "Help" screens so there's no need for constant reference to the manual or easily misplaced reference cards. Requires 48K Apple II Plus (Applesoft in ROM), 1 or 2 disk drives, (DOS 3.3)

The Perfect Companion to The Slide Show!

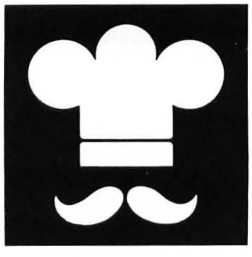
Higher Text II by Synergistic is required! (List price $40.00)

PACKAGE PRICES:

Higher Text Extended + Higher Text II . . . . . $53.00 (Save $ 9.50)
Higher Text Extended + The Slide Show . . . . . $60.00 (Save $12.50)
Higher Text Extended + Higher Text II + Slide Show . . . . $97.00 (Save $15.50)

Higher Text II is a trademark of Synergistic Software

Apple is a trademark of Apple Computer, Inc.



The Menu][

$39.95

(+$2.00 shipping)

The MENU][stores your favorite recipes in any of 6 "user-defined" categories. You "browse" through the alphabetized recipe list in these categories, choosing your menu selections with a simple keystroke. You can write your menu for up to 2 weeks (42 meals); the rest is automatic! You can request a printed copy of your menu, your recipes (in any serving size up to 1,295 people) and an alphabetized shopping list of the ingredients needed to prepare the meals you've selected (a check list of misc. items can be added to the list with the touch of a button). The shopping list automatically combines "like" ingredients and adjusts each recipe for the number of people eating the meal. Recipes and menus can be changed, deleted, or added to at any time. Also included is a Special Meal/Party planner menu option, and 2 "user-defined" counters (calories, sodium intake, etc.) Since you won't be overbuying anymore, "THE MENU][may pay for itself on your very first shopping trip!

Operates with 1 or 2 disk drives (DOS 3.3). Requires 48K Apple, Disk Drive, and Printer. Written in APPLESOFT and MACHINE LANGUAGE.

The Slide Show

$49.95

(+$2.00 shipping)

Finally, high-res pics are more versatile than 35mm slides...and less expensive! The "slide" show allows you to use your high-res pics in a programmed sequence. Not another Graphics Package, but a versatile enhancement to all of the excellent Graphics Packages already available. Think of it as a standard slide show on your television with the advantage of "TV-like" special effect transitions between "slides." 20 different transitions (including: horizontal and vertical wipes, ripple effects, column wipes, etc.). Works with standard Apple pics. Includes the capability for producing a stand-along "Run-Time" package. Hi-res pics (slides) load in approximately 4 seconds. Supports single or multiple disc drives. Free run, timed, keypress or game button slide advance. User friendly. NOT copy protected! Compatible with NTSC TV signal processing hardware. Prints slide/transition listing. Written in Applesoft/Machine Language, 48K, 1 or more disk drives, printer optional. (DOS 3.3)

PERFECT FOR:

- Educators
- Salespeople
- Businesspeople
- Executives
- Cable or closed circuit TV nets
- Free running store displays
- Lectures
- Presentations
- Exhibits
- Broadcast Television

See your favorite APPLE dealer or order direct. Send check or money order to:

C & H VIDEO
110 W. Caracas Ave.
Hershey, PA 17033

DEALER INQUIRIES WELCOME
PA Res. Add 6% sales tax
717-533-8480
Between 9am and 9pm

VISA®

master charge THE INTERBANK CARD

bombs, and at higher altitudes fires pulse cannons. My first mission is to destroy the radar. I note that I'm flying at high altitude and glance at the passing scenery below. Homes, micro-wave antennas, fuel depots and other miscellaneous edifices await destruction. But I don't see a radar unit — yet.

Time, then, to acquire the skills necessary to fly the jet and to garner precious points. Fuel consumption and remaining-bombs gauges are located at the top of the screen. Having started with 3000 gallons of fuel and 30 bombs, I'm not quite sure how careful I must be with my supplies.

By pulling down on the joystick, my jet descends to bombing level. A push on the joystick button, and bombs are released from the jet. By manipulating the jet's speed, I cause the bombs to arc forward and strike objects otherwise impossible to hit directly from above. This is indeed the case with the antenna towers, for they must be avoided and flown over by my jet. Unfortunately, this requires a height that negates bombing. So, by increasing speed while approaching an antenna tower, and suddenly pulling back on the joystick, I can cause my bomb to pitch forward arc upward, and deposit itself on the enemy tower. I must also allow ample time to fly above the tower. This takes practice, and I find myself, on many occasions, showering the landscape with my vaporized bits and pieces.

"Looks like a giant condor's snagged my fuel... Well, no use moaning over gobbled gas."

Every so often, a supply plane flies across the top of the screen, and parachutes a package meant for my jet's use. Failing to snag this supply chute results in the contents falling to earth and exploding. This is usually followed by my jet falling to earth and exploding — out of fuel. Unfortunately, no other method exists for refueling and rearming my jet.

With a careful eye on the display that indicates the number of jets remaining, (I started with three.) I finally learn how best to fly the jet. Drats, fuel almost gone...wait, there's the supply ship...must climb to intercept the chute...what's this! A bird? Looks like a giant condor and it's snagged my fuel.

Repeated cannon blasts aimed at the miscreant buzzard prove fruitless. The bird has stolen my supplies. Well, no use moaning over gobbled gas. By keeping alert, I should be able to spot the cargo plane and power up to get the package before the thieving bird strikes again.

A successful re-supply, and there's the radar, camouflaged between some trees. Bombs away — missed! But wait, this radar unit's out in the open. Lower, lower, strike! Got it!

The screen shows my jet zooming off and a new mission — "Waste The Tank" — is displayed. This

shouldn't be too hard. Tank vs. jet. No contest. New screen, and where did all those enemy jets come from. Blast'em!

I lost my jet. Incoming enemy planes pulverized me. But, I'm getting another chance. My second jet starts in the phase where the previous craft was destroyed. This time, I fly beneath the enemy jets. Here's the tank, rumbling along the desert. Power on, move above the tank, drop the bombs...I don't believe this! The tank's faster than I am. Several games later, I found the method necessary to destroy the tank. But you wouldn't want me to pass this secret information along and ruin the game for you.

The third mission, "Bomb The ICBM," is no picnic, to be sure. Not only do I have to find the ICBM, but avoid the enemy planes constantly releasing sky mines in my direction. If I concentrate too much on avoiding or shooting the mines and planes, I'll miss my target, which appears to be an orange mushroom. Once again, I must be resupplied, or I'm doomed. Believe me, the enemy planes do all in their power to relieve me of my re-supply package.

I manage, somehow, to blast the ICBM, and am faced again with the tank. This mission is entitled "Demolish The Tank, Again," which is a very apt pronouncement. Since I succeeded once before in a battle with the tank, an identical plan of attack should be met with victory. Here I go. The bugger is firing heat-seeking missiles at me! Down I go, splattered all over the desert sands.

"...there's only one thing left to do — start over. This time, I'll watch for the towers, get a headstart on the tank, fly lower than the skymines."

With more care, my third jet strikes true. The tank is no more, thanks to a combination of luck and dexterity with the joystick. The final mission, "Wipe Out Bungeling Headquarters," is revealed. I'm supposed to keep my eyes open for a building that looks like a green and white traffic control tower. But, I'd also better watch out for the enemy planes, the enemy skymines, and the enemy heat-seeking missiles.

I'm burning up a lot of fuel avoiding this, that, and the other thing. The cargo plane's chute doesn't make it to me — it dropped right onto a sky mine. The game is over. I have no more jets left. I've failed. I'd like to meet that Tony Suzuki and tell him a thing or two.

The ESC key allows one to halt play to author one's last will and testament, and CTRL-S toggles the sound on and off, whichever you prefer. I would hate to think of someone trying to play this game without a joystick, but a keyboard mode is provided.

In the meantime, there's only one thing left to do — start over. This time, I'll watch for the towers, get a headstart on the tank, fly lower than the skymines. Yes, this time I'll make it — I hope.

"SOFTALK" REVIEWED HOME FINANCIAL PROGRAMS AND HERE'S WHAT THEY HAD TO SAY...

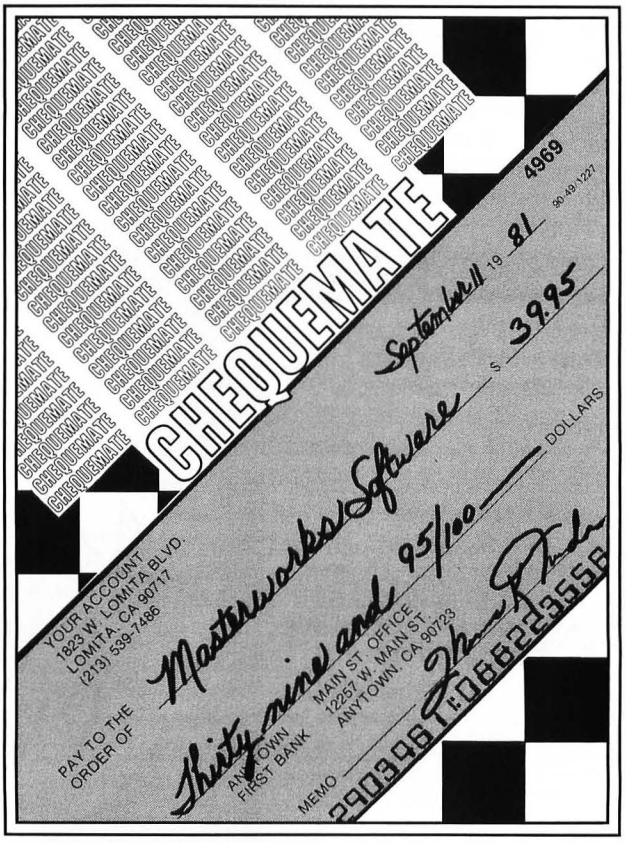
"At $39.95 ChequeMate™ is the Bargain of the Bunch."

Masterworks delivers more **value.** Our ChequeMate was reviewed in SOFTALK as the best bargain in home financial programs.

But as popular as ChequeMate is, Masterworks doesn't stop there. Our full line of home and business software programs offers you one of the most complete packages available. At the most attractive, "best selling" prices.

Here's a shopping list of our innovative programs:

ChequeMate
ChequeMate PLUS
(available for IBM PC & APPLE)
MasterDisk
MasterDOS
PAYMaster
ACCOUNTS Payable
ACCOUNTS Receivable



General LEDGER
ORDER Entry
FIXED Assets
BUSINESS Master

What's so great about Masterworks?

Plenty. Including the fact that with Masterworks Software you can get many of the features of large computer systems, previously unavailable for micros, for your APPLE II and IBM Personal Computer. This is made possible through Masterworks' unique programming system, exclusive to this complete software line.

You want value for your money — you want Masterworks software. For more information or the dealer nearest to you call (213) 539-7486.

Masterworks Software, Inc.

(213)539-7486.

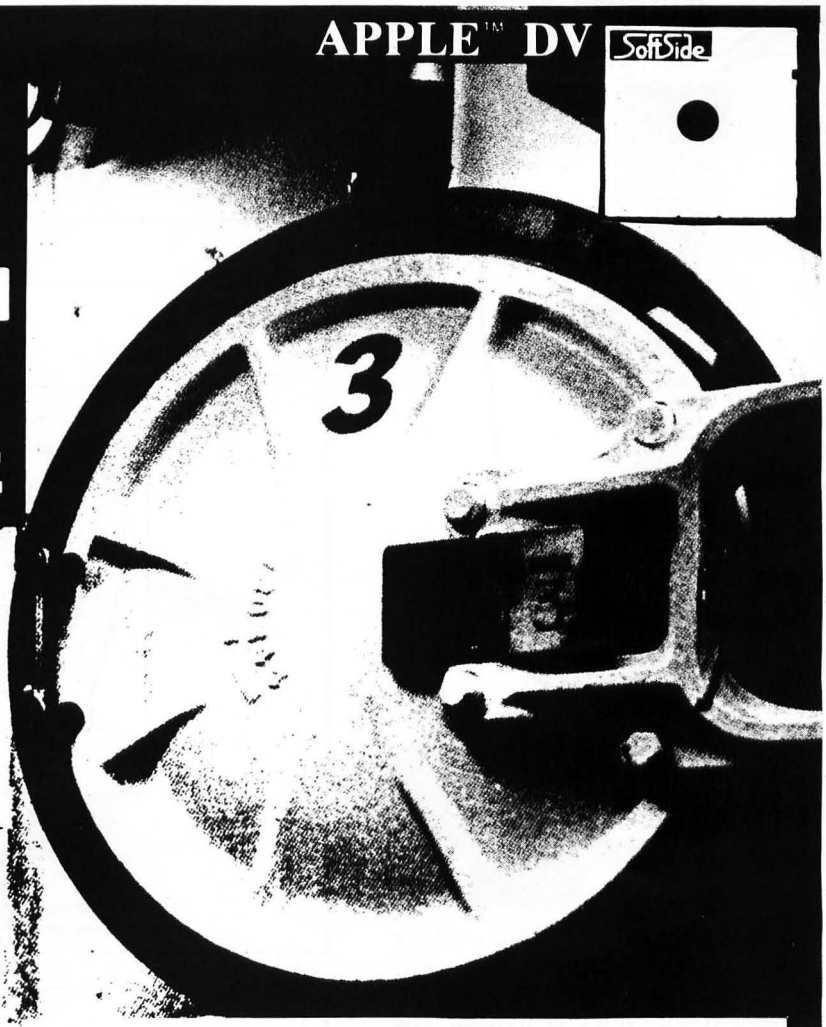
25834 Narbonne Ave.
Lomita, California 90717

master charge THE INTERBANK CARD

VISA

SUB HUNT

by Matt Clark



***Sub Hunt* is an arcade-style game for an Apple II™ with Applesoft, 48K RAM, and game paddles. It is included as the bonus program on Issue 35 Apple DV.**

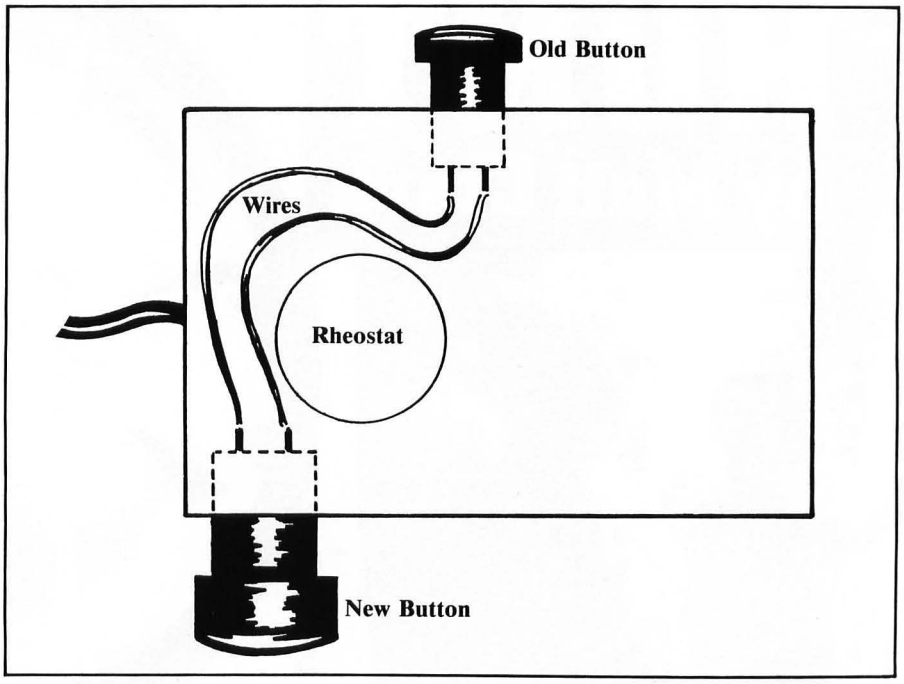
You are the commander of the anti-submarine forces of that guardian of Truth and Justice, namely, *Our Side*. Your mission is to eradicate from the high seas the menacing submarines of that anathema of all good, namely, *The Enemy*. Armed with your supply of depth charges, controls held firmly in hand, you bravely sweep the oceans in fulfillment of your mission.

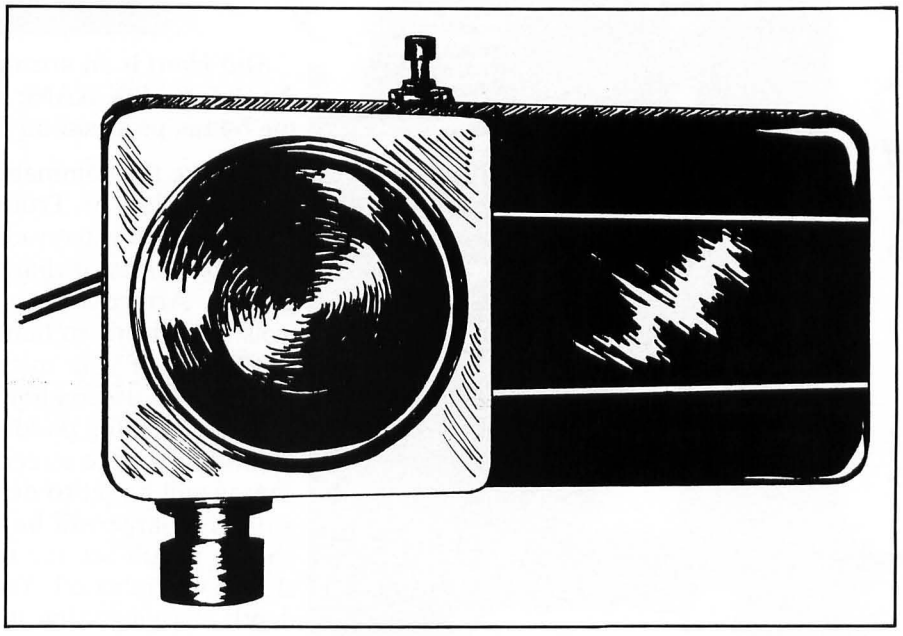
You will use paddle 0 to control the firing of depth charges. Turning paddle 0 will cause a dashed line to move up and down the screen. This line indicates where a depth charge will be set to detonate. When you push the button, a depth charge will be released. If you move the line, the charge is still set for the depth indicated at the time the charge was released. You must try to time the release of the depth charge so that it will explode when one of the submarines passes by. No more than five charges may be active at one time.

The points you receive for destroying a submarine are based on its depth. The subs nearest the surface are worth 5 points, those roaming the very bottom of the ocean are worth 20. Fast subs are worth 10 extra points. Try to get as many points as possible before time runs out.

A Big Button For Your Apple™ Paddle

by Richard Sturtz





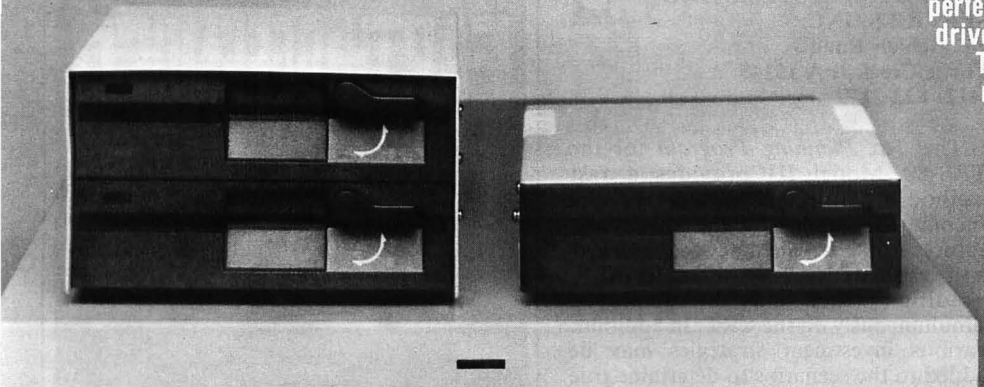
Is your finger sore from pushing the button on your Apple game paddle? Mine was, so I decided to replace it with a larger button. The button I chose is a Radio Shack #275-609. (Not all buttons will fit, so check before you buy one.)

Looking inside your paddle is easy. Just one Phillips head screw holds the back cover on the paddle. (Some units just pull apart.) There's not much room so you must be careful in the placement of the new button. Since I didn't have enough room in the location of the original button, I decided to put the new button on the opposite side (see illustrations). Because I didn't have a drill large enough to accommodate the new button, I had to enlarge the hole with a round file. I recommend using a drill press, as it is easy to drill too deep with a hand electric drill and damage the potentiometer (variable resistor) inside.

Wiring the new button before installation is simple. Two wires run from the original button to the new button. This way, both the new and old buttons will work. Since there is very little room inside, I had to temporarily move the potentiometer to one side in order to install the button. Pull the dial off and remove the nut holding the potentiometer in place. After installing the new button, return the potentiometer and dial to their original locations.

With this new, larger button on the left side of the paddle, I can hold the paddle in my left hand, press the button with my left thumb and turn the dial with my right hand. Left handed people may want to locate the button on the other side. That's all there is to it. Now, when I play *Sneakers*, my finger doesn't give out before I lose all my ships. All I have to do now is get my ships to last longer.

Space Saving Storage



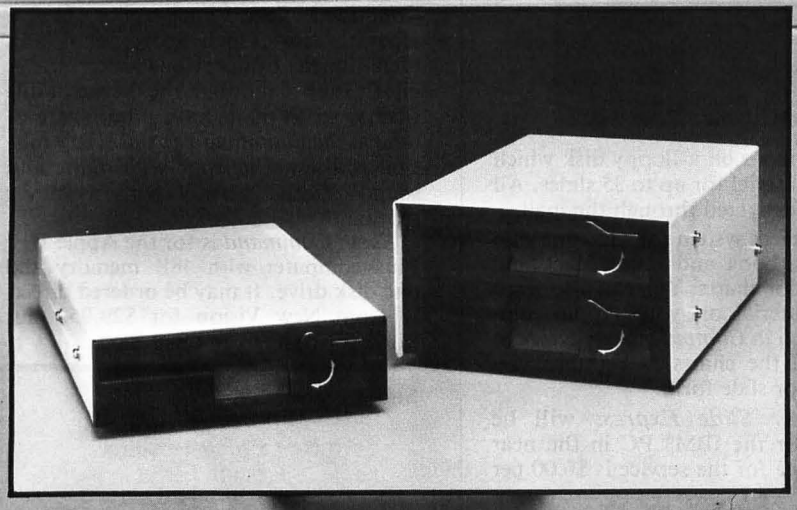
At last, ASAP Computer Products Inc. has created the perfect storage unit for the Apple II. This new data drive slimline out performs any of its competition. This pint-size drive, works 8 times faster than most other drives, saving you space and time. Compare the features of the new ASAP slimline to what you're using now. We're sure you'll agree, this slimline out performs the others.

- 8 times faster than APPLE II Drives.
- Direct drive motor of extremely high quality.
- Only ½ the size of a regular drive.
- 100% APPLE[1] compatible (including "half track").
- Can be used with IBM PC[2] and other computers by simply disconnecting Interface Board.
- Mechanism and read - write electronics made by TEAC the world leader in Audio recording machines.
- 1 (one) full year warranty at no extra cost!!!!!
- The SLIMLINE is a 35/40 track drive and can take advantage of our ENHANCER Diskette which will give the user 15% more storage capacity & up to 163K Bytes.
- Slimline ½ height 163K.
- Dual Slimline drives ½ height in one box.
- Slimline ½ height (1) Megabyte with controller.

Contact your local dealer for more information.
Dealer inquiries invited.

asap computer products, inc.

Manufactured by Professional Data Systems, Tustin, California.



[1]APPLE is TM of APPLE Corporation. [2]IBM is TM of International Business Machines.

asap computer products, inc.
1198 E. Willow St., Signal Hill, CA 90806
Call Toll Free (800) 421-7701
In California (213) 595-6431
or (714) 891-2663
In Canada Call ASAP Computer Products LTD., 116 Viceroy Rd., D-12 Concord
Toronto, Ontario, Canada L4K 1A9
(416) 738-0500 (800) 268-1996

ADVANCED FINANCIAL PLANNING
20922 Paseo Olma
El Toro, CA 92630
(714) 966-5611

Retirement Planning, a new software program for the Atari® 400/800, is a unique financial program. It allows the user to establish a retirement financing plan which takes into account their personal situation relative to inflation, investment returns, retirement income needs, etc. The program then analyzes the plan to determine if changes must be made and offers suggestions as to what these changes should be.

Retirement Planning requires an Atari 400/800 with 32K, disk drive and Atari BASIC cartridge. It is priced at $29.95 and is available from Advanced Financial Planning.

COMPUTER SYSTEMS AND SERVICES, INC.
762 Beulah Road
Turtle Creek, PA 15145
(412) 823-2179

The *Tax Planning Program* for the TRS-80® Model III provides a fast method of projecting personal Federal Income Tax liability for the current and future years. It calculates tax brackets and estimated over- or under-withholding, which may be displayed simultaneously on the CRT. In addition, various investment strategies may be added to the scenarios to determine true after-tax cost of the project as well as ROI. This is also very helpful in determining or evaluating contributions to an IRA.

The Tax Planning Program requires 32K of memory and is available on disk only. Price of the *Tax Planning Program* is $69.95. IBM® PC and Model II versions will be released shortly.

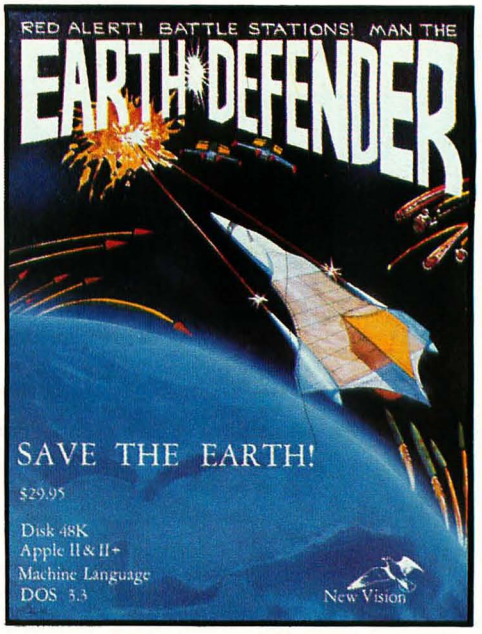
VISUAL HORIZONS
180 Metro Park
Rochester, NY 14623
(716) 424-5300

Computer Slide Express turns an Apple™ computer into an art generating machine. It allows Apple owners to convert computerized charts, designs, graphs and graphics to 35mm color slides, standard size or enlarged, color or black and white prints, or overhead transparencies.

Information can be transmitted over ordinary telephone lines or mailed to Visual Horizons on a floppy disk which can hold material for up to 35 slides. All material is delivered through the mail.

It is a perfect system for someone who is writing a book and wants black and white or color charts. You can also word process material on your Apple computer, punch in *Computer Slide Express*, and get all the charts and graphs you need in color slide form.

Computer Slide Express will be available for the IBM® PC in the near future. Price for the service is $6.00 per slide.



NEW VISION
5105 Peachtree Industrial Blvd.
Chamblee, GA 30341
(404) 455-3688

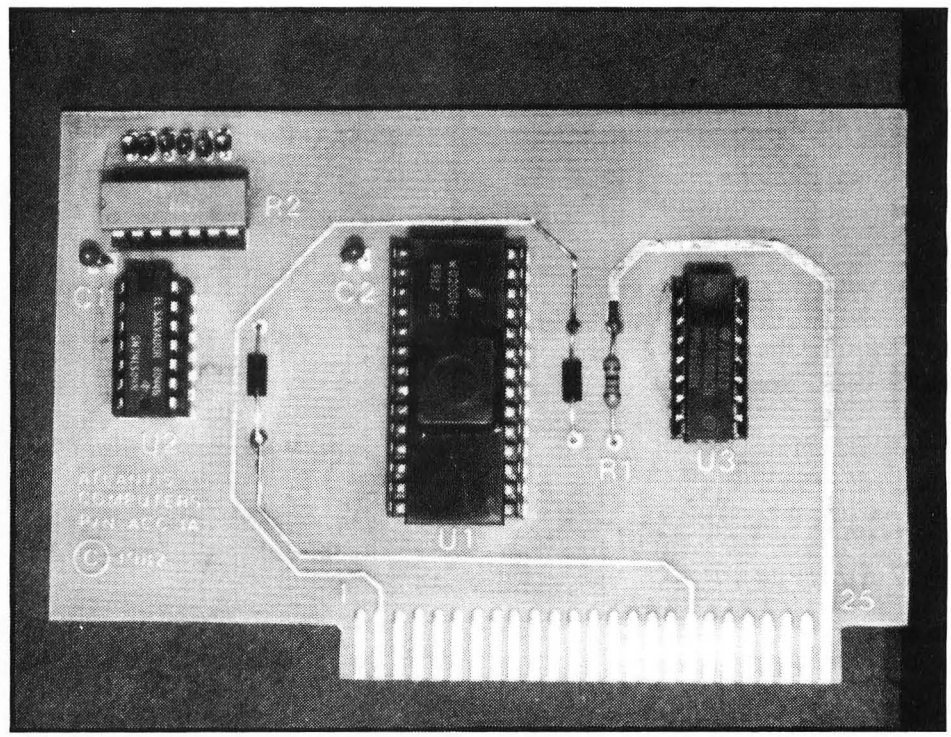
In your orbiting space ship, the Earth Defender, your nuclear powered lasers are ever-ready to save the world. As blue oceans and green continents roll beneath your ship, you soar through 3D space, blasting asteroids from your path. Suddenly, your ship's computer flashes a warning — *killer UFO's ahead*. Aim fast, and fire your wing mounted laser cannons. But be careful! A near-miss triggers a dodging attack and enemy missiles.

The foregoing scenario is from an exciting new game called *Earth Command*. One to eight pilots may join the *Earth Command* fleet. Each pilot has a separate score, high score, and rank (determined by previous success). A pilot's rank determines the degree of difficulty, so all pilots have a fair share of time at their command posts. Every mission is a new challenge with music and sound effects. Enemy patterns never repeat.

Earth Command is for the Apple II™ Plus computer with 48K memory and one disk drive. It may be ordered directly from New Vision for $29.95, plus $1.55 for postage and handling.

Submissions may be sent to:
***SoftSide* Magazine**
New Products Editor
6 South Street
Milford, NH 03055

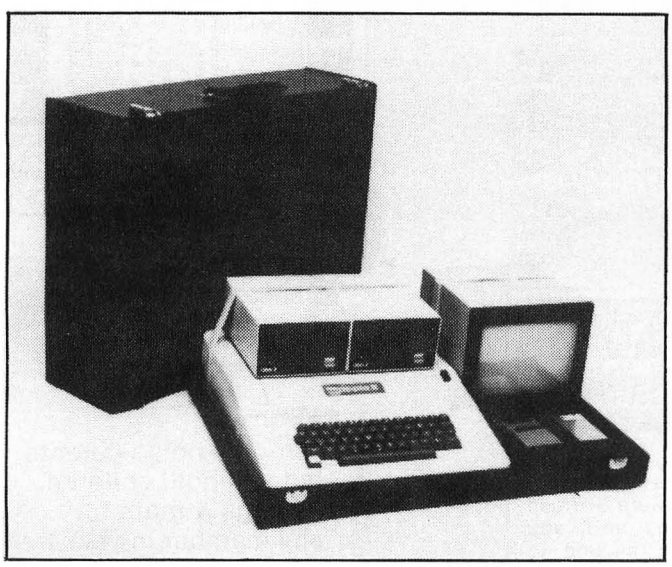
ATLANTIS COMPUTERS
31-14 Broadway
Astoria, NY 11106
(212) 728:6700

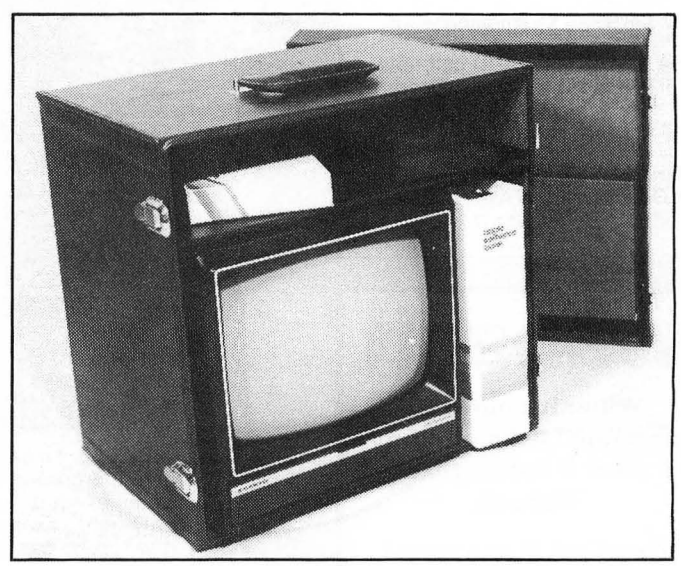


Datalok makes available, to all Apple computer users, the very powerful Data Encryption Standard (DES) Algorithm for complete and absolute data protection. It prevents unauthorized individuals, with physical access to your disks, from understanding and using any information they may contain.

The interactive software provided with the package is user friendly and facilitates encryption and decryption of data by simply responding to a few questions from the terminal. No programming is required on the part of the user for system operation. An illustrative user's manual accompanies the package.

Hardware requirements for *Datalok* include an Apple II with 48K RAM, at least one disk drive, Apple DOS 3.2 or 3.3 and Applesoft BASIC. The *Datalok* System, model ACS-1A, is priced at $349.00 and is available from Atlantis Computers.





COMPUTER CASE COMPANY
5650 Indian Mound Ct.
Colombus, OH 43213
(614) 868-9464

The *Comp Case* has been designed specifically to hold the Apple™ computer system in a fully operational configuration. There is no need to disconnect and reconnect cables each time the computer is moved. The removable top has storage space for manuals, disks, tapes, working papers and other necessities. An elastic strap in the base also provides storage for floppy disks or tapes.

The *Comp Case* facilitates transporting your Apple computer to and from the office, or carrying it as luggage on the airlines. It also provides security and dust protection, as well as temporary storage while your desk is being used for other things.

The *Comp Case* for an Apple II with two disk drives and a nine inch monitor, as shown, has a suggested retail price of $129.00. Cases are also available for the TRS-80®, for printers and other peripheral equipment such as the 12" monitor shown above, which sells for $99.00.

MARKET/SIDE

Advertiser's Index

***SOFTSIDE* PUBLICATIONS**

Advertising Representatives
Sue Rowland
Bob Mackintosh
(603) 673-0585

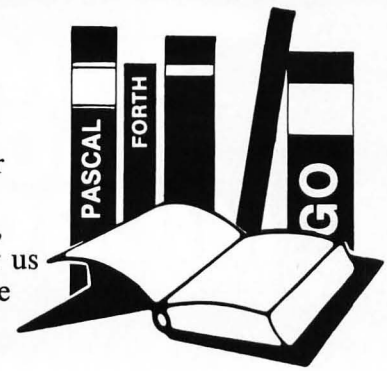
National Representative
Christopher Smith
Inter-Marketing Associates
(603) 827-3976

Associate Publisher
Nancy Lapointe

Publisher
John G. Grow

Next time in SoftSide #36....

Alternatives to BASIC



SoftSide has long been known as "Your BASIC Software Magazine." However, the time has come for us to address some of the other languages your computer can speak.

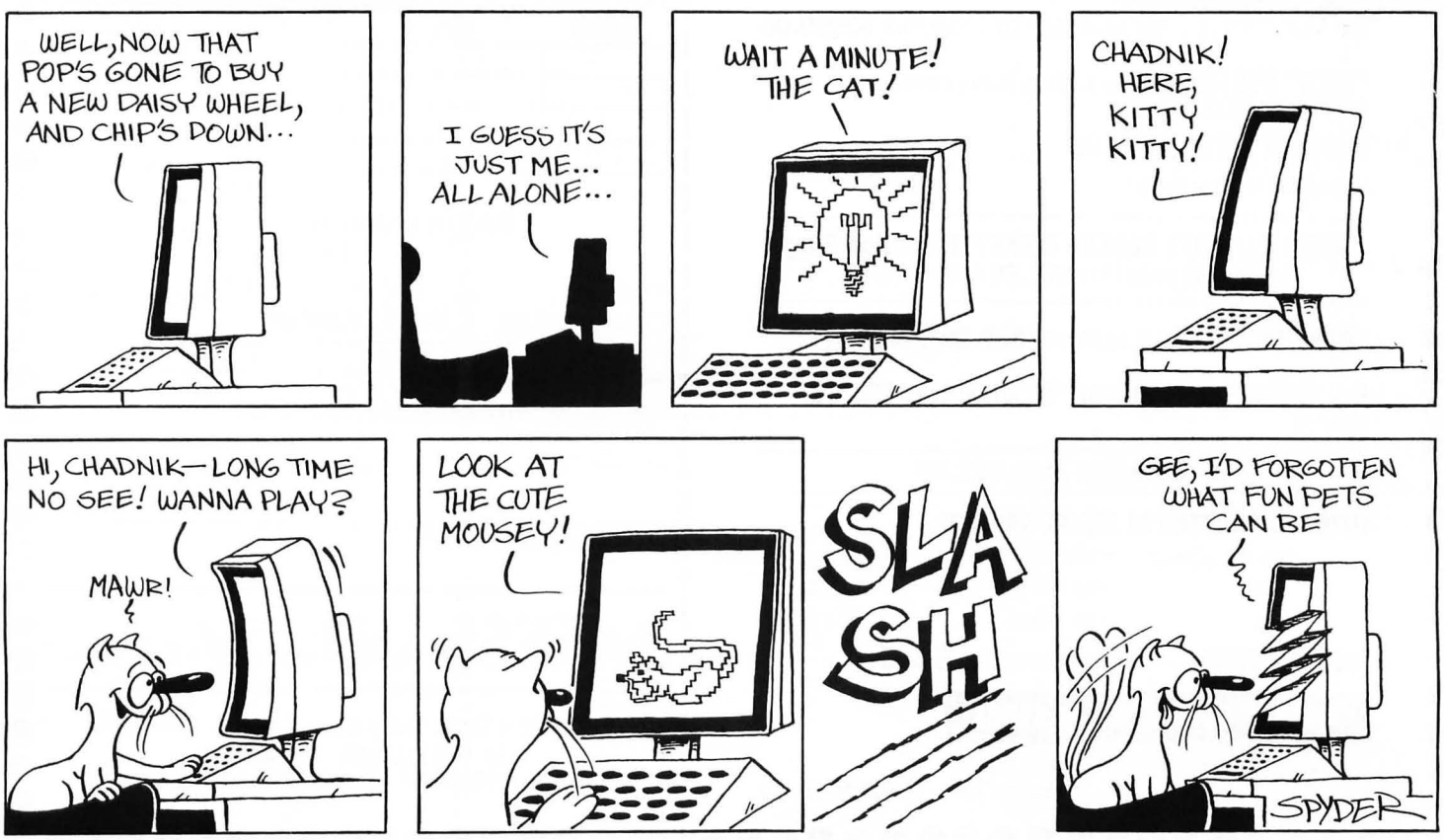
Logo, C, APL, Pascal, FORTH and many other special languages are now available or under development for most microcomputers. We'll discuss some of them in depth and give a brief overview of others. All in all, you should come away with a better understanding of computer languages, in general.

Who knows, you may even learn a few new things about BASIC — its history, its applications and new developments in its implementation.

Ready to teach your computer and yourself a new language? Don't miss *SoftSide* #36 — it could help you decide which dialect to choose!

MACHINE HEAD

BY SPYDER



ANNOUNCING THE NEW ACCESS UNLIMITED MICRO SHOPPING CENTER

OVER $100,000 IN NEW GAMES & BUSINESS SOFTWARE

WE SUPPORT "TI", "ATARI", "COMMODORE", "IBM", "APPLE", "PERCOM", "WANG", "IBEX", "TRS-80"!

MANUFACTURER OVER-PRODUCTION

LESS THAN DISTRIBUTOR COST—BEST BUYS IN U.S.
"IBEX" 64K Business System/Word Processor
Model 2200

- ▶ Hi Resolution Green Phosphor Screen.
- ▶ 2 80KB Ea. Floppy Drives, Includes CP/M* Operating System Basic
- ▶ Interpreter and "C'ITOH" Letter Quality Full Size Printer.

YOU SAVE $3408.00—Reg. Retail $6621.00—Sale Price: **$2960.00**

"IBEX" SYSTEM as above w/Billing Quality instead—
Sale Price: $1938.00.

Equally great buys on up to 11.5 MB.

Expandable Hard Disc System

NEW! 5¼" SOFT SECTOR DISKETTES W/Hub Ring
ATHANA 1 YR WARRANTY **$19.90** SSSD Box Of 10

"BASF" 5¼" LIFETIME WARRANTY **$22.90** SSDD Box of 10

For "WANG" Hard Sector "8" Diskettes $22.90
Box of 10

8" Diskettes From $22.75

OTHER DISKETTES AT EQUAL SAVINGS

Flip 'N File Boxes—Holds 50 Diskettes

5¼"	Reg. $34.95	Sale $18.95
8"	Reg. $49.95	Sale $23.95

LIBRARY CASES—Holds 10 Diskettes. Sale **$2.95** ea.

"IBM PC" COMPUTERS IN STOCK

Bare Drives (without Controller) For "IBM PC"
—Internal or External— **$269.95 ea.**

OVER $1,000,000 IN NEW COMPUTERS, PERIPHERALS AND SUPPLIES!
NEW PRINTERS IN STOCK: "CENTRONICS", "C'ITOH", "OKIDATA", "STAR", "TALLY", "RICOH", "SMITH-CORONA TP", "EPSON".
VALUES TO $3500.00 SALE: $325 to $1695
Call for pricing and fast delivery on other hard-to-get micro peripherals and software.
Top 100 Software programs normally maintained in stock and sold at our special Access Unlimited Micro Shopping Center prices.

" " Reg. Trademarks

☐ Please send me a FREE catalog. I'm not ready to order at this time.
☐ YES, I'm taking advantage of your Sales prices.

Name _____
Company Name _____
Address _____
City _____ State _____ Zip _____
Phone Number (_____) _____

Quantity	Item	Unit Price	Subtotal
		Subtotal	
		State Sales Tax (Texas residents only)	
		handling charge	
		Total	

Check one:
☐ payment enclosed ☐ Visa ☐ MasterCard*

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

*If MasterCard, numbers above name: ☐☐☐☐

Expiration Date: ☐☐ — ☐☐

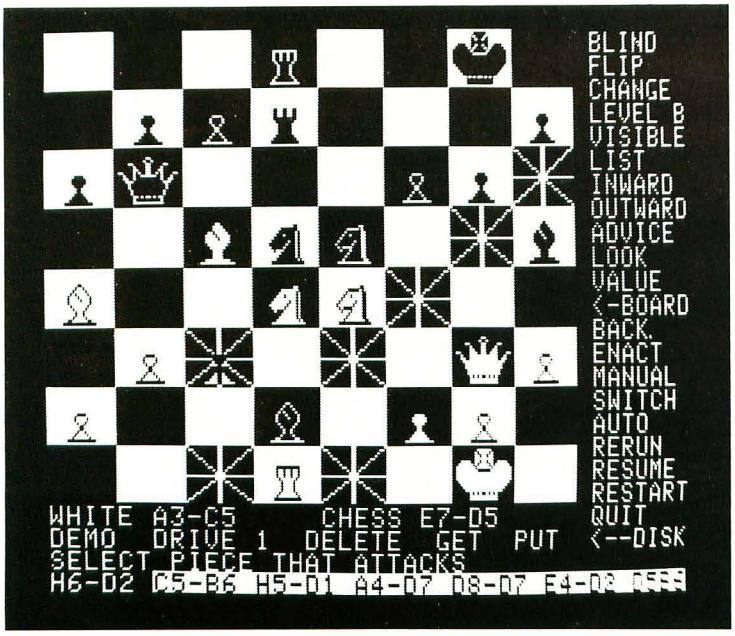
_____
Authorized signature, if charged

ACCESS UNLIMITED

DEPT. D/401 N. Central Expwy. #600/Richardson, Texas 75080
Tel. 1-800/527-3475 214/340-5366
214/690-0207 Sat. and Evenings Only

SS

Explore the Frontiers of Intelligence



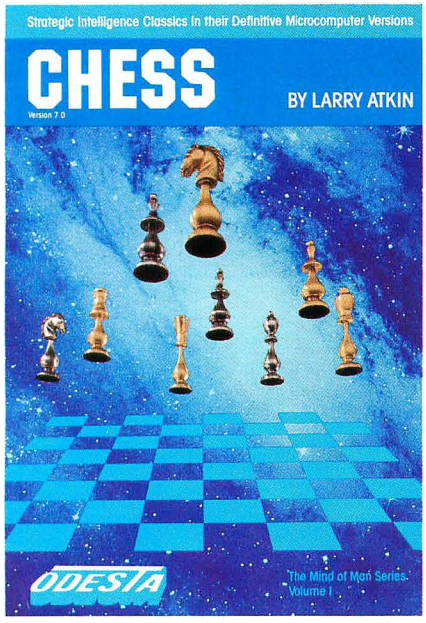
- Variations of blind-fold play—camouflaged or invisible pieces
- Invert board to play black on bottom
- Change pieces on board during game, or set up position
- Change between 15 levels of play, plus postal and mate-finder modes
- Show move that **Chess** is thinking about
- List played moves for each side
- Lines of force in: attacks and defenses on a square
- Lines of force out: squares attacked and defended
- **Chess** suggests a move
- Show moves **Chess** thinks you will make, and its responses
- Evaluation of a position
- Return to board or switch to command menu
- Take back a move (repeatable)
- Play move suggested by look-ahead search
- **Chess** plays neither side
- Switch sides
- **Chess** plays against itself—one level against another
- Replay through most advanced position
- Skip to most advanced position
- Start new game
- Leave program
- Save, get, and delete games to and from disk

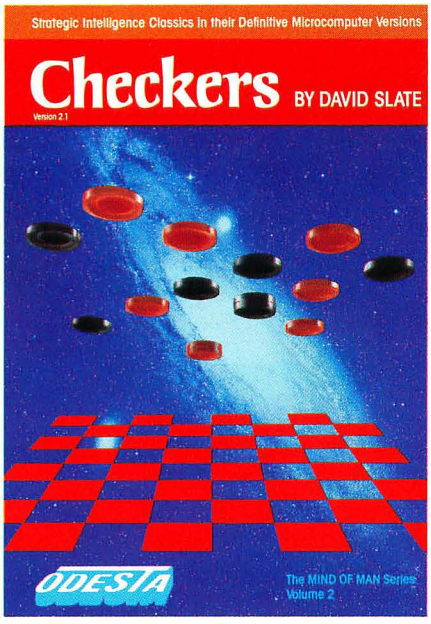
All features self-documented; all choices cursor-controlled
Screen shows "outward" and "look" features being used

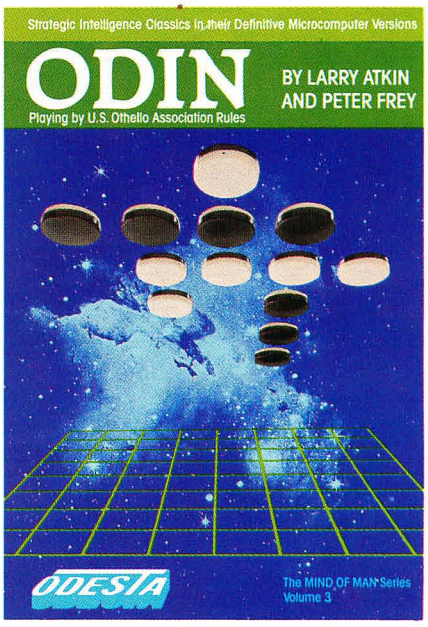
THE PEOPLE BEHIND THE PROGRAMS:

Larry Atkin & David Slate: Authors of the Northwestern University Chess 4.7 program—World Computer Chess Champion, 1977-1980

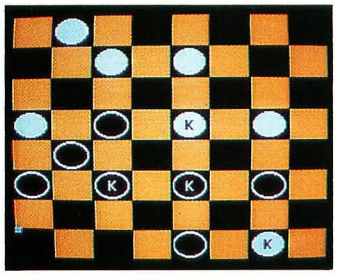
Peter Frey: Northwestern University professor
Editor: **Chess Skill in Man and Machine**
One of U.S. Othello Assoc.'s top-ranked players



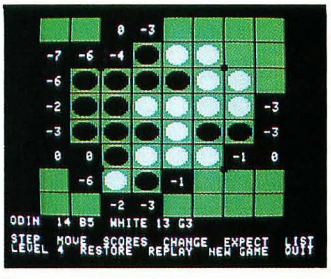




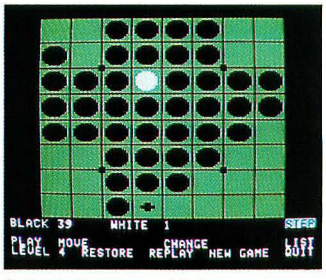
Checkers' features



Black to move and win
(From Checkers documentation)



"Scores" feature in Odin



A clue to the secret of Odin:
Black is destined to lose.

ODESTA

930 Pitner
Evanston, IL 60202
(U.S.A.)

Chess: $69.95
Checkers: $49.95
Odin: $49.95

See your local software dealer, or order (Mastercard or Visa):
800-323-5423
(in Illinois, call 312-328-7101)

For **Apple II, Apple II Plus** 48K disk systems, and **Atari** 48K disk systems. Odin is also available for **TRS-80 Model 1 & 3** 32K disk systems.

SINGLES' NIGHT AT MOLLY'S

By Norman J. Wazaney Jr.

You never dreamed Solitaire could be so fascinating.

Solitaire in a saloon can be fun but it's better on your Apple*. Fair warning: if you get hooked on Solitaire, beware of this game! "Singles' Night at Molly's" is actually two basic solitaire card games with several variations permitting you to use the skill level and strategy you enjoy most. Play alone or against other players, where a rating system declares the winner. Features High Resolution color graphics, full user documentation and various scoring potentials.

There are hours, days, even years of pleasure to be derived from this intriguing game. Available now for only $29.95 at computer stores.

To order by phone, call 800-526-9042 and use your Visa or MasterCard. All shipments made the same day in which orders are received. To order by mail, add $1 for shipping charges and send your check to:

Soft Images, 200 Route 17, Mahwah, NJ 07430.

"Singles' Night" runs on 48K Apple II, Applesoft in ROM. 3.3 DOS/One Disk Drive

*Apple is a registered trademark of Apple Computer Co.

Soft images
DIVISION OF DECISION SYSTEMS INC