No. 14 NOVEMBER/DECEMBER $3.00

THE MAGAZINE FOR ATARI® COMPUTER OWNERS

ANALOG COMPUTING

TOOLS AND UTILITIES ISSUE

HEXPAD

RETROFIRE

LUMBERJACK

CONVERSIONS

BASIC DISASSEMBLER

0 74470 12385 14

THE SECRETS OF PERFECT MEMORY: ONE AND ONE HALF EARTH DOLLARS

AT LAST: THE WHOLE TRUTH ABOUT FLOPPIES.

Amazing book reveals all!

How to keep from brainwashing your disk so it never loses it's memory.

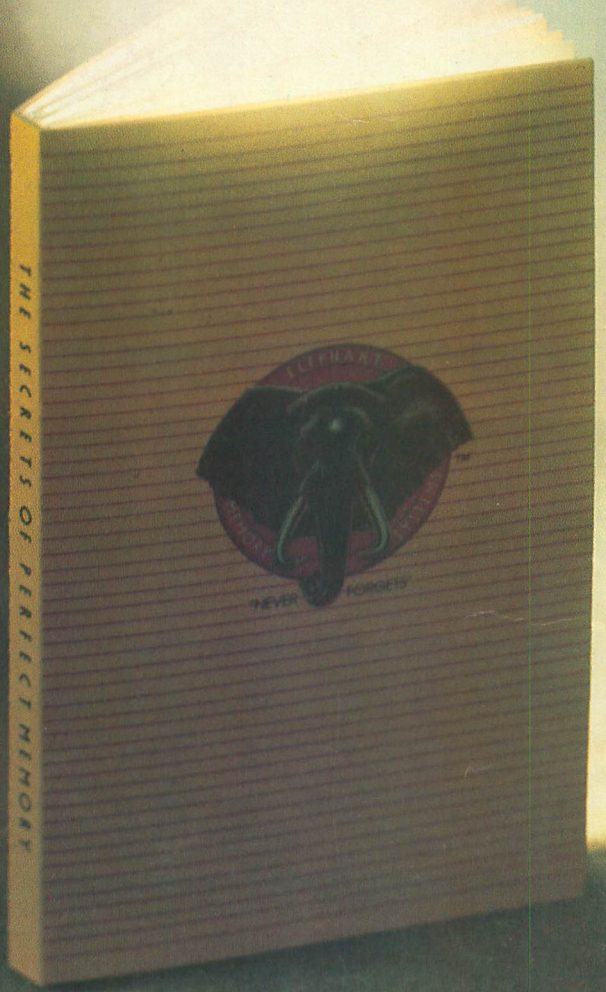
How fingerprints can actually damage disks. Unretouched Kirlian photographs of UFO's (Unidentified Floppy Objects)! The incredible importance of making copies: the Department of Redundancy Department—and what goes on when it goes on! Powerful secret methods that scientists claim can actually prevent computer amnesia! All this, and much more . . .

In short, it's an 80-page plain-English, graphically stunning, pocket-sized definitive guide to the care and feeding of flexible disks.

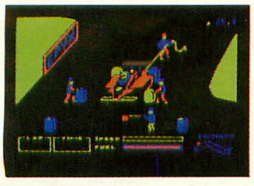
For The Book, ask your nearest computer store that sells Elephant™ disks, and bring along one and one half earth dollars.

For the name of the store, ask us.

ELEPHANT MEMORY SYSTEMS® Marketed exclusively by Leading Edge Products, Inc., Information Systems and Supplies Division, 55 Providence Highway, Norwood, MA 02062. Call toll free 1-800-343-8413, In Massachusetts, call collect (617) 769-8150, Telex 951-624.



PITSTOP.™
WHERE WINNING IS THE PITS.



You'll never make Grand Prix champion just driving in circles.

You've got to stop sometime. The question is when. Right now you're in the lead. But the faster you go, the more gas you consume. And the quicker your tires wear down.

If you do pull into the pits, though, you lose precious seconds. So it's up to you to make sure the pit crew is quick with those tires. And careful with that gas. Otherwise, poof! you're out of the race.

See your retailer for available computer formats.

So what'll it be, Mario? Think your tires will hold up for another lap? Or should you play it safe and go get some new ones?

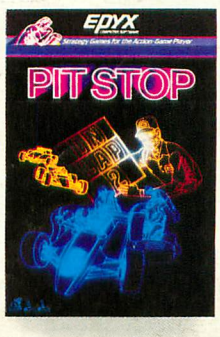
Think it over. Because Pitstop™ is the one and only road race game where winning is more than just driving. It's the pits.

Goggles not included.

One or two players; 6 racecourses, joystick control.

EPYX™ COMPUTER SOFTWARE

STRATEGY GAMES FOR THE ACTION-GAME PLAYER.



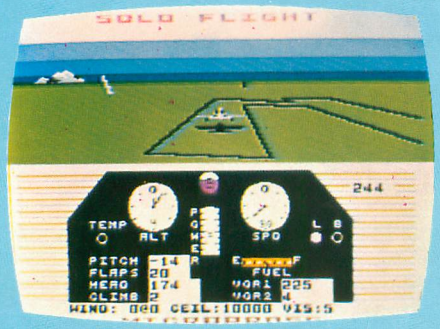
AT LAST!!! A REAL FLIGHT SIMULATOR!!!

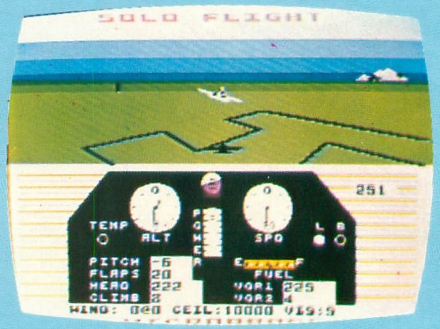
— From The Creators of HELLCAT ACE!

A real flight simulator allows the pilot to practice takeoffs, landings, cross-country navigation, and emergency procedures in Day, Night, Cross-Wind, and Instrument Flying scenarios! This practice requires realistic three-dimensional terrain graphics, actual instrument panel configurations, and a realistic simulation of the thrill and challenge of the flight experience! The experienced fighter pilots of MicroProse bring you the excitement and joy of flying with SOLO FLIGHT!

- Three Dimensional Terrain Graphics
- Multiple Airports and Runways
- Dual Radio Navigation Instruments
- Full Feature, Actual Configuration Instrument Panel
- Multiple Air Navigation Charts
- Instrument Approaches

- Takeoffs
- Landings
- Cross-Country Navigation — Day, Night, VFR and IFR Flying
- Emergency Procedures Practice
- Air Mail Delivery Game for 1-4 Players
- Smooth Landing & Good Navigation Scoring
- For All Atari Computers, Disk or Cassette . . . $34.95
- Coming Soon for Commodore 64





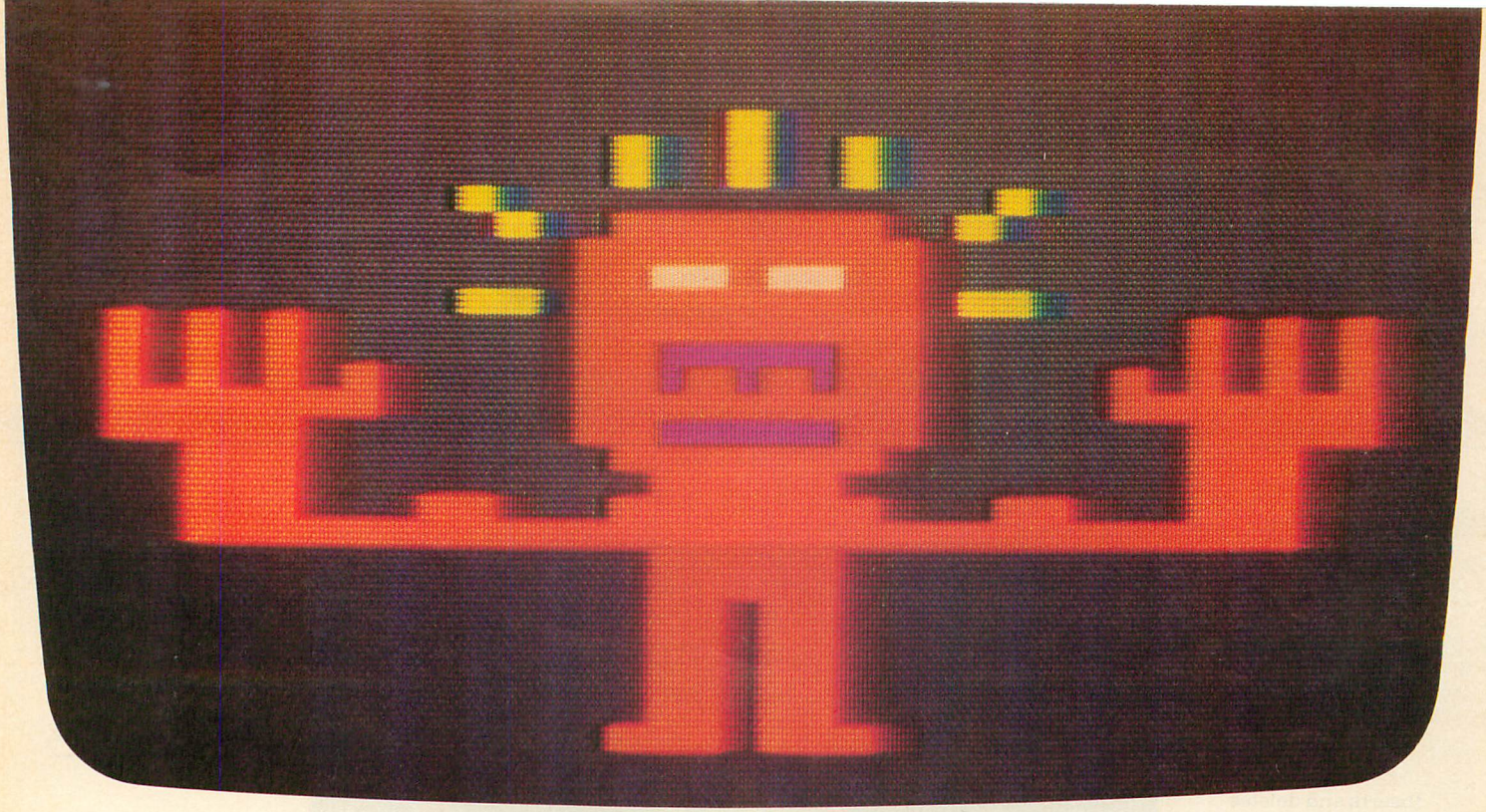
Look for MicroProse's two new exciting real time combat simulations — MIG ALLEY ACE & NATO COMMANDER. Write or call for our Free Catalog.

If you cannot find our games at your local store, you can order by MasterCard or VISA, Money Order, COD or Check. Add $2.50 for Postage and Handling. Maryland Residents add 5% Sales Tax.

MicroProse Software

DEALER INQUIRIES INVITED.

10616 Beaver Dam Road, Hunt Valley, MD 21030
(301) 667-1151



WOULD YOU SHELL OUT $1000 TO MATCH WITS WITH THIS?

SUSPENDED,™ The WITNESS,™ PLANETFALL,™ ENCHANTER,™ and INFIDEL™—has become an instant best-seller. For the simple reason that Infocom offers you something as rare and valuable as anything in software—real entertainment.

Meet your match. Meet Infocom games: perhaps the best reason in software for owning a personal computer.

In fact, people have been known to purchase computers and disk drives solely for the purpose of playing our games. And they haven't been disappointed. Because Infocom's prose stimulates your imagination to a degree nothing else in software approaches. Instead of putting funny little creatures on your screen, we put you inside our stories. And we confront you with startlingly realistic environments alive with situations, personalities, and logical puzzles the like of which you won't find elsewhere. The secret? We've found the way to plug our prose right into your imagination, and catapult you into a whole new dimension.

If you think such an extraordinary experience is worth having, you're not alone. Everything we've ever written—ZORK® I, II, and III, DEADLINE,™ STARCROSS,™

At last, you can fritter away your evenings playing a computer game without feeling like you're frittering away your computer investment.

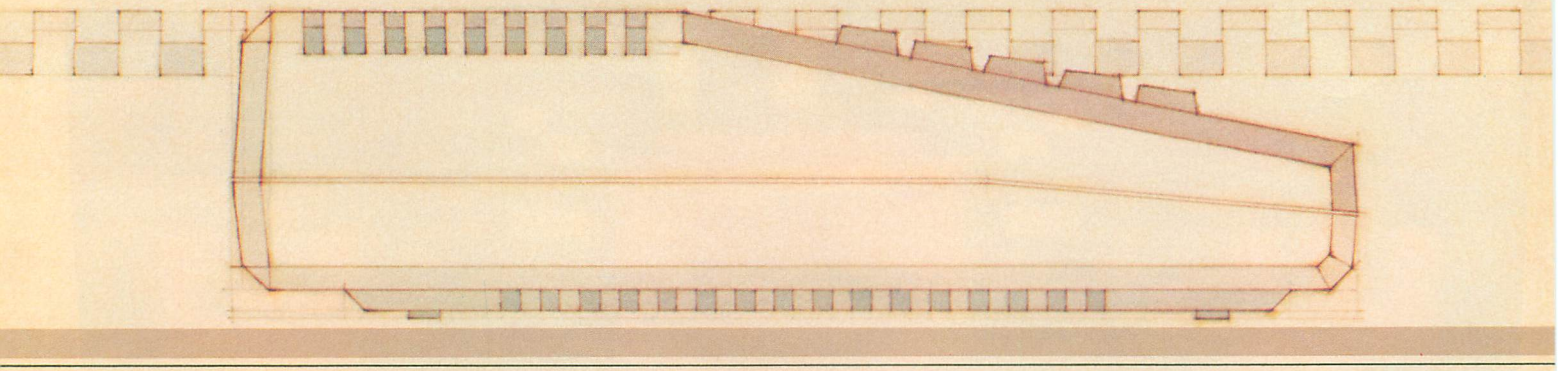
Step up to Infocom. All words. No pictures. The secret reaches of your mind are beckoning. A whole new dimension is in there waiting for you.

(For more information on Infocom games contact: Infocom, Inc., P.O. Box 855, Garden City, NY 11530.)

INFOCOM™

The next dimension.

For your: Apple II, Atari, Commodore 64, CP/M 8," DEC Rainbow, DEC RT-11, IBM, MS-DOS 2.0, NEC APC, NEC PC-8000, Osborne, TI Professional, TI 99/4A, TRS-80 Model I, TRS-80 Model III.

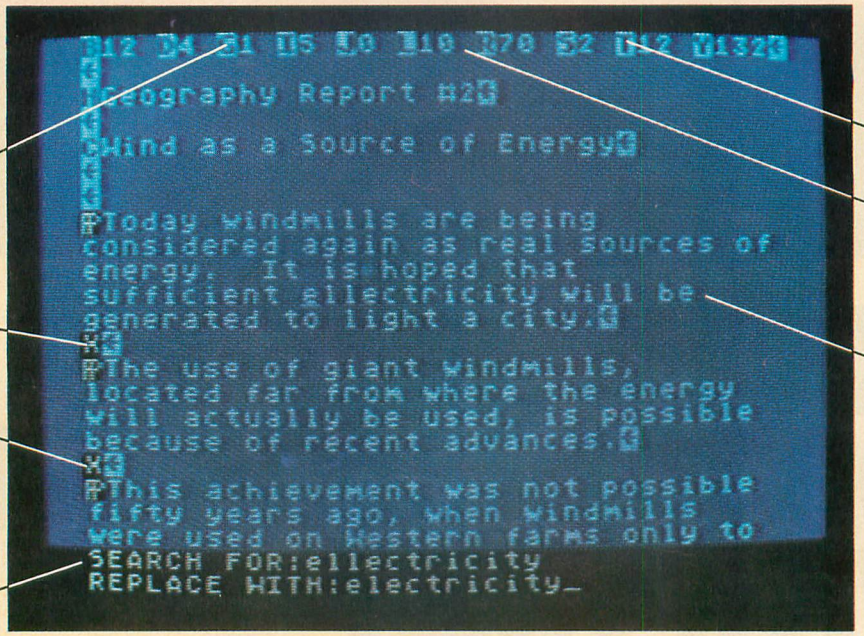


Prints with built-in format or lets you create your own: center, underline. Boldface, elongated, proportional and condensed print.

Inserts and deletes characters, lines or blocks of text.

Duplicates and moves blocks of text.

Searches for and/or replaces words or phrases, one at a time or all at once.



Changes line spacing, margins, print type, paragraph indents anywhere in the document.

Instantly reformats.

Enters text easily, with word wrap, one main menu and helpful prompts.

Creates multi-line headers and footers.

Prints double columns, form letters, multiple copies of a page or document, at the touch of a key.

Numbers pages and creates section numbers automatically.

Corrects spelling (with extra software).

Chains and merges files.

Has "print preview": you can check your copy before it's printed.

You can't find a friendlier, more powerful word processor at twice the price. New AtariWriter.™ Under $100.

Now you can do multi-featured word processing at home, simply. At a family budget price.

Our ROM-based cartridge technology means you can use new AtariWriter on any ATARI® Home Computer (even 16K) for personal and business correspondence, term papers, committee reports, mailings, etc.

It also lets you choose between cassette and disk storage systems.

One very special AtariWriter feature: you can correct as you write, without switching back and forth between Create and Edit modes.

And our memory buffer offers an "undo" command to let you change your mind, and restore text you've just deleted.

Check into our remarkable AtariWriter, and our choice of letter quality and dot matrix printers, also reasonably priced, at Atari dealers. Call 800-538-8543 for dealer nearest you. In California, call 800-672-1404.

You'll do more with Atari Home Computers.

 A Warner Communications Company

No. 14 NOVEMBER/DECEMBER

THE MAGAZINE FOR ATARI® COMPUTER OWNERS™

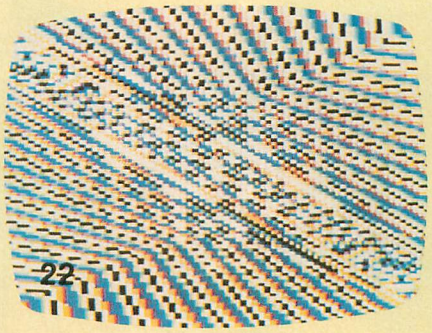
ANALOG COMPUTING

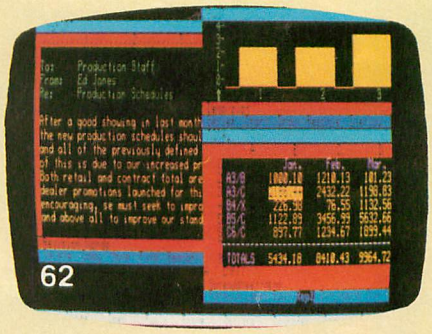
FEATURES/ARTICLES

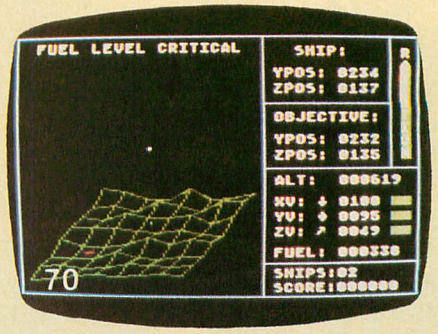
REVIEWS/PROFILES

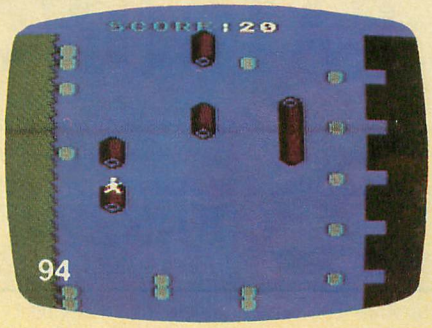
PROGRAM LISTINGS

COLUMNS









SAIGON: THE FINAL DAYS

By Jyym Pearson, Robyn Pearson & Norman Sailer

Gritty realism and historic fact blend to form a unique adventuring experience that plunges you into a controversial chapter of recent history.

Visit your local dealer *today*, or call us for your copy!

800-327-7172

Available for your 16K TRS-80 Model 1 or 3, in either tape or disk format, for only $24.95.
The Atari tape version (needs 16K) is $24.95.
The Apple version has great full-color hi-res graphics for $34.95.

Adventure® INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.

BOX 3435 • LONGWOOD, FL 32750
(305) 862-6917

149

Photo by Paul Grupp

A.N.A.L.O.G. STAFF

Editors/Publishers
MICHAEL DESCHENES
LEE H. PAPPAS

Managing Editor
JON A. BELL

Technical Editor
BRIAN MORIARTY

Contributing Editors
JOEL GLUCK
BRADEN GRIFFIN, M.D.
TONY MESSINA
DR. GARY ROSE
MARCIA ROSE

Art Director
BOB DESI

Technical Division
CHARLES BACHAND
TOM HUDSON
KYLE PEACOCK

Advertising Manager
MICHAEL DESCHENES

Distribution
PATRICK J. KELLEY

Typography
HENDRICKX &
LARRIVEE Co., Inc.

Production/Distribution
LORELL PRESS, INC.

Contributors
EDWARD BEVER
PETER C. BUDGELL
MAURICE ELLIOT
BERNARD ERTL
JOHN EUKER
RANDAL C. GIBSON
ARTHUR A. NEVOLA
RICHARD SELTZER
KEITH VALENZA

ABOUT THE COVER

This issue's cover illustrates the theme: ATARI tools. The utilities in this issue can help ATARI programmers exploit special features of their computer, or they can be used to manipulate other programs. Strewn about a polished brass sheet are tools—both unusual and ordinary. And, dominating the center of the frame is a representation of the tool most important to our readers. It is a representation of perhaps the ultimate tool—a computer.

A.N.A.L.O.G. COMPUTING (ISSN 0744-9917) is published bi-monthly for $14 per year by **A.N.A.L.O.G.** Magazine Corp., 565 Main Street, Cherry Valley, MA 01611, Tel. (617) 892-3488. Second-class postage paid at Worcester, MA and additional mailing offices. POSTMASTER: Send address changes to **A.N.A.L.O.G. COMPUTING**, P.O. Box 615, Holmes, PA 19043. Program listings should be provided in printed form. Articles should be furnished as typed copy in upper and lower case with double spacing. By submitting articles to **A.N.A.L.O.G. COMPUTING**, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of **A.N.A.L.O.G.** If not accepted for publication, the articles and/or programs will remain the property of the author. If submissions are to be returned, please supply self-addressed, stamped envelope. U.S.A. Newstand distribution by Eastern News Distributors, Inc., 111 Eighth Ave., New York, NY 10011.

EDITORIAL

by Jon A. Bell

A.N.A.L.O.G. Computing is going monthly as of January 1st. We have contemplated going monthly since our first issue. However, we had to consider the single most-asked question from our readers — can we keep up our quality?

I believe that we can maintain our editorial standards. **A.N.A.L.O.G. Computing** will continue to publish the honest, in-depth product reviews that have earned us the respect of the ATARI community. Our programming columns, feature tutorials and advanced utility programs will help you get the most from your computer investment. And **A.N.A.L.O.G.** will always publish the finest public-domain games to be found in any magazine, in both BASIC and machine language (with full commented source code). We are lucky to have an extremely talented readership, judging from the quality of recent program and article submissions. I can promise a number of fantastic things to appear in future issues.

Editorial content aside, there are other aspects of going monthly that should be mentioned. If you already subscribe to **A.N.A.L.O.G.**, you will still receive the correct number of issues, but your subscription will run out sooner. The new subscription rates are as follows: a one-year subscription will be $28.00 (a saving of 20% over the cover price) and a two-year subscription will be $52.00. We will be offering half-year cassette and disk subscriptions for those who do not want to pay an initial large fee for a year's worth of programs. A half-year cassette subscription will be $48.00; a full year, $90.00. A half-year disk subscription will be $72.00 or $130.00 for a full year.

While we're on the subject of rates, I ought to explain the increase in our cover price. It's no secret that magazine publishing is an expensive affair, and that printing costs are rising steadily. We have remained at a cover price of $2.50 for almost a year, while our number of pages has increased from 124 to 156 for this issue. I believe that our mixture of software reviews, tutorials, public-domain games and previews of new products is well worth our new $3.00 price.

A.N.A.L.O.G. Computing has had a toll-free number for new subscribers for over a year. The number is 1-800-345-8112 (in Pennsylvania, 1-800-662-2444). I'd like to emphasize that this number is for new subscriptions only. The operators on duty work for a magazine fulfillment service which handles **A.N.A.L.O.G. Computing** as well as other magazines. They cannot answer programming questions, advertising questions or queries about the editorial content of upcoming issues, so please don't waste your time — or theirs.

A lot of people are wondering where the **A.N.A.L.O.G. Compendium** is. There have been some printing delays in our first book, but we think our readers will agree that it was worth the wait. We have added a number of new programs that did not appear in **A.N.A.L.O.G. Computing**, along with enhanced versions of some old favorites. We are also going to include the complete assembly-language source code for Tony Messina's Disk Tool utility. These last-minute additions have raised the total page count from 150 to over 190 pages — with no increase in the cover price. The **A.N.A.L.O.G. Compendium** will be out before Christmas gift-giving time. We thank you for your patience and patronage.

Finally, if you have a programming question, we would prefer that you send in your query by mail, along with an SASE if you want a personal reply. Send to:

Technical Division
A.N.A.L.O.G. Computing
P.O. Box 23
Worcester, MA 01603

It is very difficult to answer programming questions over the phone. We do not accept collect calls dealing with program questions or subscription problems.

1027 update.

Additional information provided from ATARI indicates that the **1027 Letter Quality Printer** is designed strictly for sheet fed paper. Contrary to what we reported last issue, roll or tractor feed paper will most likely jam in the printer. **AtariWriter** is reportedly being updated to work on the soon-to-be-released **1027.** □

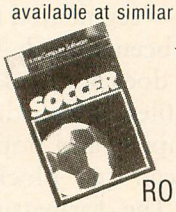
Great Software • Great Service • ~~Great~~ incredible Prices

from Sierra On-Line

Frogger

16K Tape / 32K Disk **$22.88**

The official Atari computer version of the smash coin-op hit. Beautiful graphics. The disk version features seven different tunes.
All Sierra On-Line programs, including Ulysses & The Golden Fleece and The Wizard & The Princess in stock and available at similar savings.



from Thorn-EMI

Soccer

ROM Cart. **$34.88**

In our opinion, this is **the** best arcade sports game available today. Play vs. the computer or human opponents. Fantastic graphics.
All Thorn—EMI Cartridges including Orc Attack, River Rescue, Submarine Commander and Jumbo Jet Pilot available at similar savings.

from Tronix

Juice!

32K Tape or Disk **$21.88**

Q-Bert fans rejoice! Bounce around the game board squares but don't collide with any of the cute little characters you see. It's challenging and, sometimes, maddening. But, most of all, it's fun.

from Synapse

Blue Max

32K Tape or Disk **$23.88**

The premier Atari software house brings you still another fabulous program. As a member of the Royal Air Force, you will command a biplane fighter/bomber and will battle the Axis on the ground and in the skies.
All Synapse programs including Shadow World, Dimension X, Pharoah's Curse, Necromancer and Shamus Case II in stock.

from Broderbund

Bank St. Writer

48K Disk **$49.88**

If you own a disk drive, but haven't yet acquired a word processor because the programs you've investigated are too complex and/or expensive for your needs, this program should excite you. Developed at the Bank Street School, it's surprisingly powerful and perfect for students and first-time users.
All Broderbund Software including AE, Choplifter and Stellar Shuttle in stock and available at similar savings.

from Infocom

The Witness

32K Disk **$34.88**

Working from a clue-laden police file and battling a 12-hour time limit, you're up against your toughest murder case to date: a sordid family affair that may land everyone from the knockout heiress to the poker-faced Oriental butler in the slammer. . . and the only testimony you can trust is that of your own eyes—because you are "The Witness."
All Infocom programs including Zork I, II, and III, Suspended, Deadline and Starcross in stock and available at similar savings.

from Lightning

Mastertype

32K Disk **$25.88**

A breakthrough in interactive educational software. Learn to type as you battle wave after wave of invading words riding in hi-res spaceships! 17 pre-programmed lessons, plus others that you can create, guarantee continuous challenge and help you develop mastery of the Atari Keyboard.

from Thorn-EMI

Home Financial Management

16K Tape **$19.88**

Set up a budget and then see where you stand financially either in figures or through bar charts. This program should pay for itself!
All Thorn-EMI tapes including Figure Fun available at similar savings.

from Datasoft

Zaxxon

16K Tape or Disk **$25.88**

The long-awaited version of the 3-D arcade hit of 1983. A must-have game for the space gamer.
All Datasoft programs, including Sands of Egypt and Text Wizard available at similar savings.

from Spinnaker

In Search of the Most Amazing Thing

48K Disk **$26.88**

The newest program from Tom Snyder, author of Snooper Troops #1 and #2. Bargain with aliens at an auction, raise currency, then buy your provisions and board your specially-equipped B-Liner in search of the Most Amazing Thing. It's educational — and it's fun. Free bonus novelette included.
All Spinnaker software including Facemaker, Story Machine and Snooper Troops #1 and #2 in stock and available at similar savings.

from MUSE

Castle Wolfenstein

48K Disk **$18.88**

One of the most popular games ever released for the Apple is now available to you. Try to find the war plans and escape from the Nazi stronghold in this exciting blend of strategy and arcade action

from Sirius

Bandits

48K Disk **$22.88**

Bouncing nerve gas balloons, energy shields, space bandits attempting to steal your supplies — they're all on the screen in one of the best space arcade games yet.
All Sirius Software including Alpha Shield, Way Out and Sneakers in stock and available at similar savings.

Please call or write for FREE CATALOG

Mail order prices do not apply at our retail location.

HOW TO ORDER:
Use mail or phone. For immediate shipment pay by Money Order, Certified Check, Cashiers Check, VISA, or Master Card (add 4% for credit card). Personal checks are accepted but require 1-3 weeks to clear. 10% deposit required for C.O.D.

SHIPPING & HANDLING:
$2.50 for orders under $50; free shipping over $50; Mass. residents add 5% sales tax. Prices are mail order only and are subject to change without notice.

SOFTWARE PLUS

(617) 369-1992
for orders and information
phone hours: 10-6 EST Mon.—Sat.

45 Walden St., Box AA, Concord, Mass. 01742

READER COMMENT

Dear Editor:

Raymond Tillman wrote a very comprehensive article regarding ATARI Pascal in issue no. 11 of **A.N.A.L.O.G.** Since we have this Pascal up and working (and very user friendly) in our high school laboratory, we thought we would share some pertinent information.

Figure 1 shows the use of memory as originally packaged by ATARI (as near as we can tell). **Figure 2** shows how we redesigned the use of memory for our laboratory (according to ATARI memory mapping). These changes were possible by overlapping the compiler's phases so that when one phase was finished the next phase started at the same beginning address.

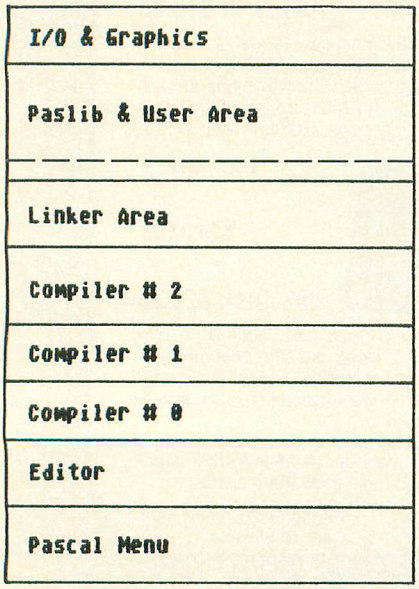


Figure 1.

We apportioned the same memory for graphics or I/O routines instead of using contiguous memory and repackaged the library. The user area grew from around 22K to 32K. We use a 48K ATARI 800 machine.

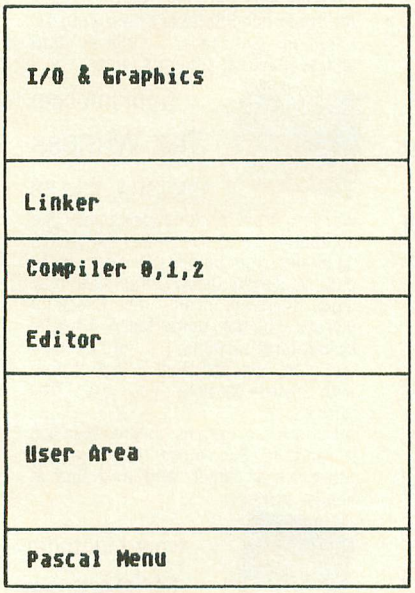


Figure 2.

The incomplete documentation was supplemented by a 19-page user's guide written by Charles Tucker, one of our senior students. We download certain debugging tools into the user's area when we need them. These include Trace, Decompose, Track, I/O Test and others. The original documentation does not give an adequate explanation of these tools.

The delivered Pascal needed four disks (not two as Mr. Tillman mentioned). One disk was the linker, one for the compiler, one for the editor and one for the addendum library and certain test functions. We repackaged the four disks into one two-sided disk. One side contains the editor, linker and Pascal menu, the obverse side contains the compiler and I/O routines.

The three library functions that do not work are ABS, SQR, and SQRT. We fixed SQR and SQRT and write an algorithm when we need absolute value. The problems with random disk access (RDA) — segmented files — have not yet been solved. The sequential I/O does not lend itself to segmented files. We are disassembling the compiler and hope to fix the block read and block write impasses. The formatting routines for real numbers also have to be fixed. The system cannot format any number smaller than one-tenth.

Compilation problems were ameliorated by correcting the error pointers so that the compiler returns to the menu instead of having the computer "lock-up" on us. The only time it "locks-up" now is for the fatal error messages. The "string" data type declaration does not work because of the interference of the screen editor associated with the ATARI 800. Apparently, strings use some of the same memory locations as the screen editor.

The debugger points to error codes in the standard ATARI DOS manual. The explanations of error codes and fatal error codes within the ATARI DOS manual are poorly done. Some of the documentation comes with the product. Our changes have increased the capacity from about 300 program lines to 500 lines per program. We are still unable to chain modules, however, this does not affect our work in the high school environment. When we accomplish our disassembling of the compiler, we hope to correct the chaining malfunction. The similarity of ATARI Pascal to MT +Pascal has allowed us to fit the ATARI Pascal into our laboratory

where 16 non-ATARI computers run CP/M with MT+Pascal.

Our environment requires the Pascal to handle the solutions of:

1) A five by five Gaussian reduction algorithm that finds the inverse matrix of the coefficients and refinds the roots by matrix multiplication.

2) Mean-Median-Mode, a large program that uses a data file built with the editor. This program works on a class roster. It sorts on the names, computes means, does a frequency distribution (mode), and finds the median of an array of numbers — i.e., the class's grades.

3) A solution for the roots of a fifth degree polynomial.

We still are working on a RDA program, binary searches, and tree sorts using files developed through the program. Because of the problems with block reads and block writes associated with disk access, these algorithms must await the reconstruction of the compiler. When we complete these programs, ATARI Pascal will satisfy the needs of the new Computer Advanced Placement curriculum.

Irwin J. Hoffman, Ph.D.
Jim Branche, E.E.
Russell Anderson, Editing
Charles Tucker, Research
Computer Laboratory
George Washington
High School

Dear Editor:

Your preliminary coverage of the new ATARI computers (issue 12) was good, but you brushed over one item which I feel deserves more attention — the phasing out of the 1200XL when the new machines become available.

Whether the 1200 is phased out or upgraded (as I have also heard), ATARI must be setting a record for shortest projected lifespan of a new product. The 1200 was introduced in mid-March, and will be obsolete by the end of the year. Eight months!

This bodes badly for those of us who bought the 1200 thinking that it was ATARI's best computer and that we'd get at least a few years out of it before it was superceded. In less than a year, my 1200 will be a museum piece, with a one-of-a-kind operating system, and will be the only ATARI computer with no provision for expansion. That is, unless we can convince ATARI to offer us some sort of trade-in for a 1400XL or (at least) an 800XL. I urge all 1200XL owners who feel as I do to write to ATARI and let them know, and maybe we'll get some results.

Now that I've gotten that off my chest, I have a question for anyone who could help me. What do I do if I have typed in a program, only to find out when I try to save it that my disk drive isn't turned on? Is there any way to activate the disk drive and load AUTORUN.SYS without clobbering whatever's in memory?

Sincerely,
Neil Weinstock
Livingston, New Jersey

The only practical way around this problem is to save the program on cassette. The computer can then be re-booted with DOS, the program loaded from cassette and re-saved on disk.

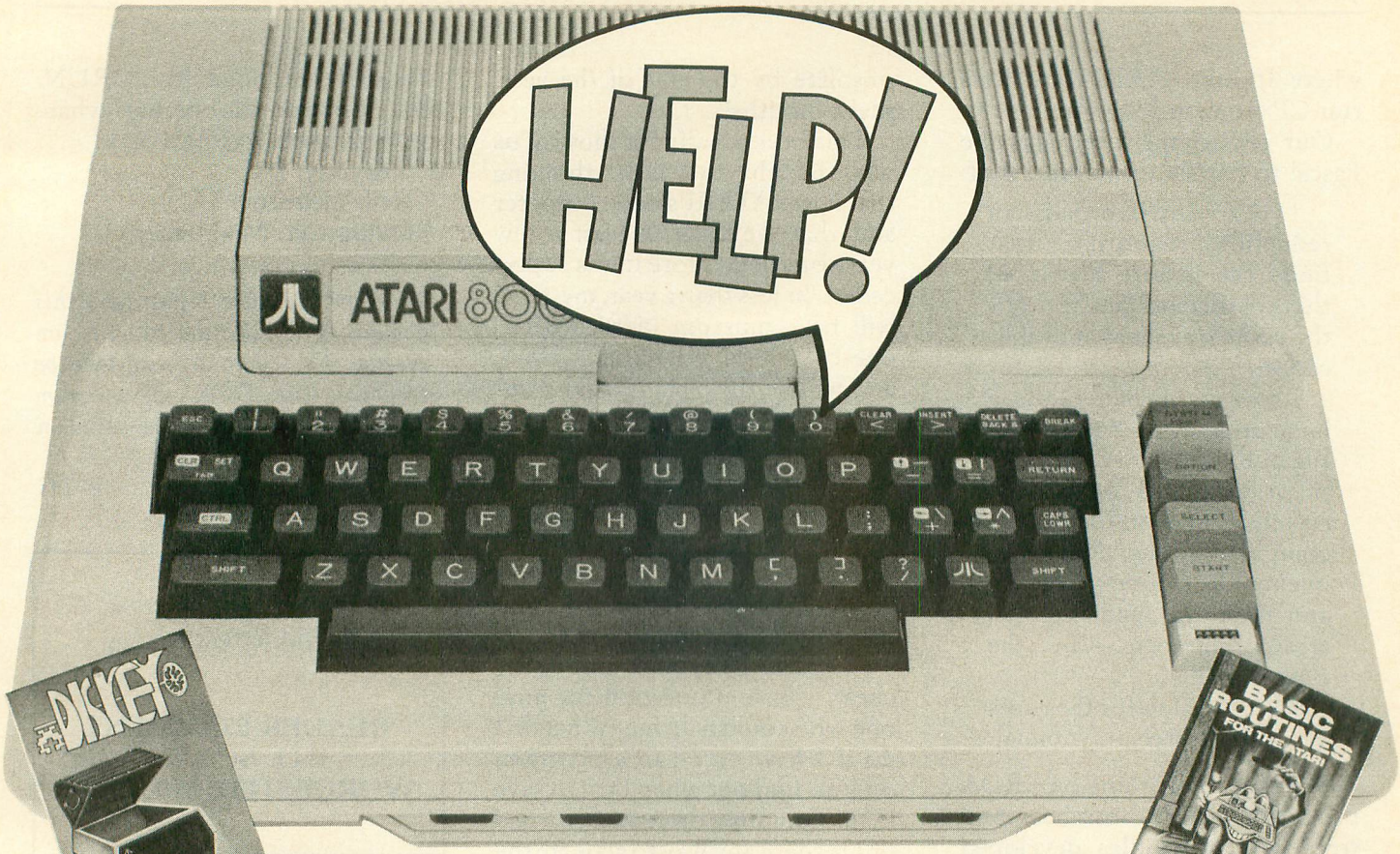
—TH

Send letters to:

READER COMMENT
P.O. BOX 23
WORCESTER, MA 01603

Attention Programmers!

A.N.A.L.O.G. Computing is interested in programs, articles, and software review submissions dealing with the ATARI home computers. If you feel that you can write as well as you can program, then submit those articles and reviews that have been floating around in your head, awaiting publication. This is your opportunity to share your knowledge with the growing family of ATARI computer owners. **A.N.A.L.O.G.** pays between $30.00-$360.00 for all articles. All submissions for publication must be typed, upper and lower case with double spacing. Program listings should be provided in printed form, and on cassette or disk. By submitting articles to **A.N.A.L.O.G. Computing**, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of **A.N.A.L.O.G.** If not accepted for publication, the articles and/or programs will remain the property of the author. If submissions are to be returned, please supply a self-addressed, stamped envelope. All submissions of any kind must be accompanied by the author's full address and telephone number. Send programs to: Editor, **A.N.A.L.O.G. Computing**, P.O. Box 23, Worcester, MA 01603.



DISKEY

by Sparky Starks

Potential into practical reality. This is the core of DISKEY — a remarkable utility program that gives ATARI disk drive owners the flexibility to accomplish tasks that other utilities either ignored or only hinted at. With DISKEY, the user will be able to actually examine a disk and its directory, and repair some files that might otherwise have been lost. DISKEY also performs a multitude of other practical functions, including the following:

- ★ Automatically lists any unreadable or destroyed sectors
- ★ Sends contents of disk to printer selectively
- ★ Up to four separate drives may be addressed
- ★ Can be used to backup some of those "unbackupable" disks
- ★ Allows direct examination of any file
- ★ Over 50 separate key functions available

As an ATARI 400 or 800 owner, you have extraordinary power and versatility at your disposal. Tap into those resources effectively with DISKEY — new from Adventure International.

ATARI 400/800 32K DISK
052-0158 $49.95
(Note: Some features will require 40K)

SOME OF DISKEY KEY FUNCTIONS

A	Clear screen and filename
B	Byte compare, D1 to D2, OS to DS
C	Copy sectors, OD to DD, OS to DS
D	Toggle destination drive
E	Erase disk (format)
F	Select file sub-menu
L	Set automatic function lower limit (OS)
M	Modify Sector Map
N	New destination sector
O	Toggle originate drive
P	Print screen to printer
Q	Query (search for hex key, drive OD, sector OS to DS)
R	Read new OS, set DS to match
S	Search for ASCII key, drive OD, sector OS to DS
T	Tape to disk
U	Upper case conversion of printer lower case
V	Toggle write verify
W	Write memory buffer to sector DS, drive DD
X	Select EOR Sector Map screen print mask
Z	Zero memory buffer
+	Read upward, next sector on disk
–	Read downward
?	Directory information
!	Select directory sub-menu
cB	Byte compare, D1 to D2, whole disk
cC	Copy D1 to D2, whole disk
cD	Decimal to hex, ASCII conversion
cE	Erase disk (without new format)
cF	Modify sector forward sector chain reference
cH	Hex to decimal, ASCII conversion
cL	Locate bad sector on drive OD
cN	Modify sector file number reference
cO	Select one-drive functions sub-program
cP	Print current Disk Map
cR	RPM test drive OD
cS	Special file copy, no directory reference from source
cV	VTOC update and repair, drive OD
cY	Toggle Sure Response prompt enable
FA	File binary load address headers to printer
FD	Delete file
FF	Select filename for all file functions
FL	Lock file
FM	Show memory address load position in file
FQ	Relative Query
FR	Rename file
FS	Relative Search

BASIC ROUTINES FOR THE ATARI

by Jerry White

Finally, a self-help system that cuts through the stuffy technical jargon and allows the user to learn effective programming techniques fast! BASIC ROUTINES FOR THE ATARI has been written especially for ATARI 400 or 800 users who wish to write programs in BASIC. This package comes complete with an extensive manual and your choice of a tape or disk which contains all of the routines from the manual — which means you'll be able to actually see each of them in action on your ATARI. Some of the BASIC routines include joystick, sound, player missile strings and much more — and presented in a way that you'll quickly be able to learn and apply to your own programs.

If your programming ability lies somewhere between beginning and advanced, then look no further — BASIC ROUTINES FOR THE ATARI is the system for you.

BASIC ROUTINES book with 16K TAPE
051-0154 $24.95
BASIC ROUTINES book with 24K DISK
052-0154 $24.95

Published by Adventure INTERNATIONAL
a subsidiary of Scott Adams, Inc.
BOX 3435 • LONGWOOD, FL 32750 • (305) 830-8194

To order, see your local dealer. If he does not have the program, then call 1 (800) 327-7172 (orders only please) or write for our free catalog.

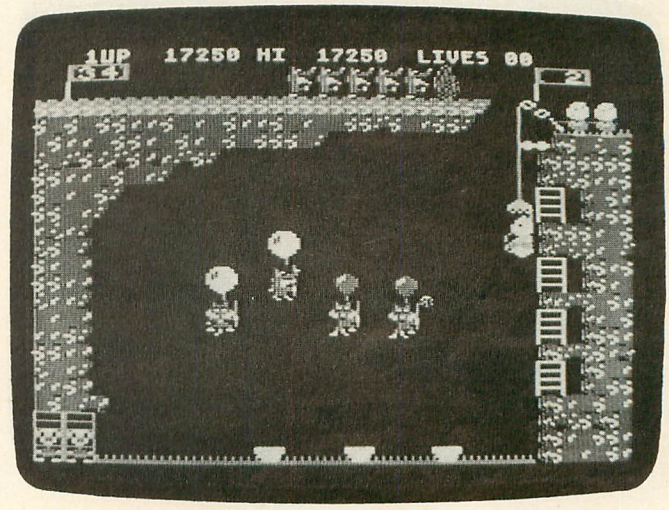
NEW PRODUCTS

by The Program Doctors

As the Christmas season nears, computer companies are releasing a great variety of new games, peripherals, educational products, and computer accessories. Most of them have not hit the stores yet, but will be there by the time you are ready to answer that age-old question from your great-uncle Ralph, "What do you want for Christmas?"

In this column we will discuss the best of the new releases and touch upon the expected new products. With the VCS market dying a slow death (many stores will drop the VCS after Christmas), more and more of the VCS companies are turning to the home computer market. Activision, Imagic and Mattel are just a few of the latest converts to the micro market, but the best of the new games come from established companies such as Gamestar, Synapse, Sirius, Broderbund, Datasoft, and yes, ATARI.

By the time you read this, the 1983 baseball season will probably have ended for all the other people in the world except for ATARI-users. **Star League Baseball** has finally arrived and to say it is realistic is an understatement. Available on 32K disk or cassette, Gamestar has thought of everything in its attempt to create a "you are there" baseball game. After hearing the National Anthem you can choose your pitcher, either "Heat" Muldoon or "Curves" Cassidy, and the make-up of your batters, the "Liners" who hit for average or the "Sluggers" who swing for the fences. You can play against a tough computer opponent who is excellent in the field and tenacious with bat or you can play with your friend. Batted balls are either grounders or fly balls (through the use of the "shadow technique"). Fielding is accomplished solely through the joystick and it takes a little practice to become an Ozzie Smith in the field. At first the fielding aspect of the game will frustrate you but as the old adage says, "practice makes perfect." Other features include a special batting practice mode to fine-tune your hitting against "Heat," bunting, and a choice in the 8th inning to bring "Knuckles" Flanagan, your ace reliever (they did think of everything), along with a few other surprises. With **Starbowl Football**, **Baja Buggies**, and now **Star League Baseball**, Gamestar has definitely cornered the sports game market for the ATARI.

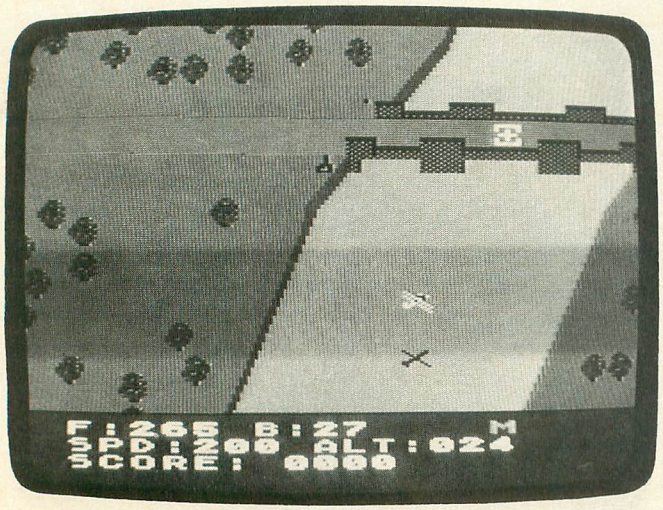


Pooyan

Synapse has enlarged its library of game software by three with the release of **Blue Max**, **Drelbs**, and **Zeppelin**. The sleeper of the trio is **Drelbs**, a fantasy arcade game with very addictive qualities. **Drelbs** are peaceful creatures whose land was invaded by evil gnomes called trollaboars who are out to destroy them. The friendly **Drelbs** have tricked these gnomes to follow them onto the atomic flip grid. Your task is to defeat the trollaboars and free all the captured **Drelbs**. This game combines the features of many different arcade games such as **Pacman**, **Ladybug**, and **Kid Grid**.

Whenever an author has written an arcade classic for the home computer most people anxiously await his next work. Such is the case with William Mataga, who wrote both **Shamus** and **Shamus II**. **Zeppelin** is his latest game. It is totally different from his previous work and includes 250 screens. As you fly your heavily armed **Zeppelin**, you navigate through seven treacherous levels (each different) encountering earthquakes, falling rocks, balloons, enemy zeppelins, buildings and towers. You are searching for the dynamite and plunger to blow up the enemy lair present on each level. This game scrolls automatically depending on the location of your **Zeppelin** on each screen. This game is not for the easily frustrated or the faint of heart. The last of

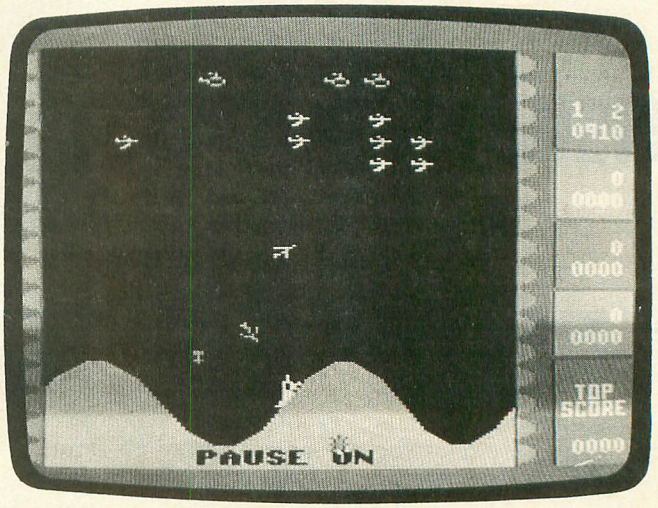
the new trio of games from Synapse is **Blue Max**. This game is simply superb in all programming and playability aspects. Flying your WWI fighter plane over land and water, you can fight enemy biplanes, bomb superdestroyers, blow up bridges, and strafe targeted buildings. Your mission is to be a good enough flying ace to earn a chance to destroy three major buildings in the awesome capitol city. Through the use of the "shadow effect" and diagonal scrolling, this 3-D game is definitely one of the top 10 games of the year.



Blue Max

Sirius has finally released **Wavy Navy**, a game which should satisfy all of the gamers who were disappointed with **Galaxian** from ATARI. This 48K disk-based game can be played with 1-4 players. It becomes constantly harder as you increase your proficiency against the enemy. You are a P.T. boat traveling in the ocean. The sky is covered with enemy bombers, backed up with machine gun helicopters. Your mission (obviously) is to rid the skies of the enemy. If you succeed, you are then confronted with sea mines, along with another fleet of bombers and choppers. This continues until you become President. Sirius has given us a quality piece of software, with top-notch sounds and graphics.

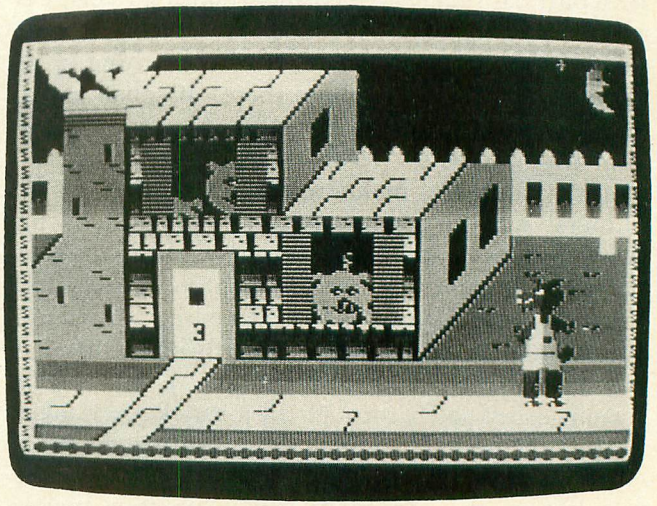
Congratulations are extended to Datasoft for their recent price decreases. Almost every product in their large line of utility and game software has been decreased in cost. Another trend-setting move has been their new policy of packaging *both* the disk and cassette version in the same box. This accomplishes two things: it reduces the number of stock-keeping units for the retailer, and gives the consumer both versions of the product if at a later date there is an upgrade to a disk drive. In the way of products, Datasoft has released **Pooyan**, an arcade game about pigs and foxes. The good guy is the pig as he attempts to prevent the foxes from invading his home. This game is an excellent re-creation of the original



Wavy Navy

licensed from Konami. Also, the long awaited **O'Riley's Mine** and **Moon Shuttle** are both now available, and while neither break any new ground they are decent arcade games for the collector.

Matchboxes from Broderbund is an excellent "Concentration" style game. Not only must you match hidden pictures, there is also a hidden word puzzle underneath that must be solved. Each picture is colorful and animated, and to aid your memory even further, each picture has its own catchy tune. This one will really appeal to the kids, as will **3 Little Pigs**, the first of Amulet Software's Magic Storybook series. The story is portrayed on the screen as the voice track reads to you, and the child can take part in the story with the use of a joystick.



3 Little Pigs

ATARI has not been asleep at the wheel (just dozing). They have released LOGO, an excellent language for children which is totally compatible with all the literature written about Apple Logo. (Read the review of LOGO on page 19.) Also by the time you read this, both **Pole Position** and **Ms. Pacman** should be in the stores and both are excellent arcade translations of the original classics.

The Challenge:
Match Wits with the Mind Games from Brøderbund!

OPERATION WHIRLWIND™

Your mind is the ultimate weapon in this unique war game. Strategy, not force, is the key to victory as you move your battalion through a series of testing skirmishes and battle actions. Your ability to command, to give orders, to move your troops skillfully determines the success of your assault and combat operations. Operation Whirlwind requires the concentration of chess—a typical game may take between one and three hours—and the ability to master the many tactics of waging a full scale war. How well you can focus your strategic energies will decide whether your victory is Questionable, Marginal, Tactical, Strategic or a Breakthrough. A great game for the thinking game player. Available on disk.

MATCHBOXES™

Pit your concentration against another player or the computer. Here's a puzzle game that will keep the whole family bemused, befuddled and playing happily for hours. A grid of 36 numbered boxes conceals an animated menagerie of colorful characters, creatures and objects. The object of the game is to match identical squares and then solve a hidden word puzzle. There are many game variations with puzzles that are frontward, backward, or scrambled. Matchboxes. It's the most memorable memory game you've ever played! Available on disk and cassette.

OPERATION WHIRLWIND AND MATCHBOXES FOR THE ATARI® HOME COMPUTERS. ATARI® is a trademark of Atari, Inc.

Make sure to look for The Arcade Machine™ and Brøderbund's new game phenomenon, Lode Runner™—also for Atari.

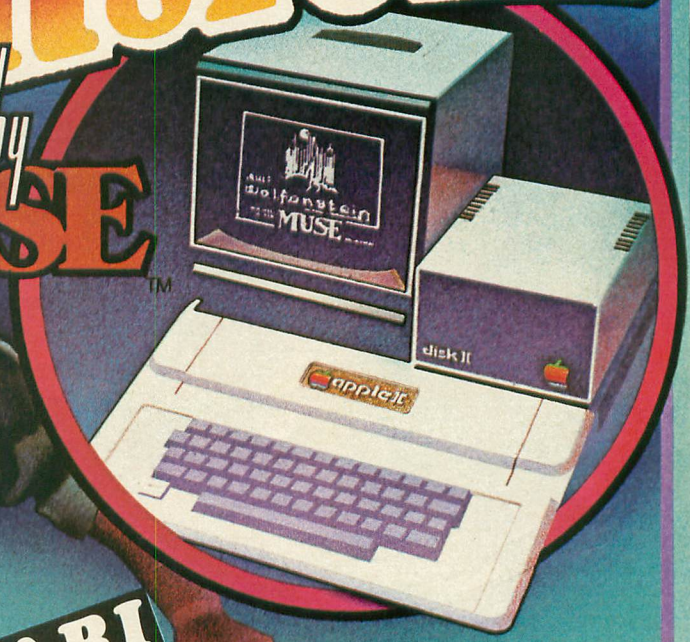
Brøderbund Software

Discover the Difference

17 Paul Drive San Rafael, CA 94903

Castle Wolfenstein™

by MUSE™



FOR APPLE & ATARI

THE MOST INTERACTIVE GAME EVER!

The wildest, most addictive adventure game to date–a genuine merger of fantasy and real-time arcade elements.

"A virtuoso feat of programming..."–*New York Magazine.*

For the Apple 11 48 K, Apple 11 E, the Atari 400 or 800, 32 K, Atari 1200 XL. $29.95

By Silas Warner.

ARCADE AWARDS 1983 ELECTRONIC GAMES MAGAZINE

Certificate of Merit For Outstanding Achievement

WALDREP

™MUSE

347 N. Charles Street
Baltimore, MD 21201 (301) 659-7212

Call or write for information and the name of your nearest MUSE dealer. Apple is a TM of Apple Computer Corp. Atari is a TM of Atari Inc.

Several new disk drives are hitting the market, and now when you decide to make the big step up to a drive it is not only a matter of *when*, but *which one?* There seems to be a lot of confusion about not only the ATARI 1050 disk drive but the other third party disk drives as well. The 1050 is not a true double density drive, but in actuality is only "one and one-half" density. Power pack problems were discovered with the first shipments and have been corrected, but more importantly, the 1050 had some problems booting third party software. The question remains "will the marketing wizards at ATARI view this as their problem or a problem of the third party manufacturers, similar to the way the 1200XL was handled?"

The Rana 1000 is a sleek-looking drive with a fancy front that has been advertised since April and has only recently become available. (A full review of this drive will appear in a future issue.)

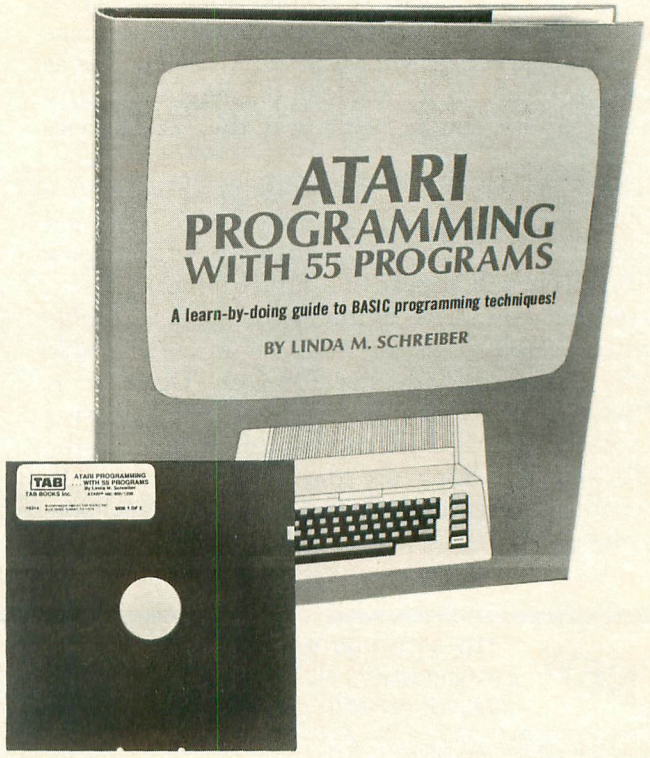
Another new drive is the Astra 1620, a double density, double drive unit priced at an unbelievable $595. Packaged with the **OS/A+** DOS, it is software-selectable as either single or double density. The Astra uses the MPI mechanism while both the 1050 and the Rana use the Tandon half-height mechanism. Percom, the leader in third-party drives for the ATARI, has just released a new single or double density unit with a built-in printer port. Consumers please note: the jury is still out on these third party drives; don't be price-conscious while drive shopping and make sure you research your purchase carefully *before* you buy, or you could be stuck with a lemon.

If you are shopping for an 850 interface, you may become frustrated in your search since there are not many to be found. No, they have not been discontinued, ATARI just stopped making them for three months during their move to Taiwan. Luckily, an alternative to the 850 has arrived from Advanced Interface Designs. The **Interfast-1** is a printer interface that is programmable with any dot matrix printer with graphics capability. It has a 4K buffer and is equipped with a 6502 microprocessor so it can be set up to print the entire ATARI character set, including the control characters. The operative word here is *printer*; you should be aware of the fact that there is no provision for a modem. The reasoning behind this is there are several new direct connect modems coming out, and by eliminating the modem port, a better printer interface could be produced at a lower cost.

In conclusion, Gary, Marcia, and Theo wish you all a happy, healthy, holiday season and a bug-free 1984.□

Now, You Can Become a Programming "Genius" with these Step-by-Step, Comprehensive Guides from Linda Schreiber and TAB

See how to tap ALL of the enormous programming potential offered by your ATARI® 400 or 800!



ATARI Programming . . . with 55 Programs

by Linda M. Schreiber, ANTIC columnist

A must-have learning guide for *every* ATARI user!

Written in non-technical, easy-to-follow language, this deluxe "programmer" provides a host of practical and fascinating programming ideas. From ATARI operation to organizing and writing your own programs . . . from mastering commands and statements to using the ANTICS® chip for more modes and colors on your screen, this is your key to discovering the exciting world of your ATARI computer. 256 pp./44 illus./7" × 10"/ Hardbound/$21.95/No. 1485.

Also available: Book plus combination disk (including all programs) for use on ATARI 400 or 800 with 16K. Combination Regularly $41.95. ANTIC Readers Save 15%. You pay Only $35.66. No. 5046.

Just Published

Advanced Programming Techniques for Your ATARI®, including Graphics and Voice Programs

by Linda M. Schreiber

Discover ATARI capabilities you never knew existed!

Now, you can move beyond the limitations imposed by BASIC to become master of your machine! With clear, step-by-step guidance from an ATARI specialist, you'll see how to use player/missile graphics, scrolling and screen flipping . . . enter a machine language subroutine to play music while a BASIC program is running . . . and a host of other unique programming tips and techniques. 244 pp./33 illus./7" × 10"/Hardbound/$19.95/No. 1545.

Also available: Book plus companion disk (including all programs) for use on ATARI 400 or 800 with 16K. Combination Regularly $44.95. ANTICS Readers Save 15%. You pay Only $38.21. No. 5047.

Buy Both Books, Save 15%!

TAB BOOKS Inc.
Blue Ridge Summit, Pa. 17214

All Orders Must Be Prepaid
(TAB pays postage and handling)

Please send me:

_____ Book No. 1485 ($21.95)
_____ Book/Disk Combination No. 5046 ($35.66)
_____ Book No. 1545 ($19.95)
_____ Book/Disk Combination No. 5047 ($38.21)
_____ Both Books at 15% Off Combined List Price No. 5045 ($35.62)

Check enclosed in the amount of $ _____

Charge my ☐ Visa ☐ MasterCard

Account No. _____ Expires _____

Name _____

Address _____

City _____ State _____ Zip _____

Signature _____

ALOG-103

ATARI LOGO
SON OF LISP

by Brian Moriarty

ATARI LOGO
by Logo Computer Systems
ATARI, Inc.
P.O. Box 61657
Sunnyvale, California 94086
16K Cartridge $99.95

"Logo with Turtle Graphics" is the hottest educational catchphrase since "Computer Literacy." BASIC cartridges across the nation are gathering chalk dust as teachers leap onto the Logo bandwagon. School departments will soon be making big purchasing decisions based on the availability of a good Logo system — and woe to the manufacturer who doesn't offer at least one.

Why is Logo so popular among educators? Reasons vary, but the two most often cited are its *simplicity* and its *versatility*.

Logo was designed from the ground up to be easy to learn. Developed by Dr. Seymour Papert at the Massachusetts Institute of Technology, Logo uses a direct, procedure-oriented command syntax that encourages logical thinking and good programming technique. Its turtle graphics system is a great motivator for young students, who delight in creating patterns and movement on a TV set with their own programs. Yet Logo belies its MIT heritage by incorporating many of the powerful list processing functions found in Lisp, Prolog and other so-called artificial intelligence languages. Logo's list-oriented structure makes it suitable for parsing, pattern recognition and other exotic applications that would be difficult to implement in a conventional language like BASIC.

ATARI has taken their sweet time about coming out with a Logo. But the new **Atari Logo** package looks as if it was worth the wait. The 16K cartridge-based language was developed for ATARI by Logo Computer Systems, Inc. (LCSI), a Toronto-based firm best known for its implementation of Logo on the Apple computer. As a result of this collaboration, the **Atari Logo** system and documentation have a clean, confident "feel" that is noticably lacking in ATARI's previous language releases.

Figure 1 shows all of the built-in commands or "primitives" in the **Atari Logo** vocabulary. These are the basic building blocks that are used to define new words called "procedures," which are in turn combined to form applications called "useful programs." Many BASIC hackers would sacrifice their little sister for the ability to define new commands. New commands are exactly what Logo programming is all about.

Figure 1. Atari Logo Vocabulary.

TURTLE GRAPHICS

ASK	BACK	BG
CLEAN	COLOR	CS
EACH	EDSH	FORWARD
GETSH	HEADING	HOME
HT	LEFT	PC
PE	PEN	PENDOWN
PENUP	PN	POS
PUTSH	PX	RIGHT
SETBG	SETC	SETH
SETPC	SETPN	SETPOS
SETSH	SETSP	SETX
SETY	SHAPE	SHOWNP
SPEED	ST	TELL
WHO	WINDOW	WRAP
XCOR	YCOR	

WORDS, LISTS & VARIABLES

ASCII	BUTFIRST	BUTLAST
CHAR	COUNT	EMPTYP
EQUALP	.EXAMINE	FIRST
FPUT	LAST	LIST
LISTP	LPUT	MAKE
MEMBERP	NAMEP	NUMBERP
SE	THING	WORD
WORDP		

MATH & LOGIC OPERATIONS

AND	COS	FALSE
INT	NOT	OR
PRODUCT	RANDOM	REMAINDER
RERANDOM	ROUND	SIN
SQRT	SUM	TRUE
+	—	*
/	<	=
>		

PROCEDURE CONTROL & EDITING

.CALL	COND	EDIT
EDNS	END	IF
OUTPUT	OVER	REPEAT
RUN	STOP	TO
TOUCHING	WAIT	WHEN

INPUT/OUTPUT

CT	FS	JOY
JOYB	KEYP	PADDLE
PADDLEB	.PRIMITIVES	PRINT
RC	RL	SETCURSOR
SETENV	SHOW	SS
TOOT	TS	TYPE

MEMORY & FILE MANAGEMENT

CATALOG	.DEPOSIT	ERALL
ERASE	ERF	ERN
ERNS	ERPS	LOAD
NODES	PO	POALL
POD	PODS	PONS
POPS	POTS	RECYCLE
SAVE	SETREAD	SETWRITE

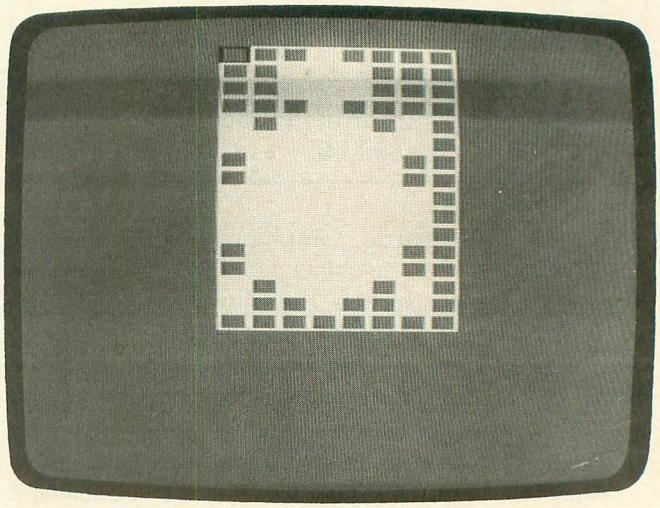
Raster reptiles.

Nearly one-third of the **Atari Logo** vocabulary is devoted to its turtle graphics system. The concept of turtle graphics may seem strange if you've been brought up on ATARI BASIC. Instead of PLOTting points on an X-Y grid and using DRAWTOs to describe lines, turtle graphics uses a "drawing pen" which is dragged around the TV screen by an imaginary entity called a "turtle." You can instruct the turtle which way to turn, how far to move, to pick up a pen, put it down or switch to another pen color. Turtles are easier for novices to grasp than PLOT-type graphics because their movements more closely resemble the familiar act of drawing on a piece of paper.

The turtle graphics in **Atari Logo** make the Apple, Tandy and Commodore Logos look primitive. By exploiting many of the ATARI's special hardware capabilities, including player/missiles and color indirection, the Logo cartridge offers a definitive implementation of the turtle graphics concept on an 8-bit micro.

Atari Logo supports four independent turtles and three drawing pens. You can control the speed, direction and absolute positioning of each turtle, individually or simultaneously. The turtle, pen and background colors can be set to any of 128 different hues. The system includes a .SETSCR command that controls the aspect ratio of the turtle's horizontal and vertical steps, and a WRAP command that determines whether or not turtles will "wrap around" if they move off the edge of the screen.

The turtles in **Atari Logo** actually look like little turtles. If you get tired of looking at them, you can use the HT (Hide Turtle) command to make them invisible, or type EDSH to enter the built-in shape editor. The editor lets you design and store up to 15 alternate turtle shapes. You can instantly change the shape of a turtle by TELLing it to assume one of your predefined designs. An enterprising Logo programmer could use this capability to create smooth animation and other special effects.



Shape Editor

Unlike BASIC, which lets you set up dozens of different graphics modes, **Atari Logo** allows only three screen formats: full text, full graphics with BASIC mode 7 resolution, or a split-screen mode with five lines of text on the bottom. Old-timers may consider this limitation somewhat restricting. But Logo isn't designed for hackers — it's for rank beginners who want to make pictures on their TV set without having to worry about graphics modes or color clocks.

The **Atari Logo** vocabulary includes a pair of interesting commands for the manipulation of sound, TOOT and SETENV. TOOT wins the award for best command name of the year. Its syntax is:

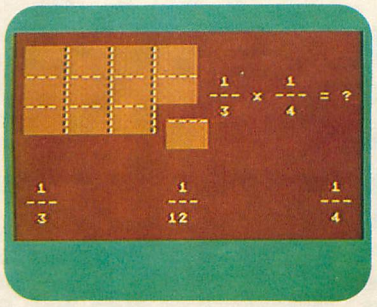
TOOT *voice frequency volume duration*

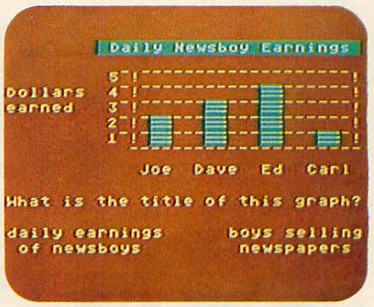
The "voice" parameter specifies one of Logo's two sound channels. Why just two, you ask? In order to improve the frequency range and resolution, Logo slaves together sound channels ½ and ¾. The benefit of this technique becomes apparent with the

(continued on page 148)

NEW! For Your Atari Computer

TRS 80 Color, I, III, 4 or Apple Computer





Over 1000 Programs with Full Time Audio Narration, Pictures & Text!

Interactive Tutorial Programs

All along you've heard there isn't much of a selection of low cost microcomputer courseware. Well, do you want to learn Physics, Psychology, or Philosophy? Accounting or Auto Mechanics? Sociology, Supervision or Statistics? Economics or Electronics? How about English as a Second Language? Would you like to teach your children Math or Reading? If you do, then we have the educational programs for you—and at affordable prices.

It's true there aren't many companies that offer full-length courses in subjects other than reading and arithmetic, and what is offered seems to be drills, tests, games, or simulations. What you really want is a course that covers the subject with, say, 16 full-length lessons called tutorial programs, where you interact with an expert programmer backed by a staff of experts. That's exactly what we have.

Is there something wrong with our software? Well, we don't ask you to enter your name so we can drop it into some later text. We don't ask you to type in your answer and refuse to accept it if it's not spelled just right. And, we don't branch around a lot when you make an error. Our programs simply let you know if you're wrong by proceeding only when you select the right multiple-choice answer.

This proven learning-by-positive-reinforcement method lets you proceed quickly and smoothly through the programs, without a lot of cute tricks. But, if you're a devoted computerist or game freak you may be disappointed at the lack of motion in most of the pictures, or our special visual effects. We have some dandies, like the mushroom cloud that rises over Hiroshima in our History series, but your learning is not distracted by needless special effects.

OVER 1000 PROGRAMS

We have 64 courses of 16 half-hour programs: 1024 programs! All are easy-reading, upper- and lower-case. All are in color. All are illustrated by frequent graphics composed of special and regular characters. And, best of all, every frame of every program is accompanied by high-quality, full-time audio narration by professional voice talent. David Stanton, James Mathers, Pam Barrymore. Recorded and played back, not synthesized or digitized!

— If your child is having trouble with Reading or Math, or if he or she is exceptional, and could benefit from professionally programmed lessons or courses, then you need our educational programs for your Atari, Apple, or TRS 80 to help your child.

— If you want to build your math skills, we have programs on Numbers and their meanings, Addition, Subtraction, Multiplication, Division, Fractions, Decimals, Percents, Angles, Graphs, Word Problems, Algebra, Statistics. There are ten series in all, 160 math programs. Again, all with color, pictures, and a friendly tutor's voice—all the time.

— If your verbal skills need polishing, try our Talk & Teach programs on the alphabet, spelling skills, and every level of vocabulary; Sight Words or Learn by Phonics; develop Reading Comprehension skills from stories and articles. There's even a series on the Great Classics.

— For self-development, try a 16-program course in Economics or Psychology. Or Supervision, Sociology, Counseling (Personal, School, or Employee), Writing, Business, Philosophy, Government, World History, U.S. History, or Accounting.

— For vocational skills, study our 16-program courses in Electronics, Fluid Power, Auto Mechanics, Shop, Carpentry, Construction, Meat Processing, Military Skills (64 programs), and there are many more.

There are 832 programs for your Atari, Apple and TRS 80 Mod I/III/4 and 160 for the Color Computer. Send for a free catalog.

Would you like to do something more than play games on your computer? Tired of squeaks and robotics and want to hear a human voice? Do you want to further your education or help your children along in theirs? Now you can. For just **$8.80**, we'll send you one cassette with 2 programs from the course of your choice. 100% guaranteed to work in your Atari, Apple, or TRS 80 computer, and you can try us out. Better yet, get one full course of 16 programs on 8 cassettes for only **$59.90**. Atari Master Cartridge, **$9.95**; Apple T/T Board, **$99.00**; and T/T Player, **$79.00**. That's less than books and tuition for most college classes. And we offer a 10 day, 100% exchange allowance. Does your alma mater? Send us your check, or call us at **1-800-654-3871** with your Visa/Master Card number. We'll send your educational software pre-paid. Please allow 15 days delivery. Or see your computer dealer. He may have some of our courses in stock. You've got nothing to lose but your games!

We're Your Educational Software Source.

Course	No. of Programs
Reading	256
Mathematics	128
Comprehension	48
History	32
Algebra	16
Spelling	16
Government	16

16 Programs in each of the following:
Carpentry-Electronics-Health Services
Office Skills-Statistics-First Aid/Safety
Economics-Business-Accounting-Psychology
And Many More!

For your Atari 400/600/800/1200, you will need the Atari cassette recorder and the Dorsett 4001 Educational Master Cartridge, $9.95. For your Apple II, you will need the Dorsett M402 T/T plug-in board, $99.00, and the M401 stereo cassette player, $79.00. All programs listed are available for TRS 80, I, III, 4, which require the M203 speaker converter, $99.00, and 401 stereo cassette player, $79.00.

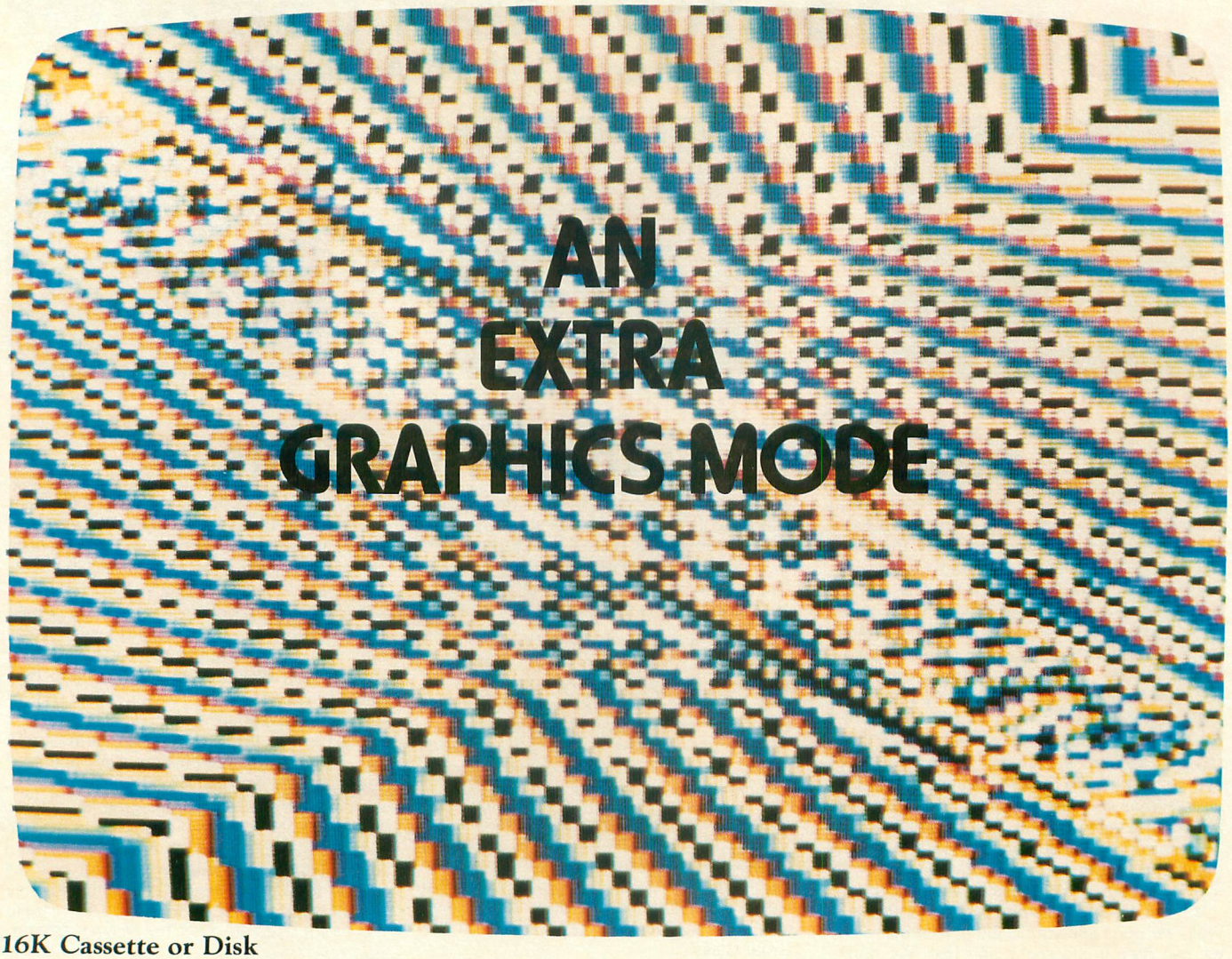
Send for a catalog of over 1000 programs for Atari, TRS 80, Apple, etc.

For more information, write or call: **TOLL FREE 1-800-654-3871**

Dealer inquiries welcome

VISA MasterCard

DORSETT
Educational Systems, Inc.
P.O. Box 1226, Norman, OK 73070



16K Cassette or Disk

by Peter C. Budgell

Including the hidden graphics modes, the ATARI supports a total of 6 text modes, 8 graphics modes plus 3 GTIA modes. However, by turning on the GTIA modes when not in GRAPHICS 8+16 (contrary to the operating system) other graphics modes appear. Most of these are useless, but in this article I implement a beautiful one.

It is called **Graphics 10/7** and features 80 by 80 pixels plus a text window, or alternately 80 by 96 pixels. It can display 7 colors simultaneously and occupies the 4K of a **Graphics 7** screen. The method to enter **Graphics 10/7** is to execute **Graphics 7**, then POKE 623,128, then POKE 87, 10. To enable the text window a display list interrupt is used (see **Listing 2**). Because of the DLI this mode cannot be used in the immediate mode, so always exit the program by hitting the SYSTEM RESET key.

Listing 1 is the demo program and it shows both the case with and the case without the text window. A simple array of lines gets lots of color onto the screen. Try changing the lines doing the plotting and assigning color to the color registers to examine this new mode.

The assignment of colors is a little odd and the reason lies somewhere inside the hardware. The following table gives the color registers and shows what color values for plotting (as in COLOR 3) will present the color register on the screen. In **Listing 1**, I simplify color choice by using the array "C" in what might be called "indirection-indirection." It is simpler!

Table 1.

Color Register	Color to access
704	0 & 1 & 4 & 5
705	2 & 6
706	3 & 7
707	nothing
708	9
709	10
710	11
711	nothing
712	8 & 12 & 13 & 14 & 15

Other interesting things can happen when the GTIA modes are switched on while in the other graphics modes and text modes. Experimentation is required, and not in all cases will the OS have to be fooled by POKEing location 87.

Have fun with these new capabilities in your machine. □

Listing 1.

```
10 REM  GRAPHICS 10/7 DEMO
11 DIM DLI$(32)
25 DIM C(6):C(0)=0:C(1)=2:C(2)=3:C(3)=
9:C(4)=10:C(5)=11:C(6)=8
30 GRAPHICS 7+16:GOSUB 40:GRAPHICS 7:G
OSUB 1000:GOSUB 40:GOTO 30
35 REM
40 POKE 623,128:REM GTIA GR.10
44 POKE 87,10:REM FOOL SCREEN HANDLER
45 REM
50 POKE 704,0:REM COLOR 0
60 POKE 705,12:REM COLOR 2
70 POKE 706,38:REM COLOR 3
80 POKE 708,98:REM COLOR 9
90 POKE 709,146:REM COLOR 10
100 POKE 710,200:REM COLOR 11
110 POKE 712,250:REM COLOR 8
120 REM
130 FOR CL=0 TO 79:TRAP 160
135 COLOR C(7*(CL/7-INT(CL/7)))
140 PLOT 0,0:DRAWTO CL,95
150 PLOT 79,95:DRAWTO 79-CL,0
155 PLOT 0,0:DRAWTO 79,CL
156 PLOT 79,95:DRAWTO 0,95-CL
160 NEXT CL
200 FOR X=1 TO 500:NEXT X:RETURN
1000 RESTORE 10000:FOR X=0 TO 31:READ
I:POKE ADR(DLI$)+X,I:NEXT X
2500 POKE PEEK(560)+256*PEEK(561)+84,1
41
2560 POKE 513,INT(ADR(DLI$)/256):POKE
512,ADR(DLI$)-PEEK(513)*256
2570 POKE 54286,192
2900 ? "      THIS IS GRAPHICS 10/7":?
"     P. BUDGELL 1983"
3000 D=PEEK(16)-128:IF D<0 THEN RETURN
3010 POKE 16,D:POKE 53774,D:RETURN
10000 DATA 72,169,0,141,10,212,141,27,
208,141,26,208,169,144,69,79,37
10001 DATA 78,141,24,208,169,10,69,79,
37,78,141,23,208,104,64
```

•

CHECKSUM DATA

(See p. 58)

```
10 DATA 317,950,776,484,270,801,217,27
2,536,282,312,350,753,689,876,7885
120 DATA 80,513,105,510,775,524,828,65
2,555,247,905,630,492,788,937,8541
3010 DATA 362,210,537,1109
```

•

Listing 2.

```
0100 ;  DLI FOR GRAPHICS 10/7
0110 ;
0120  *=$600
0130 ;
0140 COLPF1=$D017 ; THE TEXT
0150 COLPF2=$D018 ; THE WINDOW
0160 COLBK =$D01A ; THE BORDER
0170 W5YNC =$D40A ; WAIT FOR SYNC
0180 COLR5H=$4F   ;  ATTRACT
0190 DRKM5K=$4E   ;  MODE
0200 PRIOR =$D01B
0210 ;
0220  PHA
0230  LDA #0
0240  STA W5YNC   ; WAIT HORIZ SYNC
0250  STA PRIOR   ; SWITCH OFF GTIA
0260  STA COLBK   ; BLACK BORDER
0270  LDA #$90
0271  EOR COLR5H  ;    FOR
0272  AND DRKM5K  ;    PROTECTION
0280  STA COLPF2  ; DARK BLUE WINDOW
0290  LDA #$0A
0291  EOR COLR5H  ;    FOR
0292  AND DRKM5K  ;    PROTECTION
0300  STA COLPF1  ; WHITE TEXT
0310  PLA
0320  RTI         ; RETURN FROM INTER.
```

•

ATARI®
MEANS BUSINESS

BUSINESS PACKAGES FOR $195.00 EACH

For the first time, Financial Software Plus introduces businesss packages for the Atari 400 or 800 computer that simulates business packages found on much larger and expensive computer systems. All our packages are written by an accountant-programmer and program designer so that you can be assured that our packages will meet your individual requirements.

INVENTORY

- Capacity of 500 part numbers, 250 customers, 250 vendors per disk.
- Maintains customer files, vendor files, and sales history by month and year-to-year date.
- Generates purchase orders, invoices, quotations, regular mailing labels, and shipping labels.
- Reports Include: Quantity in stock, quantity on order, backorder, reorder, and sales activity.
- Ability to limit reports by part numbers and vendor or both.

PAYROLL

- Capacity of 100 Employees per disk.
- User changeable tax codes-five different tax tables may be set up.
- Miscellaneous earnings and deductions categories are user specified and changeable at any time.
- Prints checks, W2's, month-end, quarterly, annual and tax reports.
- Fixed or variable deduction capability.

COMMON FEATURES

- Will run on a one or two disk drive system at any time.
- All data disks are created automatically-unlimited storage capacity.
- All reports can be printed to the screen or printer-printer is optional.
- Can act as a stand alone package or a fully integrated system with our general ledger package.

Stop on down to your nearest dealer and ask for a demonstration.

General Ledger, Accounts Payable, and Accounts Receivable Packages Available Soon.

FINANCIAL SOFTWARE PLUS

121 WEST CEDAR
KALAMAZOO, MICH. 49007
(616) 345-8546

DISTRIBUTOR AND DEALER INQUIRIES WELCOME

FINE SCROLLING PART II

16K Cassette or Disk

by Kyle Peacock

Well, you can stop holding your breath. We're ready to continue our discussion of fine scrolling. This installment will deal with horizontal scrolling. By the end of Part II, we'll be able to selectively smooth scroll any line of text (or bit mapped graphics) to the left or right. This effect is accomplished with three simple steps:

1st: A slightly modified display list.
2nd: An organized memory map for screen display.
3rd: Changing a hardware register called HSCROL (location: 54276, $D404 hex)

You should already be versed in setting up customized display lists, so I'll just mention a few things to add to your existing knowledge. The most important display list instruction needed to accomplish horizontal scrolling is the Load Memory Scan (LMS). This 3-byte instruction tells ANTIC where to "fetch" its screen RAM. Last issue, we found that by adding 64 ($40 hex) to a display list opcode and tacking on a two byte address in lo-byte, hi-byte fashion we got an LMS instruction. Now let's go one step further and set the "horizontal scroll bit" on a display list opcode.

For those of you who feel a wee bit (no pun intended) confused when I say it's time to do some bit setting, have no fear. All that needs to be done to set the horizontal scroll bit of a display list opcode is to add the decimal number 16 ($10 hex) to that instruction. Any display list instruction that has its horizontal scroll bit set will smooth scroll upon changing the hardware register HSCROL. Let's take a look at what I mean.

ANTIC Instruction		Explanation
Decimal	**Hex**	
02	$02	Display 1 GR.0 line of text
+16	+$10	Add on to set horizontal scroll bit
18	$12	New ANTIC instruction
66	$42	GR.0 LMS instruction
xx	XX	LO-byte of screen RAM address
yy	YY	HI-byte of screen RAM address
+16	+$10	Add on to set horizontal scroll bit
82	$52	New ANTIC instruction
xx	$XX	LO-byte of screen RAM address
yy	$YY	HI-byte of screen RAM address

Note that only the opcode was changed, NOT the lo and hi bytes pointing to the screen RAM.

Here's a short BASIC program that sets the horizontal scroll bit on the first three lines of a graphics mode zero screen, and smoothly scrolls the text on these lines.

Listing 1.

```
10 REM ********************************
20 REM *   HORIZONTAL SCROLL DEMO     *
30 REM *         LISTING #1           *
40 REM *     BY KYLE S. PEACOCK       *
50 REM *    ANALOG COMPUTING #14      *
60 REM ********************************
70 REM
80 GRAPHICS 0
90 FOR X=0 TO 10
100 PRINT "THIS IS A TEST OF HORIZONTA
L SCROLL"
110 NEXT X
```

```
120 REM
130 REM --- FIND THE DISPLAY LIST
140 REM
150 DLIST=PEEK(561)*256+PEEK(560)
160 BIT1=PEEK(DLIST+3)
170 BIT2=PEEK(DLIST+6)
180 BIT3=PEEK(DLIST+7)
190 REM
200 REM --- NOW ADD 16 (HEX: $10) TO
210 REM --- DISPLAY LIST INSTRUCTIONS
220 REM
230 POKE DLIST+3,BIT1+16
240 POKE DLIST+6,BIT2+16
250 POKE DLIST+7,BIT3+16
260 REM
270 REM --- HORIZONTAL SCROLL
280 REM
290 FOR X=0 TO 7
300 POKE 54276,X
310 FOR T=0 TO 100:NEXT T
320 NEXT X
330 GOTO 290
```

•

CHECKSUM DATA
(See p. 58)

```
10 DATA 771,951,963,487,621,781,263,16
6,278,279,761,80,396,86,210,7093
160 DATA 142,151,158,101,230,354,82,31
5,325,331,94,0,100,330,335,3048
310 DATA 531,768,725,2024
```

•

As mentioned in Part I, ANTIC sets up its GR.0 display list somewhere near the top of available memory. The 1st byte of the display list's address is stored in LO-byte, HI-byte at locations 560 and 561 ($230/231 hex). Experience tells us that a GR.0 display list looks something like:

Display List Instruction

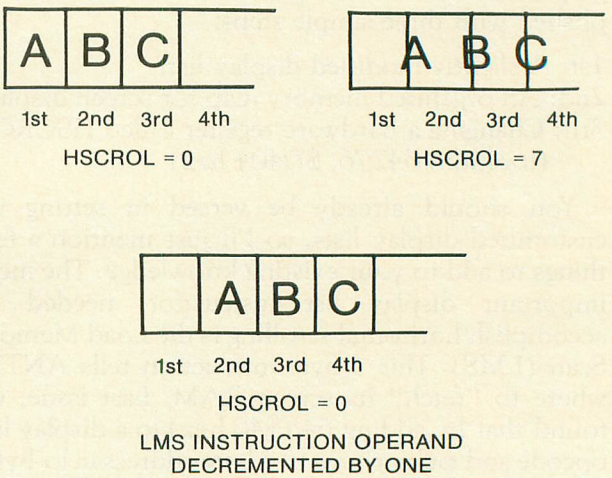
Decimal	Hex	Explanation
112	$70	Blank 8 Scan Lines
112	$70	Blank 8 Scan Lines
112	$70	Blank 8 Scan Lines
*66	$42	GR.0 LMS Instruction
xx	$XX	LO-byte of screen RAM
yy	$YY	HI-byte of screen RAM
*02	$02	Display a GR.0 line (where ANTIC last left off)
*02	$02	Display another GR.0 line (Remainder of Display List)

By setting the horizontal scroll bit of the instructions with an * next to them, the characters on these lines will be shifted "N" color clocks (or positions) to the right. ("N" being the value stored in HSCROL.) To make our display scroll to the left, change line 290 to read: FOR X = 7 TO 0 STEP -1.

Three things become apparent when looking at the display generated once the horizontal scroll bits are set. First, the text on the screen appears shifted over slightly. (There might even be a little garbage in the lower right corner of the screen.) When horizontal scrolling is enabled ANTIC needs more bytes of data for the screen RAM. Instead of using 40 bytes for a GR.0 line of text, it uses 48 bytes. Consequently, the entire display gets shifted over 8 bytes for every line having its horizontal scroll bit set. This problem will be remedied in **Listing 2.**

Second, you may notice how the text on the first three lines smooth-scrolls for a while then "jumps" back to its original position and repeats again. Let's think about what ANTIC is doing. By changing the value of HSCROL from zero all the way to seven, we shift the text to the right. When HSCROL has a value of 0, the 1st character on the line is positioned next to the second, the second character next to the third, and so on. When HSCROL finally gets a value of 7, the 1st character is *very nearly* where the second character *used to be*, the 2nd character is *close to* where the 3rd character *used to be*, etc. When HSCROL goes back to 0 again the characters resume their normal positions. To achieve true horizontal scrolling, we should reset HSCROL back to zero AND decrement the operands on our LMS instructions (or increment if we're scrolling to the left).

Gee Kyle, that last paragraph flew right over my head. How about an illustration? Sure!

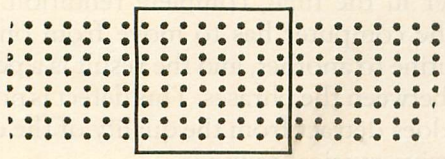


You may notice some occasional screen flicker while the program smooth-scrolls the text. This is due to the way ANTIC generates its display and the way BASIC executes a program. If we try to change HSCROL while ANTIC is drawing a line that has its horizontal scroll bit set, the screen flickers. This problem is a little more difficult to solve and requires some assembly language programming. Let's save this problem for my last installment, "Fine Scrolling Part IV: Taking The Plunge."

You may be wondering why **Listing 1** only used a scrolling range of 0 to 7. Well, devoted reader, the value stored in HSCROL tells ANTIC how many half-color clocks to shift the display. Graphics 0 characters are only 8 half-color clocks wide. That's why I restricted the scroll to no more than seven units.

Hey Kyle, can we get rid of that sloppy 8-byte shift?

We can accomplish this by setting up a display list where every line consists of an LMS instruction and the range of displayed memory on the LMS operands is 256 bytes (exactly one page). Then our screen and screen RAM will look something like this:



Screen arrangment.

By arranging our display list in this fashion and creating an organized memory map, the screen becomes a smaller window on a larger piece of memory. We can then change the value of HSCROL and update the LMS operands of our display list, and VOILA! Horizontal scrolling is achieved.

As an example of what I mean, here is **Listing 2**, exemplifying the fine horizontal scrolling capabilities of the ATARI computer. □

Listing 2.

```
10 REM ********************************
20 REM *   HORIZONTAL SCROLL DEMO     *
30 REM *         LISTING #2           *
40 REM *      BY KYLE S. PEACOCK      *
50 REM *    ANALOG COMPUTING #14      *
60 REM ********************************
70 REM
80 REM - CLEAR OUT MESSAGE SPACE
90 REM
100 DIM MES$(60):? "{INITIALIZING..."
;
110 FOR X=0 TO 256
120 POKE 49*256+X,0
130 POKE 50*256+X,0
140 POKE 51*256+X,0
150 POKE 52*256+X,0
160 POKE 53*256+X,0
170 NEXT X:POKE 712,64
180 REM
190 REM - READ WHERE IN MEMORY
200 REM - MESSAGE GOES, ADD ON FOR
210 REM - COLOR, & MESSAGE. THEN
220 REM - INSTALL INTO MEMORY
230 REM
240 FOR X=1 TO 5
250 READ WHERE,PLUS,MES$
260 FOR Y=1 TO LEN(MES$)-2
270 POKE WHERE+Y-1,ASC(MES$(Y+1,Y+1))
-32+PLUS
280 NEXT Y:NEXT X
290 REM
300 REM - NOW READ & INSTALL
310 REM - CUSTOMIZED DISPLAY LIST
320 REM
330 TRAP 380
340 READ DLIST
350 POKE 6*256+ADD,DLIST
360 ADD=ADD+1
370 GOTO 340
380 REM
390 REM - NOW TELL ANTIC WHERE TO
400 REM - FIND OUR NEW DISPLAY LIST
410 REM
420 POKE 560,0:POKE 561,6
430 REM
440 REM - CHANGE VALUE OF 'HSCROL'
450 REM
460 FOR X=7 TO 0 STEP -1
470 POKE 54276,X
474 REM
475 REM - LOOP VALUE OF 'Y' DICTATES
476 REM - SPEED OF HORIZONTAL SCROLL
477 REM
480 FOR Y=0 TO 50:NEXT Y:NEXT X
490 REM
500 REM - NOW UPDATE LMS INSTRUCTIONS
510 REM
520 LMS1=PEEK(6*256+11)+1
530 LMS2=PEEK(6*256+25)+1
534 REM
540 REM - TEST FOR WRAP AROUND
545 REM
550 IF LMS1>60 THEN LMS1=0:LMS2=0
560 POKE 6*256+11,LMS1
570 POKE 6*256+25,LMS2
580 GOTO 460
600 REM
610 REM - MESSAGES
620 REM
800 DATA 12544,64
810 DATA "HORIZONTAL SCROLLING"
820 DATA 12820,192
830 DATA "NEXT TIME WE'LL EXAMINE VERT
ICAL SCROLLING"
840 DATA 13056,128
850 DATA "    A.N.A.L.O.G."
860 DATA 13332,192
870 DATA "SEE YOU IN SIXTY DAYS..."
880 DATA 13568,64
890 DATA "     ISSUE #14"
899 REM
900 REM - DISPLAY LIST DATA
910 REM
920 DATA 112,112,112
930 DATA 70,00,49
940 DATA 112,112,112,112
950 DATA 86,00,50
960 DATA 112,112,112,112
970 DATA 70,00,51
980 DATA 112,112,112,112
990 DATA 86,00,52
1000 DATA 112,112,112,112,112
1010 DATA 70,00,53
1020 DATA 65,00,06
```

●

CHECKSUM DATA

(See p. 58)

```
10 DATA 771,951,966,487,621,781,263,49
5,267,22,352,653,631,637,643,8540
160 DATA 649,153,98,251,732,160,57,85,
314,781,595,50,542,103,827,5397
310 DATA 685,84,731,61,235,969,724,102
,514,758,83,311,89,585,95,6026
460 DATA 59,358,105,993,93,108,389,107
,414,85,207,222,95,324,99,3658
550 DATA 358,1,20,739,84,563,90,56,632
,259,192,269,183,277,678,4401
880 DATA 99,136,124,775,93,303,55,803,
56,809,45,815,74,93,163,4443
1020 DATA 154,154
```

●

Rhymes & Riddles
Spinnaker Software
215 First Street
Cambridge, MA 02142
48K Disk, $29.95

by Edward Bever

I remember playing the word game "hangman" and its commercial derivatives pretty often when I was a child. In the game's simplest form, one kid would draw a gallows and a blank for each letter in a word, and the other kid would try to guess the letters. If the guesser guessed right, the other kid had to fill in all instances of the letter; if the guesser guessed wrong, the other filled in a feature of a figure hanging from the gallows. If the guesser could guess the word before the figure was complete, he won; if not, he lost.

Rhymes and Riddles contains a set of three educational computer games based on "hangman" for children ages five to nine. In the first game, the computer displays blank spaces for the letters in a line of a nursery rhyme; in the second, it presents a riddle and the blanks corresponding to the answer; in the third, it is looking for the letters that form a famous saying. The child enters a guess by pressing the letter on the keyboard. If it is right, the computer fills it in; if it is wrong, the computer adds it to the list of wrong answers, and adds a feature to a melancholy face. In all three games, the child must fill in the blanks before making six wrong guesses. If the child makes a sixth, the computer completes the sad face, emits a mechanical gnash, flashes a discouraged message, and fills in the remaining blanks.

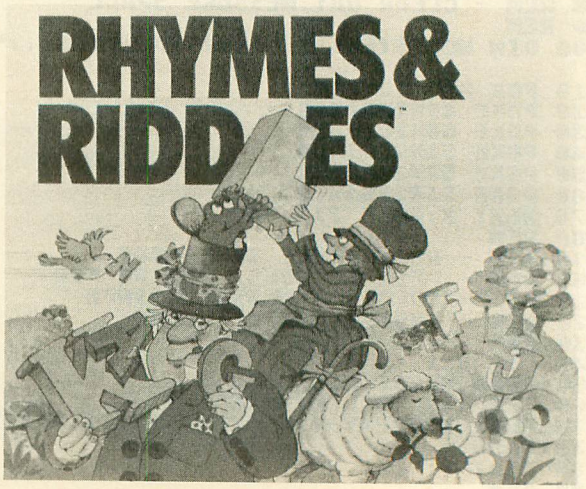
In both the riddle and famous saying games, the hidden message is a one liner. If the child successfully completes it, the computer rewards him or her with a display of sounds and graphics: music and a happy face, jet planes roaring by, or a swirl of lines and a metallic hiss of approval. In the nursery rhyme game, the passage is longer, and the computer rewards successful completion of each line with a smiling face and a musical rendition of the portion just completed. Whether the child succeeds or fails, after its response the computer proceeds as before. If the child successfully completes the last line, the computer will play the entire song, and display a drawing based on it. If not, the machine goes through its standard lament.

Overall, the package is well done. It comes in Spinnaker Software's commendable standard container: a flat plastic box that is sturdy and closes securely, yet is easy for small hands to open. The documentation is concise and comprehensive. The graphics are well done, and the musical arrangements are impressive. A simple menu appears automatically when the drive boots the disk, and the computer randomly picks one of what appears to be a large number of possibilities when a game is selected. The machine requests and uses the child's name, which is especially exciting to young children. The program will only respond to letter keys, and, in contrast to Spinnaker's otherwise excellent **Kindercomp**, the program has yet to reveal any hidden bugs.

Rhymes and Riddles does have a number of minor flaws. To begin with a trivial one, the documentation advises the user to turn off the disk drive before removing the disk, but I have heard that it is better to remove disks first. More seriously, while the musical phrases that reward each line of the nursery rhymes are quite good, they do not flow together in the final, complete rendition. Presumably, the computer has to move from one BASIC subroutine to another, and the result is a perceptible pause between the phrases. This defect is not critical, but it does detract from the quality of the otherwise impressive musical rewards.

My most serious reservation about the package is the way it responds to a child's failure. I think that mistakes during learning exercises should not be disparaged, but instead treated as normal, even necessary. When a kid knows what to do but willfully refuses, a rebuke may be in order, but when he or she has difficulty mastering a task to begin with, remorse is as inappropriate as censure. A crisp, unreproving signal would be preferable to the programs' mechanical lamentations.

None of these drawbacks is critical, and on balance **Rhymes and Riddles** is a very well done product: educational and entertaining, easy enough for a child to use alone, and diverse enough to long retain its value. My biggest problem with the package is that my son does not read well enough yet to really enjoy it. However, he is still slightly younger than the suggested age range, so I have no one to blame but my over-eager self. Fortunately, educational software is like hand-me-down clothes; if it is too big now, it will fit fine next year. □



ComputAbility™

Your ATARI Discount Software Store That's As Close As Your Phone!!

ATARI 800 48K	**CALL**
ATARI TOUCH TABLET	69.95
ATARI TRAK-BALL	49.95
COMMUNICATOR II	CALL
REMOTE CONTROL JOYSTICK	CALL
ATARI 800XL 64K	CALL
ATARI 1050 DISK DRIVE	CALL
ATARI 850 INTERFACE	CALL
ATARI 1010 RECORDER	77.00
HOME MANAGER KIT	64.95
ATARI 1027 PRINTER	289.00

MOSAIC

64K Ram/400	149.00
64K Ram + Cable Kit/400/800	169.00
48K Ram Kit	94.00
16/32 Expander	64.95
32K Ram	77.95
Mosaic Adaptor	49.95
64K Ram Utility Software - Superdrive & Handyman	Call!!
Koala Touch Tablet	89.95

DISK DRIVES: **PERCOM, RANA, TRAK, ASTRA** ..... CALL FOR PRICES.

"Advanced Interface Devices"
Programmable Printer Interface
w/4K Buffer 139.00

PRINTERS

PROWRITER 8510 AP ... CALL
AXIOM AT-100
W/PARALLEL PORT.... 269.00
GEMINI 10X ..... CALL
OKIDATA PRINTERS .... CALL

FREE* SOFTWARE AND ACCESSORIES

SEE STOCKING STUFFING AD ON PAGE 119 FOR COUPON PROGRAM

ATARI

Conversational Languages - T	43.95
Programming 2 & 3 - T	21.95
Music Composer - C	32.95
My First Alphabet - D	26.95
Touch Typing - T	19.95
Home Filing Manager - D	37.95
Mailing List - T	19.95
Asteroids - C	26.95
Caverns of Mars - D	28.95
Computer Chess - C	26.95
Missile Command - C	26.95
Super Breakout - C	26.95
Star Raiders - C	32.95
Assembly Editor - C	44.95
Basic - C	39.95
Macroassembler - D	65.95
Microsoft Basic - D	65.95
Pilot (Home Package) - C	58.95
Invitation to Programming I - T	18.95
Speed Reading - T	55.95
Basketball - C	26.95
Graph-it - T	15.95
Juggles House - D/T	22.95
Pilot (Educator) - C	97.95
Video Easel - C	26.95
Defender - C	32.95
Galaxian - C	32.95
Qix - C	32.95
Dig Dug - C	32.95
ET - C	37.95
Timewise - D	23.95
Atariwriter - C	74.95
Donkey Kong - C	37.95
Ms. Pac-Man - C	39.95
Tennis - C	35.95
Eastern Front - C	32.95
Donkey Kong Jr. - C	39.95
Pengo - C	35.95
Logo - C	79.95
Robitron - C	35.95
Pole Position - C	39.95
Microsoft Basic II - C	65.95
Paint - D	33.95
Caverns of Mars - C	32.95

PARKER BROS.

Q-Bert - Cart	34.95
Frogger - Cart	34.95
Astro Chase - Cart	34.95
Popeye - Cart	34.95
Tutankhamen - Cart	34.95
Risk - Cart	39.95
Chess - Cart	39.95

SUPER SPECIALS

ANY SYNAPSE ARCADE TITLE

$22.95 ..... DISK/TAPE
$29.95 ..... CARTRIDGE

PRICES EFFECTIVE NOW THROUGH DECEMBER 31, 1983

THORN

War Games - Cart	31.95
Hockey - Cart	31.95
Soccer - Cart	31.95
ORC Attack - Cart	31.95
Submarine Commander - Cart	34.95
Jumbo Jet Pilot - Cart	34.95
River Rescue Cart	31.95

SPINNAKER

Up For Grabs - Cart	31.95
Cosmic Life - Cart	27.95
Facemaker - Cart/Disk	27.95
Alphabet Zoo - Cart	27.95
Fraction Fever - Cart	27.95
Kids on Keys - Cart	27.95
Delta Music - Cart	CALL
Delta Drawing - Cart	CALL
Story Machine - Disk	27.95
Rhymes & Riddles - D	23.95
Amazing Things - D	31.95
Hey Diddle Diddle - D	23.95
Snooper Troops 1 or 2 - D	34.95

ADVENTURE INTERNATIONAL

Preppie - D/T	23.95
S.A.G.A. Adventures - D	31.95
Sea Dragon - D/T	27.95
Stratos - D/T	27.95
Bugoff - D/T	23.95
Prepple II - D/C	27.95
Stone of Sisyphus - D	31.95
Eliminator - D/T	20.95
Rally Speedway - Cart	37.95
Ultra Disassembler - D	39.95
Diskey - D	39.95
Triad - D/T	27.95

SIRIUS

Bandits - D	27.95
Way Out - D	31.95
Repton - D	31.95
Twerps - D	27.95
Blade of Blackpoole - D	31.95
Type Attack - D	31.95
Beer Run - D	27.95
Critical Mass - D	31.95
Wavy Navy - D	27.95

CBS

Solar Fox - Cart	31.95
Blue Print - Cart	31.95
Wings - Cart	31.95
Omega Race - Cart	31.95
Domino Man - Cart	31.95
Satan's Hollow - Cart	31.95
Madden's Football - Cart	31.95
K-Razy Shoot-Out - Cart	29.95
K-Razy Antiks - Cart	29.95
K-Star Patrol - Cart	29.95
K-Razy Kritters - Cart	29.95

FIRST STAR

Astro Chase - D/T	23.95
Bristles - D/T	23.95
Flip Flop - D/T	23.95
Rent Wars - D/T	23.95

DATASOFT

Pooyan - D/T	23.95
Moon Shuttle - D/T	23.95
Sands of Egypt - D	23.95
Text Wizard - D	37.95
Spell Wizard - D	37.95
Basic Compiler - D	59.95
Teletalk - D	37.95
Canyon Climber - D/T	20.95

SEGA

Buck Rogers - Cart	31.95
Star Trek- Cart	31.95
Tac Scan- Cart	31.95
Congo Bongo- Cart	31.95

EPYX

Gateway To Apshai - Cart	31.95
Pitstop - Cart	31.95
Lunar Outpost - Cart	31.95
Swat Rescue - Cart	31.95
Silicon Warrior - Cart	31.95
Fun With Music - Cart	31.95
Jumpman Jr. - Cart	31.95
Dragonriders of Pern - D/C	31.95
Temple Of Apshai - D/C	31.95
Hellfire Warrior - D/C	23.95

INFOCOM

Deadline - D	37.95
Witness - D	37.95
Zork I, II, III - D	31.95
Starcross - D	31.95
Planetfall - D	37.95
Suspended - D	37.95

MISCELLANEOUS

Spelunker - D	31.95
Ken Uston's Blackjack- D	54.95
Beta Lyrae - D	27.95
Cryptocube - D	31.95
Spellicopter - D	31.95
Creature Creator - D	31.95
Pinball Construction - D	31.95
Night Mission - D/T	23.95
Letter Perfect - D	99.95
Juice - D/T	23.95
Castle-Wolfenstein - D	23.95
Monkey Wrench II - Cart	49.95
PM Animator - D	27.95
Teletari - D	31.95
Starbowl Football - D/T	25.50
Data Perfect - D	79.95
Ultima II - D	44.95
Crisis Mountain - D	27.95
Star League Baseball - D/T	25.50
Home Accountant - D	54.95
Omnimon	79.95

D - Disk T - Cassette C - Cartridge

ATARI is a trademark of ATARI, Inc.

We Carry Hundreds of Items for ATARI 400/800, Ask for Our FREE CATALOG.

Mastercard/VISA Order Toll Free MasterCard **800-558-0003** VISA **In Wisc. Call 414/351-2007**

No surcharge for credit cards

COMPUTABILITY
P.O. Box 17882
Milwaukee, WI 53217

ORDERING INFORMATION: To order by mail, send money order, certified check, or personal check (allow 14 days to clear) to COMPUTABILITY. Include $2.00 shipping on all software orders. Include 3% shipping on all hardware orders, minimum $2.50. Mastercard and Visa please include card number and expiration date. WI residents please add 5% sales tax. Canadian, APO & FPO software orders include 5% shipping minimum $5.00. All other foreign software, please add 15% shipping, minimum $7.00 (US Funds only). Prices subject to change without notice.

ORDER HOURS
Mon-Fri 12-9 PM (CST)
Sat 12-5 PM (CST)

GETAWAY
by Mark Reid
ATARI PROGRAM EXCHANGE
P.O. Box 3705
Santa Clara, CA 95055
32K Cassette/Disk $29.95

by Kyle Peacock

Most games nowadays are your basic shoot-em-up. No matter how you cut it, by the time the screen reads GAME OVER, someone has "bought the farm." Destroying wave upon wave of aliens may sound sadistic, but it's the law of the universe.

Mark Reid has managed to bring that sadistic edge planetside. The object of **Getaway** is to drive your gangster-mobile through the busy streets of a large city, picking up various items with different cash values. The game begins with your gangster-mobile parked at your hideout. Here you are protected against the police. Once you depart the safety of your refuge, you must contend with various police cars, stop signs (which constantly relieve you of any on-hand cash), and road blocks (which cause leaks in your gas tank). And if this isn't enough to make you turn yourself in, you must constantly be wary of the amount of gas in your tank. Should you run dry during a high speed chase you'll receive a long and rather embarrassing trip up the river.

The stash.

Getaway may sound like it's oriented towards the long arm of the law, but it does include several thief-oriented goodies. Large dollar bills, diamonds, crosses, hearts, goblets, rings and magic wands lie in the middle of the roadway. Driving over these items allows your on-hand cash to accumulate. At first, the police aren't very interested in your illegal activities. But as your cash increases you'll soon find yourself being pursued by several cruisers and paddy wagons. Should you be caught before reaching your hideout, you lose all your cash and are carted off to jail. Since you only start with three gangster-mobiles it's not a good idea to get caught.

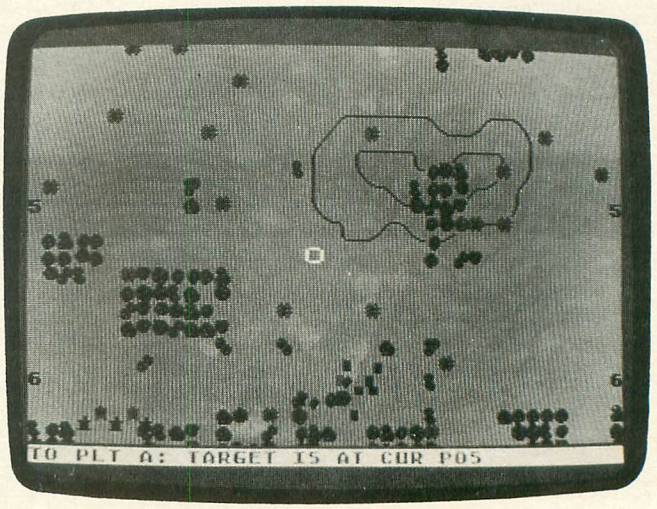
Probably the best item in **Getaway** is a roaming armored truck. You can bet this little white vehicle is stuffed to the brim with money. However, pulling off this heist puts your picture on the front page of the paper and in hot water with the cops. The chase that ensues reminds me of a multi-screen **Pacman** (without those life-saving power pellets).

COMBAT LEADER
by David Hille
SSI/Rapidfire
883 Stierlin Rd., Bldg. A-200
Mountain View, CA 94043
48K Disk/Cassette $39.95

by Pat Kelley

"Attention all units. This is a message from Battalion HQ. At 04:00 hours, all units will proceed to a previously designated area and engage enemy forces. This movement is vital to the survival of our forces in Western Europe, and the survival of the free world. Good luck, and Godspeed."

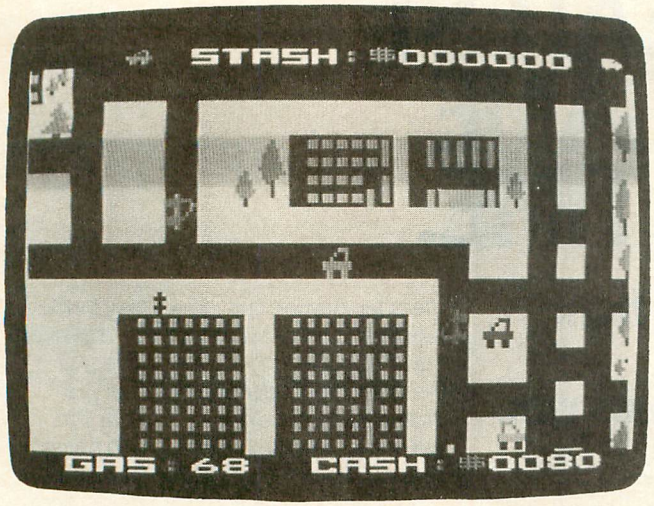
Although the beginning of this review may sound like very serious stuff, I think it is in keeping with the tone of the new SSI strategic/action game **Combat Leader**. This game is quite innovative in many respects, and is refreshingly different than the other "tactical simulations" I have seen for the ATARI Computer. In **Combat Leader**, you are in command of a full-combat unit comprised of tanks, armored troop carriers, support vehicles, and anti-tank troops. You can choose the speed and mode of play that best suits you. You and your men can take the offensive, pushing forward into enemy territory on a



Combat Leader

search-and-destroy mission. You collect points for the number of enemy units you eliminate. Or, if you like, you can command a troop of fast-reconnaisance vehicles to probe enemy lines or gather intelligence for your unit. The game also offers another interesting feature: the ability to construct your own battlefield. With this, you can specify to the letter the strengths/weaknesses of your opponents.

If you feel like taking the upper hand, you can instruct the computer to give your enemy only a token force of armor, infantry or firepower, thus



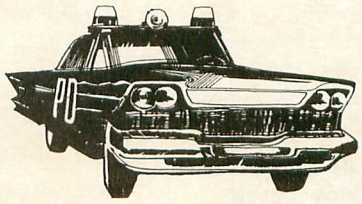
Getaway

Darkness falls over the city.

An interesting feature of **Getaway** is the ever-increasing level of difficulty. As day turns to night the police get much craftier. Most criminals would appreciate the darkening skies, but your gangster-mobile isn't equipped with headlights so the road becomes increasingly more difficult to see. The police usually nab you while speeding around corners, and they have a nasty habit of knowing the city inside out.

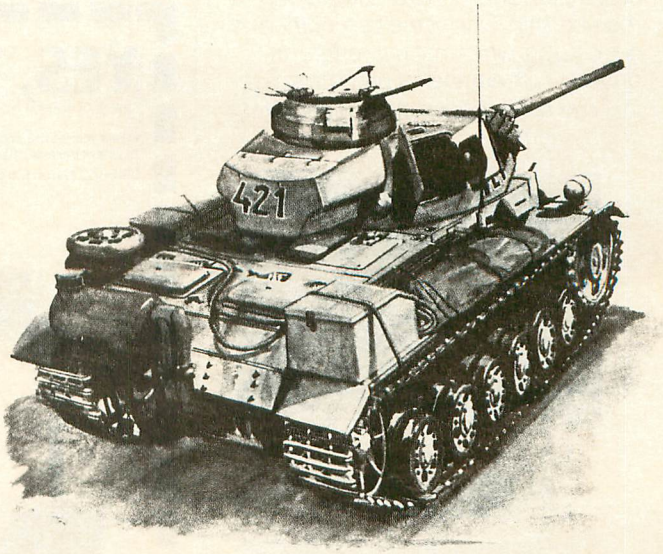
The city itself consists of thirty-five different screens. Your gangster-mobile is always positioned in the screen's center while the city smooth scrolls about in **Eastern Front** type fashion. Mark must have spent an incredible amount of time just redefining his character set. The city includes a warehouse district, high school, golf course, airport, even an attractive blue river. (Now we *know* Mark's city wasn't fashioned after New York.) And of course he couldn't forget the usual complement of trees, grass and houses.

Getaway is a great game. It utilizes many of the ATARI graphic and sound capabilities. As mentioned earlier, it is a bit on the sadistic side. It brainwashes you into thinking it's okay to break the law. But in the end, you'll get caught, put on ice, and come to the bone-chilling realization that crime doesn't pay. □



stacking the deck in your favor. Or, if you prefer the General McAuliffe method of combat you can defend your territory against numerically superior aggressors, and garner points by keeping your casualties at a minimum. Another interesting feature is a chart provided with the game, giving you all the specifications on current and historical tanks. This is great if you wish to construct a game that pits current day Abrams M-1s against Hitler's Panzers.

The game's graphics and playability are quite good, enhanced by detailed sound effects. When your tanks move out, you can hear the rumble of the engines, along with the whistle of incoming mortar fire and the crack of small-arms when in combat. You control the movement of your forces by cursor — simply place the cursor over the area on which you wish your forces to go and issue the command "GO CURSOR." Your forces will behave splendidly, and move out at a brisk pace across the full-scrolling terrain map. But, if you feel the progress they are making is not fast enough, simply issue the command "HURRY." This will speed up the assault, but cut down their ability to deliver suppressing fire. I promise you this — the frustration you will feel as you watch one of your advances become bogged down will give you a level of involvement not found in your average comuter game. Watching your units become cannon fodder for your adversary must simulate the frustration that military commanders throughout history have felt on the battlefield. This is one game that you will find hard to walk away from without first dealing your computer-controlled adversary at least a minor drubbing. If you are a budding Patton, Rommel or Giap at heart, then SSI's **Combat Leader** may be the one for you. Move out! □



THE GREATEST ATARI® GAME OF ALL TIME.

Object: Capture more programs than from any other source. Score: The best prices for programs win.

Actually, every member of CompuClub is a winner, because no one pays lower prices for Atari® programs than our members.

And no other source offers as many programs, with a descriptive catalog covering every piece of software we offer!

CompuClub has hundreds of Atari® programs: games, education and business. It's an astonishing selection, but just as incredible are our prices and our catalog.

Prices always at least 25% below retail

Fact is, usually our prices on selected programs are even lower than that. And we're always running sales with savings of at least 50% from the list price on some of your favorite programs.

And there are several ways to pay: MasterCard, Visa, or the ever popular check or money order.

Annotated catalog, updated every 45 days

We don't like to knock the competition, even by implication, but there are definite advantages to a CompuClub membership. One of the best of them is our catalog, which not only offers an unbelievable number of programs, but includes a description of each and every one of them.

In fact, the catalog is so big and filled with so much information that we print a separate price list. And we keep adding so many programs that we have to update the catalog nine times a year during the course of your membership.

Think of it! No more wondering what's behind the fancy label, no more shooting in the dark or depending on the word of a clerk who may not be familiar with the program you're interested in.

Exclusivity for Atari® owners

We're not trying to be all things to all people. We are definitely the best thing since the floppy disk for Atari® owners, and only Atari® owners. CompuClub is very exclusive.

Our catalog is thick with hundreds of Atari® programs, and only Atari® programs. No more fumbling your way through thickets of strange symbols and codes for the different computers everybody else's catalogs try to cover.

The rules

The rules are simple. To play the CompuClub "game," you've got to be a member. Anyone can join, as long as they fork over a mere five bucks, and agree to buy three programs during the year of membership. Order and buy your programs at any time during the year, but we're sure with our selection and prices that you'll want to get going right away.

Five dollars buys you a one-year membership, exclusive Atari® program offerings, a fully annotated catalog with 9 updates during the year, a current price list (and any necessary updates), a subscription to our newsletter, and discounted prices at all times, including announcements of our periodic sales, which feature savings of 50% off and more.

You can join by filling out the coupon below, or get an instant membership by calling our toll-free number and giving us your MasterCard or Visa identifying number. We'll forward your registration and sales order materials, your catalog and price list, so you can begin ordering right away.

That's all there is to it.

If you think you play a good game, you've got to join CompuClub. It's easy to play, and it saves you money.

We think it's the best Atari® game going, because with us, everybody wins.

CompuClub
Where Atari owners belong.

YES, **CompuClub,** PO Box 652, Natick MA 01760

I want to play the greatest Atari® game of all. Please enroll me in CompuClub for one year, and rush my registration material, catalog and price list to me. I agree to buy any three programs during the course of my year of membership. I have enclosed my payment of $5, or authorization to bill my charge card account.

Please make your check payable to CompuClub
Payment enclosed ☐ check ☐ money order.
Bill my ☐ MasterCard ☐ Visa ID #_____

Expires_____

Signature_____

Name_____

Address_____

City_____ State_____ Zip_____

Computer Model_____ Disk ☐ Tape ☐

If you are not satisfied in any way with the conditions of your membership, you may cancel before ordering any programs from CompuClub. Your membership fee of $5 will be cheerfully refunded.

To join by phone, call toll-free.

800-631-3111

Please have your MasterCard or Visa card ready when you call. In Massachusetts, call 1-617-879-5232. We will forward your membership materials after confirming authorization for your charge.

THE HOTTEST NEW SOFTWARE IN THE GALAXY!!!

BRINGING COMPUTERS DOWN TO EARTH.

PROGRAMMIN' STUFF

is a utility program for the basic programmer! The package contains: the *PMI RENUMBER* utility, the *PMI BASIC CRUNCHER* — program compacter, *BASICDOS* — gives you the most commonly used *DOS* functions from *BASIC*, *PMI AUTORUN* system — allows you to create your own *AUTORUN.SYS* file using up to a 50 character *BASIC* statement, the *PMI RPM TEST* — disk drive speed utility, the *PMI PROGRAM SELECTOR* — allows you to run any *BASIC* program by selecting a number off the menu, and the *PMI PROGRAM ANALYZER* — will list all variables in your program, how much *RAM* it occupies and line numbers that the variables first appear in. Requires *32K RAM*, *BASIC* cartridge and at least one disk drive.

$26.95

NEW

WORDPOWER

WORDPOWER, by Barbara Wolterbeek, is a powerful vocabulary building program. WORDPOWER provides an easy, systematic method for improving your English vocabulary skills. With 32 sets of 25 words each broken down into categories of 'Root Words,' 'Synonyms' and 'Antonyms,' WORDPOWER makes expanding your vocabulary fun! Designed for high school students in preparation for college entrance examinations. But you may find this program a challenge no matter what your educational background. Requires *32K RAM* and one disk drive.

$27.95

BIOGRAPH

Want to know if this is your day or not? Or when your day will be? Your biorhythms can tell you and *BIOGRAPH* can show you. *BIOGRAPH* will give you a colorful graphic chart of your biorhythms for any month you choose in any year. *BIOGRAPH* will also point out critical days (ones you should stay in bed on). A simple to use, educational and enjoyable program. Requires *32K RAM*, *BASIC* cartridge and one disk drive.

$24.95

PROGRAMMERS!

If you have written a program that is technically sophisticated and also "user-friendly", PMI would be interested in publishing your work. For information on how to submit your material write to: PMI, P.O. Box 2895, Winter Park, FL 32790-2895, Or phone (305) 644-3822.

VARICOMMANDER

is a state of the art variable manipulator with extremely powerful programming abilities. You can manipulate variable names virtually any way you wish. Options include automatic generation of one or two character names, or all character one name (Atascii character). You can even rename all variables in your program automatically to two character names and retain string and dimension integrity. Designed for the novice or the advanced programmer. *VARICOMMANDER* is ergonomically designed with extensive error protection. Requires *32K RAM*, *BASIC* cartridge and one or more disk drives.

$29.95

RESTORE

List the unlistable! *RESTORE* by William Fletcher will allow you to make a *LISTABLE, EXECUTABLE* copy of any *BASIC* program. *RESTORE* is extremely user-friendly and the simplicity of operation is astounding. *RESTORE* is a compiled program and executes at machine language speed. Utilizing the graphics and sound capability of the Atari, *RESTORE* is not only an indispensible programmer's tool, but is fun to use. If you have two disk drives *RESTORE* will automatically seek out relocated directories and give you the option to either restore all *BASIC* programs on the disk or make a copy of the disk putting the directory in its normal location. Requires *48K RAM* and one or more disk drive.

$27.95

NEW

T.I.P.S.

THOROUGHBRED INFORMATION PROCESSING SYSTEM, by Samuel Corkin and Richard Lamb, is the most comprehensive, accurate and user-friendly performance projection program ever written for use on a micro computer. NO program available will give you more reliable results on thoroughbred horse racing than T.I.P.S. Everything the handicapper needs to understand and implement the program is included in a most readable 46 page instructions manual. T.I.P.S. is without a doubt the most thorough program of its type. If you want a winner T.I.P.S. is your best bet! Requires *48K RAM* and one disk drive.

$45.95

COM-CON

Have you ever wondered what your computer would have to say about that party you had last weekend? Or the shape the world is in today? Well now you can find out. *COM-CON* has arrived! *COM-CON* by William Austin is an unique, exciting and entertaining program for everyone. Your computer can answer your questions on virtually any topic. While carrying on a discussion with *COM-CON* you will notice that your computer actually has a "personality" of its own. And like any human being the answers will sometimes be as vague and nebulous as a political candidate. And at other times quite philosophical. If you're 4 to 104 *COM-CON* is for you. *COM-CON* is available for the Votrax Type 'n Talk, $31.95, the Alien Group Voice Box $31.95, *SAM* $31.95, and in a text only version $26.95. Requires *48K RAM*, *BASIC* cartridge, one disk drive and optional voice synthesizer.

SPECIAL OFFER

VARICOMMANDER *and* PROGRAMMIN' STUFF - only $45.95.

All *PMI* programs are available at the best software stores throughout the world, or order direct from *PMI:*

PMI, P.O. Box 2895, 2500 Lee Road, Suite 210, Winter Park, FL 32790-2895

☐ *PROGRAMMIN' STUFF* ☐ *T.I.P.S.* ☐ *BIOGRAPH* ☐ *VARICOMMANDER* ☐ *RESTORE*
☐ *COM-CON* ☐ *VARICOMMANDER and PROGRAMMIN' STUFF PACKAGE* ☐ *WORDPOWER*

Total $ _____, plus $3.00 postage & handling (Florida residents add 5% sales tax)

☐ CHECK (allow 3 weeks for personal checks to clear) ☐ MONEY ORDER
☐ C.O.D. (add an additional $2.00 for C.O.D. orders)

NAME (print) _____

Address _____

City _____ State _____ Zip _____

S.A.M. is a trademark of DON T. ASK

DEALER INQUIRIES INVITED

ATARI is a trademark of ATARI INC

We just made owning an Atari computer a lot more logical.

Introducing the Rana 1000 disk drive. It's a whole new game for Atari computers.

This two digit LED readout displays a code that tells you everything you need to know.

This beeping button tells you your write protect feature is keeping your information safe.

The remaining buttons beep when touched, and provide readouts on density storage, error status, and drive number.

This button beeps when you touch it, and the LED readout tells you what track you're on.

When Rana Systems introduced the Elite Series of Apple® compatible disk drives, we didn't know what a tremendous impact they would make. It turned out to be a line so outstanding in performance, styling, capacity, and price, that it instantaneously made us a major force in the market. Well, needless to say, the response was so great that we were forced to create the same highly advanced disk drive for Atari.® A disk drive that when coupled with Atari's computer, could perform everything from accounting, financial planning, and stock charting, to word processing, business management, and letting you write your own programs. Plus, we made it simple enough for a child to use, for learning anything from the alphabet to a foreign language.

Working with a diskette versus playing with a cassette.

Let's face it. The only reason Atari made a cassette option to their computer was to make it affordable. But now you don't have to settle for less. Because now you can get a diskette for your Atari computer which outperforms their cassette and costs ⅓ less than their disk drive. With Atari's cassette you only get half the functions of a computer compared to what our floppy disk can give you. Their cassette is not only limited in the software available, but it also takes 20 times longer to get the information you need. And Rana's disk drive offers twice the storage capacity of either their cassette or disk drive.

Why even stylewise our new low profile design not only looks 100 times more spectacular, but it occupies 3 times less space. And our new Rana 1000 also gives you a piece of its mind every time you use it, because our disk drive gives you information as well as takes it. And we think that says a lot.

The disk drive that has all the answers.

Rana offers you a myriad of features Atari couldn't even conceive of. Like five electronic functions on the front panel that actually beep and give you a LED readout when touched. Our disk drive tells you what track you're on, and what density and how much information you're storing. It lets you switch from a single density of 90,000 letters to a double density of 180,000 letters, on a single diskette. And, we have a write protect feature which protects your diskette from being erased. In fact, no other disk drive can offer you that.

As you can see, it was easy to build a disk drive superior to Atari's. Because for every reason you buy a disk drive, Rana has superior technology.

The Rana 1000 disk drive. It brings your Atari computer to a higher level of sophistication for a price one third lower than Atari's. So your choice shouldn't even be a matter of logic.

Just common sense.

RanaSystems *Always a step ahead.*

21300 Superior Street, Chatsworth, CA 91311 213-709-5484. For dealer information call toll free: 1-800-421-2207. In California only call: 1-800-262-1221. Source Number: TCT-654

® Apple is a registered trademark of Apple Computer, Inc. ® Atari is a registered trademark of Atari, Inc., a Warner Communications Company. See us at the West Coast Computer Show. Circle 44

Books&Software

BOOKS for ATARI 400/600XL/800XL 1200XL

ATARI BASIC – Learning by Using
An excellent book for the beginner. Many short programs and learning exercises. All important features of the ATARI computers are described (screen drawings, special sounds, keys, paddles, joysticks, specialized screen routines, graphics, sound applications, peeks, pokes, and special stuff).
Order-No. 164 $7.95

Games for the ATARI Computer
This book describes advanced programming techniques like player-missile-graphics and use of the hardware-registers. Contains many ready to run programs in BASIC and one called GUNFIGHT in machine language.
Order-No. 162 $7.95

How to program your ATARI in 6502 Machine Language
Introduction to machine language for the BASIC programmer.
Order-No. 169 $9.95

FORTH on the ATARI – Learning by Using
Introduction, programs, applications, learning exercises.
Order-No. 170 $7.95

All programs from book No. 170 on disk.
Order-No. 7319 $22.00 only!

A Look into the Future–ASTROLOGY on your ATARI 800. How to calculate your own horoscope. Including listing of the program.
Order-No. 171 $9.95

HACKERBOOK for your Atari computer
Tips+tricks–Very important subroutines in 6502 machine language. How to make bootable cassettes, disks, and EPROMs. Complete construction article and software on how to build an EPROM burner.
Order-No. 172 $9.95

SMALL BUSINESS SOFTWARE FOR ALL ATARI COMPUTERS

SUPERMAIL (500 addresses on 1 disk) Completely written in FORTH. Comes on autoboot disk. No cartridge, no DOS, no FORTH language required.
Order-No. 7312 $49.00

SUPERINVENTORY (1000 items p.disk) Completely written in FORTH. Same as above. (Disk only)
Order-No. 7320 $49.00

BUSIPACK-1 (written in FORTH). Complete order entry, inventory, mailing and invoicing. (Disk only).
Order-No. 7313 $98.00

Microcomputer Hardware Handbook
Order-No. 29 $ 14.95

ATCASH
Convert your ATARI 800 into a powerful cash register. (Disk only).
Order-No. 7303 $49.95

Invoicing program in BASIC
Order-No. 7201 (C) $29.95
Order-No. 7200 (D) $39.95

Mailing List in BASIC
Order-No. 7212 (C) $19.95
Order-No. 7213 (D) $24.95

Inventory control in BASIC
Order-No. 7214 (C) $19.95
Order-No. 7215 (D) $24.95

SOFTWARE IN MACHINE LANGUAGE FOR ATARI

ATMONA-1
Machine language monitor.
Order-No. 7022 (C) $19.95

ATMONA-2
This is a tracer (debugger) that lets you explore the ATARI RAM/ROM area. You can stop at previously selected address, opcode, or operand. Also very valuable in understanding the microprocessor. At each stop, all registers of the CPU may be changed. Includes Atmona-1.
Order-No. 7049 cassette $49.95
Order-No. 7050 disk $54.00

ATMAS
Macro-Assembler for ATARI-800/48K. One of the most powerful editor assemblers on the market. Versatile editor with scrolling. Up to 17k of source-code. Very fast, translates 5k source-code in about 5 seconds. Source code can be saved on disk or cassette (Incl. Atmona-1).
Order-No. 7099 disk $89.00
Order-No. 7999 cartridge $129.00

ATMS APPLICATION DISK
All programs and machine language subroutines from Book No. 169 on disk.
Order-No. 7311 $20.00

ATAS
Same as ATMAS but without macro-capability. (32K and 48K RAM)
Order-No. 7098 $49.95

ATEXT-1
This wordprocessor is an excellent buy for your money. It features screen oriented editing, scrolling, string search (even nested), left and right margin justification. Over 30 commands. Text can be saved on disk or cassette.
Order-No. 7210 cassette $29.95
Order-No. 7216 disk $34.95
Order-No. 7217 cartridge $69.00

FORTH for the ATARI
FORTH from ELCOMP PUBLISHING, Inc. is an extended Fig-Forth-version, Editor and I/O package included. Utility package includes decompiler, sector copy, Hexdump (ASCII), ATARI filehandling, total graphic and sound, joystick program and player missile. Extremely powerful!
Order-No. 7055(D) reg. $99.00 sale $39.95

Floating point package with trigonometric functions (0 – 90°).
Order-No. 7230 disk $29.95

Learn FORTH
A subset of Fig-Forth for the beginner. On disk (32k RAM) or cass. (16k RAM).
Order-No. 7053 reg. 79.00 sale $19.95

HARDWARE-ADD-ONs for ATARI

PRINTER INTERFACE
This construction article comes with printed circuit board and software. You can use the EPSON printer without the ATARI printer interface. (Works with gameport 3 and 4).
Order-No. 7211 $19.95

300 Baud serial interface (RS232+5V)
Software with connector and construction article.
Order-No. 7291 $19.95

EPROM BURNER for ATARI 400/800 KIT. Printed circuit board incl. software and extensive construction article.
Order-No. 7292 $49.00

EPROM BOARD (CARTRIDGE)
Holds two 4k EPROMs (2532). EPROMs not included.
Order-No. 7043 $29.95

EPROM BOARD KIT
Same as above but bare board only with description.
Order-No. 7224 $14.95

Astrology and Biorhythm for ATARI
Order-Nol 7223 D/C $29.95

Birth control with the Atari(Knaus Ogino)
Order-No. 7222 disk only! $29.95

The APPLE in your Hand, BRAND NEW! Book includes introduction to 6502 Machine Language and FORTH. BASIC programs never published before!
Order-No. 178 (200 pages) $12.95

CP/M –MBASIC and the OSBORNE computer
Business Applications, compl. listings of mailing list, datablock, inventory, invoicing and more.
Order-No. 177 $9.95

SUPERSOFTWARE f.the Commodore-64

BLIZTEXT! – The best wordprocessor for the C-64 in the whole universe. Includes terminal software for electronic mail.
Order-No. 4965 (62 pages manual) $89.00

MACROFIRE – Editor/Assembler for the C-64
The best macroassembler you can buy!
Order-No. 4963 $89.00

SUPERBOOKS for your C-64

The Great Book of Games, Vol. I
64 programs for the Commodore-64.
Order-No. 182 $9.95
Programs from this book on disk $19.95

MORE ON THE SIXTYFOUR
Tips, tricks, hints, very important subroutines.
Order-No. 182 $9.95
Programs from this book on disk $19.95

How to program in 6502 Machine Language on your C-64
Order-No. 184 $12.95

Commodore-64 Tune-up, Vol. I
How to expand your C-64.
Order-No. 185 $12.95

Small Business Programs for the C-64
Order-No. 186 $12.95

HARDWARE ADD-ONs for your C-64

Parallel printer interface, KIT
Order-No. 4990 $19.95

Universal Experimenter Board
Order-No. 4970 $9.95

Expansion Board (holds 4 exp. boards)
Order-No. 4992 $29.95

BOOKS, SOFTWARE, ADD-ONs for VIC-20, APPLE II, OSBORNE, TIMEX + OSI computers

Tricks for VICs (Book)
Order-No. 176 $9.95

Universal Experim.board for the VIC-20
(Save money with this great board). This board plugs right into the expansion slot of the VIC-20.
Order-No. 4844 reg. $19.95 sale $9.95

Programming in BASIC and machine language on the ZX-81 (82) TIMEX1000
Order-No. 174 (book) $9.95

The Custom APPLE + Other Mysteries
A complete guide to customizing the APPLE software and hardware.
Order-No. 680 $24.95
We also stock the boards which are used in the book No. 680 (bareboards).

HOFACKER
Elcomp Publishing, Inc, 53 Redrock Lane, Pomona CA 91766, Phone: (714) 623-8314, Telex: 29 81 91

PAYMENT: Check, money order, VISA, MASTER CARD, Eurocheck, ACCESS, Interbank
Prepaid orders add $3.50 for shipping (USA)
$5.00 handling for C.O.D.
All orders outside USA: add 15% shipping, California residents add 6.5 % sales tax.

HEXPAD

16K Cassette or Disk

by Randal C. Gibson

Hexpad is a subroutine that will allow you to easily add DATA statements to your BASIC programs. I decided to submit this article after using **Hexpad** to quickly and accurately enter the DATA statements for the program **Livewire!**, published in the 12th issue of **A.N.A.L.O.G. Computing**. In my mind, **Livewire!** is certainly the best public-domain game ever published for ATARI computers.

But there are probably some of you out there who have not yet keyed in some of Tom Hudson's other games, such as **Fill'er Up!** from **A.N.A.L.O.G. Computing** #10, or this issues' **Retrofire**, because of the large number of hexadecimal DATA statements in those programs. I hope a lot of you will use this routine to turn your ATARI computer's keyboard into a hex/numeric keypad to facilitate easy entry of BASIC DATA statements. This method of data entry should also help you to be more accurate. After keying in all 100 of the DATA statements in **Livewire!** and then running the D:CHECK2 error correction program, I discovered that I had mis-keyed only five single characters.

Using the routine.

After keying in the **Hexpad** routine, LIST it to cassette or disk with a LIST"C: or a LIST"D:HEXPAD.LST (or use any filespec you want). The line numbers of the routine start at 3100, so that it may be merged into an already existing program. This means that you do not have to key in the DATA statements of the main program first.

For example, if you are keying in a program that has program lines from 1-990, DATA statements from 1000-1990, and more program lines from 2000-on, then you can first key in the program lines from 1-990. You would then merge in the HEXPAD subroutine with an ENTER"C: or an ENTER"D:HEXPAD.LST. To start entering the DATA statements you would then type in "GOTO 31000". The subroutine will then prompt you to enter the starting line number and the increment of the DATA statements. If the line numbers of the DATA statements are 1000, 1010, 1020, 1030, and so on, then you would enter "1000,10" and press RETURN.

At this point the text "1000 DATA" will be displayed on the screen and the cursor will be flashing one space after the DATA keyword, waiting for you to enter the data. Also, the keyboard will have been forced into lowercase mode. In this mode the hex/numeric keypad is active; if you manually switch into uppercase or use the SHIFT keys then the keypad is disabled and all of the letters may be entered on the screen.

As you enter your data, the subroutine checks each key that you press and then outputs a character to the screen. Pressing the space bar, the m key (lowercase M), or of course the 0 key will cause a 0 to be printed on the screen. Likewise, the j,k, and l keys are converted to 1, 2, and 3, the uio keys to 456, and the bn-gh-ty keys are converted to the uppercase AB-CD-EF characters. Pressing any other keys will output the normal characters or control functions.

Hexpad contains a machine language subroutine that does all of the key-code conversions, so that if you hold down any key (such as a cursor control key) it will auto-repeat just as quickly as normal.

I used the lowercase mode in the conversions so that all uppercase alphabetic characters may be entered. But the space bar will produce a 0 character in either upper or lowercase modes, so you will have to generate a space character by using the CTRL-cursor right function if you are at the end of the line. You can also use the CTRL-INSERT function to insert spaces into the middle of the line.

ATARI SINGS YOUR FAVORITE SONGS!!!

THE Original VOICE BOX Speech Synthesizer by the ALIEN GROUP has received rave reviews.

MICRO COMPUTING–"The VOICE BOX injects an endearing personality to your computer. The possibilities are enormous."

COMPUTE–"The VOICE BOX offers more human-like tones and does not blank out the screen."

CREATIVE COMPUTING–"English text and phonetic code may be freely intermixed rather than requiring separate modes as is the case without exception with every other speech system. A mode called talking face displays an animated face with impressive lip sync animation."

ANTIC–"There is a great potential for teaching children to spell and an added dimension to games overall. I believe the VOICE BOX is well worth the price tag."

ANALOG–"For ATARI owners who want to add speech to their programs the Alien Group VOICE BOX is probably the best choice."

POPULAR SCIENCE–"The speech quality is excellent. Besides creating speech the software has a bit of fun with graphics."

and on the new VOICE BOX II

TIME MAGAZINE–"Machine of the Year" "The VOICE BOX by the Alien Group enables an ATARI to say aloud anything typed on its keyboard in any language. It also sings "Amazing Grace" and "When I'm 64" or anything else that anyone wants to teach it.

INCORPORATE THE SINGING HUMAN FACE INTO YOUR PROGRAMS AND GAMES

VOICE BOX II
Speech & Singing Synthesizer

The New **VOICE BOX II** for **ATARI** plugs into the serial port of the **ATARI** 400/800 with sound coming out of the TV/monitor. 48K DISK is required. It has all of the features of the original **VOICE BOX** plus many exciting new hardware and software features:

- The ability to sing with voice and 3 part music.
- A library of 30 famous songs.
- A comprehensive music system that allows the user to easily enter or modify new songs.
- Software that can convert the bottom two rows of the **ATARI** keyboard into a piano with a range of 3½ octaves using the shift and control keys.
- Programmable musical sound effects such as tremolo, vibrato, glissando and click track.
- A singing *human* face with lip-sync animation designed by Jerry White.
- A talking or singing *ALIEN* face with software that allows the user to change the face and 8 mouth patterns as he sees fit.
- The ability to speak with inflection and feeling.
- Can speak in a foreign language with correct foreign spelling as input.
- A talk and spell program by Ron Kramer. Users can program any vocabulary for this spelling game. In fact, this program can even speak in a foreign language like French, where the user must spell the correct word in English, or vice versa.
- **GREEN GOBLINS**–A talking arcade game by John Wilson.
- Random Sentence Generator–An amusing grammar game that helps teach school children to identify parts of speech and recognize a variety of sentence structures.
- **NUMBER SPEAK**–A subroutine by Scott Matthews that converts up to a 9 digit number into normal English pronunciation. Ideal for building your own math games.
- **STUD POKER**–A talking poker game by Jerry White.
- The screen never blanks out while talking or singing.
- Singing or speaking subroutines can be incorporated into your programs, requiring as little as 100 bytes of RAM plus 5 bytes for each word.
- Entries into the $5000 talking or singing game contest can be written using the **VOICE BOX II**–send for contest information.
- Price $169.00 includes **VOICE BOX II** and all of the above software.
- Inquire about our discounts for educational institutions.

To order by mail send a check or money order to the ALIEN GROUP for $169. Then, try the VOICE BOX II for 10 days, and if it isn't the finest value you've ever seen in a computer peripheral, the most challenging and provocative addition you've ever made to your system, return it in its original condition for a full refund.

THE ALIEN GROUP
27 West 23rd Street (212) 741-1770
New York, NY 10010

ALSO AVAILABLE AT LEADING COMPUTER STORES THROUGHOUT THE WORLD.

When you have finished keying in the data, press the RETURN key. The line will then be entered into the main program, the screen will be cleared, and the next DATA line number will be displayed. Each new line number is calculated by adding the increment you specified to the previous line number. If the DATA statements stop following the previous line number sequence or increment, and you find that the line number displayed is incorrect, just press the TAB key and you will be prompted for a new starting line number and increment.

When all of your DATA statements are entered, or if you want to stop to enter some more BASIC statements, then you need to exit the data entry subroutine by pressing the BREAK key. You can do this when a DATA line is displayed but you might still be in lowercase mode. However, when you press the TAB key to return to the starting line #, increment prompt, the keyboard is put into the uppercase mode. So if you just entered the last data line (1990 in our example) and the prompt "2000 DATA" is on the screen, you can quickly press TAB and then BREAK to exit the subroutine in uppercase mode. At this point you may continue entering BASIC lines or you can enter more DATA statements by typing in "GOTO 31000", RETURN.

When you have finished entering all of your DATA statements, you may want to "get rid of" the **Hexpad** subroutine. You can do this by simply typing in "GOTO 32000", RETURN. This will cause every line of the **Hexpad** subroutine to be deleted, leaving only your main program lines and DATA statements.

How Hexpad works.

LINE 31000: The first line sets the margins turns on the flashing cursor, and clears the screen. The RESTORE function is used to make sure that the data that is READ in the next line is the data at line 31012 and not any other data that might be in the main program.

31001: This line stores the machine language subroutine into Page 6 of memory. The variables in this subroutine begin with QX, so they hopefully will not conflict with the variables in the main program.

31004: The line first closes IOCB #1 just in case it was previously opened either by this subroutine or the main program. Poking a 0 into location 702 forces the keyboard into lowercase mode. Lastly, the keyboard is opened for input, one character at a time.

31005: This line clears the screen, prints two blank lines and prints the current line number with the keyword DATA. The two blank lines are required for the "Automatic Screen Statement Processing" to work correctly. The ASSP is the method by which the DATA lines that you key in are entered into the main program. This will be discussed further at line 31009.

31006: The GET statement in this line will place the ATASCII value of the next key that you press into the variable QXD. If you press the RETURN key (ATASCII value 155) the subroutine will GOTO line 31009.

31007: If you press the TAB key then the keyboard will be forced into uppercase mode and you will be prompted to enter another starting line number and increment.

31008: The USR on this line will run the machine language subroutine previously poked into Page 6. The ML subroutine will return into the variable QXR the converted keycode as discussed at line 31013.

31009: This line first prints a CONTinue keyword to the screen for later processing. The cursor is positioned at the top of the screen and the ASSP is turned on by poking 13 into location 842. When the STOP statement is executed, the computer begins reading statements off the screen as if you had keyed them in yourself. Since we moved the cursor to the top of the screen, the computer will first "input" the DATA statment thay you just keyed in. That line will be added to the main program. Next, the computer will process the CONT statement and jump back into **Hexpad** where it left off by executing line 31010.

31010: This line first pokes 12 into location 842 to turn off the ASSP. If you BREAK out of the **Hexpad** subroutine, you will then be in the normal processing mode. Then the increment is added to the last line number. If the new line number is less than 30000, you will be prompted to enter another DATA line.

31012: The first line of data is the machine language code that performs the keycode conversion.

31013: This second line of data is the ATASCII values of the 14 keys to be changed and the ATASCII codes of the new keys that should be printed to the screen. Any keycodes that are not among the first 14 numbers in this list will remain the same. The first number in this list (32) corresponds to the space bar. If your DATA statements require spaces in them, you can change this code to 48 so that the 0 key will just be converted to itself. By changing these numbers you can have any 14 keys converted to any other 14 characters on the screen.

32000: These last two lines are really not part of the subroutine. They are only here to allow easy deletion of the entire **Hexpad** utility. The first line will print all of the line numbers of the utility to the screen.

32001: This last line will print three direct-mode statements to the screen for later processing. The cursor is positioned at the top of the screen and the ASSP is turned on. All of the lines of **Hexpad** are deleted from the main program and the three direct-mode statements will be executed. Poking 12 into location 842 turns off the ASSP. Finally, the screen is cleared and you will see the READY prompt. □

Listing 1.

```
31000 CLR :POKE 82,2:POKE 83,39:POKE 7
52,0:? CHR$(125):RESTORE 31012
31001 FOR QXI=1536 TO 1589:READ QX5:PO
KE QXI,QX5:NEXT QXI
31002 ? CHR$(125):? :? "ENTER STARTING
 LINE #, INCREMENT :":INPUT QX5,QXI
31003 QX5=INT(QX5):QXI=INT(QXI):IF QX5
<1 OR QX5>29999 OR QXI<1 OR QXI>9999 T
HEN 31002
31004 CLOSE #1:POKE 702,0:OPEN #1,4,0,
"K"
31005 ? CHR$(125):? :? :? QX5;" DATA "
;
31006 GET #1,QXD:IF QXD=155 THEN 31009
31007 IF QXD=127 THEN POKE 702,64:GOTO
 31002
31008 QXR=USR(1536,QXD):? CHR$(QXR);:G
OTO 31006
31009 ? :? "CONT":POSITION 2,0:POKE 84
2,13:STOP
31010 POKE 842,12:QX5=QX5+QXI:IF QX5<3
0000 THEN 31005
31011 CLOSE #1:STOP
31012 DATA 162,13,104,104,104,221,26,6
,240,10,202,16,248,133,212,169,0,133,2
13,96,189,40,6,76,13,6
31013 DATA 32,109,106,107,108,117,105,
111,98,110,103,104,116,121,48,48,49,50
,51,52,53,54,65,66,67,68,69,70
32000 CLOSE #1:? CHR$(125):? :? :FOR Q
XD=31000 TO 31013:? QXD:NEXT QXD:? 320
00:? 32001
32001 ? "CLR:POKE842,12:?CHR$(125)":PO
SITION 2,0:POKE 842,13:STOP
```

•

CHECKSUM DATA

(See p. 58)

```
31000 DATA 131,772,656,463,722,424,619
,431,970,181,733,603,916,907,884,9412
32001 DATA 253,253
```

•

Assembly Language Listing

```
0100          *=  $600
0110          LDX #13
0120          PLA
0130          PLA
0140          PLA
0150 CMPKY    CMP OLDKC,X
0160          BEQ GOTKY
0170          DEX
0180          BPL CMPKY
0190 NEWKY    STA $D4
0200          LDA #0
0210          STA $D5
0220          RTS
0230 GOTKY    LDA NEWKC,X
0240          JMP NEWKY
0250 OLDKC    .BYTE " mjkluiobnghty"
0260 NEWKC    .BYTE "00123456ABCDEF"
0270          .END
```

•

GRIFFIN'S LAIR

A NEW EDUCATIONAL COLUMN

by Braden E. Griffin, M.D.

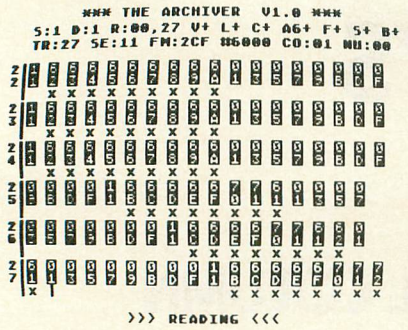
Recently, the public has become inundated with an overwhelming number of computer games, spinoffs from arcade successes and spinoffs of spinoffs. The plethora of great games and realization that another great game is just around the corner, combined with the continued high cost of software has made the public much more discriminating. Always searching for new markets, many software manufacturers have turned to educational software to fill this void. These companies are in the business of product sales, not necessarily to provide education. In this context, many of these programs fall far short of the claim to be educational. I am not so naive as to think that the purest of motives, i.e., to educate our young, is the driving force behind this movement. There are always bad products and great products, but most products are between these extremes, suited to different tastes and different budgets. Educational software is no exception. With this in mind, the editors of **A.N.A.L.O.G.** have set aside this space for the discussion and review of educational materials for the ATARI.

What are my qualifications for writing such a column? I am a Pediatrician, a teacher, a student, a computer enthusiast, and a parent. Not unlike most of you who are interested in this subject. (My children have always thought the M.D. stood for "my dad.") I am no more an expert on education than you. I do not have large numbers of children to test the efficacy of these programs in a scientific manner — nor do most reviewers. I *do* have access to newly released software. If everyone had similar access without having to first purchase an item, we probably could eliminate reviews of all software, but alas, we must rely on others for this. I hope to describe educational software as simply as possible; commenting on its contents, presentation, documentation, accuracy, and consistency, and not so much on its inherent educational value. Children have quite disparate needs. What is educational for one child may be of little benefit to another. Parents are still the best judge of what their children need.

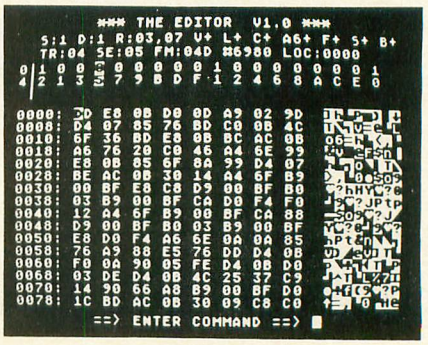
The use of the computer in education has created some controversy. The fear that the computer *will* replace teachers is no more likely than teachers being replaced by the television, a common fear during the 1950's. Computer-aided instruction (C.A.I.) is simply another educational tool, not the only tool. If it makes learning easier, more enjoyable, fine, but it must be kept in perspective. One may learn as much from an adventure game as from a math tutorial. The development of logical thinking acquired from BASIC programming or the reinforcement of grammatical skills when using a word processor may be of greater educational benefit than software specifically designed for education. A simple program seeking the answer to a mathematical operation can easily be designed for a child's individual needs and be personalized at the same time. I hope to explore these areas as well as review the new releases. □

(Dr. Griffin's reviews of the newest educational software releases start on the next page)

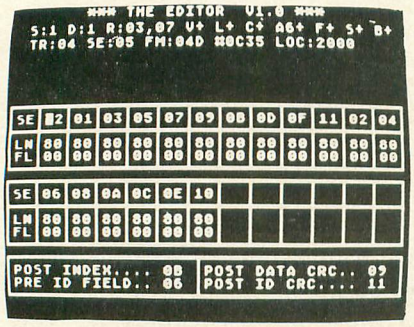
810 POWER!



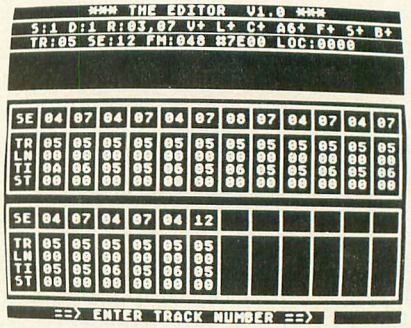
ARCHIVER



EDITOR



FORMATTER



MAPPER

The **CHIP**™ with Archiver Software allows:
Automatic Back-ups of **any** Disk Program. Easy enough for a beginner to use.
Also includes Sector Editor, Disassembler, Mapper, and Custom Formatter.

All this for **only $99.95** plus shipping

Happy owners? . . . . . .
Update your Happy 810 Enhancement now with **ARCHIVER/EDITOR** Software.
Supports multiple drives and makes your Happy drive compatible with the **CHIP!**

Just **$39.95** plus shipping

Visa & MasterCard or C.O.D. call now (815) 229-2999 for immediate shipment
or send check or M.O. to:

MIGHTY BYTE COMPUTER
828 Green Meadow Ave., Dept. AG
Rockford, Illinois 61107

For Technical Support
and Drive Service
Call 7pm-11pm CST

Thank You for making our 1983 an Outstanding Success!

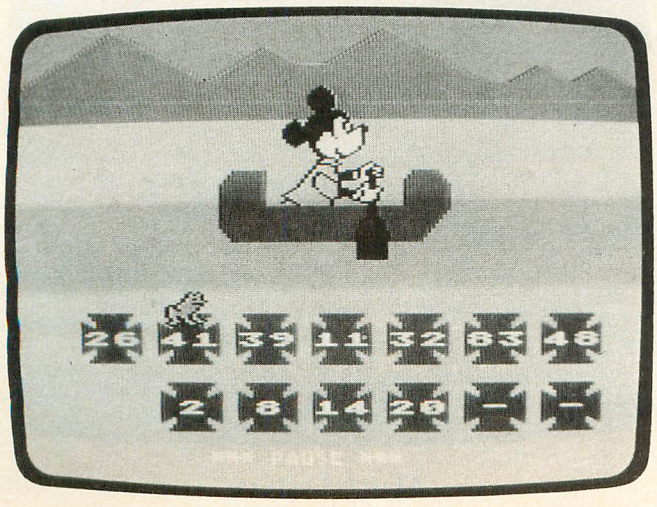
The CHIP and ARCHIVER Software are products of Spartan Software of MN, Inc.

MICKEY IN THE GREAT OUTDOORS
***Walt Disney* Personal Computer Software**
Distributed through ATARI, Inc.
32K Disk/16K Cassette $29.95

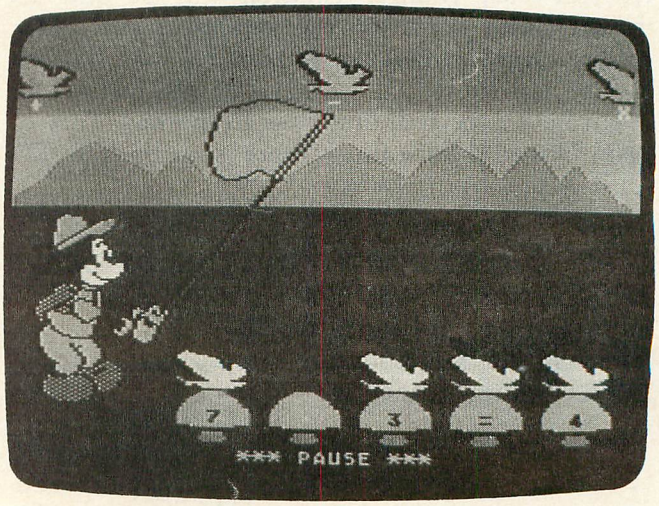
Those of you who associate the name Disney with quality and class will not be disappointed with **Mickey In The Great Outdoors**. The theme parks of the same name have always been characterized as wholesome and super clean. This educational software does not differ in those respects. Fancy frills and gimmicks are not to be found here. The graphics and animation are described as crisp and that is just what they are. Pleasant music and interesting sound effects further enhance this interactive learning game.

A learning adventure.

The adventures referred to on the packaging should not be confused with the usual computer adventure. The program consists of two parts, each of which is made up of two activities. **Mickey Goes Hiking** is designed to improve and develop grammar and spelling skills. The first activity here involves the completion of a five word sentence. The child must select the missing word, either a noun, a verb, an adjective, or an adverb to form a complete grammatical sentence. Sounds important but not too exciting, right? Wrong! The missing part of speech is contained in a cloud floating overhead interspersed with three other clouds containing incorrect words. Mickey must aim his bow and shoot his arrow into the appropriate cloud. This takes a little getting used to for old folks, but is a cinch for the kids. If the right cloud is shot then the sentence bridge is completed and Mickey crosses to the other side. Although hundreds of sentences are available, one can only complete sixty-four bridges each game. To make it more challenging, after the first two bridges have been crossed, a turtle appears and swims toward the empty stepping stone (the missing word). If the turtle touches the stone before the sentence is completed, the computer will fill in the missing word and Mickey crosses the bridge. This adds some excitement to the playing and minimizes possible frustration if the correct word cannot be shot. After fifteen bridges, the clouds and the turtle speed up and makes it even more difficult. A score is kept for each correctly completed sentence. If the computer completes three sentences for the child then it moves on to the next activity.



The second activity requires the child to unscramble a random pattern of four letters into a word. Mickey is in a garden, holding a magic wand. By using the joystick and fire button of the screen and the child is scored relative to how much time has elapsed before the word is unscrambled. After the computer completes three of the words, game-play ends and a final score is displayed for the combination of the two activities. This type of game is fun for all ages and hundreds of word combinations are available.



Mickey goes exploring.

The second adventure concentrates on two math skills: equation solving and number sequencing. In the first of these, an equation is contained in toadstools at the bottom of the screen with either a number or an operator missing. For example, 5 ? 2 = 10 is missing the multiplication sign. Four butterflies move across the top of the screen each containing numbers of operators. The correct butterfly must be caught using Mickey's net and placed over the toadstool. Ten equations can be completed in this phase during one game. Again if the computer completes three equations the game is ended. There is no time limit in this one, so the computer only responds with the answer after the wrong butterfly is chosen twice.

In the last activity, a sequence of numbers is found in a series of lilly pads and the final two numbers in the sequence must be supplied. Here Mickey is in a

canoe above a row of seven lily pads containing numbers. A frog begins to hop across the water on these pads and as it lands on the pad with the correct number Mickey must tap it with his paddle. This is a fun game, and after the first two patterns have been completed, the top row of lily pads float left and right across the screen making it more difficult for Mickey to tap it. Correctly completing the first ten patterns allows game-play to continue indefinitely. As with the others, three completions by the computer ends the game.

Learning can be fun.

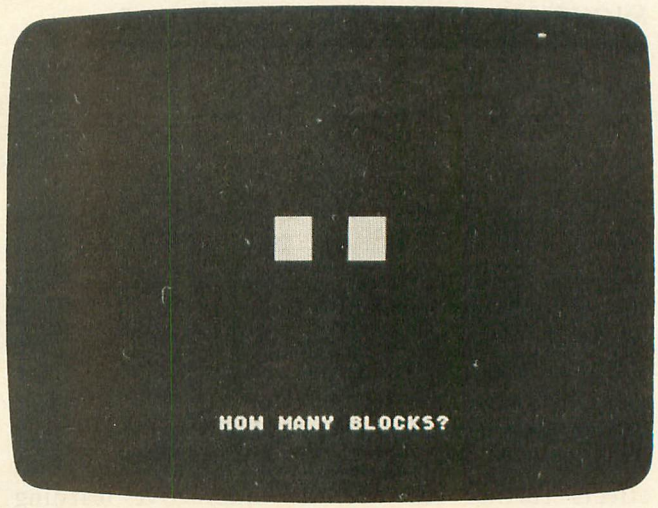
The documentation is excellent and the instruction booklet even contains some suggested activities for the parents to consider which utilize many of the skills stressed in the program. Easily loaded and protected from hitting the wrong key, children will not be frustrated with silly errors. I could find no grammatical or mathematical errors, although use of the colloquialism "snuck" instead of sneaked bothered me a bit. Although intended for seven-to-ten-year-olds, younger children will be able to interact with some of the activities, and older children will enjoy and learn from it as well. Mickey Mouse has entered the world of ATARI just as one might have expected, first class all the way. Thanks again Disney, I hope much more is on the way. □

EARLY GAMES FOR YOUNG CHILDREN

Early Games
Suite 140 Shelard Plaza North
Minneapolis, MN 55426
24K Disk/ 16K Cassette

A set of nine educational games for children ages 2 ½ to 6 years, **Early Games For Young Children** provides a good introduction to the computer for the pre-schooler. Although the packaging states that "no adult assistance is needed" and that very young children can select and play the games "all by themselves," most children will require some help initially. The format is straightforward and frustration is at a minimum. The program is automatically loaded with the BASIC cartridge in place (Run "C" with the cassette version). A menu with symbols representing each activity is paged through and the child simply presses any key to select the desired game.

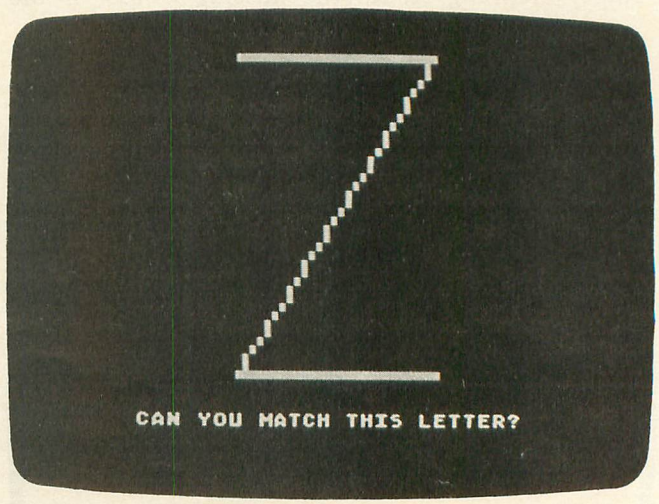
The first game is **Match Numbers**. A full screen number from one to nine is displayed and the child must select the matching numeric key on the keyboard. If there is a match, a trill is elicited and another number is displayed. Pressing any key but the correct one has no effect. **Count** displays from one to nine colorful blocks and requires that the child find and press the number corresponding to the number of blocks. **Add** and **Subtract** are the



Count

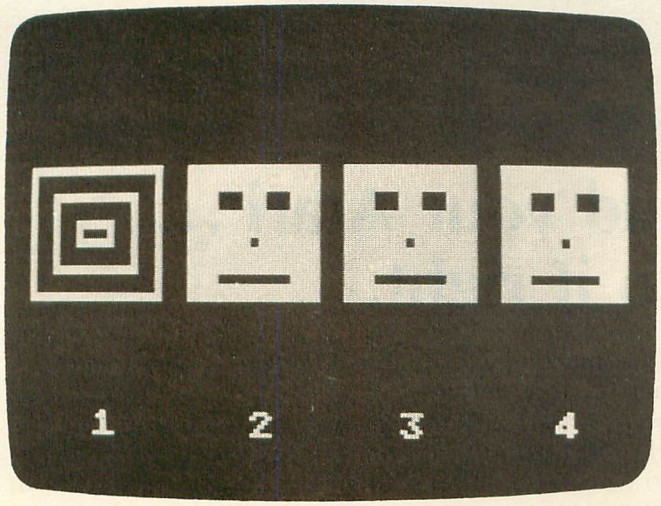
next two games. With these activities, two stacks of blocks are displayed with either a plus or minus sign between them and followed by an equals sign. The child must count the blocks and perform the function. The total of the numbers is never greater than nine and the games become a little more difficult as you progress. Parental supervision is needed for the subtraction game since this is a little tough for very young children.

The next three games deal with the letters of the alphabet. **Match Letters** is similar to the first game, with large letters being displayed instead of numbers. **Alphabet** is designed to teach the order of the alphabet. It begins with the letter "A" and the child is to press the key corresponding to the next letter of the alphabet. If correct, the letter is displayed and the computer asks for the next one. If incorrect, the computer displays the alphabet up to the letter in question, while with the next incorrect response the correct letter is displayed. This is a very pleasant activity for the child, and is not frustrating since it



Match Letters

does not require a correct response for the game to continue. **Names** requires adult attention as the child's name is entered when prompted by the computer. The child then practices by typing in his or her name. After the entire name is entered by the child, the computer displays the name in large letters. This game can be somewhat difficult for the younger children since it requires each letter in correct order for it to continue. If a child does not press the correct key, the computer just sits there.



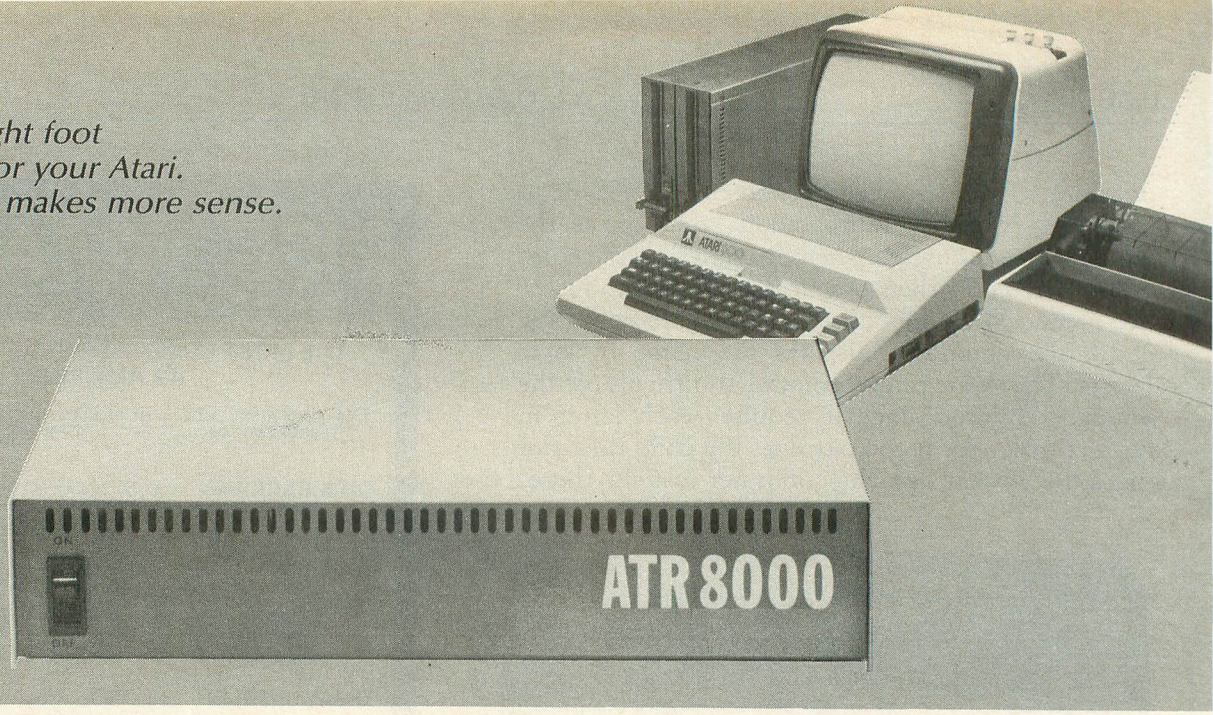
Compare Shapes

Compare Shapes is similar to the "which of these things doesn't belong" scenario made popular by a children's educational TV show. Four shapes are displayed and the child is to select the one that is different. This is well done, with a variety of colorful and distinctively designed shapes.

Draw is the last activity and the most difficult. Pictures are drawn by pressing keys on the keyboard. Upper keys draw up, side keys horizontally, lower keys down, and corner keys diagonally. The color may be changed by pressing the function keys and space bar. It is difficult to always be sure which keys go in which direction, and this game would be much improved if the joystick were utilized. (Since the program is written in BASIC it might not be too difficult for an individual to adapt the program to accept input from the joystick.) The pictures can be saved by pressing CONTROL-P and entering a name. Here again, adult supervision is necessary.

In general, this is an appealing, non-frustrating, and enjoyable learning package. It would appear to help introduce young children to the world of the computer with a minimum of adult input. Recent evidence has shown that pre-school education is probably of no great long-term benefit to an individual child, but with this program at least the child will have some fun and maybe learn a little at the same time. □

Start off on the right foot
with an ATR8000 for your Atari.
In the long run, it makes more sense.



You're going to expand your Atari®... So why not do it right.

Your first step, purchase of an Atari 400 or 800, was smart. Your next logical step is to buy a SWP ATR8000™ interface unit.

An ATR8000 gives you more capability for less money. You can start small, but you can build big . . . from an outstanding home computer to a powerful CP/M™ machine. And prices for the ATR8000 start at only $349.95.

Yes . . . I'd like to know more about SWP's ATR8000 and CO-POWER-88 for my Atari 400/800 computer. Please rush me free literature.

Send to:
SWP Microcomputer Products, Inc.
2500 E. Randol Mill Rd. - 125
Arlington, Texas 76011
200

name

street address

city state zip

MAIL TODAY!

DEALER INQUIRIES INVITED.

SWP inc
MICROCOMPUTER PRODUCTS

2500 E. Randol Mill Rd
Arlington, Texas 76011
817/469-1181

We used to be Software Publishers, Inc.

What you get first, in the amazingly adaptable ATR8000, is plug-together interfacing to all ordinary system devices:

- To floppy disk drives — The ATR8000 lets you hook up from one to four low-cost add-on drives, 5" or 8", one- or two-sided, in any mix . . . and go right to work with either a single- or double-density DOS. (DOS is optional.)
- To a printer — The ATR8000 parallel "Centronics-interface" is supported by a ROM driver program that you can access from Atari Basic.
- To a modem, to a serial data terminal or to any other RS-232 communicating device — The comprehensive users manual includes an Atari Basic serial printer driver.
- To Atari peripherals such as the 810 Disk Drive.

And of course the ATR8000 interfaces to the 400 or 800 computer itself.

But system versatility is just the beginning; the ATR8000 also opens the door to another world of microcomputing.

Atari®-CP/M™ — A whole new world of personal computing

Quite simply, CP/M is the dominant operating system of the 8-bit microworld. Make the ATR8000 your first system expansion unit, and you can put the CP/M world in your own little jug.

Since the ATR8000 is actually a self-contained Z-80 computer, a low-cost upgrade converts your Atari-ATR8000 system into a full-capability CP/M machine. Of course you can also start right off with a CP/M-ATR8000. In either case, you get CP/M program-development capability, plus access to an enormous base of proven application programs that just load and go. The CP/M upgrade, which includes 64 Kbytes of RAM, costs just $249.95.

And it doesn't end there!

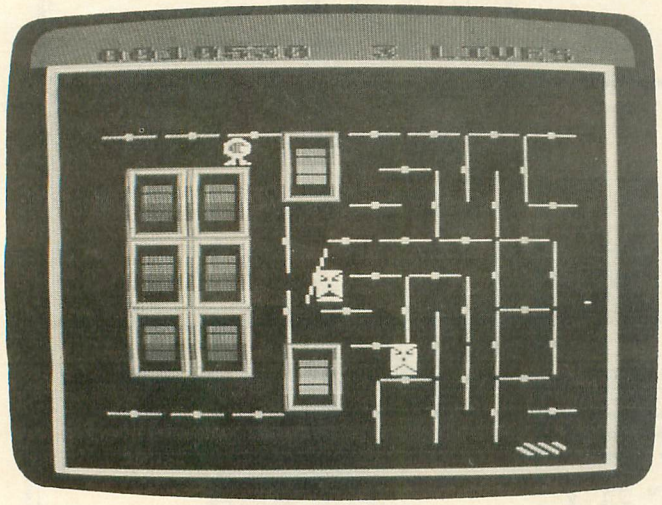
You can further upgrade your CP/M-ATR8000 to 16-bit microprocessing power . . . to CP/M-86 or MSDOS operation. Moreover, with SWP's CO-POWER-88™ upgrade, you not only retain your 8-bit processing capability, but you also can use CO-POWER-88 RAM — up to 256 Kbytes — as fast, simulated disk storage for CP/M operation.

Like the ATR8000, CO-POWER-88 comes with comprehensive documentation. Prices begin at $749.95 with 128 Kbytes of RAM and CP/M-86 operating system.

DRELBS
Synapse Software
5221 Central Ave.
Richmond, CA 94804
32K Cassette or Disk $34.95

by Lee Pappas

Here we have a game as strange as its name. The Synapse Dictionary defines a Drelb as "a small hapless creature with one huge eye" (indeed its whole head is one big eye). In times long past, Drelbs were enslaved by the evil Trollaboars, and as the only remaining "free" Drelb, you must find and rescue your captured cohorts.



Drelbs

When the game starts you find yourself on the "atomic grid." Here you must hop about, flipping the gates in the grid 90 degrees at a time in order to form enclosed boxes. Normally, this might be an easy chore; however, there are the Trollaboars to contend with. The Trollaboars patrol the grid, flipping gates as they move. Once you have formed a square, it is safe, except from the Grogolytes. These creatures can appear in a completed square and reset it so that you must again seal it off.

The only part of the grid free from the Trollaboar threat is the perimeter. However that would make your task too easy, so "Screwhead Tanks" circle the border firing random projectiles that can wander into the grid, bouncing off of the gates. Should you ever come into contact with any of these unlikeables, your poor Drelb is a deader.

Occasionally the image of a sad, but pretty (as far as computer graphics are concerned) girl will appear in a Gargolyte square. If your Drelb is fast enough to steal a kiss from this poor lady, you'll briefly go to the bonus screen and receive extra points.

With all of these menaces, what can the little Drelb do in defense? A Drelbs' best protection is to *run*! Because of the Drelbs' superior speed, it can attempt (and occasionally succeed) in trapping a Trollaboar in a square. At this point the frowning Trollaboar will fill up the square, until it frees itself and resets the square.

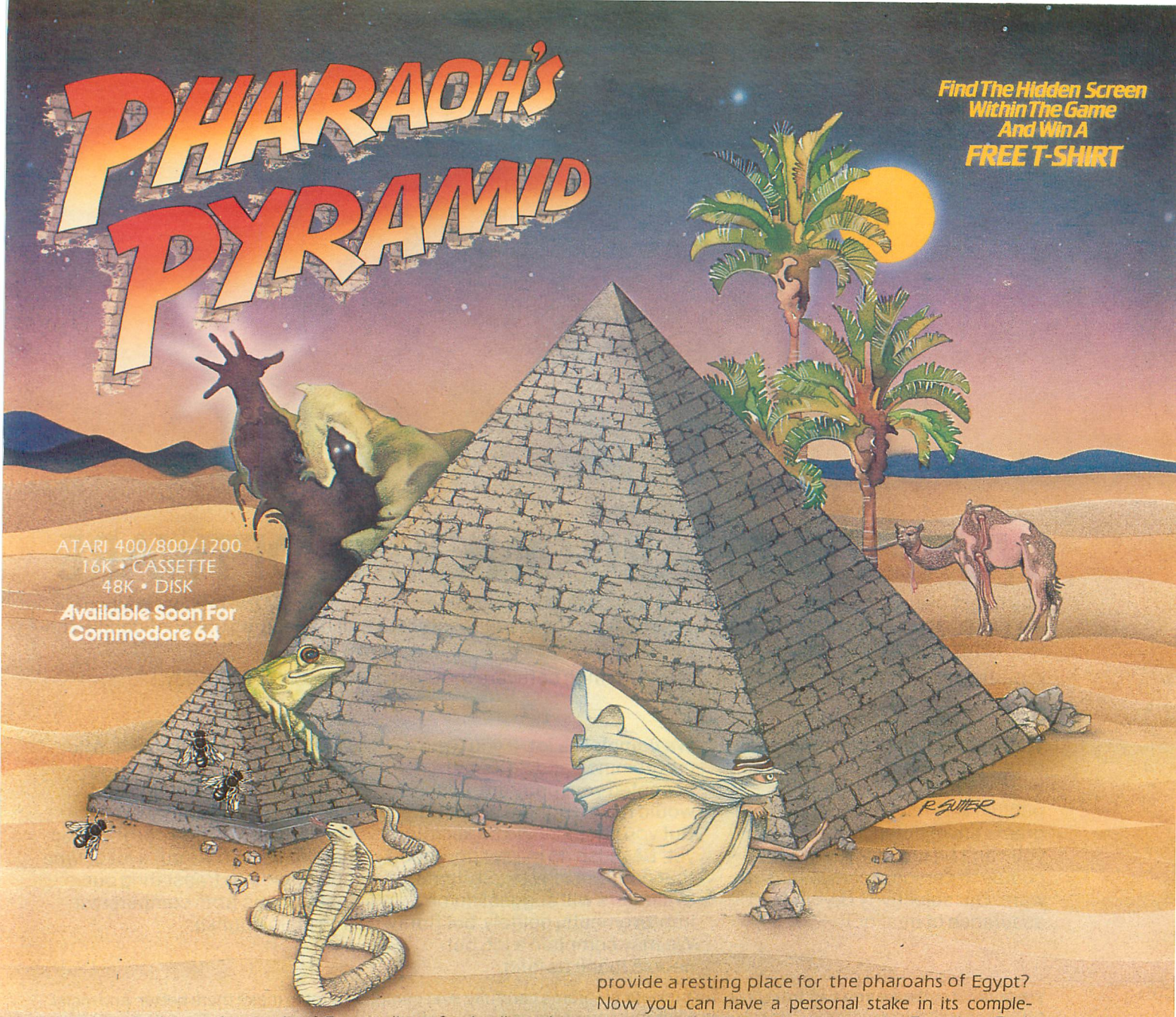
At times during the game a red heart will appear, and by touching it before it disappears, you can stun the Trollaboars for 5 seconds, then run to them and temporarily trap them in a square. If a blue diamond appears, you can touch this and open a port to the place here Drelbs are held captive. The other way to get to the Drelb prison is by flipping all of the grids so that all possible squares are built, then quickly attempting to hop into the proper square.

Once into the prison, you try and free all of the other Drelbs by touching them. Unfortunately, here reside the Gorgolytes, and they will do their best to try and stop you. The game ends when you use up your 5 lives or free all of the Drelbs after 8 grueling levels.

If you are on the lookout for something different, **Drelbs** combines interesting graphics with a challenging scenario and should entertain a wide age group. □

PHARAOH'S PYRAMID

Find The Hidden Screen Within The Game And Win A FREE T-SHIRT

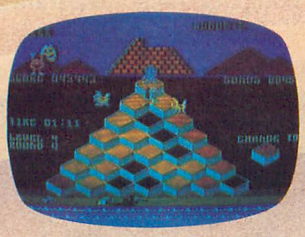


ATARI 400/800/1200
16K • CASSETTE
48K • DISK
Available Soon For Commodore 64

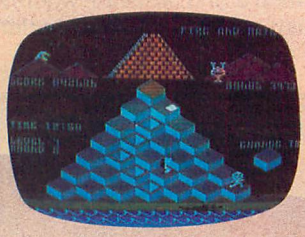
It sits on the sands of Egypt–silent, foreboding; the blue waters of the Nile sliding silently by. Pharoah's Pyramid beckons to us across millennia, mysterious, alluring. Where did the stones come from? How were they laid one on top of another? How many lives were lost under hot suns and cool moons to provide a resting place for the pharoahs of Egypt? Now you can have a personal stake in its completion. Match wits with the ancient Egyptian God, Osiris. Take advantage of gifts given you by the beautiful Isis. Fight through the plagues of Moses as brick by brick you build one of the wonders of the world. There hasn't been a greater challenge since the original Pharoah's Pyramid!

HISTORY PROVES THAT ALL GREAT GAMES HAVE THREE THINGS IN COMMON.

FAST-PACED FUN THAT LASTS



EVER INCREASING CHALLENGE



GRAPHICS THAT LIVE

PHAROAH'S PYRAMID is a fast action game with that special kind of challenge that makes you say... "This time I go all the way!"

AVAILABLE NOW FROM... Master Control Software Inc

The company that loves great games as much as you do, see your local dealer or call toll free. 1•800•624•5596

P.O. BOX 26714 • SALT LAKE CITY • UTAH • 84126

WE'RE VERY HARD ON OUR SOFTWARE

You might even say we're perfectionists. Because at Softsmith,™ we give our software the hardest workout, the toughest testing, the most rigorous evaluation. The result is software that has earned our confidence, and will justify your trust.

In particular, we do three things that make Softsmith software the most dependable you can buy.

1.

We're picky. Out of the hundreds of programs Softsmith evaluates every month, we choose to publish very few. A lot of good programs are rejected; but we think you can't be too picky when it comes to personal computer software. Our selectivity is your best assurance of quality.

2.

We complain a lot. If you were a programmer, and Softsmith accepted your program, you would have a right to be proud. But you shouldn't go on vacation yet. Because no matter how good that program may be, Softsmith evaluators will suggest some improvements; politely, but firmly. We may complain a lot, but people thank us later.

3.

We insist on plain English. After we've made the best program better, we're still not finished. Because we know that even the best program is no good if it's too hard to use. So we put a lot of time and effort into translating our instructions from computerese into plain English.

We publish software you can trust. Yes, we pick our programs carefully. And complain a lot to make them better. And insist on plain English instructions. The result is a library of personal computer software you can depend on. Even if you don't know a Pascal compiler from an emulation subroutine.

Softsmith has programs you can trust for all the most popular personal computers. Programs for Education, Home Management, Entertainment, Word Processing, Business, Communications and Programming. Ours is the largest library of quality software under one brand name.

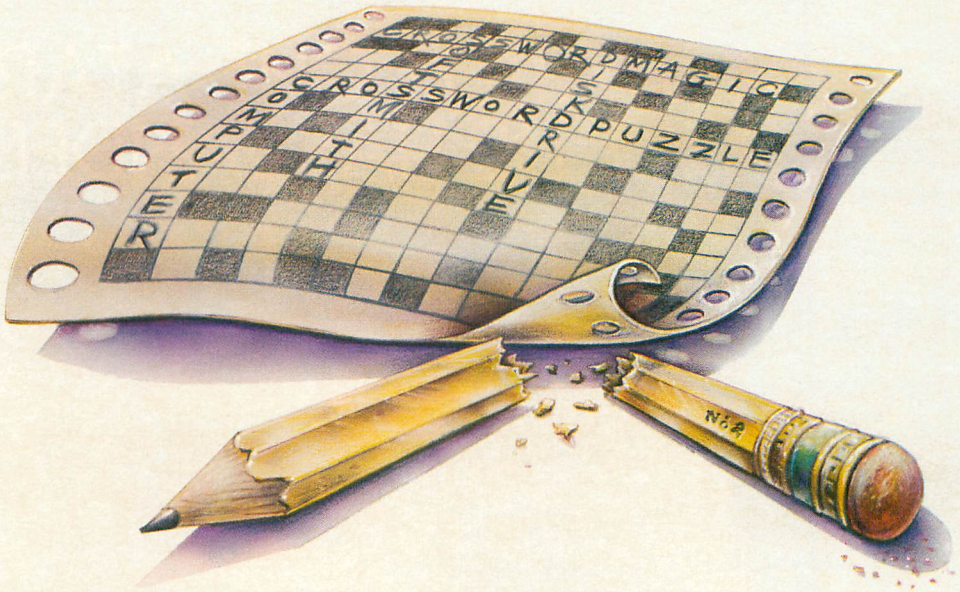
So before you choose a software package for your computer, make sure someone's taken the time to be hard on it. Make sure it's Softsmith, the software you can trust.

Ask for Softsmith brand software wherever computers or software are sold. Or call us TOLL-FREE at **(800) 341-4000** for the name and location of your nearest dealer.

Softsmith Corp., 1431 Doolittle Dr., San Leandro, CA 94577. A company of The Software Guild.™

SOFTSMITH™

SOFTWARE LIBRARY



CROSSWORD MAGIC™ A GAME DESIGNED TO GET YOU DOWN (AND ACROSS)

What's a five-letter word meaning report? The capital of a Baltic country, 6 letters, starts with W? If these questions get you thinking, you've obviously got the crossword bug. There is no cure, but Crossword Magic, from Softsmith™ Corporation, is just the treatment the doctor ordered. It is a challenging, intelligent game that will help you improve your vocabulary while you're having fun.

With Crossword Magic, you create your own puzzles to challenge your friends. First you choose a topic; then you enter words using the keyboard. Crossword Magic automatically fits the common letters together to form a familiar crossword grid. If there is no opening for the word you enter, the program saves it and fits it in later.

Next comes the fun. After the puzzle is full, Crossword Magic prompts you for a clue to go with each word. Make your clues as obscure, humorous or serious as you want. You can save the puzzle and clues on disk for later play on the screen, or have them printed out to send to friends.

Crossword Magic runs on the Apple® II, II + and IIe; Franklin Ace™ and Atari 800® computers with 48K RAM. A disk drive is required; printer is optional.

More than just a game, Crossword Magic is also an excellent educational tool for helping children of all ages improve their spelling and vocabulary skills. It is one of the programs in the Softsmith library of quality software — the largest library of programs under one brand name.

Softsmith has the most thoroughly tested, refined and clearly documented brand of personal computer software you can buy. You can depend on Softsmith programs for Education, Home Management, Entertainment, Word Processing, Business, Communications and Programming. And we back them with our Toll-Free customer service number.

Ask for Softsmith brand software wherever computers or software are sold. Or call us Toll-Free at (800) 341-4000 for the name and location of the dealer nearest you.

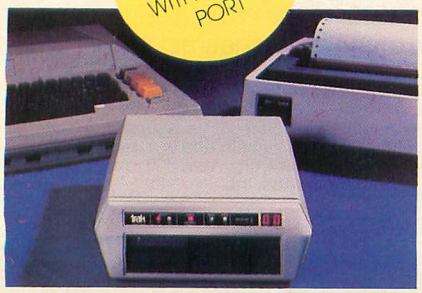
Dealer inquiries are invited.

SOFTSMITH™

SOFTWARE LIBRARY

WATCH OUT ATARI!
AT-D2 HAS LANDED!

AVAILABLE **NOW!** WITH PRINTER PORT



TRAK'S floppy and hard disk drive systems for the Apple and IBM-PC are available at many Computerland and other fine stores worldwide.

. . .WITH IT'S FLEET OF THE MOST ADVANCED SINGLE AND DOUBLE DENSITY DRIVE SYSTEMS IN THE UNIVERSE - TOTALLY COMPATIBLE WITH ATARI* COMPUTER MODELS 400, 800, 600XL, 1200XL, 1400XL, AND 1450XL.

DOCK YOUR FAVORITE PRINTER DIRECTLY TO OUR DRIVE'S BUILT-IN PARALLEL PRINTER INTERFACE. THERE'S NO NEED TO BUY AN EXPENSIVE ATARI 850 MODULE.

THE COMMAND CONTROL PANEL FEATURES READ, WRITE AND DENSITY INDICATORS, A TRACK COUNTER TO LOCATE EVERY BIT OF YOUR DATA AND A TOUCH-SENSITIVE SWITCH TO PROTECT IT.

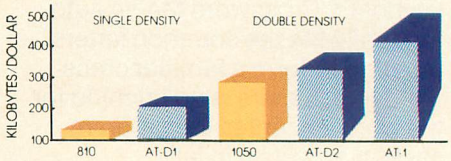
WARP SPEED OF UP TO 18 TIMES FASTER THAN **ANY** OTHER ATARI DRIVE IS ACHIEVED WITH OUR OPTIONAL TURBO-CHARGED SOFTWARE.

AWESOME POWER — DESTROY A PLANET OR LEARN HOW ONE IS CREATED. BALANCE YOUR CHECKBOOK OR WRITE A BESTSELLER. LEARN A NEW LANGUAGE OR CREATE ONE.

IT'S MAGNIFICENTLY SIMPLE! — JUST PLUG IT IN AND LIFT-OFF!

*Atari is a registered trademark of Atari, Inc.

TRAK GIVES YOU THE MOST STORAGE FOR YOUR DOLLAR!



OUR FLAGSHIP, AT-D2, WITH TWICE THE STORAGE CAPACITY OF THE ATARI 810 DRIVE PLUS A PRINTER INTERFACE AND COMMAND CONTROL PANEL, HAS A VALUE OF OVER $1000 YET COSTS LESS THAN $500!

SEE YOUR DEALER NOW FOR A TEST FLIGHT, OR

call toll free: **1-800-323-4853**
in Illinois, call collect: 1-312-968-1716
TWX 910-222-1848

trak™ microcomputer corporation 1511 Ogden Ave., Downers Grove, IL 60515

A commitment to quality

CONVERSIONS

16K Cassette 24K Disk

by Arthur A. Nevola

This program allows you to obtain some useful conversions without having to look them up in a book. The program will help you convert from U.S. Standard Measures to Metric, Metric to U.S., Standard U.S. Measures, Household Measures and a Metric Prefix Chart for other conversions not included in this program.

The program is quite simple to use. After you have typed it in, SAVE it either to tape or disk. When you are ready to RUN the program, the first screen you see will be the *Title Screen* followed by the MENU. If you wish to convert meters to feet, then you would choose Metric to U.S. Standard. Each of the listed categories contain only the most widely used conversions that we hear about each day. A large number of gasoline stations have their pumps dispensing in liters and not gallons. Do you know how many liters are in a gallon? This program will give the answer to this and many other problems requiring a form of converting.

After you have entered in your numbers, press the RETURN key for the answer, You will then be asked if you want to continue in the same field. If you do just keep entering in your figures for answers. If you do not want to stay in the same field, your entry of N will take you back to the MENU.

Also included is a Metric Prefix Chart which will help you to understand what each prefix used in the Metric system stands for. For example, a *Kilo*gram is 1000 grams but a *Milli*gram is 1/1000's of a gram.

I hope you enjoy using this program and that it makes converting problems easier and more enjoyable. □

```
5 GOSUB 5200
10 SETCOLOR 2,12,1:SETCOLOR 4,12,1
30 ? "K":? "      M E N U      ":? :? :?
60 ? "1. METRIC TO U.S. STANDARD":?
70 ? "2. U.S. STANDARD TO METRIC":?
80 ? "3. U.S. STANDARD MEASURES":?
90 ? "4. HOUSEHOLD MEASURES":?
95 ? "5. METRIC PREFIX CHART":? :?
110 POKE 752,1:? :? "CHOOSE ONE OF T
HE ABOVE"
120 OPEN #1,4,0,"K:":GET #1,S
140 IF S<49 OR S>53 THEN CLOSE #1:GOTO
 10
150 GOSUB 3370:ON S-48 GOTO 690,960,20
0,430,5000
200 OPEN #1,4,0,"K:"
220 ? "  U.S. STANDARD MEASURES  "
230 ? :? :? "1. INCHES TO FEET"
240 ? "2. FEET TO YARDS"
250 ? "3. SQUARE INCHES TO SQUARE FEE
T"
260 ? "4. SQUARE FEET TO SQUARE YARDS
"
270 ? "5. FEET TO MILES"
280 ? "6. SQUARE FEET TO ACRES"
290 ? "7. ACRES TO SQUARE FEET":?
300 ? "8. [RETURN TO MENU]":? :?
310 ? "  CHOOSE ONE OF THE ABOVE  "
320 GET #1,F
330 IF F<49 OR F>56 THEN ? "K":GOTO 22
0
340 GOSUB 3370:ON F-48 GOTO 2250,2320,
2390,2460,2530,2600,2670,10
430 OPEN #1,4,0,"K:"
450 ? "  HOUSEHOLD CONVERSIONS  ":? :?
470 ? :? "1. TEASPOONS TO TABLESPOONS
"
480 ? "2. TABLESPOONS TO CUPS"
490 ? "3. CUPS TO FLUID OUNCES"
500 ? "4. CUPS TO PINTS"
510 ? "5. CUPS TO QUARTS"
520 ? "6. PINTS TO QUARTS"
530 ? "7. PINTS TO GALLONS"
540 ? "8. QUARTS TO GALLONS":?
550 ? "9. [RETURN TO MENU]":? :?
560 ? "  CHOOSE ONE OF THE ABOVE  "
570 GET #1,D
580 IF D<49 OR D>57 THEN ? "K":GOTO 45
0
```

```
590 GOSUB 3370:ON D-48 GOTO 2740,2810,
2880,2950,3020,3090,3160,3230,10
690 OPEN #1,4,0,"K:"
710 ? "▶ METRIC TO U.S. ":? :? :?
730 ? "▶1. CENTIMETERS TO INCHES"
740 ? "▶2. METERS TO FEET"
750 ? "▶3. KILOMETERS TO MILES"
760 ? "▶4. KILOGRAMS TO POUNDS"
770 ? "▶5. GRAMS TO OUNCES"
780 ? "▶6. LITERS TO QUARTS"
790 ? "▶7. LITERS TO GALLONS"
800 ? "▶8. DEGREES CELCIUS TO FAHRENHE
IT"
810 ? :? "▶9. [Return to Menu]":? :?
830 ? "▶ CHOOSE ONE OF THE ABOVE "
840 GET #1,5
850 IF 5<49 OR 5>57 THEN ? "K":GOTO 71
0
860 GOSUB 3370:ON 5-48 GOTO 1180,1250,
1320,1400,1470,1540,1680,1610,10
960 OPEN #1,4,0,"K:"
980 ? "▶ U.S. STANDARD TO METRIC ":?
990 ? :? "▶1. INCHES TO CENTIMETERS"
1000 ? "▶2. FEET TO METERS"
1010 ? "▶3. MILES TO KILOMETERS"
1020 ? "▶4. POUNDS TO KILOGRAMS"
1030 ? "▶5. OUNCES TO GRAMS"
1040 ? "▶6. QUARTS TO LITERS"
1042 ? "▶7. GALLONS TO LITERS"
1050 ? "▶8. DEGREES FAHRENHEIT TO CELC
IUS"
1060 ? :? "▶9. [RETURN TO MENU]":? :?
1070 ? "▶ CHOOSE ONE OF THE ABOVE "
1080 GET #1,A
1090 IF A<49 OR A>57 THEN ? "K":GOTO 9
80
1100 GOSUB 3370:ON A-48 GOTO 1750,1820
,1890,1960,2030,2100,2241,2180,10
1180 ? "▶ CENTIMETERS TO INCHES ":? :?
1210 ? "HOW MANY CENTIMETERS";:INPUT C
1230 ? :? C;" CENTIMETERS = ";C*0.39;"
 INCHES"
1240 GOSUB 3300:GOTO 1210
1250 ? "▶ METERS TO FEET ":? :?
1280 ? "HOW MANY METERS";:INPUT M:?
1300 ? M;" METERS = ";M*3.28;" FEET"
1310 GOSUB 3300:GOTO 1280
1320 ? "▶ KILOMETERS TO MILES ":? :?
1360 ? "HOW MANY KILOMETERS";:INPUT K
1380 ? :? K;" KILOMETERS = ";K*0.62;"
MILES"
1390 GOSUB 3300:GOTO 1360
1400 ? "▶ KILOGRAMS TO POUNDS ":? :?
1430 ? "HOW MANY KILOGRAMS";:INPUT K
1450 ? :? K;" KILOGRAMS = ";K*2.2;" PO
UNDS"
1460 GOSUB 3300:GOTO 1430
1470 ? "▶ GRAMS TO OUNCES ":? :?
1500 ? "HOW MANY GRAMS";:INPUT G:?
1520 ? G;" GRAMS = ";G*0.035;" OUNCES"
1530 GOSUB 3300:GOTO 1500
1540 ? "▶ LITERS TO QUARTS ":? :?
1570 ? "HOW MANY LITERS";:INPUT L:?
1590 ? L;" LITERS = ";L*1.0567;" QUART
S"
1600 GOSUB 3300:GOTO 1570
1610 ? " DEGREES CELCIUS TO FAHRENHEIT
":? :?
1640 ? "DEGREES CELCIUS";:INPUT D:?
1660 ? D;" CELCIUS = ";D*1.8+32;" FAHR
ENHEIT"
1670 GOSUB 3300:GOTO 1640
1680 ? "▶ LITERS TO GALLONS ":? :?
1710 ? "HOW MANY LITERS";:INPUT LL:?
1730 ? LL;" LITERS = ";LL*0.264175;" G
ALLONS"
1740 GOSUB 3300:GOTO 1710
1750 ? "▶ INCHES TO CENTIMETERS ":?
1780 ? :? "HOW MANY INCHES";:INPUT I
1800 ? :? I;" INCHES = ";I*2.54;" CENT
IMETERS"
1810 GOSUB 3300:GOTO 1780
1820 ? "▶ FEET TO METERS ":? :?
1850 ? "HOW MANY FEET";:INPUT F:?
1870 ? F;" FEET = ";F*0.3048;" METERS"
1880 GOSUB 3300:GOTO 1850
1890 ? "▶ MILES TO KILOMETERS ":? :?
1920 ? "HOW MANY MILES";:INPUT M:?
1940 ? M;" MILES = ";M*1.609;" KILOMET
ERS"
1950 GOSUB 3300:GOTO 1920
1960 ? "▶ POUNDS TO KILOGRAMS ":? :?
1990 ? "HOW MANY POUNDS";:INPUT P:?
2010 ? P;" POUNDS = ";P*0.45;" KILOGRA
MS"
2020 GOSUB 3300:GOTO 1990
2030 ? "▶ OUNCES TO GRAMS ":? :?
2060 ? "HOW MANY OUNCES";:INPUT O:?
2080 ? O;" OUNCES = ";O*28.35;" GRAMS"
2090 GOSUB 3300:GOTO 2060
2100 ? "▶ QUARTS TO LITERS ":? :?
2140 ? "HOW MANY QUARTS";:INPUT Q:?
2160 ? Q;" QUARTS = ";Q*0.946;" LITERS
"
2170 GOSUB 3300:GOTO 2140
2180 ? "▶ FAHRENHEIT TO CELCIUS ":? :?
2210 ? "DEGREES FAHRENHEIT";:INPUT D:?
2230 ? D;" FAHRENHEIT = ";5/9*(D-32);"
 CELCIUS"
2240 GOSUB 3300:GOTO 2210
2241 ? "▶ GALLONS TO LITERS ":? :?
2244 ? "HOW MANY GALLONS";:INPUT GA:?
2246 ? GA;" GALLONS = ";GA*3.785;" LIT
ERS"
2247 GOSUB 3300:GOTO 2243
2250 ? "▶ INCHES TO FEET ":? :?
2280 ? "HOW MANY INCHES";:INPUT I:?
2300 ? I;" INCHES = ";I/12;" FEET"
2310 GOSUB 3300:GOTO 2280
2320 ? "▶ FEET TO YARDS ":? :?
2350 ? "HOW MANY FEET";:INPUT F:?
2370 ? F;" FEET = ";F/3;" YARDS"
2380 GOSUB 3300:GOTO 2350
2390 ? "▶ SQUARE INCHES TO SQUARE FEET
":? :?
2420 ? "HOW MANY SQUARE INCHES";:INPUT
 S:?
2440 ? S;" SQUARE INCHES = ";S/144;" S
QUARE FEET"
2450 GOSUB 3300:GOTO 2420
2460 ? "▶ SQUARE FEET TO SQUARE YARDS
":? :?
2490 ? "HOW MANY SQUARE FEET";:INPUT S
:?
2510 ? S;" SQUARE FEET = ";S/9;" SQUAR
E YARDS"
2520 GOSUB 3300:GOTO 2490
2530 ? "▶ FEET TO MILES ":? :?
2560 ? "HOW MANY FEET";:INPUT F:?
2580 ? F;" FEET = ";F/5280;" MILES"
2590 GOSUB 3300:GOTO 2560
2600 ? "▶ SQUARE FEET TO ACRES ":? :?
2630 ? "HOW MANY SQUARE FEET";:INPUT S
:?
2650 ? S;" SQUARE FEET = ";S/43560;" A
CRES"
2660 GOSUB 3300:GOTO 2630
2670 ? "▶ ACRES TO SQUARE FEET ":? :?
2700 ? "HOW MANY ACRES";:INPUT A:?
2720 ? A;" ACRES = ";A*43560;" SQUARE
FEET"
2730 GOSUB 3300:GOTO 2700
2740 ? "▶ TEASPOONS TO TABLESPOONS ":?
 :?
2770 ? "HOW MANY TEASPOONS";:INPUT T:?
2790 ? T;" TEASPOONS = ";T/3;" TABLESP
OONS"
2800 GOSUB 3300:GOTO 2770
2810 ? "▶ TABLESPOONS TO CUPS ":? :?
2840 ? "HOW MANY TABLESPOONS";:INPUT T
:?
2860 ? T;" TABLESPOONS = ";T/16;" CUPS
"
2870 GOSUB 3300:GOTO 2840
2880 ? "▶ CUPS TO FLUID OUNCES ":? :?
2910 ? "HOW MANY CUPS";:INPUT C:?
2930 ? C;" CUPS = ";C*8;" FLUID OUNCES
"
2940 GOSUB 3300:GOTO 2910
2950 ? "▶ CUPS TO PINTS ":? :?
2980 ? "HOW MANY CUPS";:INPUT C:?
3000 ? C;" CUPS = ";C/2;" PINTS"
3010 GOSUB 3300:GOTO 2980
3020 ? "▶ CUPS TO QUARTS ":? :?
```

```
3050 ? "HOW MANY CUPS";:INPUT C:?
3070 ? C;" CUPS = ";C/4;" QUARTS"
3080 GOSUB 3300:GOTO 3050
3090 ? "▶ PINTS TO QUARTS ":? :?
3120 ? "HOW MANY PINTS";:INPUT P:?
3140 ? P;" PINTS = ";P/2;" QUARTS"
3150 GOSUB 3300:GOTO 3120
3160 ? "▶ PINTS TO GALLONS ":? :?
3190 ? "HOW MANY PINTS";:INPUT P:?
3210 ? P;" PINTS = ";P/8;" GALLONS"
3220 GOSUB 3300:GOTO 3190
3230 ? "▶ QUARTS TO GALLONS ":? :?
3260 ? "HOW MANY QUARTS";:INPUT Q:?
3280 ? Q;" QUARTS = ";Q/4;" GALLONS"
3290 GOSUB 3300:GOTO 3260
3300 CLOSE #1
3310 OPEN #1,4,0,"K:"
3320 ? :? :? "▶ CONTINUE IN THIS FIELD
 [Y/N] "
3330 GET #1,T
3340 IF T<>78 AND T<>89 THEN 3320
3350 IF T=78 THEN CLOSE #1:? "N":GOTO
10
3360 IF T=89 THEN ? :RETURN
3370 CLOSE #1:? "N";:RETURN
5000 OPEN #1,4,0,"K:"
5010 ? "▶ METRIC PREFIX CHART ":? :?
5020 ? "PREFIX","SYMBOL","MEANING":?
5025 ? "tera","T","1,000,000,000,000"
5030 ? "giga","G","    1,000,000,000"
5035 ? "mega","M","        1,000,000"
5040 ? "kilo","k","  1,000"
5045 ? "hecto","h","    100"
5050 ? "deka","da","     10"
5055 ? "deci","d","      .1"
5060 ? "centi","c","     .01"
5065 ? "milli","m","    .001"
5070 ? "micro","u","            .000 001"
5075 ? "nano","n","       .000 000 001"
5080 ? "pico","p"," .000 000 000 001"
5085 ? "femto","f       .000 000 000 00
0 001"
5090 ? "atto","a  .000 000 000 000 000
 001"
5095 ? :? :? :? "▶PRESS 0 FOR MENU "
5100 GET #1,R
5105 IF R<>48 THEN 5100
5110 GOSUB 3370:GOTO 10
5200 GRAPHICS 17:SETCOLOR 4,0,0
5210 POSITION 5,5:? #6;"CONVERSION"
5220 POSITION 6,7:? #6;"PROGRAMS"
5230 FOR I=1 TO 150:NEXT I
5240 POSITION 1,19:? #6;"BY ARTHUR A.N
EVOLA":FOR I=1 TO 100:NEXT I:GRAPHICS
0:RETURN
```

•

CHECKSUM DATA

(See p. 58)

```
5 DATA 648,648,22,985,167,3,219,913,57
5,310,164,603,260,540,145,6202
240 DATA 550,631,422,525,215,499,710,9
86,555,44,169,273,871,571,413,7434
490 DATA 293,628,506,713,941,285,730,5
,572,68,953,295,466,541,545,7541
750 DATA 468,605,686,241,34,564,701,2,
584,134,931,292,216,98,419,5975
1010 DATA 199,115,763,741,906,542,558,
752,839,918,705,47,27,593,331,8036
1250 DATA 746,766,386,338,397,222,112,
347,79,863,426,347,627,408,272,6336
1530 DATA 347,999,780,85,354,869,700,4
97,361,11,51,506,361,629,681,7231
1800 DATA 794,368,955,586,305,375,319,
453,162,375,253,798,999,354,522,7618
2060 DATA 750,425,333,26,761,798,335,1
08,145,466,335,194,924,59,358,6017
2250 DATA 707,704,122,342,900,573,579,
349,810,949,270,349,267,331,968,8220
2520 DATA 356,903,580,885,363,120,331,
334,363,127,405,151,363,274,221,5776
2790 DATA 814,370,332,383,872,377,363,
642,754,377,981,649,622,356,30,7922
3050 DATA 621,957,335,726,466,292,335,
95,473,87,342,28,768,595,349,6469
3300 DATA 866,456,928,885,9,142,791,71
7,450,265,817,244,29,946,367,7912
5045 DATA 292,211,216,278,356,988,87,1
49,611,746,264,876,640,893,185,6792
5210 DATA 496,402,314,904,2116
```

•

WHAT IS D:CHECK/C:CHECK?

Most program listings in **A.N.A.L.O.G.** are followed by a table of numbers appearing as DATA statements, called "CHECKSUM DATA." These numbers are to be used in conjunction with D:CHECK, which appeared in issue no. 10, and C:CHECK, which appeared in issue no. 11.

D:CHECK and C:CHECK are programs by Istvan Mohos and Tom Hudson. They are designed to find and correct typing errors when entering programs from the magazine. For those readers who do not have a copy of either article, send a pre-addressed, stamped, business-sized envelope to:

D:CHECK ARTICLE
P.O. BOX 23
WORCESTER, MA 01603

Ask Sally Forth

by Sally Forth

My nerdy little brother says that a program written in FORTH can take up less memory than the same program written in machine code! I bet him a week's allowance that he's wrong. Do I win?

Nervous in Nevada

Looks like you'll have to split the kitty, kids. It *is* possible for a FORTH program to take up less space than machine code — but it isn't very likely on an ATARI computer. Let's take a look at the factors that control the code efficiency of FORTH and M/L programs.

At the innermost core of every FORTH system is a block of machine-language routines called the **kernel**. The kernel can be thought of as FORTH's operating system; it maintains the stacks, controls memory allocation and performs all the dirty little housekeeping duties that make FORTH look like FORTH. When you compile new words into a FORTH dictionary, all you are doing is defining new execution patterns for the fundamental FORTH routines inside the kernel.

Each of the FORTH systems available for the ATARI come with a "bare bones" kernel of fixed size. For example, the **valFORTH 1.1** kernel takes up about 7.5K of RAM; the Team ATARI fig-FORTH kernel, about 8.9K. Because the kernel must be permanently linked with your program in order for it to run, the size of the kernel determines the absolute minimum size of your program. So even if you wrote a **valFORTH** program consisting of just one word:

```
( Change background color to black )
: PROGRAM 0 710 C! ;
```

the final product would still occupy at least 7.5K!

Machine language laughs at the idea of a kernel. It requires no overhead at all because it communicates directly to the 6502 microprocessor at the lowest possible level. In fact, you can write a little routine in assembly like this one:

```
A900      LDA #0
8DC602    STA 710
```

that performs exactly the same function as our FORTH word PROGRAM in only five bytes.

It's the size that counts.

When you compare the sparseness of machine code to FORTH's kernel overhead, it's hard to conceive of an application where FORTH would be more efficient. Yet even as the kernel taketh away, the kernel giveth — in the form of generality.

A FORTH kernel is like a Swiss Army knife. It consists of a number of all-purpose tools (subroutines) built around a simple, versatile control structure (the stack). A knowledgeable FORTH programmer can exploit the built-in features of the kernel to concisely implement all sorts of elaborate procedures, much as an assembly hacker uses OS subroutines as often as possible to simplify his work. But a FORTH kernel is far more versatile than a machine-specific OS; and although the initial size of a FORTH program is large, its threaded structure

Lyco Computer Marketing & Consultants

TO ORDER CALL US

TOLL FREE 800-233-8760

In PA 1-717-327-1824

600XL $$$$$149.00
800 XL... $CALL$
1200 XL. $499.00
1400 XL . $CALL$
1450 XL . $CALL$

1025 Printer $399.00
1020 Color Printer . $245.00
1027 Printer $299.00
1010 Recorder $75.00
410 Recorder $75.00
810 Disk Drive $399.00
1050 Disk Drive $335.00
850 Interface..... $CALL$

ADVENTURE

Preppie..... $19.95
Preppie 2..... $19.95
Diskey..... $39.95
Sea Dragon $25.50
Stratos $25.00
Treasure Quest $13.50
Series 1-12..... $15.95
Saga Series 1-6..... $24.95
Stone of Sisyphus $29.95
Eliminator..... $18.95

BOOK of ATARI SOFTWARE 1983

346 pages $16.75

CX415 HOME FILING MANAGER $41 75
CXL4007 MUSIC COMP $33 75
CXL4002 ATARI BASIC $45 75
CX8126 MICROSOFT $65 75
CX4119 FRENCH $45 00
CX4118 GERMAN $45 00

PARKER BROTHERS

Tutankham R..... $33.75
Super Cobra R $33.75
Astro Chase R..... $33.75
Frogger R $33.75
QBert R $33.75
Popeye R..... $33.75
Risk R $42.75
Chess R $42.75

SPINNAKER

Story Machine R $26.75
Face Maker R $24.75
Kinderomp R..... $20.75
Fraction Fever R $24.75
Delta Drawing R..... $26.75

EASTERN HOUSE

Monkey Wrench 2 ..... $52.75

INHOME

Baseball..... $29.95

IDSI

Speedway Blast..... $29.95
Pool 1.5..... $26.95

GALAXIAN $29.75
DEFENDER $29.75
DIG DUG $29.75
SPEED READING $53.75
ATARI WRITER..... $54.75
BOOKKEEPER..... $102.75
CX4018 PILOT HOME $54.75
CX 405 PILOT EDU..... $91.75
CX404 WORD PRO $99.75
CXL4013 ASTEROID $25.75
CXL4020 CENTIPEDE $29.75
CXL4022 PACMAN $29.75
CXL4011 STAR RAIDER $29.75
CXL4004 BASKETBALL $25.75
CXL4006 SUPER BREAK $25.75

SSI

Battle of Shilo C/D..... $26.75
Tigers in the Snow C/D.... $26.75
Battle for Normandy C/D .. $26.75
Knights of the Desert C/D . $26.75
Cosmic Balance C/D $26.75

ON-LINE

Frogger $24.95
Wizard & Prin $26.95

ROKLAN

Wizard of War..... $29.75
Gorf $29.75
Delux Invader $27.95

BIG 5

Miner 2049..... $32.75

APX

Eastern Ft.41 ..... $25.50
DeRay Atari $19.95
Math-Tic-Tac $15.95
Pres of US..... $15.95
3R Math $19.95
Typo Attack $24.95
Family Budget $19.95
F. Cash Flow..... $19.95

BRODERBUND

Bank Street Writer D..... $44.75
AE D..... $24.75
Apple Panic D..... $23.75
Choplifter ROM $32.75
David's Midnight..... $24.75
Stellar Shuttle C/D..... $18.75
Ft. Apocalypse..... $24.75

HES

Gridrunner R..... $27.75
Sword Point D..... $24.75

INFOCOM

Zork 1/2/or 3 ..... $29.95

BUSINESS

Visicalc $89.75
Letter Perfect..... $89.75
Letter Perfect..... $89.75
Data Perfect..... $99.75
Text Wizzard $49.75
Spell Wizzard $64.75
File Manager..... $69.75
Home File Mgr..... $69.75
Bookeeper..... $119.75
C.R.I.S..... $199.75
Atari Word Pro..... $109.75
Tax Advantage $35.75
Home Accountant $59.75
Bank Street W $49.75
ATARI WRITER..... $74.75

32K RAM $65.75
48K RAM $89.75
64K RAM..... $109.75
TECHNICAL NOTE $29.75
B KEYBOARD $79.75

BLANK DISKETTES

ELEPHANT SS/SD..... $18.25
ELEPHANT SS/DD..... $21.75
KANGAROO SS/DD with storage case $24.75
MAXELL MD I $29.75
MAXELL MD II $39.75
DISK CASE (holds 10) $4 95
DISK CASE (holds 50) $19 75
ROM CASE (holds 10) $19 75

PERCOM

SINGLE DRIVE AT88 S1 ... $349.00
DUAL DRIVE AT88 S2 ..... $535.00
AT88 with printer port..... $449.00
AT88 Doubler Board $129.00
DUAL HEAD 44S1 ..... $575.00
DUAL DRIVE 44S2 ..... $899.00
SINGLE DRIVE 40S1 ..... $499.00
ADD ON DRIVES ..... $CALL

MODEMS

ANCHOR MARK I $74.75
MARK II $74.75
HAYES 1200 ..... $509.75
MICRO 2..... $274.75
SMART $214.75
NOVATION CAT $144.75
D-CAT..... $155.75
J-CAT..... $114.75
MICROBIT..... $159.75

MONITORS

NEC JB1260 ..... $115.00
NEC JB1201 ..... $145.00
NEC TC1201 ..... $315.00

Amdek Color I $275.00
Amdek 300 Green $149.00
Amdek 300 Amber $149.00
GORILLA GREEN $88.00
K TV (13" Color 1 yr. war.). $229.00

PRINTER RIBBONS AVAILABLE

PROWRITER 2P..... $679.00
GEMINI 15..... $439.00
PRINTMASTER $1589.00
SMITH CORONA TP1... $549.00
CITOH 8600B..... $1025.00
STARWRITER..... $1099.00
OKIDATA 82 ..... SAVE
OKIDATA 83 ..... LOWEST PRICE
OKIDATA 84P .... AVAILABLE
OKIDATA 93 ..... SAVE
TRACTOR..... $49.75
OKIDATA 92..... $SAVE

SAVE on these in-stock PRINTERS

GORILLA GX-100 ..... $195.00
GEMINI 10X $279.00
PROWRITER $349.00
NEC 8023..... $379.00
DELTA 10 ..... $479.00
EPSON LINE..... $$CALL$$
DIABLO 630 ..... $1749.00

MasterCard **TO ORDER** VISA

CALL TOLL FREE
800-233-8760
In PA 1-717-327-1824

or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore, PA 17740

POLICY

In-stock items shipped within 24 hours of order. Personal checks require four weeks clearance before shipping. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. PA residents add sales tax. All products subject to availability and price change. Advertised prices show 4% discount offered for cash, add 4% for Master Card or Visa. DEALER INQUIRIES INVITED.

makes it grow less quickly than a machine-language program. Theoretically, a point can be reached where FORTH and machine code take up the same amount of RAM (see **Figure 1**), after which the FORTH application will require less memory than the same job written in pure machine code.

When is this break-even point reached? It depends on the nature of the application, the power of the FORTH kernel and the skill of the programmer. Generally speaking, your FORTH program has to get fairly large before it will begin to compete with the best RAM-cramming efforts of an assembly hacker. That's why you're not likely to realize the potential efficiency of FORTH with a computer as small as the ATARI. On larger machines, however, FORTH can make a big difference when it comes to saving memory.

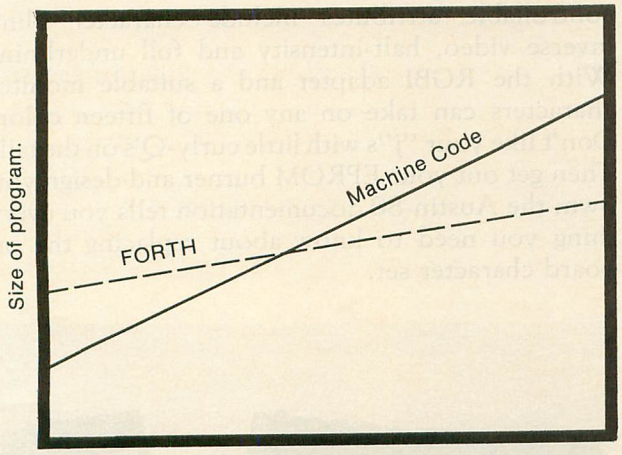


Figure 1.

One way to reduce the size of a FORTH program is to run it through a utility called a **target compiler.** A target compiler analyzes your application and strips away all the unnecessary gunk and dribble, leaving you with a tight package of object code that is smaller and maybe even a little faster than the original. None of the FORTH systems available for the ATARI offer a target compiler, although Valpar International is supposed to have a **valFORTH** compiler in the works.

As I mentioned above, the threaded architecture of FORTH makes a big contribution to its efficiency. But it takes good, structured programming techniques to realize this benefit. We'll discuss the controversial subject of *structured programming* in my next installment.

Sally welcomes your questions about the FORTH programming language, and will publish the most interesting letters in future columns. Write to her c/o A.N.A.L.O.G. Computing, P.O. Box 23, Worcester, MA 01603.

A NEW 80 COLUMN BOARD

by Brian Moriarty

AUSTIN-80 VIDEO PROCESSOR
AUSTIN FRANKLIN ASSOCIATES, Inc.
43 Grove Street
Ayer, Massachusetts 01432
Composite Monochrome $289.95
RGBI Color Adapter $69.95 additional
Requires ATARI 800 Computer

The ATARI's 40-column text format is fine for BASIC programming and an occasional letter with **AtariWriter**. But if you want to get serious about word processing or other business applications, sooner or later you're going to wish you had a professional 80-column screen. How do you get 80-columns on an ATARI, you ask? By adding an 80-column video board and an appropriate video monitor to your system, that's how.

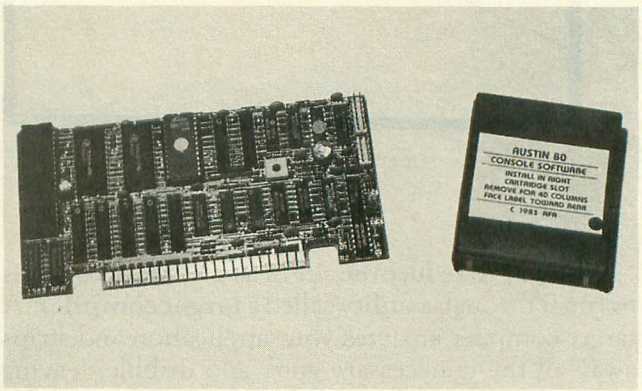
Austin Franklin Associates' new Austin-80 Video Processor is the latest entry in the ATARI 80-column market. AFA is best known to the ATARI community for their superbly engineered memory boards. Austin-80 continues this tradition, offering the serious ATARI 800 user features and capabilities found in no other product of its kind.

What it comes with.

A basic Austin kit consists of the main video processor board, a video I/O cable, a 4K ROM cartridge containing the 80-column software drivers and brief but complete instructions. Installation requires no tools and only a few minutes of your time. The video board goes into memory slot #3; if you're already using three memory boards, you'll need a new 32K or 48K board to replace the lost RAM. The I/O cable snaps into a connector on the video board and snakes out the back of your console, underneath the top cover. It's terminated with a 5-pin plug for the 800's monitor output, an RCA-type video output jack and a 9-pin input plug intended for a light pen. No sound output is provided, but a pin-out diagram in the documentation shows how to add one yourself.

The Austin-80 system requires a monochrome monitor of 12 MHz or better bandwidth for best performance. The big mail-order houses are currently selling 12 MHz "green screens" for less than $100. For color operation, you have to get AFA's $69.95 Color Adapter and an RGBI monitor, now available for under $400. You shouldn't use a color or black-and-white TV set or a composite color monitor with the Austin-80.

Each character in the 80-column by 25-line display is formed in an 8-by-10 dot matrix. The supplied character set is made up of 7-by-9 letters with true lower-case descenders. Software-controllable attributes include character blink, inverse video, half-intensity and full underlining. With the RGBI adapter and a suitable monitor, characters can take on any one of fifteen colors. Don't like your "j"s with little curly-Q's on the tails? Then get out your EPROM burner and design your own; the Austin-80 documentation tells you everything you need to know about replacing the on-board character set.

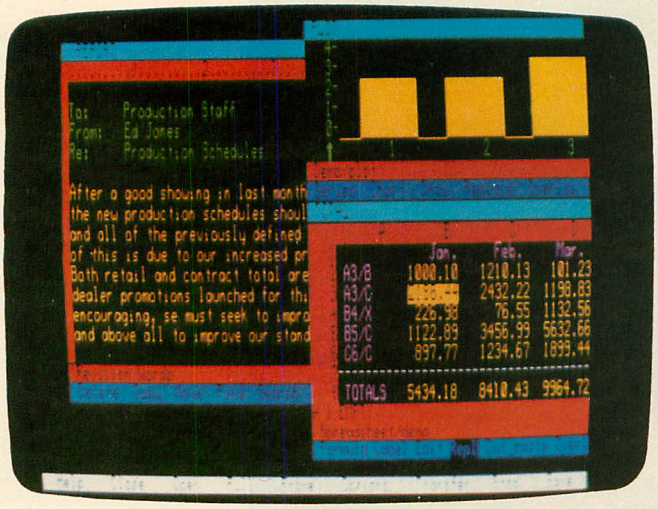


The Austin 80-column Board.

The special video cable included with the RGBI Adapter fits the 9-pin connector found on the back of virtually all new RGB monitors. You've got to see Austin-80 in RGB color to appreciate it. The hues are brilliant and highly saturated, with crisp, sharply-formed letters. I had a little trouble making out some of the inverse video characters, but the RGB monitor I used wasn't in perfect alignment. The same inverse characters looked great on my Amdek 300 green screen. (See the color screen shot on the next page.)

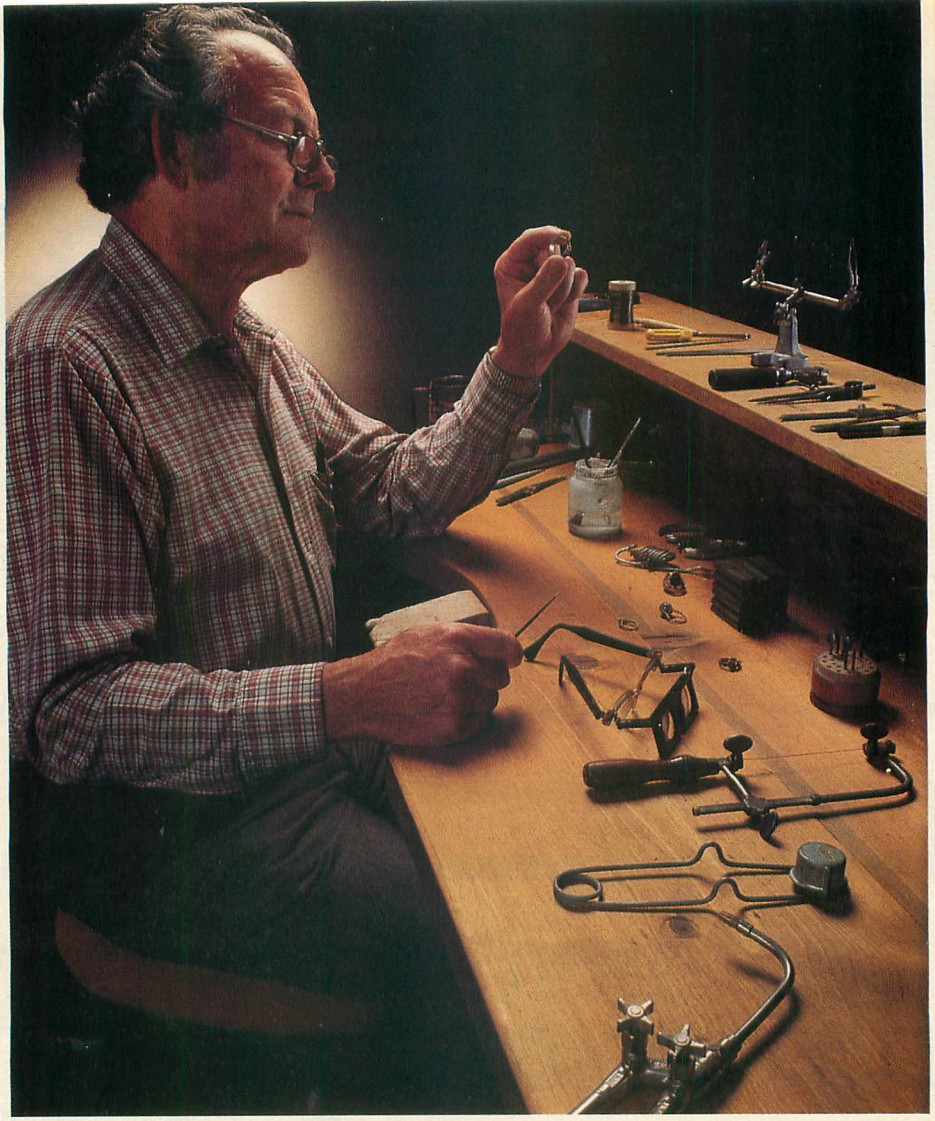
I do have a minor gripe with the Austin-80's blinking cursor. You can make it visible or invisible, but you can't make it stop blinking, even while

(continued on next page)



Sample screen display of the Austin 80-Column Board.

(review continued on page 65)



Craftsmen Need Precision Tools . . . Programmers! Demand Precision Software!

BASIC XL has twice the speed and twice the power of Atari® BASIC. And yet, as befits a fine craftsman's tool, BASIC XL is even easier to use and more dependable, while including such outstanding major *additions* as structured programming, string arrays, programming aids, enhanced graphics, and business capabilities.

Atari BASIC is a good starting point. We should know. We wrote it in 1978. Buy BASIC XL. Take advantage of five more years of experience!

So, prepare yourself for some exploration into imaginative programming with BASIC XL! Cartridge, excellent tutorial, reference manual . . . $99.

Functional, honest, and beautiful describe the simple lines of a craftsman's tools. For the jeweler these tools are an extension of the human hand to better execute complex designs. For you, the programmer, Precision Software tools keep complications out of your programming while allowing you to produce intricate programs.

See the complete collection of OSS Precision Software™ tools!

MAC/65: The fastest 6502 macro assembler/editor package on cartridge . . . $99.

BUG/65: A powerful debugger. On disk, with OS/A+ . . . $35.

C/65: The first native mode "small c" compiler for Atari and Apple® computers. On disk . . . $80.

ACTION!: The fastest, small computer language ever. A feature-packed cartridge at only . . . $99.

PRECISION SOFTWARE
A DIVISION OF O.S.S., INC.
1173D S. Saratoga/Sunnyvale Road
San Jose, CA 95129 • (408) 446-3099

you're typing. The purpose of a blinking cursor is to help you find your place after you've taken your eyes off the screen. It shouldn't continue to blink while you're moving around the screen, and I don't like it when it does. **Text Wizard**, **AtariWriter** and the 40-column **Letter Perfect** all stop blinking temporarily when you touch a key; so should the Austin-80.

Invisible ROM.

The Austin-80 ROM cartridge is an intriguing little device. This "video operating system" sits in the right cartridge slot and takes control of your 800 whenever you turn it on or hit SYSTEM RESET. It automatically reconfigures the ATARI's built-in screen handlers to make them work with the 80-column processor. Unlike a conventional ROM, the Austin-80 cartridge doesn't "steal" any of your precious memory — in fact, it actually gives you an extra 993 bytes of free workspace (!) by eliminating the display overhead normally required by a GRAPHICS 0 screen.

Austin-80 is smart enough to know when you don't need 80 columns. Any CIO GRAPHICS call other than mode 0 will instantly switch control of the display back to ANTIC and GTIA, so you can enjoy all the hi-res graphics and player/missiles you like. A firm press of the RESET key will drop you back into the Austin-80. And because the 80-column driver is on cartridge, you don't have to rip apart your entire system to play an occasional round of **Archon**. Just pull out the ROM and the video board disappears as far as your 800 is concerned.

What it works with.

The Austin-80 display handler is designed to work with any program that uses standard CIO calls to access a GRAPHICS 0 screen. In practical terms, that means it's very easy to write software for the Austin board. Just use the familiar BASIC PRINTs, INPUTs and POSITIONs (or their machine language equivalents) and everything will look just fine. Sneaky hackers who like to POKE things directly into screen RAM had better steer clear of this board, though.

Austin Franklin Associates is working with a number of major software publishers to create compatible versions of the best ATARI business software. Austin-ready versions of LJK's **Letter Perfect** and **Data Perfect** should be available by the time you read this. Synapse Software is reportedly preparing 80-column editions of their upcoming **SynApps** business series, including the **SynCale** spreadsheet, **SynFile** database and **SynText** word processor. If this integrated package is as good as some of Synapse's latest games, Austin users will have nothing to complain about. AFA also has a powerful 80-column telecommunications package in the works that fully supports the RGBI color option.

Everything published on disk by Optimized Systems Software (OSS) seems to work with Austin-80. This comes as no surprise, since Bill Wilkinson is a fanatic about sticking to CIO protocol. Compatible products include **MAC/65**, **BASIC A+**, **OS/A+** and **C/65**. Words cannot describe the joy of using **MAC/65** in eighty columns! Let's hope the upcoming cartridge versions of OSS BASIC and **MAC/65** stay nice and compatible.

On the negative side, the current release of OSS's bank-switching **Action**! cartridge doesn't take kindly to the Austin board, and neither do **Text Wizard**, **VisiCalc** or **valFORTH** (drat), but **Extended fig-FORTH** from APX is supposed to work okay. You can't use a 16K ROM cartridge in conjunction with the Austin ROM. That disqualifies **Atari Logo** and **Atari Microsoft BASIC II** from the 80-column sweepstakes. The **AtariWriter** cartridge takes a particularly dim view of the Austin-80. You have to completely remove the video processor board from slot #3 or it won't work at all. But if you're considering the purchase of an 80-column board, you probably need a word processor with more *oomph* than **AtariWriter** anyway.

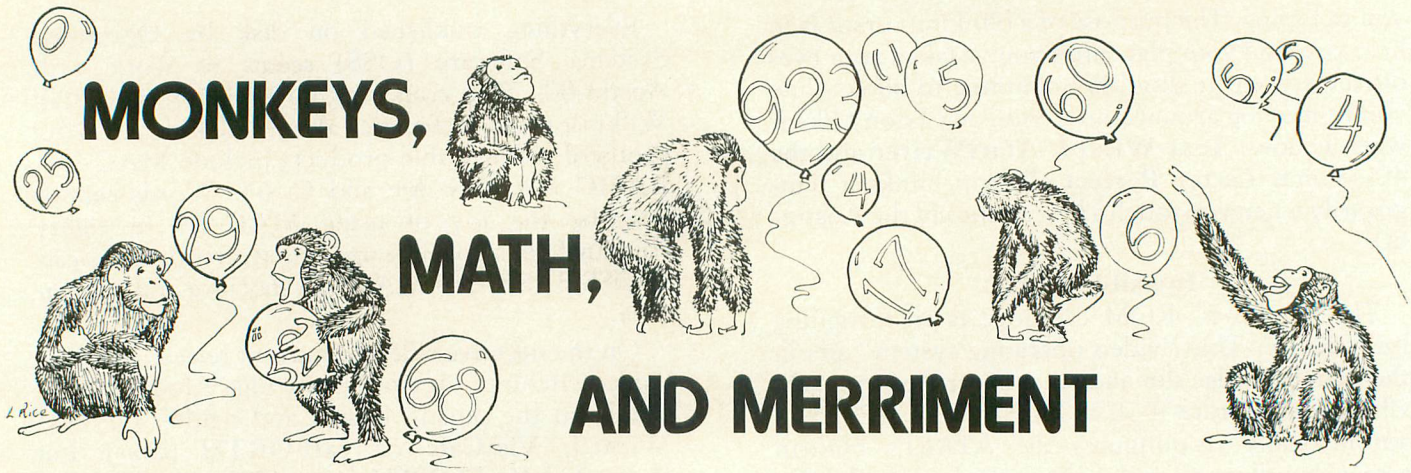
8K ATARI BASIC is almost totally happy to share its ROM cradle with the Austin cartridge. I say "almost" because Austin-80 limits the length of a logical line to 80 characters instead of the usual 120. If a line of BASIC code exceeds 80 characters (not uncommon), you won't be able to edit the line without getting a syntax error. This should pose no problems for new programs written with 80 columns in mind. But watch out for those mega-lines in your old programs. This warning also applies to **BASIC A+** users.

If you want a real selection of software goodies for your 80-column ATARI, check out the ATR8000 Module from Software Publishers. This periheral works together with the Austin-80 board to put an entire library of CP/M and MS-DOS software at your fingertips. I recently had the pleasure of booting up the legendary **WordStar** word processor on my lowly ATARI 800 system! Austin-80 and ATR8000 look like a fascinating combo; a closer inspection of the ATR8000 system will appear in a later issue.

Video value.

The Austin-80 Video Processor proves that professional computer hardware doesn't have to be expensive. Used with a decent monitor, it delivers a level of performance that equals VDTs costing many times as much as an ATARI 800 system. If your loyalty has been wavering lately, and you've caught yourself gazing wistfully at the IBM PCs in your local computer store, get a demo of the Austin-80 board on a good RGB monitor. It'll make you feel a whole lot better about owning an ATARI. □

MONKEYS, MATH, AND MERRIMENT



by Keith Valenza

Monkey Up A Tree
By Joe Grande
ATARI PROGRAM EXCHANGE
P.O. Box 3705
Santa Clara, CA 95055
24K Cassette or Disk $24.95

Monkeymath
By Dennis Zander
ARTWORX Software Co.
150 North Main St.
Fairport, NY 14550
16K Cassette ($24.95) or Diskette ($28.95)

If you're frustrated by the low quality of educational programs available for your ATARI, you may want to consult the latest **APX** and **Artwork Software** catalogs. Two recent releases — **Monkey Up a Tree** and **Monkey Math** are excellent learning games for elementary children. In both games, children learn basic arithmetic facts while playing fun games at the same time.

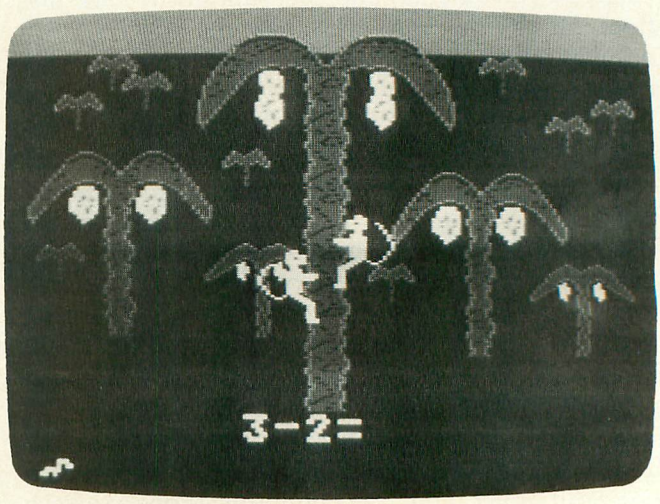
In APX's **Monkey Up a Tree**, the setting is a jungle with banana trees and one or two hungry monkeys, depending on the number of players. The object of the game is for the player's monkey to climb to the top of the tree and pick a banana. After the hungry ape takes his treasure, he jumps to the ground and eats it, tossing the peel aside. The player whose monkey reaches its goal three times is the winner. To do this, the player must solve a series of arithmetic problems which appear at the bottom of the screen.

When the player registers a correct number, the monkey climbs a short distance toward the top. If the player types the wrong answer or waits too long, the monkey slides down part of the way. After the incorrect answer has been typed, the correct answer is displayed at the bottom of the screen.

Adults will discover that they can compete equally with children without going easy on them because of the game's handicapping feature. This feature, which allows for individual differences in ability, is dependent upon the player's speed and accuracy. Skilled players quickly move on to more difficult problems. The program is designed to keep the players at the lowest level at which they show some difficulty.

With the exceptions of the brown monkeys, the yellow bananas, and the bright blue sky, the screen's colors are limited to various shades of green. However, this is a minor weakness in the program. The monkey's tail-wagging and banana-eating are nice graphic touches.

The author could have been more creative in his uses of sound effects. Only two are used in this game-the sound of the monkey sliding down the tree after an incorrect answer is typed, and a melody heard after every correct answer. The melody increases in pitch as the monkey nears the top of the tree.

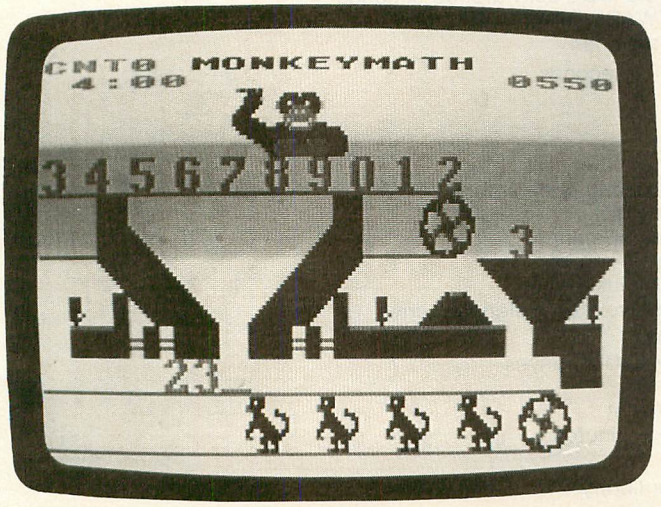


Monkey Up A Tree

If your children enjoy playing arcade games, they'll enjoy Artwork Software's **Monkeymath**. This game uses the elements of arcade action to teach basic arithmetic skills.

The setting of the game is a "number factory" run by apes. The high-resolution graphics include a conveyor belt (which contains the digits 0-9), a gorilla which stands above the conveyor belt, and chimpanzees which carry the correct answer off the screen.

Users have a choice of three skill levels and five operations — counting, addition, subtraction, multiplication, or division. When the problem appears at the bottom of the screen, the player uses the joystick to move the gorilla above the correct digit and presses the fire button. The number then falls through a chute to the bottom of the screen where the chimps are waiting for it. If the answer is correct, the player scores fifty points. The chimps then carry the answer off the screen and bring out a new problem. Incorrect answers are removed without penalty.



Monkey Math

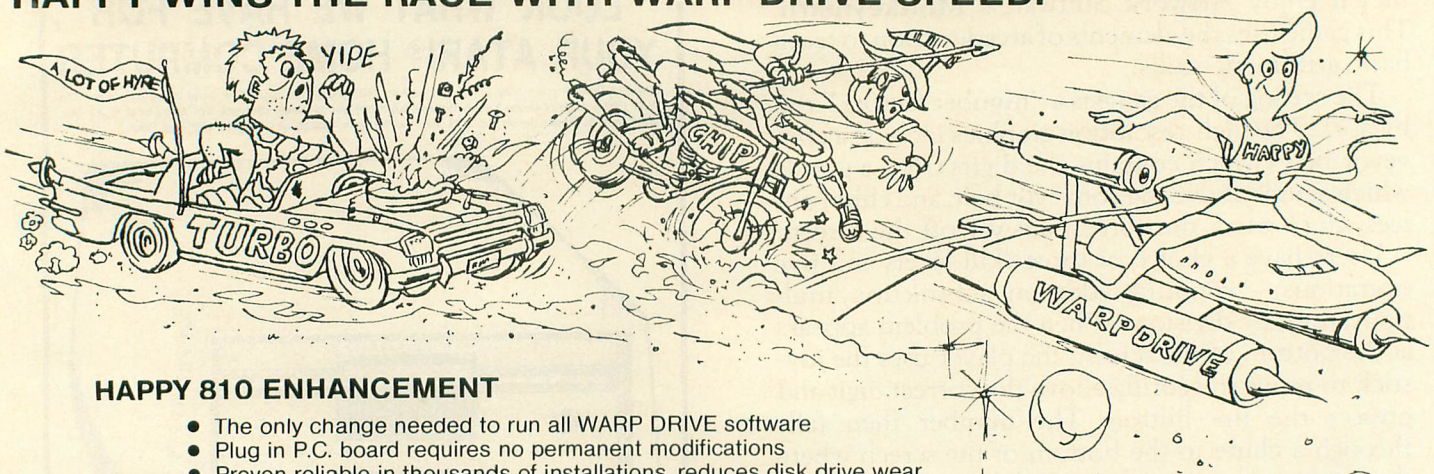
Although the documentation is fairly well-written, it does not fully explain how to play the counting game. However, with a little practice, children can easily figure out how to play it.

Each game, which lasts one minute, is the equivalent of one simulated eight-hour work shift. The "day" begins at eight o'clock, includes a "lunch" break at noon, and ends at five o'clock. During lunch and at the end of the shift, the gorilla eats "bonus bananas," which are awarded to each correct answer after the fifth one and score additional points.

The clever sound effects include these of the conveyor belt, and the blowing of the whistle at the beginning and at the end of each shift.

Monkeymath is a well-designed teaching game for elementary children. The most obvious arcade element — the incentive to score higher points as quickly as possible — will cause a child to memorize important math facts without realizing it. The faster he or she answers the question, the higher the score. □

HAPPY WINS THE RACE WITH WARP DRIVE SPEED!



HAPPY 810 ENHANCEMENT

- The only change needed to run all WARP DRIVE software
- Plug in P.C. board requires no permanent modifications
- Proven reliable in thousands of installations, reduces disk drive wear
- Comes completely assembled and tested, just plug in and use
- Full one year parts and labor guarantee, compatible with existing software
- High quality printed circuit board with gold connectors

NO ONE ELSE HAS THIS PERFORMANCE
Unenhanced whole disk (ATARI rev B format) read time: 112 seconds
Unenhanced whole disk (ATARI rev C fast format) read time: 89 seconds
ENHANCED 810 whole disk (any format) read time with standard software: 68 seconds
ENHANCED 810 whole disk (any format) read time with WARP DRIVE software: 43 seconds
Standard software whole disk write and verify time: 238 seconds
WARP DRIVE software whole disk write and verify time: 62 seconds

NEW HAPPY WARP DRIVE SOFTWARE

WARP SPEED HAPPY BACKUP PROGRAM
- Completely automatic: nothing to figure out, insert disks and press return
- Only program on the market guaranteed to backup any disk
- Can write to a blank disk: format write and verify in one operation
- Automatic program tracing: copies only the tracks that are used
- Efficient memory utilization: reduces the number of disk insertions
- Requires only one ENHANCED disk drive, backups will work on a standard drive

WARP SPEED MULTI DRIVE HAPPY BACKUP PROGRAM
- Same features as above plus support of multiple ENHANCED drives
- Can be used with up to 4 ENHANCED drives
- Source and all destination drives read and write in parallel
- Format write and verify 3 complete disks in less than 3 minutes

WARP SPEED HAPPY COMPACTOR PROGRAM
- Reduces the number of disks required to backup your library
- Combines up to 8 self booting disks into 1 disk with a menu
- Compacted disks run only on an ENHANCED drive
- Pays for itself by saving on disks
- Single or dual ENHANCED drive operation

HAPPY WARP DRIVE DOS
- Improves ATARI DOS 2.0S to use warp speed reading and write with verify
- Use all features of BASIC, PILOT, FMS, and DUP at top warp speed
- Warp speed I/O software module available separate from DOS

HAPPY WARP DRIVE SECTOR COPY PROGRAM
- Standard format whole disk read, write and verify in 105 seconds
- Use with sngle or dual drives, mix ENHANCED and NON-ENHANCED drives

HAPPY CUSTOMIZER PROGRAM (sold separately $99.95)
- Creates custom format disks of any specification
- Any type bad sector, duplicate sector numbers, or interleave
- Easy to use but requires an advanced level user to interpret the results

REVIEWED IN POPULAR MAGAZINES

A.N.A.L.O.G. COMPUTING—July/August 1983 *"...The installation instructions for the Happy 810 Enhancement are among the best I have ever seen. ...The Happy 810 Enhancement is one of the most powerful hardware modifications available to ATARI computer owners."*

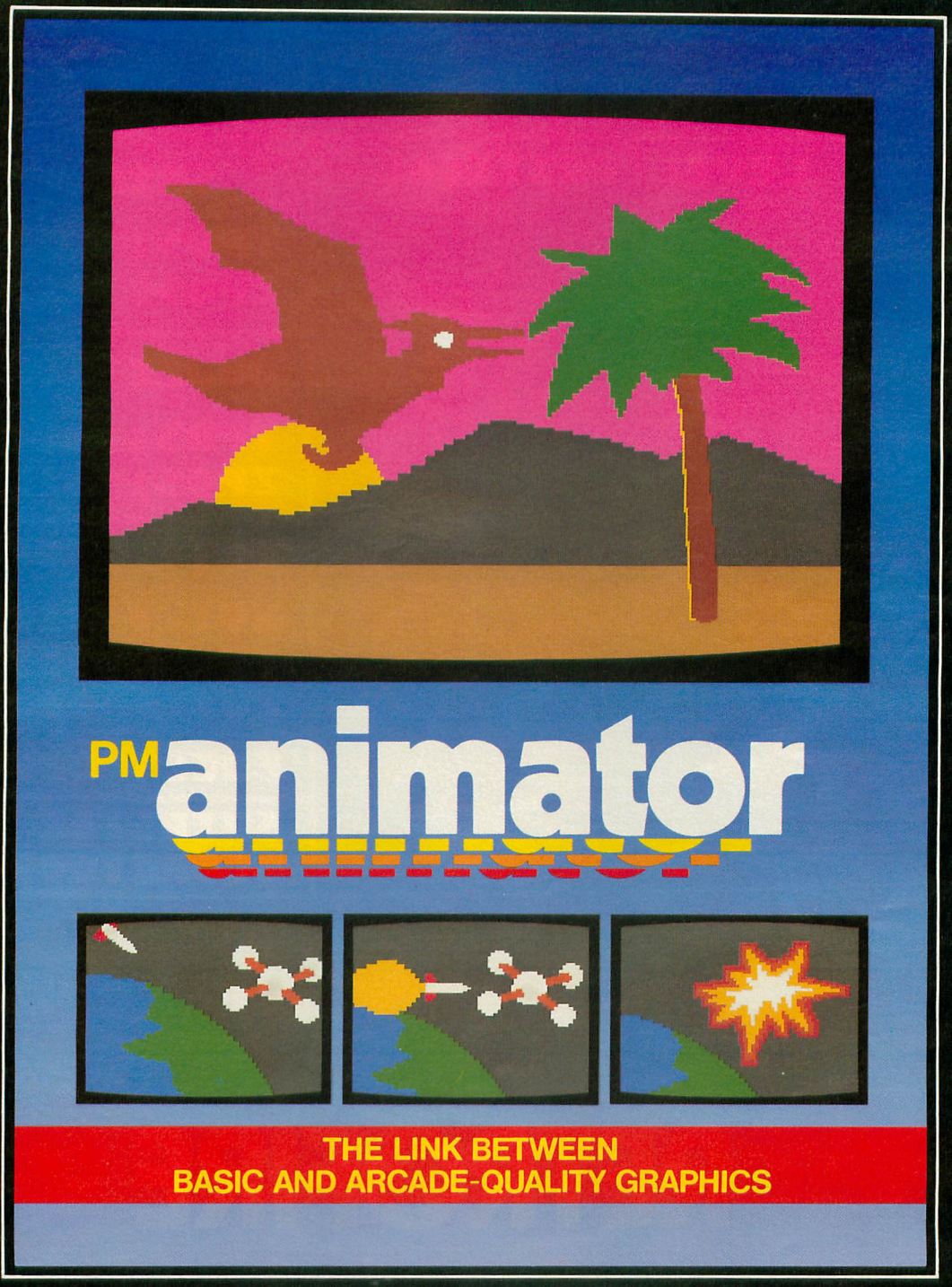
ANTIC—July 1983 *"The difference between a normal ATARI 810 disk drive and one equipped with Happy is like the contrast between mass transit and the automobile. A car costs you more initially, but improves the quality of your life. Similarly, if you use your disk drive a lot, installing Happy will markedly enhance your programming life."*

SPECIAL SUGGESTED RETAIL PRICE BEFORE DECEMBER 31, 1983: *Get the HAPPY 810 ENHANCEMENT with the single and multi drive HAPPY BACKUP PROGRAM, plus the HAPPY COMPACTOR PROGRAM, plus the HAPPY DRIVE DOS, plus the HAPPY SECTOR COPY, all with WARP DRIVE speed, including our diagnostic for $249.95. Existing registered ENHANCEMENT owners may upgrade to WARP DRIVE speed for $15.00 with no hardware changes.!*

Price includes shipping by air mail to U.S.A. and Canada. Foreign orders add $10.00 and send an international money order payable through a U.S.A. bank. California orders add $16.25 state sales tax. Cashiers check or money order for immediate shipment from stock. Personal checks require 2-3 weeks to clear. Cash COD available by phone order and charges will be added. No credit card orders accepted. ENHANCEMENTS for other ATARI compatible drives coming soon, call for information. Please specify -H model for all drives purchased new after February 1982, call for help in ENHANCEMENT model selection. Dealers now throughout the world, call for the number of the dealer closest to you.

HAPPY COMPUTING • P.O. Box 32331 • San Jose, California 95152 • (408) 251-6603

Get pm ANIMATOR
And put your fantasies on the screen.



A product from Don't Ask. Programmed by Roger Bush.

Produce dazzling Player-Missile animation on your Atari, just by writing a BASIC program. Make your creative ideas real.

All you need is **pm ANIMATOR**, the Player-Missile control kit from **DON'T ASK**.

All the colors, motions, and shapes you can make with Player-Missile Graphics are now available from BASIC. Just use **pm ANIMATOR**'s magical editors and subroutines. Draw pictures and animate them with ease. Edit your animation sequences. Save them on a disk to reload later. And build your moving figures right into your own BASIC programs. Flying dragons, spinning rockets, leaping athletes – you create them all.

You don't need to know machine language. And you won't have to do bit-mapping.

Get **pm ANIMATOR**, and write that arcade game you've been dreaming about. The power is yours.

Have your dealer demonstrate pm ANIMATOR.

Comes with numerous demo programs and tutorial on Player-Missile Graphics.
Requires disk drive, 32K RAM Atari computer.

SUGGESTED RETAIL PRICE: $44.95

To order by mail from TRONIX, add $2.00 shipping/handling to your check or money order. California residents add 6% sales tax (6-1/2% if you reside in L.A. County).

Dealer inquiries welcome.

TRONIX™
8295 South La Cienega Blvd., Inglewood, CA 90301
(213) 215-0529

RETROFIRE

By

Tom Hudson

16K Cassette 24K Disk

This issue, **A.N.A.L.O.G.** continues its series of public-domain assembly-language game programs with **Retrofire**, a game of skill for one player.

Typing the program.

Before typing anything into your computer, take a look at the listings accompanying this article.

Listing 1 is the main data and data checking routine. This listing is used to create both cassette and disk versions of Retrofire. The data statements are listed in hexadecimal (base 16), so the program will fit in 16K cassette systems. This makes typing rather difficult, but it's a necessary evil.

Listing 2 must be added to **Listing 1** if you are using a 410 or 1010 cassette recorder.

Listing 3 must be added to **Listing 1** if you are using a disk drive.

Listing 4 is the assembly-language source code for **Retrofire**, created with the ATARI Macro Assembler. You DO NOT have to type in this listing to play the game! It is provided for those readers interested in assembly language.

Follow the instructions below to make either a cassette or disk version of **Retrofire**.

Cassette instructions.

1. Type **Listing 1** into your computer and verify your typing with C:CHECK. (See page 58.)

2. After **Listing 1** has been entered into your computer, type in **Listing 2**. The program lines will merge with **Listing 1**. Make sure these lines were typed correctly. It's a good idea to CSAVE the entire program at this point.

3. Type RUN and press RETURN. The program will begin checking the DATA lines, printing the line number as it checks each one. It will alert you if it finds any problems. Fix any incorrect lines and re-RUN the program if necessary until all errors are eliminated.

4. When all data lines are correct, the program will ask you to "READY CASSETTE AND PRESS RETURN." Place a blank tape in your recorder, press RECORD and PLAY simultaneously and press RETURN. The message "WRITING FILE" will appear and the program will create a boot tape version of **Retrofire**, printing each DATA line number as it goes. When the READY prompt appears, you're ready to load and play the game. Make sure your BASIC program has been CSAVED before continuing.

5. From this point on, whenever you want to play **Retrofire**, do the following: Rewind the tape created by the BASIC program to the beginning. Turn your computer OFF and remove any cartridges. Press PLAY on your recorder, then turn your computer ON while holding down the START key. The computer will BEEP once. Press RETURN and **Retrofire** will load and run automatically.

Disk instructions.

1. Type **Listing 1** into your computer and check it with D:CHECK II (see p. 58) to eliminate any typing errors.

2. After **Listing 1** is correctly typed into your computer, type in **Listing 3**. The lines in this listing will merge with those in **Listing 1**. It's a good idea to SAVE the entire BASIC program at this point.

3. Type RUN and press RETURN. The program will begin verifying the DATA lines, printing the line number of each as it goes. It will alert you if it finds any problems. Fix incorrect lines and re-RUN the program if necessary until all errors are eliminated.

4. When all the date lines are correct, the program will ask you to "INSERT DISK WITH DOS, PRESS RETURN." Place a disk with DOS in drive 1 and press RETURN. The message "WRITING FILE" will appear and the program will create an AUTORUN.SYS file, printing each data line number as it goes. When the READY prompt appears, you're ready to play the game. Make sure your BASIC program has been SAVEd before continuing.

5. To play **Retrofire**, place the disk containing the AUTORUN.SYS file in drive 1. Turn the computer OFF, remove any cartridges and turn the computer ON. **Retrofire** will load and run automatically.

The game.

Your spacecraft was orbiting Jupiter when suddenly the radiation shielding failed. Forced to eject in an escape capsule, you must now land on Io, the only moon with a safe landing area. Your computers are able to take you to within 20,000 feet of the surface, where you must take the controls for the final approach and landing. Along with the obvious danger of crashing into the surface, you must avoid intense radiation from Jupiter and erupting volcanoes on Io's surface!

Retrofire requires one joystick in port 1.

When the game begins you will be shown a display of your escape pod's console, with the game credits. You have two factors that influence game difficulty, GRAVITY and FUEL.

The gravity level can be set to L (low), M (medium) or H (high). Low gravity is recommended for beginners. Select the desired gravity level by pressing the OPTION key.

The fuel level can range from 5,000 to 14,000 units. Select the desired fuel amount by pressing the SELECT key. Normal gravity/fuel amounts are: Low gravity, 7,000 units; Medium gravity, 10,000 units; and High gravity, 14,000 units.

There is one other option before beginning the game. The "C" key on the keyboard will toggle the playfield colors, enabling you to select the color combination you prefer.

Once you have selected the game difficulty, press START to begin the game.

The Retrofire display.

Figure 1 shows the **Retrofire** game display. This screen shows you all the information necessary to safely land your escape pod.

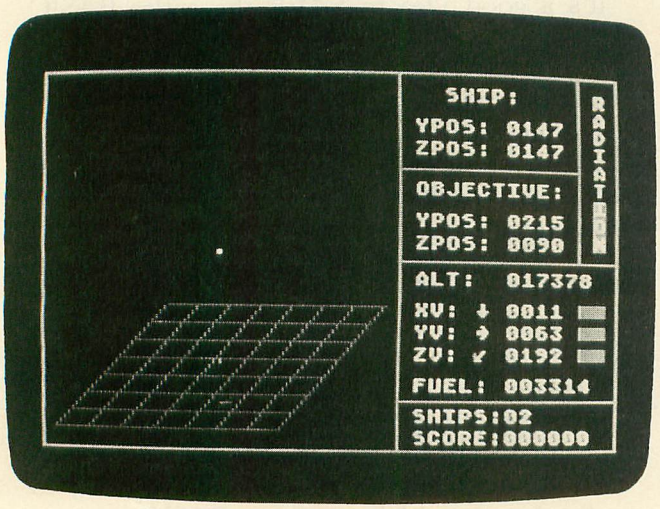


Figure 1.

Starting at the upper right side of the display, you will see a box with the word "RADIATION." This is a vertical bar graph indicating the amount of radiation you have accumulated on your descent. The higher your ship is in Io's thin atmosphere, the faster you accumulate radiation from Jupiter's radiation belt, so it's a good idea to get to a lower altitude as fast as possible.

Radiation can also be accumulated from Io's many volcanoes, described below.

If the radiation level ever gets too high, your ship's vital electronics will overload, and the craft will explode.

To the left of the radiation display is the navigational position readout. This display shows your ship's coordinates as well as the coordinates of the landing pad.

While the base's position is always known, at above 1,000 feet the navigational computer can only provide an approximation of your ship's position. Once you pass below 1,000 feet, the computer must use landmarks to show your exact position. When you match your coordinates to the base's coordinates, you are directly over the base and can land safely.

Below the navigational position displays are your five most important readouts.

The ALT reading shows your ship's altitude (in feet) in relation to the landing pad. Beware: Mountain heights are not taken into account here!

The next three readouts are what make **Retrofire** different from other "Lunar Lander" type programs: three dimensions!

These three readings are the X, Y and Z velocities of your ship in feet per second. To the left of each velocity reading is an arrow indicating the direction of movement. The axis labeling in non-standard, so read carefully.

The X velocity (XV) tells how fast your ship is going up or down. If the arrow is pointing up, you are ascending. If it is pointing down, you are descending. Pushing your joystick up will fire the main retrorockets, slowing your descent. Pushing the stick down will force you down toward Io's surface. The X velocity MUST be lower than 11 feet per second for a safe landing.

The Y velocity (YV) tells how fast your ship is going to the right or left over the terrain grid (described below). Once again, the arrow to the left of the velocity value indicates the direction of movement. Push your stick to the left to decrease the Y velocity and to the right to increase it. Your Y velocity MUST be lower than 6 feet per second for a safe landing.

The Z velocity (ZV) tells how fast your ship is going diagonally over the terrain grid, the third dimension in this game. You can think of this dimension as depth into your TV screen. Push your stick to the lower left to decrease this value and to the upper right to increase it. Your Z velocity MUST be lower than 6 feet per second for a safe landing.

Your ship has a "terminal velocity" of 500 feet per second. That is, your ship cannot go faster than 500 feet per second in any direction, no matter how much you try.

To the right of each velocity indicator is a color-coded engine temperature light. When green, the engine temperature is OK, and the engine is working normally. As you fire each engine, it heats up. When the engine is not being fired, it will cool down. Wise use of the engines will keep the engines cool and safe.

If, however, you fire an engine for too long, it will begin to overheat. As the engine heats up, the temperature light will go from green to yellow to red. As the engine heats up to the yellow and red zones, it will begin to fail, losing efficiency. If the engine is forced to operate in the red zone too long, the indicator will turn dark gray, indicating engine burnout. If this happens, the engine is dead and you will probably crash. Whenever the temperature status of an engine changes, you will be alerted with a short tone.

Below the velocity indicators is the ship's fuel level. As you fire the engines, fuel is subtracted from your initial supply. When your fuel supply drops below 1000 units you will receive a warning message and a tone. If the fuel level reaches zero, an "OUT OF FUEL" message is displayed with a lower pitch tone.

Directly below the fuel indicator is a display showing the number of ships you have left.

The game score is shown below the number of ships left. Each time you land safely, you are awarded ten points for each unit of fuel remaining. This

feature is primarily for competition between two players.

The largest area of the display is the graphic terrain display on the left side of the screen. This display shows the terrain directly below your ship. Your ship is displayed as a white square over the terrain grid, with a gray "shadow" on the grid to indicate your exact position and the terrain height below your ship.

Your objective is to land in the grid square which contains your landing pad, indicated by a flashing red "+."

As your ship begins its descent, the terrain will appear very flat. This is because surface details cannot be seen at this altitude. As your ship descends below 800 feet, the computer will "zoom in" on the square your ship is over, enlarging it to a new 7 x 7 grid. At this point, terrain detail will begin to show up. You will notice that your spacecraft appears to move faster at this magnification, due to the "zoom" effect.

As your ship descends below 1000 feet, once again the computer will "zoom in" on the square below your ship, enlarging it to a new 7 x 7 grid. This is the final stage of your approach toward the surface. At this magnification level you must pay close attention to the terrain and your shadow. Your shadow indicates how close you are to the ground, and if your ship hits any peaks, you will be destroyed. A good rule of thumb is to cruise over the surface at an altitude of around 500 feet.

While you are below 1000 feet, you must watch out for volcanoes. These erupting mountains constantly spew radioactive debris. If your ship passes directly over a volcano at an altitude of less than 500 feet, you will be exposed to a potentially lethal dose of radiation. It's a good idea to keep an eye on your radiation indicator at this point. The number of volcanoes increases each time you land successfully.

At the top of the graphic display window is a warning message area. This area will display fuel warning and other messages as necessary.

If you need to pause the game during descent, simply press the space bar. The game can be resumed by pressing the space bar a second time.

Some notes on playing Retrofire.

Those who are expecting **Retrofire** to be a fast-action shoot-em-up are in for a surprise. This is a fairly realistic *simulation* program with game elements, *not* something you'd expect to find in an arcade.

Retrofire can be quite difficult on the higher gravity settings but with practice it can be mastered.

Probably the toughest thing about **Retrofire** is getting accustomed to the X-Y-Z axis system and the joystick control.

Figure 2 shows the relationship of the joystick to movement on the screen. It's actually very simple, but takes some practice to activate the Z-axis engines, which are diagonals on the joystick.

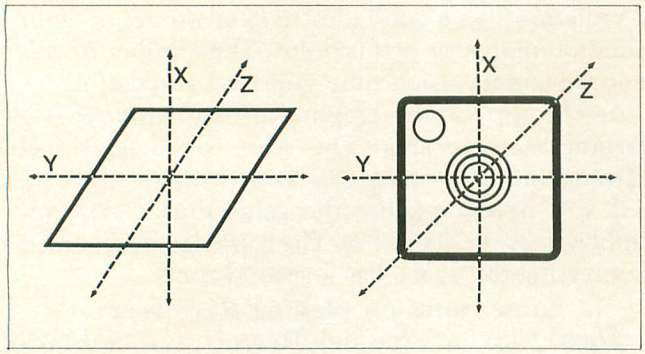


Figure 2.

The other concept which can be confusing to beginning players is the "zoom" function. This occurs at altitudes of 8000 and 1000 feet.

Figure 3 shows how this magnification works. Although the highest grid (altitude 8000 feet) is made up of 49 squares (7 x 7 squares), each of these squares, when enlarged at altitudes less than 8000 feet, can be shown to contain another 49 smaller squares. When the altitude drops to less than 1000 feet, the square the ship is over enlarges further still, showing that each of these squares also contains 49 smaller squares.

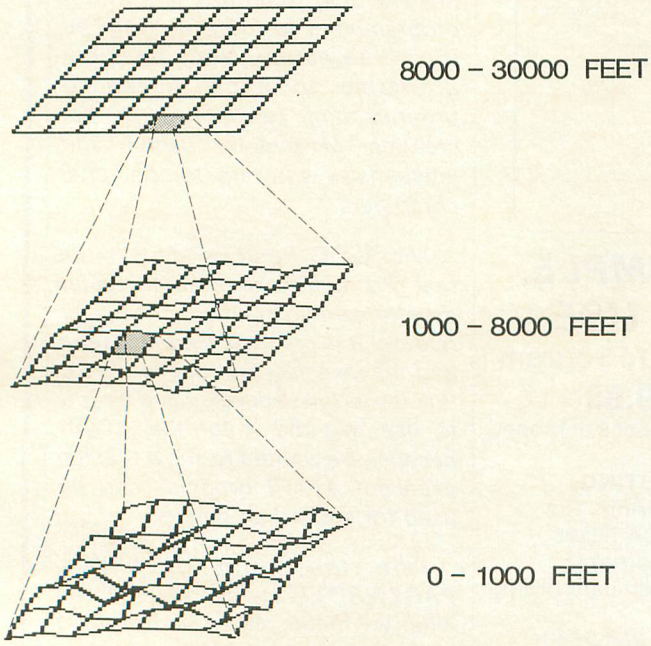


Figure 3.

By doing some quick math, you can see that the highest 7 x 7 grid actually represents 343 x 343 squares, or a total surface of 117,649 squares!

With over 117,000 possible places to land, it's obvious that the best strategy is to always keep yourself positioned over the grid square with the flashing base symbol. If you miss the square with the base symbol, you can always fly across to the base by using the navigational coordinates at the top of the screen. This requires more concentration than using the grid, so be forewarned!

Once again, this game is somewhat tricky, so just stick with it, and soon you'll be qualified to land almost any spacecraft. □

Listing 1.

```
1 REM *** RETROFIRE ***
10 DATA 0,1,2,3,4,5,6,7,8,9,0,0,0,0,0,
0,0,10,11,12,13,14,15
20 DIM DAT$(91),HEX(22):FOR X=0 TO 22:
READ N:HEX(X)=N:NEXT X:LINE=990:RESTOR
E 1000:TRAP 60:? "CHECKING DATA"
25 LINE=LINE+10:? "LINE:";LINE:READ DA
T$:IF LEN(DAT$)<>90 THEN 110
28 DATLIN=PEEK(183)+PEEK(184)*256:IF D
ATLIN<>LINE THEN ? "LINE ";LINE;" MISS
ING!":END
30 FOR X=1 TO 89 STEP 2:D1=ASC(DAT$(X,
X))-48:D2=ASC(DAT$(X+1,X+1))-48:BYTE=H
EX(D1)*16+HEX(D2)
35 IF PASS=2 THEN PUT #1,BYTE:NEXT X:R
EAD CHKSUM:GOTO 25
40 TOTAL=TOTAL+BYTE:IF TOTAL>999 THEN
TOTAL=TOTAL-1000
45 NEXT X:READ CHKSUM:IF TOTAL=CHKSUM
THEN 25
50 GOTO 110
60 IF PEEK(195)<>6 THEN 110
100 ? "WRITING FILE":PASS=2:LINE=990:R
ESTORE 1000:TRAP 60:GOTO 25
110 ? "BAD DATA: LINE ";LINE:END
1000 DATA 488A48A6C5BD201C8D0AD48D15D0
E6C568AA6840A9288590A900858FA90885B3A9
0085B2A000B18F91B2C8D0F9,856
1010 DATA E690E6B3A590C940D0EF4C3A08A2
BEA90F9D001FCAD0FAA9708D001F8D011F8D02
1FA94F8D031F8D6B1FA9208D,30
1020 DATA 051FA9108D041FA9308D6D1FA900
8D6C1FA9418DBC1FA91F8DBE1FA9008DBD1FA9
8F8D1F1F8D391F8D761F8D80,363
1030 DATA 1F8D8A1FA9018DCE1AD82065E420
0413A900A27F9580CA10FB8D2F028D0ED48D08
D28DC6028DC80220EE12A9FF,395
1040 DATA A2059D91079D9B079DA507CA10F4
A97385DF206511A210BDD61A95EB9D251CCA10
F5A90A8DC502A90F8DC002A9,654
1050 DATA 048DC102A9C08D03D0A9088D0102
A9008D0002A9008D3002A91F8D3102A9008D07
D4A215A0A1A907205CE4A93E,884
1060 DATA 8D2F02A9038D1DD0A9C08D0ED4A9
028598A20086BFBDC41848BDD31848BDE21848
BDB518F00F68858368858168,270
1070 DATA 85852087174C4F09688582688580
6885842018176A6BFE8E00FD0CBA24986A4BDF1
1885A1BD3B1985A2BD851985,638
1080 DATA A3208E13A6A4CA10E7AD0AD2291F
18692885B6AD0AD2291F18692885B8A900A205
9D821CCA10FA8D361C8D371C,239
1090 DATA 8D381C8D391CA9C48D221C8D231C
8D241CA97985E6A90F8D7907A900A22F9D0007
E8E07A90F820251220321220,504
1100 DATA 4512205B12207112207E12208B12
A90085AE2044132008A14201C13201F14ADCE1A
F006CECE1A4CD40BA5E8D0FC,324
1110 DATA A5AAF0034C570FA90185AA854DA5
EBC9039004A90085E7A5E7D025A5E5D009A6AE
BDF91BF01A85E7A5E6C930B0,555
```

```
1120 DATA 034CBD0A38E90185E6AAA90F9D00
07A90085E5F8AD361CF0064CB00C4CC90CA5AD
38E5EF85ADA5EDE5EE85EDA5,409
1130 DATA ECE90085ECA5EBE90085EBC99090
E0D8A5EE05F105F3D05CA5EFC911B056A5F2C9
06B050A5F4C906B04AA5CCC5,560
1140 DATA DCD044A201A00E206813A9FF85AA
A5AAD0FCA20306F726F626F5CA10F7F8A20218
B5F575F995F9CA10F7D8208B,998
1150 DATA 12A9F085AAA5AAD0FCA5DFC9FFF0
0818690A85DF2065114CA00CA5E6C92FD004A2
05D002A202A00E206813A5BE,588
1160 DATA 38E922A2139DB31CCA10FAA5BA38
E9300AA2139D9F1CCA10FAA90085BA85BEA213
AD0AD209109DC71CAD0AD209,811
1170 DATA 109DDB1CA9009DEF1C9D031DAD0A
D20910293F9D8B1CCA10DB3004A5AAD0FCA901
85AAA2138E171DBD8B1CF036,641
1180 DATA 8D9F1C8580BDB31C858220311 7A0
00BDB118518F918FAE171DDE8B1CF018BDC71C
187DEF1C9DEF1C08E00AB00C,668
1190 DATA 28BD9F1CE9004C690B4C9A0B28BD
9F1C69009D9F1CC9C0B033C90A902F8580BDDB
1C187D031D9D031DBDB31CE9,333
1200 DATA 009DB31CC90A901785822031 7A0
00BDB118518F918FAE171DCA300D4C220BAE17
1DA9009D8B1CF0EDA213A900,660
1210 DATA 1D8B1CCA10FAC900F0034C180BA9
FF85AAA5AAD0FCC6F830034C9D0CA200A00E20
6813ADFC02C912D014AEAA18,876
1220 DATA ADAB188DAA188EAB18A9FF8DFC02
4C3A08A208A01E206813A209A02D206813A20A
A037206813AECF1ABDD31A8D,439
1230 DATA 831AA207A04B206813ADCC1A0910
8D6A1AADCD1A09108D6B1AA206A055206813A5
AAD0FCAD1FD0C907F09EC906,147
1240 DATA F849C903D012ADCF1A186901C903
D002A9008DCF1A4C7C0CADCD1A1869018DCD1A
C90AD00EA9008DCD1AADCC1A,878
1250 DATA 1869018DCC1AADCC1AF00CADCD1A
C905D005A9008DCC1AA91E85AA4CD40BAD1FD0
C907D0F9ADCC1A8DE01AADCD,447
1260 DATA 1A0A0A0A0A8DE11A4C9008207E12
A20CBDD61A95EB9D251CCA10F54C7109A5AD18
65EF85ADA5ED65EE85EDA5EC,699
1270 DATA 690085ECA5EB690085EBD8202512
201610A200A5EBD00CA202A5ECDDA6189003CA
D0F6E4AEF052B02986AEA901,399
1280 DATA 85C1A5B620DB0FB5D60A0A0A0A18
65B485B6A5B820DB0FB5D90A0A0A0A1865B485
B84C2F0D86AEA90185C1A5B6,355
1290 DATA 20C40F85B698A6AE95D5A5B820C4
0F85B898A6AE95D8201C13200A14201F14AD28
1CC905F02BF8AD2A1C18AECF,383
1300 DATA 1A7DD01A8D2A1C85F0AD291C6900
8D291C85EFAD281C69008D281C85EEA203A000
20CB11203212AE780286ABBD,674
1310 DATA 1D1BD0034C3B0EA5F505F605F7D0
1FA5EA2901D016A5EA090185EAA9FF8D04D2A9
1485E9A204A00E2068134CD9,811
1320 DATA 0DA5F5D022A5F6C910B01CA5EA29
02D016A5EA090285EAA9508D04D2A91485E9A2
03A00E206813A4ABBE1D1BBC,179
1330 DATA 871CB9F51BF005CD0AD2B0034C3B
0EA90A85C3E6B1A5B12901D08FF8A5F738E902
85F7A5F6E90085F6A5F5E900,443
1340 DATA 85F5D8207112A6ABBD5D1B85A8BD
4D1B85A9BC2D1BC999F00AB9251CC905F0214C
280EB9251CC995D007B9261C,534
1350 DATA C900F010BC3D1BBD2D1BAA20B511
204512205B12A00620E90FA6AEBCAA1B469066
8F88D0F9AD371CD00EA5B718,261
1360 DATA 658F85B79010E6B64C6A0EA5B738
E58F85B7B002C6B6A90085C085C2A5B6C9B4B0
26C970904338E97085B6A6AE,582
1370 DATA F014B5D5186901C907D009A90095
D5CAD0F0300295D5E6C04CBD0E18697085B6A6
AEF012B5D538E9011009A906,123
1380 DATA 95D5CAD0F2300295D5E6C0A90185
C2A6AEA5B64A4A4A4A95D6A900E8E003F00495
D6D0F7A00220E911A00820E9,381
1390 DATA 0FA6AEBCAA1B4690668F88D0F9AD
381CD00EA5B918658F85B99010E6B84C060FA5
B938E58F85B9B002C6B8A5B8,264
1400 DATA C9B4B02AC970904738E97085B8A6
AEF014B5D8186901C907D009A90095D8CAD0F0
300295D8E6C0A90285C24C57,343
1410 DATA 0F18697085B8A6AEF012B5D838E9
011009A90695D8CAD0F2300295D8E6C0A90385
C2A6AEA5B84A4A4A4A95D9A9,274
1420 DATA 00E8E003F00495D9D0F7A00320E9
11A5C0F03B201C13A6C2BCAD1BA9078591BEB1
1BBD3A1CA6919D7A1C88C691,946
1430 DATA 10F0200A14A6C2BCAD1BA9078591
A691BD7A1CBED11B9D3A1C88C69110F0201F14
AD1FD0C907D0034CEB09AD1F,958
1440 DATA D0C907D0F94C3A08A000C9109007
38E910C84CC60F85910A1865910A6591608 5B4
A90785B5A90020A411A6AE60,826
1450 DATA B9EC00290F858FB9EC004A4A4A4A
AAA58F187D6D1B858FB9EB00290FAAA58F187D
8118858FA9007D7718859060,683
1460 DATA EAA5ED290F85B2A90085B3A5ED4A
4A4A4AAAA5B2187D6D1B85B2A5EC290FAAA5B2
187D811B85B2A5B37D7718 85,308
1470 DATA B3A5EC4A4A4A4AAAA5B2187D951B
85B2A5B37D8B1B85B3A5EB290FAAA5B2187DA3
1B85B2A5B37D9F1B85B346B3,115
1480 DATA 66B246B366B2A6AEBDA71B85B5A5
B285B4A5B320A411A5B84A4A85BFA5B64A1865
BF18693885BAA5B84A85BFA9,152
1490 DATA C938E5BF85BF38E5B485BEA5B64A
48290785CAA90738E5CA85CA6829F885CCA5B8
4A48290785CBA90738E5CB85,221
1500 DATA CB684A4A4A1865CC85CCC5E2D016
A5E3F012A5EBD00EA5ECC905B008A90185E5A9
0085E7A6CCBD3A1C85CDBD3B,428
1510 DATA 1C85CEA5CB85C720341185CFA6CC
BD421C85CDBD431C85CEA5CB85C720341185CE
A5CF85CDA5CA85C720341185,333
1520 DATA CFA5BF38E5CF85BCC5BEB0034CBD
0A60A5CDC5CE901138E5CE85C6208A11A5C84A
4A4A1865CE60A5CE38E5CD85,597
1530 DATA C6A90838E5C785C7208A11A5C84A
4A4A1865CD60A205AD0AD22907C907F0F795D0
CA10F2A5D20A0A0A1865D585,984
1540 DATA DCA00020E911A00120E91160A900
85C9A2080A26C906C690071865C79002E6C9CA
D0EF85C860A20806B42AC5B5,217
1550 DATA 9004E5B5E6B4CAD0F260F8BD261C
1865A89D261C95ECBD251C65A99D251C95EBC9
50B007A90099361CD860A900,816
1560 DATA 38FD261C95ECA900FD251C95EBA9
01D0E8B9031C85A2BE071CB4D0B90B1C85FCB9
121C85FDB4D1B9191CF81865,407
1570 DATA FD85FDA5FC690085FCB5D21865FD
85FDA5FC690085FCD8A90285A6A2114C9512A9
6285A2A90385A6A2004C9512,561
1580 DATA AD361C20C612A97085A2A90285A6
A2034C9512A902186D371C20C612A97A85A2A9
0285A6A2064C9512A9041 86D,926
1590 DATA 381C20C612A98485A2A90285A6A2
084C9512A99285A2A90385A6A20A4C9512A9A0
85A2A90185A6A20D4C9512A9,913
1600 DATA AA85A2A90385A6A20EA91F85A186
A5A6A5B5EB20A912E6A5C6A6D0F36085A74A4A
4A4A091085A3208E13E6A1A5,756
1610 DATA A7290F091085A3208E13E6A16048
4AAABD171B859EBD1A1B859FA000680A0A0AAA
A90785A6BDE71A919EE89818,456
1620 DATA 6928A8C6A6D0F160A900A8990003
9900049900059900069900078 8D0EE60A21DA9
208590A900858FA8918F88D0,397
1630 DATA FBCAD00160E690D0F2A90085BAA9
2F8590A9B2858FA248A015A900918F8810FBCA
D00160A58F186928858F90EA,496
1640 DATA E690D0E6A92285 90A942858FA208
A015A900918F8810FBCAD00160A58F18692885
8F90EAE690D0E684A2BCCF19,674
1650 DATA A90285A1A21686AF84B0A6AFA4B0
B9DA1985A3208E13C6AFD00160E6A1E6B0D0E9
A5A2858FA9008590068F2690,834
1660 DATA 068F2690068F2690A58F18691085
9EA5906920859F068F2690068F2690A58F1865
9E859EA590659F859FA59E18,730
1670 DATA 65A1859E9002E69FA5A3858FA900
859085A0068F2690068F2690068F2690A59018
69E08590A4A0B18FA000919E,200
1680 DATA E6A0A5A0C908F00DA59E18692885
9E90E7E69FD0E360A23FA4AEA D0AD2290FD9A3
18B0F69D3A1CCA10F060D8A9,475
1690 DATA 0085E3A2008A9D0006CAD0FAA6AE
F013A200B5D6D5D0D053B5D9D5D3D04DE8E4AE
D0EFA6AEB5D30A0A85BFB5D0,479
```

Tired of high prices, poor service and hidden charges you get from other mail order companies . . .

Try RISING SUN SOFTWARE!

ENTERTAINMENT

ARTWORX
Strip Poker (D) $25.00
S.P. Data Disks 20.00

ATARI (ROM only)
Centipede $32.00
Defender 32.00
Dig Dug 32.00
Donkey Kong 35.00
Eastern Front 35.00
Galaxian 32.00
Joust 35.00
Missle Command 27.00
Ms. Pac-Man 35.00
Pac-Man 32.00
Pengo 32.00
Qix 32.00
Robotron 32.00
Space Invaders 27.00
Star Raiders 32.00

BIG FIVE
Miner 2049er (R) $35.00

BRODERBUND
A.E. (D) $25.00
Arcade Machine (D) ... 42.00
Choplifter (D) 25.00
Choplifter (R) 32.00
Oper. Whirlwind (D) ... 25.00
Sea Fox (D) 21.00
Sea Fox (R) 28.00
Sky Blazer (D) 28.00

BUDGECO
Raster Blaster (D) $21.00

DATAMOST
Airstrike (D/C) $28.00
Bilestoad (D) 28.00
Mating Zone (D) 25.00

DATAMOST
Micropainter (D) $25.00
Pooyan (D/C) 21.00
Zaxxon (D/C) 28.00

DON'T ASK
Poker Sam (D/C) $20.00
S.A.M. (D) 42.00

EDU-WARE
Prisoner 2 (D) $28.00
Rendezvous (D) 28.00

ELECTRONIC ARTS
Archon (D) $28.00
Hard Hat Mack (D) 25.00
M.U.L.E. (D) 28.00
Murder on the Zunderneuf (D) 28.00
Pinball Construction Set (D) 28.00
Worms? (D) 25.00

EPYX
Jumpman (D) $28.00
Jumpman Jr. (D) 28.00

INFOCOM (disk only)
Deadline $35.00
Enchanter 35.00
Planet Fall 35.00
Starcross 28.00
Suspended 35.00
The Witness 35.00
Zork I 28.00
Zork II 28.00
Zork III 28.00

INTELLIGENT STATEMENTS
Pro. Blackjack (D) ... $49.00

LIGHTNING SOFTWARE
Master Type (D) $28.00

MUSE
Castle Wolfenstein ... $21.00

ODESTA
Chess (D) $49.00
Checkers (D) 35.00
Odin (D) 35.00

PARKER BROTHERS
Astrochase (R) $35.00
Chess (R) 42.00
Frogger (R) 34.00
Popeye (R) 35.00
Q-Bert (R) 35.00
Risk (R) 42.00
Super Cobra (R) 35.00
Tutankam (R) 35.00

ROKLAN
Deluxe Invaders (R) .. $28.00
Gorf (D) 28.00
Gorf (R) 32.00
Wizard of Wor (R) 32.00
Wizard of Wor (D) 28.00

SIERRA ON-LINE
Crossfire (R) $25.00
Crossfire (D/C) 21.00
Frogger (D/C) 25.00
Mission Asteroid (D) .. 20.00
Sammy Lightfoot (R) .. 27.00
Ultima II (D) 42.00
Ulysses (D) 28.00
Wiz & Princess (D) 25.00

HARDWARE DEALS

MODEMS
Apple Cat II .. $299.00
Micromodem II 265.00
Micromodem w/terminal prog. 295.00
212 Apple Cat . 580.00

MONITORS
Amdek
Color I $299.00
Color II RGB ... 599.00
Color III RGB .. 399.00
RGB Card 149.00

USI
Pi1 9" Green . $119.00
Pi2 12" Green . 149.00
Pi3 12" Amber. 159.00
Pi4 9" Amber .. 129.00
Color 1400 .... 299.00

PRINTERS
C. Itoh
GX-100 ..... $229.00
Prowriter 399.00
Okidata
Microline 80 ... 349.00
Microline 82A . 449.00
Microline 92 ... 549.00

STRATEGIC SIMULATIONS
Battle for Normandie (D/C) .. $28.00
Battle of Shiloh (D/C) 28.00
Combat Leader (D) 28.00
Cosmic Balance (D/C) 28.00
Cosmic Balance II (D/C) 28.00
Cytron Masters (D) 28.00
Galactic Gladiator (D) . 28.00
Knights of the Desert (D/C0 ..... 28.00
Shattered Alliance (D) . 28.00
Tigers in the Snow (D/C) 28.00

SYNAPSE
Blue Max (D/C) $25.00
Dimension X (D/C) 25.00
Ft. Apocalypse (D/C) .. 25.00
Necromancer (D/C) ... 25.00
Pharoah's Curse (D/C) 25.00
Shadow World (D/C) .. 25.00
Shamus (D/C) 25.00
Shamus (R) 32.00
Shamus II (D/C) 25.00
Survivor (D/C) 25.00
Zepplin (D/C) 25.00

THORN EMI
Hockey (R) $28.00
Jumbo Jet Pilot (R) 35.00
River Rescue (R) 28.00
Soccer (R) 35.00
Submarine Commander (R) 35.00

UTILITIES & LANGUAGES

ADVENTURE INTERNATIONAL
Diskey (D) $35.00

ATARI
Assembler Editor (R) . $45.00
Atari Basic (R) 42.00
Macro Assembler (R) .. 68.00
Microsoft Basic II (R) .. 68.00
PILOT (R) 60.00

DATASOFT
BASIC Compiler (D) .. $63.00
Lisp Interpreter (D) ... 70.00
Edit 6502 (R) 125.00

OPTIMIZED SYSTEMS
Basic A+ (D) $56.00
Bug 65 (D) 25.00
Mac/65 (D) 56.00

EDUCATION

ATARI
Conversational Languages
French (C) $42.00
German (C) 42.00
Italian (C) 42.00
Spanish (C) 42.00
Educator Kit 117.00
Inv. to Programming 1 ..... 18.00
Inv. to Programming 2 ..... 23.00
Inv. to Programming 3 ..... 23.00
Juggles House (D) 23.00
Juggles House (C) 18.00
Juggles Rainbow (D) .. 23.00
Juggles Rainbow (C) .. 23.00
My First Alphabet (D) . 27.00
Programmer Kit 53.00
Tough Typing (cass) .. 18.00

EDU-WARE
Compu-Read (D) $21.00
Compu-Read (C) 15.00

SPINNAKER
Face Maker (D) $28.00
Hey Diddle Diddle (D) . 21.00
Kindercomp (D) 21.00
Most Amazing Thing (D) 28.00
Rhymes & Riddles (D) . 21.00
Snooper Troops 1 (D) . 32.00
Snooper Troops 2 (D) . 32.00

RISING SUN SOFTWARE
4200 PARK BLVD.
OAKLAND, CALIFORNIA 94602
(415) 482-3391

Ordering Information: We'll accept any form of payment—cash, personal check, money order, VISA/MasterCard, or C.O.D. Send cash at your own risk. Add $2.00 for UPS shipping; $3.00 for Blue Label Air. California residents add applicable sales tax. ALL orders shipped same day received. If we are out of stock on a particular item we will include a special bonus with your order when shipped.

master charge

CALL TOLL FREE 24 HOURS (ORDERS ONLY)
(800) 227-1617, ext. 114 (Outside California)
(800) 772-3545, ext. 114 (Inside California)

VISA

```
1700 DATA 0A0A0A1865BF18693D8D02D0B5D3
0A0A0A85BFA9C438E5BFA8A204BDFC1B990006
C8CA10F6A6AEE002D013A6DC,757
1710 DATA A9009D3A1C9D3B1C9D421C9D431C
4CE514A5AEC902D053AD0AD2C5DFB04CAD0AD2
2907C905B0F718690185E1AD,870
1720 DATA 0AD22907C905B0F718690185E00A
0A0A1865E185E2A5E10A0A85BFA5E00A0A0A18
65BF18693B85E3A5E10A0A0A,430
1730 DATA 85BFA9BA38E5BF85E4A6E2A9074C
7D14A901859BA9108599A900858485859C85
9AA900859BA8A5991865 9A99,312
1740 DATA 8000A9AA38E59B38A69CFD3A1C99
8200A59BD0062018174C1F1520871 7A5991869
088599E69CA001A59B186908,895
1750 DATA 859BC93990C7A9108599A59A1869
10859AC97190B3A900859BA9FF859DE69DA59D
859CA900859AA8A599186 59A,33
1760 DATA 998000A9AA38E59BA69C38FD3A1C
998200A59AD0062018174C7A1520871 7A59C18
6908859CA001A59A18691085,619
1770 DATA 9AC97190C9A5991869088599A59B
186908859BC93990AC60D8A90085C5ADFC02C9
21D00BA5E849FF85E8A9FF8D,838
1780 DATA FC02A5E8F0034C62E4ADC3021869
01290709308DC302A5AAF002C6AAA5E7F002C6
E7A5E3F0298D02D0AD0AD22D,551
1790 DATA 0AD2A4E499000699040 6AD0AD22D
0AD2990206AD0AD2293C990606A93A8DC202D0
15E6DDA5DD290FD00DE6DEA5,487
1800 DATA DE2901AABD011C8DC202A6BBA9FC
3D00039D0003A9FC3D01039D0103A6BDA9F33D
00039D0003A9F33D01039D01,747
1810 DATA 03A5BA8D04D08D05D0A6BE86BBA9
021D00039D0003A9021D01039D0103A6BC86BD
A9081D00039D0003A9081D01,466
1820 DATA 039D0103A9828D00D2A9088D02D2
A5C38D01D24A4A4A8D03D2E6C4A5C42907D006
A5C3F002C6C3AE7802BD1D1B,683
1830 DATA 85ACF01DAACABD851CC918B014BD
821C1869019D821CBD851C69009D851C20FB16
A202C6ACE4ACF026BD851CC9,33
1840 DATA 18B01FC900D007BD821CC9029014
BD821C38E9029D821CBD851CE9009D851C20FB
16CA10D3A5E9F00B38E90185,54
1850 DATA E94A09A08D05D24C62E4291F4A4A
4A9D881CA8B9F11BDD221CF00B9D221CA91485
E90A8D04D26020311749 88D,902
1860 DATA B11839A9188591BDAD18A000318F
0591918F60A5820A858FA9002A8590068F2690
068FA58F85922690A5908593,810
1870 DATA 068F2690068F2690A58F18659285
8FA59065938590A91018658F858FA920659085
90A58029074AAAA5844AA580,862
1880 DATA 6A4A4A18658F858FA59069008590
60A583C582900B38E582858CA901858ED00BA5
8238E583858CA9FF858EA585,558
1890 DATA C584901BD006A581C5809013A581
38E580858AA585E584858BA901858DD011A580
38E581858AA584E585858BA9,710
1900 DATA FF858DA900858885868587858 9A5
8BD006A58AC58C9017A58B8597859 5A58A8596
8594A5954AA5946A85884C0F,711
1910 DATA 18A900859785 95A58C8596859 44A
8586A5960597D00160A58818658C8588A58969
008589C595901CD006A588C5,198
1920 DATA 949014A58838E5948588A589E595
8589A58218658E8582A5861865 8A8586A58765
8B8587C595903AD006A586C5,183
1930 DATA 949032A58638E5948586A587E595
8587A58D30141002D0A5A5801865 8D8580A584
690085844C8E18A58038E901,547
1940 DATA 8580A584E900858420181 7A59638
E9018596A597E900859705 96D0CD6001040800
80100055AAFF3FCFF3FCC030,739
1950 DATA 0C0300010101010001000100100
0100010001010000000000010100000101010A
36360A0AC0C0C03636C0C020,8
1960 DATA 20200A00AB4B40A0AB45E5E9D9D35
350B5D1B1C1D1E1F191A1B1C1D191A1B1C1D19
1A1B1C1D1E1F202122191A1B,152
1970 DATA 1C1D191A1B1C1D191A1B1C191A1B
191A1B191A1B191A1B1C1D191A1B1C1D1E191A
1B1C1D1E2525252525252525,445
1980 DATA 2510101010101E1E1E1E1E282828
28283A3A3A3A3A3A3A3A3A3A484848484852 52
5252526262626270707 07A7A,234
1990 DATA 7A8484849292929292A0A0A0A0A0
A0AAAAAAAAAAAA1119212931394149513328 29
301A39302F331A3A302F331A,568
2000 DATA 2F222A2523342936251A39302F33
1A3A302F331A212C341A3836 1A39361A3A361A
2635252C1A3328293033 1A33,504
2010 DATA 232F32251A3221242921342 92F2E
00162C42586E849AB8C6DC0000000000002721
2D25002F3625320000000000,630
2020 DATA 0000000033352323253333263 52C
002C212E24292E270000000000262134212C00
3328293000233221332800 00,952
2030 DATA 00002635252C002C2536252C0023
3229342923212C0000000000000002F3534002F
26002635252C000000000000,80
2040 DATA 0000322124292134292F2E002F36
25322C2F212400000000000000002635252C1A
0000001010100000000000,53
2050 DATA 0000000000273221362934391A00
0000000000000000000000003225344322F26
2932250100000000000000,808
2060 DATA 000022791A00346F6D0028756473
6F6E0000000000000212E212C2F2700236F6D
707574696E67000000000500,19
2070 DATA 000A18302C2D2802000000000000
0000000100000200000000181818 7E3C180000
183C7E1818180000180C7E7E,87
2080 DATA 0C180000183 07E7E301800003C1C
3464C0000000000C98B8E0F000AD3DCD313334
0000000000000302000300 02,356
2090 DATA 0001010000000000000080600 08
0006000303000000000000000201000200010 0
0000000000000000000000000,398
2100 DATA 9900990000990000000000000001
0100990099000199000000A141E28323C46505A
0000000101010202030300 64,882
```

(continued on page 79)

Unbelievable, Great Savings!

600XL . . . CALL
800XL . . . CALL
1400XL . . CALL
1450XLD . CALL

Atari Inc.

1010 Recorder	$ 72
1050 Disk Drive	$339
1027 Printer	Call
1025 Printer	$409
830 Modem	$145
850 Interface	$159
Entertainer System	$ 64
482 Educator	$110
483 Programmer	$ 52
Communicator II	$209
Atari Accounting	$169
CX4104 Mailing List	$ 19
CXL 4007 Music Composer	$ 42
Programming 2 & 3	Ea. $ 22
Conversational Languages	$ 42
CX4018 Pilot	$ 55
CX405 Pilot	$ 92
CXL4003 Assembler Editor	$ 42
CXB126 Microsoft Basic II	$ 62
CXL4022 Pac-Man	$ 30
CXL4020 Centipede	$ 30
CXL4006 Super Breakout	$ 26
CXL4008 Space Invaders	$ 26
CXL4009 Computer Chess	$ 26
CXL4011 Star Raiders	$ 30
CXL4012 Missile Command	$ 26
CXL4013 Asteroids	$ 26
The Bookkeeper	$102
Home Filing Manager	$ 65
Atari Speed Reading	$ 54
Home Manager Kit	$ 55
Family Finance	$ 36
Time Wise	$ 23
Galaxian	$ 30
Defender	$ 30
Paint	$ 30
Qix	$ 30
Dig Dug	$ 30
ET Phone Home	$ 34
Atari Writer	$ 65
Donkey Kong	$ 36
Donkey Kong Jr.	$ 38
Atari Logo	$ 72
Mickey in the Great Outdoors/D	$ 34
My First Alphabet/D	$ 25
Peter Pan's Daring Escape	$ 34

Business & Utilities

Visicalc	$169
Computari's Financial Wizard	$ 45
Color Accountant	$ 65
Mail Merge	$ 20
Data Perfect	$ 95
Letter Perfect	$105
Bank Street Writer	$ 45
Text Wizard	$ 55
File Manager 800 +	$ 65
Datasm 652.0	$ 59
Syn Assembler	$ 34
K-Dos	$ 59
Lisp Interpreter	$ 79
Basic Compiler	$ 55
Datalink	$ 27
Atari World	$ 39
Color Print	$ 27
Graphics Generator	$ 17
Micropainter	$ 23
Graphics Master	$ 17
P.M.P. Property Management	$149

Modems

Hayes Smartmodem 300 Baud	$209
Hayes Smarmodem 1200	$499
Signalman Modem II	$ 79
Atari 830 Modem	$155

Educational

Pre-School

Sammy the Sea Serpent (C)	$ 13 (D) $ 19
Pre-School I.Q. Builder	(C) $ 13 (D) $ 24
My First Alphabet (D)	$ 26
Alien Counter/Face Flash (D, C)	$ 26
Humpty Dumpty/Jack & Jill (C)	$ 25
Facemaker (D)	$ 23
Juggles Rainbow (D, C)	$ 22
Juggles House	$ 22
Alphabet Zoo (CT)	$ 20

Music

Music Composer (CT)	$ 42
Jerry White Music Lessons (C)	$ 20
Magic Melody Box	$ 14

Social Studies and Geography

Globemaster (D)	$ 27
States and Capitals (C)	$ 12
European Countries and Capitals (C)	$ 12

Math

Monkey Up A Tree(D,C)	$19
Video Math Flash Cards(D,C)	$13
Algicalc(D,C)	$19
Compumath-Fractions	(C)$23(D)$29
Compumath-Decimals	(C)$23(D)$29
Golf Classic/Compubar (Angles)	(D,C)$26
Cash Register	(C)$13(D)$19
Big Math Attack	(C)$17(D)$22
Computation Concentration	(C) $13 (D) $15

Reading and Language Arts

Wordmaker (D, C)	$ 19
Spelling Genie (D, C)	$ 19
Compuread	(C) $ 17 (D) $ 23
Memory Builder/ Concentration	(C) $ 13 (D) $ 19
Let's Spell (C)	$ 13
Do-It-Yourself Spelling (C)	$ 16
S.A.T. College Board Prep (C)	$ 89
Vocabulary Builder I	(C) $ 13 (D) $ 19
Fishing for Homonyms (C)	$ 13
Hey Diddle Diddle (D)	$ 20
Snooper Troops 1 & 2 (D)	Ea. $ 30
Story Machine (D)	$ 23
Crossword Magic (D)	$ 34
Delta Drawing (CT)	$ 23

Programming Techniques

Pilot (Cons. of Educator)	(C) $ 59 (D) $ 99
Invitation to Prog. #2 (C)	$ 22
Invitation to Prog. #3 (C)	$ 22
Tricky Tutorials — Ed Software	
TT#1 Display Lists (C, D)	$ 17
TT#2 Horiz/Vert. Scrolling (C, D)	$ 17
TT#3 Page Flipping (C, D)	$ 17
TT#4 Basics of Animation (C, D)	$ 17
TT#5 Player Missile Graphics (C, D)	$ 24
TT#6 Sound & Music (C,D)	$ 24
TT#7 Disk Utilities (D)	$ 24
TT#8 Character Graphics	$ 19
TT#9 GTIA	$ 19
TT#10 Sound Effects	$ 19
Page 6	$ 20
The Next Step	$ 27

Stocking Stuffers

WICO "BOSS" Joystick	$ 15.00
WICO Redball Joystick	$ 24.00
Elephant Disks s/s	$ 18.00
Verbatim Disks s/d	$ 24.00
Disk Savers (Plastic Sleeves) Multi Colored 1 Doz.	$ 4.50
Disk Mailers	$ 3.50
Flip 'n File Diskette Holder w/Lock (holds 25)	$ 18.50
Flip 'n File Diskette Box (holds 50)	$ 21.00
Library Carrying Case (holds 10)	$ 2.50

Super Savers

Gorilla Banana Printer	$ 209.00
Percom Single Density Drive	$ 389.00
Percom Double Density Drive	$ 515.00
Amdek Color I Monitor	$ 299.00

Call Toll Free

1-800-634-6766

Order Line Only — Ask for Operator CO58

Information & Order Inquiries (702) 369-5523

We accept VISA and MasterCard

Dealers' Inquiries Invited

New Hit List

In Search of the Most Amazing Thing	$ 27
Witness	$ 34
Cosmic Balance II	$ 27
Temple of Apshai	$ 27
Raster Blaster	$ 20
Deadline	$ 34
Richochet	$ 15
Wiz & Princess	$ 22
Ali Baba and the Forty Thieves	$ 22
Canyon Climber	$ 20
Crush, Crumble & Chomp	$ 20
Zork I, II & III	Ea. $ 27
Frogger	$ 23
Choplifter	$ 23
Curse of Ra	$ 15
Ulysses and the Golden Fleece	$ 23
Tigers in the Snow	$ 27
David's Midnight Magic (D)	$ 23
Sky Blazer (D)	$ 22
Serpentine (D)	$ 22
Sea Fox (D)	$ 20
Sands of Egypt (D)	$ 27
Pool 400 (CT)	$ 27
Gorf	(D)$ 27 (CT)$ 30
Wizard of Wor	(D)$ 27 (CT)$ 30
Cyborg (D)	$ 23
Gold Rush (D)	$ 23
Bandits (D)	$ 23
Way Out (D)	$ 27
Fast Eddy (CT)	$ 24
World War (CT)	$ 24
The Cosmic Balance (D)	$ 27
Chess (D)	$ 45
Checker (D)	$ 34
Raptillian (D,C)	$ 23
Submarine Commander (CT)	$ 34
Jumbo Jet Pilot (CT)	$ 34
Soccer (CT)	$ 34
Starcross (D)	$ 27
Zaxxon (D,C)	$ 27
Miner 2049er (CT)	$ 34
Twerps (D)	$ 23
Flip Out (D)	$ 20
The Birth of the Phoenix	$ 16
Protector II	(D)$ 23 (CT)$ 29
Baseball (CT)	$ 34
Preppie II (D,C)	$ 23
Arcade Machine (D)	$ 39
Cap n' Cosmos (D)	$ 29
Spy's Demise (D)	$ 15
Repton (D)	$ 27
Critical Mass (D)	$ 27
Millionaire (D)	$ 55
Poker Sam (D)	$ 17
Jump Man (D)	$ 27
Hellfire Warrior (D,C)	$ 27
Trion (D)	$ 27
Adventure in Time (D)	$ 20
Wavy Navy (D)	$ 23
Final Orbit (CT)	$ 23
Pharoah's Curse (D,CT)	$ 23

Printers

NEC 8023A	$ 439
Epson FX80	$ 599
Epson FX100FT	$ 799
Okidata ML82A	$ 399
Okidata ML83A	$ 615
Okidata ML84P	$ 979
Okidata 92	$ 469
Okidata 93	$ 884
Citoh 8510 Prowriter	$ 365
Mannesmann Tally MT160L	$ 599
Qume Sprint 11 +	$1349
Gemini 10X	$ 299
Gemini 15	$ 479

Monitors

Leading Edge Gorilla Hi-Res 12"	$ 85
Amdek Color I	$ 299
USI Amber	$ 149
NEC Green Screen	$ 169

1095 E. Twain, Las Vegas, Nevada 89109 (702) 796-0296 • Mon. — Fri. 8 A.M. to 6 P.M. — Sat. 9 A.M. to 5 P.M.

ORDERING INFORMATION AND TERMS: *For Fast Delivery send cashier checks, money orders or direct bank wire transfers. Personal and company checks allow 3 weeks to clear. Charges for C.O.D. orders are $3.00 minimum or 1% for orders over $300. School purchase orders welcomed. Prices reflect a cash discount only and are subject to change without notice. Please enclose your phone number with any orders.* ***SHIPPING — Software:*** *$2.50 minimum.* ***SHIPPING — Hardware:*** *(Please call)* ***SHIPPING — Foreign Orders:*** *APO & FPO orders: $10 minimum and 15% of all orders over $100. Nevada residents add 5 3/4% sales tax. All goods are new and include factory warranty. Due to our low prices, all sales are final. All returns must be accompanied by a return authorization number. Call 702-369-5523 to obtain one before returning goods for replacement.*

```
2110 DATA C82C90F458BC2084000307060F13
171B1F2300E8D0B8A0887058402800274E7500
1020305816030704010 70F17,849
2120 DATA 1F38393A3B3C3D3E3F0001020304
050607070F171F272F373F0008101820283038
0001020304050607383 93A3B,146
2130 DATA 3C3D3E3F0008101820283038070F
171F272F373FC4163402FFC85A0078FF008040
F0201000324852 1E28000306,851
2140 DATA 0900000001010202004998479645
9400071421283542 16C4000000000000000000
0000000000000000000000000,966
```

•

CHECKSUM DATA

(See p. 58)

```
1 DATA 883,955,686,427,745,192,617,545
,276,445,496,549,150,838,360,8164
1020 DATA 262,191,90,722,60,802,965,81
8,722,119,175,234,47,59,134,5400
1170 DATA 136,137,207,899,90,108,995,2
91,428,213,923,42,160,921,937,6487
1320 DATA 111,204,113,909,31,858,964,1
84,928,963,983,38,771,167,225,7449
1470 DATA 213,133,178,272,189,423,978,
759,16,231,22,694,753,57,44,4962
1620 DATA 633,21,40,7,789,733,131,276,
138,53,934,994,749,106,822,6426
1770 DATA 197,933,200,978,658,4,284,12
9,113,697,696,953,796,910,682,8230
1920 DATA 775,701,645,666,830,697,476,
704,306,162,873,59,2,614,340,7850
2070 DATA 201,116,424,770,384,230,428,
639,3192
```

•

Listing 2.

```
2 REM *** CASSETTE VERSION ***
65 IF PASS=2 THEN FOR X=1 TO 33:PUT #1
,0:NEXT X:CLOSE #1:END
70 ? "READY CASSETTE AND PRESS RETURN"
;:OPEN #1,8,128,"C:":RESTORE 200:FOR X
=1 TO 40:READ N:PUT #1,N:NEXT X
200 DATA 0,41,216,7,255,7,169,0,141,47
,2,169,60,141,2,211,169,0,141,231,2,13
3,14,169,56,141,232,2
210 DATA 133,15,169,58,133,10,169,8,13
3,11,24,96
```

•

Listing 3.

```
2 REM *** DISK VERSION ***
65 IF PASS=2 THEN PUT #1,224:PUT #1,2:
PUT #1,225:PUT #1,2:PUT #1,20:PUT #1,4
0:CLOSE #1:END
70 ? "INSERT DISK WITH DOS, PRESS RETU
RN";:DIM IN$(1):INPUT IN$:OPEN #1,8,0,
"D:AUTORUN.SYS"
90 PUT #1,255:PUT #1,255:PUT #1,0:PUT
#1,40:PUT #1,54:PUT #1,60
```

•

(Assembly language listing for **Retrofire** *starts on next page)*

```
;       ======================
;            RETROFIRE!
;
;          BY: TOM HUDSON
;       A.N.A.L.O.G. COMPUTING
;       ======================

;       ----------------
;       PAGE ZERO USAGE
;       ----------------

        ORG $80

PLOTX   DS  1           ;PLOT X VALUE
DRAWX   DS  1           ;DRAW X VALUE
PLOTY   DS  1           ;PLOT Y VALUE
DRAWY   DS  1           ;DRAW Y VALUE
PLXHI   DS  1           ;PLOT X HI BYTE
DRXHI   DS  1           ;PLOT Y HI BYTE
ACCX    DS  1           ;DRAW X ACCUMULATOR
ACCXHI  DS  1
ACCY    DS  1           ;DRAW Y ACCUMULATOR
ACCYHI  DS  1
DELTAX  DS  1           ;DRAW WORK
DELXHI  DS  1           ;DITTO
DELTAY  DS  1           ;DITTO
INCX    DS  1           ;DRAW X INCREMENT
INCY    DS  1           ;DRAW Y INCREMENT
LO      DS  1           ;INDIRECT ADDRESS
HI      DS  1
HOLD    DS  1           ;TEMP STORAGE
LOHLD   DS  1
HIHLD   DS  1
ENDPT   DS  1           ;DRAW ENDPOINT
ENDHI   DS  1
COUNTR  DS  1           ;DRAW COUNTER
CNTHI   DS  1
COLOR   DS  1           ;PLOT COLOR
XOFSET  DS  1           ;PLOT X OFFSET
X       DS  1           ;GRID COORD...
Y       DS  1           ;HOLD AREAS
HGTIX   DS  1           ;GRID HEIGHT
INDEX   DS  1           ;GRID INDEX
DESTLO  DS  1           ;MOVE DESTINATION...
DESTHI  DS  1           ;HOLD AREA
COPCNT  DS  1           ;BYTE COPY COUNT
CHARX   DS  1           ;CHARACTER X LOC
CHARY   DS  1           ;CHARACTER Y LOC
CHARNO  DS  1           ;CHARACTER #
CHINDX  DS  1           ;CHARACTER INDEX
SNUMIX  DS  1           ;NUMBER INDEX
DIGITS  DS  1           ;# OF DIGITS TO DISPLAY
SHOBYT  DS  1           ;NUMBER HOLD
VELADL  DS  1           ;VELOCITY ADD LOW
VELADH  DS  1           ;VELOCITY ADD HIGH
TIMER   DS  1           ;GENERAL EVENT TIMER
STKHLD  DS  1           ;STICK POS. HOLD
STKPOS  DS  1           ;STICK POSITION
ALTLOB  DS  1           ;LOW ALTITUDE BYTE
GRDTYP  DS  1           ;GRID TYPE
MSGCT   DS  1           ;MESSAGE COUNT
MSGPTR  DS  1           ;MESSAGE POINTER
THRCNT  DS  1           ;THRUST COUNTER
BALO    DS  1           ;BINARY ALTITUDE LOW
BAHI    DS  1           ;BINARY ALTITUDE HI
RESULT  DS  1           ;DIVISION RESULT
DIVISR  DS  1           ;DIVISOR
SHIPY   DS  1           ;SHIP GRID Y
SHIPYL  DS  1           ;SHIP Y LOW
SHIPZ   DS  1           ;SHIP GRID Z
SHIPZL  DS  1           ;SHIP Z LOW
SHPLRX  DS  1           ;SHIP PLAYER X POS
LASTSY  DS  1           ;PREVIOUS SHIP Y
SHADOY  DS  1           ;SHIP SHADOW Y
LSHADY  DS  1           ;PREVIOUS SHADOW Y
SHPLRY  DS  1           ;SHIP PLAYER Y
SHTEMP  DS  1           ;TEMPORARY STORAGE
NEWGRD  DS  1           ;NEW GRID FLAG
NOSPOS  DS  1           ;NO SHIP SHOW FLAG
GRDDIR  DS  1           ;SHIP'S MOVEMENT DIRECTION
THRVOL  DS  1           ;THRUST VOLUME
THRTIM  DS  1           ;THRUST TIME
DLICNT  DS  1           ;DLI COUNTER
MULT1   DS  1           ;MULTIPLY #1
MULT2   DS  1           ;MULTIPLY #2
LRES    DS  1           ;LOW MULT RESULT
HRES    DS  1           ;HIGH MULT RESULT
SUBY    DS  1           ;SHIP Y/Z
SUBZ    DS  1           ;WORK AREAS
TERRIX  DS  1           ;TERRAIN POSITION INDEX
T1      DS  1           ;TERRAIN...
T2      DS  1           ;HEIGHT...
T3      DS  1           ;WORK AREAS
BASEY   DS  3           ;BASE Y POSITIONS
BASEZ   DS  3           ;BASE Z POSITIONS
SYP     DS  3           ;SHIP Y POSITIONS
SZP     DS  3           ;SHIP Z POSITIONS
BASEIX  DS  1           ;BASE POSITION INDEX
PFLASH  DS  1           ;PAD FLASH...
PFLIX   DS  1           ;INDEXES
VCHANC  DS  1           ;VOLCANO CHANCE
VOLY    DS  1           ;VOLCANO Y AND Z...
VOLZ    DS  1           ;POSITIONS
VOLIX   DS  1           ;VOLCANO POSITION INDEX
VOLPX   DS  1           ;VOLCANO PLAYER X
VOLPY   DS  1           ;VOLCANO PLAYER Y
VOLRAD  DS  1           ;VOLCANO RADIATION FLAG

RADLVL  DS  1           ;RADIATION LEVEL
RADTIM  DS  1           ;RADIATION LEVEL TIMER
PAUSED  DS  1           ;PAUSE FLAG
SNDCNT  DS  1           ;SOUND COUNTER
SSTATS  DS  1           ;SOUND STATUSES
NUMBRS  DS  3           ;ALTITUDE
        DS  3           ;X VELOCITY
        DS  2           ;Y VELOCITY
        DS  2           ;Z VELOCITY
        DS  3           ;FUEL
        DS  1           ;SHIPS LEFT
        DS  3           ;SCORE
POSNUM  DS  2           ;SHIP/BASE POSITIONS

;       ------------------
;       MISC. MEMORY USAGE
;       ------------------

PMAREA  =   $0000       ;PLAYERS/MISSILES
MISSLS  =   PMAREA+768
PL0     =   PMAREA+1024
PL1     =   PMAREA+1280
PL2     =   PMAREA+1536
PL3     =   PMAREA+1792
DLIST   =   $1F00       ;DISPLAY LIST
DISP    =   $2010       ;TOP OF DISPLAY
DISP2   =   $3000       ;BOTTOM DISPLAY

;       ---------------
;       SYSTEM EQUATES
;       ---------------

VDSLST  =   $200        ;DLI POINTER
WSYNC   =   $D40A       ;WAIT FOR SYNC
KEY     =   $2FC        ;KEYBOARD KEY
CONSOL  =   $D01F       ;CONSOLE KEYS
PMBASE  =   $D407       ;P/M BASE ADDRESS
RANDOM  =   $D20A       ;RANDOM NUMBER
SETVBV  =   $E45C       ;VBI SETUP
XITVBV  =   $E462       ;VBLANK EXIT
COLBK   =   $2C8        ;COLOR REGISTERS
COLPF0  =   $2C4
COLPF1  =   $2C5
COLPF2  =   $2C6
COLPF3  =   $2C7
AUDC1   =   $D201       ;AUDIO CONTROLS
AUDC2   =   $D203
AUDC3   =   $D205
AUDC4   =   $D207
AUDF1   =   $D200
AUDF2   =   $D202
AUDF3   =   $D204
AUDF4   =   $D206
AUDCTL  =   $D208
PRIOR   =   $026F       ;P/M PRIOITY
ATTRAC  =   $4D         ;ATTRACT MODE
DMACTL  =   $22F        ;DMA CONTROL
DLISTL  =   $230        ;DISP LIST POINTER
GRACTL  =   $D01D       ;GRAPHICS CONTROL
NMIEN   =   $D40E       ;INTERRUPT ENABLE
PCOLR0  =   $D012       ;P/M COLORS
COLPM0  =   $2C0
COLPM1  =   $2C1
COLPM2  =   $2C2
COLPM3  =   $2C3
HPOSP0  =   $D000       ;PLAYER HORIZ POSITIONS
HPOSP1  =   $D001
HPOSP2  =   $D002
HPOSP3  =   $D003
HPOSM0  =   $D004
HPOSM1  =   $D005
STICK   =   $278        ;JOYSTICK 1

        ORG $2800       ;LOAD ADDRESS
        LOC $0800       ;EXECUTE ADDRESS

;       ------------------------
;       DISPLAY LIST INTERRUPT
;       ------------------------

DLI     PHA             ;SAVE ACCUM
        TXA             ;SAVE X REGISTER
        PHA
        LDX DLICNT      ;GET DLI NUMBER
        LDA ECOLOR,X    ;AND RELATED COLOR
        STA WSYNC       ;WAIT FOR SYNC
        STA $D015       ;AND SAVE COLOR
        INC DLICNT      ;NEXT DLI
        PLA             ;RESTORE X
        TAX
        PLA             ;RESTORE ACCUM
        RTI             ;ALL DONE!!

;       ----------------------------
;       RELOCATE PROGRAM TO $0800
;       (USED BY DISK VERSION ONLY!)
;       ----------------------------

MOVEIT  LDA #>2800/256  ;SET UP ORIGIN
        STA HI          ;ADDRESS IN
        LDA #>2800&255  ;2-BYTE
        STA LO          ;POINTER,
        LDA #>0800/256  ;SET UP DESTINATION
        STA BAHI        ;ADDRESS IN
        LDA #>0800&255  ;2-BYTE
        STA BALO        ;POINTER
        LDY #0          ;RESET Y REG

MOVELP  LDA (LO),Y      ;MOVE A 256-BYTE
        STA (BALO),Y    ;BLOCK USING THE
        INY             ;Y REG.
        BNE MOVELP
        INC HI          ;NEXT 256-BYTE
        INC BAHI        ;BLOCK OF DATA
        LDA HI
        CMP #$40        ;DONE YET?
        BNE MOVELP      ;NO, KEEP MOVING!
        JMP PFIRST      ;ALL DONE, START GAME!!

;       ------------------
;       MAIN PROGRAM START
;       ------------------

PFIRST  LDX #190
        LDA #>0F        ;GR.8
BDLOOP  STA DLIST,X     ;BUILD DISP LST
        DEX
        BNE BDLOOP
        LDA #>70        ;BLANK 8 LINES
        STA DLIST
        STA DLIST+1
        STA DLIST+2
        LDA #>4F        ;LOAD MEM SCAN
        STA DLIST+3
        STA DLIST+107
        LDA #DISP/256   ;DISP ADDRESS
        STA DLIST+5
        LDA #DISP&255
        STA DLIST+4
        LDA #DISP2/256  ;DISP 2 ADDRESS
        STA DLIST+109
        LDA #DISP2&255
        STA DLIST+108
        LDA #>41        ;JVB
        STA DLIST+188
        LDA #DLIST/256  ;DLIST ADDRESS
        STA DLIST+190
        LDA #DLIST&255
        STA DLIST+189
        LDA #>8F        ;DLI'S
        STA DLIST+31
        STA DLIST+57
        STA DLIST+118
        STA DLIST+128
        STA DLIST+138
        LDA #1          ;IT'S OUR...
        STA FTIME       ;FIRST TIME!

PLANET  CLD             ;NO DECIMAL MODE!
        JSR $E465       ;SETUP SOUND REGS
        JSR CLSCRN      ;CLEAR SCREEN

        LDA #0          ;ZERO OUT PAGE ZERO
        LDX #127        ;ONLY LAST 128 BYTES
CLPA00  STA $80,X       ;CLEAR BYTE
        DEX             ;MORE TO CLEAR?
        BPL CLPA00      ;YUP!

        STA DMACTL      ;SET ALL THESE
        STA NMIEN       ;VARIABLES TO
        STA AUDCTL      ;ZERO
        STA COLPF2
        STA COLBK

        JSR PMCLR       ;CLEAR P/M AREAS

        LDA #>FF        ;SET UP ENGINE...
        LDX #5          ;TEMP DISPLAYS
PSLOOP  STA PL3+145,X   ;IN PLAYER 3,
        STA PL3+155,X   ;EACH 6 SCAN LINES
        STA PL3+165,X   ;HIGH.
        DEX             ;LOOP UNTIL DONE
        BPL PSLOOP

;       ----------------------------
;       INITIALIZE STARTING STATUS
;       ----------------------------

        LDA #115        ;VOLCANO CHANCE
        STA VCHANC      ;(+-45%)

        JSR RNDBAS      ;RANDOM BASE LOCATION

        LDX #16         ;INITIALIZE THE
ININUM  LDA ININBR,X    ;ESSENTIAL NUMERIC
        STA NUMBRS,X    ;VALUES USING
        STA UNSNUM,X    ;ININBR (INITIAL NUMBERS)
        DEX             ;MORE TO MOVE?
        BPL ININUM      ;YUP!

        LDA #>0A        ;COLOR 1 WHITE
        STA COLPF1

        LDA #>0F        ;PLAYER 0 WHITE
        STA COLPM0      ;(SHIP)
        LDA #>04        ;PLAYER 1 GRAY
        STA COLPM1      ;(SHADOW)
        LDA #192        ;PLAYER 3 (ENGINE TEMPS)
        STA HPOSP3      ;HORIZONTAL POSITION

        LDA #DLI/256    ;SET UP DLI
        STA VDSLST+1    ;ADDRESS
        LDA #DLI&255
        STA VDSLST
        LDA #DLIST&255  ;SET UP DISPLAY LIST
```

```
        STA DLISTL          ;ADDRESS
        LDA #DLIST/256
        STA DLISTL+1
        LDA #PMAREA/256     ;SET UP PLAYER-MISSILE
        STA PMBASE          ;ADDRESS
        LDX #VBI/256        ;SET UP VBI
        LDY #VBI&255
        LDA #7
        JSR SETVBV
        LDA ##3E            ;TURN DMA ON
        STA DMACTL
        LDA ##3             ;ENABLE P/M
        STA GRACTL
        LDA ##C#            ;ENABLE INTERRUPTS
        STA NMIEN

        LDA #2              ;MAKE BORDER...
        STA COLOR           ;COLOR 2

;       ------------------------
;       DRAW BORDER USING TABLES
;       ------------------------

        LDX ##              ;START WITH POINT ##
BDRLP   STX SHTEMP          ;SAVE INDEX
        LDA XHIGH,X         ;GET X COORD HI
        PHA                 ;SAVE ON STACK
        LDA XLOW,X          ;GET X COORD LOW
        PHA                 ;SAVE ON STACK
        LDA YLOW,X          ;GET Y COORD
        PHA                 ;SAVE ON STACK
        LDA LINTYP,X        ;PLOT?
        BEQ ITSPLT          ;YUP!
        PLA                 ;PULL DRAW Y
        STA DRAWY           ;AND STORE
        PLA                 ;PULL DRAW X LOW
        STA DRAWX           ;AND STORE
        PLA                 ;PULL DRAW X HI
        STA DRXHI           ;AND STORE
        JSR DRAW            ;DRAW THE LINE
        JMP NXTBDR          ;DO NEXT POINT
ITSPLT  PLA                 ;PULL PLOT Y
        STA PLOTY           ;AND STORE
        PLA                 ;PULL PLOT X LOW
        STA PLOTX           ;AND STORE
        PLA                 ;PULL PLOT X HI
        STA PLXHI           ;AND STORE
        JSR PLOTCL          ;PLOT THE POINT
NXTBDR  LDX SHTEMP          ;RETRIEVE INDEX
        INX                 ;NEXT POINT
        CPX #15             ;DONE?
        BNE BDRLP           ;NO!

;       --------------------------------
;       NOW SET UP CHARACTERS ON SCREEN
;       --------------------------------

        LDX #73             ;74 CHARACTERS
SETCH   STX CHINDX          ;SAVE INDEX
        LDA XP,X            ;GET CHAR X POS
        STA CHARX           ;AND SAVE
        LDA YP,X            ;GET CHAR Y POS
        STA CHARY           ;AND SAVE
        LDA CH,X            ;GET CHAR #
        STA CHARNO          ;AND SAVE
        JSR SHONUM          ;SHOW THE CHARACTER
        LDX CHINDX          ;GET INDEX,
        DEX                 ;MORE CHARACTERS?
        BPL SETCH           ;YOU BET!

SHOWEM  LDA RANDOM          ;GET RANDOM #,
        AND ##1F            ;LIMIT TO 31 AND
        CLC                 ;ADD 4# TO GET RANGE
        ADC #4#             ;OF 4#-71
        STA SHIPY           ;STORE IT.

        LDA RANDOM          ;REPEAT FOR Z
        AND ##1F
        CLC
        ADC #4#
        STA SHIPZ

        LDA ##              ;NOW WE ZERO OUT
        LDX #5              ;ALL THE ENGINE
CLTEMP  STA TEMP1,X         ;TEMPERATURES (COOL)
        DEX
        BPL CLTEMP

        STA SIGNS           ;AND ZERO THE
        STA SIGNS+1         ;SIGN VARIABLES
        STA SIGNS+2
        STA SIGNS+3

        LDA ##C4            ;ENGINE COLORS ALL GREEN
        STA ECOLOR+2
        STA ECOLOR+3
        STA ECOLOR+4

        LDA #121            ;INIT RADIATION
        STA RADLVL          ;LEVEL TO LOW
        LDA ##0F            ;AND SHOW WITH
        STA PL3+121         ;PLAYER 3

        LDA ##              ;NEXT WE ERASE ANY
        LDX #47             ;OTHER RADIATION
CLRAD   STA PL3,X           ;GRAPHICS IN PLAYER 3
        INX
        CPX #122            ;ALL CLEAR?
        BCC CLRAD           ;NOT YET!

        JSR SHOALT          ;SHOW ALTITUDE,
        JSR SHOXV           ;X VELOCITY,
        JSR SHOYV           ;Y VELOCITY
        JSR SHOZV           ;Z VELOCITY
        JSR SHOFUL          ;FUEL
        JSR SHOSHP          ;SHIPS LEFT
        JSR SHOSCO          ;AND SCORE
        LDA ##              ;GRID TYPE #...
        STA GRDTYP          ;(HI ALTITUDE)
        JSR CLMESS          ;CLEAR MESSAGE AREA
        JSR RNDGRD          ;RANDOMIZE LAND GRID
        JSR CLGRDA          ;CLEAR GRID GRAPHICS AREA
        JSR GRID            ;DRAW LAND GRID
        LDA FTIME           ;FIRST TIME?
        BEQ MAINLN          ;NO, CONTINUE
        DEC FTIME           ;NO LONGER 1ST TIME,
        JMP RESTRT          ;GO SHOW OPTIONS

MAINLN  LDA PAUSED          ;PAUSED?
        BNE MAINLN          ;YES, WE'RE PAUSED!
        LDA TIMER           ;READY FOR PROCESSING?
        BEQ DOMESS          ;YES!
        JMP CKCONS          ;NO, CHECK CONSOLE
DOMESS  LDA #1              ;RESET TIMER
        STA TIMER           ;TO 1/6# SEC
        STA ATTRAC          ;NO ATTRACT MODE

;       ---------------------------
;       CHECK RADIATION LEVEL TIMER
;       ---------------------------

        LDA NUMBRS          ;OVER 3####0 FT?
        CMP #3
        BCC CKRTIM          ;NO!
        LDA ##              ;YES, HIGH RAD!
        STA RADTIM
CKRTIM  LDA RADTIM          ;READY FOR MORE RADIATION?
        BNE ADJALT          ;NOT YET.
        LDA VOLRAD          ;IS IT VOLCANO?
        BNE GOVOLR          ;YES!
        LDX GRDTYP          ;GET GRID TYPE (ALT)
        LDA RADINI,X        ;IS THERE RADIATION?
        BEQ ADJALT          ;NO RADIATION
        STA RADTIM          ;RESET RADIATION TIMER
GOVOLR  LDA RADLVL          ;GET RADIATION AMOUNT
        CMP #48             ;FATAL?
        BCS RADOK           ;NO, WE'RE OK.
        JMP CRASH           ;KABOOM!!!
RADOK   SEC                 ;ONE...
        SBC #1              ;MORE...
        STA RADLVL          ;RADIATION UNIT!
        TAX                 ;USE AS INDEX,
        LDA ##0F            ;AND SHOW ON SCREEN
        STA PL3,X           ;USING PLAYER 3
        LDA ##              ;RESET VOLCANO
        STA VOLRAD          ;RADIATION FLAG

;       ---------------
;       ADJUST ALTITUDE
;       ---------------

ADJALT  SED                 ;SET DECIMAL MODE
        LDA SIGNS           ;ADD OR SUBTRACT?
        BEQ SUBALT          ;SUBTRACT!
        JMP ADDALT          ;ADD!

JENDAJ  JMP ENDAAJ

SUBALT  LDA ALTLOB          ;GET ALT FRACTION BYTE
        SEC                 ;SUBTRACT
        SBC NUMBRS+4        ;X VELOCITY LO
        STA ALTLOB          ;AND SAVE BACK
        LDA NUMBRS+2        ;GET ALT LO BYTE
        SBC NUMBRS+3        ;SUB XV MED
        STA NUMBRS+2        ;AND SAVE
        LDA NUMBRS+1        ;GET ALT MED BYTE
        SBC ##              ;SUB # & CARRY
        STA NUMBRS+1        ;AND SAVE
        LDA NUMBRS          ;GET ALT HI BYTE
        SBC ##              ;SUB # & CARRY
        STA NUMBRS          ;AND SAVE
        CMP ##9#            ;NEGATIVE ALT?
        BCC JENDAJ          ;NO, STILL IN AIR!

;       ------------------------------
;       AT THIS POINT, WE'VE HIT GROUND
;       ------------------------------

        CLD                 ;NO MORE DECIMAL MODE
        LDA NUMBRS+3        ;TALLY HIGH-ORDER
        ORA NUMBRS+6        ;VELOCITIES TO SEE
        ORA NUMBRS+8        ;IF ANY ARE >#
        BNE CRASH           ;TOO FAST!
        LDA NUMBRS+4        ;IS X VELOCITY
        CMP ##11            ;> 1# FEET/SEC?
        BCS CRASH           ;YES!
        LDA NUMBRS+7        ;IS Y VELOCITY
        CMP ##6             ;>5 FEET/SEC?
        BCS CRASH           ;YES!
        LDA NUMBRS+9        ;IS Z VELOCITY
        CMP ##6             ;>5 FEET/SEC?
        BCS CRASH           ;YES!
        LDA TERRIX          ;DID WE LAND
        CMP BASEIX          ;ON THE BASE?
        BNE CRASH           ;NO!
        LDX #1              ;SAFE LANDING
        LDY #14             ;AT 14TH SCREEN LINE
        JSR SHOMSG          ;SHOW THE MESSAGE
        LDA #255            ;WAIT 255/6#
        STA TIMER           ;(4.25) SECONDS
LDELAY  LDA TIMER           ;TIME UP?
        BNE LDELAY          ;NO, LOOP BACK

        LDX #3
FSHFLP  ASL NUMBRS+12       ;MULT
        ROL NUMBRS+11       ;FUEL
        ROL NUMBRS+1#       ;BY 1#
        DEX                 ;(THIS IS 4 SHIFTS
        BPL FSHFLP          ;IN DECIMAL MODE)
        SED                 ;SELECT DECIMAL MODE
        LDX #2
        CLC
ADSCLP  LDA NUMBRS+1#,X     ;ADD
        ADC NUMBRS+14,X     ;FUEL X 1# TO
        STA NUMBRS+14,X     ;SCORE
        DEX
        BPL ADSCLP
        CLD                 ;NO MORE DECIMAL
        JSR SHOSCO          ;SHOW SCORE

SCOTIM  LDA #24#            ;WAIT 4 SECS
        STA TIMER
STWAIT  LDA TIMER           ;TIME UP?
        BNE STWAIT          ;NO, WAIT MORE
        LDA VCHANC          ;IS VOLC CHANCE
        CMP #255            ;AT MAXIMUM?
        BEQ GOAGIN          ;YES!
        CLC                 ;NO, INCREMENT
        ADC #1#             ;VOLCANO CHANCE
        STA VCHANC          ;BY ABOUT 4%
        JSR RNDBAS          ;GET NEW BASE LOC.
GOAGIN  JMP NEWLND          ;AND START LANDING CYCLE

CRASH   LDA RADLVL          ;WAS DEATH DUE
        CMP #47             ;TO RADIATION?
        BNE IMPCRS          ;NO, DUE TO IMPACT
        LDX #5              ;POINT TO RAD MESSAGE
        BNE SHOCMS          ;GO SHOW THE MESSAGE
IMPCRS  LDX #2              ;POINT TO IMPACT MSG
SHOCMS  LDY #14             ;14TH SCREEN LINE
        JSR SHOMSG          ;SHOW THE MESSAGE!
CRSHLP  LDA SHPLRY          ;CONVERT PLAYER Y
        SEC                 ;COORD TO SR.8
        SBC #34
        LDX #19             ;AND PUT IN ALL
SCEXY   STA EXYP,X          ;EXPLOSION Y COORDS
        DEX
        BPL SCEXY
        LDA SHPLRX          ;CONVERT PLAYER X
        SEC                 ;COORD TO SR.8
        SBC #48
        ASL A
        LDX #19             ;AND PUT IN ALL
SCEXX   STA EXXP,X          ;EXPLOSION X COORDS
        DEX
        BPL SCEXX
        LDA ##              ;POSITION SHIP
        STA SHPLRX          ;AND SHADOW
        STA SHPLRY          ;OFF-SCREEN
        LDX #19             ;NOW RANDOMIZE
SCEXIS  LDA RANDOM          ;EXPLOSION...
        ORA ##1#
        STA EXXI,X          ;X INCREMENT,
        LDA RANDOM
        ORA ##1#
        STA EXYI,X          ;Y INCREMENT,
        LDA ##
        STA EXXA,X          ;X ACCUMULATOR,
        STA EXYA,X          ;Y ACCUMULATOR,
        LDA RANDOM
        ORA ##1#
        AND ##3F
        STA EXLV,X          ;PIXEL LIFE.
        DEX
        BPL SCEXIS
        BMI SETEXT
EXGO    LDA TIMER           ;TIME TO ADVANCE EXPL?
        BNE EXGO            ;NOT YET!
SETEXT  LDA #1              ;IT'S TIME!
        STA TIMER           ;RESET TIMER
        LDX #19             ;2# PIXELS
EXLP    STX TEMPCX          ;SAVE INDEX
        LDA EXLV,X          ;PIXEL ALIVE?
        BEQ JXDEC           ;NO!
        LDA EXXP,X          ;NOW WE ERASE OLD
        STA PLOTX           ;PIXEL, SETTING UP
        LDA EXYP,X          ;THE X AND Y
        STA PLOTY           ;COORDINATES
        JSR PCALC           ;GET ADDRESS OF PLOT
        LDY ##
        LDA BMASK2,X        ;MASK OFF PIXEL
        EOR (LO),Y
        STA (LO),Y          ;AND ERASE IT!
        LDX TEMPCX          ;GET INDEX BACK
        DEC EXLV,X          ;DECREMENT LIFE
        BEQ JXDEC           ;IT'S DEAD!
        LDA EXXI,X          ;NOW MOVE PIXEL
        CLC                 ;IN X DIRECTION
        ADC EXXA,X
        STA EXXA,X
        PHP                 ;SAVE CARRY FLAG
```

```
        CPX #10          ;IF PIXEL >9,
        BCS ADEXX        ;ADD INCREMENT
        PLP              ;RESTORE CARRY
        LDA EXXP,X       ;OTHERWISE
        SBC #0           ;SUBTRACT IT!
        JMP STOEXX       ;GO STORE RESULT
JXDEC   JMP EXDEC
ADEXX   PLP              ;RESTORE CARRY
        LDA EXXP,X       ;AND ADD
        ADC #0           ;THE CARRY FLAG
STOEXX  STA EXXP,X       ;SAVE X POSITION
        CMP #192         ;IN DISPLAY WINDOW?
        BCS KILEXB       ;TOO FAR RIGHT!
        CMP #10          ;OK ON LEFT?
        BCC KILEXB       ;TOO FAR LEFT!
        STA PLOTX        ;SAVE IN PLOT COORD.
        LDA EXYI,X       ;INCREMENT Y POSITION
        CLC
        ADC EXYA,X
        STA EXYA,X
        LDA EXYP,X
        SBC #0           ;SUBTRACT INCREMENT (UP)
        STA EXYP,X       ;SAVE IT
        CMP #10          ;ON SCREEN?
        BCC KILEXB       ;OFF THE TOP!
        STA PLOTY        ;SAVE IN PLOT COORD.
        JSR PCALC        ;GET PLOT ADDRESS
        LDY #0
        LDA BMASK2,X
        EOR (LO),Y
        STA (LO),Y       ;AND PLOT PIXEL!
EXDEC   LDX TEMPCX       ;MORE PIXELS?
        DEX
        BMI CHKEXF       ;NO!
        JMP EXLP
KILEXB  LDX TEMPCX       ;GET INDEX OF PIXEL
        LDA #0           ;AND ZERO OUT
        STA EXLV,X       ;LIFE BYTE (IT'S DEAD)
        BEQ EXDEC        ;DO NEXT PIXEL
CHKEXF  LDX #19          ;TALLY ALL LIFE
        LDA #0           ;INDICATORS
ACCUEX  ORA EXLV,X
        DEX
        BPL ACCUEX
        CMP #0           ;ANY ALIVE?
        BEQ EXDONE       ;NO, EXPLOSION DONE!
        JMP EX60         ;YES, DO MORE

EXDONE  LDA #255         ;WAIT 255/60 (4.25) SECS
        STA TIMER
CTIMWT  LDA TIMER        ;TIME UP?
        BNE CTIMWT       ;NO, WAIT MORE
        DEC NUMBRS+13    ;ONE LESS SHIP
        BMI OUTSHP       ;NO MORE SHIPS!
        JMP NXTSHP       ;GO DO NEXT SHIP
OUTSHP  LDX #0           ;GAME OVER MESSAGE
        LDY #14          ;14TH LINE ON SCREEN
        JSR SHOMS0       ;SHOW THE MESSAGE
RESTRT  LDA KEY          ;GET KEYBOARD KEY
        CMP #$12         ;IS IT A "C"?
        BNE NOCTGL       ;NO TOGGLE
        LDX COLORS+1     ;SWAP COLORS 1 & 2
        LDA COLORS+2
        STA COLORS+1
        STX COLORS+2
        LDA #$FF         ;CANCEL KEY
        STA KEY
        JMP PFIRST       ;REDRAW SCREEN
NOCTGL  LDX #0           ;TITLE
        LDY #30          ;30TH LINE
        JSR SHOMS0
        LDX #9           ;AUTHOR
        LDY #45          ;45TH LINE
        JSR SHOMS0
        LDX #10          ;MAGAZINE
        LDY #55          ;55TH LINE
        JSR SHOMS0
        LDX GRAVTY       ;GET GRAVITY INDEX
        LDA GCHAR,X      ;GET L/M/H
        STA GRVMS0+15    ;PUT IN MESSAGE
        LDX #7           ;GRAVITY
        LDY #75          ;75TH LINE
        JSR SHOMS0
        LDA IFUEL1       ;GET FUEL CHAR1
        ORA #$10         ;GET CHARACTER OFFSET
        STA IFUMS0+12    ;PUT IN FUEL MESSAGE
        LDA IFUEL2       ;GET FUEL CHAR2
        ORA #$10         ;GET CHARACTER OFFSET
        STA IFUMS0+13    ;PUT IN FUEL MESSAGE
        LDX #6           ;FUEL
        LDY #85          ;85TH LINE
        JSR SHOMS0
RSTTIM  LDA TIMER        ;CONSOLE TIME READY?
        BNE RSTTIM       ;NOT YET
        LDA CONSOL       ;GET CONSOLE KEY
        CMP #7           ;ANY PRESSED?
        BEQ RESTRT       ;NO!
        CMP #6           ;START KEY?
        BEQ RESTCK       ;YES! RESTART
        CMP #3           ;OPTION KEY?
        BNE NOTOPT       ;NO, IT'S SELECT!
        LDA GRAVTY       ;GET GRAVITY
        CLC              ;ADD 1 TO GET NEXT
        ADC #1           ;GRAVITY LEVEL
        CMP #3           ;BEYOND 0-2?
        BNE STGRAV       ;NO, STORE IT.
        LDA #0           ;WRAP TO 0

STGRAV  STA GRAVTY       ;STORE GRAVITY
        JMP CDELAY       ;AND DEBOUNCE CONSOLE

NOTOPT  LDA IFUEL2       ;INCREMENT FUEL
        CLC              ;AMOUNT (5000-14000)
        ADC #1
        STA IFUEL2
        CMP #10
        BNE CKFLIM
        LDA #0
        STA IFUEL2
        LDA IFUEL1
        CLC
        ADC #1
        STA IFUEL1
CKFLIM  LDA IFUEL1       ;FUEL > 9000?
        BEQ CDELAY       ;NO, GO DEBOUNCE CONSOLE
        LDA IFUEL2
        CMP #5           ;FUEL = 15000?
        BNE CDELAY       ;NO
        LDA #0           ;RESET FUEL TO 5000
        STA IFUEL1

CDELAY  LDA #30          ;WAIT 30/60 (1/2) SEC
        STA TIMER
        JMP RESTRT       ;KEEP CHECKING CONSOLE

RESTCK  LDA CONSOL       ;CONSOLE RELEASED?
        CMP #7
        BNE RESTCK       ;NOT RELEASED, WAIT
        LDA IFUEL1       ;GET SELECTED FUEL
        STA ININBR+10    ;AND PLACE IN
        LDA IFUEL2       ;INITIAL NUMBER
        ASL A            ;TABLE FOR PROPER
        ASL A            ;INITIAL FUEL
        ASL A
        ASL A
        STA ININBR+11
        JMP PLANET       ;RESTART GAME

NXTSHP  JSR SHOSHP       ;SHOW # OF SHIPS
NEWLND  LDX #12          ;NOW WE MUST
REINI   LDA ININBR,X     ;RE-INITIALIZE
        STA NUMBRS,X     ;ALL VITAL NUMBERS
        STA UNSNUM,X     ;AND UNSIGNED NUMBERS
        DEX
        BPL REINI
        JMP SHOWEM       ;AND RESTART LANDING

ADDALT  LDA ALTLOB       ;THIS ROUTINE
        CLC              ;ADDS OUR X VELOCITY
        ADC NUMBRS+4     ;TO THE ALTITUDE
        STA ALTLOB       ;WHEN WE'RE GOING UP.
        LDA NUMBRS+2
        ADC NUMBRS+3
        STA NUMBRS+2
        LDA NUMBRS+1
        ADC #0
        STA NUMBRS+1
        LDA NUMBRS
        ADC #0
        STA NUMBRS
ENDAAJ  CLD              ;NO MORE DECIMAL MODE
        JSR SHOALT       ;SHOW ALTITUDE
        JSR SHCALC       ;SHIP CALCS.
        LDX #0           ;FIND THE SCALE
        LDA NUMBRS       ;OF THE LANDING GRID
        BNE GOTSIZ       ;BASED ON OUR ALTITUDE
        LDX #2
CKSIZE  LDA NUMBRS+1     ;COMPARE ALTITUDE TO
        CMP GRDLIM,X     ;GRID SCALE LIMITS
        BCC GOTSIZ
        DEX
        BNE CKSIZE
GOTSIZ  CPX GRDTYP       ;SAME SCALE AS PREVIOUS?
        BEQ DOGRAV       ;YUP!
        BCS GODOWN       ;NO, WE'RE GOING DOWN!
        STX GRDTYP       ;SAVE NEW GRID TYPE
        LDA #1           ;GET RID OF SHIP
        STA NOSPOS       ;TEMPORARILY
        LDA SHIPY        ;CALCULATE NEW
        JSR CSCUP        ;SHIP Y COORD
        LDA BYP,X
        ASL A
        ASL A
        ASL A
        ASL A
        CLC
        ADC RESULT
        STA SHIPY
        LDA SHIPZ        ;AND NEW Z COORD
        JSR CSCUP
        LDA BZP,X
        ASL A
        ASL A
        ASL A
        ASL A
        CLC
        ADC RESULT
        STA SHIPZ
        JMP GENGRD       ;GENERATE NEW GRID

GODOWN  STX GRDTYP       ;SAVE NEW GRID TYPE
        LDA #1           ;GET RID
        STA NOSPOS       ;OF SHIP
        LDA SHIPY        ;CALCULATE NEW
        JSR CSCDN        ;SHIP Y COORD
        STA SHIPY
        TYA
        LDX GRDTYP
        STA BYP-1,X
        LDA SHIPZ        ;AND SHIP Z COORD
        JSR CSCDN
        STA SHIPZ
        TYA
        LDX GRDTYP
        STA BZP-1,X
GENGRD  JSR CLGRDA       ;CLEAR GRID AREA
        JSR RNDGRD       ;RANDOMIZE NEW GRID
        JSR GRID         ;AND DRAW GRID

DOGRAV  LDA UNSNUM+3     ;HAS SHIP REACHED
        CMP #$05         ;TERMINAL VELOCITY?
        BEQ NOGRAV       ;YES--DON'T ACCELLERATE
        SED              ;GO TO DECIMAL MODE
        LDA UNSNUM+5     ;GET X VELOCITY
        CLC
        LDX GRAVTY       ;GET GRAVITY INDEX 0-2
        ADC GRAVS,X      ;ADD PROPER GRAVITY
        STA UNSNUM+5     ;SAVE NEW XV
        STA NUMBRS+5
        LDA UNSNUM+4
        ADC #0
        STA UNSNUM+4
        STA NUMBRS+4
        LDA UNSNUM+3
        ADC #0
        STA UNSNUM+3
        STA NUMBRS+3
        LDX #3           ;POINT TO XV
        LDY #0           ;SIGN INDEX
        JSR NEGHAN       ;CHECK FOR NEGATIVE
NOGRAV  JSR SHOXV        ;SHOW XV ON SCREEN
RDSTIK  LDX STICK        ;GET STICK POSITION
        STX STKHLD       ;SAVE IT
        LDA ENGIX,X      ;THRUSTING?
        BNE THRUST       ;YUP!
JADJYZ  JMP ADJYZP       ;ADJUST Y/Z POS
THRUST  LDA NUMBRS+10    ;TALLY FUEL
        ORA NUMBRS+11
        ORA NUMBRS+12
        BNE GOFUEL       ;WE'VE GOT FUEL
        LDA SSTATS       ;DID WE ALREADY DO
        AND #1           ;OUT-OF-FUEL SOUND?
        BNE NOFUEL       ;YES!
        LDA SSTATS       ;SET OUT-OF-FUEL FLAG
        ORA #1
        STA SSTATS
        LDA #255         ;LOW PITCH
        STA AUDF3        ;IN SOUND 3
        LDA #20          ;20/60 (1/3) SEC
        STA SNDCNT       ;SOUND DURATION
        LDX #4           ;OUT OF FUEL MESSAGE
        LDY #14          ;14TH SCREEN LINE
        JSR SHOMS0       ;SHOW MESSAGE
NOFUEL  JMP NOTHR        ;DON'T THRUST (NO FUEL)
GOFUEL  LDA NUMBRS+10    ;FUEL>10000?
        BNE FUELOK       ;YES, WE'RE OK
        LDA NUMBRS+11
        CMP #$10         ;FUEL>9999?
        BCS FUELOK       ;YES, STILL OK
        LDA SSTATS       ;HAVE WE DONE
        AND #2           ;FUEL LOW SOUND?
        BNE FUELOK       ;YES, GO DO THRUST
        LDA SSTATS       ;SET FUEL LOW FLAG
        ORA #2
        STA SSTATS
        LDA #80          ;MEDIUM PITCH
        STA AUDF3        ;IN SOUND 3
        LDA #20          ;20/60 (1/3) SEC
        STA SNDCNT       ;SOUND DURATION
        LDX #3           ;FUEL LOW MESSAGE
        LDY #14          ;14TH SCREEN LINE
        JSR SHOMS0       ;SHOW THE MESSAGE
FUELOK  LDY STKHLD       ;GET STICK TO FIND
        LDX ENGIX,Y      ;WHICH ENGINE'S FIRING
        LDY TEMPS-1,X    ;GET ENGINE TEMPERATURE
        LDA ENGTHR,Y     ;AND THRUST FOR THAT TEMP
        BEQ NOTHR        ;ENGINE'S DEAD!
        CMP RANDOM       ;ENGINE MAY BE FIRING
        BCS THROK        ;IT IS!
NOTHR   JMP ADJYZP       ;NO THRUST!
THROK   LDA #10          ;SET UP
        STA THRVOL       ;THRUST VOLUME
        INC THRCNT       ;ONLY ALLOW THRUST
        LDA THRCNT       ;EVERY OTHER TIME
        AND #1
        BNE JADJYZ       ;NO THRUST!
        SED              ;GO TO DECIMAL MODE
        LDA NUMBRS+12    ;AND SUBTRACT
        SEC              ;2 UNITS
        SBC #2           ;FROM FUEL
        STA NUMBRS+12
        LDA NUMBRS+11
        SBC #0
        STA NUMBRS+11
        LDA NUMBRS+10
        SBC #0
        STA NUMBRS+10
        CLD              ;NO MORE DECIMAL MODE
        JSR SHOFUL       ;SHOW THE NEW FUEL AMT.
        LDX STKHLD       ;GET STICK FOR INDEX,
        LDA VAL,X        ;AND GET AMOUNT
        STA VELADL       ;TO ALTER VELOCITY,
```

```
        LDA VAH,X       ;EITHER 1 OR -1
        STA VELADH      ;(##1 OR #99)
        LDY FUELIX,X    ;GET INDEX TO VELOCITIES
        CMP ##99        ;NEGATIVE VEL ALTER?
        BEQ CKNLIM      ;YES
        LDA UNSNUM,Y    ;AT MAXIMUM VELOCITY
        CMP ##65        ;OF 500 FT/SEC?
        BEQ ADJYZP      ;YES!
        JMP ADDTHR      ;GO ADD THE THRUST
CKNLIM  LDA UNSNUM,Y    ;AT MINIMUM VELOCITY
        CMP ##95        ;OF -500 FT/SEC
        BNE ADDTHR      ;NO!
        LDA UNSNUM+1,Y  ;AT -500 FT/SEC?
        CMP ###0        ;
        BEQ ADJYZP      ;YES!
ADDTHR  LDY FSIGNS,X    ;GET VELOCITY DIRECTION
        LDA FUELIX,X    ;AND VELOCITY INDEX
        TAX
        JSR ADDV        ;ADD THE VELOCITY
        JSR SHOYV       ;SHOW Y VELOCITY
        JSR SHOZV       ;AND Z VELOCITY

;       ------------------
;       ADJUST THE Y COORD
;       ------------------

ADJYZP  LDY #6          ;CONVERT YV
        JSR CVDBIN      ;TO BINARY
        LDX GRDTYP      ;GET GRID TYPE FOR SCALE
        LDY VELSHF,X    ;AND DIVIDE BY
SHFYLP  LSR HI          ;THE PROPER FACTOR
        ROR LO          ;FOR THE GRID'S
        DEY             ;MAGNIFICATION
        BNE SHFYLP
        LDA SIGNS+1     ;IS VELOCITY NEGATIVE?
        BNE SUBYV       ;YES, SUBTRACT IT
        LDA SHIPYL      ;ADD SHIP'S VELOCITY
        CLC             ;TO THE SHIP'S
        ADC LO          ;Y POSITION
        STA SHIPYL
        BCC CKYLIM
        INC SHIPY
        JMP CKYLIM      ;AND CHECK LIMIT
SUBYV   LDA SHIPYL      ;SUBTRACT VELOCITY
        SEC             ;FROM THE SHIP'S
        SBC LO          ;POSITION
        STA SHIPYL
        BCS CKYLIM
        DEC SHIPY
CKYLIM  LDA #0          ;NOT ON
        STA NEWGRD      ;NEW GRID YET
        STA GRDDIR      ;ZERO GRID DIRECTION
        LDA SHIPY       ;IS SHIP Y
        CMP #180        ;< 0?
        BCS YLO         ;YES!
        CMP #112        ;IS IT > 111?
        BCC NAVY        ;NO, GO CHECK Z
        SEC             ;ADJUST IT BACK
        SBC #112        ;DOWN WITHIN THE
        STA SHIPY       ;Y LIMIT
        LDX GRDTYP      ;NOW ADJUST THE
        BEQ SETGYI      ;SHIP'S NAVIGATIONAL
NXTGYI  LDA GYP-1,X     ;COORDINATES IF
        CLC             ;WE'RE NOT ON
        ADC #1          ;GRID 0 (HIGHEST)
        CMP #7
        BNE STGYI
        LDA #0
        STA GYP-1,X
        DEX
        BNE NXTGYI
        BMI SETGYI
STGYI   STA GYP-1,X
SETGYI  INC NEWGRD      ;GOING TO NEW GRID
        JMP NAVY        ;GO ADJUST Z
YLO     CLC             ;ADJUST Y UP
        ADC #112        ;BY 112
        STA SHIPY
        LDX GRDTYP      ;ADJUST NAVIGATIONAL
        BEQ SETGYD      ;COORDINATES
NXTGYD  LDA GYP-1,X     ;IF WE'RE NOT
        SEC             ;ON GRID TYPE 0
        SBC #1
        BPL STGYD
        LDA #6
        STA GYP-1,X
        DEX
        BNE NXTGYD
        BMI SETGYD
STGYD   STA GYP-1,X
SETGYD  INC NEWGRD      ;GOING TO NEW GRID
        LDA #1          ;DIRECTION 1
        STA GRDDIR

NAVY    LDX GRDTYP      ;THIS ROUTINE FINDS
        LDA SHIPY       ;THE SHIP'S "Y"
        LSR A           ;NAVIGATIONAL POSITION
        LSR A
        LSR A
        LSR A
        STA GYP,X
        LDA #0
SYFINE  INX
        CPX #3
        BEQ NOSYF
        STA GYP,X
        BNE SYFINE

NOSYF   LDY #2          ;SHOW SHIP Y
        JSR SHOPOS      ;NAVIGATIONAL POSITION

;       ---------------
;       NOW THE Z COORD
;       ---------------

        LDY #8          ;THIS ROUTINE IS THE
        JSR CVDBIN      ;SAME AS THE ABOVE
        LDX GRDTYP      ;"Y" ROUTINE, EXCEPT
        LDY VELSHF,X    ;THAT IT ADJUSTS THE
SHFZLP  LSR HI          ;SHIP "Z" COORD.
        ROR LO
        DEY
        BNE SHFZLP
        LDA SIGNS+2
        BNE SUBZV
        LDA SHIPZL
        CLC
        ADC LO
        STA SHIPZL
        BCC CKZLIM
        INC SHIPZ
        JMP CKZLIM
SUBZV   LDA SHIPZL
        SEC
        SBC LO
        STA SHIPZL
        BCS CKZLIM
        DEC SHIPZ
CKZLIM  LDA SHIPZ
        CMP #180
        BCS ZLO
        CMP #112
        BCC CKCONS
        SEC
        SBC #112
        STA SHIPZ
        LDX GRDTYP
        BEQ SETGZI
NXTGZI  LDA GZP-1,X
        CLC
        ADC #1
        CMP #7
        BNE STGZI
        LDA #0
        STA GZP-1,X
        DEX
        BNE NXTGZI
        BMI SETGZI
STGZI   STA GZP-1,X
SETGZI  INC NEWGRD
        LDA #2          ;DIRECTION 2
        STA GRDDIR
        JMP CKCONS
ZLO     CLC
        ADC #112
        STA SHIPZ
        LDX GRDTYP
        BEQ SETGZD
NXTGZD  LDA GZP-1,X
        SEC
        SBC #1
        BPL STGZD
        LDA #6
        STA GZP-1,X
        DEX
        BNE NXTGZD
        BMI SETGZD
STGZD   STA GZP-1,X
SETGZD  INC NEWGRD
        LDA #3          ;DIRECTION 3
        STA GRDDIR

CKCONS  LDX GRDTYP      ;FIND SHIP'S Z
        LDA SHIPZ       ;NAVIGATIONAL
        LSR A           ;COORDINATES
        LSR A
        LSR A
        LSR A
        STA GZP,X
        LDA #0
SZFINE  INX
        CPX #3
        BEQ NOSZF
        STA GZP,X
        BNE SZFINE
NOSZF   LDY #3          ;SHOW SHIP'S Z
        JSR SHOPOS      ;NAVIGATIONAL POSITION
        LDA NEWGRD      ;ARE WE ON NEW GRID?
        BEQ NONEWG      ;NO
        JSR CLGRDA      ;CLEAR GRID AREA
        LDX GRDDIR      ;GET GRID DIRECTION AND
        LDY GDIX,X      ;SAVE THE HEIGHTS OF
        LDA #7          ;THAT GRID SIDE.
        STA HOLD
CGSIDE  LDX HSAVE,Y
        LDA HEIGHT,X    ;SAVE
        LDX HOLD
        STA HGTSAV,X    ;SIDE'S
        DEY             ;COORDS
        DEC HOLD
        BPL CGSIDE

        JSR RNDGRD      ;RANDOMIZE NEW GRID

        LDX GRDDIR      ;MOVE THE COORDINATES
        LDY GDIX,X      ;WE JUST SAVED TO
        LDA #7          ;THE NEW GRID'S
        STA HOLD        ;OPPOSITE SIDE
RSTSID  LDX HOLD
        LDA HGTSAV,X
        LDX MREST,Y
        STA HEIGHT,X
        DEY
        DEC HOLD
        BPL RSTSID

        JSR GRID        ;DRAW THE GRID
NONEWG  LDA CONSOL      ;ARE ANY
        CMP #7          ;CONSOLE KEYS PRESSED?
        BNE WTRELS      ;YES, WAIT FOR RELEASE
        JMP MAINLN      ;NO, KEEP PROCESSING
WTRELS  LDA CONSOL      ;IS CONSOLE KEY
        CMP #7          ;RELEASED?
        BNE WTRELS      ;NOT YET!
        JMP PFIRST      ;RESTART GAME

;       ------------------------
;       DESCENT SCALING ROUTINE
;       ------------------------

CSCDN   LDY #0          ;DIVIDE SHIP COORD
CSCLP   CMP #16         ;BY 16.  PUT QUOTIENT
        BCC NOCADJ      ;IN Y AND REMAINDER
        SEC             ;IN ACCUMULATOR
        SBC #16
        INY
        JMP CSCLP
NOCADJ  STA HOLD        ;NOT MULT A BY 7
        ASL A           ;*2
        CLC
        ADC HOLD        ;*3
        ASL A           ;*6
        ADC HOLD        ;*7
        RTS             ;ALL DONE!

;       ------------------------
;       ASCENT SCALING ROUTINE
;       ------------------------

CSCUP   STA RESULT      ;DIVIDE SHIP COORD
        LDA #7          ;BY 7
        STA DIVISR
        LDA #0
        JSR DIVIDE
        LDX GRDTYP      ;GET GRID TYPE
        RTS             ;AND EXIT.

;       ------------------------------
;       DECIMAL TO BINARY WORK ROUTINE
;       ------------------------------

CVDBIN  LDA NUMBRS+1,Y  ;GET DECIMAL NUMBER
        AND ##0F        ;MASK OFF LOW DIGIT
        STA LO          ;SAVE
        LDA NUMBRS+1,Y  ;GET DECIMAL
        LSR A           ;MASK OFF HIGH DIGIT
        LSR A
        LSR A
        LSR A
        TAX             ;PUT IN INDEX
        LDA LO          ;GET PREVIOUS DIGIT
        CLC             ;AND ADD
        ADC ALT1L,X     ;DIGIT2 X 10
        STA LO          ;SAVE IT
        LDA NUMBRS,Y    ;GET NEXT DECIMAL #
        AND ##0F        ;MASK OFF LOW DIGIT
        TAX             ;PUT IN INDEX
        LDA LO          ;GET PREVIOUS RESULT
        CLC             ;AND ADD
        ADC ALT2L,X     ;DIGIT3 X 100 PART 1
        STA LO          ;SAVE IT
        LDA #0          ;AND CARRY
        ADC ALT2H,X     ;DIGIT3 X 100 PART 2
        STA HI          ;AND SAVE IT
        RTS             ;ALL DONE!

;       --------------
;       SHOW SPACESHIP
;       --------------

SHCALC  NOP

;       ------------------------------
;       CONVERT BCD ALTITUDE TO BINARY
;       ------------------------------

        LDA NUMBRS+2    ;SIMILAR TO ABOVE
        AND ##0F        ;ROUTINE BUT WILL
        STA BALO        ;WORK FOR 3 BCD DIGITS
        LDA #0          ;AND IS DEDICATED
        STA BAHI        ;FOR ALTITUDE ONLY
        LDA NUMBRS+2
        LSR A
        LSR A
        LSR A
        LSR A
        TAX
        LDA BALO
        CLC
        ADC ALT1L,X
        STA BALO
        LDA NUMBRS+1
```

```
        AND #$0F
        TAX
        LDA BALO
        CLC
        ADC ALT2L,X
        STA BALO
        LDA BAHI
        ADC ALT2H,X
        STA BAHI
        LDA NUMBRS+1
        LSR A
        LSR A
        LSR A
        LSR A
        TAX
        LDA BALO
        CLC
        ADC ALT3L,X
        STA BALO
        LDA BAHI
        ADC ALT3H,X
        STA BAHI
        LDA NUMBRS
        AND #$0F
        TAX
        LDA BALO
        CLC
        ADC ALT4L,X
        STA BALO
        LDA BAHI
        ADC ALT4H,X
        STA BAHI

        LSR BAHI        ;DIVIDE ALT. BY
        ROR BALO        ;4
        LSR BAHI
        ROR BALO
        LDX GRDTYP      ;NOW DIVIDE BY
        LDA GRDDIV,X    ;GRID SCALING FACTOR
        STA DIVISR
        LDA BALO
        STA RESULT
        LDA BAHI
        JSR DIVIDE      ;DO THE DIVISION

;       ----------------------
;       CALC SHIP PLYR X COORD
;       ----------------------

        LDA SHIPI       ;0-111
        LSR A           ;DIVIDE BY 4
        LSR A           ;(NOW 0-27)
        STA SHTEMP      ;SAVE IT
        LDA SHIPY       ;0-111
        LSR A           ;/2 = 0-55
        CLC
        ADC SHTEMP      ;NOW 0-82
        CLC
        ADC #56         ;ADD LEFT LIMIT
        STA SHPLRX      ;GOT X POS. OF PLAYER!

;       ----------------------
;       CALC SHIP PLYR Y COORD
;       ----------------------

        LDA SHIPZ       ;0-111
        LSR A           ;NOW 0-55
        STA SHTEMP      ;SAVE IT
        LDA #201        ;BOTTOM OF GRID
        SEC
        SBC SHTEMP      ;NOW 146-201
        STA SHTEMP
        SEC             ;NOW WE MUST SUBTRACT
        SBC RESULT      ;SCALED ALTITUDE
        STA SHPLRY      ;DONE!

;       ------------------------------------
;       NOW ADJUST SHADOW HEIGHT FOR TERRAIN
;       ------------------------------------

        LDA SHIPY       ;0-111
        LSR A           ;0-55
        PHA             ;SAVE ON STACK
        AND #7          ;MASK TO 0-7
        STA SUBY
        LDA #7          ;NOW SUBTRACT
        SEC             ;THIS FROM 7
        SBC SUBY        ;TO FLIP IT AROUND
        STA SUBY
        PLA             ;GET SHIPY/2
        AND #$F8        ;MASK OFF TOP 5 BITS
        STA TERRIX      ;SAVE IN TERRAIN INDEX
        LDA SHIPZ       ;0-111
        LSR A           ;0-55
        PHA             ;SAVE
        AND #7          ;MASK TO 0-7
        STA SUBZ
        LDA #7          ;NOW SUBTRACT
        SEC             ;THIS FROM 7
        SBC SUBZ        ;TO FLIP IT AROUND
        STA SUBZ
        PLA             ;GET SHIPZ/2
        LSR A           ;DIVIDE BY 8
        LSR A           ;(/16 TOTAL)
        LSR A
        CLC
        ADC TERRIX      ;AND ADD TO
        STA TERRIX      ;TERRAIN INDEX

;       ---------------------
;       HIT VOLCANO ERUPTION?
;       ---------------------

        CMP VOLIX       ;IS SHIP OVER VOLCANO?
        BNE NOVHIT      ;NO
        LDA VOLPX       ;IS VOLCANO ACTIVE?
        BEQ NOVHIT      ;NO
        LDA NUMBRS      ;IS ALT>9999?
        BNE NOVHIT      ;YES, NO VOLCANO HIT
        LDA NUMBRS+1    ;IS ALT>499?
        CMP #5
        BCS NOVHIT      ;YES, SHIP SAFE
        LDA #1          ;UH-OH, WE'VE GOT
        STA VOLRAD      ;VOLCANO RADIATION!!!
        LDA #8          ;FORCE
        STA RADTIM      ;VOLCANO RADIATION

NOVHIT  LDX TERRIX      ;THIS ROUTINE GETS
        LDA HEIGHT,X    ;THE HEIGHTS OF THE
        STA T1          ;CORNERS OF THE
        LDA HEIGHT+1,X  ;TERRAIN WE'RE OVER
        STA T2          ;AND CALCULATES THE
        LDA SUBZ        ;HEIGHT OF THE
        STA MULT2       ;PARTICULAR POINT
        JSR TERCAL      ;UNDER OUR SHIP.
        STA T3          ;THIS ROUTINE IS VERY
        LDX TERRIX      ;COMPLICATED, AND
        LDA HEIGHT+8,X  ;SPACE DOES NOT PERMIT
        STA T1          ;IN-DEPTH DISCUSSION
        LDA HEIGHT+9,X  ;OF THE TECHNIQUE.
        STA T2
        LDA SUBZ
        STA MULT2
        JSR TERCAL
        STA T2
        LDA T3
        STA T1
        LDA SUBY
        STA MULT2
        JSR TERCAL
        STA T3
        LDA SHTEMP
        SEC
        SBC T3
        STA SHADOY      ;GOT SHADOW HEIGHT
        CMP SHPLRY      ;SHIP BELOW SHADOW?
        BCS NOMTHT      ;NO, SHIP'S AIRBORNE
        JMP CRASH       ;SPLAT!!!
NOMTHT  RTS             ;ALL DONE

;       ---------------------------
;       TERRAIN HEIGHT CALCULATOR
;       ---------------------------

TERCAL  LDA T1          ;THIS SUBROUTINE
        CMP T2          ;IS USED BY THE ABOVE
        BCC TC1         ;SHADOW HEIGHT
        SEC             ;ROUTINE TO FIND
        SBC T2          ;THE HEIGHT OF THE
        STA MULT1       ;GROUND AT A
        JSR MULT        ;SPECIFIED POINT.
        LDA LRES
        LSR A
        LSR A
        LSR A
        CLC
        ADC T2
        RTS
TC1     LDA T2
        SEC
        SBC T1
        STA MULT1
        LDA #8
        SEC
        SBC MULT2
        STA MULT2
        JSR MULT
        LDA LRES
        LSR A
        LSR A
        LSR A
        CLC
        ADC T1
        RTS

;       ------------
;       RANDOMIZE BASE
;       ------------

RNDBAS  LDX #5          ;6 COORDS
INBASE  LDA RANDOM      ;GET A NUMBER
        AND #7          ;FROM 0-6
        CMP #7          ;IS IT 7?
        BEQ INBASE      ;YES, DON'T ALLOW!
        STA BASEY,X     ;STORE IN Y/Z POSITION
        DEX
        BPL INBASE

        LDA BASEY+2     ;NOW CALCULATE BASE
        ASL A           ;GRID INDEX WITH THE
        ASL A           ;FORMULA:
        ASL A
        CLC             ;IX=BASEY*8+BASEZ
        ADC BASEZ+2
        STA BASEIX

        LDY #0          ;SHOW BASE...
        JSR SHOPOS      ;Y POSITION
        LDY #1          ;SHOW BASE...
        JSR SHOPOS      ;Z POSITION

        RTS             ;AND EXIT.

;       ------------------------------
;       MULTIPLY ROUTINE
;
;       THIS ROUTINE DOES AN UNSIGNED
;       MULTIPLY OF THE VALUES IN MULT1
;       AND MULT2, PLACING THE RESULT
;       IN HRES AND LRES.
;       ------------------------------

MULT    LDA #0
        STA HRES
        LDX #8
MSHIFT  ASL A
        ROL HRES
        ASL MULT1
        BCC MLOOP
        CLC
        ADC MULT2
        BCC MLOOP
        INC HRES
MLOOP   DEX
        BNE MSHIFT
        STA LRES
        RTS

;       ------------------------------
;       DIVIDE ROUTINE
;
;       THIS ROUTINE DIVIDES A 16-BIT
;       NUMBER (LO 8 BITS IN RESULT,
;       HI 8 BITS IN ACCUMULATOR) BY THE
;       VALUE IN DIVISR. THE QUOTIENT IS
;       PLACED IN RESULT.  THE REMAINDER
;       IS DISCARDED.
;       ------------------------------

DIVIDE  LDX #8
DIVL1   ASL RESULT
        ROL A
        CMP DIVISR
        BCC DIVL2
        SBC DIVISR
        INC RESULT
DIVL2   DEX
        BNE DIVL1
        RTS

;       ---------------
;       ADD TO VELOCITY
;       ---------------

ADDV    SED             ;WANT DECIMAL MODE
        LDA UNSNUM+1,X  ;ADD VELOCITY INCREMENT
        CLC             ;AMOUNT TO VELOCITY
        ADC VELADL      ;INDICATED BY
        STA UNSNUM+1,X  ;X REGISTER.
        STA NUMBRS+1,X
        LDA UNSNUM,X
        ADC VELADH
        STA UNSNUM,X
        STA NUMBRS,X
NEGMAN  CMP #$50        ;IS IT NEGATIVE?
        BCS NEGVEL      ;YES!
        LDA #0          ;SET SIGN TO 0 (+)
STOSGN  STA SIGNS,Y     ;STORE SIGN
        CLD             ;NO MORE DECIMAL
        RTS             ;AND EXIT!
NEGVEL  LDA #0          ;SUBTRACT UNSIGNED
        SEC             ;VELOCITY VALUE FROM
        SBC UNSNUM+1,X  ;ZERO AND STORE IN
        STA NUMBRS+1,X  ;SIGNED VALUE
        LDA #0
        SBC UNSNUM,X
        STA NUMBRS,X
        LDA #1          ;PUT 1 (-)...
        BNE STOSGN      ;IN SIGN

;       ------------------------------
;       SHOW SHIP/OBJECTIVE POSITIONS
;       ------------------------------

SHOPOS  LDA PYPOS,Y     ;GET Y POSITION
        STA CHARY       ;OF MESSAGE
        LDX POSNIX,Y    ;GET COORD INDEX
        LDY BASEY,X     ;GET COORD DIGIT 1
        LDA HI1,Y       ;CONVERT COORD
        STA POSNUM      ;TO DECIMAL
        LDA HI2,Y       ;AND PLACE IN
        STA POSNUM+1    ;WORK VARIABLE
        LDY BASEY+1,X   ;GET COORD DIGIT 2
        LDA MED1,Y      ;ADD DIGIT 2'S
        SED             ;DECIMAL VALUE
        CLC             ;TO WORK AREA
        ADC POSNUM+1
        STA POSNUM+1
        LDA POSNUM
        ADC #0
        STA POSNUM
```

```
        LDA BASEY+2,X    ;ADD DIGIT 3'S
        CLC              ;DECIMAL VALUE
        ADC POSNUM+1     ;TO WORK AREA
        STA POSNUM+1
        LDA POSNUM
        ADC #0
        STA POSNUM
        CLD
        LDA #2           ;2 DIGITS TO
        STA DIGITS       ;DISPLAY
        LDX #17          ;17TH IN "NUMBRS"
        JMP SHOALL       ;SHOW THE NUMBER
;       --------------
;       SHOW ALTITUDE
;       --------------
SHOALT  LDA #98          ;98TH SCREEN LINE
        STA CHARY
        LDA #3           ;3 DIGITS TO SHOW
        STA DIGITS
        LDX #0           ;0TH IN "NUMBRS"
        JMP SHOALL       ;SHOW THE NUMBER
;       ----------------
;       SHOW X VELOCITY
;       ----------------
SHOXV   LDA SIGNS        ;GET 1ST SIGN
        JSR SHOARR       ;SHOW ITS ARROW
        LDA #112         ;112TH SCREEN LINE
        STA CHARY
        LDA #2           ;2 DIGITS TO SHOW
        STA DIGITS
        LDX #3           ;3RD IN "NUMBRS"
        JMP SHOALL       ;SHOW IT
;       ----------------
;       SHOW Y VELOCITY
;       ----------------
SHOYV   LDA #2           ;GET 2ND SIGN
        CLC
        ADC SIGNS+1
        JSR SHOARR       ;SHOW ITS ARROW
        LDA #122         ;122ND SCREEN LINE
        STA CHARY
        LDA #2           ;2 DIGITS TO SHOW
        STA DIGITS
        LDX #6           ;6TH IN "NUMBRS"
        JMP SHOALL       ;SHOW IT
;       ----------------
;       SHOW Z VELOCITY
;       ----------------
SHOZV   LDA #4           ;GET 3RD SIGN
        CLC
        ADC SIGNS+2
        JSR SHOARR       ;SHOW ITS ARROW
        LDA #132         ;132ND SCREEN LINE
        STA CHARY
        LDA #2           ;2 DIGITS TO SHOW
        STA DIGITS
        LDX #8           ;8TH IN "NUMBRS"
        JMP SHOALL       ;SHOW IT
;       ---------
;       SHOW FUEL
;       ---------
SHOFUL  LDA #146         ;146TH SCREEN LINE
        STA CHARY
        LDA #3           ;3 DIGITS TO SHOW
        STA DIGITS
        LDX #10          ;10TH IN "NUMBRS"
        JMP SHOALL       ;SHOW IT
;       -----------
;       SHOW SHIPS
;       -----------
SHOSHP  LDA #160         ;160TH SCREEN LINE
        STA CHARY
        LDA #1           ;1 DIGIT TO SHOW
        STA DIGITS
        LDX #13          ;13TH IN "NUMBRS"
        JMP SHOALL       ;SHOW IT
;       -----------
;       SHOW SCORE
;       -----------
SHOSCO  LDA #170         ;170TH SCREEN LINE
        STA CHARY
        LDA #3           ;3 DIGITS TO SHOW
        STA DIGITS
        LDX #14          ;14TH IN "NUMBRS"
;       -------------
;       SHOW NUMBERS
;       -------------
SHOALL  LDA #31          ;31ST CHAR ACROSS
        STA CHARX
        STX SNUMIX       ;SAVE "NUMBRS" INDEX
SNUMLP  LDX SNUMIX       ;GET INDEX
        LDA NUMBRS,X     ;GET BCD VALUE
        JSR SHOBCD       ;SHOW IT
        INC SNUMIX       ;NEXT BCD VALUE
        DEC DIGITS       ;1 LESS DIGIT
        BNE SNUMLP       ;NOT FINISHED YET!
        RTS              ;ALL DONE
;       ----------------
;       BCD CHAR DISPLAY
;       ----------------
SHOBCD  STA SHOBYT       ;SAVE BCD VALUE
        LSR A            ;SHIFT RIGHT
        LSR A            ;TO GET
        LSR A            ;FIRST DIGIT
        LSR A
        ORA #$10         ;ADD CHAR. OFFSET
        STA CHARNO       ;SAVE FOR DISPLAY
        JSR SHONUM       ;AND DISPLAY IT
        INC CHARX        ;NEXT POS. ON SCREEN
        LDA SHOBYT       ;GET BCD BACK
        AND #$0F         ;GET SECOND DIGIT
        ORA #$10         ;ADD CHAR OFFSET
        STA CHARNO       ;SAVE FOR DISPLAY
        JSR SHONUM       ;AND DISPLAY IT
        INC CHARX        ;NEXT SCREEN POSITION
        RTS              ;AND EXIT
;       ------------
;       SHOW ARROWS
;       ------------
SHOARR  PHA              ;SAVE ARROW #
        LSR A            ;DIV BY 2
        TAX              ;USE FOR INDEX
        LDA ARRLO,X      ;GET SCREEN ADDRESS
        STA DESTLO       ;FOR THE ARROW
        LDA ARRHI,X      ;AND SAVE ON
        STA DESTHI       ;PAGE ZERO
        LDY #0           ;INITIALIZE COPY INDEX
        PLA              ;GET ARROW #
        ASL A            ;MULTIPLY BY 8 FOR
        ASL A            ;OFFSET INTO ARROW
        ASL A            ;GRAPHICS TABLE
        TAX              ;SAVE INDEX
        LDA #7           ;THERE ARE 8 BYTES
        STA DIGITS       ;IN EACH ARROW
COPARR  LDA ARROWS,X     ;GET GRAPHICS DATA,
        STA (DESTLO),Y   ;PUT ON SCREEN
        INX              ;NEXT ARROW BYTE
        TYA              ;ADD 40 TO Y REGISTER
        CLC              ;TO POINT TO NEXT
        ADC #40          ;GRAPHICS 8 LINE
        TAY              ;(40 BYTES PER LINE)
        DEC DIGITS       ;MORE ARROW DATA?
        BNE COPARR       ;YES, KEEP COPYING
        RTS              ;ALL DONE
;       --------------
;       CLEAR P/M AREA
;       --------------
PMCLR   LDA #0           ;PUT ZEROS IN
        TAY              ;ALL 256 POSITIONS
PMCLP   STA MISSLS,Y     ;OF MISSILES,
        STA PL0,Y        ;PLAYER 0
        STA PL1,Y        ;PLAYER 1
        STA PL2,Y        ;PLAYER 2
        STA PL3,Y        ;PLAYER 3
        DEY              ;DONE?
        BNE PMCLP        ;NOT YET
        RTS              ;EXIT
;       -------------
;       CLEAR SCREEN
;       -------------
CLSCRN  LDX #29          ;CLEAR 30*256 BYTES
        LDA #DISP/256    ;PUT DISPLAY
        STA HI           ;ADDRESS IN
        LDA #0           ;PAGE ZERO
        STA LO           ;WORK AREA
CLSLP1  TAY              ;ZERO Y REGISTER
CLSLP2  STA (LO),Y       ;CLEAR A BYTE
        DEY              ;DONE WITH 256?
        BNE CLSLP2       ;NOPE!
        DEX              ;DONE WITH 30?
        BNE NXTCLR       ;NOPE!
        RTS              ;ALL DONE!
NXTCLR  INC HI           ;NEXT 256-BYTE GROUP
        BNE CLSLP1       ;LOOP BACK
;       -------------------------
;       CLEAR GRID AREA OF SCREEN
;       -------------------------
CLGRDA  LDA #0           ;POINT TO 100TH LINE
        STA SHPLRX       ;OF SCREEN
        LDA #[DISP+2+100*40]/256
        STA HI
        LDA #[DISP+2+100*40]&255
        STA LO
        LDX #72          ;CLEAR 72 GROUPS
CGRDL1  LDY #21          ;OF 22 BYTES
        LDA #0
CGRDL2  STA (LO),Y       ;CLEAR A BYTE
        DEY              ;DONE WITH 22?
        BPL CGRDL2       ;NOT YET!
        DEX              ;DONE 72 GROUPS?
        BNE NXTCGD       ;NOPE!
        RTS              ;ALL DONE!
NXTCGD  LDA LO           ;POINT TO NEXT LINE
        CLC              ;OF SCREEN
        ADC #40          ;(40 BYTES IN GR.8)
        STA LO
        BCC CGRDL1
        INC HI
        BNE CGRDL1
;       -------------------
;       CLEAR MESSAGE AREA
;       -------------------
;       POINT TO 14TH LINE OF SCREEN
CLMESS  LDA #[DISP+2+14*40]/256
        STA HI
        LDA #[DISP+2+14*40]&255
        STA LO
        LDX #8           ;CLEAR 8 GROUPS
CMESS1  LDY #21          ;OF 22 BYTES
        LDA #0
CMESS2  STA (LO),Y       ;CLEAR A BYTE
        DEY              ;DONE 22 BYTES?
        BPL CMESS2       ;NOT YET!
        DEX              ;DONE 8 GROUPS?
        BNE NXTCMS       ;NOPE!
        RTS              ;ALL DONE
NXTCMS  LDA LO           ;ADVANCE 40 BYTES
        CLC              ;TO NEXT SCREEN LINE
        ADC #40
        STA LO
        BCC CMESS1
        INC HI
        BNE CMESS1
;       -------------
;       SHOW MESSAGE
;       -------------
SHOMSG  STY CHARY        ;SET SCREEN LINE
        LDY MSGIX,X      ;POINT TO MESSAGE TEXT
        LDA #2           ;2ND CHAR ON SCREEN
        STA CHARX
        LDX #22          ;22 CHARACTERS TO SHOW
        STX MSGCT
        STY MSGPTR
MSGLP   LDX MSGCT        ;GET COUNT IN MESSAGE
        LDY MSGPTR       ;POINT TO MESSAGE CHAR
        LDA MESSGE,Y     ;GET CHARACTER
        STA CHARNO       ;GET READY TO SHOW IT
        JSR SHONUM       ;SHOW IT!
        DEC MSGCT        ;LAST CHARACTER?
        BNE NXTMBT       ;NOT YET!
        RTS              ;ALL DONE
NXTMBT  INC CHARX        ;NEXT X POSITION
        INC MSGPTR       ;NEXT CHAR
        BNE MSGLP        ;LOOP BACK
;       ---------------------------
;       GRAPHICS 8 CHAR. GENERATOR
;       ---------------------------
SHONUM  LDA CHARY
        STA LO           ;SAVE Y POS
        LDA #0
        STA HI
        ASL LO           ;*2
        ROL HI
        ASL LO           ;*4
        ROL HI
        ASL LO           ;*8
        ROL HI
        LDA LO
        CLC
        ADC #DISP&255    ;ADD DISPLAY START
        STA DESTLO
        LDA HI
        ADC #DISP/256
        STA DESTHI
        ASL LO           ;*16
        ROL HI
        ASL LO           ;*32
        ROL HI
        LDA LO
        CLC
        ADC DESTLO       ;GET CHARACTER'S
        STA DESTLO       ;SCREEN ADDRESS
        LDA HI
        ADC DESTHI
        STA DESTHI
        LDA DESTLO       ;ADD X OFFSET
        CLC              ;FOR FINAL ADDRESS
        ADC CHARX
        STA DESTLO
        BCC NODHIN
        INC DESTHI
NODHIN  LDA CHARNO       ;GET OFFSET INTO
        STA LO           ;CHARACTER SET
        LDA #0
        STA HI
        STA COPCNT       ;ZERO COPY COUNT
        ASL LO           ;*2
        ROL HI
        ASL LO           ;*4
        ROL HI
```

```
        ASL LO          ;*8
        ROL HI
        LDA HI          ;GET INFORMATION
        CLC             ;FROM O.S. CHARACTER
        ADC #$E0        ;SET AT $E000
        STA HI
COPNUM  LDY COPCNT
        LDA (LO),Y      ;GET CHAR. IMAGE BYTE
        LDY #0
        STA (DESTLO),Y  ;PUT ON SCREEN
        INC COPCNT      ;NEXT CHARACTER BYTE
        LDA COPCNT      ;ARE ALL 8 BYTES
        CMP #8          ;IN CHARACTER MOVED?
        BEQ FINISH      ;YES!
        LDA DESTLO      ;ADD 40 BYTES TO
        CLC             ;POINT TO NEXT LINE
        ADC #40         ;ON SCREEN
        STA DESTLO
        BCC COPNUM
        INC DESTHI
        BNE COPNUM
FINISH  RTS             ;ALL DONE!

;       ------------------------
;       RANDOMIZE GRID HEIGHTS
;       ------------------------

RNDGRD  LDX #63         ;64 POINTS ON GRID
        LDY GRDTYP      ;GET GRID TYPE
RGLP    LDA RANDOM      ;GET RANDOM NUMBER BUT
        AND #$0F        ;LIMIT IT TO THE
        CMP MAXHGT,Y    ;MAXIMUM HEIGHT
        BCS RGLP        ;OF THE GRID TYPE
        STA HEIGHT,X    ;SAVE HEIGHT
        DEX             ;DONE?
        BPL RGLP        ;NO!
        RTS             ;ALL DONE, EXIT

;       --------------
;       DRAW THE GRID
;       --------------

GRID    CLD
        LDA #0
        STA VOLPX       ;NO VOLCANO
        LDX #0          ;CLEAR OUT
        TXA             ;PLAYER 2
CP2LP   STA PL2,X
        DEX
        BNE CP2LP
        LDX GRDTYP      ;GRID TYPE 0 (HIGH)?
        BEQ GOTBAS      ;YES, SHOW BASE
        LDX #0          ;COMPARE SHIP'S
BASCHK  LDA GYP,X       ;GRID POSITIONS
        CMP BASEY,X     ;TO BASE'S.
        BNE NOBASE
        LDA GZP,X
        CMP BASEZ,X     ;IF THEY'RE NOT EQUAL,
        BNE NOBASE      ;DON'T SHOW BASE!
        INX
        CPX GRDTYP
        BNE BASCHK
GOTBAS  LDX GRDTYP

;       ------------------
;       CALC BASE X COORD
;       ------------------

        LDA BASEZ,X     ;THIS ROUTINE IS
        ASL A           ;SIMILAR TO THE
        ASL A           ;SHIP PLAYER POSITION
        STA SHTEMP      ;CALCULATOR ABOVE.
        LDA BASEY,X     ;IT FINDS THE VOLCANO'S
        ASL A           ;P/M COORDINATES.
        ASL A
        ASL A
        CLC
        ADC SHTEMP
        CLC
        ADC #61
        STA HPOSP2

;       ------------------
;       CALC BASE Y COORD
;       ------------------

        LDA BASEZ,X
        ASL A
        ASL A
        ASL A
        STA SHTEMP
        LDA #196
        SEC
        SBC SHTEMP
        TAY

;       ----------------
;       COPY BASE SYMBOL
;       ----------------

        LDX #4          ;5 BYTES IN BASE PIC.
BSCOPY  LDA BASPIC,X    ;GET BASE PICTURE BYTE
        STA PL2,Y       ;PUT IN PLAYER 2
        INY             ;NEXT PLAYER BYTE
        DEX             ;NEXT IMAGE BYTE
        BPL BSCOPY      ;COPY ALL 5 BYTES!

;       ------------------------
;       MAKE BASE ZERO HEIGHT IF
;       WE'RE ON FINAL APPROACH
;       ------------------------

        LDX GRDTYP      ;IF GRID IS NOT
        CPX #2          ;TYPE 2 (LOW)...
        BNE NOBASE      ;LEAVE TERRAIN ALONE!
        LDX BASEIX      ;OTHERWISE WE MUST
        LDA #0          ;MAKE THE 4 CORNERS
NEWHGT  STA HEIGHT,X    ;OF THE LANDING GRID
        STA HEIGHT+1,X  ;ALL ZERO HEIGHT!
        STA HEIGHT+8,X
        STA HEIGHT+9,X
        JMP NOVOLC      ;AND BYPASS NEXT CODE.

;       ------------------------------
;       IF THERE'S NO BASE, LET'S TRY TO
;       PLACE A DANGEROUS VOLCANO!
;       ------------------------------

NOBASE  LDA GRDTYP      ;ARE WE AT LOW ALT...
        CMP #2          ;(GRID TYPE 2)?
        BNE NOVOLC      ;NO, NO VOLCANO!
        LDA RANDOM      ;GET RANDOM #
        CMP VCHANC      ;IS IT > VOLCANO CHANCE?
        BCS NOVOLC      ;YES, NO VOLCANO!
VOLCZ   LDA RANDOM      ;GET RANDOM VOLCANO
        AND #7          ;Y AND Z COORDINATES
        CMP #5          ;FROM 1-5
        BCS VOLCZ
        CLC
        ADC #1
        STA VOLZ
VOLCY   LDA RANDOM
        AND #7
        CMP #5
        BCS VOLCY
        CLC
        ADC #1
        STA VOLY        ;NOW SET VOLCANO
        ASL A           ;POSITION INDEX
        ASL A
        ASL A
        CLC
        ADC VOLZ
        STA VOLIX

;       -------------------------
;       CALC VOLCANO PLAYER COORDS
;       -------------------------

        LDA VOLZ        ;THIS ROUTINE IS
        ASL A           ;SIMILAR TO THE BASE
        ASL A           ;PLAYER POSITION
        STA SHTEMP      ;ROUTINE.  IT USES
        LDA VOLY        ;VOLY & VOLZ TO
        ASL A           ;CALCULATE THE
        ASL A           ;VOLCANO'S PLAYER
        ASL A           ;COORDINATES.
        CLC
        ADC SHTEMP
        CLC
        ADC #59
        STA VOLPX

        LDA VOLZ
        ASL A
        ASL A
        ASL A
        STA SHTEMP
        LDA #184
        SEC
        SBC SHTEMP
        STA VOLPY

        LDX VOLIX       ;NOW SET THE 4 CORNERS
        LDA #7          ;OF THE VOLCANO TO
        JMP NEWHGT      ;HEIGHT 7.

NOVOLC  LDA #1          ;SELECT PLOT COLOR
        STA COLOR
        LDA #16         ;SET PLOT X OFFSET
        STA XOFSET

;       ------------------------------
;       THE FOLLOWING CODE TAKES THE
;       HEIGHT VALUES FROM THE "HEIGHT"
;       TABLE AND PLOTS THE GRID ON THE
;       SCREEN.
;       ------------------------------

        LDA #0
        STA PLXHI
        STA DRXHI
        STA HGTIX
        STA X
NEXTX1  LDA #0          ;ZERO Y INDEX
        STA Y
        TAY             ;PLOT FLAG
NEXTY1  LDA XOFSET      ;GET X POSITION
        CLC
        ADC X
        STA PLOTX,Y     ;STORE IN PLOTX/DRAWX
        LDA #170        ;GET Y POSITION
        SEC
        SBC Y
        SEC
        LDX HGTIX
        SBC HEIGHT,X    ;SUBTRACT GRID HEIGHT
        STA PLOTY,Y     ;STORE IN PLOTY/DRAWY
        LDA Y           ;DRAWING?
        BNE DRWIT1      ;YES!
        JSR PLOTCL      ;PLOT THE POINT
        JMP ENDRW1      ;DO NEXT POINT
DRWIT1  JSR DRAW        ;DRAW TO THE POINT
ENDRW1  LDA XOFSET      ;ADD 8 TO X OFFSET
        CLC
        ADC #8
        STA XOFSET
        INC HGTIX       ;NEXT HEIGHT VALUE
        LDY #1          ;NOW DRAW
        LDA Y           ;ADD 8 TO Y INDEX
        CLC
        ADC #8
        STA Y
        CMP #57         ;PAST LIMIT?
        BCC NEXTY1      ;NO, LOOP BACK.
        LDA #16         ;RESET X OFFSET
        STA XOFSET
        LDA X           ;AND INCREMENT X
        CLC             ;FOR NEXT GRID LINE
        ADC #16
        STA X
        CMP #113        ;PAST LIMIT?
        BCC NEXTX1      ;NO, LOOP BACK.

;       ------------------------------
;       NOW DO OTHER DIRECTION OF GRID
;       ------------------------------

        LDA #0          ;ZERO OUT Y
        STA Y
        LDA #$FF        ;SET INDEX TO -1
        STA INDEX
NEXTY2  INC INDEX       ;NEXT INDEX
        LDA INDEX
        STA HGTIX       ;SAVE IN HEIGHT INDEX
        LDA #0          ;ZERO OUT X
        STA X
        TAY             ;SET Y REG FOR PLOT
NEXTX2  LDA XOFSET      ;FIND X POSITION
        CLC
        ADC X
        STA PLOTX,Y     ;STORE IN PLOTX/DRAWX
        LDA #170        ;GET Y POSITION
        SEC
        SBC Y
        LDX HGTIX
        SEC
        SBC HEIGHT,X    ;ADJUST FOR HEIGHT
        STA PLOTY,Y     ;STORE IN PLOTY/DRAWY
        LDA X           ;PLOTTING?
        BNE DRWIT2      ;NO, DRAW IT.
        JSR PLOTCL      ;PLOT THE POINT
        JMP ENDRW2      ;GO DO NEXT POINT
DRWIT2  JSR DRAW        ;DRAW TO THE POINT
ENDRW2  LDA HGTIX       ;NEXT HEIGHT VALUE
        CLC
        ADC #8
        STA HGTIX
        LDY #1          ;NOW DRAWING
        LDA X           ;NEXT X POSITION
        CLC
        ADC #16
        STA X
        CMP #113        ;PAST LIMIT?
        BCC NEXTX2      ;NO
        LDA XOFSET      ;NEXT X OFFSET VALUE
        CLC
        ADC #8
        STA XOFSET
        LDA Y           ;NEXT Y VALUE
        CLC
        ADC #8
        STA Y
        CMP #57         ;PAST LIMIT?
        BCC NEXTY2      ;NO, LOOP
        RTS             ;ALL DONE.

;       ---------------
;       VERTICAL BLANK
;       ---------------

VBI     CLD             ;NO DECIMAL MODE!
        LDA #0          ;RESET DLI INDEX
        STA DLICNT

;       ---------------
;       CHECK FOR PAUSE
;       ---------------

        LDA KEY         ;GET KEYBOARD KET
        CMP #$21        ;SPACE BAR?
        BNE PCHK        ;NO
        LDA PAUSED      ;FLIP THE
        EOR #$FF        ;PAUSE STATUS
        STA PAUSED
        LDA #$FF        ;RESET KEYPRESS
        STA KEY
PCHK    LDA PAUSED      ;WE PAUSED?
        BEQ NOPAU       ;NO
        JMP XITVBV      ;PAUSED, NO PROCESSING!
NOPAU   LDA COLPM3      ;CYCLE
```

HAVE YOU FLOWN YOUR ATARI TODAY?

Imagine yourself at the controls of a small, single-engine plane, 10,000 feet in the air, on your approach to the runway and safety. You're running low on fuel, but your instruments show that you're on the glide path, and lined up with the runway. It's a beautiful, sunny day, and you can see the airport in the distance, across the grassy fields. But the crosswind is tricky, and it will take all your skill to land safely. You're coming down now, and the runway is getting closer. A bit left, OK, now lower the power, fine, now put down the flaps. Pull the nose up a bit more, you're a little low. Watch the power! Don't stall. OK. Here comes the runway. You hear the squeal of tires on pavement, your pulse quickens, you're down, but watch it, you're pulling right! Brakes, brakes! Left more! You've stopped safely! Good job. The first real-time flight simulator for ATARI is now available from MMG Micro Software. Written entirely in machine language, there are four levels of difficulty, landings in clear or foggy weather, landings with or without instruments, and with or without the real-time view from the cockpit. **Final Flight!** requires Atari 400/800, 24K, 1 joy stick, and is offered on tape or disk for the same suggested retail price of $29.95.

Final Flight! is available at your local dealer or direct from **MMG Micro Software.** Just send check or money order to P.O. Box 131, Marlboro, N.J. 07746 or for Mastercard, Visa, and C.O.D. deliveries call **(201)431-3472.** Please add $3.00 for postage and handling. New Jersey residents add 6% sales tax.

WHEN EAGLES FIRST RULED THE SKIES...

EAGLES™ flies you over France during World War I to join the men who first rose to the skies like some glorious birds of prey. This realistic strategy simulation gives you fourteen historic planes — including the feared Fokker DR-1 Triplane (the Red Baron's favorite) and the legendary Sopwith Camel — each accurately rated for speed, rate of climb and maneuverability. If you like, you can even fly planes of your own design. You can play solitaire against enemy planes flown by the computer, or against other players. Fly balloon-busting missions, set up duels between single aircraft, or create grand dogfights involving as many as twenty planes. Choose historic scenarios, or let the computer set up the battles.

So head on down to your nearest computer/software or game dealer and get your wings today!

FOR APPLE®, ATARI® & COMMODORE 64®

On 48K mini floppy disk for Apple® II with Applesoft ROM Card, II+, IIe & Apple III. On 40K disk for Atari® 400/800/1200 with BASIC cartridge. On 64K disk for the Commodore 64®. **$39.95.**

SSI

STRATEGIC SIMULATIONS INC

If there are no convenient stores near you, VISA & Mastercard holders can order direct by calling 800-227-1617 ext. 335 (toll free). In California, call 800-772-3545, ext. 335. To order by mail, send your check to: Strategic Simulations Inc, 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. (California residents, add 6.5% sales tax.) Please specify computer format.

WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.

```
        CLC             ;RADIATION
        ADC #1          ;COLOR
        AND #$07        ;BRIGHTNESS
        ORA #$30
        STA COLPM3
        LDA TIMER       ;HANDLE
        BEQ NOTDEC      ;GENERAL TIMER
        DEC TIMER
NOTDEC  LDA RADTIM      ;HANDLE
        BEQ NORDEC      ;RADIATION TIMER
        DEC RADTIM
NORDEC  LDA VOLPX       ;VOLCANO?
        BEQ FLSHP2      ;NOPE
        STA HPOSP2      ;POSITION PLAYER 2
        LDA RANDOM      ;DO RANDOM
        AND RANDOM      ;ERUPTION GRAPHICS
        LDY VOLPY
        STA PL2,Y
        STA PL2+4,Y
        LDA RANDOM
        AND RANDOM
        STA PL2+2,Y
        LDA RANDOM
        AND #$3C
        STA PL2+6,Y
        LDA #$3A
        STA COLPM2
        BNE NOPFLS      ;NO PAD FLASH
FLSHP2  INC PFLASH      ;INCREMENT
        LDA PFLASH      ;PAD FLASH TIMER
        AND #$0F
        BNE NOPFLS      ;NO FLASH YET
        INC PFLIX       ;FLIP FLASH COLOR INDEX
        LDA PFLIX
        AND #1
        TAX             ;USE FOR INDEX
        LDA PCOLRS,X    ;GET PAD COLOR
        STA COLPM2      ;PUT IN PLAYER 2 COLOR
NOPFLS  LDX LASTSY      ;NOW WE ERASE
        LDA #$FC        ;THE OLD SHIP...
        AND MISSLS,X
        STA MISSLS,X
        LDA #$FC
        AND MISSLS+1,X
        STA MISSLS+1,X
        LDX LSHADY      ;AND SHADOW
        LDA #$F3
        AND MISSLS,X
        STA MISSLS,X
        LDA #$F3
        AND MISSLS+1,X
        STA MISSLS+1,X
        LDA SHPLRX      ;POSITION
        STA HPOSM0      ;THE SHIP
        STA HPOSM1      ;AND SHADOW
        LDX SHPLRY
        STX LASTSY      ;SAVE POSITION,
        LDA #$02        ;DRAW SHIP...
        ORA MISSLS,X
        STA MISSLS,X
        LDA #$02
        ORA MISSLS+1,X
        STA MISSLS+1,X
        LDX SHADOY      ;AND SHADOW
        STX LSHADY
        LDA #$08
        ORA MISSLS,X
        STA MISSLS,X
        LDA #$08
        ORA MISSLS+1,X
        STA MISSLS+1,X

        LDA #130        ;SET UP THRUST
        STA AUDF1       ;FREQUENCY 1
        LDA #0
        STA AUDF2       ;AND 2
        LDA THRVOL      ;GET THRUST VOLUME
        STA AUDC1       ;STORE IN AUDIO 1
        LSR A           ;DIVIDE BY 8
        LSR A
        LSR A
        STA AUDC2       ;AND PUT IN AUDIO 2
        INC THRTIM      ;TIME TO DECREMENT
        LDA THRTIM      ;THRUST VOLUME?
        AND #7
        BNE DOTEMP      ;NOT YET
        LDA THRVOL      ;IS VOLUME 0?
        BEQ DOTEMP      ;YES, DON'T CHANGE!
        DEC THRVOL      ;LESS VOLUME!

DOTEMP  LDX STICK       ;GET STICK
        LDA ENGIX,X     ;AND ENGINE NUMBER
        STA STKPOS      ;IS ENGINE BEING USED?
        BEQ COOLIT      ;NO, COOL ALL ENGINES
        TAX
        DEX
        LDA TEMP2,X     ;GET ENGINE TEMP
        CMP #$10        ;IS ENGINE DEAD?
        BCS COOLIT      ;YES, DON'T HEAT MORE!
        LDA TEMP1,X     ;INCREMENT THE
        CLC             ;TEMPERATURE.
        ADC #1
        STA TEMP1,X
        LDA TEMP2,X
        ADC #0
        STA TEMP2,X
        JSR SETCLR      ;GO DO ENGINE COLORS

COOLIT  LDX #2          ;COOL ALL 3 ENGINES
        DEC STKPOS
COOLP   CPX STKPOS      ;IS THIS ENGINE IN USE?
        BEQ NXCOOL      ;IF SO, CON'T COOL IT!
        LDA TEMP2,X     ;GET ENGINE TEMP
        CMP #$10        ;IS IT DEAD?
        BCS NXCOOL      ;YES, DON'T COOL IT.
        CMP #0          ;IS ENGINE HOT?
        BNE COOLOK      ;YES, OK TO COOL
        LDA TEMP1,X     ;GET LOW TEMP BYTE
        CMP #2          ;IS ENGINE COLD?
        BCC NXCOOL      ;YES, DON'T COOL
COOLOK  LDA TEMP1,X     ;COOL OFF THE
        SEC             ;ENGINE BY
        SBC #2          ;SUBTRACTING 2
        STA TEMP1,X     ;FROM THE TEMP
        LDA TEMP2,X
        SBC #0
        STA TEMP2,X
        JSR SETCLR      ;SHOW ENGINE COLOR
NXCOOL  DEX             ;MORE ENGINES?
        BPL COOLP       ;YUP!
VBDONE  LDA SNDCNT      ;GET WARNING TONE COUNT
        BEQ NOSND       ;IF 0, NO TONE.
        SEC             ;OTHERWISE
        SBC #1          ;SUBTRACT 1 FROM
        STA SNDCNT      ;THE COUNT,
        LSR A           ;DIVIDE BY 2
        ORA #$A0        ;AND MAKE IT THE
        STA AUDC3       ;VOLUME.
NOSND   JMP XITVBV      ;VBI DONE!

;       ------------------
;       SET ENGINE COLORS
;       ------------------

SETCLR  AND #$1F        ;MASK OFF HI TEMP,
        LSR A           ;DIVIDE BY 8
        LSR A
        LSR A
        STA TEMP8,X     ;GET TEMP COLOR INDEX
        TAY
        LDA ENBCLS,Y    ;GET COLOR
        CMP ECOLOR+2,X  ;SAME AS CURRENT?
        BEQ NOCCHG      ;YES, NO CHANGE
        STA ECOLOR+2,X  ;NEW COLOR:
        LDA #20         ;SET UP ENGINE TEMP
        STA SNDCNT      ;TONE COUNT
        ASL A           ;AND FREQUENCY
        STA AUDF3       ;OF 40.
NOCCHG  RTS             ;EXIT

;       --------------------------------
;       GR. 8+ PLOTTER ROUTINE
;
;       THIS ROUTINE PLOTS IN GRAPHICS 8
;       IN 4 COLORS, DETERMINED BY 0-3
;       IN "COLOR".
;       --------------------------------

PLOTCL  JSR PCALC
        LDY COLOR       ;GET COLOR
        LDA BMASK2,X    ;AND MASK OFF
        AND COLORS,Y    ;PIXEL POSITION
        STA HOLD        ;SAVE IT.
        LDA BMASK1,X    ;MASK OFF PIXEL
        LDY #0          ;OF THE ADDRESS
        AND (LO),Y      ;TO BE ALTERED
        ORA HOLD        ;SET THE PLOT
        STA (LO),Y      ;BITS AND STORE!
        RTS             ;FINIS!

;       ---------------
;       PLOT CALCULATOR
;       ---------------

PCALC   LDA PLOTY
        ASL A
        STA LO
        LDA #0
        ROL A
        STA HI          ;*2
        ASL LO          ;*4
        ROL HI
        ASL LO
        LDA LO
        STA LOHLD
        ROL HI
        LDA HI
        STA HIHLD       ;*8
        ASL LO
        ROL HI          ;*16
        ASL LO
        ROL HI          ;*32
        LDA LO
        CLC
        ADC LOHLD
        STA LO
        LDA HI
        ADC HIHLD
        STA HI          ;+*8=*40
        LDA #DISP&255   ;ADD THE DISPLAY
        CLC             ;ADDRESS TO GET
        ADC LO          ;THE ACTUAL
        STA LO          ;ADDRESS OF THE
        LDA #DISP/256   ;BYTE THAT WILL
        ADC HI          ;BE ALTERED FOR
        STA HI          ;THE PLOT.
        LDA PLOTX       ;MASK PLOTX FOR
        AND #7          ;PLOT INDEX,
        LSR A
        TAX             ;PLACE IN X.
        LDA PLXHI
        LSR A
        LDA PLOTX       ;GET PLOTX AND
        ROR A
        LSR A           ;DIVIDE
        LSR A           ;BY 8,
        CLC             ;ADD TO
        ADC LO          ;PLOT ADDRESS
        STA LO          ;FOR FINAL PLOT
        LDA HI          ;ADDRESS.
        ADC #0
        STA HI
        RTS             ;EXIT!

;       --------------------------------
;       GRAPHICS 8+ DRAW HANDLER
;
;       THIS ROUTINE IS A MODIFICATION
;       OF THE GRAPHICS 7+ HANDLER FROM
;       A.N.A.L.O.G. #11.
;       --------------------------------

DRAW    LDA DRAWY       ;GET DRAWTO Y
        CMP PLOTY       ;IS DRAWY>PLOTY?
        BCC YMINUS      ;NO!
        SEC             ;SUBTRACT
        SBC PLOTY       ;PLOTY FROM DRAWY
        STA DELTAY      ;AND SAVE DIFFERENCE
        LDA #1          ;Y INCREMENT
        STA INCY        ;=1 (DOWN)
        BNE XVEC        ;NOW DO X
YMINUS  LDA PLOTY       ;SUBTRACT
        SEC             ;DRAWY FROM PLOTY
        SBC DRAWY
        STA DELTAY      ;AND SAVE DIFFERENCE
        LDA #255        ;Y INCREMENT
        STA INCY        ;=-1 (UP)
XVEC    LDA DRXHI       ;IS DRAWX
        CMP PLXHI       ;> PLOTX?
        BCC XMINUS      ;NO!
        BNE NOCMP       ;MAYBE...
        LDA DRAWX
        CMP PLOTX
        BCC XMINUS      ;NO!
NOCMP   LDA DRAWX       ;SUBTRACT PLOTX
        SEC             ;FROM DRAWX
        SBC PLOTX
        STA DELTAX      ;AND SAVE DIFFERENCE
        LDA DRXHI
        SBC PLXHI
        STA DELXHI      ;SAVE HI BYTE TOO
        LDA #1          ;X INCREMENT
        STA INCX        ;=1 (RIGHT)
        BNE VECSET      ;BRANCH!
XMINUS  LDA PLOTX       ;SUBTRACT
        SEC             ;DRAWX FROM PLOTX
        SBC DRAWX
        STA DELTAX      ;AND SAVE DIFFERENCE
        LDA PLXHI
        SBC DRXHI
        STA DELXHI      ;SAVE HI BYTE TOO
        LDA #255        ;X INCREMENT
        STA INCX        ;=-1 (LEFT)
VECSET  LDA #0          ;ZERO OUT
        STA ACCY        ;ALL ACCUMULATORS
        STA ACCX
        STA ACCXHI
        STA ACCYHI
        LDA DELXHI      ;IS DELTAX>DELTAY?
        BNE XMAX        ;YES!
        LDA DELTAX      ;MAYBE...
        CMP DELTAY
        BCC YMAX        ;NO, Y IS LARGER
XMAX    LDA DELXHI      ;SET UP DRAW
        STA CNTHI       ;COUNTER AND
        STA ENDHI       ;ENDPOINT
        LDA DELTAX
        STA COUNTR
        STA ENDPT
        LDA ENDHI       ;DIVIDE ENDPOINT
        LSR A           ;BY 2 FOR
        LDA ENDPT       ;Y ACCUMULATOR
        ROR A
        STA ACCY
        JMP DRAW80      ;START DRAWING
YMAX    LDA #0          ;DELTA Y LARGER,
        STA CNTHI       ;STORE IT IN
        STA ENDHI       ;THE DRAW COUNTER
        LDA DELTAY      ;AND ENDPOINT
        STA COUNTR
        STA ENDPT
        LSR A           ;DIVIDE BY 2 AND
        STA ACCX        ;SET X ACCUMULATOR
DRAW80  LDA COUNTR      ;IF COUNTR>0...
        ORA CNTHI
        BNE BEGIN       ;WE DRAW!
        RTS             ;OTHERWISE, EXIT
BEGIN   LDA ACCY        ;ADD DELTA Y
        CLC             ;TO Y ACCUMULATOR
        ADC DELTAY
        STA ACCY
        LDA ACCYHI
        ADC #0
```

```
        STA ACCYHI
        CMP ENDHI           ;AT ENDPOINT?
        BCC BEGIN2          ;NO, DO X
        BNE ADJY
        LDA ACCY            ;CHECK LOW BYTE
        CMP ENDPT
        BCC BEGIN2          ;DO X
ADJY    LDA ACCY            ;SUBTRACT ENDPOINT
        SEC                 ;FROM Y ACCUMULATOR
        SBC ENDPT
        STA ACCY
        LDA ACCYHI
        SBC ENDHI
        STA ACCYHI
        LDA PLOTY           ;INCREMENT Y
        CLC                 ;POSITION
        ADC INCY
        STA PLOTY
BEGIN2  LDA ACCX            ;ADD DELTA X TO
        CLC                 ;X ACCUMULATOR
        ADC DELTAX
        STA ACCX
        LDA ACCXHI
        ADC DELXHI
        STA ACCXHI
        CMP ENDHI           ;AT ENDPOINT?
        BCC PLOTIT          ;NO, GO PLOT!
        BNE ADJX
        LDA ACCX            ;CHECK LOW BYTE
        CMP ENDPT
        BCC PLOTIT          ;GO PLOT
ADJX    LDA ACCX            ;SUBTRACT ENDPOINT
        SEC                 ;FROM X ACCUMULATOR
        SBC ENDPT
        STA ACCX
        LDA ACCXHI
        SBC ENDHI
        STA ACCXHI
        LDA INCX            ;WHICH DIRECTION?
        BMI SUBX            ;GOING LEFT
        BPL ADDX            ;GOING RIGHT
BBEGIN  BNE BEGIN           ;LEAPFROG JUMP
ADDX    LDA PLOTX           ;ADD INCREMENT
        CLC                 ;TO PLOTX
        ADC INCX
        STA PLOTX
        LDA PLXHI
        ADC #0
        STA PLXHI
        JMP PLOTIT          ;GO DO THE PLOT
SUBX    LDA PLOTX           ;SUBTRACT 1
        SEC                 ;FROM PLOTX
        SBC #1
        STA PLOTX
        LDA PLXHI
        SBC #0
        STA PLXHI
PLOTIT  JSR PLOTCL          ;PLOT THE POINT
        LDA COUNTR          ;ONE LESS PIXEL
        SEC                 ;TO PLOT
        SBC #1
        STA COUNTR
        LDA CNTHI
        SBC #0
        STA CNTHI
        ORA COUNTR          ;DONE PLOTTING?
        BNE BBEGIN          ;NOT YET!
        RTS                 ;FINIS!

;       ---------
;       GRID DATA
;       ---------

MAXHGT  DB  1,4,8           ;MAXIMUM GRID HEIGHTS

GRDLIM  DB  $00,$80,$10     ;ALTITUDE LIMITS (X100')

;       ------------
;       PLOT TABLES
;       ------------

COLORS  DB  $00,$55,$AA,$FF
BMASK1  DB  $3F,$CF,$F3,$FC
BMASK2  DB  $C0,$30,$0C,$03

;       -----------------
;       BORDER PLOT DATA
;       -----------------

LINTYP  DB  0,1,1,1,1,0,1,0 ;0=PLOT, 1=DRAW
        DB  1,0,1,0,1,0,1,1
XHIGH   DB  0,1,1,0,0,0,0,0
        DB  1,1,0,0,1,1,1,1
XLOW    DB  10,54,54,10,10,192,192   ;COORDS
        DB  192,54,54,192,192,32,32
        DB  32
YLOW    DB  10,10,180,180,10,10,180
        DB  94,94,157,157,53,53,11
        DB  93

;       ----------------------------------
;       SCREEN CHARACTER POSITIONS, DATA
;       ----------------------------------

XP      DB  27,28,29,30,31,25,26,27
        DB  28,29,25,26,27,28,29
        DB  25,26,27,28,29,30,31,32
        DB  33,34,25,26,27,28,29
        DB  25,26,27,28,29
        DB  25,26,27,28,25,26,27
        DB  25,26,27,25,26,27,25
        DB  26,27,28,29,25,26,27
        DB  28,29,30,25,26,27,28
        DB  29,30,37,37,37,37,37
        DB  37,37,37,37
YP      DB  16,16,16,16,16,30,30,30
        DB  30,30,40,40,40,40,40
        DB  58,58,58,58,58,58,58,58
        DB  58,58,72,72,72,72,72
        DB  82,82,82,82,82
        DB  98,98,98,98,112,112,112
        DB  122,122,122,132,132,132,146
        DB  146,146,146,146,160,160,160
        DB  160,160,160,170,170,170,170
        DB  170,170,17,25,33,41,49
        DB  57,65,73,81
CH      DB  51,40,41,48,26,57,48,47
        DB  51,26,58,48,47,51,26
        DB  47,34,42,37,35,52,41
        DB  54,37,26,57,48,47,51,26
        DB  58,48,47,51,26
        DB  33,44,52,26,56,54,26
        DB  57,54,26,58,54,26,38
        DB  53,37,44,26,51,40,41
        DB  48,51,26,31,35,47,58
        DB  37,26,58,33,36,41,33
        DB  52,41,47,46

;       ------------------------------------
;       TOP-OF-SCREEN MESSAGES & INDEXES
;       ------------------------------------

MSGIX   DB  0,22,44,66,88,110,132
        DB  154,176,198,220

MESSGE  DB  0,0,0,0,0,0,0,39,33,45,37 ;GAME OVER
        DB  0,47,54,37,58,0,0,0,0,0
        DB  0,0

SUCLAN  DB  0,0,51,53,35,35,37,51   ;LANDED OK
        DB  51,38,53,44,0,44,33,46
        DB  36,41,46,39,0,0

CRSLAN  DB  0,0,0,38,33,52,33,44    ;CRASH
        DB  0,51,40,41,48,0,35,58
        DB  33,51,40,0,0,0

LOWFU   DB  0,38,53,37,44,0,44,37   ;LOW FUEL
        DB  54,37,44,0,35,58,41,52
        DB  41,35,33,44,0,0

OUTFU   DB  0,0,0,0,0,47,53,52,0,47 ;OUT OF FUEL
        DB  38,0,38,53,37,44,0,0,0
        DB  0,0,0

RADCRS  DB  0,0,50,33,36,41,33,52   ;RADIATION
        DB  41,47,46,0,47,54,37,58
        DB  44,47,33,36,0,0

IFUMSG  DB  0,0,0,0,0,0,38,53,37,44 ;INIT FUEL
        DB  26,0,0,0,0,16,16,16,0,0,0
        DB  0,0

GRVMSG  DB  0,0,0,0,0,0,0,39,58,33,54 ;GRAVITY
        DB  41,52,57,26,0,0,0,0,0,0,0
        DB  0,0

TTLMSG  DB  0,0,0,0,0,0,0,58,37,52,58 ;TITLE
        DB  47,38,41,58,37,1,0,0,0
        DB  0,0,0

AUTMSG  DB  0,0,0,0,0,34,121,26,0,52  ;AUTHOR
        DB  111,109,0,40,117,100
        DB  115,111,110,0,0,0,0

MAGMSG  DB  0,0,0,33,46,33,44,47,39 ;MAGAZINE
        DB  0,35,111,109,112,117
        DB  116,105,110,103,0,0,0

IFUEL1  DB  0               ;INITIAL FUEL
IFUEL2  DB  5
FTIME   DB  0
GRAVTY  DB  0               ;GRAVITY LEVEL
GRAVS   DB  10,24,48        ;GRAVITY FACTORS
GCHAR   DB  44,45,46        ;L,M,H

;       ----------------------
;       INITIAL NUMERICAL DATA
;       ----------------------

ININBR  DB  $02,$00,$00   ;ALT
        DB  $00,$00,$00   ;XV
        DB  $00,$00       ;YV
        DB  $00,$00       ;ZV
        DB  $01,$00,$00   ;FUEL
        DB  $02           ;SHIPS
        DB  $00,$00,$00   ;SCORE

;       ----------------
;       DIRECTION ARROWS
;       ----------------

ARROWS  DB  0,$18,$18,$18,$7E,$3C,$18,0
        DB  0,$18,$3C,$7E,$18,$18,$18,0
        DB  0,$18,$0C,$7E,$7E,$0C,$18,0
        DB  0,$18,$30,$7E,$7E,$30,$18,0
        DB  0,$3C,$1C,$34,$64,$C0,0,0
        DB  0,0,$C,$98,$B0,$E0,$F0,0
ARRLO   DB  LOW [DISP+29+112*40]
        DB  LOW [DISP+29+122*40]
        DB  LOW [DISP+29+132*40]
ARRHI   DB  HIGH [DISP+29+112*40]
        DB  HIGH [DISP+29+122*40]
        DB  HIGH [DISP+29+132*40]

;       ---------------
;       JOYSTICK TABLES
;       ---------------

ENGIX   DB  0,0,0,0,0,0,3,2  ;ENGINE NUMBERS
        DB  0,3,0,2,0,1,1,0

FUELIX  DB  0,0,0,0,0,0,8,6  ;FUEL DATA POINTERS
        DB  0,8,0,6,0,3,3,0

FSIGNS  DB  0,0,0,0,0,0,2,1  ;DIRECTIONS
        DB  0,2,0,1,0,0,0,0

VAH     DB  0,0,0,0,0,0,0,0  ;VELOCITY ADD VALS.
        DB  0,$99,0,$99,0,0,$99,0
VAL     DB  0,0,0,0,0,0,1,1
        DB  0,$99,0,$99,0,1,$99,0

;       -------------------------
;       DECIMAL TO BINARY DATA
;       -------------------------

ALT1L   DB  0,10,20,30,40
        DB  50,60,70,80,90
ALT2H   DB  0,0,0,1,1
ALT2L   DB  1,2,3,3,3
ALT3H   DB  0,$64,$C8,$2C,$90
ALT3L   DB  $F4,$58,$BC,$20,$84
ALT4H   DB  0,3,7,$B,$F
ALT4L   DB  $13,$17,$1B,$1F,$23
        DB  0,$E8,$D0,$B8,$A0
        DB  $88,$70,$58,$40,$28
ALT4H   DB  0,$27,$4E,$75
ALT4L   DB  0,$10,$20,$30

GRDDIV  DB  80,22,3         ;GRID DIVISION FACTORS

VELSHF  DB  7,4,1           ;VELOCITY DIVISION FACS

;       ------------------------------
;       GRID TERRAIN SAVE WORK AREAS
;       ------------------------------

GDIX    DB  7,15,23,31
HSAVE   DB  56,57,58,59,60,61,62,63
        DB  0,1,2,3,4,5,6,7
        DB  7,15,23,31,39,47,55,63
        DB  0,8,16,24,32,40,48,56
HREST   DB  0,1,2,3,4,5,6,7
        DB  56,57,58,59,60,61,62,63
        DB  0,8,16,24,32,40,48,56
        DB  7,15,23,31,39,47,55,63

ENGCLS  DB  $C4,$16,$34,$02 ;ENGINE COLORS
ENGTHR  DB  255,200,90,0    ;ENG EFFECTIVENESS

RADINI  DB  120,255,0       ;RADIATION TIMES

BASPIC  DB  $80,$40,$F0,$20,$10 ;PAD IMAGE
PCOLRS  DB  $00,$32             ;PAD COLORS

PYPOS   DB  72,82,30,40
POSNIX  DB  0,3,6,9
HI1     DB  0,0,1,1,2,2
HIZ     DB  0,$47,$96,$47,$96,$45,$94
MED1    DB  0,$07,$14,$21,$28,$35,$42

ECOLOR  DB  $16,$C4,0,0,0   ;DLI COLORS
UNSNUM  DS  17              ;UNSIGNED NUMBERS
SIGNS   DS  4               ;SIGN INFORMATION
HEIGHT  DS  64              ;GRID HEIGHT
HGTSAV  DS  8               ;HEIGHT SAVE
TEMP1   DS  3               ;ENGINE...
TEMP2   DS  3               ;TEMPS
TEMPS   DS  3               ;0,1,2,3 TEMPS

;       EXPLOSION DATA

EXLV    DS  20              ;EXPLOSION LIFE
EXXP    DS  20              ;X POSITIONS
EXYP    DS  20              ;Y POSITIONS
EXXI    DS  20              ;X INCREMENT
EXYI    DS  20              ;Y INCREMENT
EXXA    DS  20              ;X ACCUM
EXYA    DS  20              ;Y ACCUM
TEMPCX  DS  1

        END MOVEIT+$2000
```

OMNITREND'S

UNIVERSE

For 200 years the people of Axia, the central planet in a cluster of colonies known as the Local Group, have relied on spacecraft from Earth for economic support. The spacecraft, propelled by Earth's prized Hyperspace Booster, arrive regularly at Axia, carrying Earth's latest technological advances and trained personnel. These shipments from Earth are strictly one-way, because the Local Group does not have a Hyperspace Booster; Earth has been dispatching the ships based only on her faith in the colonists' ability to survive.

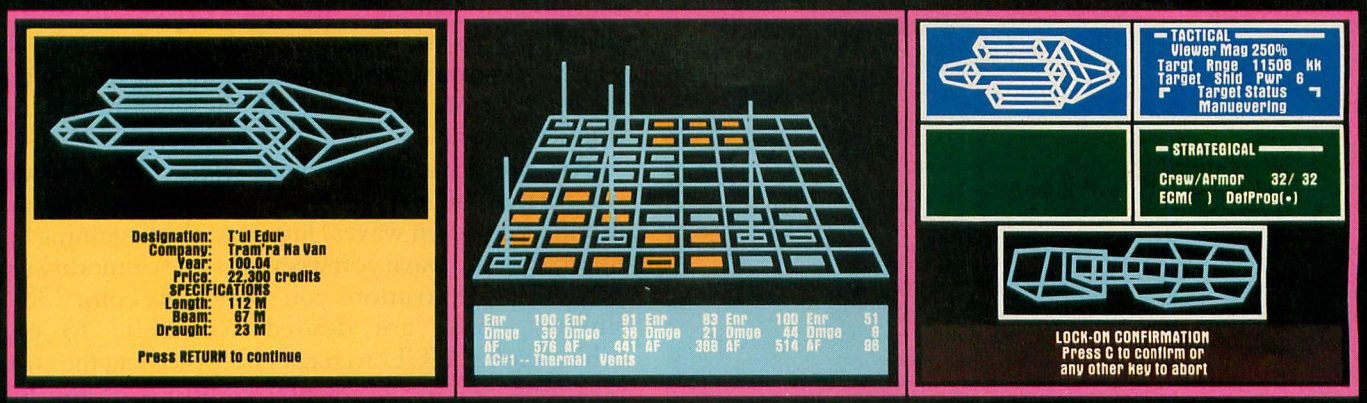
Four months ago, the expected ship did not arrive. The colony has been caught in a wave of desperate concern and wild speculation, for without Earth's assistance, technological deterioration is certain.

Fifteen days ago, evidence of a second Hyperspace Booster, lost somewhere in the Local Group, surfaced. Discovery of this second Booster would mean a fortune for those who found it, and would renew contact with Earth; failure to find the Booster would mean the eventual destruction of the colonies. You and your colleagues have decided to search for it.

Your search takes you through a multitude of star systems and planets, using true three dimensional flight, orbits, and orbital transfers. But your voyage will not be free. You must earn money to maintain spacecraft and crew. You will need to use your ship for passenger transport, mining, trading in exotic goods, or, for the desperate, piracy. You may need to defend yourself, for there are others who are eager for profit and power. You will have to construct the spacecraft most suited to your endeavor and decide what is needed to survive in deep space while contending with unknowns.

This real time game, with hundreds of kilobytes of data, features intelligent enemy ships and total control of on-board facilities such as computers, high-resolution scanners, weapons, assault capsules, ore processors, orbital shuttles, rescue pods, and much more.

Using high-resolution graphics, and more than 30 custom displays—distributed on four disks—Omnitrend's Universe allows you to experience the life of a starship captain in search of the lost Hyperspace Booster.

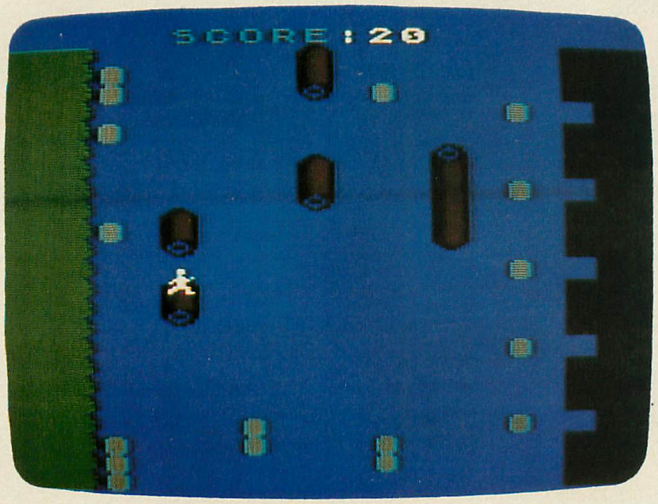


Omnitrend's Universe™ fits your Atari® 400, 800 or 1200 XL computer. Requires 48K and one Atari 810 disk drive.

To order, see your local dealer. If he does not have the program, send your check for $89.95 plus $2 for shipping and handling to Omnitrend Software, P.O. Box 3, West Simsbury, CT 06092. Connecticut residents add 7 1/2 sales tax.

For Master Card, Visa and C.O.D orders, phone 203-658-6917.

OMNITREND
SOFTWARE
P.O. BOX 3 WEST SIMSBURY, CT 06092



LUMBERJACK

24K Cassette 32K Disk

by Bernard Ertl

You are a lumberjack determined to cross a treacherous river to get to your sawmills. Not knowing how to swim, you must use floating logs and slippery stones to accomplish this feat, while avoiding dangers that will hinder your progress. Be sure to land on both feet when jumping because the stones and logs are slippery! Some stones have been known to sink into the mud soon after being stepped on. Also, beware of jumping fish. While not deadly themselves, they will knock you off the logs if they hit you.

To begin play, select color or black and white by pressing the SELECT key, and the press the START key. It will take about a minute for initialization. You will have five lumberjacks to gain as many points as possible. Points are gained by jumping forward for ten points, jumping in a sawmill for fifty points, and filling all the sawmills for five hundred points.

For the adventurous lumberjack, this program was written so that it would be easy to create and modify your own waves. Just go to the beginning line number of the wave you wish to create or modify and plot the stone positions you want using color 238. If sinking stones are desired set UUU to one, otherwise set UUU to zero. After plotting the stone positions, set the horizontal positions of the logs (players 1-3) using the chart and variables F, G, and H. Next, the log speeds must be set using variables to AA x BB, and CC when all five sawmills are full to keep the speeds of the logs between five and negative five. The variables AAA, BBB and CCC are added to AA, BB and CC when all five sawmills are full to speed up or slow down the logs. Also set variables D and E as shown in previous waves and include a RETURN statement at the end. Finally, under the "Wave Complete" section, enter an IF-THEN statement similar to the previous ones setting WA to the line number where your wave begins if WAVE equals the wave you desire. □

If you are serious about personal finance...

- **Budget Forecast** – 26 expense categories
- **Check Entry** – easy data entry – scan & modify 26 major & 36 sub-categories – information block
- **Check Search** – single or multiple parameters – (up to seven) to search entries
- **Tabulations** – detailed expense vs. budget comparisons by month, year-to-date, category
- **Bar Graphs** – screen displays in graph form expenses vs. budget – by month or category – printing with graphic capable printers
- **Check Reconciliation** – fast clearing of resident checks & deposits, complete summary report
- **Checkwriter** – prints your custom checks
- **Printouts** – most popular printers
- **Multi-Colored Graphics** • **Audio Enhancements**
- **7 Utility Programs** • **User-Friendly Operation**
- **Easy To Use Instruction Manual** • **Audit Report**
- **Handsome Tinted Plastic Storage Case**

COMPUTE!
The Leading Magazine Of Home, Educational, And Recreational Computing

In a feature editorial.

"If you want to use a finance system, but don't want to spend several days trying to learn how to use one, then A Financial Wizard by Computari may be just what you need."

"The illustrated manual that comes with this program is clear, direct, and very thorough."

"It appears that this finance system was designed to achieve the best and most comfortable working relationship between the user and the program."

"The check entry routine is the most attractive feature of this finance system. Data prompts are very clear and the category item names are displayed at all times during data entry for your convenience."

"The file search capabilities of this program are superior. You are offered seven ways to look up the checks."

"The system is disk intensive. All data is saved automatically and immediately following all routines that either enter data or modify it."

"Scanning your entries is made possible by pressing START. You can see records very quickly this way."

"This is an excellent finance system – entertaining, accurate, and fun to use."

ANALOG COMPUTING
THE MAGAZINE FOR ATARI COMPUTER OWNERS

Analog Magazine in a comprehensive study of personal finance systems for Atari* computers.

"A Financial Wizard from Computari is by far the best of these programs and will be the standard of comparison for the others."

"The check entry mode is easy to use..."

"The way a Financial Wizard handles your tabulations is excellent. You can chart your actual expenses vs. your budget by month, by category or year to date."

"...where it really outshines the rest is in the check reconciliation."

"In effect it gives you your bank statement on the screen, a complete list by month of all your checks and deposits."

"A Financial Wizard has one disk that does everything..."

"Graphics, while really not a factor in the quality of programs of this type, do make your budgeting chores a little more pleasant. Again A Financial Wizard comes out on top."

"Everything about this program is excellent..."

Antic
The ATARI Resource

In a Report from Antic.

"Like most Atarians, I am captivated by the graphic, color and sound capabilities of my machine. Nothing quite discourages me more than to boot up an applications program (personal, business, etc.) and to be presented with the standard graphic 'o' white characters on a blue screen.

Of course the usefulness and effectiveness of a program is of primary importance. However, enhancing the dullest of applications programs with some of Atari's charms, is a great asset. A Financial Wizard, a personal finance program by Computari's Bill McLachlan, is an excellent example of an applications program that integrates many of the Atari's features into a well conceived and executed program."

"The use of color and sound in the data input prompts and error checking routines are so well done that it's quite simple to boot up the disk, follow along with the very clear documentation, and be 'up and running' in short order."

"I give A Financial Wizard high marks in ease of use, documentation and performance. If a disk-based home finance package is in your future, The Wizard should get serious consideration."

Computari's
A Financial Wizard 1.5
The logical choice.

The system is designed for Atari computers having a minimum of 32K and operating from a disk drive. The cost is only $59.95 plus $3 for handling/postage.

If your dealer does not have **A Financial Wizard**... Telephone orders are accepted on Mastercharge or Visa credit cards. Mail order must be accompanied by check or money-order or credit card #.

Dealer inquiries invited.

*trademark of Atari Inc.

OK residents ad 4% Tax.

Exclusively thru **ON LINE** COMPUTER CENTERS of OKC (405)751-7925

10944 North May, Oklahoma City, OK 73120

CHECK ENCLOSED ☐ VISA ☐ MASTERCHARGE

NAME (Print)_____

Address _____

City _____

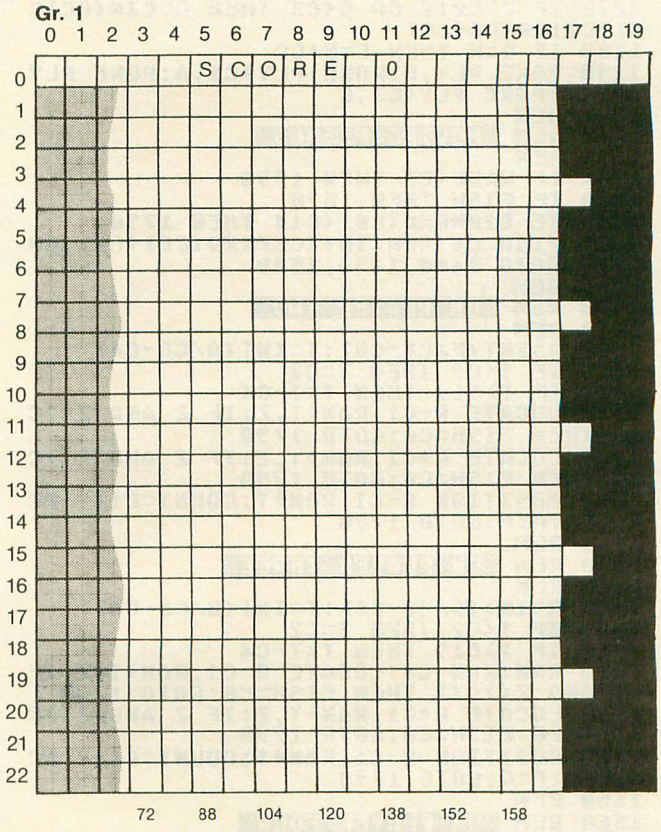
State _____ Zip _____

Card # _____ Exp._____

Signature_____

Variable List

PLX - Location to Poke Player-0's Hor. Position
PLY - Location to Poke Player-0's Vert. Position
E - Y Position of Player 0
A - Y Position of Player 1
B - Y Position of Player 2
C - Y Position of Player 3
D - Y Position of Player 0
F - Y Position of Player 1
G - Y Position of Player 2
H - Y Position of Player 3
Q - Checks for Joystick Release
SC - Score
WAVE - Current Wave
V(0)-V(4) - Remembers if Sawmill Occupied
X - USR for PM Routine
Y - Lives
U - 1=Man On Stone, 0=Man Not On Stone
UU - Counts Time Until Stone Disappears
UUU - 1=Disappearing Stones Desired
I - Loop Variable
WA - Line Number Where Current Wave Is Initialized
AA - Speed of Player 1
BB - Speed of Player 2
CC - Speed of Player 3
AAA - Increments AA when Sawmills Filled
BBB - Increments BB when Sawmills Filled
CCC - Increments CC when Sawmills Filled
FISH — Fish is Present if Fish > 0
RAN - Random for Fish Location
R - Hor. Conversion of Players 1-3 to GR. 1
J - Hor. Conversion of Player 0 to GR. 1
T - Vert. Conversion of Players 1-3 to GR. 1
K - Vert. Conversion of Player 0 to GR. 1
Z - Locate Variable for Fishes
COUNT - Timer for Fish Jumps
P - Log Fish is Jumping
W - Game Over if W=1
ST - Stores Value of Stick(0)
L - Locate Variable for Player 0 to GR. 1
N - Sawmills Full if N=5
S - Loop Variable for Music
M - Color of Lumberjack when Drowning
CHAR$ - Stores Characters to be Edited
CHSET - Location of Edited CHSET
CHORG - Location of CHSET in ROM
CHPOS - Location of Character to be Edited
Z1 - Color of Lumberjack
Z2 - Color of Logs
Z3 - Color of Background
Z4 - Luminance of Background
Z5 - Color of Sawmills
Z6 - Color of Stones
Z7 - Color of Shoreline
PLL - Height of Player 0
PM - Location for PM Set-Up
PMBASE - Location for PM Set-Up




```
1000 REM      LUMBERJACK
1010 REM  by B.Ertl & J.Euker
1020 REM
1030 REM  SETTING UP GAME
1040 REM
1050 C1=1:C2=C1+C1:C3=C2+C1:C4=C3+C1:C
5=C4+C1:C6=C5+C1:C7=C6+C1
1060 C8=C7+C1:C9=C8+C1:C10=C9+C1:C11=C
10+C1:C12=C11+C1:C13=C12+C1:C14=C13+C1
:C15=C14+C1:C16=C15+C1
1070 C17=C16+C1:C18=C17+C1:C19=C18+C1:
C20=C19+C1:C21=C20+C1
1080 C22=C21+C1:C23=C22+C1:C30=C23+C7:
C32=C30+C2:C54=C32+C22:C56=C54+C2:C66=
C56+C10:C186=C30*C6+C6
1090 C210=C7*C30:C228=C210+C18:C238=C2
28+C10:C256=C16*C16:C704=704:CONSOL=53
279:K275=2.75
1100 DIM CHAR$(C9),V(C4):GOTO 3360
1110 GOSUB 3510:GOSUB 3860:? #C6;"K":G
OSUB 3970
1120 A=C1:B=C210:C=C1:Q=C0:SC=C0:WAVE=
C1
1130 V(C0)=C0:V(C1)=C0:V(C2)=C0:V(C3)=
C0:V(C4)=C0:X=C0:Y=C5:WA=C1:U=C0:UU=C0
1140 COLOR C0:FOR I=C0 TO C4:PLOT C17,
I*C4+C4:V(I)=C0:NEXT I:POSITION C0,C0:
? #C6;"                    "
1150 POSITION C5,C11:? #C6;"          "
:GOTO 2900
1160 REM
1170 REM  MOVING LOGS
1180 REM
1190 A=A+AA:B=B+BB:C=C+CC
1200 POSITION C5,C0:? #C6;"score:";SC
1210 U=U+C1:IF UUU=C1 AND UU=C1 AND U=
C15 THEN COLOR C0:PLOT J,K:GOTO 3280
1220 IF E<C30 OR E>C210+C7 THEN 3280
1230 IF A>C210 OR A<C3 THEN A=C3*(AA>C
0)+C210*(AA<C0)
1240 IF D=F THEN E=E+AA
1250 IF B>C210 OR B<C3 THEN B=C3*(BB>C
0)+C210*(BB<C0)
```

```
1260 IF D=G THEN E=E+BB
1270 IF C>C210 OR C<C3 THEN C=C3*(CC>C
0)+C210*(CC<C0)
1280 IF D=H THEN E=E+CC
1290 POKE PLY,E:POKE PLY+C1,A:POKE PLY
+C2,B:POKE PLY+C3,C
1300 REM
1310 REM  MOVING FISHES
1320 REM
1330 IF WAVE<C3 THEN 1790
1340 IF FISH THEN 1670
1350 IF C20*RND(C0)<C14 THEN 1790
1360 FISH=C1:RAN=INT(C3*RND(C0)+C1):ON
 RAN GOTO 1400,1490,1580
1370 REM
1380 REM  ON PLAYER TWO
1390 REM
1400 R=INT(F/C8-C6):T=INT(A/C8-C4)
1410 IF T<C2 THEN T=C2
1420 IF T>C19 THEN T=T-C4
1430 LOCATE R-C1,RAN+T,Z:IF Z AND Z<>C
32 THEN FISH=C0:GOTO 1790
1440 LOCATE R+C1,RAN+T,Z:IF Z AND Z<>C
32 THEN FISH=C0:GOTO 1790
1450 POSITION R-C1,RAN+T:COUNT=C1:? #C
6;"F":P=F:GOTO 1790
1460 REM
1470 REM  ON PLAYER THREE
1480 REM
1490 R=INT(G/C8-C6):T=INT(B/C8-C4)
1500 IF T<C2 THEN T=C2
1510 IF T>C19 THEN T=T-C4
1520 RAN=RAN-C4:LOCATE R-C1,RAN+T,Z:IF
 Z AND Z<>C32 THEN FISH=C0:GOTO 1790
1530 LOCATE R+C1,RAN+T,Z:IF Z AND Z<>C
32 THEN FISH=C0:GOTO 1790
1540 POSITION R-C1,RAN+T:COUNT=C1:? #C
6;"F":P=G:GOTO 1790
1550 REM
1560 REM  ON PLAYER FOUR
1570 REM
1580 R=INT(H/C8-C6):T=INT(C/C8-C4)
1590 IF T<C2 THEN T=C2
1600 IF T>C19 THEN T=T-C4
1610 RAN=RAN-C8:LOCATE R-C1,RAN+T,Z:IF
 Z AND Z<>C32 THEN FISH=C0:GOTO 1790
1620 LOCATE R+C1,RAN+T,Z:IF Z AND Z<>C
32 THEN FISH=C0:GOTO 1790
1630 POSITION R-C1,RAN+T:COUNT=C1:? #C
6;"F":P=H:GOTO 1790
1640 REM
1650 REM  FISH PRESENT
1660 REM
1670 COUNT=COUNT+C1:IF COUNT=C7 THEN F
ISH=FISH+C1:ON FISH-C1 GOTO 1690,1720,
1750
1680 GOTO 1790
1690 COUNT=C1:POSITION R-C1,RAN+T:? #C
6;" ":POSITION R,RAN-C1+T:? #C6;"P":K=
INT(E/C8-K275)
1700 IF D=P AND (K=RAN+T OR K=RAN-C1+T
) THEN D=D+C8:GOTO 3280
1710 GOTO 1790
1720 COUNT=C1
1730 POSITION R,RAN-C1+T:? #C6;" ":POS
ITION R+C1,RAN+T:? #C6;"Q"
1740 GOTO 1790
1750 FISH=C0:COUNT=C0:POSITION R+C1,RA
N+T:? #C6;" "
1760 REM
1770 REM  CHECKS FOR PLAYER MOVE
1780 REM
1790 IF PEEK(CONSOL)=C6 THEN W=C0:POKE
 C704,Z1:POSITION C5,C11:? #C6;"
   ":GOTO 1120
1800 IF Q OR W THEN 1910
1810 ST=STICK(C0)
1820 IF ST=C15 THEN 1190
1830 POKE 77,C0
1840 IF ST=C7 THEN SOUND C0,C54,C10,C1
5:D=D+C16:GOTO 1960
1850 IF ST=C14 THEN SOUND C0,C54,C10,C
15:E=E-C8:GOTO 2100
1860 IF ST=C11 THEN SOUND C0,C54,C10,C
15:D=D-C16:GOTO 2230
1870 IF ST=C13 THEN SOUND C0,C54,C10,C
15:E=E+C8:GOTO 2370
1880 GOTO 1190
1890 SOUND C0,C0,C0,C0:POKE PLX,D:POKE
 PLY,E
1910 Q=C0:IF STICK(C0)<>C15 THEN Q=C1
1920 GOTO 1190
1930 REM
1940 REM  RIGHT MOVE
1950 REM
1960 IF UU=C1 THEN COLOR C0:PLOT J,K
1970 IF D=F OR D=G OR D=H THEN UU=C0:G
OTO 2040
1980 J=INT(D/C8-C6):K=INT(E/C8-K275):I
F J=C17 THEN SC=SC+C30+C20:GOTO 2650
1990 IF K>C23 THEN 3280
2000 LOCATE J,K,L
2010 IF L=C32 OR  NOT L THEN 3280
2020 IF UUU=C1 THEN COLOR C228:PLOT J,
K:U=C0:UU=C1
2030 SC=SC+C10:GOTO 1890
2040 IF D=F THEN 2560
2050 IF D=G THEN 2580
2060 GOTO 2600
2070 REM
2080 REM  UP MOVE
2090 REM
2100 IF UU=C1 THEN COLOR C0:PLOT J,K
2110 IF D=F OR D=G OR D=H THEN UU=C0:G
OTO 2170
2120 J=INT(D/C8-C6):K=INT(E/C8-K275)
2130 LOCATE J,K,L
2140 IF L=C32 OR  NOT L THEN 3280
2150 IF UUU=C1 AND J>C3 THEN COLOR C22
8:PLOT J,K:U=C0:UU=C1
2160 GOTO 1890
2170 IF D=F THEN 2500
2180 IF D=G THEN 2520
2190 GOTO 2540
2200 REM
2210 REM  LEFT MOVE
2220 REM
2230 IF UU=C1 THEN COLOR C0:PLOT J,K
2240 SC=SC-C10:IF D=F OR D=G OR D=H TH
EN UU=C0:GOTO 2310
2250 IF D<C56 THEN D=D+C16
2260 J=INT(D/C8-C6):K=INT(E/C8-K275):I
F K>C23 THEN 3280
2270 LOCATE J,K,L
2280 IF L=C32 OR  NOT L THEN 3280
2290 IF UUU=C1 AND J>C3 THEN COLOR C22
8:PLOT J,K:U=C0:UU=C1
2300 GOTO 1890
2310 IF D=F THEN 2500
2320 IF D=G THEN 2520
2330 GOTO 2540
2340 REM
2350 REM  DOWN MOVE
2360 REM
2370 IF UU=C1 THEN COLOR C0:PLOT J,K
2380 IF D=F OR D=G OR D=H THEN UU=C0:G
OTO 2440
2390 J=INT(D/C8-C6):K=INT(E/C8-K275):I
F K>C23 THEN 3280
2400 LOCATE J,K,L
2410 IF L=C32 OR  NOT L THEN 3280
2420 IF UUU=C1 AND J>C2 THEN COLOR C22
8:PLOT J,K:U=C0:UU=C1
2430 GOTO 1890
2440 IF D=F THEN 2500
2450 IF D=G THEN 2520
2460 GOTO 2540
2470 REM
2480 REM  MAN ON LOGS
2490 REM
2500 IF E+C7>A AND E<A+C8 OR E>A+C21 A
ND E+C9<A+45 THEN 1890
2510 GOTO 3280
2520 IF E+C7>B AND E<B+C30+C1 THEN 189
0
2530 GOTO 3280
2540 IF E+C7>C AND E<C+C12 OR E+C9>C+4
7 AND E+C9<C+C66 THEN 1890
2550 GOTO 3280
2560 IF E+C7>A AND E<A+C8 OR E>A+C21 A
ND E+C9<A+45 THEN SC=SC+C10:GOTO 1890
2570 SC=SC-C10:GOTO 3280
2580 IF E+C7>B AND E<B+C21+C10 THEN SC
=SC+C10:GOTO 1890
```

```
2590 SC=SC-C10:GOTO 3280
2600 IF E+C7)C AND E(C+C12 OR E)C+C20+
C18 AND E+C9(C+C66 THEN SC=SC+C10:GOTO
 1890
2610 SC=SC-C10:GOTO 3280
2620 REM
2630 REM  SAM MILLS
2640 REM
2650 UU=C0:POKE C704,Z3*C16+Z4:POKE PL
X,C0:POKE PLY,C10*C10+C1:FOR I=C0 TO C
4
2660 IF K=I*C4+C4 AND  NOT V(I) THEN C
OLOR 346:GOTO 2680
2670 NEXT I:SC=SC-C20-C30:GOTO 3280
2680 PLOT C17,K:V(I)=C1
2690 FOR I=C9*C6 TO C0 STEP -C1:SOUND
C0,I,C10,C15:NEXT I
2700 N=C0:FOR I=C0 TO C4:IF V(I)=C1 TH
EN N=N+C1
2710 NEXT I:IF N=C5 THEN 2760
2720 D=C56:E=INT(C186*RND(C0)+C30):POK
E C704,Z1:GOTO 1890
2730 REM
2740 REM  WAVE COMPLETE
2750 REM
2760 SC=SC+C10*C5*C5:COLOR C0:FOR I=C0
 TO C4:PLOT C17,I*C4+C4:V(I)=C0:NEXT I
2770 WAVE=WAVE+C1:D=C56:E=C10*C10+C1:P
OKE C704,Z1:AA=AA+AAA:BB=BB+BBB:CC=CC+
CCC
2772 REM
2773 REM  CHARGE THEME
2774 REM
2775 RESTORE 2776:FOR J=C0 TO C18:READ
 A:SOUND C0,A,C10,C15:FOR I=C0 TO C5:N
EXT I:NEXT J
2776 DATA 85,0,64,0,50,0,42,0,0,50,0,4
2,42,42,42,42,42,0
2780 IF WAVE=C3 THEN WA=C2:GOTO 2900
2790 IF WAVE=C5 THEN WA=C3:GOTO 2900
2800 IF WAVE=C6 THEN WA=C3:GOTO 2900
2810 IF WAVE)C6 THEN WA=C3:GOTO 2900
2812 REM
2820 REM  CHANGE LINE 2810 TO:
2830 REM  IF WAVE=7 THEN WA=C4:GOTO 29
00
2840 REM  ONLY IF YOU ADDED YOUR
2850 REM          OWN WAVE
2860 GOTO 1890
2870 REM
2880 REM  SETTING UP NEW WAVE
2890 REM
2900 COLOR C0:FOR I=C3 TO C15:PLOT I,C
0:DRAWTO I,C23:NEXT I:FISH=C0
2910 ON WA GOSUB 2970,3040,3120,3200
2920 POKE PLX,D:POKE PLX+C1,F:POKE PLX
+C2,G:POKE PLX+C3,H
2930 GOTO 1190
2940 REM
2950 REM  FIRST WAVE
2960 REM
2970 COLOR C238:FOR I=C4 TO C20 STEP C
4:PLOT C15,I:NEXT I:PLOT C3,C2:PLOT C3
,C3:PLOT C3,C5:PLOT C3,C10
2980 PLOT C3,C21:PLOT C3,C22:PLOT C3,C
23:PLOT C7,C20:PLOT C7,C21:PLOT C11,C2
1:PLOT C11,C22:PLOT C11,C3
2990 UUU=C0:F=88:AA=C3:AAA=C1/C2:G=152
:BB=-C9/C4:BBB=-C3/C4:H=C30*C4:CC=C1
3000 CCC=C3/C2:D=C56:E=INT(C186*RND(C0
)+C30):RETURN
3010 REM
3020 REM  SECOND WAVE
3030 REM
3040 COLOR C238:PLOT C9,C2:DRAWTO C11,
C2:PLOT C9,C3:DRAWTO C11,C3:PLOT C9,C1
0
3050 DRAWTO C11,C10:PLOT C9,C22:DRAWTO
 C11,C22
3060 PLOT C3,C10:PLOT C3,C23:PLOT C3,C
3:FOR I=C4 TO C20 STEP C4:PLOT C15,I:N
EXT I
3070 UUU=C0:D=C56:E=INT(C186*RND(C0)+C
30):F=152:G=104:H=88:AA=C3:AAA=C3/C2
3080 BB=-C3:BBB=-C1:CC=C5/C2:CCC=AAA:R
ETURN
3090 REM
3100 REM  THIRD WAVE
3110 REM
3120 COLOR C238:FOR I=C4 TO C20 STEP C
4:PLOT C15,I:NEXT I:FOR I=C4 TO C20 ST
EP C4:PLOT C13,I:NEXT I
3130 PLOT C5,C1:DRAWTO C5,C23:PLOT C9,
C1:DRAWTO C9,C23
3140 D=C56:E=INT(C186*RND(C0)+C30):F=C
18*C4:G=104:H=136:AA=C2+(WAVE=C6):AAA=
C0:BB=-C3-(WAVE=C6):BBB=C0
3150 CC=K275+((WAVE=C6)+C1/C4):UUU=C1:
CCC=C0:RETURN
3160 REM
3170 REM  FOURTH WAVE
3180 REM  YOUR WAVE GOES HERE!
3190 REM  GOOD LUCK
3250 REM
3260 REM  DEATH
3270 REM
3280 UU=C0:Y=Y-C1:POKE PLX,D:POKE PLY,
E
3290 FOR M=C14 TO Z4 STEP -C2:SOUND C0
,C10,C8,M:POKE C704,Z3*C16+M:FOR I=C0
TO C54
3300 NEXT I:NEXT M:D=C56:E=INT(C186*RN
D(C0)+C30)
3310 IF  NOT Y THEN W=C1:D=C0:POSITION
 C5,C11:? #C6;"game over":GOTO 1890
3320 POKE C704,Z1:GOTO 1890
3330 REM
3340 REM  TITLE PAGE
3350 REM
3360 GRAPHICS C17:SETCOLOR C4,C2,C3
3370 POSITION C1,C5:? #C6;"*** lumber j
ack ***":? #C6:? #C6;"   by bernie ert
l":? #6;"   and john euker"
3380 Z7=C0:POSITION C4,C12:? #C6;"   c
olor?   ":Z1=42:Z2=C32:Z3=C9:Z4=C0:Z5=
C0:Z6=C4:Z7=194
3390 FOR I=C0 TO C32:NEXT I
3400 IF PEEK(CONSOL)=C5 THEN 3420
3410 GOSUB 3460:GOTO 3400
3420 POSITION C4,C12:? #C6;"black/whit
e?":Z1=C14:Z2=C0:Z3=C0:Z4=C10:Z5=148:Z
6=C10*C7:Z7=C0
3430 FOR I=C0 TO C32:NEXT I
3440 IF PEEK(CONSOL)=C5 THEN 3380
3450 GOSUB 3460:GOTO 3440
3460 IF PEEK(CONSOL)=C6 THEN POSITION
C1,C12:? #C6;"one moment please.":GOTO
 1110
3470 RETURN
3480 REM
3490 REM  P/M GRAPHICS
3500 REM
3510 POKE 752,C1
3520 RESTORE 3650:GOSUB 3530:POKE PLL,
C10:POKE PLL+C1,C12*C4:POKE PLL+C2,43:
POKE PLL+C3,69:RETURN
3530 FOR I=1536 TO 1706:READ A:POKE I,
A:NEXT I
3540 FOR I=1774 TO 1787:POKE I,C0:NEXT
 I
3550 PM=PEEK(106)-C16:PMBASE=C256*PM
3560 FOR I=PMBASE+1023 TO PMBASE+2047:
POKE I,C0:NEXT I
3570 FOR I=PMBASE+1025 TO PMBASE+1034:
READ A:POKE I,A:NEXT I
3580 FOR I=PMBASE+1281 TO PMBASE+1328:
READ A:POKE I,A:NEXT I
3590 FOR I=PMBASE+1537 TO PMBASE+1579:
READ A:POKE I,A:NEXT I
3600 FOR I=PMBASE+1793 TO PMBASE+1861:
READ A:POKE I,A:NEXT I
3610 POKE C704,Z1:POKE C704+C1,Z2:POKE
 C704+C2,Z2:POKE C704+C3,Z2
3620 PLX=53248:PLY=1780:PLL=1784
3630 POKE 559,62:POKE 623,C16:POKE 178
8,PM+4:POKE 53277,3:POKE 54279,PM
3640 X=USR(1696):RETURN
3650 DATA 162,3,189,244,6,240,89,56,22
1,240,6,240,83,141,254,6,106,141
3660 DATA 255,6,142,253,6,24,169,0,109
,253,6,24,109,252,6,133,204,133
3670 DATA 206,189,240,6,133,203,173,25
4,6,133,205,189,248,6,170,232,46,255
```

```
3680 DATA 6,144,16,168,177,203,145,205
,169,0,145,203,136,202,208,244,76,87
3690 DATA 6,160,0,177,203,145,205,169,
0,145,203,200,202,208,244,174,253,6
3700 DATA 173,254,6,157,240,6,189,236,
6,240,48,133,203,24,138,141,253,6
3710 DATA 109,235,6,133,204,24,173,253
,6,109,252,6,133,206,189,240,6,133
3720 DATA 205,189,248,6,170,160,0,177,
203,145,205,200,202,208,248,174,253,6
3730 DATA 169,0,157,236,6,202,48,3,76,
2,6,76,98,228,0,0,104,169
3740 DATA 7,162,6,160,0,32,92,228,96
3750 DATA 24,24,16,26,60,88,24,60,100,
70,24,60,126,255,255,255,255,255,255,2
55,255,255,255,231,195,153,153,66
3760 DATA 38,24,0,0,0,0,0,0,0,24,60,
126,255,255,255,255,255,255,255,255,25
5,255,231,195,153,153,66,38,24
3770 DATA 24,38,66,153,153,195,231,255
,255,255,255
3780 DATA 255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255
3790 DATA 255,255,255,255,255,255,255,
255,255,255,255,255,255,126,60,24,
24,60,126,255,255,255,255,255
3800 DATA 255,255,255,255,255,255,255,
255,255,231,195,153,153,66,38,24
3810 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0
3820 DATA 0,24,60,126,255,255,255,255,
255,255,255,255,255,255,255,255,255,23
1,195,153,153,66,38,24
3830 REM
3840 REM EDITING CHSET
3850 REM
3860 DATA 60,126,126,126,126,126,126,6
0,255,255,255,255,255,255,255,255
3870 DATA 255,255,255,255,255,255,255,
0,56,186,214,124,56,16,56,108
3880 DATA 254,255,254,255,255,254,252,
254,0,0,8,28,46,126,252,56
3890 DATA 0,48,24,28,186,127,190,28,0,
0,0,16,96,48,120,126,0,24,60,60,60,60,
24,0
3900 RESTORE 3860:CHAR$="NXYZWFPQD":CH
SET=(PEEK(106)-C32)*C256:CHORG=57344
3910 FOR I=C0 TO C256+C256-C1:POKE CHS
ET+I,PEEK(CHORG+I):NEXT I:FOR I=C1 TO
C9
3920 CHPOS=CHSET+(ASC(CHAR$(I))-C32)*C
8:FOR J=C0 TO C7:READ A:POKE CHPOS+J,A
:NEXT J:NEXT I
3930 FOR I=C32 TO 41:POKE CHSET+I,C256
-C1-PEEK(CHORG+I):NEXT I:POKE 756,CHSE
T/C256:RETURN
3940 REM
3950 REM DRAW PLAYING FIELD
3960 REM
3970 COLOR 216:PLOT C19,C1:DRAWTO C19,
C23:PLOT C18,C1
3980 DRAWTO C18,C23:PLOT C17,C1:PLOT C
17,C2:PLOT C17,C5:PLOT C17,C6
3990 PLOT C17,C9:PLOT C17,C10:PLOT C17
,C13:PLOT C17,C14:PLOT C17,C17
4000 PLOT C17,C18:PLOT C17,C21:DRAWTO
C17,C23
4010 COLOR 217:FOR I=C3 TO C19 STEP C4
:PLOT C17,I:NEXT I
4020 FOR I=C1 TO C23:POSITION C0,I:? #
C6;"xxw":NEXT I
4030 POKE C704+C5,Z7:POKE C704+C6,Z5:P
OKE C704+C7,Z6:SETCOLOR C4,Z3,Z4:RETUR
N
```

•

CHECKSUM DATA

(See p. 58)

```
1000 DATA 820,889,277,969,279,674,8,69
8,61,166,467,774,267,362,735,7446
1150 DATA 242,284,443,286,340,588,316,
213,993,859,8,869,23,879,545,6888
1300 DATA 284,162,286,392,9,961,629,29
1,144,293,402,34,302,92,87,4368
1450 DATA 292,293,881,295,413,36,304,4
79,89,297,295,661,297,417,45,5094
1600 DATA 306,485,91,302,297,806,299,5
6,747,200,892,743,44,50,746,6064
1750 DATA 667,302,671,304,502,9,189,98
8,972,912,903,206,925,747,582,8879
1910 DATA 940,744,305,988,307,703,25,8
8,666,217,994,468,7,610,620,7682
2060 DATA 713,284,773,286,675,3,477,22
3,0,573,733,598,608,726,283,6955
2210 DATA 569,285,681,819,39,926,230,7
,580,733,598,608,726,290,578,7669
2360 DATA 292,688,22,932,229,6,576,739
,604,614,732,296,317,298,0,6345
2510 DATA 738,663,740,759,742,836,19,9
79,21,674,16,297,524,299,732,8039
2660 DATA 703,436,874,817,803,596,920,
301,255,303,720,120,309,632,313,8102
2775 DATA 433,37,687,693,688,691,306,6
92,492,763,618,754,308,654,310,8126
2900 DATA 722,85,582,747,308,26,310,89
4,675,992,644,280,435,282,615,7597
3050 DATA 53,203,925,492,288,3,283,429
,843,128,718,288,473,960,550,6636
3250 DATA 290,93,292,294,412,84,29,437
,291,968,293,535,73,613,411,5115
3400 DATA 180,367,2,408,200,375,164,80
2,299,660,294,178,513,68,318,4828
3550 DATA 564,869,590,611,641,631,891,
28,476,275,458,200,687,665,752,8338
3700 DATA 322,551,816,665,20,333,957,2
81,316,879,524,176,84,306,213,6443
3850 DATA 308,366,216,933,244,366,821,
666,631,310,74,312,564,527,319,6657
4000 DATA 576,131,900,851,2458
```

•

CONTROL CHARACTERS

Some program listings reproduced in A.N.A.L.O.G. may contain "strange" characters not shown on the ATARI keyboard. These are special characters which use the CTRL, ESC and "ATARI LOGO" (INVERSE) keys. Shown below is a list of these characters and the keystrokes used to get them. □

♥ --- CTRL ,
├ --- CTRL A
┤ --- CTRL B
┘ --- CTRL C
┼ --- CTRL D
┐ --- CTRL E
╱ --- CTRL F
╲ --- CTRL G
◢ --- CTRL H
▪ --- CTRL I
◣ --- CTRL J
▝ --- CTRL K
▘ --- CTRL L
▔ --- CTRL M
▁ --- CTRL N
▗ --- CTRL O
♣ --- CTRL P
▖ --- CTRL Q
▂ --- CTRL R
+ --- CTRL S
● --- CTRL T
▃ --- CTRL U
▏ --- CTRL V
┬ --- CTRL W
┴ --- CTRL X
▌ --- CTRL Y
└ --- CTRL Z
ESC --- ESC ESC
↑ --- ESC CTRL UP-ARROW
↓ --- ESC CTRL DOWN-ARROW
← --- ESC CTRL LEFT-ARROW
→ --- ESC CTRL RIGHT-ARROW
● --- CTRL .
♠ --- CTRL ;
↖ --- ESC SHIFT CLEAR
◀ --- ESC BACK S
▶ --- ESC TAB
♥ --- INVERSE CTRL ,
▌ --- INVERSE CTRL A
▐ --- INVERSE CTRL B
▟ --- INVERSE CTRL C
▙ --- INVERSE CTRL D
▛ --- INVERSE CTRL E
▜ --- INVERSE CTRL F
▚ --- INVERSE CTRL G
▞ --- INVERSE CTRL H
▛ --- INVERSE CTRL I
▜ --- INVERSE CTRL J
▙ --- INVERSE CTRL K
▟ --- INVERSE CTRL L
▐ --- INVERSE CTRL M
▐ --- INVERSE CTRL N
▛ --- INVERSE CTRL O
▜ --- INVERSE CTRL P
▙ --- INVERSE CTRL Q
▟ --- INVERSE CTRL R
▚ --- INVERSE CTRL S
▞ --- INVERSE CTRL T
▀ --- INVERSE CTRL U
▌ --- INVERSE CTRL V
▐ --- INVERSE CTRL W
▛ --- INVERSE CTRL X
▜ --- INVERSE CTRL Y
▙ --- INVERSE CTRL Z
▲ --- ESC DELETE
▲ --- ESC INSERT
▶ --- ESC CTRL TAB (CLR)
▶ --- ESC SHIFT TAB (SET)
▐ --- INVERSE SPACE
▁ --- INVERSE _
● --- INVERSE CTRL .
♠ --- INVERSE CTRL ;
▐ --- INVERSE |
▀ --- ESC CTRL 2
◀ --- ESC CTRL BACK S
▶ --- ESC CTRL INSERT

SYNCHRONIZING VOICE AND PROGRAM IN ATARI PILOT

by Richard Seltzer

For some kinds of teaching, voice is not just interesting — it's essential. Foreign language vocabulary drill is one of these. Others are spelling and beginning reading.

My seven-year-old son, Bobby, could use practice in spelling — not prepackaged instruction, but rather the very words his class is studying that week. My five-year-old daughter, Heather, needs practice associating spoken words with written ones. Bobby and my wife, Barbara, who are taking German lessons, need drill in the vocabulary their teachers introduce each week.

It seemed that ATARI **Pilot** had the capability to enable me to write programs to help them. After all, a number of commercially available programs in BASIC, such as ATARI's foreign language and **Invitation to Programming** series, used taped voice.

But information from ATARI implies that to synchronize a program with a voice tape in **Pilot** you have to use the "TSYNC" instruction and, therefore, need special equipment to record digital signals on one track and voice on another.

After a lot of experimenting, I found a simple way to get around that problem, so with one main program and no fancy additional equipment (just an audio cassette recorder), I could quickly tailor-make new tapes for a variety of purposes whenever needed.

The player hears a spoken word and responds by typing the answer — which might be the correct spelling, a synonym, an antonym, an English equivalent of a foreign spoken word, a foreign equivalent of an English word...the possibilities are limitless. You set up whatever pattern of response you want when you prepare the tape.

The program allows for six separate games of twenty words each on a single tape. The player selects a game and, on completion, can choose which one to try next.

To tailor-make such a tape, load the main program and list it. Then input a set of answer words in sequence, one word per line in lines 160 through 1650, in the form $WORD=YOUR ANSWER, such as $WORD=ELEPHANT or $WORD=GESUNDHEIT. Just substitute your answers for the ones shown in the sample program listing.

If you don't need all six games, you can change the number of choices in lines 70 and 90.

Once you've made these changes in the main program, take a fresh tape and save the program on it. (Be sure to rewind, start your counter at zero, and begin the SAVE C instruction when the tape is at #10.)

Once the program is on the tape, do not rewind or fast forward. Remove the cassette and put it in an ordinary audio cassette recorder. (If it has a counter, note the setting.) Type NEW to clear the computer's memory. Then type in the following short program:

```
10 *START
20 ,     S0:25
30 ,     PA:20
40 ,     S0:0
50 ,     PA:340
60 ,     C:#C=#C+1
70 ,     J(#C<120):*START
80 ,     E:
```

Now, turn up the volume on your TV, and type RUN. At the same time that you hit RETURN to start the program going, push the buttons on your audio recorder that will start it recording. The program will produce a distinctive high-pitched tone once every six seconds, 120 times. When it finishes, rewind your tape (still in your audio machine). If it has a counter, go to the setting where the main program ends and the six-second interval sounds begin. If you don't have a counter, approximate and play the tape until you find that point. (The main program makes a distinctive noise. It's easy to tell when it ends.)

Next, in the same sequence that you entered them in the main program, speak the words aloud and record them on your audio recorder. First record your first word, two or three times in succession for clarity. Push STOP. Push PLAY. Listen for the next high-pitched tone. Push STOP. Push RECORD. Say your second word aloud, two or three times, and so on until you've gone through your whole list. (If you have more than one word to say, and hence need more than six seconds, just add time to line 50 above (60=1 second) and the same amount of time to line 1820 in the main program.)

After you've recorded your words, put the cassette back in the ATARI program recorder. Rewind to the beginning. Reset the counter to zero. Advance to #10. Type NEW, and load the revised program (LOAD C, etc.).

Once the program is loaded, leave the Play button on. Type RUN, and test what you have done.

The heart of the program is the subsection labeled *TPAE, which turns the cassette machine on for six seconds, then turns it off. For that six seconds, the screen will read "WAIT." Then it will say "ANSWER NOW," and wait for your answer. (If you begin typing before "ANSWER NOW," most of what you type will be lost, and you'll have to delete that particular response and type it again.)

If you made a mistake in your audio recording (for instance, skipping a word or not pronouncing a word clearly enough), it's easy to correct using your audio recorder. Just advance to the spot where you made the mistake, and record your correction over it — being careful to listen for and not record over the high-pitched cue sounds.

As you go through your test, note the number on the counter at the end of each game and later insert those numbers as the value of #Z in lines 400 (game 1), 620 (game 2), 890 (game 3), 1140 (game 4), 1400 (game 5), and 1660 (game 6). Then save this revised program (SAVE C, etc.) on the same tape, starting again at counter #10. (It will take up the same space it did before.) If you want to play the games out of sequence, those numbers will tell you where you should be on the tape. And if for some reason you get out of sync, you can use those numbers (or simply listen on an audio cassette machine or on the program recorder with TAPE:ON) to readjust.

To prepare a tape with 120 words should take you about an hour. That's about four weeks worth of spelling words in first, second, or third grade, or all the words in the Dr. Seuss learning-to-read classic *Hop on Pop*. And for your effort you have a tape that specifically meets the needs of your family or your class. □

Documentation

Lines 10 - 140 — choice of games; putting screen in graphics mode

Lines 150 - 400 — answers for game 1

Lines 410 - 660 — answers for game 2

Lines 670 - 920 — answers for game 3

Lines 930 - 1180 — answers for game 4

Lines 1190 - 1440 — answers for game 5

Lines 1450 - 1700 — answers for game 6

Lines 1710 - 1780 — beginning instructions for player

Lines 1790 - 1880 — turning the tape on for six seconds

Lines 1890 - 1940 — comparing the response with the right answer

Lines 1950 - 2050 — you get three tries

Lines 2060 - 2190 — after you make three mistakes in a row, the correct answer appears on the screen, and the program moves on to the next question

Lines 2200 - 2440 — if the answer is correct, as soon as you hit RETURN, that correct answer appears in large letters in the middle of the screen and a little tune is played before moving on to the next question

Lines 2450 - 2510 — after 20 words, the score (number right) is displayed

Lines 2510 - 2600 — you then have a choice of playing another game or stopping

```
10 J:*YES
20 *YES
30 ,     C:#5=0
40 ,     C:@B1373=16
50 ,     C:@B1374=2
60 ,     WRITE:5,K
70 ,     T:WHICH GAME DO YOU WANT TO PL
AY? (TYPE 1,2,3,4,5,6)
80 ,      A:
90 ,     M:1,2,3,4,5,6
100 ,     TN:WHAT?
110 ,     PAN:200
120 ,     JN:*YES
130 ,     JM:*ONE,*TWO,*THREE,*FOUR,*FI
VE,*SIX
140 ,     E:
150 *ONE
160 ,     C(#C=0):$WORD=UP
170 ,     C(#C=1):$WORD=PUP
180 ,     C(#C=2):$WORD=CUP
190 ,     C(#C=3):$WORD=IS
200 ,     C(#C=4):$WORD=IN
210 ,     C(#C=5):$WORD=ON
220 ,     C(#C=6):$WORD=MOUSE
230 ,     C(#C=7):$WORD=HOUSE
240 ,     C(#C=8):$WORD=ALL
250 ,     C(#C=9):$WORD=TALL
260 ,     C(#C=10):$WORD=SMALL
270 ,     C(#C=11):$WORD=WE
280 ,     C(#C=12):$WORD=ARE
290 ,     C(#C=13):$WORD=WALL
300 ,     C(#C=14):$WORD=BALL
310 ,     C(#C=15):$WORD=FALL
320 ,     C(#C=16):$WORD=OFF
330 ,     C(#C=17):$WORD=THE
340 ,     C(#C=18):$WORD=PLAY
350 ,     C(#C=19):$WORD=DAY
360 ,     C:#Z=111
370 ,     C:#A=1
380 ,     J(#C=0):*START
390 ,     J(#C)0):*TAPE
400 ,     E:
410 *TWO
420 ,     C(#C=0):$WORD=NIGHT
430 ,     C(#C=1):$WORD=FIGHT
440 ,     C(#C=2):$WORD=HE
450 ,     C(#C=3):$WORD=ME
460 ,     C(#C=4):$WORD=AFTER
470 ,     C(#C=5):$WORD=HIM
480 ,     C(#C=6):$WORD=SEE
490 ,     C(#C=7):$WORD=BEE
500 ,     C(#C=8):$WORD=THREE
510 ,     C(#C=9):$WORD=TREE
520 ,     C(#C=10):$WORD=NOW
530 ,     C(#C=11):$WORD=FISH
540 ,     C(#C=12):$WORD=HOW
550 ,     C(#C=13):$WORD=CAN
560 ,     C(#C=14):$WORD=THAT
570 ,     C(#C=15):$WORD=RED
580 ,     C(#C=16):$WORD=THEY
590 ,     C(#C=17):$WORD=CALL
600 ,     C(#C=18):$WORD=BED
610 ,     C(#C=19):$WORD=AM
620 ,     C:#Z=150
630 ,     C:#A=2
640 ,     J(#C=0):*START
650 ,     J(#C)0):*TAPE
660 ,     E:
670 *THREE
680 ,     C(#C=0):$WORD=AND
690 ,     C(#C=1):$WORD=PAT
700 ,     C(#C=2):$WORD=SAT
710 ,     C(#C=3):$WORD=HAT
720 ,     C(#C=4):$WORD=CAT
730 ,     C(#C=5):$WORD=BAT
740 ,     C(#C=6):$WORD=NO
750 ,     C(#C=7):$WORD=SIT
760 ,     C(#C=8):$WORD=DO
770 ,     C(#C=9):$WORD=SAD
780 ,     C(#C=10):$WORD=DAD
790 ,     C(#C=11):$WORD=BAD
800 ,     C(#C=12):$WORD=HAD
810 ,     C(#C=13):$WORD=VERY
820 ,     C(#C=14):$WORD=WHAT
830 ,     C(#C=15):$WORD=THING
840 ,     C(#C=16):$WORD=SING
850 ,     C(#C=17):$WORD=SONG
860 ,     C(#C=18):$WORD=LONG
870 ,     C(#C=19):$WORD=YOU
880 ,     C:#A=3
890 ,     C:#Z=188
900 ,     J(#C=0):*START
910 ,     J(#C)0):*TAPE
920 ,     E:
930 *FOUR
940 ,     C(#C=0):$WORD=SAY
950 ,     C(#C=1):$WORD=WALK
960 ,     C(#C=2):$WORD=LIKE
970 ,     C(#C=3):$WORD=TALK
980 ,     C(#C=4):$WORD=HOP
990 ,     C(#C=5):$WORD=POP
1000 ,     C(#C=6):$WORD=TOP
1010 ,     C(#C=7):$WORD=STOP
1020 ,     C(#C=8):$WORD=MUST
1030 ,     C(#C=9):$WORD=NOT
1040 ,     C(#C=10):$WORD=BROWN
1050 ,     C(#C=11):$WORD=SIDE
1060 ,     C(#C=12):$WORD=DOWN
1070 ,     C(#C=13):$WORD=WHERE
1080 ,     C(#C=14):$WORD=THERE
1090 ,     C(#C=15):$WORD=OUT
1100 ,     C(#C=16):$WORD=OF
1110 ,     C(#C=17):$WORD=TOWN
1120 ,     C(#C=18):$WORD=BACK
1130 ,     C(#C=19):$WORD=BLACK
1140 ,     C:#Z=225
1150 ,     C:#A=4
1160 ,     J(#C=0):*START
1170 ,     J(#C)0):*TAPE
1180 ,     E:
1190 *FIVE
1200 ,     C(#C=0):$WORD=CAME
1210 ,     C(#C=1):$WORD=WITH
1220 ,     C(#C=2):$WORD=SNACK
1230 ,     C(#C=3):$WORD=EAT
1240 ,     C(#C=4):$WORD=WITH
1250 ,     C(#C=5):$WORD=JUMP
1260 ,     C(#C=6):$WORD=BUMP
1270 ,     C(#C=7):$WORD=FAST
```

```
1280 ,     C(#C=8):$WORD=PAST
1290 ,     C(#C=9):$WORD=WENT
1300 ,     C(#C=10):$WORD=TENT
1310 ,     C(#C=11):$WORD=SENT
1320 ,     C(#C=12):$WORD=INTO
1330 ,     C(#C=13):$WORD=WET
1340 ,     C(#C=14):$WORD=GET
1350 ,     C(#C=15):$WORD=DOG
1360 ,     C(#C=16):$WORD=HELP
1370 ,     C(#C=17):$WORD=YELP
1380 ,     C(#C=18):$WORD=TOMORROW
1390 ,     C(#C=19):$WORD=THEY
1400 ,     C:#Z=258
1410 ,     C:#A=5
1420 ,     J(#C=0):*START
1430 ,     J(#C>0):*TAPE
1440 ,     E:
1450 *SIX
1460 ,     C(#C=0):$WORD=HILL
1470 ,     C(#C=1):$WORD=WILL
1480 ,     C(#C=2):$WORD=WENT
1490 ,     C(#C=3):$WORD=STILL
1500 ,     C(#C=4):$WORD=FATHER
1510 ,     C(#C=5):$WORD=MOTHER
1520 ,     C(#C=6):$WORD=SISTER
1530 ,     C(#C=7):$WORD=BROTHER
1540 ,     C(#C=8):$WORD=ONE
1550 ,     C(#C=9):$WORD=MY
1560 ,     C(#C=10):$WORD=OTHER
1570 ,     C(#C=11):$WORD=READ
1580 ,     C(#C=12):$WORD=LITTLE
1590 ,     C(#C=13):$WORD=BIT
1600 ,     C(#C=14):$WORD=WORD
1610 ,     C(#C=15):$WORD=IF
1620 ,     C(#C=16):$WORD=IT
1630 ,     C(#C=17):$WORD=BIG
1640 ,     C(#C=18):$WORD=CONSTANTINOPL
E
```

```
1650 ,     C(#C=19):$WORD=TIMBUKTU
1660 ,     C:#Z=289
1670 ,     C:#A=6
1680 ,     J(#C=0):*START
1690 ,     J(#C)0):*TAPE
1700 ,     E:
1710 *START
1720 ,     T:WIND TAPE TO NUMBER #Z.THE
N PUSH DOWN PLAY BUTTON AND LEAVE IT D
OWN. WHEN  YOU ARE READY, TYPE R.
1730 ,     A:
1740 ,     M:R
1750 ,     TN:WHAT?
1760 ,     JN:*START
1770 ,     JM:*TAPE
1780 *TAPE
1790 ,     T:R
1800 ,     T:WAIT
1810 ,     TAPE:ON
1820 ,     PA:330
1830 ,     TAPE:OFF
1840 ,     T:R
1850 ,     T:ANSWER NOW
1860 ,     T:
1870 ,     J:*ANSWER
1880 ,     E:
1890 *ANSWER
1900 ,     A:
1910 ,     M:$WORD
1920 ,     JN:*WRONG
1930 ,     JM:*RIGHT
1940 ,     E:
1950 *WRONG
1960 ,     C:#T=#T+1
1970 ,     SO:13
1980 ,     PA:30
1990 ,     SO:7
2000 ,     PA:50
```

```
2010 ,     50:0
2020 ,     J(#T=3):*THIRDWRONG
2030 ,     T:TRY AGAIN.
2040 ,     J:*ANSWER
2050 ,     E:
2060 *THIRDWRONG
2070 ,     C:#T=0
2080 ,     T:
2090 ,     T:THE RIGHT ANSWER IS:
2100 ,     T:        $WORD
2110 ,     PA:200
2120 ,     C:#C=#C+1
2130 ,     J(#A=1):*ONE
2140 ,     J(#A=2):*TWO
2150 ,     J(#A=3):*THREE
2160 ,     J(#A=4):*FOUR
2170 ,     J(#A=5):*FIVE
2180 ,     J(#A=6):*SIX
2190 ,     E:
2200 *RIGHT
2210 ,     WRITE:5,K
2220 ,     WRITE:5,
2230 ,     WRITE:5,
2240 ,     WRITE:5,
2250 ,     WRITE:5,     $WORD
2260 ,     T:
2270 ,     T:CORRECT!
2280 ,     C:#5=#5+1
2290 ,     C:#C=#C+1
2300 ,     50:13
2310 ,     PA:20
2320 ,     50:20
2330 ,     PA:20
2340 ,     50:25
2350 ,     PA:30
2360 ,     50:0
2370 ,     J(#C=20):*END
2380 ,     J(#A=1):*ONE
2390 ,     J(#A=2):*TWO
2400 ,     J(#A=3):*THREE
2410 ,     J(#A=4):*FOUR
2420 ,     J(#A=5):*FIVE
2430 ,     J(#A=6):*SIX
2440 ,     E:
2450 *END
2460 ,     WRITE:5,K
2470 ,     WRITE:5,  YOU GOT #5 RIGHT.
2480 ,     WRITE:5,  CONGRATULATIONS
2490 ,     J:*AGAIN
2500 ,     E:
2510 *AGAIN
2520 ,     T:YOU JUST FINISHED GAME #A.
2530 ,     T:WANT TO PLAY AGAIN?
2540 ,     A:
2550 ,     M:YES,NO
2560 ,     CY:#C=0
2570 ,     JM:*YES,*NO
2580 ,     E:
2590 *NO
2600 ,     E:
```

•

CHECKSUM DATA

(See p. 58)

```
10 DATA 416,327,446,310,52,227,648,155
,223,137,21,207,784,189,92,4234
160 DATA 196,135,126,202,168,178,658,6
57,90,322,580,44,268,580,515,4639
310 DATA 528,253,273,611,250,887,750,7
58,756,183,139,622,618,163,172,6963
460 DATA 651,104,87,74,630,322,287,553
,285,229,599,271,623,554,226,5495
610 DATA 30,886,745,752,750,205,657,82
,123,102,95,94,97,199,138,4955
760 DATA 197,82,228,229,218,610,599,49
7,591,614,605,346,765,941,746,7268
910 DATA 744,199,380,131,330,330,335,1
37,149,280,610,616,281,634,371,5527
1060 DATA 401,655,650,522,181,462,349,
566,62,944,849,587,368,584,488,7668
1210 DATA 568,464,215,577,599,586,551,
574,598,436,437,417,527,483,462,7494
1360 DATA 408,463,186,472,79,951,854,5
92,373,324,547,580,589,526,658,7602
1510 DATA 717,737,44,279,87,628,372,97
2,480,436,176,194,463,32,152,5769
1660 DATA 101,965,866,604,378,600,334,
369,725,342,512,184,599,837,191,7607
1810 DATA 96,994,299,835,47,432,575,38
9,777,372,34,526,499,388,579,6842
1960 DATA 385,768,749,582,722,528,605,
914,550,364,530,994,412,42,522,8667
2110 DATA 968,257,659,706,821,610,570,
719,371,537,725,321,322,323,247,8156
2260 DATA 416,420,362,267,745,723,747,
725,754,730,542,505,670,717,825,9148
2410 DATA 614,574,723,375,293,736,813,
645,172,374,485,816,25,366,272,7283
2560 DATA 804,574,382,98,377,2235
```

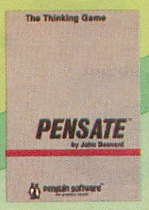
•

penguin software™

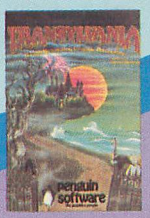
the graphics people

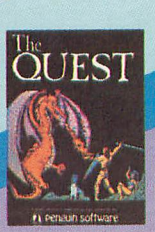
The Next Generation Is Here!

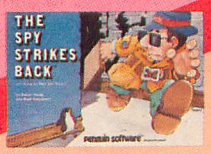
Strategy

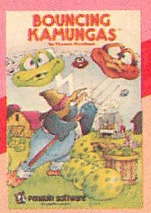


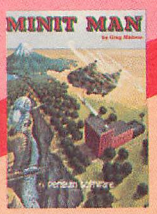
Exciting Adventures



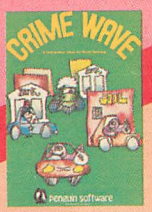








Arcade Fun



Fantasy Role Playing

Games for Apple, Atari, Commodore, and IBM computers

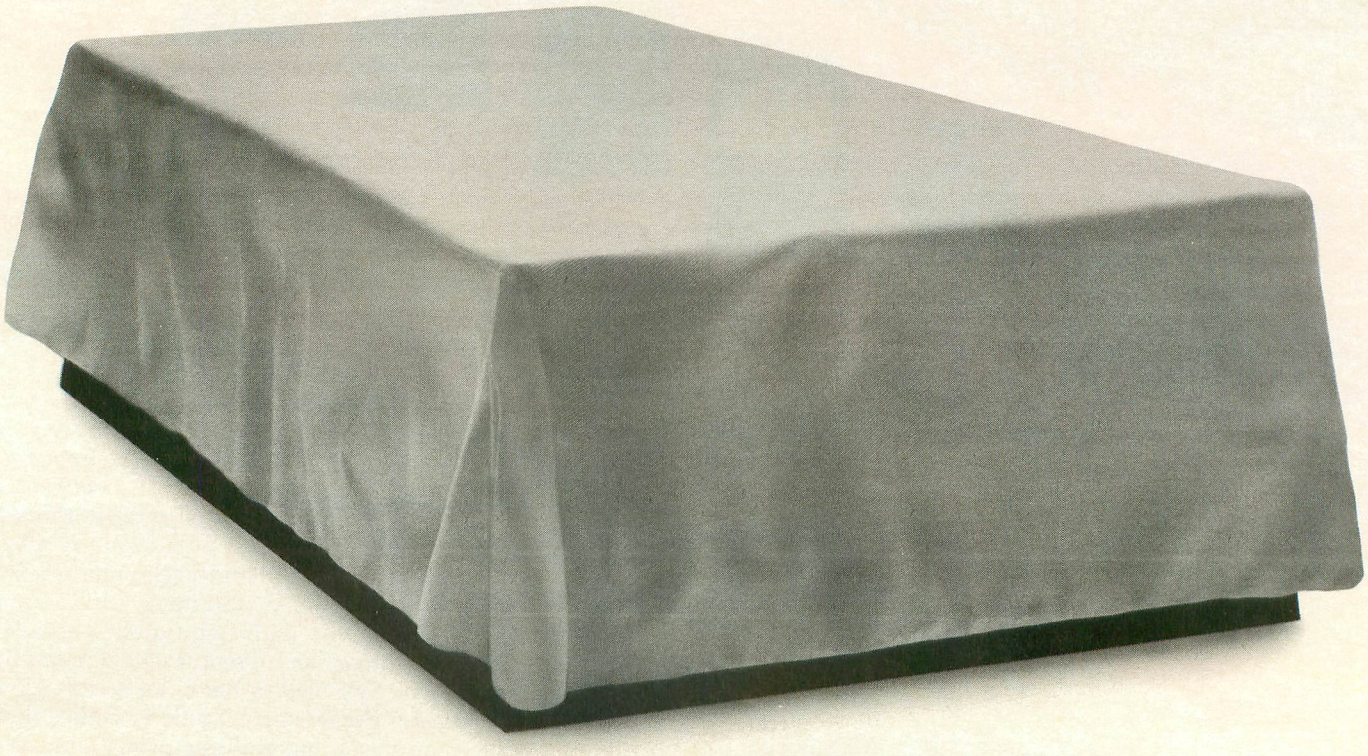
At Penguin, we don't rely on the same old game formulas, adding to the mish-mash of look-alike games already on the market. We look for creativity, originality, and innovation. Games like **Minit Man,** challenging you with two types of arcade action simultaneously; **Coveted Mirror,** an adventure with animation and arcade games throughout; **The Spy Strikes Back,** an arcade game with strategy and a touch of adventure; **Pensate** and **Tactic 9,** games that make you think; and **Expedition Amazon,** an intriguing and humorous fantasy game.

And Each Is Only $19.95! More Fun For Your Money From Penguin Software

Write for a free catalog

830 Fourth Avenue, Box 311 Dept. G Geneva, IL 60134 For information - Call (312) 232-1984 Dealer orders only - Call (800) 323-0116

Looks like a
Drives like a
Parks like a



For additional information, call 1-800-334-4387. In California, call 1-800-544-4387.

Ferrari. Rolls. Beetle.

Introducing the all-new 1984 Indus GT™ disk drive. The most advanced, most complete, most handsome disk drive in the world.

A flick of its "Power" switch can turn an Atari into a Ferrari.

Or an Apple into a Red Hot Apple.

Ask your favorite computer dealer to unveil the remarkable Indus GT for you. Test drive it. Then, get ready to drive home a world-class driving machine.

Looks Like a Ferrari.

The Indus GT is incredibly sleek. Only 2.65" high! Its smoked-glass dust-cover front-loading front-end width is a perfectly proportioned 6.27." Eye-catching. Distinctively European-Gran styled.

Touch its LED-lit CommandPost™ function controls. Lay a finger on its AccuTouch™ buttons. Feel its elegant, fluid form.

Admire its smooth, sculptured, masculine lines. Wonder at its ingenius slimline engineering.

Marvel at how responsive it makes *every* Atari or Apple home computer.

Drives Like a Rolls.

Nestled into its soundproofed 10.4" length chassis is the quietest and most powerful disk drive power system money can buy. At top speed (which it can handle all day long), it's virtually unhearable. Whisper quiet.

Flat out, the GT will drive your Atari track-to-track 0-39 in *less than one second!* Increasing data transfer an incredible 400%.

Built-in, low-friction GT CruiseControl metal band head-positioning assures Apple and Atari owners of trouble-free performance. So you can finish every project you enter.

And quietly tucked into every 1984 Indus GT disk drive is the exclusive GT DrivingSystem™ of software programs (included as standard equipment):

The GT DrivingSystem allows you to shift smoothly into world-class word processing with the GT *Estate WordProcessor.*™ And your dealer will describe the two additional new GT Driving System programs that allow you to advance your driving skills. Instantly. Smoothly. Easily.

The Indus GT and GT DrivingSystem are covered with the GT PortaCase.™ Which doubles as a 20-disk storage file when you're at home. Or as a portable case to flaunt your good taste and judgement when you travel. Safely. Smartly.

Parks Like a Beetle.

And it's sticker-priced like a Beetle, too. $449 for Atari. $329 for Apple.

The all-new, 1984 Indus GT.

Its small size makes it easy to park. And its low price makes it easy to buy.

So see and test drive the GT at your nearest computer dealer today.

The drive will be well worth it.

INDUS GT™

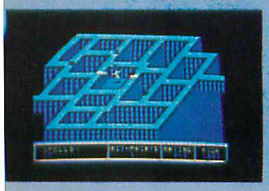
The all-new 1984 Indus GT Disk Drive.

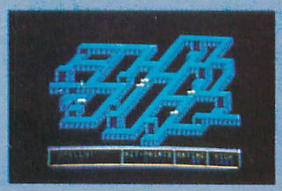
The cover comes off at your computer dealer's showroom.

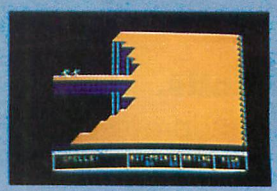
ZOMBIES©

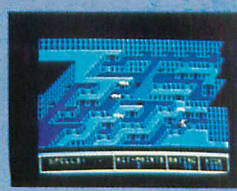
*Atari is a trademark of Atari, Inc.

Scrolling 3D graphics, on-line instructions, one or two player cooperative, seven different dungeons, 74 different screens, high score save to disk, full sound and color, zombies, poisonous snakes, giant spiders, evil orbs, scrolls, talismans, magic spells, lost crowns and spectacular underground scenery.







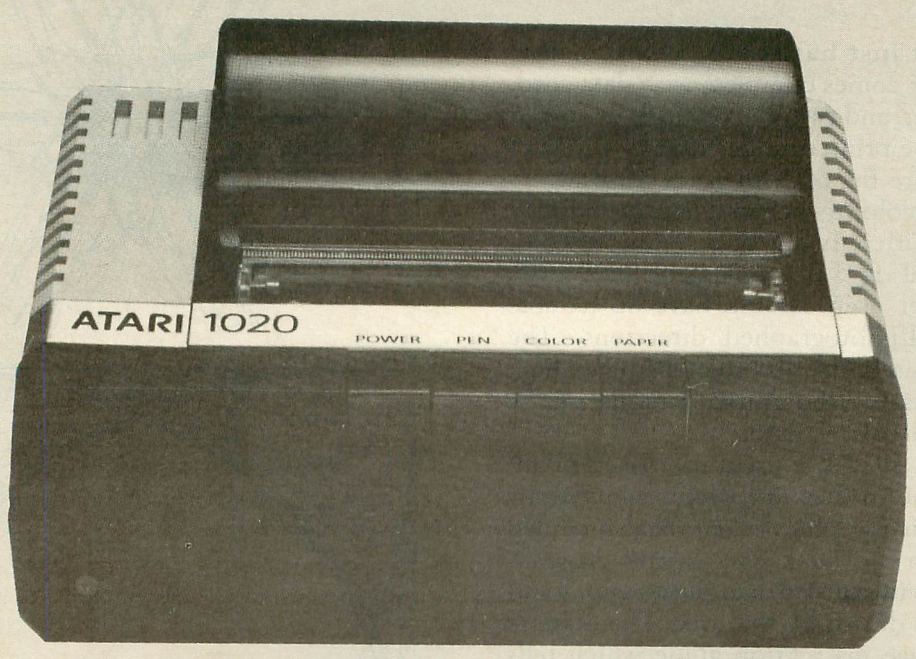


BRAM Inc.
18779 Kenlake Place N.E.
Seattle, Washington 98155
(206) 486-8428

A fast action arcade fantasy for Atari's.*

By Mike Edwards from BRAM Inc.

Disk and Tape
$34.95



THE ATARI 1020 COLOR PRINTER

by Tom Hudson

ATARI 1020 COLOR PRINTER
ATARI, Inc.
P.O. Box 60657
Sunnyvale, CA 94086
$299.00

If you're looking for low-cost, high-quality graphics hardcopy, the ATARI 1020 Color Printer may be for you. This compact printer is capable of four-color text and graphics printing with resolution much higher than that of the ATARI computer graphics displays.

Packed with features.

To be honest, I was disappointed at first by the 1020's small size. It's slightly smaller than the 410 Program Recorder, and uses standard four-and-a-half inch roll paper.

All my fears about the 1020 disappeared the first time I saw it plot. Unlike a conventional dot-matrix printer, the 1020 uses four special ball-point cartridges and *draws* each letter when printing text. The fact that the letters are drawn allows the 1020 to print characters in 64 *different sizes* with 20, 40 or 80 characters per line.

The 1020 is able to plot in four colors, selected either from the printer's control panel or through program control. The printer comes with a spare set of pens, an unexpected extra.

The printer is also able to print international characters, making it compatible with all the new ATARI computer systems.

In addition to all the above features, the printer has the ability to print text in *four* different directions, even upside down! This allows easy labeling of axes on charts or graphs plotted with the printer.

Not just text.

The text-handling features of the 1020 Printer are impressive but the real beauty of this printer is the ability to plot high-resolution, four-color graphics.

The 1020's graphics mode has a resolution of 0-480 in the X axis and -999 to 999 in the Y axis. This is a total resolution of 481 by 1999 or over 961,500 individual points. Watch out, GRAPHICS 8 fans!

Of course, the graphics mode of the 1020 allows you to draw lines from one point to another and even allows multiple plots with one PRINT statement. These lines are normally solid, but the 1020 can draw dotted lines with 15 different length dots. A simple three-character command does the trick.

Another unexpected feature of the 1020 is its ability to draw X and Y axes of graphs with automatic scale marks. This simplifies graph generation immeasurably.

Not just hardware.

The 1020 Printer comes complete with data cord, power supply, paper, and two sets of black, blue, red and green pens. The printer plugs into any I/O port and does not require the 850 Interface Module.

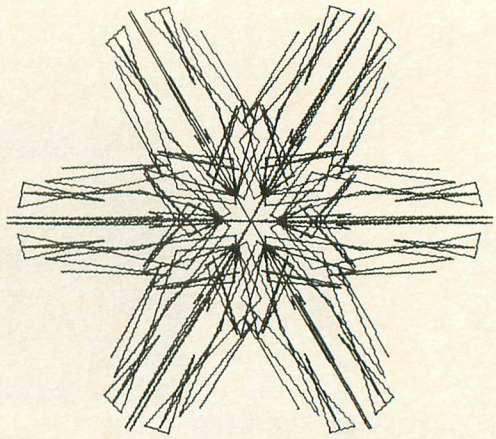
An 18-page, full color owner's manual is included with the printer. With only one or two minor exceptions, this manual is very complete and even beginning computer users should have no trouble following the well-photographed directions. The owner's manual contains two program listings which demonstrate the use of the graphics mode of the 1020, but there's an extra bonus.

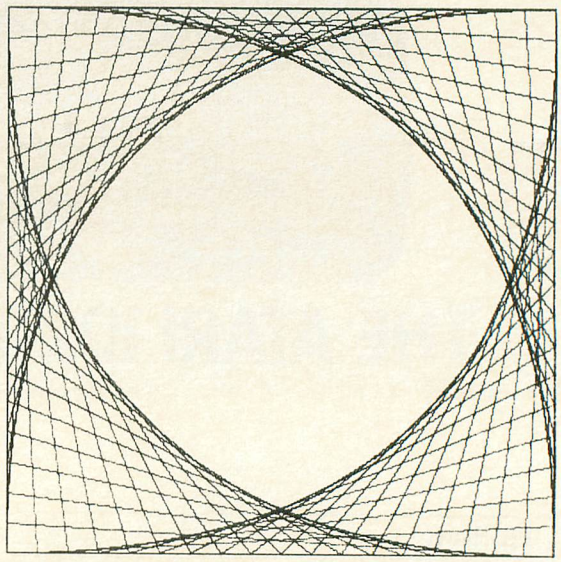
Packed with the 1020 is a program cassette containing several demonstration programs which draw a variety of interesting graphic designs. Also included on the cassette is a "Joystick Sketchpad" program which, when used with a disk drive, allows saving and loading of pictures drawn by the user. The cassette comes with a 3-page instruction guide which tells how to use the programs and transfer them to disk.

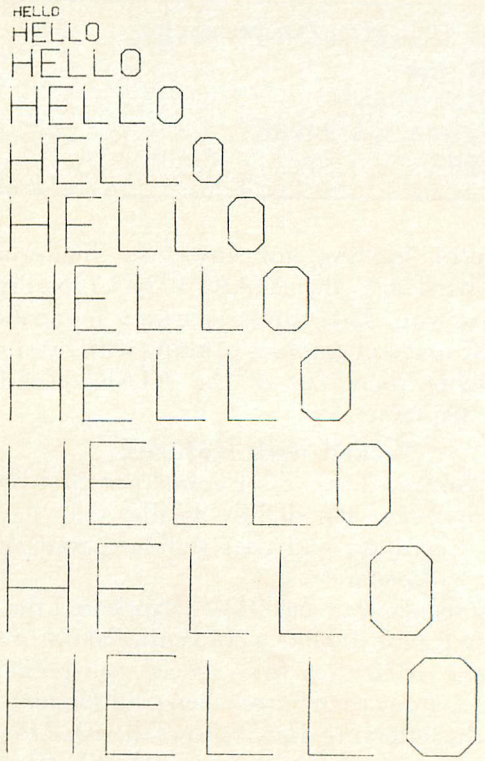
A product to be proud of.

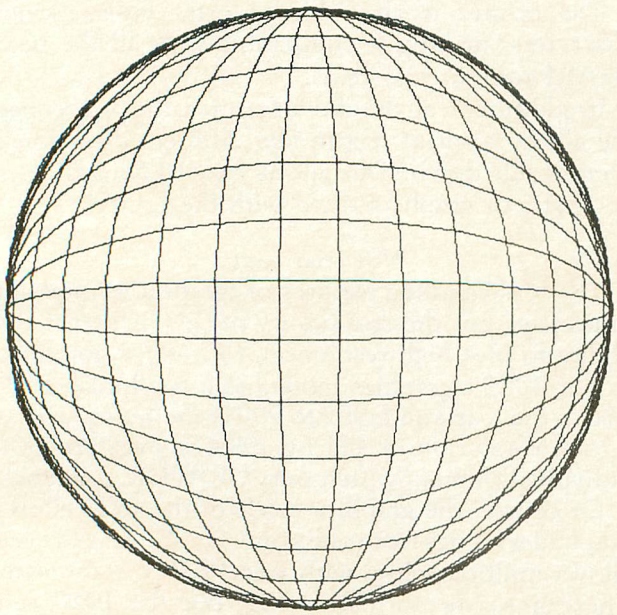
The ATARI 1020 Printer is a fine example of the difference between a hardware item and a *product*. The inclusion of excellent user manuals and a demonstration program cassette make this package complete.

ATARI should be very proud of the 1020 Printer. And you can bet that it will be supported here in the pages of **A.N.A.L.O.G.** in the future. □









mUSE

A Basic Memory Monitor

32K Disk

by Brian Moriarty

Suppose you've just finished a long and complicated piece of BASIC code. Trembling with anticipation, you SAVE the program out to disk, type RUN and press RETURN. Your 48K ATARI hesitates for a moment as it initializes dozens of strings and multi-dimensional arrays. The screen blinks as it tries to enter GRAPHICS 24 — and suddenly jumps back to mode 0 with an ERROR 2 (Memory Insufficient) message. You type PRINT FRE(0) and discover that there isn't enough memory left over to accomodate a mode 24 screen. It's Optimization Time!

There are all sorts of things you can do to a BASIC program to save memory. Removing REM statements, cleaning out the variable tables and eliminating constants are just a few of the techniques available to the RAM-hungry programmer. But the only way to check the effectiveness of your cramming is to use the FRE(0) function, which tells you absolutely nothing about *where* your program needs tightening.

mUse (Memory Usage) is a co-resident, machine-language utility that takes up where FRE(0) leaves off. It analyzes your BASIC program and shows you not only how much RAM it is using, but also *how* it is using it. With **mUse**, you can improve the optimization of your BASIC code by concentrating on the things that take up the most space.

Design considerations.

A BASIC utility like **mUse** is tricky to implement for two reasons: you need a safe place to put it, and a convenient way to use it. I wanted to avoid USR calls, extra DATA statements and page 6 because so many other BASIC utility programs use these facilities. I also wanted the routine to be completely transparent to the user until it was actually needed. **mUse** meets all of these requirements by exploiting the AUTORUN.SYS feature of DOS 2.0S.

When you turn on your ATARI, the operating system checks to see if a cartridge is inserted and, if so, whether or not the Disk Option Byte at location 49149 ($BFFC) is set. This byte tells the OS whether or not it's okay to boot a disk. Language cartridges like BASIC and Logo have the option byte set: game cartridges like **Star Raiders** do not.

Assume that the disk in drive #1 contains a standard ATARI DOS.SYS file. As soon as DOS.SYS is booted into memory, the disk directory is scanned to see if there is a file named AUTORUN.SYS. If present, the file is loaded and executed *before* control is passed to the BASIC cartridge. That means you can use AUTORUN.SYS to reserve blocks of memory and to "steal" important operating system vectors to suit your own devious purposes.

mUse does both. It loads into memory just above DOS.SYS and protects itself by changing the low memory pointer MEMLO at address $2E7. Then it alters the DOSINI vector ($0C) so that your ATARI will execute the **mUse** routine whenever you hit the SYSTEM RESET key. Finally, control is allowed to pass into the BASIC cartridge, which initializes itself to operate above the RAM block reserved by **mUse**. You can't overwrite **mUse** with BASIC unless you deliberately alter the value of MEMLO, or start POKEing around inside **mUse**'s reserved memory area.

Typing the program.

Type **Listing 1** into your computer exactly as you see it printed. Be especially careful with the DATA statements in Lines 1000-1340, as these constitute the actual machine code for the **mUse** routine. Then LIST the program out to disk and use D:CHECK2 to verify the accuracy of your typing. When everything is perfect, re-ENTER the BASIC program, type RUN and hit RETURN.

The line number of each DATA statement will appear as the byte values are verified. Bad data or missing line numbers will produce an appropriate error message. Otherwise you will be prompted to insert a disk containing DOS 2.0S into drive #1. Make sure this disk has enough room on it to fit **mUse** (about 7 sectors), and that there is no other AUTORUN.SYS file on the disk. Everything okay? Then press the START key and an AUTORUN

version of **mUse** will be written out to your disk. Make sure you've SAVEd a copy of the BASIC program; you can use it to make fresh copies of **mUse**.

Getting used to mUse.

Put the disk with the AUTORUN.SYS file in drive #1, power down your ATARI and turn it back on. After DOS boots in, you should be greeted with a "mUse 1.0 OK" message along with the familiar BASIC "READY" prompt. Type PRINT FRE(0) and you will find that **mUse** has stolen about 1300 bytes from your BASIC workspace. Don't worry, though. **mUse** is needed only as a diagnostic tool; you can get rid of it after you're done optimizing your program.

Now brace yourself for a cheap thrill. Put your finger on the SYSTEM RESET key and give it a firm, confident press. Surprise! Instead of a simple "READY" prompt, the screen is filled with all sorts of interesting statistics about your BASIC program. Let's go through them one at a time:

#Lines: The total number of lines in your BASIC program.

#Stmnts: The total number of statements in your program. This number will be equal to # **Lines** if you put just one statement on every line. It will be larger if you used multiple statements per line. By dividing # **Stmnts** by # **Lines**, you can estimate the density of your program listings.

 # REM chars: This figure gives you an idea of how much program space is being wasted by REM statements. A simple REM counts as one byte; each text character after the REM adds an additional byte. Reducing the number and size of REM statements is one of the most effective ways to cut the size of your BASIC code.

Program RAM: The size of your tokenized BASIC program, not including the RAM used by variables, strings and arrays. **Program RAM** lets you monitor the actual memory efficiency of the token program, regardless of variable name length and other unrelated factors.

 # Vars: The total number of simple (scalar) variables in your program, including "dead" variables that have been removed from your program but continue to take up space. Each variable requires eight bytes in the Variable Value Table and one or more bytes in the Variable Name Table. Obviously, it pays to keep the number of variables to a minimum.

 # Arrays: The number of one- and two-dimensional numeric arrays in your program, both DIMensioned, unDIMensioned and deleted.

Array RAM: Arrays really eat up RAM, as this number readily proves. Each array element requires six bytes of binary-coded decimal. That means a one-dimensional array with 10 elements takes up 60 bytes, while a 2-D array DIMed to (10,10) requires a whopping 600 bytes! Note that **Array RAM** includes only those arrays which have been officially DIMensioned.

 # Strings: The number of DIMensioned, unDIMensioned and deleted strings in your program.

String RAM: Strings are more forgiving than arrays since they use only one byte per DIM allotment. The value shown by **String RAM** does not include strings which have not been DIMensioned.

Varname RAM: The total number of bytes used by variable names, including strings and arrays. You can save space by keeping the length of variable names to a minimum, often at the cost of reduced readability.

Free RAM: The same as FRE(0), except that it includes the space taken up by the last immediate mode line.

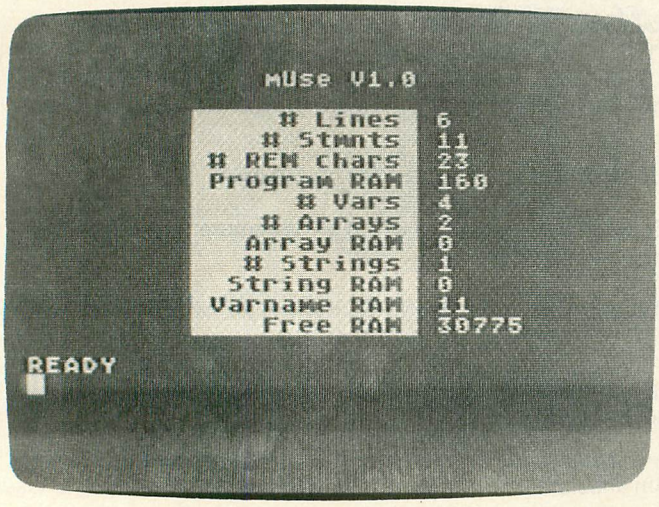
(continued on page 114)

A short demonstration.

Type in the short BASIC program in **Listing 2** but do not RUN it. After it's typed, press SYSTEM RESET and check to be sure **mUse** is showing the following values:

 # Lines: 6
 # Stmnts: 11
 # REM Chars: 23
Program RAM: 160
 # Vars: 4
 # Arrays: 2
Array RAM: 0
 # Strings: 1
String RAM: 0
Varname RAM: 11
Free RAM: (varies)

Why are **Array RAM** and **String RAM** equal to zero? Remember that **mUse** doesn't count arrays and strings that haven't been DIMensioned, and that doesn't happen until the DIM statements are executed. So type RUN and, after the READY prompt appears, hit SYSTEM RESET again. You should now see the 660 bytes reserved for the arrays A() and B(), with 14 bytes reserved for S$.



Screen format.

Play around with the demo program by adding or deleting REM characters, lines and individual statements. Try deleting a variable and see how BASIC retains old variables, even when you don't want them anymore. To clean up unused variables, LIST the demo out to disk, type NEW and re-ENTER the program. **mUse** will now return the correct number of variables in the program. When you're done playing around, type NEW and hit RESET to see what a clean slate looks like.

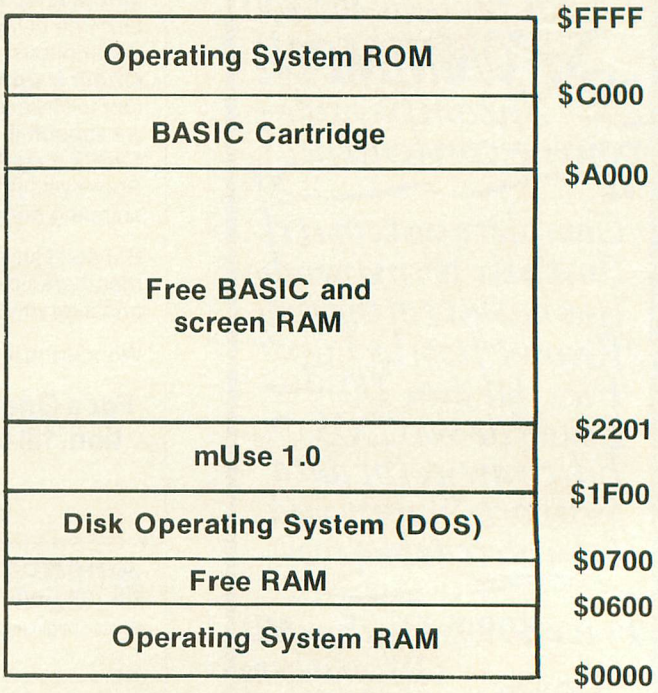
Beware of DOS!

The only BASIC command that will definitely affect the operation of **mUse** is the DOS command. When you type DOS, BASIC performs an indirect jump to a routine that loads ATARI's DUP.SYS file right over the space occupied by **mUse.** DUP.SYS will work just fine; you can even use option "B" to return to the BASIC cartridge. But don't hit the SYSTEM RESET key after calling DOS, or your ATARI will go sailing into never-never land! If you need to use **mUse** after running DUP.SYS, you're better off re-booting the system.

mUse starts at location $1F00, so it's safe to use with **OS/A+** by Optimized Systems Software. Just change the filenames in Line 230 of **Listing 1** from AUTORUN.SYS to MUSE.COM before you RUN the BASIC program. When you boot up into BASIC, type DOS to enter **OS/A+**, type MUSE and hit RETURN. The **mUse** file will load and run automatically. You can also use the STARTUP.EXC feature of **OS/A+** to execute the MUSE.COM file at boot time.

For more information.

You'll get more out of **mUse** if you familiarize yourself with the structure of ATARI BASIC. Chapter 10 of *De Re Atari* (available from the ATARI Program Exchange) includes a brief but useful summary of BASIC memory conservation techniques. Robert Howell's article "RAM Cram Techniques for Atari" (*Creative Computing*, August 1981) offers an informative and entertaining treatment of the subject. And Bill Wilkinson's *Atari BASIC Sourcebook* (published by COMPUTE! Books) delves into the deepest secrets of everybody's favorite ROM cartridge. □



mUse 1.0 Memory Map (48K System)

Listing 1.

```
100 REM ========================
110 REM mUse Memory Usage Monitor
120 REM Version 1.0
130 REM by Brian Moriarty
140 REM ANALOG Computing #14
150 REM ========================
160 ? "K↓Verifying DATA line: ";SUM=0:
RESTORE 1000
170 FOR LINE=1000 TO 1340 STEP 10
180 POSITION 23,1:? LINE;FOR I=1 TO 24
:READ BYTE:SUM=SUM+BYTE:NEXT I
190 READ CHECK:IF CHECK<>SUM THEN ? "B
ad DATA at line ";LINE:END
200 IF PEEK(183)+256*PEEK(184)<>LINE T
HEN ? "Line ";LINE;" missing!":END
210 NEXT LINE:? "↓Insert a disk with D
OS in Drive #1.":? "Press START to wri
te mUse file."
220 IF PEEK(53279)<>6 THEN 220
230 ? "↓Writing AUTORUN.SYS.":TRAP 290
:OPEN #2,8,0,"D:AUTORUN.SYS"
240 PUT #2,255:PUT #2,255:PUT #2,0:PUT
 #2,31:PUT #2,71:PUT #2,34
250 FOR LINE=1000 TO 1340 STEP 10
260 RESTORE LINE:FOR I=1 TO 24:READ BY
TE:PUT #2,BYTE:NEXT I:READ CHECK
270 NEXT LINE:PUT #2,224:PUT #2,2:PUT
#2,225:PUT #2,2:PUT #2,1:PUT #2,34
280 CLOSE #2:? "↓mUse disk OK.":END
290 ? "Disk I/O error #";PEEK(195)
300 END :REM * M/L PROGRAM DATA
1000 DATA 32,255,255,216,169,1,141,231
,2,169,34,141,232,2,169,11,133,82,169,
15,133,85,169,0,2846
1010 DATA 133,86,169,3,133,84,169,89,1
60,33,32,254,32,32,79,33,165,136,133,2
03,165,137,133,204,5643
1020 DATA 160,1,177,203,48,59,230,212,
208,2,230,213,200,177,203,133,208,200,
177,203,133,209,230,205,9664
1030 DATA 208,2,230,206,200,177,203,20
8,13,230,201,208,2,230,202,200,177,203
,201,155,208,243,164,209,13944
1040 DATA 196,208,208,222,24,165,208,1
01,203,133,203,144,195,230,204,208,191
,169,103,160,33,32,254,32,17770
1050 DATA 32,24,33,169,117,160,33,32,2
54,32,165,205,133,212,165,206,133,213,
32,24,33,169,131,160,20637
1060 DATA 33,32,254,32,165,201,133,212
,165,202,133,213,32,24,33,169,145,160,
33,32,254,32,216,56,23598
1070 DATA 165,140,229,136,233,3,133,21
2,165,141,229,137,133,213,32,24,33,32,
79,33,165,134,133,203,26735
1080 DATA 165,135,133,204,165,203,197,
136,208,9,165,204,197,137,208,3,76,99,
32,160,0,177,203,240,30191
1090 DATA 94,48,96,230,206,41,1,240,11
3,32,68,218,216,56,160,4,177,203,233,1
,133,212,200,177,33350
1100 DATA 203,233,0,133,213,56,200,177
,203,233,1,133,201,200,177,203,233,0,1
33,202,165,201,208,4,37062
1110 DATA 165,202,240,26,32,170,217,32
,182,221,32,68,218,165,201,133,212,165
,202,133,213,32,170,217,40710
1120 DATA 32,219,218,32,210,217,216,24
,165,212,101,208,133,208,165,213,101,2
09,133,209,76,82,32,230,44355
1130 DATA 205,208,23,230,207,41,1,240,
17,216,24,160,6,177,203,101,210,133,21
0,200,177,203,101,211,47859
1140 DATA 133,211,216,24,165,203,105,8
,133,203,165,204,105,0,133,204,76,196,
31,169,159,160,33,32,50927
1150 DATA 254,32,32,68,218,165,205,133
,212,32,24,33,169,173,160,33,32,254,32
,165,206,133,212,32,53936
1160 DATA 24,33,169,187,160,33,32,254,
32,165,208,133,212,165,209,133,213,32,
170,217,32,182,221,32,57184
1170 DATA 68,218,169,6,133,212,32,170,
217,32,219,218,32,27,33,169,201,160,33
,32,254,32,165,207,60223
1180 DATA 133,212,32,24,33,169,215,160
,33,32,254,32,165,210,133,212,165,211,
133,213,32,24,33,169,63252
1190 DATA 229,160,33,32,254,32,56,165,
132,229,130,133,212,165,133,229,131,13
3,213,32,24,33,169,243,66554
1200 DATA 160,33,32,254,32,56,173,229,
2,229,144,133,212,173,230,2,229,145,13
3,213,32,24,33,169,69626
1210 DATA 2,133,82,76,77,160,162,0,157
,68,3,152,157,69,3,169,11,157,66,3,169
,14,157,72,71745
1220 DATA 3,138,157,73,3,76,86,228,32,
170,217,32,230,216,160,255,200,177,243
,16,251,41,127,145,75021
1230 DATA 243,200,169,155,145,243,152,
162,0,157,72,3,138,157,73,3,165,243,15
7,68,3,165,244,157,78295
1240 DATA 69,3,169,9,157,66,3,32,86,22
8,32,68,218,216,96,162,28,169,0,149,20
1,202,16,251,80925
1250 DATA 96,109,85,115,101,32,86,49,4
6,48,32,32,32,29,155,160,160,160,160,1
60,163,160,204,233,83532
1260 DATA 238,229,243,160,32,160,160,1
60,160,163,160,211,244,237,238,244,243
,160,32,160,163,160,210,197,87896
1270 DATA 205,160,227,232,225,242,243,
160,32,160,208,242,239,231,242,225,237
,160,210,193,205,160,32,160,92526
1280 DATA 160,160,160,160,160,163,160,
214,225,242,243,160,32,160,160,160,160
,163,160,193,242,242,225,249,96879
1290 DATA 243,160,32,160,160,160,193,2
42,242,225,249,160,210,193,205,160,32,
160,160,160,163,160,211,244,101163
1300 DATA 242,233,238,231,243,160,32,1
60,160,211,244,242,233,238,231,160,210
,193,205,160,32,160,214,225,105820
1310 DATA 242,238,225,237,229,160,210,
193,205,160,32,160,160,160,160,198,242
,229,229,160,210,193,205,160,110417
1320 DATA 32,165,12,141,1,31,165,13,14
1,2,31,169,0,133,12,169,31,133,13,169,
1,141,231,2,112355
1330 DATA 169,34,141,232,2,169,47,160,
34,32,254,32,162,0,142,68,2,232,134,9,
76,0,160,125,114771
1340 DATA 29,109,85,115,101,32,49,46,4
8,32,79,75,155,0,0,0,0,0,0,0,0,0,0,0,1
15726
```

•

CHECKSUM DATA

(See p. 58)

```
100 DATA 218,776,141,491,523,233,448,7
7,66,881,783,489,818,34,671,6649
250 DATA 73,753,708,63,988,216,846,214
,526,920,800,417,238,672,504,7938
1090 DATA 983,342,680,773,645,457,230,
697,394,462,543,412,446,376,509,7949
1240 DATA 923,398,317,272,162,83,52,33
3,739,635,270,4184
```

•

Listing 2.

Demo program.

```
10 REM mUse V1.0 Demo Program
20 DIM A(10),B(10,10),S$(14)
30 X=0:Y=1:Z=Y:PI=3.14159
40 PRINT "THIS IS A TEST"
50 ? S$:? :?
60 END
```

Assembly Listing.

```
0100 ; ====================
0110 ; mUse V1.0
0120 ; by Brian Moriarty
0130 ; ANALOG Computing #14
0140 ; ====================
0150 ;
0160 ; BASIC addresses
0170 ;
0180 VNTP =  $82
0190 VNTD =  $84
0200 VVTP =  $86
0210 STMTAB = $88
0220 STARP = $8C
0230 MEMTOP = $90
0240 COLD =  $A000
0250 WARM =  $A04D
0260 ;
0270 ; OS equates
0280 ;
0290 BOOT? = $09
0300 DOSINI = $0C
0310 LMARGN = $52
0320 ROWCRS = $54
0330 COLCRS = $55
0340 COLDST = $0244
0350 RUNAD = $02E0
0360 HIMEM = $02E5
0370 MEMLO = $02E7
0380 ICCOM = $0342
0390 ICBADR = $0344
0400 ICBLEN = $0348
0410 ICAUX1 = $034A
0420 ICAUX2 = $034B
0430 CIOV =  $E456
0440 ;
0450 ; Floating point equates
0460 ;
0470 FR0 =   $D4
0480 INBUFF = $F3
0490 FASC =  $D8E6
0500 IFP =   $D9AA
0510 FPI =   $D9D2
0520 ZFR0 =  $DA44
0530 FMUL =  $DADB
0540 FMOVE = $DDB6
0550 ;
0560 ; Internal program equates
0570 ;
0580 ORIGIN = $1F00
0590 REMS =  $C9      ; # REM chars
0600 PNTR =  $CB      ; Z-page pointer
0610 STATS = $CD      ; # statements
0620 LNOFF = $D0      ; line offset
0630 STOFF = $D1      ; stmnt offset
0640 TEMP =  REMS     ; scratch pad
0650 SVARS = STATS    ; # simple vars
0660 ARRYS = $CE      ; # arrays
0670 STRGS = $CF      ; # strings
0680 ADIMS = LNOFF    ; array RAM use
0690 SDIMS = $D2      ; string RAM use
0700 ;
0710     *=  ORIGIN
0720 ;
0730 START
0740     JSR $FFFF    ; handle DOS
0750     CLD          ; for safety
0760     LDA # <NEWMEMLO
0770     STA MEMLO
0780     LDA # >NEWMEMLO
0790     STA MEMLO+1
0800 ;
0810     LDA #11
0820     STA LMARGN  ; change margin
0830 ;
0840 ; Position/print title
0850 ;
0860     LDA #15
0870     STA COLCRS
0880     LDA #0
0890     STA COLCRS+1 ; X=15
0900     LDA #3
0910     STA ROWCRS   ; Y=3
0920     LDA # <TITLE
0930     LDY # >TITLE
0940     JSR PRINT    ; "mUse V1.0"
0950 ;
0960     JSR VCLEAR  ; init vars
0970 ;
0980 ; Init BASIC line pointer
0990 ;
1000     LDA STMTAB
1010     STA PNTR
1020     LDA STMTAB+1
1030     STA PNTR+1
1040 ;
1050 ; Check for last lineno ($8000)
1060 ;
1070 NEWLINE
1080     LDY #1
1090     LDA (PNTR),Y
1100     BMI LNPRINT ; exit if done
1110 ;
1120 ; Increment line count
1130 ;
1140 LINEINC
1150     INC FR0
1160     BNE GETOFFS
1170     INC FR0+1
1180 ;
1190 ; Fetch line & stmnt offsets
1200 ;
1210 GETOFFS
1220     INY
1230     LDA (PNTR),Y
1240     STA LNOFF
1250     INY
1260 NEWSTOFF
1270     LDA (PNTR),Y
1280     STA STOFF
1290 ;
1300 ; Increment statement count
1310 ;
1320     INC STATS
1330     BNE FINDREMS
1340     INC STATS+1
1350 ;
1360 ; Is this a REM statement?
1370 ;
1380 FINDREMS
1390     INY
1400     LDA (PNTR),Y ; get token
1410     BNE ENDPNTR? ; 0=REM
1420 ;
1430 ; Count # of chars in REM stmnt
1440 ;
1450 RCOUNT
1460     INC REMS
1470     BNE SKIP
1480     INC REMS+1
1490 SKIP
1500     INY
1510     LDA (PNTR),Y
1520     CMP #$9B    ; EOL?
1530     BNE RCOUNT
1540 ;
1550 ; Any more stmts on this line?
1560 ;
1570 ENDPNTR?
1580     LDY STOFF
1590     CPY LNOFF
1600     BNE NEWSTOFF
1610 ;
1620 ; Add LNOFF to PNTR
1630 ;
1640     CLC
1650     LDA LNOFF
1660     ADC PNTR
1670     STA PNTR
1680     BCC NEWLINE
1690     INC PNTR+1
1700     BNE NEWLINE
1710 ;
1720 ; Print # lines & statements
1730 ;
1740 LNPRINT
1750     LDA # <L1
1760     LDY # >L1
1770     JSR PRINT   ; "# Lines"
```

```
1780      JSR NUMPRINT ; in FR0
1790 ;
1800      LDA # <L2
1810      LDY # >L2
1820      JSR PRINT    ; "# Stmnts"
1830      LDA STATS
1840      STA FR0
1850      LDA STATS+1
1860      STA FR0+1
1870      JSR NUMPRINT
1880 ;
1890 ; Print # REM characters
1900 ;
1910      LDA # <L3
1920      LDY # >L3
1930      JSR PRINT    ; "# REM chars"
1940      LDA REMS
1950      STA FR0
1960      LDA REMS+1
1970      STA FR0+1
1980      JSR NUMPRINT
1990 ;
2000 ; Calculate & print size of
2010 ; token table
2020 ;
2030      LDA # <L4
2040      LDY # >L4
2050      JSR PRINT    ; "Program RAM"
2060      CLD
2070      SEC
2080      LDA STARP
2090      SBC STMTAB
2100      SBC #3       ; line $8000
2110      STA FR0
2120      LDA STARP+1
2130      SBC STMTAB+1
2140      STA FR0+1
2150      JSR NUMPRINT
2160 ;
2170 ; Count # of variables
2180 ;
2190      JSR VCLEAR
2200      LDA VVTP
2210      STA PNTR
2220      LDA VVTP+1
2230      STA PNTR+1
2240 ;
2250 ; End of value table?
2260 ;
2270 TABEND?
2280      LDA PNTR
2290      CMP STMTAB
2300      BNE VARCNT
2310      LDA PNTR+1
2320      CMP STMTAB+1
2330      BNE VARCNT
2340      JMP VPRINT   ; none left
2350 ;
2360 VARCNT
2370      LDY #0
2380      LDA (PNTR),Y ; get v-type
2390      BEQ SVPLUS   ; 0 = simple var
2400      BMI SPLUS    ; $8n = string
2410 ;
2420 ; Handle a numeric array
2430 ;
2440      INC ARRYS
2450      AND #$01     ; is it DIMed?
2460      BEQ ADD8     ; not yet
2470      JSR ZFR0
2480 ;
2490 ; Fetch DIM1, subtract 1,
2500 ; store in FR0
2510 ;
2520      CLD
2530      SEC
2540      LDY #4
2550      LDA (PNTR),Y ; lsb
2560      SBC #1
2570      STA FR0
2580      INY
2590      LDA (PNTR),Y ; msb
2600      SBC #0
2610      STA FR0+1
2620 ;
2630 ; Fetch DIM2, subtract 1,
2640 ; store in TEMP
2650 ;
2660      SEC
2670      INY
2680      LDA (PNTR),Y ; lsb
2690      SBC #1
2700      STA TEMP
2710      INY
2720      LDA (PNTR),Y ; msb
2730      SBC #0
2740      STA TEMP+1
2750 ;
2760 ; If DIM2=0, this is a
2770 ; one-dimensional array
2780 ;
2790      LDA TEMP
2800      BNE MULTIPLY ; must be 2-D
2810      LDA TEMP+1
2820      BEQ ADDA     ; must be 1-D
2830 ;
2840 ; Multiply DIM1 by DIM2
2850 ; to determine the number of
2860 ; cells in this 2-D array
2870 ;
2880 MULTIPLY
2890      JSR IFP      ; DIM1 to FP
2900      JSR FMOVE    ; into FR1
2910      JSR ZFR0
2920      LDA TEMP     ; put DIM2
2930      STA FR0      ; into FR0
2940      LDA TEMP+1
2950      STA FR0+1
2960      JSR IFP      ; DIM2 to FP
2970      JSR FMUL     ; DIM2 * DIM1
2980      JSR FPI      ; into integer
2990 ;
3000 ; Add the result to the total
3010 ; number of array cells
3020 ;
3030 ADDA
3040      CLD
3050      CLC
3060      LDA FR0
3070      ADC ADIMS
3080      STA ADIMS
3090      LDA FR0+1
3100      ADC ADIMS+1
3110      STA ADIMS+1
3120      JMP ADD8     ; whew!
3130 ;
3140 ; Handle a simple variable
3150 ;
3160 SVPLUS
3170      INC SVARS    ; simple enough
3180      BNE ADD8
3190 ;
3200 ; Handle a string
3210 ;
3220 SPLUS
3230      INC STRGS
3240      AND #$01     ; DIMed yet?
3250      BEQ ADD8     ; guess not
3260      CLD
3270      CLC
3280      LDY #6
3290      LDA (PNTR),Y
3300      ADC SDIMS
3310      STA SDIMS
3320      INY
3330      LDA (PNTR),Y
3340      ADC SDIMS+1
3350      STA SDIMS+1
3360 ;
3370 ; Point to next variable
3380 ;
3390 ADD8
3400      CLD
3410      CLC
3420      LDA PNTR
3430      ADC #8
3440      STA PNTR
3450      LDA PNTR+1
3460      ADC #0
3470      STA PNTR+1
```

```
3480      JMP TABEND?
3490 ;
3500 ; Print number of variables
3510 ;
3520 VPRINT
3530      LDA # <L5
3540      LDY # >L5
3550      JSR PRINT    ; "# Vars"
3560      JSR ZFR0
3570      LDA SVARS
3580      STA FR0
3590      JSR NUMPRINT
3600 ;
3610 ; Print number of arrays
3620 ;
3630      LDA # <L6
3640      LDY # >L6
3650      JSR PRINT    ; "# Arrays"
3660      LDA ARRYS
3670      STA FR0
3680      JSR NUMPRINT
3690 ;
3700 ; Print amount of array RAM
3710 ;
3720      LDA # <L7
3730      LDY # >L7
3740      JSR PRINT    ; "Array RAM"
3750      LDA ADIMS
3760      STA FR0
3770      LDA ADIMS+1
3780      STA FR0+1
3790      JSR IFP
3800      JSR FMOVE
3810      JSR ZFR0
3820      LDA #6
3830      STA FR0
3840      JSR IFP
3850      JSR FMUL     ; ADIMS * 6
3860      JSR PRINTFP
3870 ;
3880 ; Print number of strings
3890 ;
3900      LDA # <L8
3910      LDY # >L8
3920      JSR PRINT    ; "# Strings"
3930      LDA STRGS
3940      STA FR0
3950      JSR NUMPRINT
3960 ;
3970 ; Print string RAM usage
3980 ;
3990      LDA # <L9
4000      LDY # >L9
4010      JSR PRINT    ; "String RAM"
4020      LDA SDIMS
4030      STA FR0
4040      LDA SDIMS+1
4050      STA FR0+1
4060      JSR NUMPRINT
4070 ;
4080 ; Calculate/print VNT RAM usage
4090 ;
4100      LDA # <L10
4110      LDY # >L10
4120      JSR PRINT    ; "Varname RAM"
4130      SEC
4140      LDA VNTD
4150      SBC VNTP
4160      STA FR0
4170      LDA VNTD+1
4180      SBC VNTP+1
4190      STA FR0+1
4200      JSR NUMPRINT
4210 ;
4220 ; Calculate/print free RAM
4230 ;
4240      LDA # <L11
4250      LDY # >L11
4260      JSR PRINT    ; "Free RAM"
4270      SEC
4280      LDA HIMEM
4290      SBC MEMTOP
4300      STA FR0
4310      LDA HIMEM+1
4320      SBC MEMTOP+1
4330      STA FR0+1
4340      JSR NUMPRINT
4350 ;
4360 ; Do a BASIC warmstart
4370 ;
4380      LDA #2
4390      STA LMARGN   ; Reset margin
4400      JMP WARM
4410 ;
4420 ; * Text Print Subroutine *
4430 ;
4440 ; ENTRY:
4450 ; Addr of text in A/Y (lsb/msb)
4460 ;
4470 PRINT
4480      LDX #0       ; IOCB #0 (E:)
4490      STA ICBADR,X ; lsb
4500      TYA
4510      STA ICBADR+1,X ; msb
4520      LDA #$0B     ; put char
4530      STA ICCOM,X
4540      LDA #14      ; fixed length
4550      STA ICBLEN,X
4560      TXA
4570      STA ICBLEN+1,X
4580      JMP CIOV     ; self-return
4590 ;
4600 ; * Number Print Subroutine *
4610 ;
4620 ; ENTRY:
4630 ; 16-bit integer in FR0
4640 ;
4650 NUMPRINT
4660      JSR IFP      ; to FP
4670 PRINTFP
4680      JSR FASC     ; to ATASCII
4690      LDY #$FF     ; find last char
4700 LOOP
4710      INY
4720      LDA (INBUFF),Y
4730      BPL LOOP
4740      AND #$7F     ; mask bit 7
4750      STA (INBUFF),Y
4760      INY
4770      LDA #$9B
4780      STA (INBUFF),Y ; install EOL
4790      TYA          ; get length
4800      LDX #0       ; IOCB #0 (E:)
4810      STA ICBLEN,X
4820      TXA
4830      STA ICBLEN+1,X
4840      LDA INBUFF   ; addr of num
4850      STA ICBADR,X
4860      LDA INBUFF+1
4870      STA ICBADR+1,X
4880      LDA #$09     ; put record
4890      STA ICCOM,X
4900      JSR CIOV     ; print number
4910      JSR ZFR0
4920      CLD
4930      RTS
4940 ;
4950 ; * Clear Variables ($C9-$E5) *
4960 ;
4970 VCLEAR
4980      LDX #$1C
4990      LDA #0
5000 EMPTY
5010      STA TEMP,X
5020      DEX
5030      BPL EMPTY
5040      RTS
5050 ;
5060 ; Message texts
5070 ;
5080 TITLE
5090      .BYTE "MUse V1.0   ↓",$9B
5100 L1
5110      .BYTE "    # Lines "
5120 L2
5130      .BYTE "   # Stmnts "
5140 L3
5150      .BYTE "# REM chars "
5160 L4
5170      .BYTE "Program RAM "
```

```
5180 L5
5190      .BYTE "      # Vars  "
5200 L6
5210      .BYTE "    # Arrays  "
5220 L7
5230      .BYTE "  Array RAM  "
5240 L8
5250      .BYTE "   # Strings  "
5260 L9
5270      .BYTE "  String RAM  "
5280 L10
5290      .BYTE " Varname RAM  "
5300 L11
5310      .BYTE "    Free RAM  "
5320 ;
5330 ; End of MUse routine
5340 ;
5350 NEWMEMLO
5360 ;
5370 ; The following code is used
5380 ; only during power-up --
5390 ; It will be erased when BASIC
5400 ; initializes itself
5410 ;
5420      LDA DOSINI
5430      STA START+1
5440      LDA DOSINI+1
5450      STA START+2
5460 ;
5470 ; Replace DOSINI with the
5480 ; addr of the MUse code
5490 ;
5500      LDA # <START
5510      STA DOSINI
5520      LDA # >START
5530      STA DOSINI+1
5540 ;
5550 ; Now set MEMLO above MUse
5560 ;
5570      LDA # <NEWMEMLO
5580      STA MEMLO
5590      LDA # >NEWMEMLO
5600      STA MEMLO+1
5610 ;
5620 ; Announce successful load
5630 ;
5640      LDA # <LOADED
5650      LDY # >LOADED
5660      JSR PRINT
5670 ;
5680 ; Set up for jump into BASIC
5690 ;
5700      LDX #0
5710      STX COLDST
5720      INX
5730      STX BOOT?
5740      JMP COLD
5750 ;
5760 ; Load OK message
5770 ;
5780 LOADED
5790      .BYTE "R↓MUse 1.0 OK",$9B
5800 ;
5810 ; Set autorun vector
5820 ;
5830      *=  RUNAD
5840      .BYTE  <NEWMEMLO
5850      .BYTE  >NEWMEMLO
5860      .END
```

In Every Sector, this is the Source for all your ATARI needs !

SOFTWARE

HARDWARE

EDUCATIONAL & BUSINESS CONSULTATION

RAM COMPUTER CENTER

AUTHORIZED SERVICE

INSTRUCTION

24 HOUR BULLETIN BOARD

Word Processing Package for your ATARI $649

Gemini 10X Printer
850 Interface
Printer Driver
Cable
ATARI WRITER

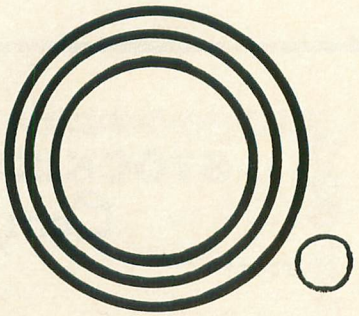
ATARI 1050 Disk Drives
Call for Information

ELECTRONIC ARTS 15% OFF
Hard Hat Mack LIST PRICE
Worms?
M.U.L.E.
Pinball Construction Set
Archon
Murder on the Zinderneuf
Word Flyer
D-Bug
Seven Cities of Gold
Axis Assassin

ATARI is TM of ATARI, Inc.

ATARI 1020 Color Printer Plotter $249

Austin Franklin 48K Board for
400/800
$89 ($15 Installation)



Pole Position : $39.95
QIX : $34.95
Galaxian : $34.95
Caverns of Mars : 34.95
Dig-Dug : $34.95
Paint : $31.95
ATARI WRITER : $79.95

Plus over 400 other titles at comparable savings !!!!!

Prices effective to December 31, 1983.

Call our 24 Hour **MODEM**
BULLETIN BOARD SERVICE
(617) 371-1855

We ship anywhere.

RAM Computer Center

427 Great Road, Acton, MA 01720 **(617) 263-0418**

Flight Simulator II

For Atari computers with 48K memory

Put yourself in the pilot's seat of a Piper 181 Cherokee Archer for an awe-inspiring flight over realistic scenery from New York to Los Angeles. High speed color-filled 3D graphics will give you a beautiful panoramic view as you practice takeoffs, landings, and aerobatics. Complete documentation will get you airborne quickly even if you've never flown before. When you think you're ready, you can play the World War I Ace aerial battle game. Flight Simulator II features include ■ animated color 3D graphics ■ day, dusk, and night flying modes ■ over 80 airports in four scenery areas: New York, Chicago, Los Angeles, Seattle, with additional scenery areas available ■ user-variable weather, from clear blue skies to grey cloudy conditions ■ complete flight instrumentation ■ VOR, ILS, ADF, and DME radio equipped ■ navigation facilities and course plotting ■ World War I Ace aerial battle game ■ complete information manual and flight handbook.

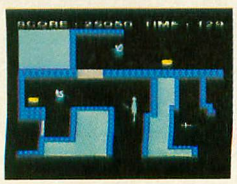
See your dealer . . .

or write or call for more information. For direct orders please add $1.50 for shipping and specify UPS or first class mail delivery. American Express, Diner's Club, MasterCard, and Visa accepted.

Order Line: 800/637-4983

subLOGIC Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

WELCOME TO APSHAI. YOU'RE JUST IN TIME FOR LUNCH.



Boy, have you taken a wrong turn. One moment you're gathering treasure and the next you're being eyed like a side of beef.

You're in the Gateway to Apshai.™ The new cartridge version of the Computer Game of the Year,* Temple of Apshai.™

Gateway has eight levels. And over 400 dark, nasty chambers to explore. And because it's joystick controlled, you'll have to move faster than ever.

But first you'll have to consider your strategy. Is it treasure you're after? Or glory? You'll live longer if you're greedy, but slaying monsters racks up a higher score.

The Apshai series is the standard by which all other adventure games are judged. And novices will not survive.

They'll be eaten.

One player; Temple of Apshai, disk/cassette; Gateway to Apshai, cartridge, joystick control.

EPYX COMPUTER SOFTWARE

STRATEGY GAMES FOR THE ACTION-GAME PLAYER.

*Game Manufacturers Association, 1981

This Christmas, Atari and Commodore* owners will go out of their minds.

It'll take about 30 seconds.

Because once you boot a Datamost game on your Atari home computer, or your Commodore 64, you'll come face to screen with the most mind-blasting games ever.

And what better time to go out of your mind than Christmas?

Our Music Will Have You Hearing Things.

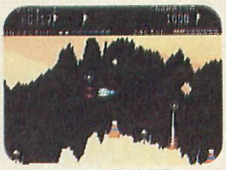
Going out of your mind never sounded so good.

Because now our games have music. You heard right. Music. Original Music. Throughout. And scored just for our newest releases. You'll go nuts over it.

Our Graphics Will Have You Seeing Things.

You and everybody else.

Because our games are so great-how great are they?-they're so great you'll want to play them again and again and again. And then your friends will go bonkers over them.



THE TAIL OF BETA LYRAE.™ Changes as you play. Unpredictable. Impossible to master. (No one has!)

And they'll want to play. And then your family will want to play. And then total strangers off the street will want to play and...

Mind-blowing arcade-quality action like this makes it one mad world, pal.

We'll Torment You Right From The Start.

No time to settle down and get comfy.

The tension begins building the moment you boot the game.

Terrific screen titles tease you with the game's objective, scenario, characters.

And while you wait, hand twitching over the Joystick, you'll find out what planet you're on.

What the fuss is all about.

Why you're involved. And perhaps of singular importance to you personally, how to keep from being obliterated. So get ready to get the most out of your mind.

By getting the most out of ours.

DATAMOST INC™

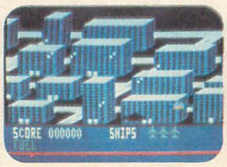
The most out of our minds.™

Datamost, Inc., 8943 Fullbright Ave., Chatsworth, CA 91311, (213) 709-1202

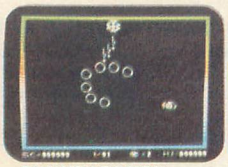
Atari is a trademark of Atari Computer. *Commodore 64 is a trademark of Commodore Business Machines, Inc. TM Registered Trademark of Datamost.

COSMIC TUNNELS.™ Four games in one. Four times the challenge! Incredible graphics.

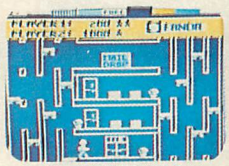
MONSTER SMASH.™ Deathly strategy. Mash the monsters! Let the visitors live.



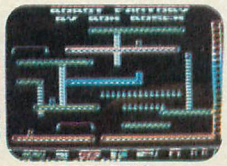
NIGHTRAIDERS.™ Strafe a city under siege with 3D angled selective firing!



ROUNDABOUT.™ Sharpens your shoot 'em up skills. 24 different game screens. Habit-forming!



COHEN'S TOWERS.™ You're the mailboy in a big city skyscraper. Work your way to the top.



MR. ROBOT.™ Screens scream with color, action and sound! Design your own screens, too.

THE HOTTEST NEW GAMES IN THE GALAXY!!!

SWORD POINT

NEW

SWORD POINT, by William A. Austin, is a very unique and exciting adventure. You will assume the role of a young man seeking to climb to the pinnacle of success in the mythical monarchy of Simafranconia, a nation not unlike feudal 17th Century Europe. It is a country of totally masculine values, where style supersedes substance, and social level is not the best thing, it is the only thing worth acquiring. Dueling, gambling, wenching and high living are among a few of the things that you will do to gain the King's attention. So tankards up . . . a toast to bravery . . . with a little luck and by dancing to the daredevil's tune on the Swords Point . . . you may just win! Requires 48K RAM and one disk drive.

$31.95

KARMIC CAVERNS

NEW

KARMIC CAVERNS, by Len Dorfman, is a 100% machine language arcade-style game that will entertain you hour after hour! You must find your way through the maze-like caverns avoiding the deadly plasmatic guards (not to mention the electrified walls) and acquire as many energy pods as you can. You must climb ever upwards striving to acquire enough energy points to reach the next level before time runs out for you. KARMIC CAVERNS utilizes the full sound and graphics capabilities of the Atari computer! Fun for all ages. Requires 48K RAM and one disk drive.

$34.95

PMI

BRINGING COMPUTERS DOWN TO EARTH.

WAMPYR'S TOMB

WARNING!! This adventure may be addictive. WAMPYR'S TOMB, by James Glines, will take you on a journey from the village of Amberpool through a 'mid-evil' countryside filled with magical spells, demons and dragons. Can you stay alive? To prevail in WAMPYR'S TOMB you must do more! You must equip yourself to do battle with the LICH and win. (Of course there is the option of dying!) We will supply you with a little money and the opportunity to go where you've never been; the courage and the desire to conquer the unknown must come from within. WAMPYR'S TOMB will not only challenge your bravery, your intelligence and your cunning, but will take you beyond your imagination into a realm of new and exciting forces where only legends survive. Requires 48K RAM and one disk drive.

$29.95

PROGRAMMERS!

If you have written a program that is technically sophisticated and also "user-friendly," PMI would be interested in publishing your work. For information on how to submit your material write to: PMI, P.O. Box 2895, Winter Park, FL 32790-2895. Or phone (305) 644-3822.

DEALER INQUIRIES INVITED

All ***PMI*** programs are available at the best software stores throughout the world, or order direct from ***PMI:***

PMI, P.O. Box 2895, 2500 Lee Road, Suite 210, Winter Park, FL 32790-2895

☐ ***SWORD POINT*** ☐ ***KARMIC CAVERNS*** ☐ ***WAMPYR'S TOMB***

Total $ _____ , plus $3.00 postage & handling (Florida residents add 5% sales tax)

☐ CHECK (allow 3 weeks for personal checks to clear) ☐ MONEY ORDER

☐ C.O.D. (add an additional $2.00 for C.O.D. orders)

NAME (print) _____

Address _____

City _____ State _____ Zip _____

by Tom Hudson

Last issue, you were introduced to the concept of various numbering systems, including base 2, base 10 and base 16. We also covered the basics of assembly language and the registers of the 6502 microprocessor.

In this issue, we'll talk about the ways the 6502 can address memory and begin looking at the 6502 instruction set.

Address unknown?

In order to perform useful work for us, the 6502 microprocessor chip must be able to get numbers from memory, manipulate them, and place the results back in memory. Each memory location has its own number, or ADDRESS. The 6502 can reference up to 65536 bytes of memory ($0000-$FFFF).

If you've used the BASIC PEEK and POKE functions, you've used the 6502's addressing ability already. Consider, for example, the BASIC command:

```
POKE 559,0
```

This command places a zero in address 559 ($22F), which turns off the computer's screen display.

Luckily for us programmers, the designers of the 6502 gave us quite a bit of flexibility in how we reference memory locations. These ways are listed below.

Immediate addressing allows us to place one number we are working with (or OPERAND) right after the operation code. The operand must be preceded with the "#" symbol. For example, the assembly instruction:

```
LDA #23
```

places the number 23 in the accumulator. In this example we specified the number in decimal. If we wanted, we could have given the number in hexadecimal (base 16):

```
LDA #$17
```

Note that decimal numbers require no special marking, but hex numbers are always preceded by a "$" symbol.

Absolute addressing tells the computer we want to get the operand from a certain address somewhere in memory. For example, let's say we want to turn off the screen as we did before in the above BASIC example. Instead of a POKE 559,0 command, we could use the following two assembly instructions:

```
LDA #0
STA 559
```

The first instruction, as we learned above, will load the accumulator with a zero. The second instruction uses the absolute addressing mode to store the contents of the accumulator into memory address 559. What could be easier?

Implied addressing means that no addresses are used in the instruction. The CLC (clear carry) and RTS (return from subroutine) instructions are good examples of implied addressing instructions.

Accumulator addressing is used for those instructions that use only the accumulator, such as ASLA (arithmetic shift left).

Indexed addressing is a useful type of addressing which makes table operations very simple. In this mode, the X or Y register is used as an index. For example, in the following instruction:

```
LDA TABLE, X
```

If the X register contains a 7, the accumulator will be loaded with whatever is in the seventh byte after TABLE. It's a very simple concept, and works the same with the Y register.

Indirect addressing is only used with the JMP (jump to location) instruction. In the following example:

JMP ($3000)

The JMP will NOT go to address $3000, but it will take the contents of $3000 and $3001 and jump to the address indicated by their contents. If, for example, $3000 contains $3F and $3001 contains $50, the program will jump to $503F. This instruction is rarely used, but it can be irreplacable under certain circumstances.

Pre-indexed indirect addressing uses the X register and an operand byte to address a byte in the first 256 bytes of memory. In the following example:

LDA ($AF,X)

If the X register contains $12, the computer adds $AF and $12, giving a result of $C1. The computer then takes the contents of $C1 and $C2 and loads the accumulator from the address contained in these bytes. For example, if location $C1 contains $50 and location $C2 contains $3F, the accumulator will be loaded from location $3F50.

Post-indexed indirect addressing uses the Y register and an address in the first 256 bytes of memory to point to another address. In the following example:

LDA ($AF),Y

The computer takes the contents of bytes $AF and $BO and adds the Y register to this address for a final address. If $AF contains $00 and $B0 contains $40 the computer first points to $4000, then uses the Y register as an offset. If the Y register contains $50, the accumulator would be loaded from $4050. This addressing mode is fairly often.

Relative addressing is used in all branch-on-condition instructions in the 6502. Usually after a comparison the programmer will branch on a condition. This is the same as an IF/THEN statement in BASIC. In the following example:

BNE START

The computer will calculate the number of bytes between the branch instruction and the location referenced by START at assembly time. During execution, the image in memory may look like:

BNE $30

This indicates that START was 48 bytes from the branch instruction. If the branch is executed, the computer will skip 48 bytes and continue executing at the part of the program labeled START. There is only one drawback to this addressing mode: The branch cannot be farther than -126 or +129 bytes. Longer branches require the use of the JMP instruction.

Assembler syntax.

Every computer language has a set of rules known as SYNTAX. These rules are established so that the programmer will enter program code in a way that the computer can understand. Assembly language has a very simple syntax, shown in **Figure 1**.

LABEL OP CODE OPERAND COMMENTS

Figure 1.

If you have ever looked at assembly language source listings in **A.N.A.L.O.G.**, you have probably noticed the neat columns of "gibberish." This is the way assembly language is structured.

Each column of information in the assembly source listing is known as a FIELD. Each field is separated by one or more spaces.

The first field, or LABEL field, is optional. If the code you are writing will be referenced elsewhere in the program, you should place an appropriate label in the label field.

A label should give some idea of what the section of code does. For example, L0001 tells nothing about the code, whereas VBLANK tells us that the code is part of the vertical blank cycle. Meaningful labels should be included whenever possible.

Lables should start with a letter, but can contain numbers within them.

Many assemblers use only the first 5 or 6 characters of a label, so the labels we use will be limited to 6 characters. This will enable the readers with assemblers other than the ATARI cartridge to use the program listings with as little modification as possible.

The second field in an assembler statement is the OPERATION CODE. This is usually a three-character standard 6502 instruction, such as LDA, STA, or JMP.

Each assembler also has a set of DIRECTIVES, or PSEUDO-OPERATIONS. These operations are not commands to the 6502, but are processed by the assembler program at assembly time. The most common directives are ".BYTE," ".WORD," "EQU" or "=" and "ORG" or "*=." These will be discussed in detail later.

The third field in an assembler statement is the OPERAND. This field contains data or addresses required by the operation code. Operands are not needed by all operation codes.

Operands are usually given in decimal or hexadecimal. Decimal numbers require no special prefix, but hex numbers must be preceded by the "$" character.

Operands can also be labels defined elsewhere in the program. For example, instead of:

JMP $4000

We could have used the EQUATE directive to define a label called START and set it to the value of $4000 as follows:

START = $4000
JMP START

By using labels in operands instead of absolute numbers, programs are easier to change if the need

arises. Imagine having to change 50 "JMP $4000" instructions to "JMP $5000." If we used "JMP START " instead, we'd only have to change the "START = $4000" to START = $5000." This would automatically change the 50 JMP instructions!

The last field in an assembler statement is the COMMENT. Comments are optional, but encouraged. Comments are like REMarks in BASIC — they help document what the programmer is doing. This is especially important in assembler programs, which are somewhat difficult to decipher.

Comments are preceded by a semicolon (;). Everything after the semicolon is ignored by the assembler. Comments should be used as often as possible, especially when a section of code is fairly complex. This will not only help others who use the program, but will help you if you need to make changes to the program at a later date.

Where to put the program?

In BASIC, the programmer doesn't really care where the program is placed in memory. BASIC handles all these messy details for the programmer, who simply writes program code. This is one of the benefits of a high-level language like BASIC.

As mentioned last issue, the assembly language programmer must know at all times what locations a program is using. Without total knowledge of a program's location, it is possible to overlap memory used by the system and cause an irrecoverable "lock-up."

Let's look at what memory locations are available to us in the ATARI computer system. This discussion will apply to users of the ATARI assembler-editor cartridge only.

Plug your cartridge into the computer and turn on the power. When the EDIT prompt appears. type SIZE and press RETURN. The cartridge will show three numbers, such as:

0700 0800 3C1F

The first number is the bottom of RAM, the second is the end of the text editor buffer, and the third is the highest available RAM address.

Since readers have different amounts of memory and since cassette and disk systems use different amounts of memory, each reader must decide where to place the object program in memory. To do this, subtract about $600 (1536 bytes) from the last number above. In this case, the number is $3C1F-$0600 = $361F. Round this down to the nearest 256 bytes and you have $3600. This will be the starting address of your object program. Use this address in the "*=" directive of the program in this column.

There are also 256 bytes available for use at $0600-$06FF, or PAGE 6. We will be using this area later for subroutines called by BASIC. The term "page" is used to refer to a 256-byte section of memory. The page number comes from the first two digits of the hex address. $0200-$02FF is page 2, $0800-08FF is page 8, etc.

The last memory available to us has special significance. This memory lies on page 0, $0000-$00FF. When the 6502 knows a byte is on page 0, it only needs the last two hex digits to address it. This allows the 6502 to access the information faster, with a smaller program, since only one byte is needed in the operand instead of the usual two needed for an address.

Since page 0 addresses can be accessed faster with less program memory, it is obviously good to use page 0 whenever possible. The problem is, the system uses some page 0 for its own needs. The entire first half page of 0 ($0000-$007F is always used by the system. The second half ($0080-$00FF) is available to assembly language programs if no cartridges are in use.

Unfortunately, the ATARI assembler editor cartridge only allows you to use locations $B0-$CF. These locations are probably sufficient for most testing purposes.

When writing assembly language programs to be called as subroutines by ATARI BASIC, only loca-

(continued on page 129)

ATARI® Printers/Etc.

600XL	CALL
1200XL	CALL *

*Reflects $100 Atari Rebate

1050 DRIVE	$335
1025 PRINTER	$399
1020 COLOR PTR	$219
1027 PRINTER	$279
1010 RECORDER	$72
850 INTERFACE	CALL
MICROBITS INFC	$78
80 COLUMN BD	$249
TECHNICAL NOTES	$25
REAL TIME CLK	$38
810 DRIVE	$419

DRIVES

RANA 1000	$319
PERCOM 88-S1	$358
PERCOM AT 88-S2	$539
PERCOM 40-S1	$510
PERCOM 40-S2	$799
PERCOM 44-S1	$639
PERCOM 44-S2	$929

MEMORIES

48K RAM (INTEC)	$95
64K RAM (INTEC)	$119
48K RAM (MOSAIC)	$109
64K RAM (MOSAIC)	$145
128K RAM DISK	$299
32K RAM (MOSAIC)	$68

ATARI SOFTWARE

ADVENTURE INT'L
Adv. 1-12 each (C) $18
Preppie (C/D) $20
Preppie II (C/D) $23
Diskey (D) $33
Sea Dragon (C/D) $23
Bug Off! (C/D) $20
Tutti-Frutti (C/D) $17
Saga 1-3 (ea.) (D) $27
Lunar Lndr (C) $11 (D) $15
Galactic Empire (C) $14
Galactic Trader (C) $14
Stratos (C/D) $23
Rear Guard (C) $14 (D) $17

APX
Eastern Front (C/D) $23
747 Land Sim. (C/D) $17
Fig-Forth (C) $30
Family Cash Flow (D) $17
Downhill (C/D) $17
Avalanche (C/D) $17
Outlaw/Howitzer (C/D) $17
Salmon Run (C/D) $17
Data Management (D) $17
Galahad/Holy Grail (D) $21
Adv. Music System (D) $21

ATARI INC.
Microsoft Basic II (R) $62
Mickey in Great Outdoors (C/D) $36
Paint (D) $30
Speed Reading (C) $54
Qix (R) $30
Dig Dug (R) $30
Atari Writer (R) $68
Time Wise (D) $23
Visicalc (D) $139
Juggles House (C/D) $22
Juggles Rnbw (C/D) $22
Pilot (Home) (R) $55
Galaxian (R) $30
Defender (R) $30
ET $34
Microsoft Basic (D) $62
Macro Ass. & Edit (D) $62
Assembler Editor (R) $42
Basic Cartridge (R) $45
Pac Man (R) $30
Centipede (R) $30
Caverns of Mars (D) $28
Star Raiders (R) $30
Conv. Lang. Ea. (C) $42
Music Composer (R) $31
Super Breakout (R) $26
My First Alphabet (D) $26
Prog.2 & E (ea.)(C) $21
Word Processor (D) $102
Pilot (Educ.) (R) $92
Touch Typing (C) $19
Home File Mngr (D) $36
Bookkeeper (D) $102
Family Finance (D) $35
Prog. 1 (C) $18
Scram (C) $18
Asteroids (R) $27
Space Invaders (R) $27
Missile Command (R) $27
Telelink (R) $21
Superman III (R) $34
Basketball (R) $24
Donkey Kong (R) $34

AVALON HILL
VC (D) $17
B-1 Nuc. Bomber (C) $12
Legionnaire (C) $23
Empire of Overmind (D) $23 (C) $20
Tanktics (D) $20 (C) $17
Comptr Stock & Bonds (D) $17 (C) $14
Dnieper River Line (D) $20
Voyager (D) $17
Controller (D) $20
GFS Sorceress (D) $23 (C) $20
Telegard (D) $19 (C) $16
Vorrak (D) $18
Lord of Karma (C) $14

AVALON HILL(Cont'd)
Moon Patrol (C) $17
Planet Miners (C) $12
Conflict 2500 (C) $12
Space Station Zulu (D) $17 (C) $14
Flying Ace (D) $21 (C) $18
Gypsy (D) $18 (C) $15
Galaxy (D) $17 (C) $14
Guns of Fort Def. (C) $14
Nuke War (C) $12
Andromeda Conquest (D) $16 (C) $13
Close Assit (D) $23 (C) $20

BRODERBUND
Sky Blazer (D) $22
Bank St. Writer (D) $46
A.E. (D) $23
Arcade Machine (D) $39
Choplifter (D) $23 (R) $29
Labyrinth (C/D) $20
Serpentine (D) $23 (R) $27
Steller Shuttle (C/D) $17
Apple Panic (C/D) $20
Genetic Drift (C/D) $20
David's Midnight (D) $23
Sea Fox (D) $20
Track Attack (D) $20
Operatn Whrlwnd (D) $23
Match Boxes (C/D) $20

CDY CONSULTING
Pogoman (C/D) $27
Mad-Netter (C/D) $23
It-Is-Balloon (C/D) $23

CBS
Mountain King (R) $27
Boulders & Bombs (R) $27
Krazy (each) (R) $27

CONTINENTAL SOFT.
Home Accountant (D) $48
Tax Advantage (D) $39

DATA MOST
Pig Pen (D) $20
Mating Zone (D) $20
Roundabout (D) $20
Bilestoad (D) $20
Night Raiders (D) $20
Monster Smash (D) $20

DATASOFT
Text Wizard (D) $65
Graphic Master (D) $27
Micro Painter (D) $23
Lisp Interpreter (D) $79
Graphics Gen.(D) $17
Basic Compiler (D) $65
Zaxxon (C/D) $27
Teletalk (D) $34
Pac. Coast Hwy (C/D) $20
Clowns/Balloons (C/D) $20
Spell Wizard (D) $53
Canyon Climber (C/D) $20
Sands of Egypt (D) $27
O'Riley's Mine (C/D) $23
Rosen's Brigade (C/D) $23
Moon Shuttle (D) $27

DON'T ASK
Sam (D) $39
P.M. Animator (D) $23
Teletari (D) $27
Poker Sam (D) $17
Word Race (D) $17 (C) $14

EDU-WARE
Prisoner II (D) $27
Spelling Bee (D) $27
Rendevous (D) $27
Compu/Read (D) $21 (C) $15
Compu/Math-FR (D) $27 (C) $21
Compu/Math-Dec (D) $27 (C) $21

EDUCATIONAL SOFT.
Tricky Tutorial 1,2,3 or 4 (C/D) $15
Tricky Tutorial 5,6 or 7 (C/D) $22

GEMINI 10X	$289
GORILLA	$199

CITOH

Prowriter	$345
Prowriter II	$629
Starwriter	$1149
Printmaster	$1448

NEC

8023 A-C	$409
3510	$1375
3530	$1579
3550	$1779
7710/7730	$1998

PROWRITER	$345
SMITH TPI	$488

SILVER REED P $669
QUME 11/40 + $1299

OKI-DATA

Microline 82A	$398
Microline 83A	$638
Microline 84P	$958
Microline 92	$488
Microline 93	$858

DIABLO

620R	$939
630R	$1719

MONITORS

AMDEK

Color I	$289
V300	$139
V300A	$149
Color II	$449

NEC

GRN (JB1260)	$115
GRN (JB1201)	$155
Color Composite	$298
RGB Color	$598

MODEMS

HAYES

Smartmodem	$209
Smartmodem 1200	$498
Micromodem II	$259

NOVATION

J-Cat	$99
Apple Cat II	$259
D-Cat	$149

ANCHOR AUTOMATION
Mark I or II Modem $78

D = DISK
R = ROM (Cartridge)
C = CASSETTE

SPECIALS

Gemini 15X Printer	$379
Axiom AT-100 Printer (with interface	$229
Astra Double Density Dual Drive	$469
Rana 1000 Drive	$319
Bit-3 80 Column Board	$245
Mannesmann Talley 160L Printer	$589
Atari 400 Keyboard (In Home)	$35
Programmer Kit $48	Entertainer Kit $64
Wico Joystick $23	Wico Trackball $49

COSMIC COMPUTERS
UNLIMITED

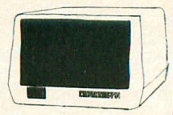
727 BREA CANYON RD., SUITE 16
WALNUT, CA 91789
ORDER LINES OPEN MON-SAT 8 am - 6 pm

(800) 626-7642
PLEASE FOR ORDERS ONLY
SORRY, NO COD'S

(714) 594-5204
TECHNICAL INFO OR IN CALIFORNIA

Add $2.00 shipping per software order in continental U.S. Add $5.00 shipping per software order for AK, HI, FPO-APO. Add $10.00 or 15% (whichever is greater) per software order for non-U.S. Call for cost of hardware shipping. Calif. residents add 6½% sales tax. Cashiers checks or money orders filled within 24 hours for items in stock. Personal checks require 4 weeks to clear. MasterCard and Visa OK for software only within continental U.S., add 3% surcharge. Include card no., expiration date and signature. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call to obtain one before returning goods for replacement or repair. Prices & availability subject to change.

*ASTRA 1620
DISK DRIVE SYSTEM



MORE FOR YOUR MONEY
DOUBLE OR SINGLE DENSITY
TWO DRIVES

SPECIAL $469

EPYX-AUTO. SIMULATION
Rescue at Rigel (C/D) $20
Datestones/Ryn (C/D) $14
Upr Rchs Apshai (C/D) $14
Ricochet (C/D) $14
Keys/Acheron (C/D) $14
Curse of Ra (C/D) $14
Danger/Drindisti (C/D) $14
Jumpman (C/D) $27
Escp Vulcans Isle (D) $20
Crypt of Undead (D) $20
Nightmare (D) $20
Armor Assault (D) $27
Monster Maze (R) $27
Alien Garden (R) $27
Plattermarna (R) $27
Morloc's Tower (C) $14
King Arthur's Heir (D) $20 (C) $14
Hellfire Warrior (C/D) $27
Invasion Orion (C/D) $17
Temple of Aps. (C/D) $27
Star Warrior (C/D) $27
Dragon's Eye (D) $20
Crush Crumble (C/D) $20

GEBELLI
Firebird (R) $27
Embargo (R) $30
Candy Factory (D) $20
Match Racers (C/D) $20
Dr. Goodcode (D) $20

HAYDEN
Go (D) $23 (C) $20
Sargon II (D) $23 (C) $20
Bulldog Pinball (C) $20

HES
Coco (C/D) $34
Coco II (C/D) $27

INFOCOM
Suspended (D) $34
Zork I, II or III (D) $27
Starcross (D) $27
Deadline (D) $34
Witness (D) $34

INHOME
Baseball (R) $23 (D) $23
Crypts of Terror (D) $23 (C) $20

INNOVATIVE DESIGN
Pool 1.5 (D) $23
Juggler (D) $20
Speedway Blast (D) $20 (R) $27
Pool 400 (R) $27

JV SOFTWARE
Jrny to Plnts (C/D) $20
Action Quest (C/D) $20
Ghost Encount. (C/D) $20

LJK
Letter Perfect (D) $104
Data Perfect (D) $74
Letter Perfect (R) $137
Edit 6502 (R) $137

MONARCH DATA SYS.
ABC Compiler (D) $48

ODESTA
Chess (D) $45
Checkers (D) $34
Odin (D) $34

OPTIMIZED SYSTEMS
C-65 (D) $58
Bug-65 (D) $23
Max-65 (D) $58
Basic A + (D) $58
Action (D) $65
Speed Read Plus (D) $40

PHOENIX
Birth of Phoenix (D) $16
Adv. In Time (D) $20

QUALITY
Fastgammon (C) $14
Name That Song (D) $13 (C) $11
Starbase Hyperion (D) $17 (C) $14
Ali Baba/40 Thvs (D) $20
Jeeper Creepers (D) $20

ROKLAN
Gorf (D) $27 (R) $30
Wizard/Wor (D) $27 (R) $30
Dlx Invaders (D) $23 (R) $27
Anti Sub Ptrl (D) $20 (C) $14
Telecom (D) $49
Space Journey (R) $30

SENTIENT
Gold Rush (D) $23
Cyborg (D) $23

SIERRA ON-LINE
Mission Asteroids (D) $17
Ulys.& Gldn Fleece (D) $27
Crossfire (D) $20 (R) $23
Mouseattack (D) $23
Jawbreaker (D) $20 (R) $23
Threshold (D) $27
Softporn (D) $20
Ultima II (D) $39

SIERRA ON-LINE
Marauder (D) $23
Lunar Leeper (D) $20
Wiz & Princess (D) $22
Frogger (C/D) $23

SIRIUS
Alpha Shield (R) $27
Wavy Navy (D) $23
Bandits (D) $23
Space Eggs (D) $20
Sneakers (D) $20
Way Out (D) $27
Type Attack (D) $27
Repton (D) $27
Critical Mass (D) $27
Fast Eddy (R) $23
Worm War (R) $23

SPINNAKER
Snooper Troop 1,2 (D) $30
Kindercomp (D) $20
Rhymes & Riddles (D) $20
Hey Diddle Diddle (D) $20
Srch Amzng Thngs (D) $27
Story Machine (D) $23
Face Maker (D) $23

STRATEGIC SIM.
Cosmic Balance (D) $27
Cosmic Balance II (D) $27
Tigers In Snow (C/D) $27
Battle of Shiloh (C/D) $27
Battle of Norm. (C/D) $27
Galactic Gladiator (D) $27
Cytron Masters (D) $27
Knights/Desert (C/D) $27
Combat Leader (C/D) $27

SUBLOGIC
Pinball (C/D) $20

SWIFTY
J. White Music Lessons (C/D) $20
Space Shuttle (D) $20

SYNAPSE SOFTWARE
File Mngr 800 + $65
Protector II (D) $23 (R) $29
Shamus (D) $23 (R) $29
Fort Apocalypse (C/D) $23
Shamus II (C/D) $23
Necromancer (C/D) $23
Pharoh's Curse (C/D) $23
Slime (C/D) $23
Page 6 (D) $23
Reptilian (C/D) $23
Picnic Paranoia (C/D) $23
Claim Jumper (C/D) $23
Drelbs (C/D) $23
Shadow World (C/D) $23
Survivor (C/D) $23
T.N.T. (C/D) $23
N.Y.C. (C/D) $23
Slamball (C/D) $23
Blue Max (C/D) $23
Nautilus (C/D) $23
Chicken (R/D) $23
Quasimodo (C/D) $23
River Quest (C/D) $23

SYNERGISTIC
Programr Wkshp (D) $23
Graph Workshop (D) $27
Disk Workshop (D) $23
Worlock's Revenge (D) $23

THORN EMI
Soccer (R) $34
Jumbo Jet (R) $34
Submarine Comm. (R) $34
Kick Back (R) $34
Pool (C) $22
Humpty Dumpty (C) $22
Hckry Dckry Dck (C) $22
Darts (C) $22

USA
Atari World (D) $39
3-D Sprgrphcs (C/D) $27
Survival Adv. (C/D) $17

VERSA
Mind Bgglrs I (D) $14 (C) $12
Globe Master (D) $20

MISCELLANEOUS
Financial Wizard (D) $41
Castle Wolfenstein (D) $20
Master Type (D) $27
Astro Chase (D) $22
Miner 2049er (R) $34
Cypher Bowel (C) $33
Bug Attack (C/D) $20
Raster Blaster (D) $20
Air Strike (C/D) $27
Prism (D) $18
Megalegs (C) $23
Cap'n Cosmo (D) $19
Spy's Demise (D) $14
Galac. Chase (D) $20 (C) $18
Kid Grid (C/D) $20
Battle Trek (D) $20
Snapper (D) $19
T&F-PMP Property Management (D) $149
Millionaire (D) $39

tions $CB-$D1 and $D4-$D5 can be used without conflict with BASIC's work areas. If an assembly subroutine needs temporary work areas, locations $D6-$F1 can be used. These areas will probably be changed by BASIC after the assembly subroutine ends, but they will work fine as temporary storage locations.

A few instructions.

Now we're ready to look at a few 6502 operation codes and see how they work. We'll start with the most frequently used instructions and work our way up to the rarely used instructions.

Without a doubt, the most frequently used 6502 operation code is LDA, or LOAD ACCUMULATOR. This instruction places a desired number in the A register, or accumulator.

The accumulator is used in all addition and subtraction operations, as well as most other arithmetic that can be performed on the 6502. You must move numbers in and out of the accumulator constantly, keeping track of the results. At times, you'll feel like a traffic cop trying to direct hundreds of cars through an ordinary doorway. After just a few hours of assembly programming, you'll see how important the accumulator is.

The LDA instruction has eight different formats, each with its own addressing method:

```
LDA #n     (IMMEDIATE)
LDA nn     (ABSOLUTE)
LDA n      (PAGE ZERO)
LDA (n,X)  (PRE-INDEXED INDIRECT)
LDA (n), Y(POST-INDEXED INDIRECT)
LDA n,X    (ZERO PAGE INDEXED X)
LDA nn,X   (INDEXED X)
LDA nn, Y  (INDEXED Y)
```

Each of these instructions work differently in order to load the accumulator. They find the address from which they are to get the number and place it in the accumulator, destroying whatever was there before. Once the number is placed in the accumulator, however, the instructions act alike.

Let's assume the number loaded into the accumulator was $94, shown below in its binary form (note the "%" sign preceding the binary number).

$94 = %10010100

All LDA instructions take special information from the number loaded and set microprocessor status flags accordingly. The two flags changed are the SIGN flag and the ZERO flag.

The zero flag is set to 1 if the number loaded was zero, and is set to 0 if the number was not zero. This flag is mainly used for branching, which we will cover later.

The sign flag is set to the value of the high-order (or leftmost) bit of the number loaded. You should remember that an 8-bit byte can contain numbers from 0-255. This is true when we are considering the numbers to be UNSIGNED. The 6502 uses a signed numbering system that can be somewhat confusing.

Whenever a number's high-order bit is a 1, the number is considered to be negative. Using this method, a byte can contain numbers from -128 to 127. How does this work? Let's start with the positive numbers.

Positive numbers in the 6502 signed number scheme range from 0 (which is always considered positive) to 127. The upper limit of 127 is set because if the number goes to 128, the high-order bit will be set to 1 and the number is negative.

Negative numbers range from -1 to -128 in the 6502 system. If we subtract 1 from zero in the 8-bit btye format, the byte's contents will "wrap around" to the bit pattern 11111111, which is 255. 255 corresponds to -1 in this scheme. An easy way to remember the relationship here is the following calculation:

UNSIGNED NUMBER — 256 =
SIGNED NUMBER

Using this formula with the unsigned number 255, we can see that 255 - 256 = -1, which is correct. We can easily find the signed counterpart to 128, or 128 - 256 = -128.

Now you can see exactly how the sign flag works. This flag will be very important later when we perform comparisons.

The next instruction, which is used almost as much as the LDA instruction is STA, or STORE ACCUMULATOR. This instruction does almost the same thing as LDA, but in reverse.

The STA instruction has the following formats:

```
STA nn     (ABSOLUTE)
STA n      (PAGE ZERO)
STA (n,X)  (PRE-INDEXED X)
STA (n), Y (POST-INDEXED Y)
STA n,X    (PAGE ZERO INDEXED X)
STA nn,X   (ABSOLUTE X)
STA nn,Y   (ABSOLUTE Y)
```

You will notice that the STA instruction has the same formats as the LDA instructor except for the IMMEDIATE format. Think about it for a minute and the reason should be obvious.

The STA instruction simply places whatever number is in the accumulator into the address specified in the operand. The number in the accumulator will be unaffected, and will still be available for your use.

The STA instruction does not affect any status flags.

A third instruction that is widely used is the JMP instruction. This instruction is just like BASIC's GOTO statement. Whenever this instruction is executed, the program will JUMP to the address

specified and continue processing. The address jumped to MUST contain executable program statements, so take care.

The JMP instruction has two formats:

JMP nn (ABSOLUTE)
JMP (nn) (INDIRECT)

As noted above in the discussion of the indirect format, the indirect jump is rarely used, but can be very helpful in special situations.

The absolute jump instruction is the most-used form of the JMP operation code. The address specified can either be a hex or decimal number or a label that is defined elsewhere in the program.

The JMP instruction does not affect any status flags.

Applying the instructions.

Now that we've described the LDA, STA and JMP instructions, let's apply them in a short program.

The program in **Figure 2** is essentially a "do-nothing." It will simply move numbers around in memory until we stop it. Type the program into your computer, remembering to set your origin value (*= in line 140) as described above.

```
0100 ;** DO-NOTHING DEMO PROGRAM **
0110 ;
0120 ;BY TOM HUDSON
0130 ;
0140         *=  $????    ;YOUR ORIGIN!
0150 ;
0160 START   LDA BYTE1    ;COPY BYTE1..
0170         STA BYTE2    ;TO BYTE2
0180         LDA #7       ;PUT A 7...
0190         STA BYTE3    ;IN BYTE3
0200         JMP PART2    ;JUMP!
0210 ;
0220 PART1   LDA BYTE2    ;MOVE BYTE2...
0230         STA BYTE4    ;TO BYTE4
0240         JMP PART3    ;AND JUMP
0250 ;
0260 PART2   LDA RANDOM   ;MOVE RANDOM..
0270         STA BYTE1    ;TO BYTE1
0280         JMP PART1    ;AND JUMP!
0290 ;
0300 PART3   LDA BYTE4    ;MOVE BYTE4...
0310         STA BYTE5    ;TO BYTE5
0320         JMP START    ;AND JUMP!
0330 ;
0340 ;DATA BYTE5 START HERE!
0350 ;
0360 BYTE1   .BYTE 1      ;NUMBER 1
0370 BYTE2   .BYTE 2      ;NUMBER 2
0380 BYTE3   .BYTE 3      ;NUMBER 3
0390 BYTE4   .BYTE 4      ;NUMBER 4
0400 BYTE5   .BYTE 5      ;NUMBER 5
0410 RANDOM =    $D20A    ;RANDOM NUMBER
0420 ;
0430         .END
```

Figure 2.

When you have entered the program and set the origin at Line 140, type ASM and press RETURN. The program will be assembled into memory and is ready to execute.

Before executing the program, let's look at **Figure 2.** The first thing you'll notice in the listing is the presence of COMMENTS. I can't overemphasize the importance of comments in an assembly language program. They're simply a MUST whenever you're writing programs, even for yourself. You'll notice that some comment lines are simply semicolons with no comment. These are used as separators to break up sections of code. For example, each label group (i.e. START, PART1, PART2, etc.) is a distinct group in the listing.

Remember, comments don't take up any program space in assembly language, so use them as often as possible!

Line 160 — loads the accumulator with the number 7, wiping out whatever was previously in the accumulator. Remember that whenever the accumulator is loaded, the contents of the accumulator before the load will be lost.

Line 190 — stores the 7 just loaded into the accumulator at the location labeled BYTE3. This is also a very common operation.

Line 200 — jumps to PART2, and execution continues there.

Line 220 — labeled PART1, loads the accumulator from the location marked BYTE2.

Line 230 — stores the value just loaded from BYTE2 into the location labeled BYTE4.

Line 240 — jumpst to PART3.

Line 260 — labeled PART2, loads a byte from the computer's random number generator at $D20A. This location gives a random number from 0-255.

Line 270 — stores the random number at the location labeled BYTE1.

Line 280 — jumps to PART1.

Line 300 — labeled PART3, loads the accumulator from the location labeled BYTE4.

Line 310 — stores the number just loaded at location BYTE5.

Line 320 — jumps to START. This causes the program to loop forever until you press the BREAK key.

Lines 360-400 — define the bytes labeled BYTE1-BYTE5. The .BYTE directive is used to assign initial values to the locations. BYTE 1 will contain 1, BYTE2 will contain 2, etc.

Line 410 — uses the EQUATE directive to define the address of the label RANDOM. This location is $D20A (53770 decimal). Whenever the label RANDOM is referenced, the computer will use the value $D20A.

Line 430 — uses the .END directive to tell the assembler the end of the source code has been reached. This directive is optional, but recommended.

Tracing the action.

Now you can execute the above program and see what it does. Note the address you used in Line 140. With the EDIT prompt on the screen, type BUG and press RETURN. The DEBUG prompt will appear.

Type L followed by the address you used in Line 140 and press RETURN. For example, if your Line 140 reads:

```
* = $4300
```

You should type L4300 and press RETURN. The computer will show how your program appears in memory, and should look something like **Figure 3**.

```
6000  AD 29 60  LDA $6029
6003  8D 2A 60  STA $602A
6006  A9 07     LDA #$07
6008  8D 2B 60  STA $602B
600B  4C 17 60  JMP $6017
600E  AD 2A 60  LDA $602A
6011  8D 2C 60  STA $602C
6014  4C 20 60  JMP $6020
6017  AD 0A D2  LDA $D20A
601A  8D 29 60  STA $6029
601D  4C 0E 60  JMP $600E
6020  AD 2C 60  LDA $602C
6023  8D 2D 60  STA $602D
6026  4C 00 60  JMP $6000
6029  01 02     ORA ($02,X)
602B  03        ???
602C  04        ???
602D  05 00     ORA #$00
602F  00        BRK
6030  00        BRK
```

Figure 3.

Your listing will probably vary from this illustration, which was assembled to location $6000. Note that the BYTE1-BYTE5 values appear in memory from $6029-$602D, and the computer tries to show the bytes as instructions (like DRA #$00). Simply ignore such instructions whenever you know they are misinterpreted data.

If your program is at the proper location, you are ready to watch its execution. Type T followed by the address in Line 140 and press RETURN.

The computer will begin tracing the execution of your program one line at a time. Each instruction will be shown along with its address and the contents of the 6502 registers after the instruction executes. Page 40 of the ATARI assembler editor manual describes the trace operation in detail.

At any time in the execution you may stop the program with the BREAK key and examine the BYTE1-BYTE5 locations (note their addresses at assembly time) by using the Dnnnn command, described on page 36 of the Assembler Editor manual.

We are interested in seeing how the instructions we used are executed and how they affect memory.

(continued on next page)

Figure 4 shows the lines of the program as they are executed and the status of the variables BYTE1-BYTE5 after each statement is executed. Note that the value present in RANDOM cannot be predicted and is indicated by "R#."

			STATEMENT	A RG.	BT. 1	BT. 2	BT. 3	BT. 4	BT. 5
160	START	LDA	BYTE 1	01	01	02	03	04	05
170		STA	BYTE 2	01	01	01	03	04	05
180		LDA	#7	07	01	01	07	04	05
190		STA	BYTE3	07	01	01	07	04	05
200		JMP	PART2	07	01	01	07	04	05
260	PART2	LDA	RANDOM	R#	01	01	07	04	05
270		STA	BYTE1	R#	R#	01	07	04	05
280		JMP	PART1	R#	R#	01	07	04	05
220	PART1	LDA	BYTE2	01	R#	01	07	04	05
230		STA	BYTE4	01	R#	01	07	01	05
240		JMP	PART3	01	R#	01	07	01	05
300	PART3	LDA	BYTE4	01	R#	01	07	01	05
310		STA	BYTE5	01	R#	01	07	01	01
320		JMP	START	01	R#	01	07	01	01
160	START	LDA	BYTE1	R#	R#	01	07	01	01
170		STA	BYTE2	R#	R#	R#	07	01	01

Figure 4.

(TRY THE NEXT 10 STEPS YOURSELF!)

As stated earlier, this is a "do-nothing" program, and will continue to execute forever unless it is stopped by the user. If you'd like a demonstration of this infinite execution, type G followed by the address in Line 140 and press RETURN. The computer will begin executing the do-nothing at unbelievable speed, and won't stop until you press BREAK. You won't see anything happen during the program's execution, but you can rest assured that the computer is following your instructions to the letter.

Stay tuned.

Next issue, we'll start digging into more 6502 operation codes, learn to add and subtract, and work with the index registers. Until then, make your own short programs using the instructions we've covered. I realize these three aren't enough to create complex programs, but knowledge of their use is essential to future lessons. □

A new column for BASIC programmers, also by Tom Hudson, will appear next issue.

OUR GAME

by Joel Gluck

Welcome to **Our Game**, the column in which you, the reader, and I, the writer, collaborate in creating and programming a BASIC computer game. You send in ideas, criticisms, and encouragement (flowers will do), and I develop all this into a *fabulous* game.

It's about time.

This is a very exciting moment, because it's the first **Our Game** to have the benefit of reader input. Translation: I got letters!

Although the response was not overwhelming, almost everyone who wrote thought that **Our Game** was a good idea (whew!). I'd like to thank everyone who sent me mail. I would love to respond personally to every one of you — unfortunately, I can't. I'd also like to apologize to anyone who writes but isn't mentioned in the colomn. It's impossible to include everyone's responses.

Donald Prakap of Alexandria, VA, writes:

I've have a hard time working with my Atari. I try to do too much without understanding the basics. I feel your column will be a good starting point. So many things come into play with the Atari P/M, vertical blank, display lists, etc., that starting a project seems like a monster."

Donald is absolutely right. One of the purposes of **Our Game** is to show Atari owners that, when programming, it isn't always necessary to wade through all of the strange features of the Atari computer. For example, I've written and sold several games for the 400/800, and I didn't use Player/Missile graphics in any of them. Of course, if the game we decide to write necessitates the use of Player/Missile or other advanced techniques, **Our Game** will try to make them digestible.

Charles G. Miller, resident of Severna Park, Md., writes:

"I, for one, would like to see a clear and comprehensive, but not overwhelming, flow chart presented during the development of our program. More important, I hope we can make the game appealing to GIRLS (for a change of pace) as well as boys. I have two daughters, and I think they get slighted — don't you?"

You should be pleased, Mr. Miller, with the organized approach **Our Game** will be using, although I do not plan to present formal flow charts. As for the game being equally appealing to both sexes, I agree with you; so here's a gentle proposal to the reader: before you send **Our Game** a game idea,

This message arrived from Karen Kujala of Taylor,

We've received several game ideas, but almost every one of them involved fighting, killing, etc. Please try to be more original — there have already been hundreds of violent, fast-action computer and arcade games. Let's try to be different!

This message arrived from Karen Kujala of Taylor, MI:

"I would like to see a game that can be played by one person, or simultaneously by two or more people. To me, it seems boring waiting for the other player to get bumped off so you can have your turn."

Nice suggestion, Karen. I, myself, lean toward games in which two or more people can play at the same time. There's nothing as much fun as human interaction.

Paul T. Sprague, of Orono, ME, wrote to say that the game we develop ought to be written in machine code. I'm sorry I can't oblige, Paul, because this column is devoted to beginning and intermediate BASIC programmers. However, don't lose heart. There's a lot one can learn about writing games — in any language, or any computer. **Our Game**, with the help of its readers, might still be able to teach you something.

Probably the most surprising mail was from Steven M. Owens, of Park Ridge, Il. Not only did Steven send me some pretty interesting (although violent) game ideas, he also sent me a really good twenty-eight page science fiction story he wrote that had inspired the ideas. Some of Steven's thoughts about user control in a game are a bit out of the ordinary — which is exactly what we're looking for! Here are a few:

"Speed could be selected by engine sound, that is, hold the joystick all the way forward. As speed increases, the

engine pitch increases. When it stops climbing, center the stick, then hold it forward again. You're in the next gear and moving faster."

From a different game concept: "*All control functions could become delayed during this game. If the amount of delay is based on your current score, the game would become increasingly difficult as you got better at it, but would still be playable by anyone.*" Steven's idea for delayed-control is based on part of his story, in which a team of "pilots" is controlling a distant battle-probe. The transmission of radio signals to and from the probe account for the delay. More ideas:

"*...one person can fly the jet, and one person can operate the weapons. (One can) use the fire button to control speed by step, maybe 1-2-3-2-1 in a loop. The gunner could use an assortment of weapons, chosing them from a menu and selecting one with the joystick.*"

These are all good ideas, and although I'd like to avoid trite violence in the game we write, such innovative user-control concepts could certainly come in handy.

Again, I'd like to thank everyone who wrote, even those not mentioned. I also hope that everyone who wrote will write again, and contribute more good ideas and suggestions to **Our Game.**

Deep thought.

It's worth thinking about the state of computer/video/arcade games. Here are a few observations to consider:

1) Video and arcade game manufacturers are hurting themselves by focusing in on one market; male teenagers. Only when a game has broken out of this rut has there been advancement and success. A good example is **Pac-Man**. Sadly, games still progress toward more realistic action-violence and higher-speed eye-hand coordination. More thought should be put into attracting females and people of different age groups.

2) Most video/arcade games deal in frustration. Yes, a person may be happy at moments during the game, but since he inevitably loses, he is bound to be frustrated. Is it possible to make a game that is more like a good movie, after which you walk away satisfied? Does this mean that a more expensive game that you would tend to play only once is the answer? Or how about games that are simply easier to play and enjoy?

3) People having the most fun in the arcades are usually with other people:

a) The **Centipede** wizard impressing his friends as they cheer him on.

b) The girl and her date taking turns at **Pac-Man**.

c) The two buddies having a go at **Joust**.

What is needed now are more two-player, simultaneous cooperative and/or competitive games. We can hope for (and work toward) computer networks that support real-time graphics, sound, and control, combined with multi-player interaction.

4) Arcade games are designed to make money. They gives short game play and have little or no depth. Not only should arcade games play more like good home computer games (possibly with more advanced hardware), but home computer games should stop trying to be like the arcades. In the future, will arcade games still strive for special hardware to remain superior, or will a room full of computers and "rent-a-games" be called an arcade?

Taking the above into consideration, it's possible to conceive of a "new genre" of computer games that breaks away from the faults of the arcades:

a) Games that give real satisfaction instead of a score;

b) Game play limited not by arbitrary number of "lives," but instead by the intelligence and creativity of the player;

c) Games that do not depend on violence and special effects as the only means of attracting an audience;

d) Games that give the player something more than an increased pulse rate — games that can teach you something.

(I'd like to thank Doug Crockford, author of **Galahad and the Holy Grail**, for inspiring this little outburst of idealism.)

If you, the reader, would like to respond to these ideas, don't hesitate to write to **Our Game**.

Of prototypes and programming.

Last issue we discussed the four steps toward writing a video game, given a good idea. Just to refresh your memory, here are those steps:

1) Transform the idea.

2) Develop experimental programs and prototypes.

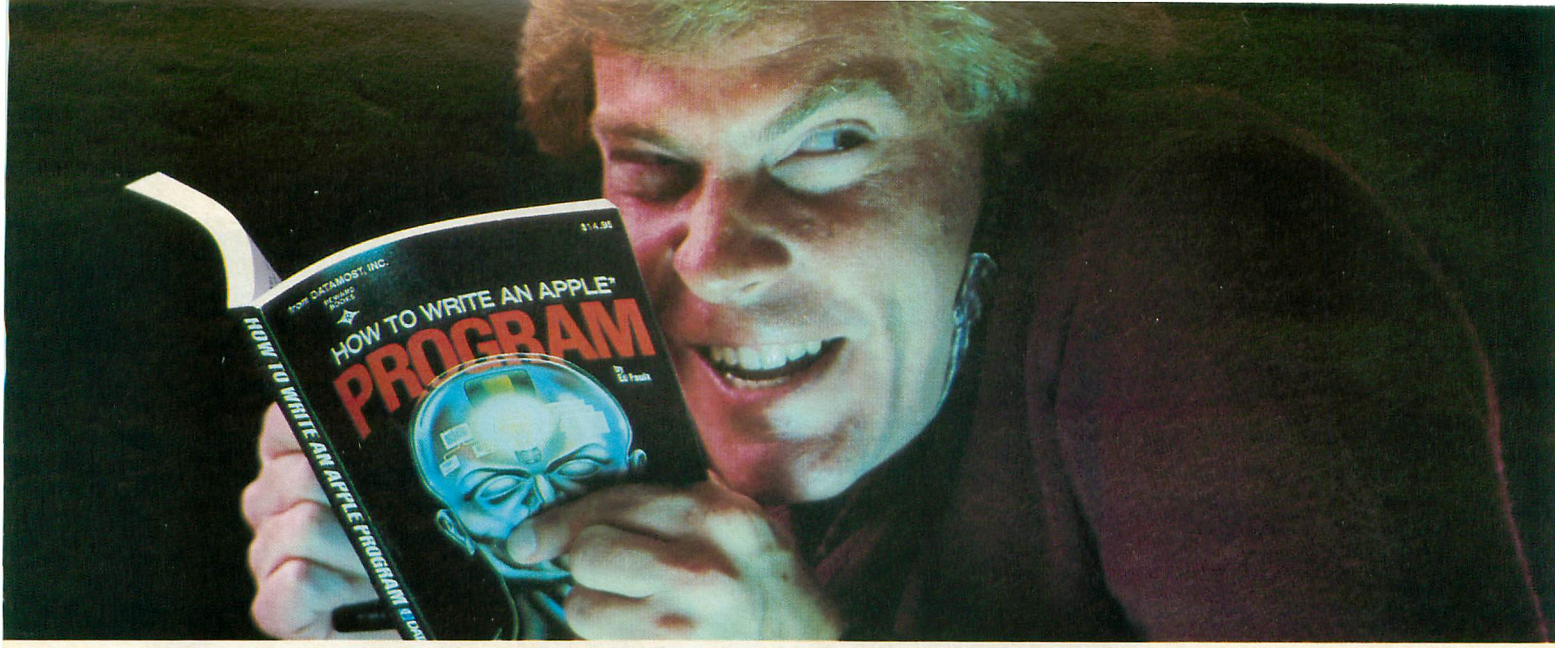
3) Write the game.

4) Document.

Last time we talked about step one. Step two concerns "prototypes." What is a prototype? Why would you want to develop one?

Well, imagine you wanted to build a giant robot hamster named Biggles. This hamster could be 20 feet high, 50 feet long, and weigh quite a bit more than you or I. Also imagine that Biggles would be equipped with the latest space-age technology, including a built-in automatic frobozz, a fusion-powered quarkatron, and an electric can-opener.

Before you built the actual giant robot hamster, you'd probably build a prototype; a smaller, simpler version of the real thing. This prototype might be only 4 feet high, 10 feet long, and not be equipped with any of those space-age frills mentioned above. You'd build this prototype in order to test your basic design — to make sure that Biggles, when finally built, would really work.



Outsmart your computer.

Show your computer who's boss. Earn its respect. With a Datamost book.

No matter what age or ability level you're at, we have the right book that talks just to you. And your computer. Whether you own an Apple.* An Atari.* Or just about any brand.

All our books are incredibly easy to understand.

Which will make it incredibly easy to understand your computer.

What If You're Scared Of Books About Computers?

Don't be.

Our books are written in friendly, familiar American English. Highlighted with cartoons. And illustrations. So they're fun to read. As well as educational.

And there's over 30 books to choose from. Basic computer learning books to programming books to coloring books.

You won't be bombarded with complicated programming routines until you're ready for complicated programming routines. And no funny technical talk until you've reached level where you don't think it's funny.

We'll tell you what you need to know. And then, when that's understood, you can take the next step toward outsmarting your computer.

Before You Spend Big Bucks On A Computer, Spend Little Bucks.

Once you've decided, kind of, almost, nearly, what type of computer you think you'd like to buy, buy a Datamost book.

It'll help you understand the Atari or the Commodore* or the Apple of your eye.

Before you spend a lot of money. Before you take your computer home.

You'll make a better purchase decision. Because you'll understand what you're doing. And what you and your computer can do together.

So there won't be any misunderstandings to ruin your new relationship.

How To Get The Most Out Of Your Computer.

Get the most out of our minds.

DATAMOST INC™

The most out of our minds.™

Datamost books for every computer.

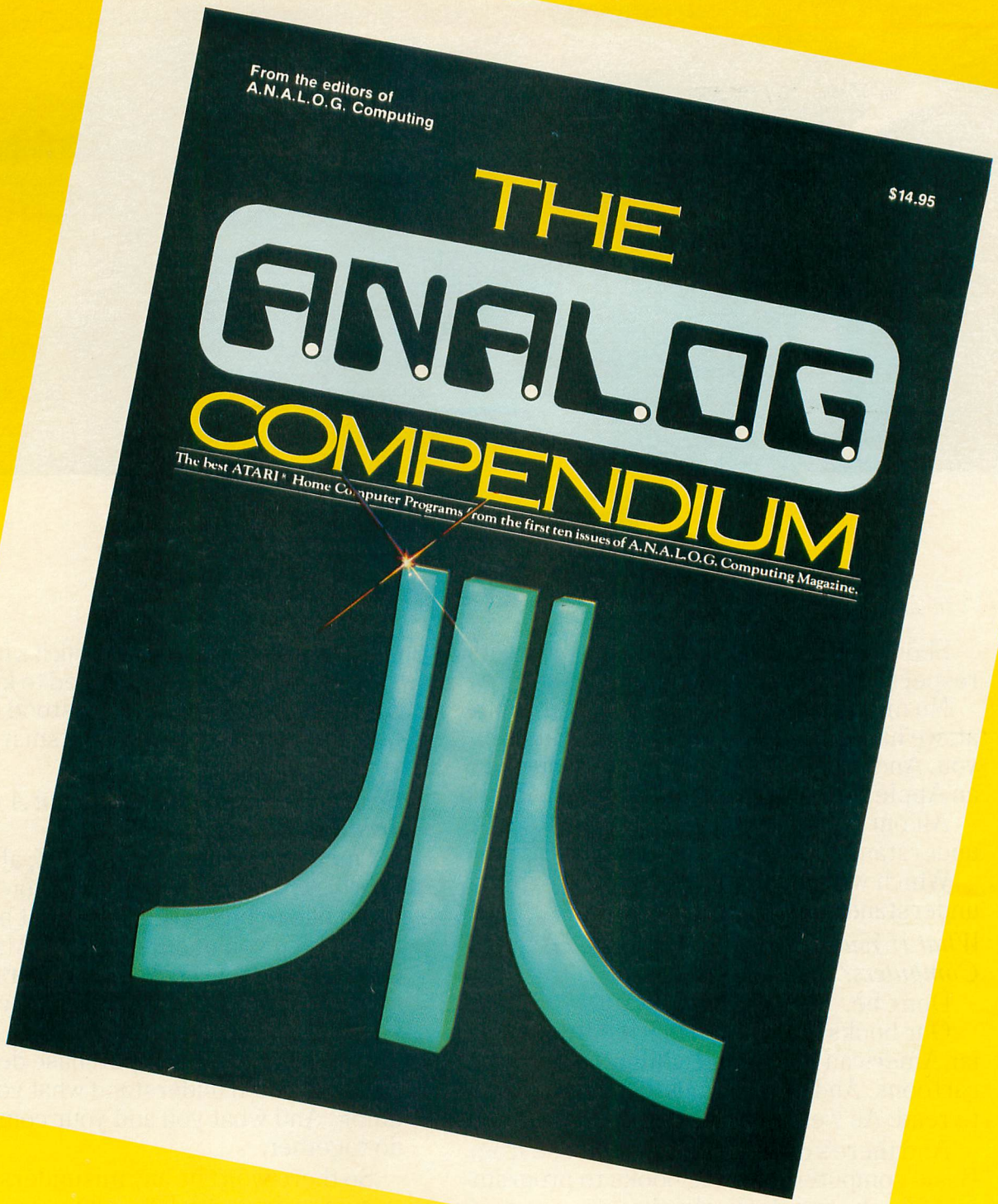
Everybody. Every level.

Pretty smart, huh?

Datamost, Inc., 8943 Fullbright Ave., Chatsworth, CA 91311, (213) 709-1202

*Atari is a trademark of Atari Computer.*Apple is a trademark of Apple Computer.

*Commodore 64 is a trademark of Commodore Business Machines, Inc.



The **A.N.A.L.O.G. Compendium** contains the best ATARI programs from the first ten issues of **A.N.A.L.O.G. Computing** Magazine, including new material not previously published. It is packed with utilities, educational and business programs, including an updated version of Tony Messina's "Disk Tool." And, to top it off, The **Compendium** contains the best game programs ever published in any magazine, including *Maniac*, *Stuntman* and the **A.N.A.L.O.G.** favorite, *Fill 'Er Up!*

The **A.N.A.L.O.G. Compendium** is available at selected book and computer stores, or you can order it direct. Send a check or money order for $14.95 + $2 shipping and handling to:

A.N.A.L.O.G. Compendium
P.O. Box 615
Holmes, PA 19043

Or you can order by phone with MasterCard or VISA. Call toll free: 1-800-345-8112 (in PA, call 1-800-662-2444).

For orders outside the U.S., add an additional $5 air mail, $2 surface.

SUPPLIES ARE LIMITED, SO ORDER YOUR COPY TODAY!

Prototypes of game programs are a lot like those large, mechanized rodents. They, too, are smaller, simpler versions of the real thing. They, too, can help you when it comes time to work on the final version.

Here are some concrete reasons for writing a prototype of a game:

1) To find out if it is really fun;

2) To do #1 without committing yourself to a big project;

3) To benefit from your mistakes (when writing the real version);

4) To make the final effort much easier and more organized, so that you can concentrate on the details.

Here are some enjoyable things to do with a prototype:

2) Have people *playtest* it, to find out if the game concept could be improved in some way;

3) Get a hard copy (list the program on paper), so you can refer to the prototype when writing the real thing.

One nasty thing that you should never do to a prototype is to tack things onto it and expand it until it becomes the final version. Imagine the pain of a prototype robot hamster being stretched into a giant robot hamster. The same damage is inflicted by lazy programmers trying to get away with only one version: a prototype/final program.

A close relative of the prototype is the "EP," or "experimental program." You write EPs before you write the final program in order to test untried or difficult techniques.

It's just like building Biggles. If you had never designed an automatic frobozz before, you might want to try building one separately before you make Biggles. This way, you can work on the experimental frobozz without interfering with the building of Biggles himself.

When programming, you may ask yourself this question: "Why don't I work on the experimental program as part of my prototype?" The reason is that if you did, it wouldn't be a simple prototype anymore. The whole purpose of organizing your design process into EP and prototype is to make things simpler, not more complex.

Dr. Ralph and the Martians.

Let's review the process of using prototypes and EPs. Say you're working on a video game called **Martian Big-Game Psychiatrist.** In this game, which you've worked out on paper, all sorts of strange Martian creatures (lions, elephants, and tigers, for example) are psychoanalyzed by the big-game psychiatrist, Dr. Ralph.

Early on in the development of this game, you'd want to write at least two programs. The first, a prototype, would be a no-frills version of the game; no title-page, little sound, and very rudimentary graphics. The strange Martian creatures could be orange squares, and Dr. Ralph could be a blue blip. You write this prototype for all the reasons mentioned above; most important of all, to determine if the game is fun.

The other programs you write would be EPs. One of them would be a program demonstrating the complex animation of the strange Martian creatures. Another EP might allow the user to operate a fairly complex looking Dr. Ralph and his examination couch with the joystick and keyboard.

The knowledge you gain from writing all of these programs, the prototype and the EPs, would definitely help you when writing the final version of **Martian Big-Game Psychiatrist.**

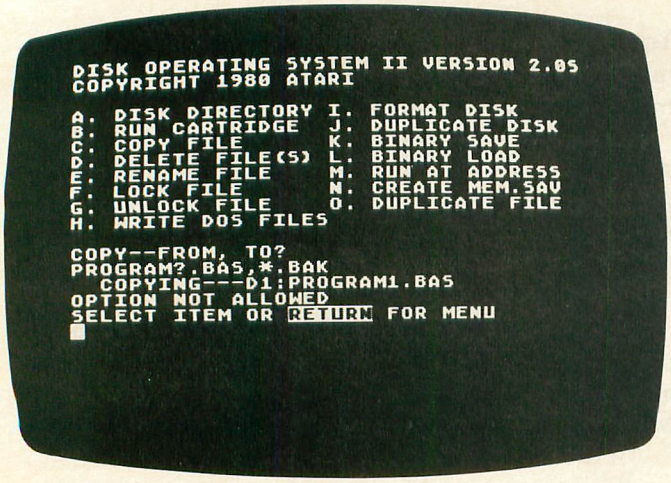
All this talk about prototypes and EPs is well and good, but how should you go about going the actual programming itself? Stay tuned for next issue, when we'll talk about "top-down" programming, and clear up other dreadfully mysterious topics as well.

Can't get no satisfACTION!

Since this issue of A.N.A.L.O.G. has the theme of tools and utilities, it's worth noting a new programming language that may become important to the game-writing public: **Action!** by Optimized

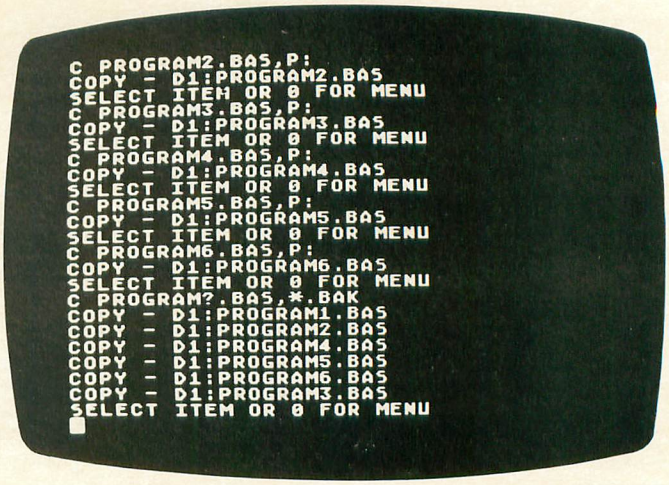
(continued on page 139)

ATARI DOS 2.0S

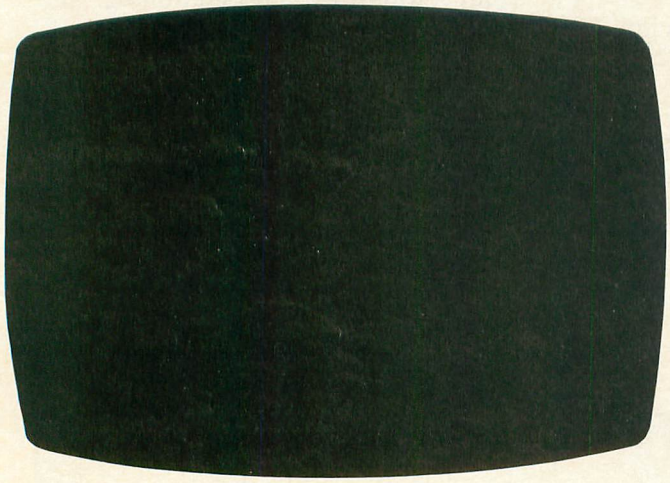


Six Lines of Work Space

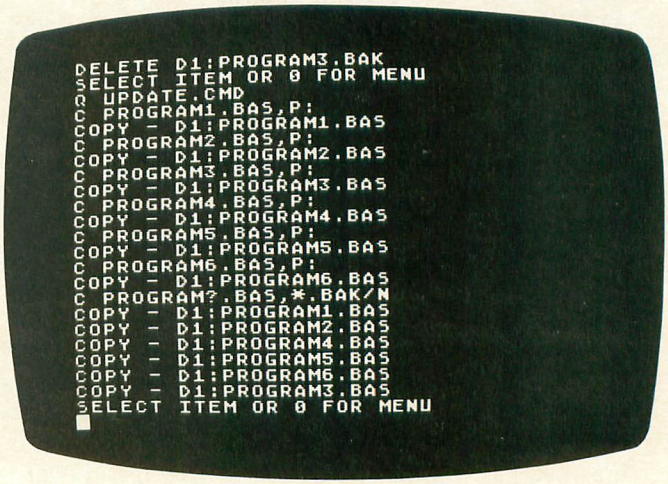
DOS-MOD



Twenty-Four Lines of Work Space



None



Command Files

Upgrade your ATARI DOS 2.0S with the modifications, enhancements, and new features of DOS-MOD.

DOS-MOD makes ATARI DOS an even more useful operating system, yet it is completely compatible with all your exisiting programs. DOS-MOD has features you would expect to find in more sophisticated systems. Compare them and see.

FULL SCREEN USE. DOS-MOD allows you four times more workspace on your screen. One-line commands and queries, a compressed menu, and a minimum of screen-clearing lets you see more of what you've been doing.

COMMAND FILES. Execute a whole sequence of commands in one easy operation. DOS-MOD's new commands and expanded functions give you a more powerful system.

EXPANDED WILDCARD CAPABILITY. In DOS-MOD, the wildcard conventions are more general and provide you greater control over COPY, DELETE, and RENAME operations.

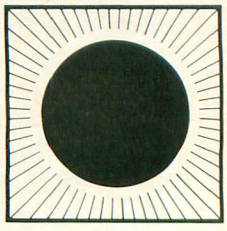
BUGS ELIMINATED. Eleven bugs in ATARI DOS 2.0S have been fixed. And the BREAK instruction has been trapped, facilitating recovery when your program gets lost.

FRIENDLY TUTORIAL. DOS-MOD's on-screen interactive tutorial helps you learn the features and uses of ATARI DOS and DOS-MOD.

GREAT PRICE. NO RISK. Only $35 for single-density and $50 for double-density version includes diskette with tutorial. DOS-MOD is guaranteed to make your ATARI DOS a more useful tool, or you may return it within 30 days for a full refund.

Order your DOS-MOD today. Send a check or money order to Eclipse, 1058-G Marigold Court, Sunnyvale, California 94086. Or call (408) 246-8325.

ECLIPSE



Systems Software. (See a full review of **Action!** in a future issue of **A.N.A.L.O.G. Computing**.)

Action! was created with one major intent — to be the fastest high-level language for the 6502. In that it succeeds; it is far faster than BASIC, FORTH, C, and just about any other language you can name other than assembly language. As a matter of fact, in most tasks **Action!** takes only twice as long as machine language.

Action! programs a lot like C, one of the most popular languages for minis and mainframes. For you hackers, the language is so fast because it is non-recursive (and therefore doesn't have to maintain a stack).

One of the major features of the **Action!** system is that it is more than just a language. It is a whole enviroment, complete with editor, monitor, and compiler. The whole system is packaged as one bright-orange cartridge.

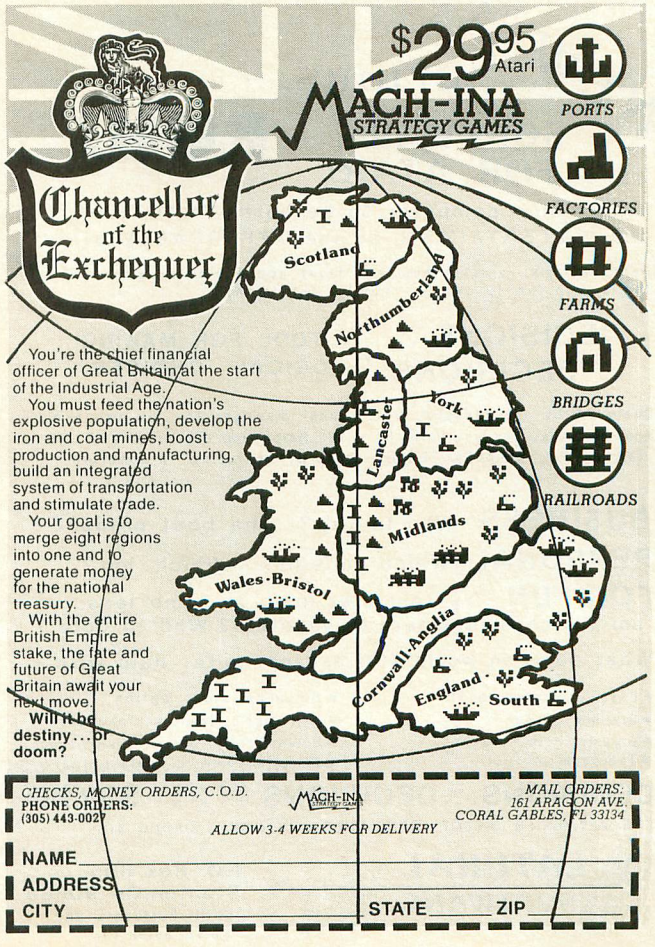
Action! is easy to use because having a real editor makes changing your program a snap, and because compilation time is very short — about 5 seconds for a 12K (object code) program. (You must compile an **Action!** program before you run it.)

Why am I mentioning this language in **Our Game?** Because of its speed and ease of use, **Action!** is potentially the best game-development tool ever for the ATARI computer. However, there are some problems.

It seems that OSS did not spend much time ironing out rough spots in the cartridge and its instruction manual before they introduced **Action!**. The language itself has faults, as does the editor. Most mysterious in **Action!** is the problem of memory management; where does a programmer put data for character sets, player/missiles, etc.?

OSS claims they will eliminate most of these problems, and will allow any **Action!** owner to send in the cartridge for free updates. They also say that they'll be coming out with a utility disk that solves the memory management problem, and that includes several sample programs and utilities.

My advice to the advanced programmer: wait a while, then get **Action!** The language is slowly being improved, and once all the bugs are out it will be a definite winner. To the beginning programmer: beware. The **Action!** manual isn't nearly good enough to teach you the language. What OSS needs is a complete step-by-step tutorial book on **Action!**. Until then, no inexperienced programmer will be



able to comfortably learn the language.

As for **Our Game**, no — we're not going to do our computer game in **Action!**. BASIC is still the only "universal" high-level language for the ATARI, and that's what we're going to use.

Hey, Mr. Postman.

It's that time. Time for me to ask you for mail. Even a postcard will do. If you can't write, have someone tape you and send me the cassette. . . whatever you do, send me your IDEAS. Not only your ideas about the game we're going to write together in the coming months, but also your ideas on the state of computer games in general — after all, that's what this column is all about. I want to know what you think. You are (even if you don't know it) a creative individual with ideas different from everyone else's.

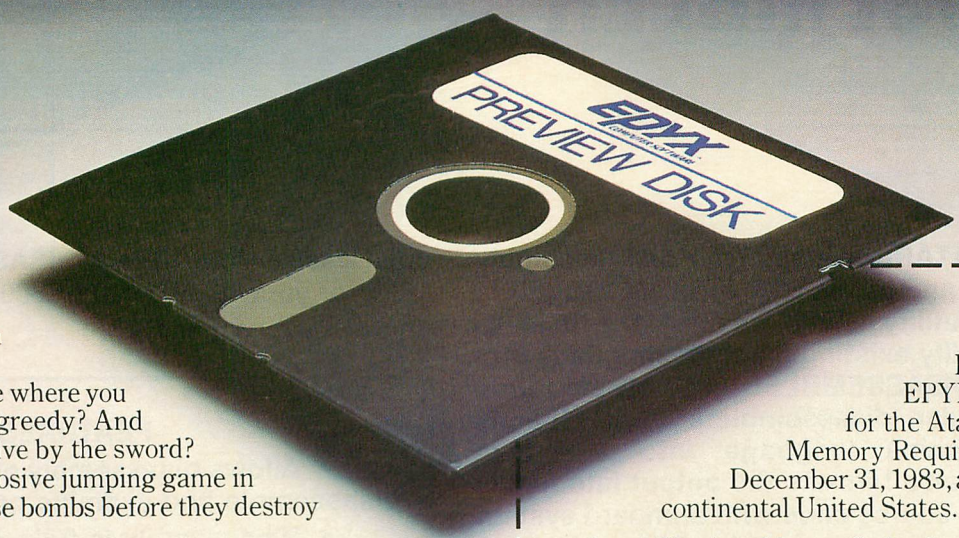
Share those ideas with us!

Send your mail to:

Our Game
c/o A.N.A.L.O.G. Magazine
P.O. Box 23
Worcester, Mass. 01603

Next time we'll be discussing all sorts of strange and wonderful ideas. Don't miss the next **Our Game**! □

PREVIEW FIVE HOT NEW EPYX COMPUTER GAMES FOR $2.50. OR LESS.



A road race game where the action's as fast in the pits as it is on the track?

An adventure game where you get rewarded for being greedy? And die a slow death if you live by the sword?

How about an explosive jumping game in which you have to defuse bombs before they destroy your command center?

You'd better be fast on the draw in one of the most famous shoot-outs of all time.

And watch out for mines while commanding a famous sub trying to torpedo moving ships.

Pitstop,™ Gateway to Aphsai,™ Jumpman Jr.™ and the Bally/Midway arcade classics Seawolf* and Gun Fight* are five hot new EPYX games. Now they're yours to preview on your Atari** home computer for only $2.50 (price covers shipping and handling).

Buy one of the games from any retailer, and we'll refund your $2.50. So the disk becomes yours for free.

Now make out your check or money order to EPYX and send it in with the coupon at right.

A disk for $2.50. Or for free.

Good fun should always be this cheap.

EPYX COMPUTER SOFTWARE

STRATEGY GAMES FOR THE ACTION-GAME PLAYER.

*Seawolf and Gun Fight are trademarks of Bally/Midway. **Atari is a registered trademark of Warner Communications.

Previews of Five New EPYX Games on a Disk for the Atari Computer. 48K Memory Required. This offer expires December 31, 1983, and is only valid in the continental United States.

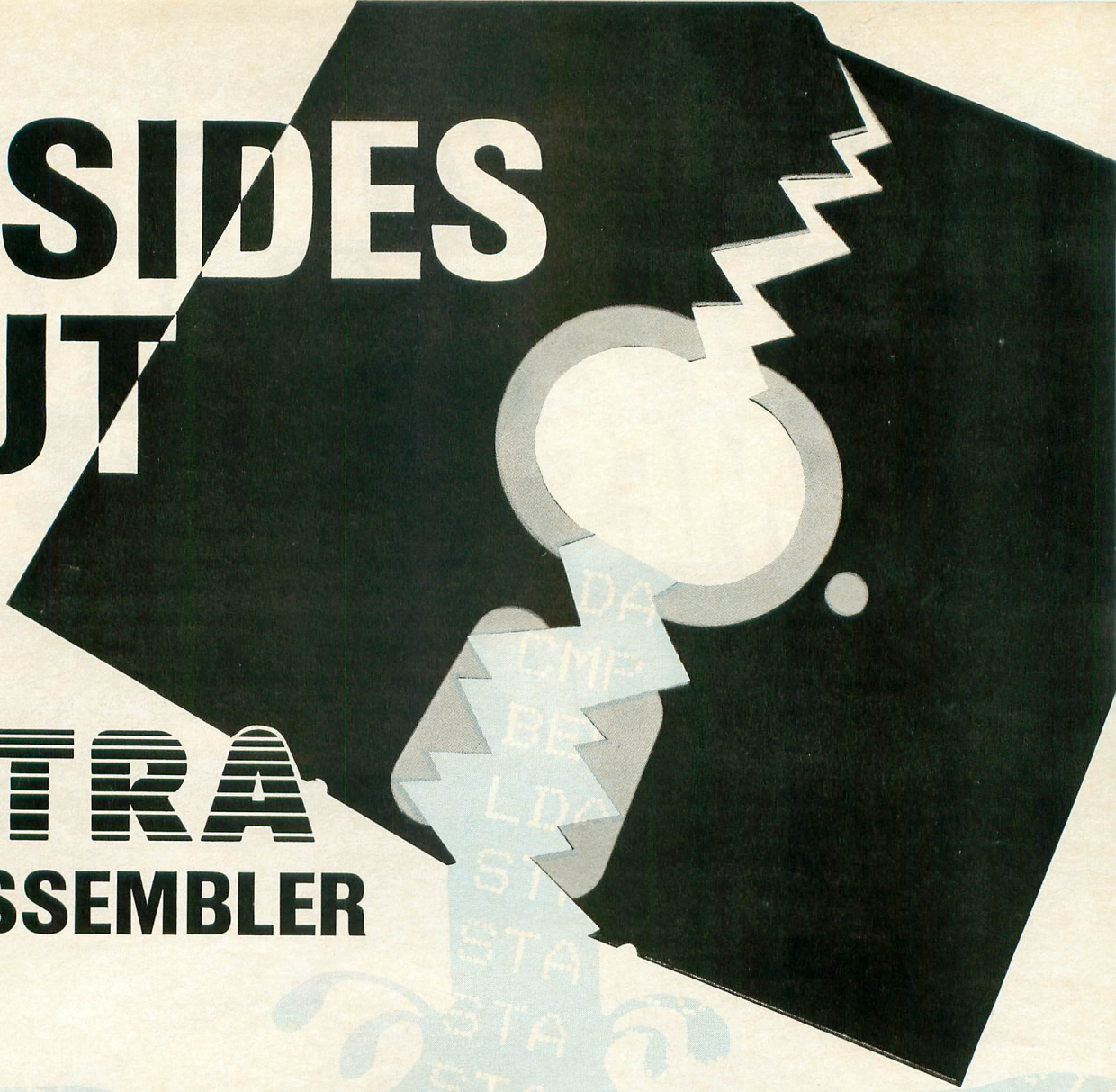
Send Check or Money Order for $2.50 to:
EPYX Preview Disk
144 Townsend St.
San Francisco, CA 94107

Name_____

Address_____ City_____

State/Zip_____ Phone_____

INSIDES OUT



ULTRA DISASSEMBLER

by Ralph Jones

Attention, ATARI owners: Your disk assembly problems are at an end. ULTRA DISASSEMBLER is a superb, new utility which enables the user to analyze and modify existing machine language programs. ULTRA DISASSEMBLER not only performs like other popular disassemblers (e.g., directly translating machine language into assembly language) but it also formats the output into highly readable pseudo-source code with standard system labels (where appropriate). Plus, ULTRA DISASSEMBLER writes the source code to disk in a form suitable for editing and reassembly with all major ATARI assemblers, including ATARI MACRO, ATARI Assembler/Editor, DATASM-65, EDASM and others.

Anyone with a working knowledge of 6502 assembly language and the ATARI operating system can put ULTRA DISASSEMBLER to work quickly and effectively.

The bottom line: If you need to analyze or modify machine language programs, you need a solid assist from ULTRA DISASSEMBLER, new for 1983 for your ATARI 400 or 800.

ATARI 400/800 32K DISK 052-0174 $49.95

FEATURES

- **Allows disassembly of object base, e.g., disassemble and examine code in a cartridge.**
- **Disassembly is fully reversible; i.e., the output code is written to disk as a pseudo-source file ready for reassembly with any of the major ATARI assemblers.**
- **Will read machine language code from DOS files or from a specified list of absolute disk sectors, not just from machine memory.**
- **Make "custom" changes to ULTRA DISASSEMBLER since Source Code is included.**
- **Extensive, user-oriented manual supplied in handsome flip-open binder.**

To order, see your local dealer. If he does not have the program, then call 1-800-327-7172 (orders only please) or write for our free catalog.
DEALER INQUIRIES ARE INVITED!

Adventure INTERNATIONAL ®

A SUBSIDIARY OF SCOTT ADAMS, INC.
BOX 3435 • LONGWOOD, FL 32750 • (305) 862-6917

A DISASSEMBLER IN ATARI BASIC

16K Cassette 24K Disk

by Maurice Elliot

I have been involved with computers for over 20 years now, so when I recently bought an ATARI 400 (for the kids, of course!) it wasn't long before I started tinkering with machine language. In order to speed up my learning, I tried examining machine language routines in published BASIC programs by translating the DATA statements back into an assembly listing form. But I quickly decided this was a task which the computer could handle much better than I could. The dis-assembler in my Assembler Editor cartridge was not much use, because I had no easy way to feed it the data out of the BASIC programs. So, since I couldn't find a published dis-assembler in BASIC, I sat down and wrote one.

The program I am about to describe is the outcome of this exercise. It is quite a simple program, though it has the flexibility — with minor changes — to dis-assemble machine language programs in formats other than the BASIC DATA statements for which it was designed. The listing it produces contains the address, operation code, and operand in the appropriate format. The address and operand are shown in both decimal and hexadecimal, so you don't have to get your conversion tables or calculator out to convert them. The program handles all 6502 instructions and addressing modes. Relative addresses in branch instructions are added to the program counter prior to printing. The result is the address of the instruction pointed to by the branch, so you can simply look it up in the listing. Indexed and indirect operands are shown in their 'proper' format (as used in the reference manual for the Assembler Editor cartridge). Of course, the output cannot include the labels or comments from the original assembly listing. The dis-assembly process ends when the program comes to an undecipherable operation code, a data value outside the range 0-255, the end of the DATA statements, or reaches the finish address you specify.

The dis-assembly process.

To print a pseudo-assembly listing, we must perform the following steps for each instruction in turn:

(a) look up the operation code to find the assembler mnemonic and addressing mode;

(b) obtain the operand (if any), and format it as required by the addressing mode;

(c) print the instruction, and increment the program counter (instruction address) to point to the next instruction.

Since I do not know whether you will need the instruction address and operand in decimal or hexadecimal, my program prints both so you won't have to convert manually from one to the other. The listing of one instruction just fits into one screen line.

Program organization.

The organization of this program is straightforward. Note first that I use high line numbers in the expectation they won't conflict with the ones in the DATA statements you want to process. The program begins with some initialization, then enters the main loop. Here it decodes the operation code to get the mnemonic and addressing mode. After printing the first half of the output line, it calls a subroutine specific to the addressing mode, which sets up and prints the operand. These subroutines are followed by the routine which sets up a one- or two-byte operand, and one which converts decimal numbers to hexadecimal form.

The program revolves around two main data structures — the instruction mnemonic table and the operation decoding table. The first table (named OPCODES) is built from the DATA statements in Lines 32310 through 32340. These list all the instruction mnemonics (as given in the Assembler Editor reference manual). The second table (named OPTABLE) is a numeric array with 256 entries. The entry at position I tells us about the operation code (if any) whose decimal value is 1. If the entry is zero, I

is not the value of any 6502 operation code. If I is the value of a 6502 operation code, the entry position I is a 3- or 4-digit number. The leading digit or two gives the position of the instruction mnemonic in the mnemonic table. The junior two digits give the addressing mode. I numbered the addressing modes arbitrarily, as follows:

Number	Description	Format
1	Implied	Blank (no operand)
2	Accumulator	Blank (no operand)
3	Immediate	$hh
4	Relative	$hhhh (gives address, not displacement)
5	Zero Page	$hh
6	Zero Page, X	$hh,X
7	Zero Page, Y	$hh,Y
8	Absolute	$hhhh
9	Absolute, X	$hhhh,X
10	Absolute,Y	$hhhh,Y
11	Indirect	($hhhh)
12	Indexed Indirect	($hh,X)
13	Indirect Indexed	($hh,Y)

For example, $06 is the operation code for a page zero ASL instruction. Thus the entry at index position 6 in OPTABLE is 305 (ASL is the third mnemonic, and zero page is addressing mode 5).

Other variables used by the program are as follows:

OPN$ — is used to hold an individual operation mnemonic.

HEXDIGIT$ — is a 16-character string containing the hexadecimal digits (0-9 and A-F). It is used by the hexadecimal-to-decimal conversion routine.

HEX$ — holds the result of the hexadecimal-to-decimal conversion routine.

PC$ — is used to arrange the instruction address for printing.

SETOPRND — holds the beginning line number of the routine that sets up the operand of an instruction, given its length.

DEC2HEX — holds the beginning line number of the routine that converts a decimal number into 4-byte hexadecimal format.

PC — contains the address of the current instruction (decimal).

FINADR — is the address at which the disassembly is to stop.

DECIMAL — is used for the decimal input to the routine DEC2HEX.

OPCODE — is the original operation code (decimal) of the current instruction.

OPN — is the entry in position OPCODE of OPTABLE.

OPNUM — is the operation number (high-order digits of OPN). It is used as an index into the operation mnemonic table.

ADMODE — is the addressing mode (junior digits of OPN). It is used to determine which routine to call to set up and print the operand.

ILEN — is set to the instruction length by the routine that handles its addressing mode. It is used to increment the program counter, and in the calculation of relative address targets.

OPRND — holds the instruction operand, in decimal form.

Program details.

The program begins on Line 30000 with a DATA statement. Its value (-1) acts as a sentinel marking the end of the user's data, and forcing the program to stop if the user does not enter an accurate finish address. Line 30010 dimensions all the arrays and strings. Lines 30020 through 30160 initialize all the tables and constants used by the program.

The next two lines ask the user for a starting and finishing address. The starting address is used to print addresses alongside the instructions in the listing. Its value is not critical unless you are PEEKing at a program in memory (see later), in

which case you must enter the proper address to start at. Similarly, the finish address is not critical — any large value is appropriate unless you want to finish at a particular address.

Line 30210 sets the tabs every 6 columns. This setting allows me to arrange the output for one instruction neatly on one line (though I have to "pack" the two forms of the instruction address into PC$ to do it).

Lines 30230 through 30310 form the main loop. For each instruction, this loop reads the operation code and decodes it, then formats and prints the instruction address and mnemonic. After this (on line 30300) it calls the routine (depending on the addressing mode) which formats and prints the operand. Finally, it increments the program counter and loops back (unless it has reached the finish address). At the end, Line 30320 restores the default tabs to 10.

The routines that print the operand in the correct format for each addressing mode are found in Lines 30500 through 31640. Each routine (except the first) performs the following steps:

(a) set the instruction length
(b) get the operand
(c) print the operand in the required format

The most complicated of these routines is the one for relative addresses, which adds the program counter and instruction length to the relative displacement to get an "absolute" address for printing.

The routine at Line 3200 gets the operand for this instruction. It uses the decimal-to-hexadecimal conversion routine at Line 32200 to convert the decimal operand to hexadecimal format, retaining only the two junior hexadecimal digits when the operand is only one byte.

The DATA statements in Lines 32310 through 32340 define the assembler mnemonics for all 6502 instructions. There are 56 mnemonics — 14 per DATA statement. And finally, with DATA statements in Lines 32600 through 32750 define the mnemonic and address mode for each possible operation code. There are 256 entries, held in 16 DATA statements of 16 entries each.

Operating instructions.

To dis-assemble a machine language program held as integers in BASIC DATA statements, proceed as follows:

1) LOAD the program containing the DATA statements, and LIST the DATA statements to cassette or disk.

2) LOAD the dis-assembler.

3) ENTER the DATA statements you saved in step one.

4) RUN the program.

When the program asks you for a start address, you should enter the location at which the machine language is being POKE'd if you want the listing to show true addresses. In reply to the prompt for finish address, you may enter the true finish address of the machine language program, or any larger number. Note that if the DATA statements have line numbers larger than 30000, you must renumber them before LISTing them in step one.

Alternative sources of input.

There are only three lines in the dis-assembler which obtain the machine language code it operates on. By changing these lines, you can use input forms other than DATA statements for the dis-assembly process.

For example, the machine language program may be set up as a string variable. Since the dis-assembler requires integers in the range 0-255, we use the ASC function on each character in the string. To dis-assemble such a program, make the following changes to the dis-assembler (substituting the name of the string variable for ML$):

```
30230 OPCODE=ASC(ML$(PC,PC))
32020 OPRND=ASC(ML$(PC+1,PC+1))
32070 X=ASC(ML$(PC+2,PC+2))
```

Since the program counter points to a position in the string, you must specify a starting address of 1, and a finish address equal to the length of the string. (You can't use *real* addresses anyway, because you don't know where in memory the string will be placed.)

The third possible source of input for the dis-assembler is directly from memory itself (either RAM or ROM). In this case, make the following changes to the dis-assembler:

```
30230 OPCODE=PEEK(PC)
32020 OPRND=PEEK(PC+1)
32070 X=PEEK(PC+2)
```

Here, you must use the proper address to start the dis-assembly, though the finish address may be any higher number.

Conclusion.

Since I wrote this program, I have used it to examine many machine language routines. It turns what was a pain in the neck into a relaxing, armchair task. If you choose to use it, I expect you will find it as helpful as I have. □

```
110 REM
120 REM ****************************
130 REM *                          *
140 REM *     DIS-ASSEMBLER        *
150 REM *                          *
160 REM * BY: MAURICE ELLIOTT      *
170 REM *     126 PARKVIEW PL. SE  *
180 REM *     CALGARY, ALBERTA     *
190 REM *     T2J 4W5  CANADA      *
200 REM *    TEL: (403) 278-9564   *
210 REM *                          *
220 REM ****************************
230 REM
```

```
30000 DATA -1
30010 DIM OPCODES$(168),OPTABLE(255),O
PN$(3),HEXDIGIT$(16),HEX$(4),PC$(10)
30020 RESTORE 32310
30030 REM SET UP ASSEMBLER MNEMONIC TA
BLE
30040 FOR I=1 TO 56
30050 READ OPN$:OPCODES$(3*I-2)=OPN$
30060 NEXT I
30070 REM SET UP TABLE TO DECIPHER OP
CODES AND ADDRESSING MODES
30080 FOR I=0 TO 255
30090 READ X:OPTABLE(I)=X
30100 NEXT I
30110 REM HEXADECIMAL DIGITS
30120 HEXDIGIT$="0123456789ABCDEF"
30130 REM SET UP LINE NUMBERS FOR SUBR
OUTINES
30140 SETOPRND=32010:REM ROUTINE TO SE
T UP INSTRUCTION OPERAND
30150 DEC2HEX=32210:REM ROUTINE TO CON
VERT A DECIMAL NUMBER TO HEXADECIMAL
30160 PC$="          "
30170 ? "START ADDRESS (DECIMAL)";:INP
UT PC:REM PC=PROGRAM COUNTER
30180 ? "FINISH ADDRESS (DECIMAL)";:IN
PUT FINADR
30190 IF FINADR<PC THEN 30170
30200 RESTORE :REM TO START OF USER'S
DATA STATEMENTS
30210 POKE 201,6:REM SET TAB WIDTH TO
6.
30220 ? "ADDR DEC    OPN   OPERAND
 DEC"
30230 REM MAIN LOOP
30240 READ OPCODE:REM READ OP CODE (WE
 HOPE) OF NEXT INSTRUCTION
30250 IF OPCODE<0 OR OPCODE>255 THEN 3
0320
30260 OPN=OPTABLE(OPCODE):IF OPN=0 THE
N ? "INVALID OP CODE AT ";PC:GOTO 3032
0
30270 OPNUM=INT(OPN/100):ADMODE=OPN-10
0*OPNUM
30280 DECIMAL=PC:GOSUB DEC2HEX:PC$(1)=
HEX$:PC$(6)=STR$(PC)
30290 ? PC$,OPCODES$(3*OPNUM-2,3*OPNUM
),
30300 ON ADMODE GOSUB 30510,30510,3061
0,30710,30810,30910,31010,31110,31210,
31310,31410,31510,31610
30310 PC=PC+ILEN:IF PC<=FINADR THEN 30
230
30320 POKE 201,10:REM RESTORE DEFAULT
TAB WIDTH.
30330 END
30490 REM THE FOLLOWING ROUTINES PRINT
 THE OPERAND FOR EACH ADDRESSING MODE.
30500 REM IMPLIED (MODE 1) AND ACCUMUL
ATOR (MODE 2)
30510 ILEN=1:REM INSTRUCTION LENGTH
30520 ? ""
30530 RETURN
30600 REM IMMEDIATE (MODE 3)
30610 ILEN=2:GOSUB SETOPRND
30620 ? "#$";HEX$,,OPRND
30630 RETURN
30700 REM RELATIVE (MODE 4)
30710 ILEN=2:GOSUB SETOPRND
30720 DECIMAL=PC+ILEN+OPRND-256*(OPRND
>127):GOSUB DEC2HEX
30730 REM CONVERT RELATIVE ADDRESS (DE
CIMAL) TO ABSOLUTE ADDRESS (HEX)
30740 ? "$";HEX$
30750 RETURN
30800 REM ZERO-PAGE (MODE 5)
30810 ILEN=2:GOSUB SETOPRND
30820 ? "$";HEX$,,OPRND
30830 RETURN
30900 REM ZERO-PAGE,X (MODE 6)
30910 ILEN=2:GOSUB SETOPRND
30920 ? "$";HEX$;",X",OPRND
30930 RETURN
31000 REM ZERO-PAGE,Y (MODE 7)
31010 ILEN=2:GOSUB SETOPRND
31020 ? "$";HEX$;",Y",OPRND
31030 RETURN
31100 REM ABSOLUTE (MODE 8)
31110 ILEN=3:GOSUB SETOPRND
31120 ? "$";HEX$,OPRND
31130 RETURN
31200 REM ABSOLUTE,X (MODE 9)
31210 ILEN=3:GOSUB SETOPRND
31220 ? "$";HEX$;",X",OPRND
31230 RETURN
31300 REM ABSOLUTE,Y (MODE 10)
31310 ILEN=3:GOSUB SETOPRND
31320 ? "$";HEX$;",Y",OPRND
31330 RETURN
31400 REM INDIRECT (MODE 11)
31410 ILEN=3:GOSUB SETOPRND
31420 ? "($";HEX$;")",OPRND
31430 RETURN
31500 REM INDEXED INDIRECT (MODE 12)
31510 ILEN=2:GOSUB SETOPRND
31520 ? "($";HEX$;",X)",OPRND
31530 RETURN
31600 REM INDIRECT INDEXED (MODE 13)
31610 ILEN=2:GOSUB SETOPRND
31620 ? "($";HEX$;"),Y",OPRND
31630 RETURN
32000 REM ROUTINE TO SET UP INSTRUCTIO
N OPERAND IN DECIMAL AND HEXADECIMAL
32010 IF ILEN=1 THEN RETURN :REM 1-BYT
E INSTRUCTIONS HAVE NO OPERAND
32020 READ OPRND:REM OPERAND FIRST (OR
 ONLY) BYTE
32030 IF OPRND<0 OR OPRND>255 THEN ? "
UNEXPECTED END AT ";PC+1:GOTO 30320
32040 IF ILEN>2 THEN 32070
32050 DECIMAL=OPRND:GOSUB DEC2HEX:HEX$
=HEX$(3,4)
32060 RETURN
32070 READ X:REM OPERAND SECOND BYTE
32080 IF X<0 OR X>255 THEN ? "UNEXPECT
ED END AT ";PC+2:GOTO 30320
32090 OPRND=OPRND+256*X:DECIMAL=OPRND:
GOSUB DEC2HEX
32100 RETURN
32200 REM ROUTINE TO CONVERT A DECIMAL
 NUMBER TO 4-BYTE HEXADECIMAL FORM
32210 HEX$="0000":I=4:DEC=DECIMAL
32220 IF DEC=0 THEN RETURN
32230 D=INT(DEC/16):D1=DEC-16*D
32240 HEX$(I,I)=HEXDIGIT$(D1+1,D1+1)
32250 DEC=D:I=I-1:IF I>0 THEN 32220
32260 RETURN
32300 REM OP CODE MNEMONICS
32310 DATA ADC,AND,ASL,BCC,BCS,BEQ,BIT
,BMI,BNE,BPL,BRK,BVC,BVS,CLC
32320 DATA CLD,CLI,CLV,CMP,CPX,CPY,DEC
,DEX,DEY,EOR,INC,INX,INY,JMP
32330 DATA JSR,LDA,LDX,LDY,LSR,NOP,ORA
,PHA,PHP,PLA,PLP,ROL,ROR,RTI
32340 DATA RTS,SBC,SEC,SED,SEI,STA,STX
,STY,TAX,TAY,TSX,TXA,TXS,TYA
32400 REM THE FOLLOWING 16 DATA STATEM
ENTS DEFINE A 256-ELEMENT
32410 REM TABLE TO INTERPRET EVERY POS
SIBLE BYTE VALUE (0-255) AS
32420 REM AN OP CODE AND AN ADDRESSING
 MODE.   WHERE THE ENTRY IS
32430 REM ZERO, THE BYTE VALUE DOES NO
T REPRESENT A VALID OP CODE.
32440 REM NON-ZERO ENTRIES ARE IN THE
FORMAT 'OPAD', WHERE OP=OP CODE
32450 REM NUMBER (=POSITION IN ABOVE L
IST OF MNEMONICS), AND AD=
32460 REM ADDRESSING MODE.  THE ADDRES
SING MODES ARE:
32470 REM 01=IMPLIED     02=ACCUMULATOR
   03=IMMEDIATE     04=RELATIVE
32480 REM 05=ZERO PAGE   06=ZERO PAGE,X
   07=ZERO PAGE,Y   08=ABSOLUTE
32490 REM 09=ABSOLUTE,X  10=ABSOLUTE,Y
   11=INDIRECT      12=INDEXED INDIRECT
32500 REM 13=INDIRECT INDEXED
32510 REM FOR EXAMPLE, A BYTE VALUE OF
 $21 IS INTERPRETED BY THE THIRD
32520 REM DATA STATEMENT ($2-), SECOND
 ENTRY ($-1). THE NUMBER THERE IS
```

```
32530 REM 212.   THUS, $21 (DECIMAL 33
) IS AN INDEXED INDIRECT 'AND'
32540 REM OP CODE (THE 2ND MNEMONIC, W
ITH ADDRESSING MODE 12).
32600 DATA 1101,3512,0,0,0,3505,305,0,
3701,3503,302,0,0,3508,308,0
32610 DATA 1004,3513,0,0,0,3506,306,0,
1401,3510,0,0,0,3509,309,0
32620 DATA 2908,212,0,0,705,205,4005,0
,3901,203,4002,0,708,208,4008,0
32630 DATA 804,213,0,0,0,206,4006,0,45
01,210,0,0,0,209,4009,0
32640 DATA 4201,2412,0,0,0,2405,3305,0
,3601,2403,3302,0,2808,2408,3308,0
32650 DATA 1204,2413,0,0,0,2406,3306,0
,1601,2410,0,0,0,2409,3309,0
32660 DATA 4301,112,0,0,0,105,4105,0,3
801,103,4102,0,2811,108,4108,0
32670 DATA 1304,113,0,0,0,106,4106,0,4
701,110,0,0,0,109,4109,0
32680 DATA 0,4812,0,0,5005,4805,4905,0
,2301,0,5401,0,5008,4808,4908,0
32690 DATA 404,4813,0,0,5006,4806,4907
,0,5601,4810,5501,0,0,4809,0,0
32700 DATA 3203,3012,3103,0,3205,3005,
3105,0,5201,3003,5101,0,3208,3008,3108
,0
32710 DATA 504,3013,0,0,3206,3006,3107
,0,1701,3010,5301,0,3209,3009,3110,0
32720 DATA 2003,1812,0,0,2005,1805,210
5,0,2701,1803,2201,0,2008,1808,2108,0
32730 DATA 904,1813,0,0,0,1806,2106,0,
1501,1810,0,0,0,1809,2109,0
32740 DATA 1903,4412,0,0,1905,4405,250
5,0,2601,4403,3401,0,1908,4408,2508,0
32750 DATA 604,4413,0,0,0,4406,2506,0,
4601,4410,0,0,0,4409,2509,0
```

•

CHECKSUM DATA

(See p. 58)

```
110 DATA 77,348,851,900,857,288,187,98
4,752,715,847,350,85,815,872,8928
30020 DATA 659,648,535,738,676,508,385
,572,665,59,434,347,675,944,341,8186
30170 DATA 3,530,749,333,154,92,191,91
7,407,930,174,686,263,633,276,6338
30320 DATA 426,544,846,663,909,364,55,
799,944,956,56,567,945,511,802,9387
30740 DATA 46,61,864,946,749,58,925,94
7,357,59,923,941,354,53,588,7871
31110 DATA 943,870,54,855,944,353,55,6
0,945,357,56,741,946,207,57,7443
31500 DATA 879,946,313,58,777,947,357,
59,345,343,908,264,304,352,62,6914
32070 DATA 921,212,150,51,920,150,676,
85,400,336,64,789,693,956,144,6547
32340 DATA 330,317,564,518,55,172,843,
108,586,664,2,855,349,129,213,5705
32540 DATA 45,811,789,153,469,440,829,
889,286,217,6,4,517,726,629,6810
32740 DATA 791,635,1426
```

•

(continued from page 20)

"frequency" parameter, which accepts values directly in Hertz (cycles per second) instead of the arbitrary values used by BASIC. Logo's extended-range sound system can accurately reproduce frequencies from 14 Hz to beyond the limits of audibility.

TOOT's "duration" parameter tells Logo how long to play a sound before shutting it off. The maximum play time is 255 "jiffies" or a little over 4 seconds. The SETENVelope command lets you control the decay (rate of volume decrease) of your sound effects.

Lists and recursion.

Like its ancestor Lisp, the Logo programming environment is organized around two fundamental data types: words (atoms) and lists (groups of words). All data and procedure definitions in Logo can be broken down into words and lists. A Logo program is itself nothing more than a list!

Logo's unitized data structure gives the language a number of fascinating capabilities. For example, it's possible to write a Logo procedure that can compose new Logo procedures based on external data. You can then instruct Logo to execute the procedures it has written — all from within a Logo procedure. Think of the weird things you could do with a program-writing program!

Another idea central to the design of Logo is *recursion*, or defining a procedure in terms of the procedure itself. This may sound like the computer equivalent of a dog chasing its tail, but in the hands of a skilled user recursion can save hours of programming effort while yielding clean, readable code.

Atari Logo provides a surprisingly broad selection of list-manipulating primitives that can pull apart, analyze and synthesize Logo lists of any type. Once you get tired of playing with turtles, take an afternoon to explore the list-processing and recursive properties of Logo. I think you'll be pleasantly impressed.

Other goodies.

The Logo cartridge includes a built-in procedure editor that works like a mini text-processor. You call the editor with the word EDIT, followed by the name of the procedure you want to work on. Logo automatically clears the screen and displays the latest definition of the specified word. When you exit the EDIT mode (by hitting the ESCape key), Logo dumps the contents of the 3840-byte edit buffer directly into Logo's workspace, exactly as if you had typed it in manually. A useful side effect of this design is that any Logo commands which appear outside a procedure definition will be executed immediately.

Another feature of **Atari Logo** that deserves special mention is its "dribble file" capability. Suppose you're a teacher who wants to keep track of a student's progress during a Logo session, but you don't have time to sit and personally monitor his/her performance. You can use **Atari Logo's** SETWRITE mode to open a disk file which will automatically record everything the student types on the TV screen. Later, you can step through the entire session (a line at a time, if you like) and see exactly what the student did or didn't do. Nice.

Still gnashing your teeth over ATARI BASIC's meaningless error codes? Then check out Logo's superb error-handling system, which tells the user exactly what went wrong in complete English sentences. "PROCEDURE DOESN'T LIKE X AS INPUT" is a whole lot friendlier than "ERROR — 8" in my book.

Documentation.

The documentation supplied with **Atari Logo** is well-written and reasonably complete. $99.95 gets you the Logo cartridge along with three books: a 160-page *Introduction To Programming Through Turtle Graphics* that offers a breezy, entertaining walk through the Logo environment; a 216-page *Reference Manual* that describes the usage and syntax of each Logo primitive in detail, with numerous useful examples; and a 16-page *Quick Reference Guide* that does exactly what its title suggests. They're all fully indexed, thoughtfully laid out and attractively printed to boot. Let's hope that future ATARI languages follow the example of Logo in this regard.

RAM and time usage.

Atari Logo is a hog-o when it comes to RAM usage. Its basic memory unit is a five-byte cell called a "node." Each word in a Logo program requires two nodes, plus an additional node for every two letters in its name. Numbers require two nodes apiece; lists use up one node for each element in the list, plus all the nodes required by each element! So the Logo list:

[LOGO EATS RAM FAST]

requires 16 nodes or 80 bytes of storage! Too bad ATARI didn't use bank-switching to make Logo "look" like an 8K cartridge — the extra 8K of user RAM would have been mighty handy.

Logo is nothing to cheer about when it comes to speed, either. The benchmarks I was able to devise suggested that Logo was at least two or three times slower than ATARI BASIC when it comes to simple repetitions and commands like .EXAMINE (POKE) and .DEPOSIT (PEEK). Recursive procedures make Logo *really* look like a turtle.

I hasten to point out that Logo was never intended to be a model of speed or memory efficiency. It's more concerned with teaching than with pinching bytes, and most educational users with a 32K system should never have to worry about running out of RAM. But Logo's greediness just might get you into trouble if you start getting ambitious with its list-

processing and recursive functions. Logo is the best introduction to these concepts you'll find on an ATARI computer, but serious list-hackers should start shopping for a bigger machine.

Dark mutterings.

No language system is perfect, and **Atari Logo** is no exception. Although the system appears to be free of any serious defects, I did run across a couple of "undocumented restrictions" the prospective user may want to watch out for.

My biggest gripe is the way the editing cursor "takes off" when you hold down one of the CTRL-arrow keys. Time and time again I found myself backing up the cursor after it had gone sailing past my desired editing point, completely out of control. I also experienced fatal editing lock-ups on more than one occasion. Even the SYSTEM RESET key wouldn't help me — which was probably just as well, since hitting RESET will erase your Logo program even if you aren't locked up.

The *Reference Manual* says that unused or "dead" words can be cleaned out of the system with the RECYCLE command. This is not completely true. If you attach a value to a variable and later edit that variable out of your program, the deleted variable will never go away, even if you write the program out to disk and load it back into Logo from a coldstart! The only way to permanently remove "deleted" variables is to use the ERN (Erase Name) command on the offending words.

I'm happy to report that **Atari Logo** does include a way to access machine-language subroutines, using the built-in .CALL command. But the manual doesn't say how to get your routines into memory, where to put them, how to pass data back and forth or how to return to Logo when you're finished. I suspect that the majority of Logo users will have very little need for machine language, but the info would be nice to have, just in case. And while we're on the subject of documentation, how about a chart that shows what frequencies to specify with TOOT when you want to play a certain musical note?

"This is the one."

The above comments notwithstanding, **Atari Logo** is one of the most intelligently-conceived and well-executed pieces of software ever published by ATARI. I highly recommend it to parents and educators as a tool for computer education, and to hobbyists interested in learning the fundamentals of list processing and recursive programming. If someone you know has been looking for a reason to buy an ATARI home computer for their family, they need look no further than this little cartridge (and maybe **Star Raiders**). □

INDEX TO ADVERTISERS

COMPUTER PALACE

Your Holiday Gift Headquarters!

ATARI®

We Know ATARI Computers... Call Us For The Latest Info: (503)683-5361

PINBALL (Much more than a game!) CONSTRUCTION SET

- A Fantastic Program!
- Design the Pinball Game of your dreams... or play one of five games included on the disk!

New from Electronic Arts — 48K Disk — **$35.10**

Blue Max

It's 1915...
You are a British Air Ace...
It's up to you to save the Empire...
Bombs away, God save the Queen!

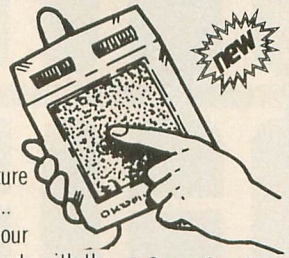
- Super scrolling screen!
- Dive, Climb, Turn

Another "Winner" from Synapse!

32K DISK/TAPE — **$31.50**

the Koala Pad

From Koala Tech.



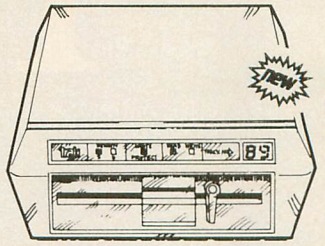
The future is here... Input your commands with the touch of a finger on your Koala Pad.

- Super Graphics & Drawing Program Included with Pad.

Retail $100.00

$87.00 + FRT

TRAK DISK DRIVE



- Single/or/Double Density

AT-1 DD. . . . . Only $348
AT-D2 DD. . . . . Only $398 + FRT

(AT-D2 drive includes built-in printer interface and printer buffer!)

Pole Position

Are You Fast Enough???

$44.90

The reviews are coming in on this one, and it sounds like another Winner From Atari!!! Super Scrolling Screen... Fast Speedway Action! 16K Cartridge.

Super Mailer PLUS

From Royal Software

48K DISK
$49.95
Dealer Inquiries Invited

- Store 500-2000 files per disk.
- Fast machine language sorts.
- Works on single or double density.
- Automatic delete of duplicates.
- Special coding feature.
- Official state abbreviations are built in.
- Re-label fields for unlimited uses.
- Merge files. • Create sub-files.
- Search files on any field.

SUPER MAILER + is perfect for organizing your Christmas card list... You'll never forget to send a card again!

Super Mailer + is the most "user-friendly" program available!!!

DUST COVERS

- Available for:
- Atari 400/800/600XL/800XL/1200XL computers
- Atari 410 program recorder and 810 disk drive
- Epson MX-80 and FX-80 printers

Only $8.95 EACH

ALL TIME FAVORITES

Star Raiders (C)	38.20
Eastern Front (D,T)	27.90
Pac Man (C)	39.50
Temple of Apshi (D,T)	35.10
ZORK I,II,III (D) EACH	35.10
Centipede (C)	39.50
Missle Command (C)	29.70
Shamus (D,T)	31.50
Ghost Hunter (D)	34.95
Ghost Hunter (T)	29.95
Choplifter (D)	31.50
Choplifter (C)	39.90
Wizard & Princess (D)	29.90
Astro Chase (D,T)	26.90
Sammy the Seaserpent (T)	15.90
Miner 2049er (C)	44.90
Defender (C)	39.50
Crossfire (D,T)	26.90
Preppie (D,T)	26.90
Ali Baba & the 40 Thieves (D)	28.90
Master Type (D)	35.10
Diskey (D)	44.90

HARDWARE

Atari 600XL computer (16K)	148.00
Atari 800XL computer (64K)	248.00
Atari 1020 printer	249.00
Atari 1025 printer	477.00
Atari 1027 printer	319.00
Atari 1050 disk drive	399.00
Atari 1010 program recorder	86.00
Atari CX77 tablet	79.00
Panasonic green screen monitor	199.00
Color monitor	299.00
Astra 1620 disk drive	488.00
Rana 1000 disk drive	349.00
Trak AT-1 disk drive	348.00
Koala Pad graphics pad	87.00
Wico "Boss" joystick	19.95
Dust covers (computers/drives/printers) EACH	8.95
Flip-N-File (diskette or cartridge)	24.90
Bright-colored diskettes (SSDD, 10/box)	29.95
Regular diskettes (SSSD, 10/box)	19.95
TV/Monitor swivel base	39.95
Oak TV/Monitor stand	29.95
Oak TV/Monitor stand (for Percom drives)	49.95

SOFTWARE

Advanced Music System (D)	29.95
Blackjack Casino (D,T)	17.95
Pascal (D)	49.95
Quarxon (D,T)	29.95
Phobos (D)	29.95
Block Buster (D,T)	17.95
Basic Commander (D)	31.50
Ali Baba & the 40 Thieves (D)	28.90
Jeepers Creepers (D)	26.90
Castle Wolfenstein (D)	26.90
Spitfire Ace (D,T)	26.90
Poker Sam (D)	22.45
Strip Poker (D)	31.50
Zaxxon (D,T)	29.90
Fort Apocalypse (C)	35.10
Fort Apocalypse (D,T)	31.50
Pharoah's Curse (D,T)	31.50
Shamus (C)	35.10
Shamus (D,T)	31.50
Necromancer (D,T)	31.50
Zeppelin (D,T)	31.50
Survivor (D,T)	31.50

SOFTWARE

Qix (C)	39.50
Way Out (D)	35.10
Repton (D)	35.10
Dark Crystal (D)	35.10
Frogger (D,T)	31.40
Arcade Machine (D)	52.50
Bank Street Writer (D)	62.50
Shadow World (D,T)	31.50
Quasimodo (D,T)	31.50

TOP-TEN PROGRAMS

Pole Position (C)	44.90
Donkey Kong (C)	44.90
Blue Max (D,T)	31.50
Baseball (D,T)	28.90
Pinball Construction Set (48K, Disk)	35.10
Football (D,T)	28.90
Monkey Wrench II (not a game) (C)	54.00
Ultima II (D)	52.50
Archon (D)	35.10
Dig Dug (C)	44.90

COMPUTER PALACE

OPEN Monday - Saturday, 9 to 6
2160 W. 11th Avenue Eugene, Oregon 97402

MasterCard VISA

USE YOUR CREDIT CARD & CALL
Toll Free 1-800-452-8013
(Orders Only, Please)
For Information Call (503) 683-5361

Shipping Note: UPS & Parcel Post-$2.00—UPS Air-$3.90
Shipping on hardware & paper supplis extra—Call for $.
24 & 48 hour shipping available... Call (503) 683-5361

We specialize in *ATARI Compatible Products*
So we know what works best!!!
Call for the latest info... (503) 683-5361

THEY'RE HERE.


ESC
1 !
2 "
3 #
4 $
5 %
6 &
7 '
8 (
9)
0
CLEAR
INSERT
DELETE BACK SPACE
BREAK
CLR TAB SET
Q
W
E
R
T
Y
U
I
O
P
RETURN

River Raid™ and Kaboom!® are here. And your Atari® home computer just became more fun than ever.

Because River Raid and Kaboom! have been re-designed to take full advantage of home computer capabilities.

Far beyond anything you've ever experienced before in video games.

Unique graphics, crisp detail and brilliant sound all come together with spectacular impact.

River Raid,™ the battle adventure up the "River of No Return."

The realism of Carol Shaw's River Raid is utterly amazing.

It challenges your reflexes, stamina and strategic savvy as you battle your way up the winding river.

Enemy choppers. Jets. Tanks. Hot-air balloons. Ships. All out to blow you to smithereens.

But you strike back.

Keeping one eye on the ever-changing terrain.

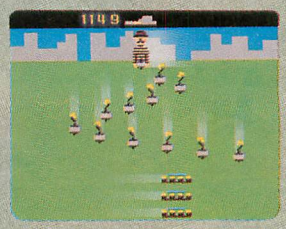
Bridges. Mountains. Canyon walls. Islands.

One false move and it's curtains.

And if you're up to it, now you can skip easier sections of the river and get right to the heart of the battle.

Everything that made River Raid such a smashing hit is here. And tons more.

Kaboom!®, the award-winning game of catch with buckets and bombs.



Larry Kaplan's Mad Bomber is back. The buckets are back.

But now, in Paul Willson's adaptation, you can drop the bombs, while someone else tries to catch them.

Imagine dropping bombs. Faster and faster.

To the ever-quickening pace of the 1812 Overture.

You shift right. Left. Back again. All the way right.

He misses! You win!

Now it's your turn to catch. The pressure mounts.

The bombs start flying. You dash to catch them.

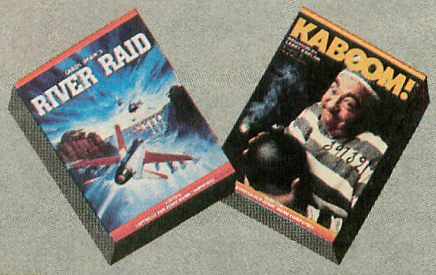
And so it goes on into the night.

And everytime you hit a new high score, it's displayed after the game, just like at the arcade.

Kaboom! and River Raid for your Atari home computer.

They're here.

Just for the fun of it.



ACTIVISION®

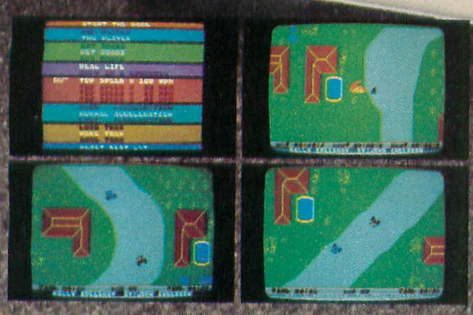
John Anderson's

RALLY SPEEDWAY

SLIP INTO SOMETHING TURBO-CHARGED

like the road-scorching speedster in RALLY SPEEDWAY — the game that turns your ATARI into a full-blown, four-wheeled demon! Take on our demanding course, or use the unique Construct-A-Track option to design your own. Push yourself to the limit for a better lap time, or get a friend in on the action for a one-on-one duel to the finish line — it's your choice!

CALL TOLL FREE 1-800-327-7172



MAKE TRACKS TO YOUR DEALER TODAY!

ATARI 400/800/1200 CARTRIDGE

053-0171 . . . . . $49.95

If your dealer doesn't have Rally Speedway in stock, phone 1-800-327-7172 with your order, or write us at Adventure International, P.O. Box 3435, Longwood, FL 32750 for your FREE catalog.

DEALER ORDERS WELCOME

155

Adventure® INTERNATIONAL

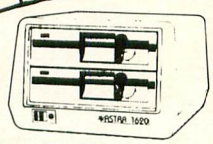
A DIVISION OF SCOTT ADAMS, INC.

COMPUTER PALACE

IF IT'S FOR THE ATARI... WE HAVE IT!!!

► Look Here For Holiday Gift Ideas! ◄

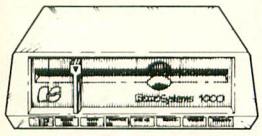
Astra 1620 DISK DRIVE



Double Density
Double Drives

Only $488 + FRT

Rana 1000 DISK DRIVE



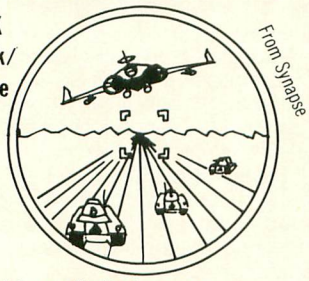
Single or Double Density

Hi-Tech, Low-Profile Styling

Only $349 + FRT

Dimension X

32K Disk/Tape



From Synapse

What has unlimited power... scrolling perspective... & excitement???

THIS DOES!!! **$31.50**

FREE Catalog

With any order, or send $1 (refundable with purchase).

Send us your name & address for FREE FLYERS with Special Offers!

Over 1000 items for your Atari...

- Synapse
- Percom
- Datasoft
- Epson
- Atari
- APX
- A.I.
- On-line
- Amdek
- Hayes
- Visicalc
- Much more...

Use Your Credit Cards!

We honor VISA, American Express, & MasterCard.

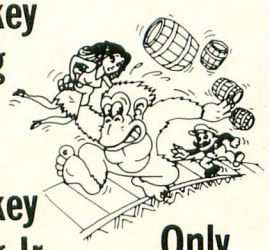
The Monkey Wrench II

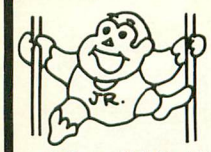
$54.00

This is a BASIC & Machine-language **Programmers Aid** for the 800. It plugs into the right slot, and works with Atari BASIC to add 16 new direct-mode commands(Incl: Auto line numbering; Delete lines; Renumber BASIC; Hex-Dec conv.; Find string; Move lines; Exchange string; Disk directory from BASIC; Mem test; More...). Also includes a monitor with 15 commands to interact with the 6502. This has been proven to be a very helpful tool when writing your own programs!

A cartridge for the right-hand Slot of your Atari 800!

Donkey Kong (or) Donkey Kong Jr.





Only $44.90 Each

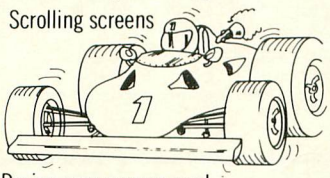
From Atari

Both on 16K Cartridge from Atari

Rally Speedway

Want'a Race???? Here's your game!!!

Scrolling screens



Design your own course!

From Adventure International

16K Cartridge **$44.90**

From Gamestar

FOOT BALL

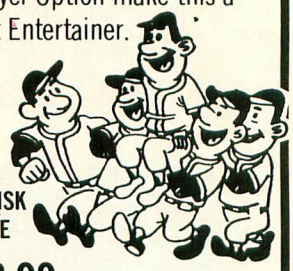
- One or Two Players
- 24K RAM Disk or Tape

★ Pass
★ Run
★ Kick

Disk/Tape
$28.90

Baseball

Super Sharp Colorful and Smooth Graphics, with 2-Player Option make this a Great Entertainer.



32K DISK or TAPE

$28.90 From GAMESTAR

SUPER SPECIALS!

LIMITED QUANTITIES—WHILE SUPPLIES LAST!

(PLEASE LIST YOUR SECOND-CHOICE WHEN ORDERING)

	Reg.	Special	
Kid Grid (D).	30.00	17.95	
Gorf (D).	40.00	19.95	
Wizard (D).	40.00	19.95	
Shamus II (T).	35.00	17.95	
Ricochet (D,T).	20.00	9.95	Wow
Rescue at Rigel (D).	30.00	17.95	
Survival of Fittest (C).	40.00	17.95	
Magic Mail (D).	60.00	29.90	
Bug-Off (D).	30.00	14.95	
Snake Byte (D).	30.00	9.95	Super
Alien Swarm (D,T).	30.00	9.95	Look
Genetic Drift (D,T).	30.00	19.90	
Chicken (D,T).	35.00	19.90	
Slime (D,T).	35.00	19.90	

	Reg.	Special	
6502 Disassembler (T).	20.00	6.95	Steal
Turmoil (C).	35.00	15.95	
6502 Assembler (T).	30.00	12.95	
Platter Mania (D).	40.00	17.95	
Sentinel One (D,T).	35.00	9.95	Wow
Guardian Gorn (D,T).	35.00	9.95	
Intruder (D,T).	35.00	9.95	Look
Atari Mailing List (T).	30.00	9.95	
Hockey (D,T).	30.00	9.95	
Pogo Man (D,T).	30.00	9.95	Great
Serpentine (D,T).	30.00	14.95	
Apple Panic (D).	30.00	14.95	
Visicalc (D).	250.00	149.95	
Capt. Beeble (C).	40.00	24.95	

EDUCATIONAL PROGRAMS

Math Tic-Tac-Toe (D,T).	17.95
Type Attack (D).	35.10
Pre-School I.Q. (D,T).	21.90
Vocabulary Builder (D,T).	21.90
Preparing for the S.A.T. (D,T).	112.00
Math UFO (D,T).	24.95
Spelling Genie (D).	24.95
Kindercomp (D).	26.90
Rhymes & Riddles (D).	26.90

BUSINESS PROGRAMS

Personal Finance (D).	67.50
Letter Perfect (D,C).	89.90
Data Perfect (D).	89.90
Spell Perfect (D).	89.90
Atari Writer (C).	87.90
Text Wizard & Spell Wizard (D).	SET 72.50
Super Mailer Plus (D).	49.95
B/Graph (D).	89.90
Payroll (D).	195.00
Label-It (D).	19.95
Visicalc (D).	149.95

BOOKS

Your Atari Computer.	16.95
Technical Users Notes.	26.90
Compute's 1st Book of Atari.	12.95
Compute's 2nd Book of Atari.	12.95
Machine Language for Beginners.	12.95
Mapping the Atari.	14.95
Atari Source Book.	12.95
Atari Graphics.	12.95
De Re Atari.	17.90
Inside Atari DOS.	19.95
Kids & the Atari.	19.95
Visicalc Book.	14.95
Book of Atari Software.	19.95

MAGAZINES

ANTIC, Vol. 2/No. 6.	2.50
ANTIC, Vol. 2/No. 5.	2.50
ANTIC, Vol. 2/No. 4.	2.50
ANTIC, Vol. 2/No. 3.	2.50
ANTIC, Vol. 2/No. 2.	2.50
A.N.A.L.O.G., No. 13.	2.50
A.N.A.L.O.G., No. 12.	2.50
A.N.A.L.O.G., No. 11.	2.50

COMPUTER PALACE

OPEN Monday - Saturday, 9 to 6

2160 W. 11th Avenue Eugene, Oregon 97402

MasterCard VISA

USE YOUR CREDIT CARD & CALL

Toll Free 1-800-452-8013

(Orders Only, Please)

For Information Call (503) 683-5361

Shipping Note: UPS & Parcel Post-$2.00—UPS Air-$3.90
Shipping on hardware & paper supplis extra—Call for $
24 & 48 hour shipping available... Call (503) 683-5361

We specialize in *ATARI Compatible Products*
So we know what works best!!!
Call for the latest info... (503) 683-5361

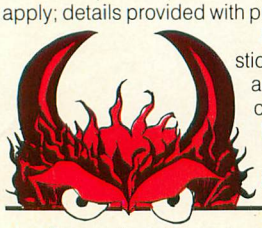
EXCITING NEW ATARI* SOFTWARE FROM JAY GEE

DEVIL'S DARE™

A STRATEGY BOARD GAME. BEAT THE COMPUTER AND GET THE GAME FREE!

A superb implementation of the Ancient Japanese game of GO-MOKU. Employs artificial intelligence techniques to provide challenging advanced levels of strategy.

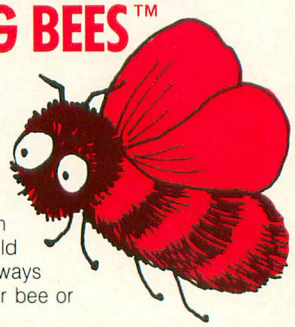
1–4 players race against time to get five tokens in a row before the devilish computer does. Defeat his fiendish strategy three times in a row and he will tell you his SECRET CODE number. Send us your number and we will refund your full purchase price. Rules and restrictions apply; details provided with purchase.



Requires 16K and joysticks. Available on diskette and cassette for $24.95 including tax and shipping. MONEY-BACK OFFER VOID IN STATES WHERE PROHIBITED BY LAW.

ATTACK OF THE SPELLING BEES™

AN EDUCATIONAL ARCADE GAME. AIM, SHOOT, AND LEARN TO SPELL!



Words with missing letters bounce around the screen while big colored letters rotate around the outside. Maneuver your bee to snatch up the correct letter and fire it at the incomplete word (aim carefully!). Score points for correct letters and extra for completing a word. Score even more for shooting a letter of your own color. Don't hit the "Wild Wasp" or you will lose points; shoot him at a word and he always adds a letter. Block your opponent's flight path with your bee or shoot down letters so he can't get them.

A random arrangement of walls makes the game different every time. The game comes complete with 250 built-in words. When you get those mastered, order additional word lists from our catalog of dozens; we have lists on all grade levels and in all categories.

For 1 or 2 players, requires 32K and joysticks, available on diskette only for $39.95 including tax and shipping.

FREE TRIAL VERSION ($5.00 S & H) available on diskette and cassette. A reduced, simplified, single-player version with 50 words. Return for 10% discount on full version.

Jay Gee entertainment software runs on all ATARI Computers and uses high-resolution graphics.

TO ORDER: Bring this ad to your local dealer or mail coupon at right (check or credit card info) or **CALL OUR TOLL FREE ORDER DESK 800-227-1617.** In California 800-772-3545–Please ask for extension 217.

The Jay Gee Programming Co.
7185 Blue Hill Drive
San Jose, CA 95129

DEALER INQUIRIES INVITED

*"ATARI" is a trademark of Atari, Inc.

Please provide the following information for our records:

Computer System: _____

_____

Memory: _____ K _____ Disk Drives

Printer: _____

_____

Cassette Recorder: ☐ Yes ☐ No

850 Interface: ☐ Yes ☐ No

Modem: ☐ Yes ☐ No

ATTACK OF THE SPELLING BEES:
☐ Send _____ copies ($39.95 each, tax, S & H included)
☐ Send Demo Version on: ☐ Diskette ☐ Cassette ($5.00 S & H)
☐ Send brochure (free)

DEVIL'S DARE:
☐ Send _____ copies on: ☐ Diskette ☐ Cassette ($24.95 each, tax, S & H included)

Payment: ☐ Check ☐ MasterCard ☐ VISA ☐ American Express

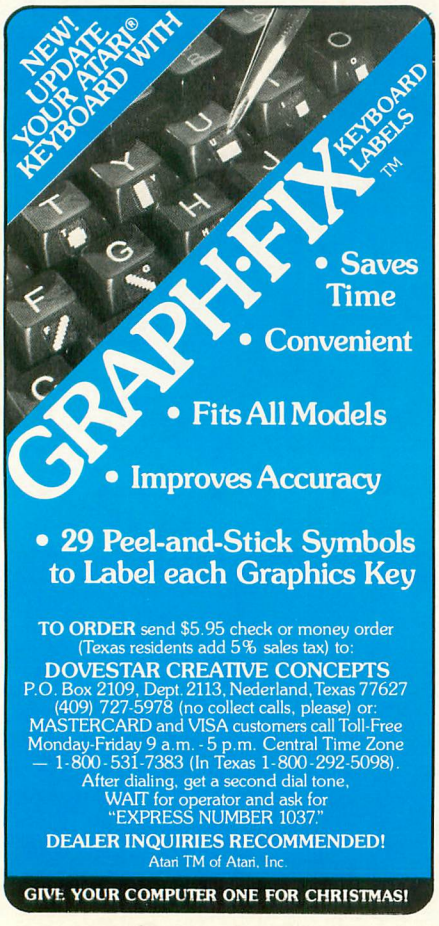
Credit Card Number: _____

Exp Date: _____ Signature: _____

Name: _____

Address: _____

City: _____ State: _____ Zip: _____



TO ORDER send $5.95 check or money order (Texas residents add 5% sales tax) to:
DOVESTAR CREATIVE CONCEPTS
P.O. Box 2109, Dept. 2113, Nederland, Texas 77627
(409) 727-5978 (no collect calls, please) or:
MASTERCARD and VISA customers call Toll-Free Monday-Friday 9 a.m. - 5 p.m. Central Time Zone — 1-800-531-7383 (In Texas 1-800-292-5098). After dialing, get a second dial tone, WAIT for operator and ask for "EXPRESS NUMBER 1037."

DEALER INQUIRIES RECOMMENDED!
Atari TM of Atari, Inc.

GIVE YOUR COMPUTER ONE FOR CHRISTMAS!

TAX COMMAND

NOW YOUR ATARI PUTS LINE-BY-LINE CONTROL OF TAX PREPARATION AT YOUR FINGER TIPS.

Calculations are automatic. All you do is enter your tax information. Tax Command does all mathematical calculations for you. **Built-in tax tables eliminate guesswork.** No more finding the right column down and right line across. Tax Command has the 1040 tax tables built right in. So it zeros in on your refund (or tax payment) amount automatically. **Tax Command is fast, easy!** It gives your computer more than just the 1040 tax form. Tax Command tells you when to income average, and has the forms you need. Plus Schedule A (Itemized Deductions, including medical), capital gains and losses, and more. Anyone who can read can use Tax Command. And the best feature of all . . . just $24.95 plus $2.00 for shipping and handling. Wisconsin residents add 5% tax. Specify computer type, tape or disk.

Practical Programs, Inc.

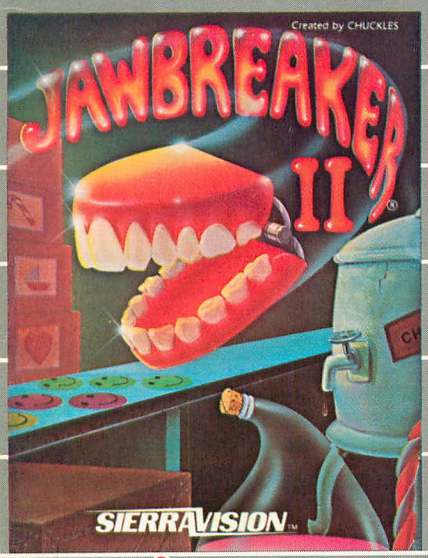
P.O. Box 93104-Y • Milwaukee, WI 53203 • (414) 278-0829
Available at fine computer stores everywhere, or by ordering direct.

III

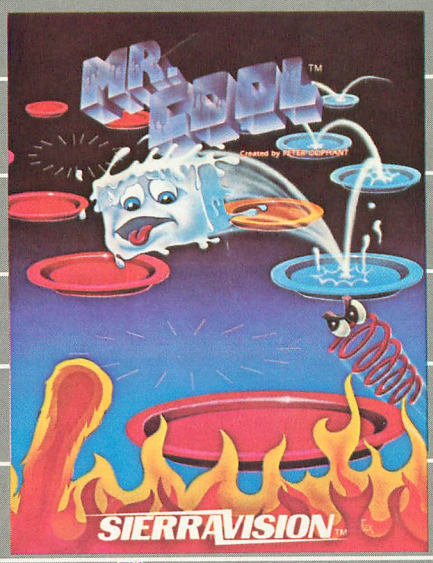
CARTRIDGES FOR

ATARI®

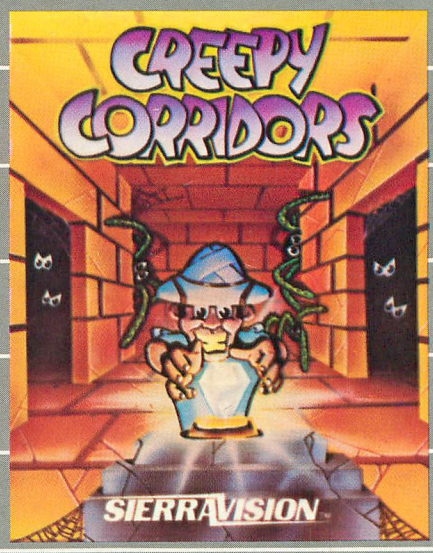
HOME COMPUTERS



JAWBREAKER® II - Now a new and deliciously different action game to follow the all-time best seller. A challenging new twist on an old favorite.



MR. COOL™ - A furnace of fun. Hop about the pyro-pyramid while avoiding menacing hot springs and shooting fireballs.



CREEPY CORRIDORS™ - Crawling critters will be chasing you all over the screen in this addicting action game. An exciting race to gather jewels. . . and save your life.

Mr. Cool, Creepy Corridors and Jawbreaker II are now available on cartridge for the Atari 400, 600, 800, 1400 and 1450 computers. These and other titles are selectively available for the Apple II, Commodore 64, IBM-PC and Vic-20 from your local software dealer.

Sierra ON-LINE

YOU'LL LIKE IT!

ACTUALLY, YOU'LL LOVE IT!

Here's why! TRIAD is nine separate pulse-pounding games in one package. Select one of nine weird alien menaces to battle from the Master Board. But careful — these guys can be tricky! Defeat three aliens in a row tic-tac-toe style and you earn a crack at the next skill level.

The whole neighborhood'll thrill to TRIAD's crisp colors, super sounds and the oddest assortment of alien "baddies" ever assembled on one screen. Snappy arcade excitement for all ages!

TRIAD — from Adventure International. Why get one game when you can have nine?

TRIAD — YOU'LL ~~LIKE~~ *love* IT!

(152)

APPLE (DOS 3.3 REQUIRED) 48K DISK $34.95
ATARI 48K DISK $34.95
COMMODORE 64 TAPE & DISK $34.95
TRS-80 CoCo 32K TAPE $34.95

ANOTHER WINNER FROM

Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.

P.O. BOX 3435 • LONGWOOD, FL 327
1-800-327-7172