NUMBER 7

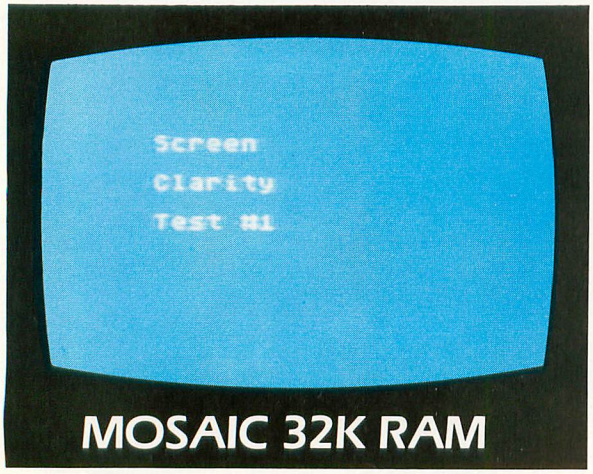
$2.25

ANALOG COMPUTING

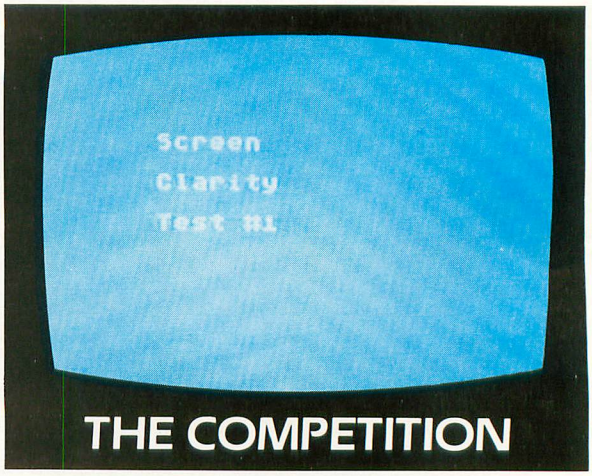
THE MAGAZINE FOR ATARI® COMPUTER OWNERS

THE MOSAIC RAM SYSTEMS FOR ATARI*

CLEARLY THE BEST



THE SCREEN CLARITY TEST



YOU CAN SEE THE MOSAIC DIFFERENCE

WHAT THE EXPERTS HAD TO SAY

A.N.A.L.O.G. 400/800 MAGAZINE said in a 32K RAM board comparison test: "The Mosaic 32K RAM showed no sign of interference and gave the best screen clarity" and "Mosaic uses what we feel are the highest quality components which may improve reliability".

INTERFACE AGE said after replacing the Atari memory board with a Mosaic 32K RAM: "Once in place there is no noticeable change in screen clarity" and "in view of its excellent performance it should be a serious choice for those Atari owners intending to expand their memory."

Each Mosaic RAM board gives you more than just the best screen clarity but also the best in reliability, flexibility, and compatibility. The Mosaic RAM systems offer you the best in every way — these features prove it.

- Works in both the Atari 400*& 800*computers.
- 4 year guarantee.
- Complete instructions.
- Test cycled 24 hours for reliability.
- Gold edge connectors for better reliability.
- Quick no-solder installation.
- Full flexible memory configuration.
- Can be used with 8K, 16K and future products.
- Designed to take advantage of Atari 800's superior bus structure.
- Designed for inter-board communication in Atari 800.
- Allows Atari 800 to have 2 slots open for future expansion.
- Always the best components used for superior screen clarity and reliability.
- Low power design for safety and reliability.
- Available companion board ($5) to allow running The Mosaic RAM systems independent of other boards.

THE MOSAIC RAM SYSTEMS FOR THE SERIOUS ATARI OWNER.

THE MOSAIC 32K RAM

For the serious Atari owner. This is the 32K RAM board you've been hearing about. It has every feature you could want from a RAM board and more features than any other board offers. Each Mosaic 32K RAM comes with complete instructions so in a few minutes you will have expanded your Atari 800 to 48K RAM. The Mosaic 32K RAM works as well in the Atari 400, but we suggest the NEW Mosaic Expander.

THE MOSAIC EXPANDER

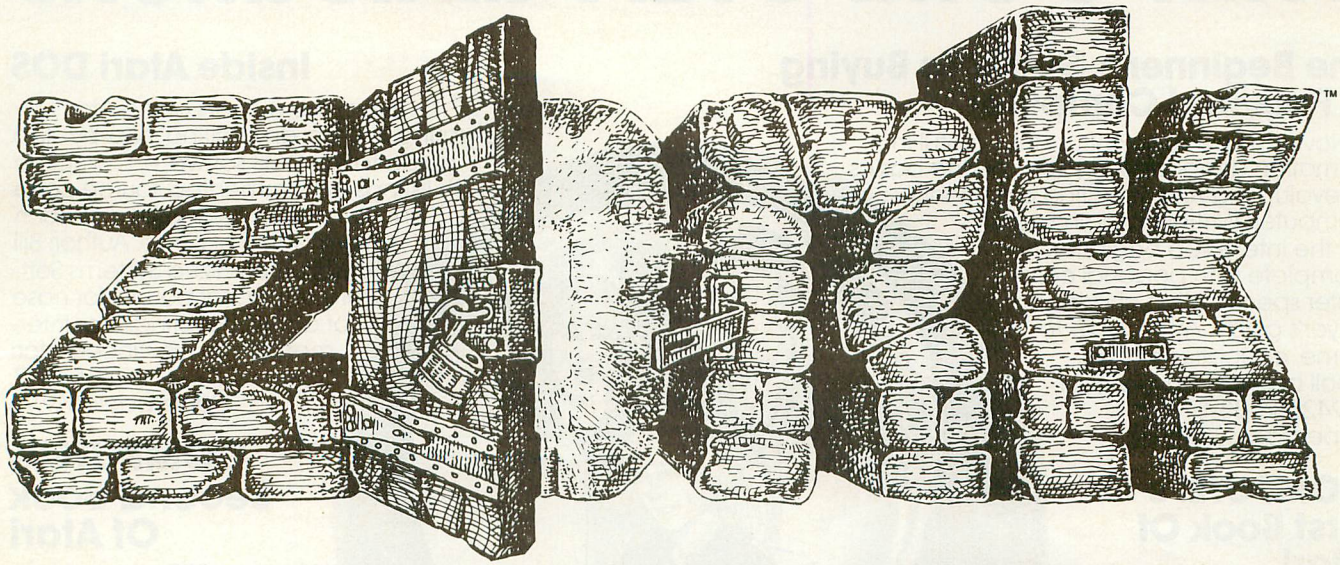
This is the most effective way you can expand to 32K RAM for your Atari 400 computer. And at almost the 16K price! The revolutionary Mosaic Expander is a memory board with 16K RAM in place and open slots to add 16K more from the Atari 16K board that comes with your Atari 400. Each board comes with complete instructions so in a few minutes you will have expanded to 32K RAM.

*Trademark of Atari, Inc.

MOSAIC
ELECTRONICS

P.O. Box 708 Oregon City, Oregon 97045
503/655-9574, Toll Free 1-800-547-2807

Many are the doors that lead to the underground.



The top prose adventure bestsellers, ZORK I and ZORK II.

Now available for

Apple® II, ATARI®400/800, IBM Personal Computer, NEC PC-8000, CP/M,® and PDP™-11.

Few are they who come back alive.
Your greatest challenges lie ahead—and below. ZORK I: The Great Underground Empire, and ZORK II: The Wizard of Frobozz will transcend any computer age adventure you've ever encountered.

Those who live to tell the tale of ZORK's mysteries speak of becoming immersed in a dimension where the extraordinary is commonplace, and the dangers are as real as any in human experience. It is said that once you have known the forbidden realm, you are changed—and that he who dares ZORK's depths twice will never be the same again.

Now your door to ZORK has opened to beckon you below. The journey begins where Infocom's new worlds of INTERLOGIC™ prose adventure—including the new mystery thriller DEADLINE™—await you, at fine computer stores everywhere.

Open your door to the underground—here's $2 for openers!

To get your $2 ZORK rebate by mail, just send Infocom this coupon with your completed warranty card and sales slip from ZORK in their original form. No reproductions will be accepted. Limit 1 rebate for household, address or organization. Offer good only in U.S.A. Void where prohibited, taxed or otherwise restricted. Rebate request must be postmarked before midnight 7/31/82. Infocom is not responsible for lost, late or misdirected mail. Allow 4 to 6 weeks for delivery.

Infocom, 55 Wheeler St., Cambridge, MA 02138
OFFER EXPIRES JULY 31, 1982

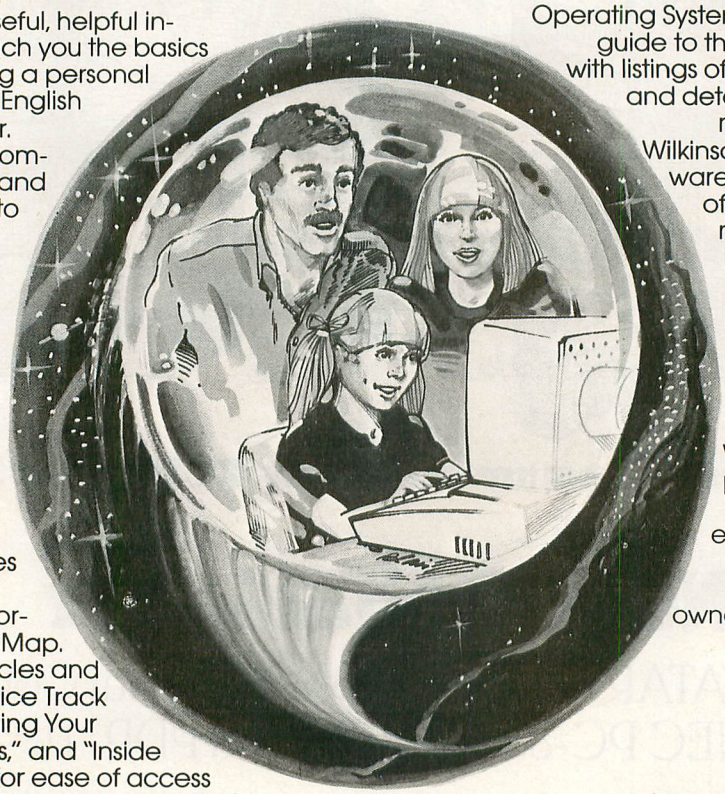
B1

INFOCOM™
55 Wheeler Street, Cambridge, MA 02138

Turn To The Future With COMPUTE! Publications

The Beginner's Guide To Buying A Personal Computer

A Novice's handbook of useful, helpful information designed to teach you the basics of evaluating and selecting a personal computer. Written in plain English for the interested beginner. Complete with personal computer specification charts and buyer's guide. Applicable to home, educational, and small business buyers. ISBN 0-942386-03-5. Paperback. $3.95.



COMPUTE!'s First Book Of Atari

192 pages of useful, informative applications and programs from **COMPUTE!** magazine issues now out of print. Includes previously unpublished information including Memory Map. Contents include such articles and programs as "Adding A Voice Track to Atari Programs," "Designing Your Own Atari Graphics Modes," and "Inside Atari BASIC." Spiral bound for ease of access to listings. For Beginner level to Advanced Atari users. ISBN 0-942386-00-0. Paperback. $12.95.

Inside Atari DOS

From the authors of the Atari Disk Operating System, an exciting step-by-step guide to the DOS software. Complete with listings of commented source code and detailed explanations of each module of code. Author: Bill Wilkinson, Optimized Systems Software, Inc. Spiral bound for ease of access to listings. For Intermediate to Advanced Atari Users. ISBN 0-942386-02-7. Paperback. $19.95

COMPUTE!'s Second Book Of Atari

With over 200 pages of unpublished Atari information specially selected by the editors of **COMPUTE!** Magazine, this new release is an excellent resource for Atari owners and users. Spiral bound. ISBN 0-942386-06-X. Paperback. $12.95

COMPUTE! Books invites dealer inquiries. Call the Toll Free Number below for Dealer Information.

COMPUTE! Magazine

A Monthly encyclopedia of informative applications articles and programs. **COMPUTE!** features articles, programs, and columns covering the spectrum of home and educational computing. Monthly reviews, complete BASIC and machine language listings of games, utilities, applications such as "Programming Your Home Insurance Inventory," "Real Estate Investment Analysis," "Telecommunications: How To Use A Modem," and much more. Written for children and parents, educators, novices to advanced programmers. Principal editorial coverage is Atari, Apple, Commodore PET/CBM, and VIC-20. Editorial coverage is expanding to include TI-99/4A, Sinclair ZX-81, and Radio Shack Color Computer. Latest issue: 224 pages.

Mail to: **COMPUTE!** Publications, P.O. Box 5406, Greensboro, NC 27403 USA

COMPUTE! MAGAZINE

My Computer Is:

☐ Commodore PET/CBM ☐ OSI
☐ VIC-20 ☐ Radio Shack Color Computer
☐ Apple ☐ TI 99/4A
☐ Atari ☐ Other _____
☐ AIM ☐ Don't yet have one

For Fastest Service Call Our **Toll-Free** US Order Line **800-345-8112**
In Pennsylvania Call 800-662-2444

☐ $20.00 One Year US Subscription ☐ $25.00 Canada and Int'l Surface Mail
☐ $36.00 Two Year US Subscription ☐ $38.00 Europe Air Delivery (Foreign orders must be pre-paid in US Funds)
☐ $54.00 Three Year US Subscription

☐ Payment Enclosed ☐ VISA ☐ MasterCard ☐ American Express
Acct. No. _____ Expires ___/___

Name _____
Address _____
City _____ State _____ Zip _____
Country _____

COMPUTE! Books

For fastest service, in the US call **Toll Free 800-334-0868.**
In NC call 919-275-9809.

Quantity		Price		Shipping/Handling	
____	**Beginner's Guide**	$ 3.95 ea.	+	$ 1.00 ea.	_____
____	**First Book of Atari**	12.95 ea.	+	2.00 ea.	_____
____	**Inside Atari DOS**	19.95 ea.	+	2.00 ea.	_____
____	**Second Book of Atari**	12.95 ea.	+	2.00 ea.	_____
				Total	_____

All orders must be prepaid (money order, check or charge). All payments must be in US funds. (Outside the US add $4.00 shippinng and handling for air mail, $2.00 for surface mail.) NC residents add 4% sales tax.

☐ Payment Enclosed ☐ VISA ☐ MasterCard ☐ American Express
Acct. No. _____ Expires ___/___

Name _____
Address _____
City _____ State _____ Zip _____
Country _____
Allow 4-6 weeks for delivery. Foreign surface delivery 2-3 months.

ANALOG COMPUTING
THE MAGAZINE FOR ATARI COMPUTER OWNERS

1982

NUMBER 7

FEATURES/ARTICLES

REVIEWS/PROFILES

PROGRAM LISTINGS

COLUMNS

"A.N.A.L.O.G. COMPUTING is published bi-monthly for $12 per year by A.N.A.L.O.G. Magazine Corp., 565 Main Street, Cherry Valley, MA 01603. Second-class postage pending 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." Tel. (617) 892-3488.

EDITORIAL

by Lee Pappas

The presence of ATARI and their computers has been felt strongly around the country at electronics shows in the first half of 1982. Last March in San Francisco the 7th West Coast Computer Faire was attended by many tens of thousands, and the Summer Consumer Electronics Show in Chicago was even larger. ATARI, per se, made a small appearance in San Francisco, with representatives from ASAP (ATARI Software Acquisition Program) only in official attendance. The C.E.S. shows (Las Vegas in winter, Chicago in summer) are quite another matter. Here ATARI makes its corporate appearance with most of the "executive types" attending. Every C.E.S. the personal computer area of the display is allotted more space. This summer much of it went towards the new computer demo modules. A new automated dealer display concept, ERIC (Electronic Retail Information Center) is the first intelligent in-store automated demonstration unit. The system detects when a person is near, then cues up the internal laser disk system to interact with the soon-to-be ATARI computer owner. This is one of several new displays (resembling something from Moonbase Alpha) that should be in some stores by the year's end.

New products released for computers include THE BOOKKEEPER, TELELINK II, THE PERSONAL FINANCIAL MANAGEMENT SYSTEM, ATARI SPEED READING, and THE HOME FILING MANAGER. New hardware announced and shown were the numeric keypad and a new serial direct connect modem. Many new cartridges for the VCS were announced, and the new ATARI 5200 advanced video game was in full swing.

Last March staff programmer Tom Hudson and I toured the vast ATARI complex in Sunnyvale. One of the most impressive setups was the 400 & 800 construction and test facilities. With the thorough burn-in period these units endure, it's difficult to believe any bad ones get through. We also toured the coin-op construction plant, which was quite an advanced setup.

Perhaps the most fascinating part of the trip was meeting with Chris Crawford. Chris is now with the (fairly new) long range computer development program. This department's main concern is what the market and technology will be like 5 to 10 years or more down the road. In a few years we may be seeing personal computers with one megabyte RAM capabilities, 1K x 1K screen resolution, and of course superior human interfacing.

Last, but not least, I thought that I would mention our name change and the new look of the magazine. We changed the name to *A.N.A.L.O.G. Computing* for a very simple reason: when (not if) ATARI comes out with their new home computer systems, they obviously will have different model numbers than "400" and "800." Therefore, we changed the name now (to save us trouble later on), so *A.N.A.L.O.G.* can be devoted to **all** ATARI computer systems.

With our increased distribution comes an increase in our quality. A.N.A.L.O.G. now has full color throughout, as well as slick paper. (You may have noticed the latter.) All of us here at A.N.A.L.O.G. hope you, the reader, enjoy the changes and will continue to make A.N.A.L.O.G. the Number 1 magazine for the ATARI. □

In issue #6, we published two programs for cassette, **Assembler Code Subroutine Adder** and **Maniac!** Here are the modifications to run them on disk.

```
20020 DIM A$(15),B$(15):A$="D:":PRINT
"WHAT IS THE NAME OF YOUR DATA FILE";:
INPUT B$:A$(3)=B$
20040 OPEN #1,4,0,A$:GET #1,X:GET #1,X
```

•

```
0 REM CHANGE LINES 1000,5000,7480
      DELETE LINES 7490-8130
1 REM MANIAC MAKER PROGRAM
2 REM DISK VERSION
3 REM
1000 OPEN #1,8,0,"D:MANIAC.OBJ"
5000 DATA 255,255,6,64,127,79,169,60,1
41,2,211,169,119,141,231,2,2031
7480 DATA 10,0,11,0,38,64,224,2,225,2,
38,64,0,0,0,0,678
```

•

ABOUT THE COVER...

I have a confession to make: everyone here at A.N.A.L.O.G. is a science fiction fan. You may have noticed occasional sf references throughout the magazine i.e., Charlie's Non-Tutorial and Pat Kelley's article ATARI:2019. In reference to the latter, I was inspired to come up with the homage to **Blade Runner** which serves as the cover art for this issue. The original artwork was drawn in black and white, and a large photographic positive (black and clear) was shot. The positive was mounted on a sheet of glass, and color was applied by taping colored acetate sheets to the positive, cutting them to the shape of the buildings. Frosted acetate was placed behind the glass to diffuse the back-lighting (a movie light.) The positive was photographed with color print film. As with the last two covers, multiple exposures were "piled up" on the same piece of film. In this instance, it involved adding the crosstar effects and rainbow diffraction ring to the scene: a futuristic city, like **Blade Runner**, in which ATARI takes an important role.

READER COMMENT...

Dear Editor,

I noticed in the A.N.A.L.O.G. #5 the letter from Bob deWitt concerning the interference band which rolls up his screen. I have had my ATARI 800 since January 1981 and this same problem has occurred with my system. I have tried different power supplies but to no avail. Since this band doesn't always appear, I've tended to ignore the problem. Is this a major ATARI problem? Have many other owners experienced it? Do you have a means of finding out from ATARI if it is correctable?

Thank you for your attention.

Sincerely,

Ed. J. Lehmann

Washington, DC

ATARI's design philosophy is to reduce as much as possible any radio frequency interference that your computer can radiate. So-called interference bars running up your TV set are usually assoicated with power supply problems. These problems usually occur when the computer's memory capacity is increased and the power supply filtering action becomes marginal at best. This type of interference is noticed more on the 400 model since the 800 has a slightly larger power supply and is better suited to handling this problem. Luckily, the interference is only slightly annoying and does not interfere with the operation of the computer. If however, the interference is so strong that it interferes with your reading what is on the screen, producing what looks like a herring bone pattern, it is possible that one of the voltage regulator integrated circuits has gone bad and needs to be replaced. I have actually seen these IC's oscillate at a rate of several million hertz (cycles per second) when they are not suppose to oscillate at all. □ **CB**

Dear A.N.A.L.O.G.,

As the number of hardware items from secondary suppliers increases for the ATARI 800, I suggest that you consider publishing an "Alert" column. One of the first "Alerts" you should consider relates to the Axlon 128K Ramdisk Module. While this device has a number of advantages, as you pointed out in an earlier article, it also has a number of restrictions which I feel should be passed on to your readers. The Ramdisk will **not** work with the following:

1. BASIC A+
2. Monkey Wrench
3. The ATARI Word Processor

None of Axlon's ads tell you this! Having spent $500 for this module, suffice it to say that I'm more than a bit disappointed!

Paul Pettennude

Wayne, New Jersey

The Axlon Ramdisk (tm) is an excellent product when used in the way that it was designed. Unfortunately, there are several Hardware/Software items that through no fault of their own go against this design philosophy. The Ramdisk memory board contains a total of 128K of RAM. This is twice the amount of memory that the computer can normally access. To get around this limitation, Axlon used a technique known as Bank-Switching in Ramdisk's design. You might want to think of the Ramdisk board as eight memory boards of 16K bytes each. The trick here is that Ramdisk will allow only one of these eight memory banks to be accessible to the computer at any one time. Power is constantly being applied to the entire Ramdisk board to insure memory retention, but only one eighth of the total 128K of memory may be used at any one time. The board is designed to work only with the ATARI 800 computer, and the board must also be mounted in the second memory slot and have regular 16K memory boards in the first and third memory slots. To enable a different memory bank of the Ramdisk, the user simply stores the bank number which can range from 0 to 7 in any address in the range of 53184 to 53247 ($CFCO to $CFFF hex) which is an area that is presently not used by the operating system, or in the range 4032 to 4095 ($OFCO to $OFFF) which is Ram and used by several programs. To simplify its usage, Axlon has written a modified Disk Operating System that will allow the Ramdisk to simulate an 810 disk drive. This method of use seems to be the one most favored by Ramdisk owners.

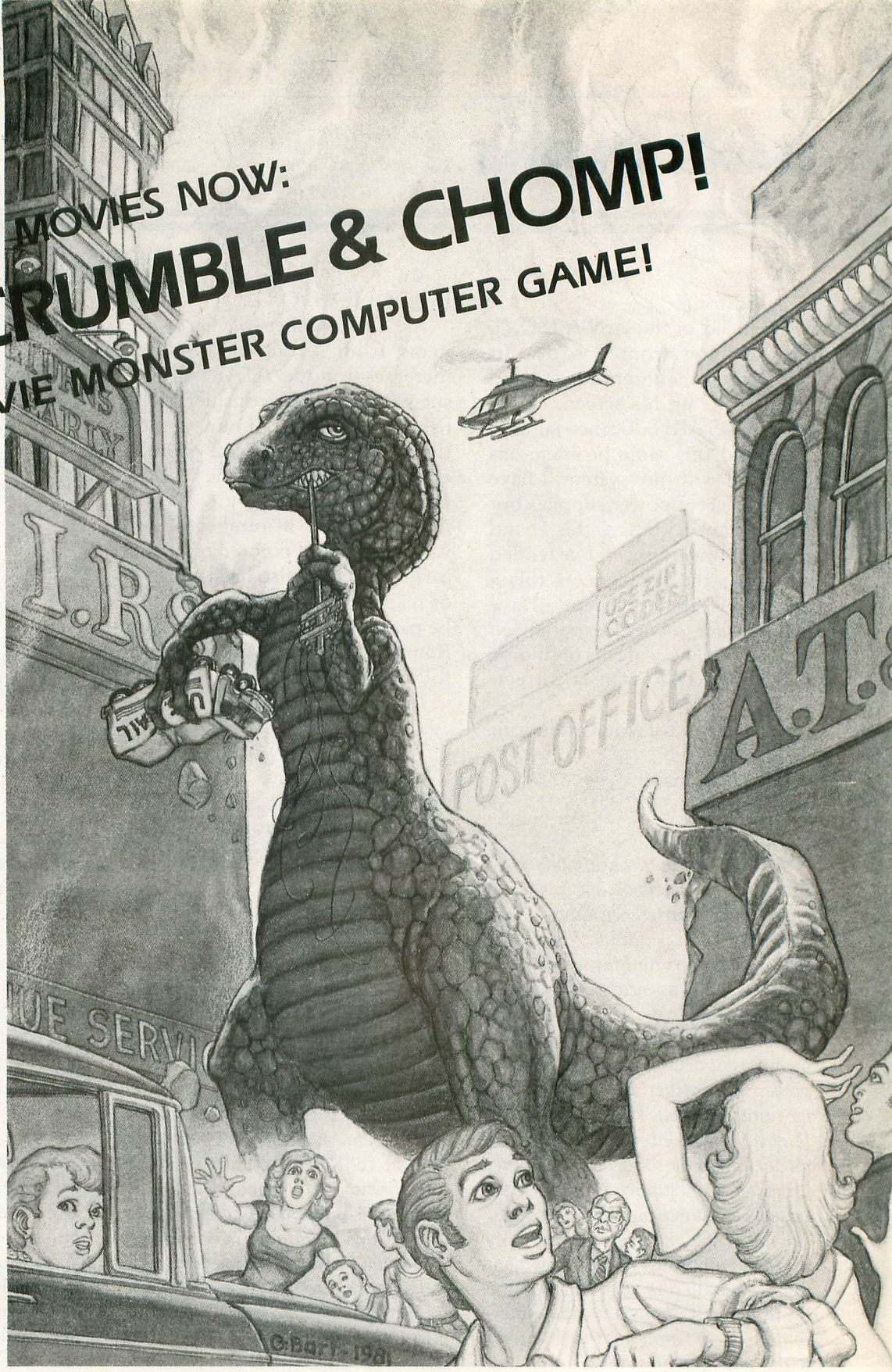
Unfortunately, Ramdisk will not work properly under certain conditions. The Axlon DOS is totally Ram resident, similar to the old ATARI DOS 1, and is quite large. Some machine language software that operates properly with DOS II will load itself over parts of the Axlon DOS, rendering it quite useless. Some software might be so large and extensive that at least parts of it will load into the Ramdisk memory. If another Ramdisk memory bank is now selected, the user's program will seem to disappear!

A final problem involves the right cartridge slot of the ATARI 800. Whenever two cartridges are inserted into the computer, as is the case when one is using Monkey Wrench by Eastern House Software, the top 16K of Ram is disabled to make room for the cartridge software. The computer uses the highest available Ram for display memory, and with the top 16K disabled, the Ramdisk now becomes the highest available Ram. The ATARI also performs another trick known as DMA (Direct Memory Access). Your computer has a special video display processor chip known as ANTIC which handles all the different character and graphics

FIRST: MONSTER MOVIES NOW:

CRUSH, CRUMBLE & CHOMP!

THE GREAT MOVIE MONSTER COMPUTER GAME!



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

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

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

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

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

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

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

COMPUTER GAMES THINKERS PLAY EDX

modes, including Player/Missile graphics. ANTIC accesses the computer's memory to produce every single line you see on your screen independently of what the 6502 microprocessor is doing. When the user starts switching memory banks with the display memory being stored in the Ramdisk, the information that the ANTIC chip uses will no longer be available. The results are unpredictable and can be quite interesting. □ **CB**

Gentlemen:

I've had my ATARI 400 Computer for four months and I am most interested in the graphics. Well, what I wanted to ask you is: **"Is there a simple way to draw a circle?!!"** I have been trying to draw a circle since I got my computer and I can't do it. If there is a simple way, would you please send it to me. Thank you.

Yours truly,
Scott C. Tucker
Las Vegas, Nevada

It is very easy to draw a circle on the ATARI computer using the trigonometric functions sine (SIN) and cosine (COS). To make things simpler, we have put together a circle drawing subroutine for you to use. The graphic mode is first set up and the color to plot is selected. Next, several variables must be assigned values. Variable XC contains the x-coordinate of the center of the circle, and variable YC contains the y-coordinate. RD is set to the radius of the circle and INC is used for the circle drawing increment in degrees. Lastly, since not all television sets are aligned equally, variable YS contains a scaling factor that is set to one initially, but should be adjusted up or down until the circles are perfectly circular. □

```
10 XC=160:YC=80
20 RD=60:INC=10:YS=0.75
30 GRAPHICS 8:COLOR 1
40 GOSUB 1000:END
1000 REM ---------------------
1010 REM CIRCLE DRAWER ROUTINE
1020 REM ---------------------
1030 REM
1040 REM XC:  x-coordinate of ce
nter
1050 REM YC:  y-coordinate of ce
nter
1060 REM RD:  circle radius
1070 REM INC: drawing increment
1-360
1080 REM YS:  y-scaling factor
1090 REM
1100 DEG :PLOT XC,YC+RD*YSCALE
1110 FOR CIRCLE=0 TO 360 STEP IN
C
1120 XCOORD=XC+SIN(CIRCLE)*RD
1130 YCOORD=YC+COS(CIRCLE)*RD*YS
1140 DRAWTO XCOORD,YCOORD
1150 NEXT CIRCLE:RETURN
```

Dear A.N.A.L.O.G.,

Congratulations on your publication excellence. In just the first five issues you have been growing and improving in both volume and content. I especially appreciate the software reviews.

One of the areas in which we ATARI owners need more information is on current software and hardware development efforts, so that we can plan our expenditures and make more informed purchasing decisions. For example, you occasionally allude to things like an expected release by Microsoft, but there is no comprehensive coverage of imminent product releases in general.

Some specific products that I would like to know about are:

1. An 80 column display package (hardware/software?)
2. Pascal
3. A single-switch system boot package (my family is intimidated by the sequence of boot operations).

Maybe the topic of coming attractions would make a good news article for your next issue. Be sure to keep up the good work.

Sincerely,
Hal Gilbert
Lakewood, CA

1) An 80-column display board is now available from Bit 3 Computer Corporation, 1890 Huron St., St. Paul, MN 55113. The board is easy to install, and plugs into the last slot of the 800 (it will not work on the 400). The board requires a monitor, preferably black & white, as the board will not work on ordinary television sets. Color monitors are not recommended, either, as their resolution is not high enough for 80-column display. The board contains its own firmware (the instructions allowing it to work) and its own display memory. Preliminary examination here at the A.N.A.L.O.G. offices have shown that the board works very well, providing a sharp, readable display. The board will be officially reviewed in a future issue of A.N.A.L.O.G. It lists for $349.

2) ATARI markets a Pascal package for the 400/800 computers which is sold through the Atari Program Exchange. It requires two disk drives and 48K, and lists for $49.95. Check the APX Summer catalog for all the details.

3) You can set up your system to boot with one switch by plugging all the computer power supplies into a multiple-plug switched outlet, available at Radio Shack. When the outlet box is turned on, the computer will start up normally, avoiding all the normal power-up procedures. □

TH

Dear Editors,

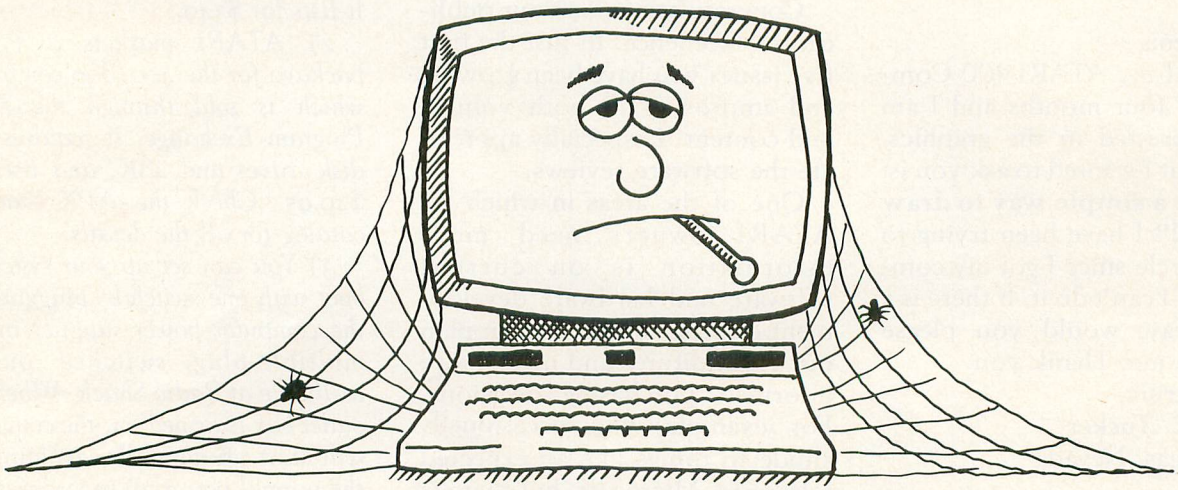
I recently decided to upgrade my ATARI system by adding a faster and higher quality printer. By doing so, I ran into a few problems concerning software compatibility for word processing. It was through one of the back issues of A.N.A.L.O.G. that I found a solution to my software problem. I appreciate your efforts and wish you great success in your undertaking. Thank you.

Sincerely,
Richard J. Boden
Tulsa, OK

SEE PAGE 26

QUICKSOFT™

CURES THE #1 TERMINAL ILLNESS:



S.D.S.

(SLOW DELIVERY SYNDROME)

QUICKSOFT'S cure doesn't come in a bottle, but it does come in handy when your computer needs software delivered promptly.

24 HOUR SHIPPING: QUICKSOFT is the software-house with a difference: **SPEED!!** When you place a phone order with **QUICKSOFT we guarantee that it will be shipped within twenty-four hours.***

WIDE SELECTION: QUICKSOFT carries thousands of software disks, tapes and cartridges for every purpose. We have business systems to cure those overflowing files, educational disks to relieve the worry of a D in chemistry, and adventure games to aid a case of the blues.

QUICKSOFT HAS THEM ALL, and right now we're having specials on these items from our "most wanted" list:

Wordrace (D)	**$19.95**
Bug Attack (T,D)	**23.95**
Canyon Climbers (T,D)	**23.95**
Protector (T,D)	**23.95**
Shooting Arcade (T,D)	**23.95**
Tumble Bugs (D)	**23.95**
Pathfinder (D)	**27.95**
Ulysses and the Golden Fleece (D)	**27.95**
Pac-man (cart)	**35.95**
Text Wizard (D)	**79.95**

Prices effective through September 4, 1982.

FREE DELIVERY: QUICKSOFT offers free delivery on *all* orders of $50.00 or more. We not only ship within 24 hours but we also pay the freight! (For orders of less than $50 add $1.50 for shipping and handling.)

CALL TOLL FREE ANYTIME. QUICKSOFT'S toll free number is available anytime of the day or night. For placing an order or for requesting our **FREE CATALOG** simply dial:

1-800-547-8009

IN OREGON CALL 1-342-1298

***GUARANTEE: QUICKSOFT** guarantees shipment within 24 hours on each phone order. If the item ordered is not readily available and cannot be shipped on time, you will receive an additional 5% OFF the regular **QUICKSOFT** price.

VISA-MASTERCHARGE and C.O.D. Welcome

QUICKSOFT™

Department 97, P.O. Box 10854 Eugene, OR 97440

Benioff At Large

by Marc Russell Benioff

Hello, and welcome to another exciting issue of **A.N.A.L.O.G.**, and another Benioff At Large.

NEW from InfoCom software!! InfoCom, Inc. is one of the largest software text adventure companies making software for microcomputers. They make software for almost all of the systems, and recently released new adventures for the ATARI! They have released three programs. Żork I and Zork II are underground adventure programs. You can type in full sentences to the computer, and then it responds with the answer. Fantastic! But, hold on. Deadline is even better. It is an extensive (and expensive) mystery adventure game. It comes in a manila file holder, allowing for about ten different items. There are notes from attorneys, police, coroners, and even five tablets found at the scene of the crime. What crime? A murder has taken place, and you must find the murderer in a huge mansion, in real time! People walk around during your investigation, phones ring, newspapers are delivered, and best of all, you can examine and interview all of the people/things. It is great. InfoCom's address is: 6 Faneuil Hall Marketplace, Boston, MA 02109.

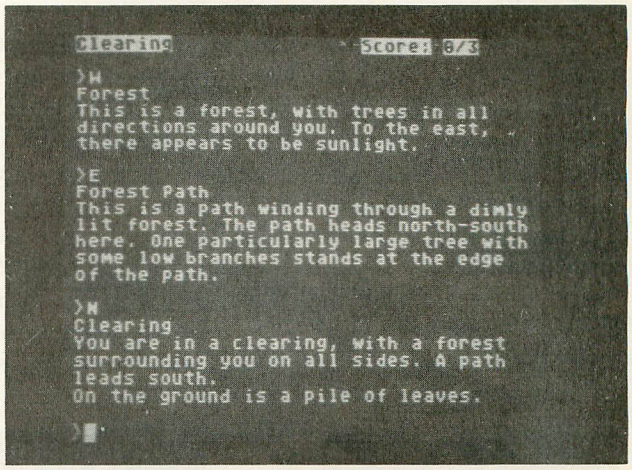
Datasoft has released three new programs: Pacific Coast Highway, Canyon Climber and Shooting Arcade. All of these are super. I don't have them yet, but have seen several demonstrations. Soon, in about four months, Datasoft will release a BASIC compiler. Anything that Datasoft has released can be considered excellent.

If you like Defender or Phoenix, you probably will like a program called Alien Swarm. It is a very addictive and original game by In-Home Software. A line moves above you; a line of birds trying to come down and steal "pods". Wow! Shoot the birds before they steal your pods! There is a two player version, and a very difficult advanced version. In-Home is at 1560 Yonge St., P.O. Box 10, Toronto, Canada, M4T1Z7.

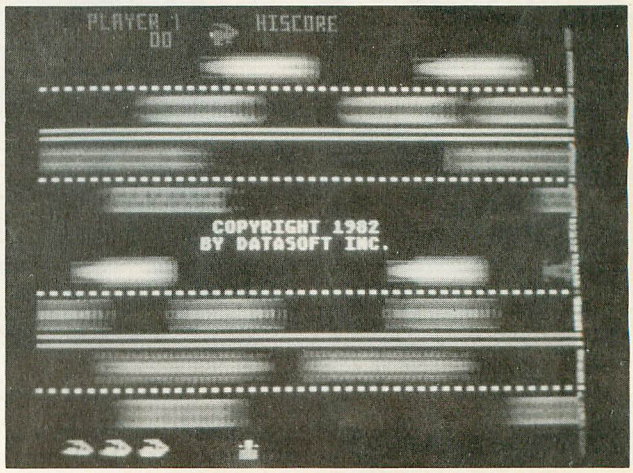
This month I bought another On-Line adventure called Ulysses and the Golden Fleece. It is a three disk adventure similiar to the Wizard and the Princess. It is very good, and if you liked Wizard, you will love Ulysses.

Gebelli told me that they will be releasing a new improved version of Andromeda, and that their game Pathfinder will be available soon. Their soon-to-be released cartridge Embargo is temporarily on hold, and is waiting for cartridge packaging.

Until next issue, this is Marc Benioff signing off! □



ZORK™



PACIFIC COAST HIGHWAY™

ULYSSES™

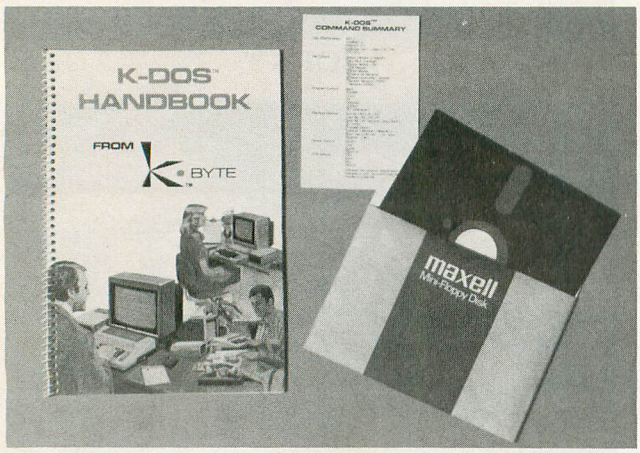
SOFTWARE REVIEW: K-DOS

K—BYTE
1705 Austin
Troy, MI 48099

by Jerry White

ATARI disk owners finally have an alternative to ATARI DOS. K-BYTE has introduced a more sophisticated Disk Operating System called K-DOS. If you can afford the $89.95 price tag, and the 5K of additional RAM K-DOS requires, you can have much greater control of your system.

I had often wished that ATARI would come out with a memory resident version of DOS 20S. Someone at K-BYTE had the same idea. No more MEM-SAVing or DUP.SYS loading over your program in memory. From BASIC or Assembler, K-DOS is always instantly at your service.



DOS 20S and K-DOS are totally compatible. DOS 20S users will quickly adapt to using K-DOS. It's a very friendly utility package with fine human engineering features such as a HELP command, and a keyboard idiot proofing routine that resets inverse video. Error messages are displayed in English as well as by number. K-DOS will stop any disk I/O at a single touch of the BREAK key. Digits are allowed as the first character of a filename, and appending files has been made more efficient.

The DC (Defined Character) command allows DOS commands to be executed when the cartridge is in control. NOTE and POINT are supported with the screen editor. Disk to cassette and cassette to disk file transfer is also supported.

The K-DOS DISKDUP provides all the options of DOS.20S and then some. The user may specify that all sectors are to be copied without regard to the disk directory. You may use or bypass the read after write verify routines. When an error is detected, you may specify "C" to continue trying, "S" to skip, "Q" to quit after the current pass, "A" to abort immediately, or "?" for help. DISKDUP will also display the number of sectors it copied, and the number of errors encountered.

K-DOS provides WARM and COLD START options, and a command called "LOM". LOM may be used to set the bottom of memory so that an area may be set aside for machine language subroutines. With no operand, LOM will display its low and high address along with the current LOM setting and the top of usable RAM.

The Machine Monitor built into this package provides valuable tools for the Assembler programmer. A LOAD map of records may be displayed as a file is loaded. The user may deal directly with memory VIA commands such as GO, PROCEED, EXAMINE, ALTER, and REGISTER. This package also comes with a system equate file, plus an equate file to entry points inside of DOS including user callable subroutines.

The user manual comes complete with Table of Contents, Detailed Explanation of Commands, Command Summary, Glossary of Terms, and an Index.

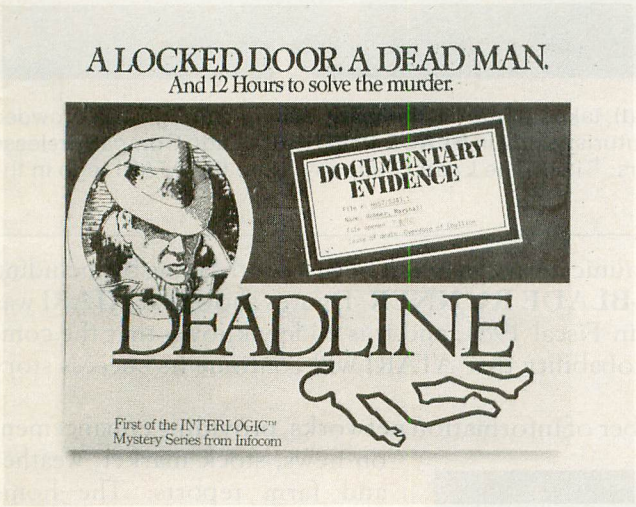
This reviewer found absolutely no bugs or problems in using this package. If the features listed above would be useful to you, K-DOS is highly recommended. □

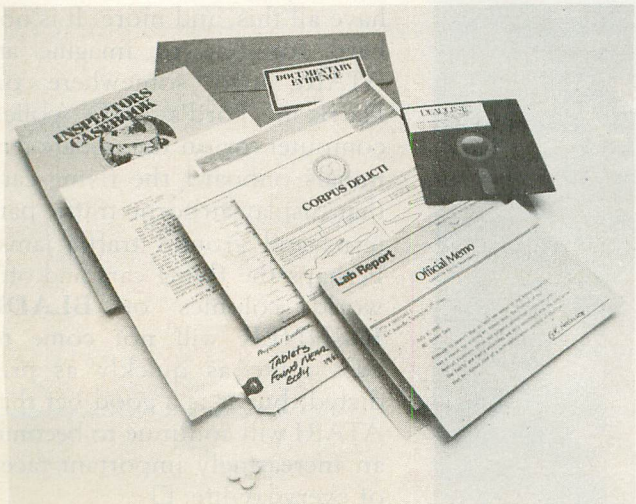
NEW PRODUCTS...

FIRST SOPHISTICATED COMPUTER MYSTERY GAME STEEPED IN TREACHERY AND SUSPENSE

CAMBRIDGE, Mass. — "A dead man, a locked door and a killer who may strike again..." are the dilemmas a computer game buff faces as the detective/player in the first sophisticated murder mystery of the computer age.

Called DEADLINE, the new mystery game was created by Infocom, Inc., developers of the best-selling ZORK I and ZORK II computer adventure games. DEADLINE will go on sale at fine computer stores and software outlets across the country beginning May 1, according to the Cambridge, Mass. firm.





The packaging for DEADLINE is as unique as the highly sophisticated prose mystery game itself. DEADLINE comes packaged in an actual dossier containing critical evidence on the crime. The dossier includes such genuine clues as a lab report, fingerprints, physical evidence collected near the victim's body, interviews with possible suspects, an 8 × 10 glossy photo of the scene of the crime, as well as a full-fledged detective's manual.

This dossier is not "window dressing," according to Joel Berez, President of Infocom, Inc. "The clues must be utilized to solve the crime. It's a whole new concept in which the player utilizes in-hand materials, as well as the software to play the game," Mr. Berez said.

In addition to tactile as well as cerebral sleuthing, DEADLINE delivers an unparalleled degree of realism because of the INTERLOGIC™ English upon which it is founded. INTERLOGIC, trademarked by Infocom, offers personal computer owners and English-based vocabulary of 600+ words, the ability to use complete English sentences, and the widest range of command options in the genre.

Only the utmost skill, logic and intuition enable the computer detective to unravel the treacherous web of motives and suspicions to solve the crime within the 12-hour "deadline." Actual playing time, according to the company, may run a good 20 hours or more.

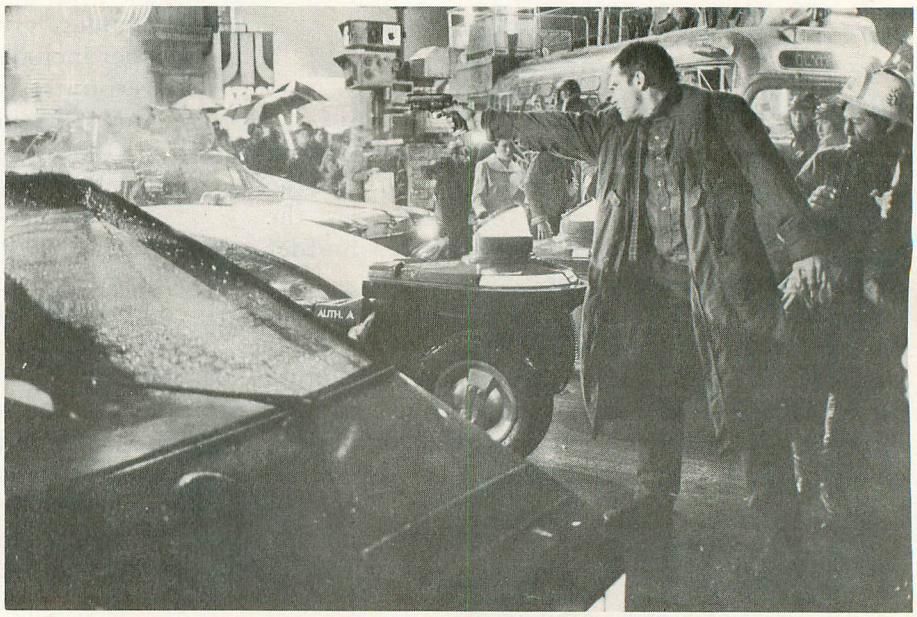
DEADLINE is available for Apple II, ATARI 400/800, IBM Personal Computer, NEC PC-8000, CP/M, and PDP-11. □

LETTER PERFECT ATARI CARTRIDGE

LJK Enterprises Incorporated has announced the release of Letter Perfect Word Processing for the ATARI Computers. The Letter Perfect Cartridge offers the same ease of operation as the disk version. Combined with the ATARI 400 or 800 computers, at least one disk drive and 16K of memory, the user complete word processor at his command. The new version has the added capability of working with any parallel printer, as well as the flexibility of multi-configuration of default formats and values. Included with the cartridge version, at no additional cost, is the Mail Merge Utility, which allows the user to do form letters, and switch between the LJK and the ATARI disk operating systems. The cartridge version and the Utility Mail Merge together sell for $249.95. The Mail Merge itself will continue to sell separately for $29.95. LJK will also continue to produce the Letter Perfect disk version for a suggested retail of $149.95. *Other ATARI* products by LJK include Data Perfect and Edit 6502. □

ATARI: 2019

by Patrick J. Kelley



Deckard (Harrison Ford) takes aim at a replicant fleeing through the crowded downtown streets of a futuristic city in **BLADE RUNNER**, a Ladd Company release through Warner Brothers. ©1982 The Ladd Company. Note the ATARI logo in the upper left of the photo.

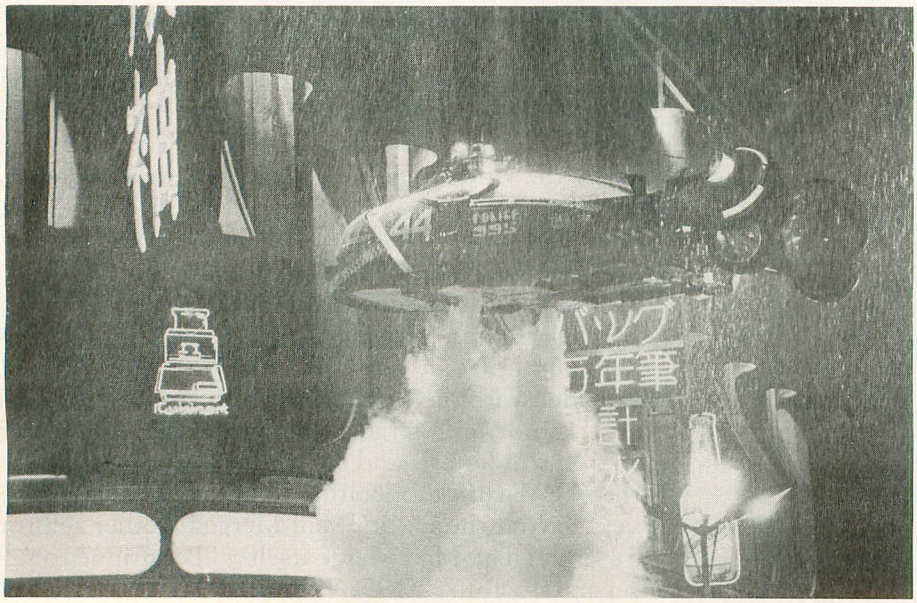
The year is 2019.

The place is Los Angeles, transformed into a bustling industrial center of over 106 million persons. The streets are clogged with perpetual traffic jams, and the harried city's police force patrols the skies in flying cars. To the average person of 20th Century North America these above scenes may sound strange and unsettling, almost alien. But a close look at the streets of 2019 shows a common denominator with those of 1982: the multi-million dollar, multi-national corporations that are a part of everyday life today still thrive in the future: TDK, RCA, Pan-Am Airlines, Coca Cola, and ATARI.

That's right, ATARI.

Warner Brothers, the filmmaking arm of Warner Communications, may not be just soothsaying by including ATARI in their filmic vision of the future, Ridley Scott's **BLADE RUNNER**. Figures show that ATARI was the most lucrative division of Warner Communications in Fiscal 1981, and it is widely known that the computer age is just dawning; it is well within the realm of probability that ATARI will continue its success story well into the next century.

Today's computer owners can tie into any one of a number of information networks, providing enhancement on news, stock market, weather and farm reports. The home computer owners of 2019 will have all this, and more. It is not hard for one to imagine an ATARI logo somewhere on Harrison Ford's 3-D police computer, or on the update terminals onboard the flying cars that display airborne traffic patterns and ground traffic jams. Perhaps the flying cars and off-world colonies of **BLADE RUNNER** will not come to pass quite as quickly as predicted, but it is a good bet that ATARI will continue to become an increasingly important facet of everyday life. □



The Spinner, a flying car used by the police in **BLADE RUNNER**, a Ladd Company release through Warner Brothers, lifts off from a downtown street. ©1982 The Ladd Company.

All photos copyright 1982 The Ladd Company. Special thanks to Christine Lyons of Jon Carter & Company, Boston, for her assistance.

Utility #2: "BUNCRUSH"

by Tony Messina

In our last episode we left our hero (Bruno Bitmangler) tearing out his hair, looking for his lost energy variable E amidst all the garbage on the T.V. screen. Meanwhile, Bruno Jr. screams ——I WANNA PLAY MISSILE COMMAND!!" and Mrs. Bitmangler shouts "Both of you get in here... DINNER is getting COLD..." If only our hero had BUNCRUSH, his problem would be solved. What's a BUNCRUSH?? It's the BASIC Unembellished No Cost Cross Reference Utility and Software Helper. If you want to get it up and running, type in Listing 2 and skip to the How To Use Buncrush section. Those of you who want to learn a little more about ATARI BASIC Token structure and how BUNCRUSH was developed should read on.

DESIGN CONSIDERATIONS

Several major considerations were involved in designing BUNCRUSH. The list I used was as follows.

1.) Build upon the concepts presented in Utility #1 — Variable Lister. (**A.N.A.L.O.G.** no. 6.)

2.) Allow use with both Cassette and Disk systems.

3.) Allow screen or printer output.

4.) Output should include the variable name, its associated line reference numbers and be neat in appearance.

5.) Make the output fast and simple.

6.) Provide flexibility for user modifications.

With these considerations in mind, I sat down and wrote BUNCRUSH. It's been rewritten three or four times. Each time it was improved and streamlined. Listing 2 is the final version. With all the above ground rules set I'll dive into the Background material, namely ATARI Token structure.

BACKGROUND

As was explained in the last utility article, variables are assigned numbers in our Token program. Names do not matter unless we want to print out a program listing. It would follow then that if we could locate the start of our Token program, scan each line for a variable # (128-255), save the line numbers that contain the variables we are looking for and print out this information, we would be all set. Of course, we would have to do this for every variable number and it could take some time. We'll worry about the time later. The first question is: where does the Tokenized version of our BASIC program begin? Glad you asked! The start location can be found at address 136,137 (Decimal) or 88,89 (Hex). This is not where the program begins but rather the pointer to where it begins. To obtain the decimal location number we would execute the following BASIC statement.

```
TOKEN=PEEK(136)+PEEK(137)*256
```

Token would be set equal to the start address of our Token program. That's nice!!! Now what?? Well... time to scan the program from start to finish for our first variable. Before we do this, I'll digress into my "Here's how a Tokenized BASIC Line is set up" tap dance routine.

I saw a hand in the back of the room... "What's this Tokenized Program you keep referring to?", I'm sorry...let me explain. When you type in a Program Line in BASIC and hit Return, several things happen, (Keep in mind this is a simple explanation.) Upon hitting Return, the BASIC interpreter (BASIC cartridge) takes each item you typed in and converts it into Tokens for its own use. Each command (GOTO, TRAP, etc.) each operator (+,-,=, etc.) and each function (STR$, SIN, COS, etc.) has its own Token associated with it. The interpreter scans, Tokenizes, places Token in program area and continues till it hits your carriage return. If everything is correct with respect to syntax, the cursor appears on the left side of the screen and you continue on with the next line. If you make a mistake, the interpreter stops the scanning and Tokenizing process and prints the line out with an error message and an inverse cursor to show you where it stopped. When you correct your mistake, the interpreter goes through the line again. This process continues until you have entered your entire program. The Token process is used to save space by converting the ASCII input to Tokens. For example, the Restore command would normally take 7 Bytes (one per letter). Through Tokenization, it only takes 1 Byte containing the number 35 Decimal. Tokens serve another important purpose. At Run-Time, the BASIC Interpreter fetches a Token. This Token is actually an index for a Jump Table. This Jump Table points to the various routines within the system. When a Token has been executed, BASIC returns, fetches the next Token and continues the process of execution. With that simple explanation out of the way, let's look at the structure of a Tokenized line of BASIC. Each line varies depending on its length and the number of multiple statements in it. Some items don't get Tokenized. ASCII strings are

an example. In a statement such as PRINT "This is a test", the PRINT statement will get Tokenized. When the interpreter encounters the quotes, it replaces them with a 15 Token (string follows Token), saves one space, then puts each letter of the string in one Byte until it hits the last quote. The Byte after the 15 then gets updated to the number of ASCII characters in the string. Similarly, numbers are put in BCD representation. BCD numbers take up 6 Bytes for the number itself. For example, with PEEK 130, the PEEK would get a Token of 70 and the "(" a Token of 58. Then a 14 would be placed next. Fourteen is the "BCD number follows Token". After the 14 would be the 6 Byte BCD representation of 130 (65 1 48 0 0 0). Don't worry, no need to memorize BCD numbers. Just remember how they appear. Anyway, our example of a simple Tokenized BASIC line follows.

BASIC LINE:

```
20 PRINT PEEK(Z)
```

TOKEN FORM (DECIMAL)

Bytes	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	20	0	10	10	32	70	58	128	44	22

```
BYTES 1 AND 2 = LINE NUMBER LSB MSB FO
RMAT.

BYTE 3 = NUMERICAL OFFSET TO THE NEXT
LINE NUMBER IN BYTES.

BYTE 4 = NUMERICAL OFFSET TO NEXT STAT
EMENT NUMBER OF BYTES.

THIS IS USED TO KEEP TRACK OF WHERE TH
E INTERPRETER IS WHEN A LINE HAS MULTI
PLE STATEMENTS.

I.E. 10 GOTO 20:GOSUB 200:PRINT X:GOTO
5

THE REMAINDER OF THE BYTES CONSISTS OF
THE TOKENIZED FORM OF OUR BASIC LINE.

BYTE 5 IS THE TOKEN OF PRINT.

BYTE 6 IS THE PEEK TOKEN.

BYTE 7 IS THE LEFT PARENTHESIS TOKEN "
(".

BYTE 8 IS THE VARIABLE NUMBER ASSIGNED
TO Z.

BYTE 9 IS THE RIGHT PARENTHESIS TOKEN
")".

BYTE 10 IS THE END OF LINE TOKEN.
```

To help you get a feel for these concepts, I've included the ATARI BASIC TOKEN TABLE 1. I've also included a short program that prints out the Tokenized version of line numbers within a program. This is Listing 1. I call it TOKLOOK. Type it in and save it using the List Command. Now load in one of your BASIC programs. When ready appears, load the TOOKLOOK program using the Enter command. When it's in, type GOTO 32500. Answer the prompt with a line number. The Tokenized version of the line will appear as well as the BASIC form. Use Table 1 and compare the Token version with the table. This little utility helped me a great deal in understanding how things get Tokenized.

BACK TO BUNCRUSH

Well, with that digression out of the way, let's look at listing 2, the Actual BUNCRUSH utility. You may notice some similarity to the Variable Lister program. I built BUNCRUSH around it. Variable names were shortened and some unnecessary items removed. There are 2 parts to BUNCRUSH. I used BASIC to handle the string manipulation tasks of finding the variable names and formatting the names/line numbers for output. The ML Routine works hand in hand with BASIC. All the ML Routine does is search the Token program for our variable numbers (we start at number 128). When it finds it, it returns the line number to BASIC. BASIC then takes the number and puts it in the string VAR$. If VAR$ exceeds the print length of 80, the program prints out that line. BASIC then jumps back into the ML Routine and the Search goes on until all variables and line references are output.

PROGRAM FLOW

```
LINE 32500 - CLEARS ALL VARIABLES AND
SETS UP THE PROGRAM PARAMETERS.

LINE 32502 - OUTPUTS HEADING CREDIT.
(GO AHEAD - PUT YOUR OWN NAME IN THERE
IF YOU WANT.)

LINE 32503 - SKIPS SOME LINES, PRINTS
OUT COLUMN HEADINGS AND READS IN THE M
L ROUTINE DATA.

LINE 32504 - GETS OUR VARIABLE NAME, O
NE CHARACTER AT A TIME.

REMEMBER FROM VARIABLE LISTER, AN INVE
RSE CHARACTER MARKS THE END OF A VARIA
BLE NAME.

IF TP>=128 THEN WE SUBTRACT 128 AND SE
T A FLAG AT 1690 FOR USE LATER ON.

I CALL IT THE VARIABLE NAME COMPLETE F
LAG.  IF TP IS NOT >=128 WE MOVE ON.
```

```
LINE 32506 - PUTS THE VARIABLE NAME IN
VAR$.  CC IS THE CHARACTER COUNT.


LINE 32508 - CHECKS OUR VARIABLE NAME
COMPLETE FLAG.  IF IT SET (=1) WE GOSU
B 32526.  IF NOT, WE FALL THROUGH.


LINE 32510 - UPDATES THE CURRENT ADDRE
SS (CA), THE CHARACTER COUNT (CC) AND
GOES BACK TO 32504 TO GET THE NEXT CHA
RACTER OF THE VARIABLE NAME.


LINE 32512 - SKIPS A FEW LINES AND PRI
NTS OUT THE VARIABLE COUNT AT THE END
OF THE PROGRAM.


LINE 32513 - ENDS THE PROGRAM.


LINE 32526 - IS A SUBROUTINE, WE JUMP
HERE FROM LINE 32508. FIRST WE CHECK
IF OUR VARIABLE NAME IS VAR$. IF YES,
POP THE STACK AND END THE PROGRAM. IF
NOT, WE DROP THROUGH.


LINE 32527 - PADS VAR$ WITH BLANKS. VA
RIABLE NAMES CAN BE UP TO 14 CHARACTER
S LONG.  IF YOU HAVE VARIABLE NAMES
LONGER THAN 15, JUST CHANGE THE 15 TO
WHATEVER YOU WANT.  I HAVEN'T HAD ANY
PROBLEMS YET.  FIFTEEN IS A SAFETY NUM
BER.


LINE 32530 - JUMPS TO OUR ML ROUTINE.
THE SOURCE LISTING IS INCLUDED AS LIST
ING NUMBER 3.  THE ML ROUTINE SEARCHES
EVERY LINE OF THE TOKEN PROGRAM LOOKIN
G FOR OUR VARIABLE NUMBER.  IT RETURNS
TO BASIC UNDER TWO CONDITIONS.


CONDITION 1: IT FINDS OUR VARIABLE NUM
BER IN A LINE.


CONDITION 2: IT ENCOUNTERS LINE NUMBER
32500, WHICH IS THE START OF THE UTILI
TY.
```

Some simplifications were necessary in writing the search program.

1.) If you find our variable, stop searching that line and return to BASIC with the line number. There is no need to search any further even if the variable appears 10 times in the line. All we care about is the line number, not how many times the variable appears therein.

2.) If we encounter a Data or REM statement skip them. There are no variables in DATA or REM statements.

3.) If we pick up a BCD Number follows Token (14) skip past it. Searching it is not healthy — we'll get an erroneous cross-reference in some instances.

4.) If we encounter a string follows Token (15) skip past the string as any inverse characters will trigger the "I found our variable" signal. Remember, we look for variable numbers from 128-255.

5.) If we hit a statement end Token (15), skip past the next Byte. It contains an offset number which can cause errors.

I won't go into too much detail of the ML Routine. It's not even very elegant, as a matter of fact. Things can be done to speed it up but as you'll see it's plenty fast enough!!! Anyway, we return to BASIC.

```
LINE 32532 - CHECKS THE CON LOCATION A
T 1680 DECIMAL.  IF SET, WE ARE CONTIN
UING - GO PROCESS THE LINE NUMBER.  IF
NOT, WE ARE DONE WITH THIS VARIABLE SO
DROP THROUGH.


LINE 32534 - ERASES THE COMMA AT THE E
ND OF THE LAST LINE NUMBER.  IF X <=16
THEN NO LINE NUMBERS WERE GENERATED FO
R THIS VARIABLE AND THEREFORE THERE AR
E NO REFERENCES TO IT.


LINE 32535 - PRINTS OUT VAR$, ZEROS OU
T THE CHARACTER COUNT, CLEARS OUT VAR$
AND NUM$ AND RETURNS TO 32510 TO GET T
HE NEXT VARIABLE FLAG.


LINE 32536 - GETS THE CURRENT LINE NUM
BER (CL) FROM LOCATIONS 1683 AND 1684
- THAT'S WHERE THE ML ROUTINE PUT THEM
.


LINE 32538 - CONVERTS THE LINE NUMBER
TO A STRING.  TI CHECKS TO SEE IF THE
LENGTH OF THIS LINE NUMBER, WHEN ADDE
D TO THE CURRENT LENGTH OF VAR$, WILL
BE GREATER THAN 80.  IF IT WOULD, VAR$
GETS PRINTED FIRST, THEN IS PADDED WIT
H 15 BLANKS.


LINE 32540 - THE LINE NUMBER GETS ADDE
D TO VAR$ AND A (COMMA SPACE) IS APPEN
DED.  HERE X IS UPDATED TO REFLECT THE
LENGTH OF VAR$.  WE THEN JUMP BACK TO
THE ML ROUTINE SO WE CAN CONTINUE ON.
```

That's the program.

HOW TO USE BUNCRUSH

Type in the program from LISTING 2. Double-check everything, especially the ML DATA, to ensure a good program. Save the program to Disk using the LIST "D:BUNCRUSH" command or to Cassette using the LIST "C:" command. To use BUNCRUSH:

1) Load in the program you want to Cross-Reference.

2) Load in BUNCRUSH using the ENTER "D:BUNCRUSH" command for Disk or the ENTER"C:" command for Cassette.

3) When READY appears, be sure your Printer and Interface are turned on.

4) Type in immediate mode GOTO 32500.

5) BUNCRUSH should now print out the title and the column header VAR LINE NUMBERS to the printer.

6) The CRT display should say *READING ML* PROGRAM. After 3-5 seconds GOOOO!! should

appear and the printer should be busy dumping out the Variable Cross-Reference.

MODIFICATIONS

The program in LISTING 2 is set up for an ATARI 825 printer with a line output of 80 columns. Modifications for other printers follow!

1) PRINTER — If you have an ATARI 40 column printer, change the >80 in LINE 32538 to 40.

2) NO PRINTER — If you don't have a printer, change all LPRINT statements to PRINT in LINES 32502, 32503, 32512, 32535 and 32538. In addition, change the 80 in LINE 32538 to 39. Everything will now be dumped out to the screen. Use the CNTRL 1 key to STOP/START the listing.

3) LINE NUMBERS — If you want to change the line numbers for BUNCRUSH in order to move it up or down, you must BEWARE of certain items. All GOTO, GOSUB references must be changed to reflect the new line numbers. The most IMPORTANT change of all is in the ML Routine itself. The ML Routine checks to see if the current line number is 32500. If you change the starting line number of BUNCRUSH, you must change the check in the ML Routine, DATA LINE 32548 item 14 is a 126 which is the MSB of the line number 32500, DATA LINE 32500 item 5 is a 244 which is the LSB of 32500. Anyway, whatever your new line number, break it down into LSB/MSB format and substitute the appropriate numbers in the above mentioned locations.

4) OTHER CHANGES — Other things which you may want to add to BUNCRUSH are ERROR CHECKING and an INPUT line which will let you title the listing in expanded print so you know what program is being Cross-Referenced. Another change which would require some work is to output an alphabetical Cross-Reference. The possibilities for additions are limited only to your imagination.

DRAWBACKS/LIMITATIONS

BUNCRUSH has some limitations which I thought should be mentioned prior to receiving a bunch of nasty phone calls and letters. Limitations on BUNCRUSH are identical to those of the Variable Lister Utility from the last issue. BUNCRUSH will not work correctly if:

1) The target program uses more than 120 variables. BUNCRUSH will abort the load procedure with an ERROR 4 (Too Many Variables).

2) Line numbers are the same as BUNCRUSH. In this case, BUNCRUSH will merge just fine with the target program but may cause problems if the target program has line numbers not contained in BUNCRUSH.

3) The target program is so large BUNCRUSH will not load due to an ERROR 2 (Insufficient Memory).

I've never had problems with item 2 or 3. I have a 48K system, however, and this may be the reason. I have encountered item 1 only once and it was with a canned program. There is a way around of all these problems; a method by which BUNCRUSH will work on ANY BASIC program. If BUNCRUSH were written entirely in Machine Language, without BASIC overhead, everything would work fine. I'll leave that as an exercise for the reader. □

PROGRAM 1

```
32500 CLR :DIM VAR$(1):ST=PEEK(136)+PE
EK(137)*256:NT=ST
32502 ? CHR$(125):? "INPUT LINE # TO E
XAMINE":INPUT A
32504 TL=PEEK(NT)+PEEK(NT+1)*256:BC=PE
EK(NT+2):IF TL=32500 OR TL>A THEN ? "L
INE NOT FOUND!!":GOTO 32512
32506 IF TL<>A THEN NT=NT+BC:GOTO 3250
4
32507 ? " LINE#","NXT LINE","NXT STMNT
"
32508 ? "LSB/MSB"," OFFSET"," OFFSET"
32509 ? " ";PEEK(NT);"/";PEEK(NT+1),"
  ";PEEK(NT+2),"    ";PEEK(NT+3)
32510 ? "TOKENIZED STATEMENT":FOR X=NT
+4 TO NT+BC-1:? PEEK(X);" ";:NEXT X:?
:? "BASIC STATEMENT":LIST A
32512 ? :? "ANOTHER LINE? Y/N ":INPUT
VAR$
32514 IF VAR$(1,1)="Y" THEN NT=ST:GOTO
 32502
32516 END
```

PROGRAM 2

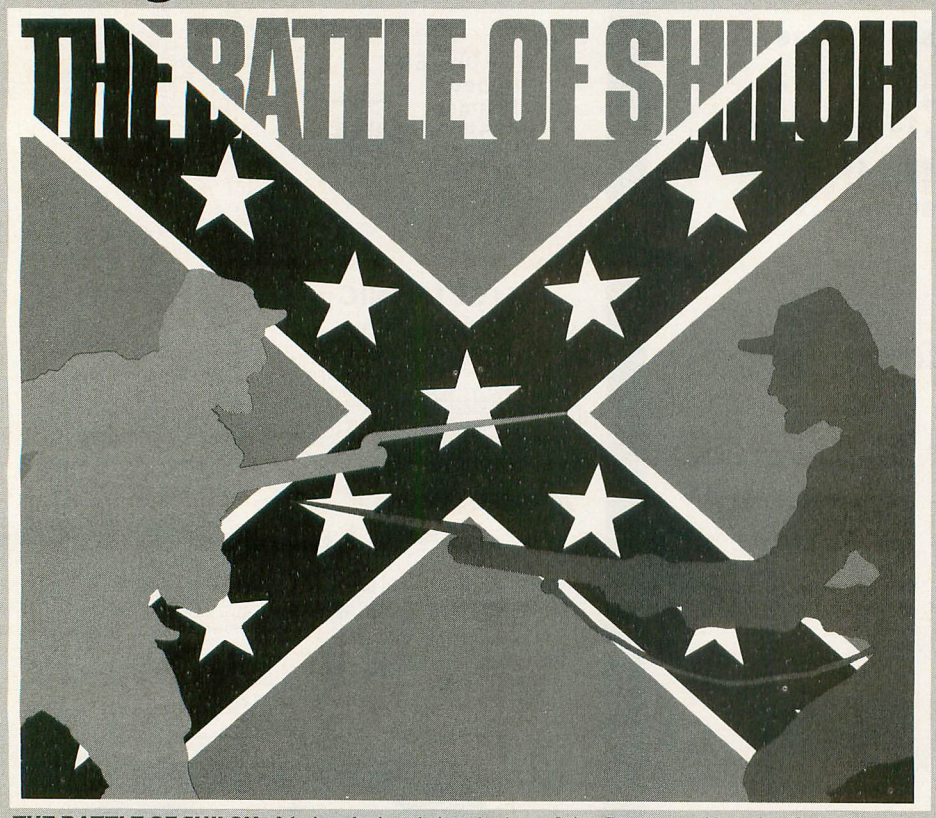
```
32500 CLR :DIM VAR$(80),NUM$(5):CA=PEE
K(130)+PEEK(131)*256:CC=1:POKE 1690,0
32502 ? "A":LPRINT "CROSS REFERENCE UT
ILITY VER. 2.5 BY TONY MESSINA NEWPORT
, RI"
32503 LPRINT :LPRINT :LPRINT "VAR
      LINE NUMBERS":LPRINT :GOSUB 325
42
32504 TP=PEEK(CA):IF TP>=128 THEN TP=T
P-128:POKE 1690,1
32506 VAR$(CC,CC)=CHR$(TP)
32508 IF PEEK(1690) THEN GOSUB 32526
32510 CA=CA+1:CC=CC+1:GOTO 32504
32512 LPRINT :LPRINT "# OF VARIABLES=
";PEEK(1681)-128
32513 END
32526 IF VAR$="VAR$" THEN POP :GOTO 32
512
32527 FOR X=CC+1 TO 15:VAR$(X,X)=" ":N
EXT X:GOTO 32530
32530 A=USR(1536)
32532 IF PEEK(1680) THEN GOTO 32536
32534 VAR$(X-1,X-1)=" ":IF X<=16 THEN
VAR$(LEN(VAR$)+1)="NO REFERENCES"
32535 LPRINT VAR$:LPRINT :CC=0:POKE 16
90,0:VAR$="":NUM$="":RETURN
32536 CL=PEEK(1683)+PEEK(1684)*256
32538 NUM$=STR$(CL):IF LEN(VAR$)+LEN(N
UM$)+2>80 THEN LPRINT VAR$:VAR$="
      "
32540 VAR$(LEN(VAR$)+1)=NUM$:VAR$(LEN(
VAR$)+1)=", ":X=LEN(VAR$):GOTO 32530
32542 RESTORE 32546:? "READING ML PROG
RAM":FOR X=1536 TO 1684:READ TP:POKE X
,TP:NEXT X
32544 ? CHR$(125):? "GOOOOO!":RETURN
32546 DATA 169,0,205,144,6,208,8,165,1
36,133,205,165,137,133,206,160
32548 DATA 0,177,205,141,147,6,200,177
,205,141,148,6,201,126,208,7
32550 DATA 173,147,6,201,244,240,82,20
0,177,205,141,143,6,160,4,177
32552 DATA 205,201,20,208,5,200,200,76
,101,6,205,145,6,240,49,201,0,240,39,2
01,1,240,35,201,14
```

```
32554 DATA 208,8,152,24,105,7,168,76,1
01,6,200,201,15,208,13,140
32556 DATA 146,6,177,205,238,146,6,24,
109,146,6,168,204,143,6,144
32558 DATA 197,32,130,6,76,15,6,141,14
4,6,32,130,6,76,128,6
32560 DATA 136,238,145,6,140,144,6,104
,96,165,205,24,109,143,6,133
32562 DATA 205,144,2,230,206,96,0,0,12
8,0,0,0
```

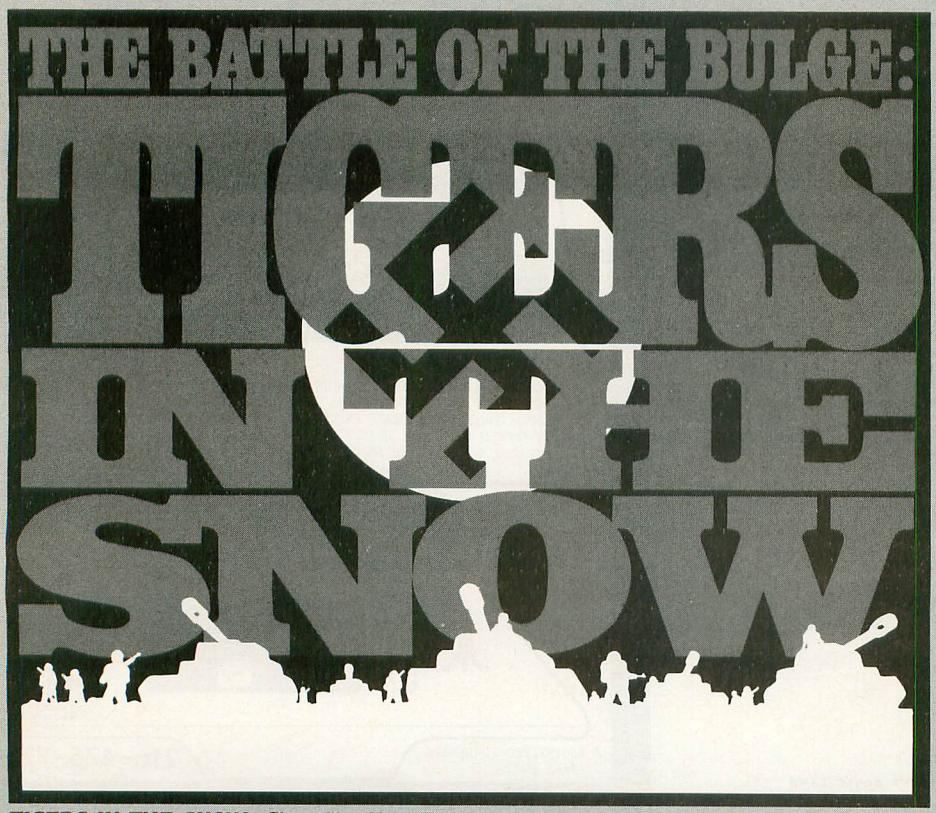
ASSEMBLY LANGUAGE LISTING

```
0100 ;*********************************
0102 ;* ML SEARCH AIDE FOR ATARI 400*
0104 ;* /800 WRITTEN BY TONY MESSINA*
0106 ;* 48 DUDLEY AVE NEWPORT,RI    *
0108 ;* 02840  VERSION 2.0          *
0110 ;*********************************
0112 ;*********************************
0114 ;*  EQUATES FOR PROGRAM FOLLOW   *
0116 ;*********************************
0118 DAT        = 1         ; DATA TOKEN
0120 REM        = 0         ; REM TOKEN
0122 BCD        = 14        ; BCD # TOKEN
0124 STR        = 15        ; STRING TOKEN
0126 STMT       = 20        ; STMNT END TOKEN
0128 TOKPTR     = $0088     ; PTR TO TOKEN PROGRAM
0130 PG0        = $00CD     ; OUR PAGE 0 LOCATION
0132 ;*********************************
0134 ;* THIS PROGRAM DOES A SEARCH TO*
0136 ;* AIDE THE BASIC CROSS-REF UTIL*
0138 ;* BASIC WAS TO SLOW SO THIS ML *
0140 ;* ROUTINE WAS WRITTEN TO SPEED *
0142 ;* THINGS UP A BIT..            *
0144 ;*********************************
0146             *=$0600
0148 MESS        LDA #0          ; LOAD A ZERO
0150             CMP CON         ; CK WITH CON
0152             BNE CONTIN      ; SKIP INIT IF NOT ZERO
0154 INIT        LDA TOKPTR      ; GET LSB OF TOKEN POINTER
0156             STA PG0         ; STORE IT
0158             LDA TOKPTR+1    ; GET MSB
0160             STA PG0+1       ; STORE IT
0162 CONTIN      LDY #0          ; START Y AT ZERO
0164             LDA (PG0),Y     ; GET LSB OF CURRENT LINE NUMBER
0166             STA LINNUM      ; SAVE IT FOR BASIC
0168             INY             ; INCREMENT OFFSET IN Y
0170             LDA (PG0),Y     ; GET MSB OF LINE NUMBER
0172             STA LINNUM+1    ; SAVE IT FOR BASIC
0174 ;
0176 ;**** CHECK THIS LINNUM FOR 32500 $7EF4 ****
0178 ;
0180             CMP #$7E        ; IS IT = TO MSB
0182             BNE NOER        ; IF NO THEN START
0184             LDA LINNUM      ; YES SO CK LSB
0186             CMP #$F4        ;
0188             BEQ DONE        ; IF EQUAL WE ARE DONE WITH THIS VAR
0190 NOER        INY             ; INC PTR TO NXT LOCATION
0192             LDA (PG0),Y     ; GET BASIC LINE BYTE CNT
0194             STA COUNT       ; SAVE IT FOR FUTURE CK'S
0196             LDY #4          ; GET NEW OFFSET
0198 START       LDA (PG0),Y     ; GET A BYTE INDIRECTLY
0200             CMP #STMT       ; CK FOR A STATEMENT END TOKEN
0202             BNE TARGCK      ; NO.. CK FOR OUR TARGET NUMBER
0204             INY             ; IT WAS A STMT..INC PAST
0206             INY             ; THE NEXT BYTE
0208             JMP CKCNT       ; SEE IF WE ARE DONE
0210 TARGCK      CMP TGT         ; CK AGAINST THE TGT NUMBER
0212             BEQ PROCIT      ; IF= GO PROCESS THIS LINE NUMBER
0214             CMP #REM        ; NO CK IF A REM
0216             BEQ SKIPIT      ; IF REM SKIP THIS LINE
0218             CMP #DAT        ; NOT REM CK DATA STATEMENT
0220             BEQ SKIPIT      ; IF DATA SKIP THIS LINE
0222             CMP #BCD        ; NOT DATA CK FOR BCD NUMBER
0224             BNE STRCK       ; IF NOT A BCD NUMBER CK FOR A STRING
0226             TYA             ; ITS A BCD NUMBER SO PUT OFFSET IN A REG
0228             CLC             ; CLEAR CARRY FOR ADD
0230             ADC #7          ; ADD 7 TO OFFSET TO SKIP THE BCD NUMBER
0232             TAY             ; NEW OFFSET TO Y REG
0234             JMP CKCNT       ; GO CK TO SEE IF WERE DONE WITH LINE
0236 STRCK       INY             ; INC Y REG TO NEXT LOCATION
0238             CMP #STR        ; CK IF ITS A STRING BYTE FOLLOWS TOKEN
0240             BNE CKCNT       ; IF NOT GO CK THE COUNT
0242             STY YSAVE       ; ITS A STRING..SAVE Y REGISTER
0244             LDA (PG0),Y     ; LOAD IN THE # OF STRING BYTES
0246             INC YSAVE       ; INCREMENT PAST THE LAST STRING BYTE
0248             CLC             ; CLEAR CARRY FOR ADD
0250             ADC YSAVE       ; ADD STRING COUNT TO OLD Y
0252             TAY             ; PUT IT BACK IN Y LIKE A GOOD DOOBIE
0254 CKCNT       CPY COUNT       ; CK IF WE ARE INTO THE NEXT BASIC LINE
0256             BCC START       ; IF NO, GO GET THIS BYTE
0258 SKIPIT      JSR TOKEUP      ; YES SO UPDATE THE TOKEN PTR ON PG 0
0260             JMP CONTIN      ; CONTINUE TO LOOK
0262 PROCIT      STA CON         ; STORE VAR NUM IN CON (MAKE NON-ZERO)
0264             JSR TOKEUP      ; NOW UPDATE THE PG0 PTR
0266             JMP BASIC       ; EXIT TO BASIC
0268 DONE        DEY             ; DEC Y TO 0
0270             INC TGT         ; UPDATE TGT NUMBER
0272             STY CON         ; ZERO OUT CON SO BASIC KNOWS WHAT TO DO
0274 BASIC       PLA             ; GET NASTYNESS OFF OF THE STACK
0276             RTS             ; GO TO BASIC
0278 ;*********************************
0280 ;*    * SUBROUTINE TOKE UP *     *
0282 ;*      *********************     *
0284 ;* THIS SUBROUTINE UPDATES THE *
0286 ;* PG0 PTR TO THE TOKEN PROGRAM*
0288 ;* THE OLD PTR IS LOADED AND   *
0290 ;* THEN THE BYTE COUNT IS ADDED*
0292 ;* CARRY SET UPDATES PG0+1 TOO *
0294 ;*********************************
0296 TOKEUP      LDA PG0         ; GET LSB OF POINTER
0298             CLC             ; CLEAR THAT CARRY
0300             ADC COUNT       ; ADD THE COUNT TO NEXT BASIC LIN#
0302             STA PG0         ; PUT IT BACK
0304             BCC OUT         ; IF CARRY CLEAR GET OUT
0306             INC PG0+1       ; OOPS CARRY SET SO INC MSB OF POINTER
0308 OUT         RTS             ; GO BACK TO CALLING ROUTINE
0310 ;
0312 ;
0314 ;******** LOCAL VARIABLES FOLLOW ********
0316 COUNT       .BYTE 0         ; BASIC LINE COUNT (IN BYTES)
0318 CON         .BYTE 0         ; CONTINUE FLAG
0320 TGT         .BYTE 128       ; VARIABLE # START AT 128
0322 YSAVE       .BYTE 0         ; Y REGISTER SAVE AREA
0324 LINNUM      .BYTE 0,0       ; BASIC LINE NUMBER LSB,MSB
0326             .END
```

A dynamic duo from SSI for the ATARI®!



THE BATTLE OF SHILOH: A brigade-level simulation of the first grand battle of the Civil War, pitting the Confederate Army against Grant's troops and Union gunboats.



TIGERS IN THE SNOW: Ghostlike Nazi Tiger tanks and infantry sweep across the dark, frozen forests of the Ardennes against a surprised U.S. force in this division/regiment-level simulation of Hitler's last desperate attack.

Atari owners, rejoice!

First, SSI brought you THE SHATTERED ALLIANCE™ and you made it the popular fantasy wargame it is today.

Now, we are proud to offer two more games for your pleasure: THE BATTLE OF SHILOH™ and TIGERS IN THE SNOW™.

In producing these games, we've taken full advantage of the Atari 400/800's brilliant graphics capability and combined it with our extensive wargame-design experience.

For example, both games have colorful map displays that can be scrolled effortlessly so you can keep track of all the action on the vast battlefields. We've also employed special rules and innovative gaming systems to enhance the sophistication, realism and playability of these simulations.

And the result of all this hard work? Two strategy games that will truly challenge and excite your game-loving mind!

At $39.95 each (available in either cassette or disc version*), they are waiting for you at your local computer/game store today!

VISA and M/C holders can order 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, 465 Fairchild Drive, Suite 108, Mountain View, California 94043.

All our games carry a 14-day money-back guarantee.

*TIGERS: 40K mini floppy disc or 40K cassette for the Atari 400/800 with Atari BASIC cartridge. SHILOH: 40K mini floppy disc or 32K cassette for same. When ordering, please specify disc or cassette version.

SSI

STRATEGIC SIMULATIONS INC

Restore Your Mental Health

by Mike Sueirro

How many times have you attempted to convert and enter a general BASIC program, from a magazine perhaps, only to find it wouldn't fit into your 16K ATARI? Or perhaps you're two-thirds of the way through typing in your masterpiece only to receive an "ERROR-2", out of memory flag? If you have, then you know true frustration. You are also beginning to understand one of the reasons why commercial BASIC games, etc., that fit into 16K on an Apple or TRS-80 require 24K on an ATARI. For those of us who chose the ATARI 400 as the best cost/performance path into personal computing, ATARI BASIC's voracious appetite for memory is disappointing. If these problems are plaguing you, friend, then what you need is mucho dinero to expand your machine's memory OR "Dr. Mike's Crash Course in Memory Improvement". If you're as frugal (broke!) as I am, then read on.

Falling into Disarray

Single and multi-dimensional arrays are powerful tools that the programmer can use to achieve ends that are difficult to satisfy using other means. But the price we pay in memory overhead to maintain an array is enormous. Since ATARI BASIC uses floating point representation for all numbers, each element in a numerical array is also a floating point number and requires 7 bytes of memory plus some stray bytes to intialize the array and store the array name. That means that a simple 8 by 8 array (a checkerboard perhaps) will require over 450 bytes. Nearly half a K of memory for a simple checkerboard array!

A more dramatic demonstration is to consider a typical method used in adventure games to determine which room a player moves into when the player selects a direction. The common practice is to use an array whose length is the number of rooms in the game and whose width is the number of possible directions one can move. Let's choose a small adventure game with only 40 rooms and only six possible directions (N,S,E,W,U, and D). This is a typical size for an adventure in a 16K machine. In ATARI BASIC this single array will consume over 1680 bytes of free memory! When you consider that ATARI BASIC in a 16K machine only has 13.3K of free memory to begin with (the remainder is controlled by the BASIC ROM), this single 40 by 6 array will occupy nearly 13% of available RAM.

This is a high price to pay for the convenience of using an array. Note that I said convenience of using an array, not the need for such use. Arrays are powerful tools for storing variables, but quite often a large array is used merely to store a table of constant values. When memory is scarce or where a program will not fit into available RAM, we simply cannot afford such luxuries. So what do we do? One simple step that can be taken is to ensure that arrays are dimensioned as tightly as possible. It's important to realize that the amount of data stored in an array is irrelevant in terms of memory dedicated to the array. BASIC will reserve an amount of memory for each array when the array is dimensioned (with a DIM statement). If you don't fill an array then the remaining elements will be filled with zeroes and they will not be available for other uses. So make your DIM statements as tight as you can. Now if you carry this process through to its logical conclusion, then the best way to reduce memory consumption is to keep reducing the size of your arrays until they disappear altogether. But if we don't use an array, what do we use?

RESTORE Your Memory

In the vast majority of programs that employ arrays to hold what is essentially a table of constant values, the data to fill the array are programmed in a string of DATA statements which we READ into the array at program execution. Do you see that if the array data is first loaded in as part of the program (in DATA statements) and then transferred to an array, that we are storing that information twice? If we never loaded that array at all the data would still be present in memory as a character string following the DATA statements. If we can find a method of accessing the list of DATA statements selectively then we can simply use these statements as our array of constants, thereby saving 7 bytes per array element.

We achieve this feat of prestidigitation through the judicious use of the RESTORE statement. A quick check of the ATARI BASIC Reference Manual tells us that the RESTORE statement can be followed by a line number. In normal operation, ATARI BASIC maintains a pointer "that keeps track of the DATA statement item to be read next." If we use RESTORE without a line number, then this pointer is returned to the line number of the first DATA statement in the program. However, if the RESTORE is followed by a line number or variable then the next DATA line to be read is that one specified in the RESTORE statement. Here is the tool that allows us to represent our array of constants exclusively through the use of DATA

statements.

To demonstrate how we apply this tool and its effectiveness, I'll describe the technique I used recently to convert an adventure program I found in a magazine into a format that would fit into my 16K ATARI 400. The program was coded relatively simply so there was little problem in converting those few statements that would not fit ATARI BASIC syntax. With that done, I began typing. The program fit within available memory so I sat back and prepared for an enjoyable romp. I typed in the RUN command and was immediately greeted with an ERROR-2, out of memory flag. An examination revealed that the game employed two large arrays, the first of which was a 40 by 8 numeric array of possible moves out of each location, just like the array described above. This array occupied over 2200 bytes, so when the BASIC ROM attempted to reserve RAM at execution time it found insufficient space available and errored out. Enter the RESTORE statement.

The first step in eliminating this array is to ensure that line numbers for the associated DATA statements follow a predictable pattern; let's say each line number is an increment of 10. Next I created a subroutine that selected the line number to RESTORE. This was done by using a base number that was 10 units lower than the line number of the first DATA statement in the list. I added onto this the appropriate room number of the current location multiplied by 10. So, if we are in room Z and the line number of the first DATA statement in the possible movement array is xxxx, our statement becomes:

```
DATA Line Number = (xxxx-10)+Z*10
```

If we RESTORE and READ the DATA statement at this line number we will have the possible moves available from room Z. Now, if we wish to extract the move for a particular direction we assign a number, 1 through 6, for each potential movement. Then, when a move command is entered, we go to the DATA statement for our current position, READ to the appropriate element (first through sixth) on the statement line and return that value. We do this by setting the ending value of a FOR-NEXT loop to the number we assigned for that direction. We read each element in the DATA statement until the loop ends. The last element read is the one we want and we return this value to the program. Presto chango, we've just saved 2200 bytes of memory!

Getting Strung Out

What's that you say? The array I've converted was 8 elements wide and I've only used 6 elements? You're right, of course, and the explanation of what happened to those elusive two elements takes us into the second mind (memory?) expanding the use of the RESTORE statement. I'll let you in on a little secret. My tale of woe regarding the discovery that this adventure wouldn't fit in memory wasn't quite accurate. You see, the second large array I mentioned earlier turned out to be a 40 by 42 string array. We all know (or soon will) that ATARI BASIC doesn't accept string arrays. So I confess that I had to overcome this problem before I even began to type. How do you represent string arrays in ATARI BASIC? Elementary, my dear Watson. Elementary, that is if the data in the strings are constant. You do the same thing that you did in the numeric array above.

In this example the strings to be manipulated were the textual descriptions of each location. The two missing elements in the movement array were merely the pointers to the descriptive text for that location. Two elements were used because some locations required more than one DATA statement. The missing elements pointed to the first and last DATA statements that describe that particular room. When the player moves to a new location, the program automatically goes to the movement table entry for the new room, pulls out two pointers (elements 7 and 8) then restores and displays the appropriate text included between the pointers. Note that the pointers could be actual line numbers, but these are frequently four digit numbers. Space is conserved by using only a two digit number and reconstructing the line number in a manner similar to the numeric array above.

Digital Amnesia Cured!

And there you have it, Lesson One in Memory Improvement. Naturally, this method can't be applied in all cases, but where it is applicable the memory savings can be tremendous. Ignoring the problem of string arrays in the ATARI, I estimate that the adventure game I've described would have required 20K of memory as originally coded. By using RESTORE, READ and DATA statements, the program is running nicely, thank you, in my 16K ATARI 400 and with 600 bytes to spare. So get busy, eliminate those arrays, and let's start seeing some good games that don't "require 24K" to run. □

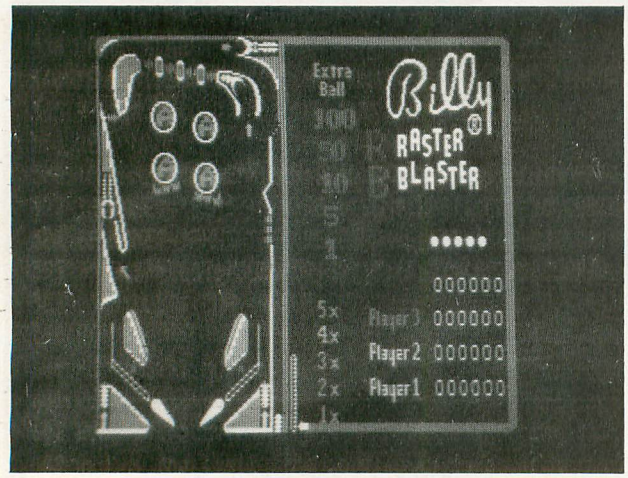
The Program Doctors

SOFTWARE REVIEW:
RASTER BLASTER

Budge Co.
428 Pala Ave.
Piedmont, CA 94611
$29.95 32K Disk

Review by The Program Doctors

When I was a child, I always wanted a pinball machine of my very own. Many years later when the opportunity arose, I bought one (Cowpoke by Gottlieb). Now ATARI owners can also be like "Tommy" and become pinball wizards in their own homes. Raster Blaster, the much celebrated Apple program, has been released by BudgeCo. on 32K disk. It is complete with bumpers, rollovers, a bonus multiplier, adjustable ball launcher, pop-up targets, independently controlled flippers and ball-catching claws.



In the center of the play field are four bumpers, each with a big, red "A." There are three right side pop-up targets, which, when completed, light the Blaster "B." Making the four top rollovers lights the Raster "R," and if you can light up both the "R" and the "B" on one ball, you will earn yourself a nice, big bonus. Upon completion of the center six pop-up targets, two of the ball-catching claws are enabled. When a ball is caught in a claw, it is held, and you receive another ball. If you can knock-out the six center targets a second time, and have captured at least one ball, the third claw is enabled. When all three claws have a ball, the balls are simultaneously released, and you begin to play three balls at once. Getting the rollovers also increases the bonus multiplier, which is re-set at the end of each ball. Make the rollovers six times on one ball, and you will be rewarded with a free ball.

The flippers are independently controlled by the use of two joystick buttons, similar to the flipper buttons on a real pinball machine. The right joystick button, besides controlling the right flipper, also manipulates the top rollover lights. Flipper response is excellent, and control is outstanding. You can transfer the ball from one flipper to the other and even catch it to aim for dead shots (you will have to find them — they are there!). Up to four people can play, and there are two levels of difficulty.

Raster Blaster is a game that has received mixed comments in its ATARI translation. As a pinball game simulation Raster Blaster is excellent. But as a pinball simulation for the ATARI, Raster Blaster could have been better. It is quite obvious that the author, Bill Budge, has placed a lot of attention to detail while writing this program. The animation of the spinner, rubber bumpers, claw springs and ball-saving shields is outstanding, and the physics of the game are extremely good. But he has not taken full advantage of the ATARI's superior sound and graphic capabilities for the overall picture.

Bill Budge's Raster Blaster has been an extremely successful Apple program, and there is an old saying, "Don't mess with success." This is one of the first translations for the ATARI, and it is the Program Doctor's fear that other translators may not take into account the ATARI's differences and its unique game qualities.

Despite this, Raster Blaster is a highly realistic and entertaining game, which should give you many hours of enjoyment, and it receives a hearty recommendation from the Program Doctor. □

Raster Blaster Rating Scale
(1 Unsatisfactory - 10 Ideal)

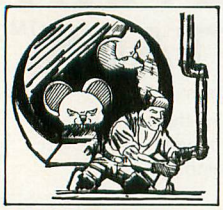
Concept	8
Originality	7
Challenge	9
Skill	7
Graphics	9
Sound	8

```
5 REM *** CUBE 'FILL' GRAPHICS DEMO ***
10 GRAPHICS 7+16:SETCOLOR 0,0,12:SETCOLOR 1,
3,2:SETCOLOR 2,7,4:HUE=1
20 FOR CUBE=1 TO 15:RAND=RND(0):MAX=15+15*RA
ND:MIN=5+5*RAND:PX=2+RND(0)*116:PY=2+RND(0)*
52:REM *** 15 CUBES ***
30 X1=PX+MIN:X2=PX+MAX:X3=X2+MIN:Y1=PY+MIN:Y
2=PY+MAX:Y3=Y2+MIN:REM *** CUBE COORDS ***
35 COLOR 0:PLOT X3+1,Y3+1:DRAWTO X3+1,Y1:DRA
WTO X2,PY-1:DRAWTO PX-1,PY-1
36 DRAWTO PX-1,Y2:DRAWTO X1,Y3+1:DRAWTO X3+1
,Y3+1
40 FOR N=1 TO MIN:PLOT PX+N,PY+N:DRAWTO X2+N
,PY+N:PLOT PX+N,PY+N:DRAWTO PX+N,Y2+N:NEXT N
50 FOR N=1 TO MAX+1:PLOT X1,Y1+N:DRAWTO X3,Y
1+N:NEXT N:REM *** 35-50 ERASE CUBE AREA ***
55 REM *** NOW DRAW & FILL CUBE SIDES ***
60 COLOR HUE:PLOT X3,Y3:DRAWTO X3,Y1:DRAWTO
X1,Y1:POSITION X1,Y3:POKE 765,HUE:XIO 18,#6,
0,0,"S:":GOSUB 200
70 COLOR HUE:PLOT X3,Y1:DRAWTO X2,PY:DRAWTO
PX,PY:POSITION X1,Y1:POKE 765,HUE:XIO 18,#6,
0,0,"S:":GOSUB 200
80 COLOR HUE:PLOT X1,Y3:DRAWTO X1,Y1:DRAWTO
PX,PY:POSITION PX,Y2:POKE 765,HUE:XIO 18,#6,
0,0,"S:"
90 PLOT X1,Y3:DRAWTO X1,Y2:DRAWTO PX,Y2:POSI
TION X1,Y3:XIO 18,#6,0,0,"S:":NEXT CUBE
100 REM *** ROTATE COLORS A WHILE ***
110 FOR ROT=1 TO 500:T=PEEK(708):POKE 708,PE
EK(709):POKE 709,PEEK(710):POKE 710,T
120 FOR DELAY=1 TO 20:NEXT DELAY:NEXT ROT:RU
N :REM *** DO IT AGAIN! ***
200 HUE=HUE+1:IF HUE=4 THEN HUE=1
210 RETURN
```

To Order Call Toll-Free
800-424-2738

ATARI®

the PROGRAM STORE™



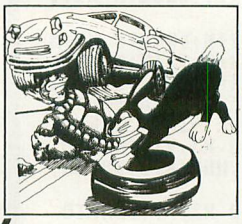
MOUSKATTACK

From On-Line by John Harris

You are Larry Bain, ace plumber, trying desperately to finish installing a pipe through the dangerous corridors of Rat Alley, before the legendary man-eating rodents can stop you! The last plumber that entered disappeared without a trace, so be careful!

48K Disk, ~~$34.95~~ *Save 20%!*

Now Thru Sep. 1 You Pay Just $27.96



PACIFIC COAST HIGHWAY

From DataSoft by Ron Rosen

Fast-action game with good graphics. You must try to cross the Highway without getting hit by the rush hour traffic. Once across the highway, you must leap from boats to rafts to cross the Pacific Ocean. One added challenge: you're a *tortoise*!

16K Tape or Disk, ~~$29.95~~ *Save 20%!*

Now Thru Sep. 1 You Pay Just $23.96

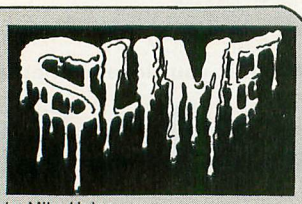
CANYON CLIMBER

From DataSoft by Tim Ferris

Great outdoor adventure! You're at the bottom of the Grand Canyon, trying to scale the world's toughest wall. If that's not enough of a problem, there are three challenges to face along the way: angry sheep, threatening Indians and attacking birds. Action-packed game you'll really enjoy.

16K Tape or Disk, ~~$29.95~~ *Save 20%!*

Now Thru Sep. 1 You Pay Just $23.96

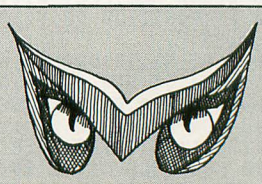


SLIME

From Synapse by Mike Hales

Large drops of "Slime" fall from the sky onto your ships. If ***one drop*** hits your ship, it will sink. Use the triangular diverters to aim Slime into buckets on the sides of the screen. If Slime falls into the ocean, the level of the ocean rises. When the ocean reaches the top of the screen, the game is over, and your score is calculated. One-player game with excellent graphics.

16K Tape or Disk, $29.95



WARLOCK'S REVENGE

From Synergistic Software by Butch Greathouse

You're leading a group of adventurers into the evil warlock's underground empire to recover the treasures he has stolen. Your party includes a cleric, a thief, a gladiator, a strongman, a magician, a wizard and an elf. You must select the best character to lead each part of the adventure, as each has special powers and abilities you may need to gain your objective. Good luck, and watch out for the Torkies!

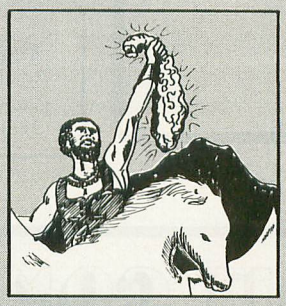
40K Disk, $34.95

NAUTILUS

From Synapse by Mike Potter

Two-player interaction combat game: the underwater cities and their destroyers against the helicopters and submarines. The helicopters bomb the ships; the subs fire torpedoes at the cities and surface to fire at the destroyers. The destroyers fire at the helicopters and drop depth charges on the subs. They can also lower men into the ocean to rebuild cities hit by torpedoes. Top and bottom halves of screen scroll separately for maximum field of play. Great graphics.

16K Tape or Disk, $29.95



ULYSSES AND THE GOLDEN FLEECE

From On-Line by Bob Davis & Ken Williams

The setting is ancient Greece, and you are Ulysses. The king has assigned you the task of retrieving the Golden Fleece, kept from men for many years under the protection of the gods. On your journey, you will encounter the song of the Sirens; one-eyed Cyclops; and the famous winged horse, Pegasus. Ulysses is known as the most daring and skilled seaman of all time. . . it's up to you to prove it!

40K Disk, $39.95

CONTROLLER

From Microcomputer Games (Avalon Hill)

You are the air traffic controller who must make quick decisions. . . or you might cause a mid-air crash and lose hundreds of lives! All actions are in REAL TIME—aircraft will maneuver, use fuel and respond to your commands as a real plane would—the computer won't wait for you. Aircraft are plotted on realistic "radar scope." Heading, velocity and altitude displayed on separate chart. Select up to 8 planes; choose from 3 types with 3 levels of difficulty.

40K Disk, $29.95

DOMINOES AND CRIBBAGE

From Creative Computing Software

Challenge your computer to a game of draw Dominoes. You can repeat or alternate draws, but you have a pretty tough opponent who knows all the rules! Or, can you beat the computer in Cribbage, and be the first to peg twice around the board? An exciting challenge with graphic display of board and cards.

16K Tape or Disk, $19.95

CROSSFIRE

The aliens have taken over the city and everyone left peacibly—except you. You are surrounded by the laser firing hordes. In this exciting arcade game your chances of survival are slim, but you can make it—if you don't get caught in the crossfire!

48K Disk or 16K Cassette, $29.95

CAVERNS OF MARS

By Greg Christensen from APEX

An award-winning arcade game that will challenge your skill for many, many plays. You command the fighter Helicon VII on a mission to invade the Martian high command. The headquarters are located deep in the caverns, so you must descend through radar, rockets, laser gates, and space mines to reach your destination. Once there, you activate the fusion bomb and try to get out before it detonates. Good luck! Joystick required.

16K Tape, $29.95 16K Disk, $37.95

Visit our other stores: Seven Corners Center, Falls Church, VA • W. Bell Plaza, 6600 Security Blvd., Baltimore, MD
Coming Soon to Columbus, Ohio

VISA **For Information Call 202-363-9797** **To Order Call Toll-Free 800-424-2738** master charge

the PROGRAM STORE™
4200 Wisconsin Avenue, NW, Dept. U208 Box 9609
Washington, D.C. 20016

MAIL ORDERS: Send check or M.O. for total purchase price, plus $1.00 postage & handling. D.C., MD. & VA.: add sales tax. Charge cards: include all embossed information on the card.

Dino Battle

by Art V. Cestaro III

Dino Battle is a game of primordial confrontation, a game of fierce battle between two players. See if you can defeat a dinosaur!

Your goal is to bit your opponent's dinosaur on the back of the neck. By moving your joystick and pressing the firing button, you can move your dinosaur and open and close his mouth. You may make a number of attempts before you succeed. Try to bite your opponent as many times as you can before the time is up.

Your score is displayed on the corresponding side of the starting position of each dinosaur at the start of the game. You receive one point each time you bite the other dinosaur. □

LINE	EXPLANATION
Y, Y1	Vertical position of dinosaur 1
3	Sets GRAPHIC mode and colors
12	Sets time and score
13-16	Draws landscape
80-81	Prints text
100-200	Main loop: checks joysticks and triggers and increments time
300-315	Moves dinosaur figures on screen
1000-1015	Turns dinosaur number 1 around
1100-1115	Turns dinosaur number 2 around
3500-3595	Makes dinosaur 1 open his mouth and try to bite the other one
3600-3710	Makes dinosaur 2 do the same thing
3800	Prints both players' scores
3900-3905	Plots cacti
3910-3930	Plots rocks
4000-4021	Plots dinosaur 1, fall routine
4500-4531	Plots dinosaur 2, fall routine
4600	Erases the dinosaur
4800-4810	Moves dinosaur away from defeated opponent
5000-5990	Plots title
7000-7110	Opening display
8000-8220	End of game
10000-10035	Sets up player/missile graphics
10040-11000	Reads shape data and stores it in the proper arrays
12000-12900	Data for shapes

NAME	VARIABLE
Time	Time in seconds of the game
Score 1	Players' scores
Score 2	
TT	Timing variable
X	Horizontal position of dinosaur 1
X2	Horizontal position of dinosaur 2
DR1	Direction dinosaur 1 is facing
DR2	Direction dinosaur 2 is facing
DF1	Area in memory where player data is poked.
DB1	
DF2	
DB2	
Y, Y1	Vertical position of Dinosaur 1
Y2, Y3	Vertical position of dinosaur 2
RT, RET, RT1	Return Flags
G, H, DD	Dummy variables
C, Z,	
I	Top of RAM: used for setting up player/missile area

ARRAYS

TF1, TF2	Flying dinosaur's front
TB1, TB2	Flying dinosaur's back
D1NF1	
D1NF2	Dinosaur front and back views
D1NB1	
D1NB2	
DHR	Dinosaur's head and mouth open

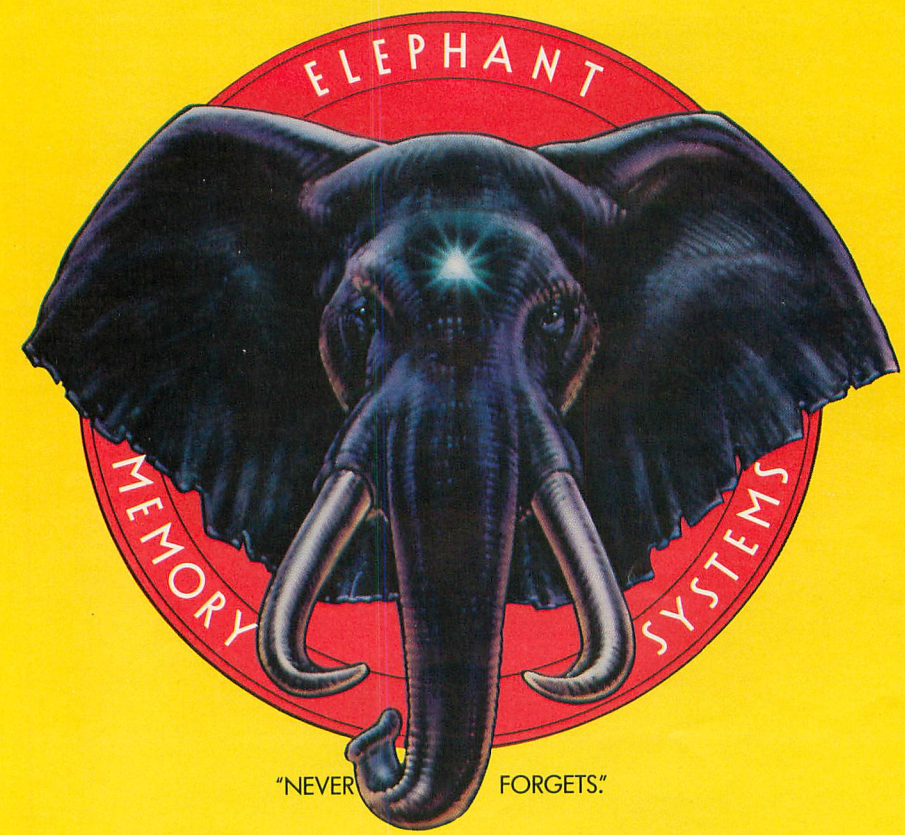
Each dinosaur is made up of two players, positioned next to each other so they make up one dinosaur shape.

```
0 REM  DINO BATTLE  REV 1.0
1 REM By Art V Cestaro III 10/13/81
3 GRAPHICS 7:CLR :POKE 752,1:POKE 712,
197:POKE 710,24:POKE 708,99:POKE 709,1
95
6 GOSUB 3930
12 TIME=59:TIM=0:SCORE1=0:SCORE2=0:COL
OR 1
13 Y=INT(RND(0)*35+10):D=1:FOR X=0 TO
150 STEP 2:Y1=INT(15*RND(0)+Y-5*D):PLO
T X,47:DRAWTO X,Y:PLOT X+1,47
14 DRAWTO X+1,(Y+Y1)/2:Y=Y1:IF Y>40 TH
EN Y=Y-10:D=2
15 IF Y<20 THEN Y=Y+10:D=1
16 NEXT X
17 GOSUB 3900:GOSUB 3910
30 GOSUB 7000
75 RET=0:GOSUB 10000:GOSUB 1000:GOSUB
1100
80 POKE 752,1:POKE 656,0:POKE 657,3:?
"  SCORE  ":POKE 656,0:POKE 657,28:? "
 SCORE  "
81 POKE 656,0:POKE 657,12:? "|↓+|↓+|";
POKE 656,0:POKE 657,27:? "|↓+|↓+|":POK
E 656,0:POKE 657,16:? "  TIME  "
82 GOSUB 3800
100 TT=TT+0.2:IF TT>1 THEN TT=0:TIME=T
IME-1:IF TIME<1 THEN TIME=59:TIM=TIM-1
104 IF STICK(0)=7 THEN X=X+2:IF DR1=1
THEN GOSUB 1000
105 IF STRIG(0)=0 THEN RT=0:GOSUB 3500
110 IF STICK(1)=7 THEN X2=X2+2:IF DR2=
2 THEN GOSUB 1110
```

```
111 IF X<55 THEN X=55
112 IF X>195 THEN X=195
115 ON DR1 GOSUB 300,305
120 IF STICK(1)=11 THEN X2=X2-2:IF DR2
=1 THEN GOSUB 1100
130 IF STICK(0)=11 THEN X=X-2:IF DR1=2
 THEN GOSUB 1010
132 IF STRIG(1)=0 THEN RT1=0:GOSUB 360
0
133 IF X2<55 THEN X2=55
134 IF X2>195 THEN X2=195
135 ON DR2 GOSUB 310,315
169 IF TIM<1 AND TIME<2 THEN POKE 656,
2:POKE 657,18:? "0:00":GOTO 8000
172 IF TIME<10 THEN POKE 656,2:POKE 65
7,18:? TIM;":0";TIME:GOTO 180
175 POKE 656,2:POKE 657,18:? TIM;":";T
IME;" "
180 POKE 77,0
200 GOTO 100
300 POKE 53248,X:POKE 53249,X-8:RETURN
305 POKE 53249,X-8:POKE 53248,X:RETURN
310 POKE 53250,X2-8:POKE 53251,X2:RETU
RN
315 POKE 53251,X2:POKE 53250,X2-8:RETU
RN
1000 DR1=2:FOR G=1 TO 4:POKE DB1+G,0:N
EXT G:Y=65:Y1=69:DF1=Y+J:DB1=Y1+J1:FOR
 G=1 TO 18:POKE DB1+G,DINB1(G)
1005 POKE DF1+G,DINF1(G):NEXT G:FOR G=
19 TO 22:POKE DF1+G,DINF1(G):NEXT G:RE
TURN
1010 DR1=1:FOR G=1 TO 4:POKE DF1+G,0:N
EXT G:Y=69:Y1=65:DF1=Y+J:DB1=Y1+J1:FOR
 G=1 TO 18:POKE DF1+G,DINB2(G)
1015 POKE DB1+G,DINF2(G):NEXT G:FOR G=
19 TO 22:POKE DB1+G,DINF2(G):NEXT G:RE
TURN
1100 DR2=2:FOR G=1 TO 4:POKE DB2+G,0:N
EXT G:Y2=65:Y3=69:DF2=Y2+J2:DB2=Y3+J3:
FOR G=1 TO 18:POKE DF2+G,DINF2(G)
1105 POKE DB2+G,DINB2(G):NEXT G:FOR G=
19 TO 22:POKE DF2+G,DINF2(G):NEXT G:RE
TURN
1110 DR2=1:FOR G=1 TO 4:POKE DF2+G,0:N
EXT G:Y2=69:Y3=65:DF2=Y2+J2:DB2=Y3+J3:
FOR G=1 TO 18:POKE DF2+G,DINB1(G)
1115 POKE DB2+G,DINF1(G):NEXT G:FOR G=
19 TO 22:POKE DB2+G,DINF1(G):NEXT G:RE
TURN
3500 ON DR1 GOTO 3510,3520
3510 BB=DB1:GG=3590:GOTO 3550
3520 BB=DF1:GG=3580
3550 GOSUB GG
3555 FOR G=50 TO 100:SOUND 0,G,10,15:S
OUND 0,100-(G-50),10,15:NEXT G:SOUND 0
,0,0,0
3560 ON DR1 GOTO 3563,3565
3563 POKE BB,0:FOR G=1 TO 6:POKE BB+G,
DINF2(G):NEXT G:GOTO 3591
3565 POKE BB,0:FOR G=1 TO 6:POKE BB+G,
DINF1(G):NEXT G:GOTO 3591
3570 RETURN
3580 POKE BB+6,224:FOR G=0 TO 5:POKE B
B+G,DHR(G+1):NEXT G:RETURN
3590 POKE BB+6,7:FOR G=0 TO 5:POKE BB+
G,DHL(G+1):NEXT G:RETURN
3591 IF RT=1 THEN RETURN
3592 IF DR1=2 AND DR2=1 AND PEEK(53260
)=12 THEN GOSUB 4500
3593 IF DR1=1 AND DR2=2 AND PEEK(53261
)=12 THEN GOSUB 4500
3595 POKE 53278,0:RETURN
3600 ON DR2 GOTO 3610,3620
3610 BB=DB2:GG=3580:GOTO 3650
3620 BB=DF2:GG=3590
3650 GOSUB GG
3655 FOR G=50 TO 100:SOUND 0,G,10,15:S
OUND 0,100-(G-50),12,10:NEXT G:SOUND 0
,0,0,0
3660 ON DR2 GOTO 3663,3665
3663 POKE BB,0:FOR G=1 TO 6:POKE BB+G,
DINF1(G):NEXT G:GOTO 3700
3665 POKE BB,0:FOR G=1 TO 6:POKE BB+G,
DINF2(G):NEXT G
3700 IF RT1=1 THEN RETURN
3701 IF DR2=2 AND DR1=1 AND PEEK(53262
)=3 THEN GOSUB 4000
3705 IF DR2=1 AND DR1=2 AND PEEK(53263
)=3 THEN GOSUB 4000
3710 POKE 53278,0:RETURN
3800 POKE 656,2:POKE 657,6:? SCORE1;"
 ":POKE 656,2:POKE 657,31:? SCORE2;"
":RETURN
3900 COLOR 2:FOR J=1 TO 4:H=INT(45+RND
(0)*10):G=RND(0)*145+10:GOSUB 3903:NEX
T J:RETURN
3901 DRAWTO G+2,H+5:DRAWTO G+2,H+3:RET
URN
3903 PLOT G,H:DRAWTO G,H+9:PLOT G,H+4:
DRAWTO G-2,H+4:DRAWTO G-2,H+1:PLOT G,H
+5
3905 DRAWTO G+2,H+5:DRAWTO G+2,H+3:RET
URN
3910 COLOR 1:FOR J=1 TO 3:H=48+RND(0)*
10:G=RND(0)*145+10:GOSUB 3913:NEXT J:R
ETURN
3911 DRAWTO G+5,H+5:DRAWTO G+3,H+9:RET
URN
3913 PLOT G,H:DRAWTO G-5,H+5:DRAWTO G+
3,H+9:DRAWTO G,H:DRAWTO G+4,H+1
3915 DRAWTO G+5,H+5:DRAWTO G+3,H+9:RET
URN
3930 COLOR 3:FOR G=79 TO 47 STEP -1:PL
OT 0,G:DRAWTO 159,G:NEXT G:RETURN
4000 BB1=DF1:BB2=DB1:GOSUB 4600
4003 Y=75:Y1=74:DF1=Y+J:DB1=Y1+J1
4005 ON DR1 GOSUB 4010,4020
4006 RT1=1:GOSUB 3600:GOTO 4810
4010 FOR G=1 TO 9:POKE DB1+G,DLF(G):PO
KE DF1+G,DLB(G):SOUND 0,120,8,15-G:NEX
T G
4011 POKE DF1+10,DLB(10):POKE DF1+11,D
LB(11):FOR G=1 TO 6:SOUND 0,120,8,15-G
:FOR HH=1 TO 10:NEXT HH:NEXT G:RETURN
4020 FOR G=1 TO 9:POKE DB1+G,DRB(G):PO
KE DF1+G,DRF(G):SOUND 0,120,8,15-G:NEX
T G
4021 POKE DB1+10,DRB(10):POKE DB1+11,D
RB(11):FOR G=1 TO 6:SOUND 0,120,8,15-G
:FOR HH=1 TO 10:NEXT HH:NEXT G:RETURN
4500 BB1=DF2:BB2=DB2:GOSUB 4600
4503 Y2=74:Y3=75:DF2=Y2+J2:DB2=Y3+J3
4505 ON DR2 GOSUB 4520,4530
4510 RT=1:GOSUB 3500:GOTO 4800
4520 FOR G=1 TO 9:POKE DF2+G,DRB(G):PO
KE DB2+G,DRF(G):SOUND 0,110,8,15-G:NEX
T G
4521 POKE DF2+10,DRB(10):POKE DF2+11,D
RB(11):FOR G=1 TO 6:SOUND 0,110,8,15-G
:FOR HH=1 TO 10:NEXT HH:NEXT G:RETURN
4530 FOR G=1 TO 9:POKE DB2+G,DLB(G):PO
KE DF2+G,DLF(G):SOUND 0,110,8,15-G:NEX
T G
4531 POKE DB2+10,DLB(10):POKE DB2+11,D
LB(11):FOR G=1 TO 6:SOUND 0,110,8,15-G
:FOR HH=1 TO 10:NEXT HH:NEXT G:RETURN
4600 FOR G=1 TO 22:POKE BB1+G,0:POKE B
B2+G,0:NEXT G:RETURN
4800 X=INT(RND(0)*145+50):ON DR1 GOSUB
 300,305:GOSUB 1100:SCORE1=SCORE1+10:G
OSUB 3800:RETURN
4810 X2=INT(RND(0)*145+50):ON DR2 GOSU
B 310,315:GOSUB 1000:SCORE2=SCORE2+10:
GOSUB 3800:RETURN
5000 COLOR 1:PLOT 26,5:DRAWTO 26,15:PL
OT 26,5:DRAWTO 31,6:DRAWTO 31,14:DRAWT
O 26,15:GOTO 5990
5100 PLOT 36,5:DRAWTO 36,15:PLOT 35,5:
PLOT 37,5:PLOT 35,15:PLOT 37,15:GOTO 5
990
5200 PLOT 42,15:DRAWTO 42,5:DRAWTO 46,
15:DRAWTO 46,5:GOTO 5990
5300 PLOT 50,5:DRAWTO 50,15:DRAWTO 55,
15:DRAWTO 55,5:DRAWTO 50,5:GOTO 5990
5400 PLOT 66,5:DRAWTO 66,15:DRAWTO 71,
15:DRAWTO 71,5:DRAWTO 66,5:PLOT 66,10:
DRAWTO 71,10:GOTO 5990
5500 PLOT 76,5:DRAWTO 81,5:DRAWTO 81,1
5:PLOT 76,5:DRAWTO 76,15:PLOT 76,10:DR
AWTO 81,10:GOTO 5990
5600 PLOT 85,5:DRAWTO 91,5:PLOT 88,5:D
RAWTO 88,15:GOTO 5990
5700 PLOT 95,5:DRAWTO 101,5:PLOT 98,5:
```

```
DRAWTO 98,15:GOTO 5990
5800 PLOT 106,5:DRAWTO 106,15:DRAWTO 1
11,15:GOTO 5990
5900 PLOT 116,5:DRAWTO 116,15:DRAWTO 1
21,15:PLOT 116,10:DRAWTO 121,10:PLOT 1
16,5:DRAWTO 121,5:GOTO 5990
5990 RETURN
7000 DD=17:DIM TF1(DD),TB1(DD),TF2(DD)
,TB2(DD)
7005 FOR G=1 TO DD:TF1(G)=0:TF2(G)=0:T
B1(G)=0:TB2(G)=0:NEXT G
7010 FOR G=1 TO 14:READ C:TF1(G)=C:NEX
T G:FOR G=1 TO 13:READ C:TB1(G)=C:NEXT
 G
7020 FOR G=1 TO 13:READ C:TB2(G)=C:NEX
T G:FOR G=1 TO 14:READ C:TF2(G)=C:NEXT
 G
7025 RET=0:GOSUB 10000
7030 POKE 704,49:POKE 705,49:FOR G=5 T
O 19:POKE DF1+G,TF1(G-4):NEXT G:FOR G=
1 TO 13:POKE DB1+G,TB1(G):NEXT G
7040 FOR X=220 TO 35 STEP -1:POKE 5324
8,X-7:POKE 53249,X:SOUND 0,X,10,6:FOR
H=1 TO 3:NEXT H:NEXT X
7045 FOR G=1 TO 18:POKE DF1+G,0:POKE D
B1+G,0:NEXT G
7051 POKE 704,49:POKE 705,49:FOR G=1 T
O 13:POKE DB1+G,TB2(G):NEXT G:FOR G=4
TO 18:POKE DF1+G,TF2(G-3):NEXT G
7060 FOR X=30 TO 210:POKE 53249,X:POKE
 53248,X+7:SOUND 0,X,10,6
7062 IF X=75 THEN GOSUB 5000
7063 IF X=85 THEN GOSUB 5100
7064 IF X=91 THEN GOSUB 5200
7065 IF X=103 THEN GOSUB 5300
7066 IF X=119 THEN GOSUB 5400
7067 IF X=130 THEN GOSUB 5500
7068 IF X=138 THEN GOSUB 5600
7069 IF X=144 THEN GOSUB 5700
7070 IF X=155 THEN GOSUB 5800
7071 IF X=165 THEN GOSUB 5900
7075 FOR H=1 TO 4:NEXT H:NEXT X
7080 ? "K            BY Art.V.Cestaro III
":SOUND 0,90,12,11:SOUND 1,91,12,12:GO
SUB 10040
7085 COLOR 0:FOR G=5 TO 10:PLOT 25,G:D
RAWTO 125,G:PLOT 25,15-(G-5):DRAWTO 12
5,15-(G-5):NEXT G
7086 SOUND 0,80,12,12:SOUND 1,81,12,14
7090 ? "K              PRESS START  "
7091 FOR G=1 TO 20:GOSUB 7098:NEXT G
7092 ? "K              PRESS START  "
7093 FOR G=1 TO 20:GOSUB 7098:NEXT G:G
OTO 7090
7098 IF PEEK(53279)=6 THEN POP :GOTO 7
100
7099 RETURN
7100 ? "K          OH OH ... FOOTSTEPS
"
7101 FOR G=1 TO 2:FOR H=15 TO 0 STEP -
1:SOUND 0,120,8,H
7105 SOUND 1,122,8,H:FOR J=1 TO 8:NEXT
 J:NEXT H:FOR F=1 TO 60:NEXT F
7107 FOR H=15 TO 0 STEP -1:SOUND 0,110
,8,H:SOUND 1,112,8,H:FOR J=1 TO 8:NEXT
 J:NEXT H:FOR F=1 TO 60:NEXT F:NEXT G
7110 ? "K":RETURN
8000 FOR G=1 TO 10:POKE 656,0:POKE 657
,15:? " GAME OVER ":SOUND 0,150,10,14:
FOR Z=1 TO 15:NEXT Z
8005 POKE 656,0:POKE 657,15:? "  GAME O
VER ":SOUND 0,100,10,14:FOR H=1 TO 15:
NEXT H:NEXT G
8009 SOUND 0,0,0,0:POKE 656,0:POKE 657
,13:? "  PRESS START "
8010 IF SCORE1>SCORE2 THEN 8020
8013 IF SCORE2>SCORE1 THEN 8030
8015 IF SCORE1=SCORE2 THEN 8040
8020 POKE 656,0:POKE 657,3:? "  SCORE
 ":FOR H=1 TO 15:GOSUB 8100:NEXT H
8021 POKE 656,0:POKE 657,3:? "  SCORE
 ":FOR H=1 TO 15:GOSUB 8100:NEXT H:GOT
O 8020
8030 POKE 656,0:POKE 657,28:? "  SCORE
 ":FOR H=1 TO 15:GOSUB 8100:NEXT H
8035 POKE 656,0:POKE 657,28:? "  SCORE
 ":FOR H=1 TO 15:GOSUB 8100:NEXT H:GO
TO 8030
8040 POKE 656,0:POKE 657,3:? "  SCORE
 ":POKE 656,0:POKE 657,28:? "  SCORE
":FOR H=1 TO 15:GOSUB 8100
8041 NEXT H
8045 POKE 656,0:POKE 657,3:? "  SCORE
 ":POKE 656,0:POKE 657,28:? "  SCORE
":FOR H=1 TO 15:GOSUB 8100
8046 NEXT H:GOTO 8040
8100 IF PEEK(53279)=6 THEN POP :GOTO 8
200
8101 RETURN
8200 SCORE1=0:SCORE2=0:TIM=0:TIME=59
8210 FOR G=250 TO 0 STEP -3:SOUND 0,G+
5,10,15:SOUND 1,G+4,10,14:SOUND 2,G+3,
10,13
8215 SOUND 3,G+2,10,12:POKE 712,RND(0)
*255:NEXT G:FOR G=0 TO 3:SOUND G,0,0,0
:POKE 53248+G,35:NEXT G
8217 POKE 712,197:GOSUB 3930:GOSUB 390
0:GOSUB 3910
8220 POKE 656,0:POKE 657,13:? "
      ":POKE 712,197:GOTO 75
10000 POKE 559,46:I=PEEK(106)-24:POKE
54279,I:POKE 53277,3:POKE 623,1
10010 J=I*256+512:J1=I*256+640:J2=I*25
6+768:J3=I*256+896
10015 FOR G=J TO J3+128:POKE G,0:NEXT
G
10020 POKE 704,165:POKE 705,165:POKE 7
06,220:POKE 707,220
10025 X=100:Y=17:Y1=16
10030 DF1=Y+J:DB1=Y1+J1:DF2=Y+J2:DB2=Y
+J3
10035 IF RET=0 THEN RETURN
10040 DD=22:DIM DINF1(DD),DINF2(DD),DI
NB1(DD),DINB2(DD),DHR(6),DHL(6)
10043 CC=11:DIM DRF(CC),DRB(CC),DLF(CC
),DLB(CC)
10045 FOR G=1 TO DD:DINF1(G)=0:DINF2(G
)=0:DINB1(G)=0:DINB2(G)=0:NEXT G
10050 RESTORE 12500:FOR G=1 TO 18:READ
 C:DINB1(G)=C:NEXT G:FOR G=1 TO 22:REA
D C:DINF1(G)=C:NEXT G
10060 RESTORE 12600:FOR G=1 TO 22:READ
 C:DINF2(G)=C:NEXT G:FOR G=1 TO 18:REA
D C:DINB2(G)=C:NEXT G
10065 RESTORE 12700:FOR G=1 TO 6:READ
C:DHR(G)=C:NEXT G:FOR G=1 TO 6:READ C:
DHL(G)=C:NEXT G
10066 RESTORE 12800:FOR G=1 TO 9:READ
C:DRF(G)=C:NEXT G:FOR G=1 TO CC:READ C
:DRB(G)=C:NEXT G
10068 RESTORE 12900:FOR G=1 TO 9:READ
C:DLF(G)=C:NEXT G:FOR G=1 TO CC:READ C
:DLB(G)=C:NEXT G
10070 X=100:X1=92:X2=150:X3=158:Y=64:Y
1=68:Y2=64:Y3=68
10071 DF1=Y+J:DB1=Y1+J1:DF2=Y2+J2:DB2=
Y3+J3
11000 RETURN
12000 DATA 1,6,28,47,63,87,175,31,28,5
6,56,24,12,4
12005 DATA 3,6,28,24,56,48,112,112,243
,252,248,249,158
12010 DATA 192,96,56,24,28,12,14,14,20
7,63,31,153,112
12020 DATA 128,96,56,244,252,234,245,2
48,56,28,28,24,48,32
12500 DATA 1,1,1,7,7,3,7,15,7,143,199,
143,198,158,188,240,224,64
12510 DATA 28,52,62,122,245,242,224,25
1,245,240,240,224,192,128,192,224,240,
112,48,96,96,248
12600 DATA 56,44,124,94,175,79,7,223,1
75,15,15,7,3,1,3,7,15,14,12,6,6,31
12610 DATA 128,128,128,224,224,192,224
,240,224,241,227,241,99,121,61,15,7,2
12700 DATA 76,104,208,254,240,224,50,2
2,11,127,15,7
12800 DATA 12,15,229,55,255,254,252,24
8,112,128,128,240,252,31,15,79,39,19,3
0,12
12900 DATA 48,240,160,231,252,127,63,3
1,14,1,1,143,63,248,240,242,228,200,12
0,48
```

REMEMBER:



MORE THAN JUST ANOTHER PRETTY FACE.

Says who? Says ANSI.

Specifically, subcommittee X3B8 of the American National Standards Institute (ANSI) says so. The fact is all Elephant™ floppies meet or exceed the specs required to meet or exceed all their standards.

But just who is "subcommittee X3B8" to issue such pronouncements?

They're a group of people representing a large, well-balanced cross section of disciplines—from academia, government agencies, and the computer industry. People from places like IBM, Hewlett-Packard, 3M, Lawrence Livermore Labs, The U.S. Department of Defense, Honeywell and The Association of Computer Programmers and Analysts. In short, it's a bunch of high-caliber nitpickers whose mission, it seems, in order to make better disks for consumers, is also to make life miserable for everyone in the disk-making business.

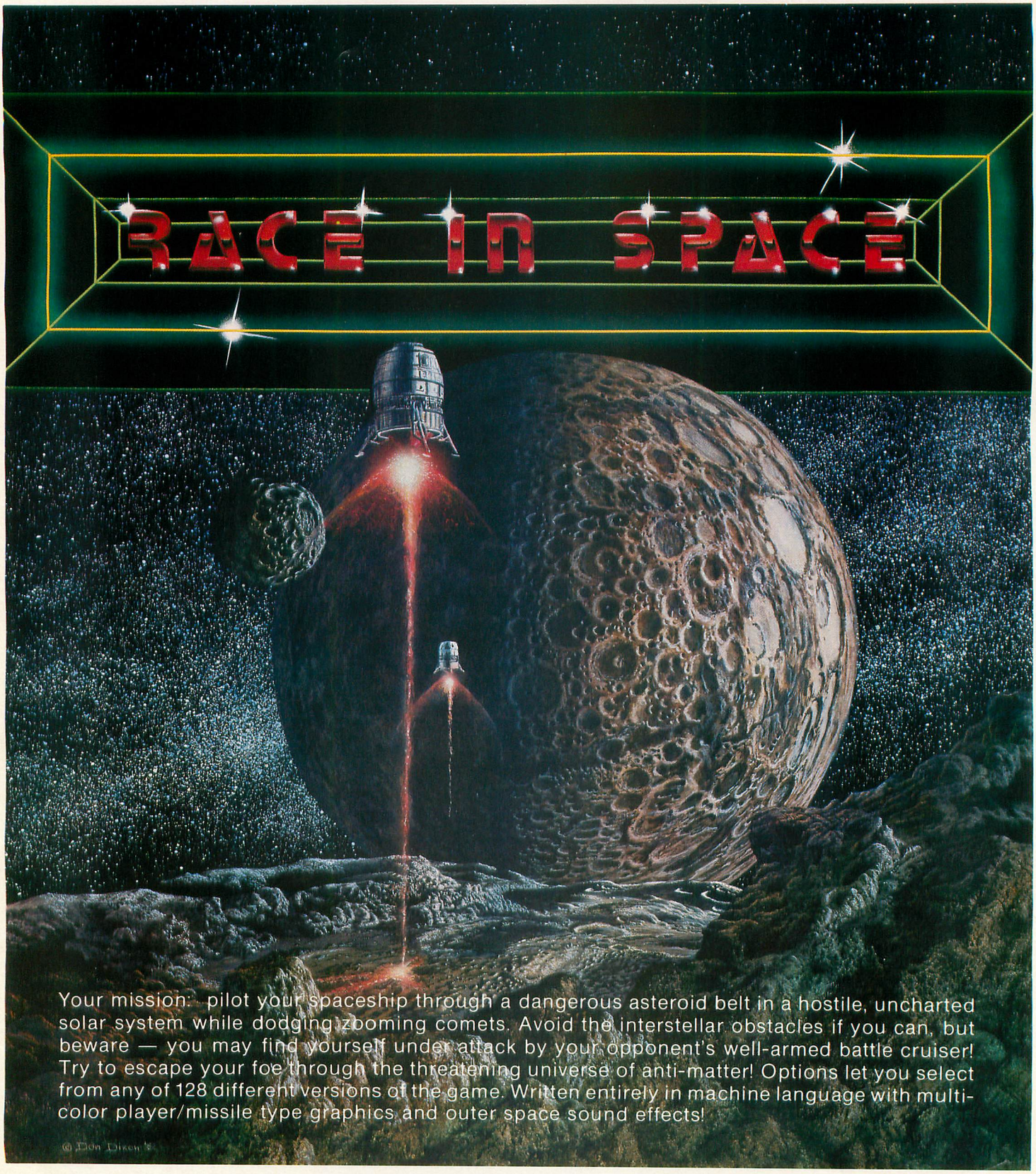
How? By gathering together periodically (often, one suspects, under the full moon) to concoct more and more rules to increase the quality of flexible disks. Their most recent rule book runs over 20 single-spaced pages—listing, and insisting upon—hundreds upon hundreds of standards a disk must meet in order to be blessed by ANSI. (And thereby be taken seriously by people who take disks seriously.)

In fact, if you'd like a copy of this formidable document, for free, just let us know and we'll send you one. Because once you know what it takes to make an Elephant for ANSI . . .

We think you'll want us to make some Elephants for you.

ELEPHANT.™ HEAVY DUTY DISKS.

Distributed Exclusively by Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021
Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.



RACE IN SPACE

THE BEST-SELLING GAME FROM

A.N.A.L.O.G. SOFTWARE

$21.95 cassette $24.95 disk

SEND CHECK OR MONEY ORDER TO:

A.N.A.L.O.G. Software

P.O. Box 23 Worcester, MA 01603

MasterCard and VISA accepted (617) 892-3488

NEW FROM
WESTON SOFTWARE

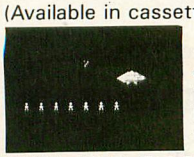
ATARI

AUTHORIZED EDUFUN! DEALER!

EduFun! Educational games for use with the Atari 400 and 800 Computers
Each EduFun!...program offers a unique combination of computer-generated learning and fun from a company with years of experience in educational publishing.
All of these games are a must for the family with young children.
Finally quality programs for your children. The games can be used by children aged 5 to 14...
We offer the New EDUFUN! programs. Each program offers a unique combination of educational design and creativity. Designed for both the home and school use, the programs make excellent use of computer graphics, sound and animation without sacrificing sound instructional principles and content. Developed by experienced educators and field tested by students and teachers in schools. The programs allow children to learn and explore basic concepts through games that teach and entertain the child.
EDUFUN! programs make learning fun. The new programs have been designed to teach basic math skills in an exciting way. The games provide the child with both practice in key mathematical skills and enrichment experiences.

A LOOK AT THE GAMES!!!! (Each package includes two games)

(Available in cassette 16K & diskette 32K)

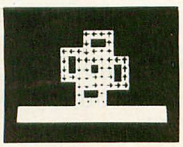


ALIENCOUNTER/FACE FLASH Ages 5 to 9

Aliencounter (counting) flying saucers...close encounters...alien visitors...how many should you land?
Face Flash (counting and base ten numeration) ready...set...now you see them...now you don't...how many were there?

THE JAR GAME/CHAOS Ages 6 to 10

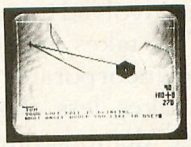
The Jar Game (problem solving and spatial visualization) follow the gold...pick the right jar...land the buzzing fly...you're sure to beat the computer.
Chaos (classification and direction) up, down, left, right...move your ship...capture the magic figures but don't crash.



GLUP!!/ARROW GRAPHICS Ages 7 to 12

Glup! (addition and multiplication drill) the race is on...faster, faster...don't get caught...watch out for those jaws! A great game to help master those basic facts.
Arrow Graphics (problem solving and spatial visualization) follow the bouncing arrow...where did it go?...retrace its steps...ring up those points!

GOLF CLASSIC/COMPUBAR Ages 9 and up



Golf Classic (angle and length estimation) fore!...play the angles...don't be long or short...a game of geometric skills for children aged 9 to 99
Compubar (reading graphs, constructing arithmetic expressions) add this bar... subtract that one...did you read them correctly?...you're on your way to becoming a Compubar master...

FRENZY/FLIP FLOP Ages 7 to 12

Frenzy (Subtraction & Division)...Frenzy is the name of the game...division and subtraction is the skill...super sound and graphics will cause the player to go wild with excitement...for children aged 7 to 12...
Flip Flop (Transformational Geometry)...rotate figures that are presented on the computer...figure out which diagrams are congruent...excellent graphic patterns for children ages 7 to 12...

BATTLING BUGS/CONCENTRACTION Ages 9 to 14

Battling Bugs (Interger Readiness)...Positive and Negative Numbers...Don't let this game bug you! Keep an eye on the bugs...they are very tricky...for children aged 9 to 14.
Concentration (Fractions Galore!)...try your skill with fractions...action with fractions can fractionize your brain...what a challenge! for children aged 9 to 14.

SUPPORT MATERIALS

Edufun! packages contain the following materials designed to help children use the programs more effectively:

- an instruction guide for children, parents and teachers
- reward stickers for good work
- ideas for additional, extending activities.

RECORD KEEPING

Diskette programs contain a record keeping program which stores information about the childs performance and provides parents and teachers with valuable data.

Mail to: **Weston Software**
P.O. Box 141
San Luis Obispo, Ca 93406
805-544-6863

All orders must be prepaid (money order, check or charge). All payments must be in US funds. (Outside the US add $4.00 shipping and handling for air mail, $2.00 for surface mail.) CA residents 6% sales tax.

Add $2.00 for shipping

☐ Payment Enclosed

Please charge my ☐ VISA ☐ MasterCard

Account No._____

Expires _____

Name_____

Address_____

City _____

State _____

Zip _____

BOOK SHELF

ATARI® Games and Recreation

A dynamite seller! Designed for the beginner as well as the experienced programmer. Provides a marvelous exploration of the types of games the Atari can run. Gives the programmer mastery of the Atari's capabilities. Has number and logic games, complete with graphs and visual tricks • letters, words and language games • appendices with flashcards, graph paper designs, and erro dictionary • game designers' toolbox included...Basic language...

288pp paper/$14.95

ATARI® Sound and Graphics

This crystal-clear guide is the first book to open up the vast creative possibilities of artistic programming to owners of the ATARI 400 and ATARI 800—the most visually advanced personal micros on the market. With this self-paced, self-teaching guide, you'll advance step-by-step through simple techniques for creating a fascinating array of sounds and images... compose and play melodies... draw cartoons... create sound effects and games... The book uses BASIC...

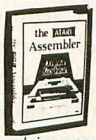
234pp paper/$9.95

Your ATARI® Computer: A Guide to ATARI® 400/800™ Personal Computers

Authored by the same team that brought you the *Apple II® User's Guide, Your ATARI® Computer* provides similar invaluable reference material for ATARI® home computer users. This is a "how-to" guide written for the novice. It covers all the common external devices, including disk drives and printers. The reader can also learn to program in ATARI® BASIC and use sound and color graphics to enhance the programs. Tables, charts and programming tips included.

458pp paper $15.00

The ATARI® Assembler



Here it is! A guide for using the *Atari Assembler Cartridge* with the Atari 400 and 800. Your customer's bonus: they'll also learn assembly language programming. Has a learn-by-doing approach • a refreshing writing style that makes learning fun • models to inspire creativity. Uses customer's BASIC background as a basis for learning.

320 pp paper/$12.95

ORDER FORM

_____	HCO1 ALIENCOUNTER/FACE FLASH (16K cassette)	$29.95
_____	HDO1 ALIENCOUNTER/FACE FLASH (32K diskette)	$32.95
_____	HCO2 THE JAR GAME/CHAOS (16K cassette)	$29.95
_____	HDO2 The JAR GAME/CHAOS (32K diskette)	$32.95
_____	HCO3 GULP!!/ARROW GRAPHICS (16K cassette)	$29.95
_____	HDO3 GULP!!/ARROW GRAPHICS (32K diskette)	$32.95
_____	HCO4 GOLF CLASSIC/COMPUBAR (16K cassette)	$29.95
_____	HDO4 GOLF CLASSIC/COMPUBAR (32K diskette)	$32.95
_____	HCO5 FRENZY/FLIP FLOP (16K cassette)	$29.95
_____	HDO5 FRENZY/FLIP FLOP (32K diskette)	$32.95
_____	HCO6 BATTLING BUGS/ CONCENTRACTION (16K cassette)	$29.95
_____	HDO6 BATTLING BUGS/ CONCENTRACTION (32K diskette)	$32.95
_____	B21 Atari Games and Recreations	$14.95
_____	B22 Atari Sound and Graphics	$ 9.95
_____	B23 Your Atari Computer: A Guide To Atari 400/800	$15.00
_____	B24 The Atari Assembler	$12.95

Faster Character Dumps

by Joseph T. Trem

If you are an avid ATARI Enthusiast as I am, then you probably have gone through quite a number of different programs. Many of the better programs use character redefinition. Unfortunately, to define a new character set one must move ATARI's character set into a new defined memory location. In BASIC, this takes time. For 1024 bytes (128 characters or 4 pages) it takes approximately eleven seconds. Darn right boring! Well...wait no more! This article demonstrates a machine language routine in string form which transfers ATARI's character set into a user chosen RAM area. It runs in a split second. Before going any further, I must state that this article is not a tutorial on character redefinition nor animation, although they are both used in the demo. There are many articles on these subjects including the ATARI Connection Volume 1, number 1 pages 22-23.

Here is a brief summary of the four sample programs included. Program one demonstrates character redefinition and the time involved in transferring 1024 bytes (line 40). This takes approximately 11 seconds. Program two incorporates the machine language routine and takes less than a second (line 30). Program three demonstrates a sample program with sound and animation. There are five redefined characters. After the program executes once, it recycles and re-executes all over again. Notice the time it takes to rerun...remember every time this program runs it is dumping 1024 bytes in under a second!! Program four shows the source code for the machine language routine. The technique used in this program is called a block move. We simply look at what is in ATARI's character base address and move that data to our new character address, one bit at a time. This technique is also good for player/missile graphics. You can zero out all player/missile data in a split second. Just think, no more time delay to clear P/M memory. Sound great? Then read on...here is the documentation for the first three programs.

PROGRAM 1

line	
20	Sets up character variables
40	Transfers characters (slow)
50-70	Reads in new character
80	Points to a new character base

PROGRAM 2

line	
20	Sets up character variables
30	Transfers characters (fast)
40-60	Reads in new character
70	Points to new character base

PROGRAM 3

line	
10	Sets up variables
100	Transfers 128 characters
1000-2170	Reads in five new characters
3000	Points to a new character base
3500-7000	Main loop for animation
10000	Sound routine

Here's some information on our USR call:

```
E$="hh,Oh,Nhh,T)♥,L)◆,M"H♥1L,NHPyfMfOhdTPp◆♥"
```

A=USR(ADR(E$),ADDR,PAGE

ADDR=address where new character set is to reside

PAGE=the number of 256 byte blocks you wish to move

In closing, I hope that everyone will enjoy the substantial increase in speed this subroutine will provide. Just think, no more "Please wait..." prompts. □

PROGRAM 1

```
10 REM ***DUMPS 1024 BYTES TO NEW CHBA
S USING ONLY BASIC (APROX. 11 SECONDS)
***
20 DIM E$(50):RAMTOP=PEEK(106)-8:POKE
106,RAMTOP:CHBAS=RAMTOP:ADDR=CHBAS*256
30 GRAPHICS 17:POSITION 0,9:? #6;"MOVI
NG CHARACTER SET"
40 FOR X=0 TO 1023:POKE ADDR+X,PEEK(57
344+X):NEXT X
50 CHAR=59:POS=ADDR+(CHAR*8)
60 DATA 0,24,36,66,153,66,60,0
70 FOR X=0 TO 7:READ A:POKE (POS+X),A:
NEXT X
80 GRAPHICS 18:POKE 752,1:POKE 756,CHB
AS
90 POSITION 10,5:? #6;"["
100 GOTO 100
```

●

PROGRAM 2

```
10 REM ***DUMPS 1024 BYTES TO NEW CHBA
S IN MACHINE LANGUAGE (NO DELAY)***
20 DIM E$(50):RAMTOP=PEEK(106)-8:POKE
106,RAMTOP:CHBAS=RAMTOP:ADDR=CHBAS*256
:PAGE=4
30 E$(1,41)="hh,Oh,Nhh,T)♥,L)◆,M"H♥1L
,NHPyfMfOhdTPp◆♥":A=USR(ADR(E$),ADDR,P
AGE):REM *DUMP ROUTINE*
40 CHAR=59:POS=ADDR+(CHAR*8)
50 DATA 0,24,36,66,153,66,60,0
60 FOR X=0 TO 7:READ A:POKE (POS+X),A:
NEXT X
70 GRAPHICS 18:POKE 752,1:POKE 756,CHB
AS
80 POSITION 9,5:? #6;"["
90 GOTO 90
```

PROGRAM 3

```
10 CLR :DIM E$(50):RAMTOP=PEEK(106):CH
BAS=RAMTOP-8:ADDR=CHBAS*256:PAGE=4:SND
=10000
100 E$(1,41)="hh[0h[Nhh[T]♥[L]X[H♥[
[[NHPUXfMfOhdTPP♦♥":A=USR(ADR(E$),ADDR,
PAGE):REM *DUMP ROUTINE (C)81 TREM*
)>*
1000 CHAR=59:POS=ADDR+(CHAR*8)
1010 DATA 0,0,144,96,144,0,0,0
1020 FOR X=0 TO 7:READ A:POKE (POS+X),
A:NEXT X
2000 CHAR=60:POS=ADDR+(CHAR*8)
2010 DATA 0,6,6,15,6,6,0,0
2020 FOR X=0 TO 7:READ A:POKE (POS+X),
A:NEXT X
2050 CHAR=61:POS=ADDR+(CHAR*8)
2060 DATA 0,0,0,20,8,20,0,0
2070 FOR X=0 TO 7:READ A:POKE (POS+X),
A:NEXT X
2100 CHAR=62:POS=ADDR+(CHAR*8)
2110 DATA 0,0,20,10,60,20,10,0
2120 FOR X=0 TO 7:READ A:POKE (POS+X),
A:NEXT X
2150 CHAR=63:POS=ADDR+(CHAR*8)
2160 DATA 0,148,104,57,86,72,2,0
2170 FOR X=0 TO 7:READ A:POKE (POS+X),
A:NEXT X
3000 GRAPHICS 17:POKE 752,1:POKE 756,C
HBAS
3010 POSITION 1,20:? #6;"ASSY CHARACTE
R DUMP"
3500 FOR X=0 TO 4:POKE 708,14:POSITION
 X,5:? #6;" [ ":GOSUB SND:POKE 708,8:P
OSITION X,5:? #6;" \ ":GOSUB SND
3510 NEXT X
3520 FOR I=1 TO 20:POKE 708,14:GOSUB S
ND:POKE 708,8:GOSUB SND
3530 NEXT I
4000 FOR X=5 TO 10:POKE 708,14:POSITIO
N X,5:? #6;" [ ":GOSUB SND:POKE 708,8:
POSITION X,5:? #6;" \ ":GOSUB SND
4010 NEXT X
4520 FOR I=1 TO 20:POKE 708,14:GOSUB S
ND:POKE 708,8:GOSUB SND
4530 NEXT I
5000 FOR X=11 TO 14:POKE 708,14:POSITI
ON X,5:? #6;" [ ":GOSUB SND:POKE 708,8
:POSITION X,5:? #6;" \ ":GOSUB SND
5010 NEXT X
6000 POSITION 15,5:? #6;"]":FOR D=14 T
O 10 STEP -1:SOUND 0,30,8,D:NEXT D
6010 POSITION 15,5:? #6;"^":FOR D=10 T
O 5 STEP -1:SOUND 0,100,8,D:NEXT D
6020 POSITION 15,5:? #6;"_":FOR D=5 TO
 0 STEP -1:SOUND 0,30,8,D:NEXT D
6030 POSITION 15,5:? #6;"_":FOR D=5 TO
 0 STEP -1:SOUND 0,30,8,D:NEXT D
6040 POSITION 15,5:? #6;"^":FOR D=10 T
O 5 STEP -1:SOUND 0,100,8,D:NEXT D
6050 POSITION 15,5:? #6;"]":FOR D=14 T
O 10 STEP -1:SOUND 0,30,8,D:NEXT D
6070 POKE 708,0:SOUND 0,0,0,0:SOUND 1,
0,0,0:SOUND 2,0,0,0
7000 GOTO 10
10000 SOUND 0,200,12,8:SOUND 2,203,12,
8:SOUND 1,RND(0)*10,10,8:RETURN
```

●

PROGRAM 4

```
10 ;CHARACTER DUMP (C)81 JOE TREM
20 OLD=$CC ;TEMP. LOCATION OF ATARI'S
CHARACTER SET
30 NEW=$CE ;TEMP. LOCATION OF NEW CHAR
ACTER SET
40 PAGE=$D4 ;NUMBER OF 256 BYTE BLOCKS
50  *=$600
60  PLA
70  PLA ;PULL HIGH BYTE OF ADDR
80  STA NEW+1
90  PLA ;PULL LOW BYTE OF ADDR
0100  STA NEW
0110  PLA ;PULL HIGH BYTE-DON'T NEED
0120  PLA ;PULL NUMBER OF BLOCKS TO MO
VE
0130  STA PAGE
0140  LDA #00 ;LOADS IN ATARI CHR.SET
0150  STA OLD
0160  LDA #$E0 ;ATARI CHR. SET IS AT $
E000 OR 57344 IN BASIC
0170  STA OLD+1
0180  LDX #1
0190  LDY #0
0200 LOOP LDA (OLD),Y
0210  STA (NEW),Y ;MOVES TO NEW AREA
0220  INY
0230  BNE LOOP
0240  INC OLD+1
0250  INC NEW+1
0260  INX
0270  CPX PAGE
0280  BNE LOOP
0290  RTS ;IF ALL BLOCKS ARE LOADED RE
TURN TO BASIC
0300    .END
```

●

Soundlab

by Dave Hallowell

You might find this program useful as a planning tool. I'll assume that you're already familiar with ATARI's four sound registers (see the BASIC Manual for background) and that you may have spent time feeding them frequency, tone and volume values by trial and error, to get a desired sound for a game or display. SOUNDLAB simply displays the numeric states of each sound register while letting you hear, blend, and change them. A delay loop is included to allow experimenting with timed pulses of sound for special effects.

The joystick control and continuous monitor of all four registers allows 'fine increment' blending of sounds to produce results you might not hit upon by punching in values from the keyboard.

USING THE PROGRAM

The following keys are active during SOUNDLAB:

's'=(Sound Register) — Allows the change of control (using the joystick) to any of the sound registers (0 through 3). Note that an asterisk marks the register under current control.

'V'=(Volume) — Lets the joystick increment the volume (0 through 15) of the chosen register.

'T'=(Tone) — Shifts joystick control to the chosen tone setting (even numbers only; 0 through 14)

'F'=(Frequency) — Allows joystick to set the frequency value of the register chosen. (values 0 through 255)

'SPACE BAR' — Initiates "Delay Loop" while displaying a delay value. Note that the delay value can be changed **only** when the fire button is depressed **and** when the end of a delay loop comes around. I'll explain why a little later.

'START' — Erases the values in all four sound registers and resets the program back to the beginning. (Handy for 'starting over' if you aren't pleased with the sounds you're producing)

'Fire Button' — Activates a 'rush loop' allowing rapid incrementing of the value being adjusted. Useful in moving through a range of frequencies quickly. Note that in the delay loop option this feature **must** be used to make adjustments.

THINGS TO TRY

The 'pure tone' setting of 10 (ex. SOUND 0,200, 10,8) gives a rather shallow sounding tone. Try blending one such sound with a near-neighbor frequency (ex. SOUND 0,200,10,8:SOUND 1, 201, 10,8). The falling into and out of phase of these two (or try more) tones will give a much 'fatter' and more pleasing sound. More realistic jet sounds (using especially tone settings 0 and 8) are possible using the same idea.

Try changing the delay loop (lines 600-700) to allow selective entry or exemption of registers into the loop. Then you could produce 'two-dimensional' sounds with a pulsation happening over a steady unbroken background.

Try writing a subroutine that writes DATA statements to a disk or tape and allows the accumulation of a melody that you have worked through note-by-note by ear.

LAST WORDS

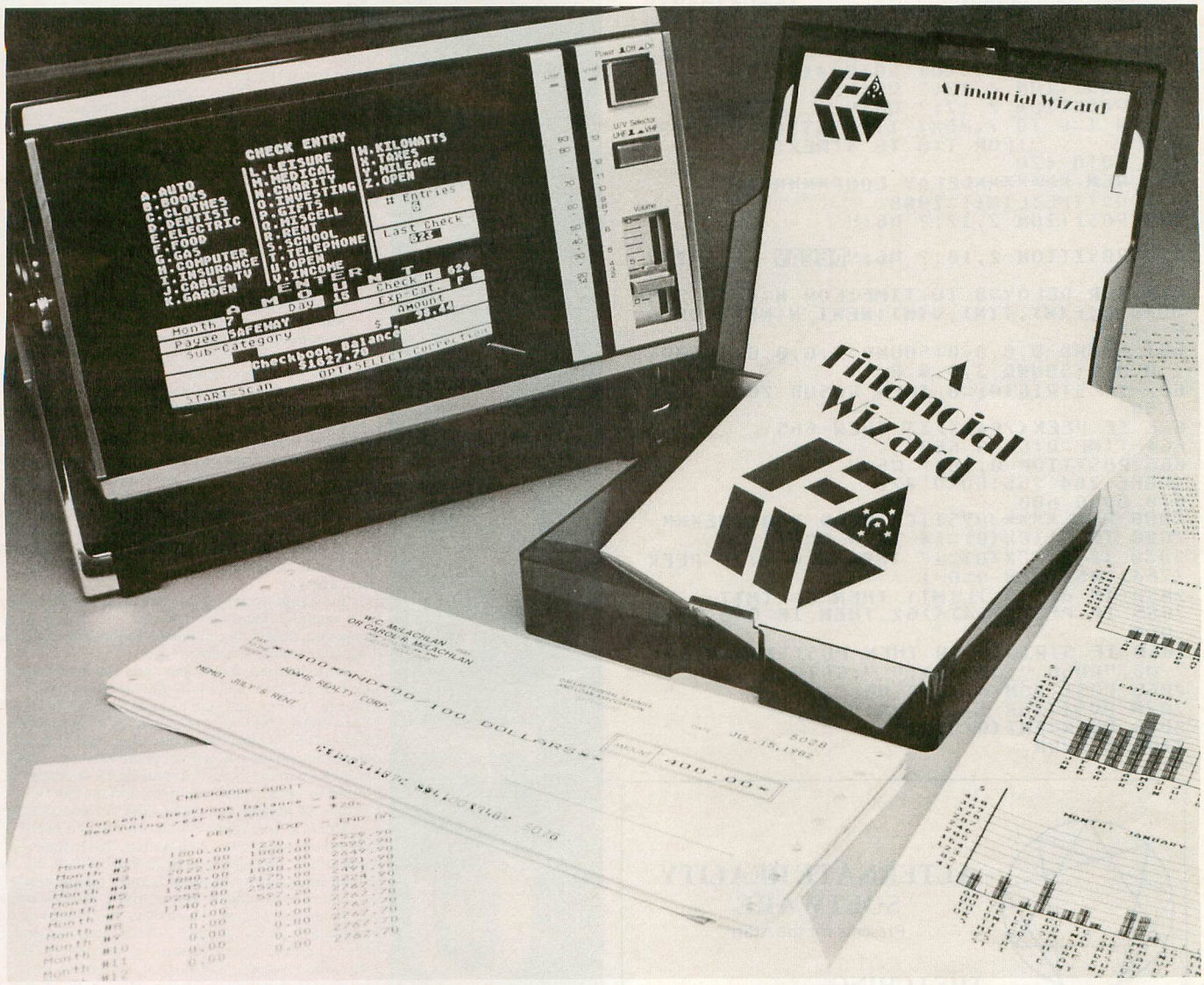
I've kept the program very simple and modular to allow you to expand it to your own liking. Regarding the delay loop; there is a reason that the lowest number is 2 and that the fire button is needed to actuate a change. You may already know that FOR-NEXT loops vary in their execution time depending on where they are sandwiched into a program. Since mine includes some unavoidable extra delay time (checking the condition of the fire button, etc.) the fastest execution possible is about what you'd get using a 2 in the fastest possible format (See example 1). I calibrated on a 2 to remind you that faster execution is possible in situations you might set up in your own programs. If I had included the whole joystick reading subroutine (lines 2000-2100) in the delay loop the execution would be much slower. That's why you have to press fire and wait for the end of a loop to get in and make a change. □

```
200 REM ****SOUNDLAB************
210 REM ************BY DAVE HALLOWELL
220 DIM F(4),T(4),V(4)
225 FOR I=0 TO 3:F(I)=0:T(I)=0:V(I)=0:
NEXT I
400 REM SETUP DISPLAY
410 GRAPHICS 17
411 POKE 712,18:POKE 710,12:POKE 708,8
:POKE 709,14
415 POSITION 5,2:? #6;"SOUNDLAB":IF PE
EK(53279)=6 THEN CLR :GOTO 200
418 H=2*N+7
420 POSITION 2,7:? #6;"SOUND 0,";F(0);
",";T(0);",";V(0);" "
422 POSITION 1,H:? #6;"►"
425 SOUND 0,F(0),T(0),V(0)
430 POSITION 2,9:? #6;"SOUND 1,";F(1);
",";T(1);",";V(1);" "
435 SOUND 1,F(1),T(1),V(1)
440 POSITION 2,11:? #6;"SOUND 2,";F(2)
;",";T(2);",";V(2);" "
445 SOUND 2,F(2),T(2),V(2)
460 POSITION 2,13:? #6;"SOUND 3,";F(3)
;",";T(3);",";V(3);" "
465 SOUND 3,F(3),T(3),V(3)
470 IF PEEK(764)=56 THEN POSITION 2,17
```

(continued on page 36)

The New Standard

A FINANCIAL WIZARD 1.5



THE ULTIMATE PERSONAL FINANCE SYSTEM FOR ATARI*

- Budget – forecast 26 expense categories
- Check Entry – easy data entry – scan & correct 26 major & 36 sub-categories – information block
- Check Search – single or multiple parameters – (up to seven) to search entries
- Tabulations – detailed expense vs. budget 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 – print your custom checks
- Printouts – most popular printers • Audit Report
- Multi-Colored Graphics • Audio Enhancements
- 7 Utility Programs • User-Friendly Operation
- Easy To Use Instruction Manual
- Tinted Plastic Storage Case

The system is designed for Atari computers having a minimum of 24K and operating from a disk drive. The price 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. (405)751-2783.

*trademark of Atari Inc. OK residents ad 4% Tax.

A Financial Wizard Exclusively thru ON LINE COMPUTER CENTERS of OKC ™

10944 North May
Oklahoma City, OK 73120
(405) 751-2781

☐ CHECK ENCLOSED ☐ VISA ☐ MASTERCHARGE

NAME (Print) _____

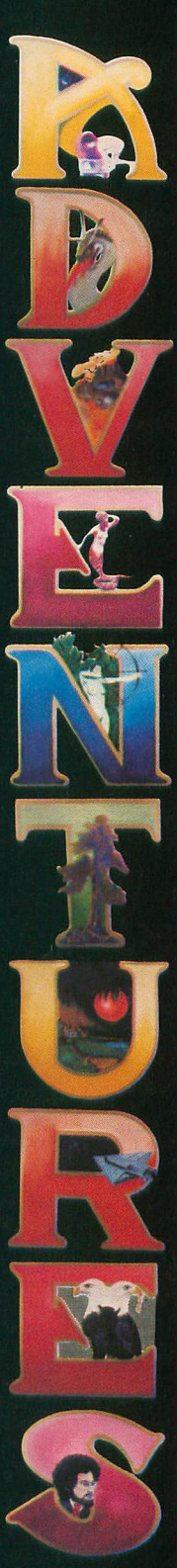
Address _____

City _____ State _____ Zip _____

Card # _____ Exp. _____ Sig. _____

```
:? #6;"    FREQUENCY       ":LIMIT=255:Q
=F(N):GOSUB 2000:F(N)=ABS(Q):GOTO 415
472 IF PEEK(764)=33 THEN POSITION 2,17
:? #6;"                    ":GOTO 600
474 IF PEEK(764)=45 THEN POSITION 2,17
:? #6;"      TONE          ":LIMIT=14:Q=T(
N)+1:GOSUB 2000:T(N)=EVEN:GOTO 415
478 IF PEEK(764)=16 THEN POSITION 2,17
:? #6;"    VOLUME          ":LIMIT=15:Q
=V(N):GOSUB 2000:V(N)=ABS(Q):GOTO 415
480 IF PEEK(764)=62 THEN POSITION 2,17
:? #6;"SOUND REGISTER":Q=-1*N:LIMIT=3:
GOSUB 2000:N=ABS(Q):GOTO 415
490 POSITION 4,17:? #6;""
495 POSITION 2,17:? #6;"     SELECT
":FOR T=0 TO 7:NEXT T:POSITION 6,17:?
#6;"       ":FOR T=0 TO 4:NEXT T
499 GOTO 470
600 REM ********DELAY LOOP********
630 Q=TIME:LIMIT=2000
635 POSITION 2,17:? #6;"
   "
650 POSITION 2,18:? #6;"DELAY  ";Q+2;"
 ";
660 FOR DELAY=0 TO TIME:FOR N=0 TO 3:S
OUND N,F(N),T(N),V(N):NEXT N:NEXT DELA
Y
661 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND
2,0,0,0:SOUND 3,0,0,0
662 IF STRIG(0)=0 THEN GOSUB 2000:GOTO
 650
663 IF PEEK(764)<>33 THEN 665
664 TIME=Q:GOTO 660
665 POSITION 0,18:? #6;"                 "
:POKE 764,255:GOTO 470
670 GOTO 600
2000 REM ****JOYSTICK READ ROUTINE***
2020 IF STICK(0)=14 THEN Q=Q+1
2030 IF STICK(0)=13 THEN Q=Q-1:IF PEEK
(764)=45 THEN Q=Q-1
2050 IF ABS(Q)>LIMIT THEN Q=LIMIT
2055 IF PEEK(764)<>62 THEN IF Q<0 THEN
 Q=0
2060 IF STRIG(0)=0 THEN POSITION 8,19:
? #6;"RUSH ";ABS(Q);" ":GOTO 2000
2065 POSITION 8,19:? #6;"                "
2070 POSITION 1,H:? #6;" "
2080 EVEN=INT(Q/2)*2
2100 RETURN
```


The Scott Adams Adventure Series
AN OVERVIEW



I stood at the bottom of a deep chasm. Cool air sliding down the sides of the crevasse hit waves of heat rising from a stream of bubbling lava and formed a mist over the sluggish flow. Through the swirling clouds I caught glimpses of two ledges high above me: one was bricked, the other appeared to lead to the throne room I had been seeking.

A blast of fresh air cleared the mist near my feet and like a single gravestone a broken sign appeared momentarily. A dull gleam of gold showed at the base of the sign before being swallowed up by the fog again. From the distance came the angry buzz of the killer bees. Could I avoid their lethal stings as I had managed to escape the wrath of the dragon? Reading the sign might give me a clue to the dangers of this pit.

I approached the sign slowly.

And so it goes, hour after hour, as you guide your microcomputer through the Adventures of Scott Adams in an effort to amass treasures within the worlds of his imagination.

By definition, an adventure is a dangerous or risky undertaking; a novel, exciting, or otherwise remarkable event or experience. On your personal computer, Adventure is that and more.

For the user, playing Adventure is a dangerous or risky undertaking in that you better be prepared to spend many addictive hours at the keyboard. If you like challenges, surprises, humor and being transported to other worlds, these are the games for you. If you dislike being forced to use your common sense and imagination, or you frustrate easily, try them anyway.

In beginning any Adventure, you will find yourself in a specific location: a forest, on board a small spaceship, outside a fun house, in the briefing room of a nuclear plant, in a desert, etc.

By using two-word commands you move from location to location, manipulate objects that you find in the different places, and perform actions as if you were really there. The object of a game is to amass treasure for points or accomplish some other goal. Successfully completing a game, however, is far easier to state than achieve. In many cases you will find a treasure but be unable to take it until you are carrying the right combination of objects you find in the various locations.

How do you know which objects you need? Trial and error, logic and imagination. Each time you try some action, you learn a little more about the game. Which brings us to the term "game" again. While called games, Adventures are actually puzzles because you have to discover which way the pieces (actions, manipulations, use of magic words, etc.) fit together in order to gather your treasures or accomplish the mission. Like a puzzle, there are a number of ways to fit the pieces together; players who have found and stored all the treasures (there are 13) of Adventure #1 may have done so in different ways.

In finding how the pieces fit, you will be forced to deal with unexpected events, apparent dead ends and Scott's humor, which is one of the best parts of the puzzles.

If you run into a barrier like not being able to discover more rooms, don't give up. Play the game with some friends; sometimes they'll think of things you haven't tried.

While I pondered how to reach the throne room — which I was sure contained the treasures of Croesus — the fog grew thicker and the hours passed. I realized I would not be able to outwit Adams today...but maybe tomorrow. I marked my present location on my tattered map and began the long trip to the surface. As I dragged myself off to bed, I thought about other possible Adventures.

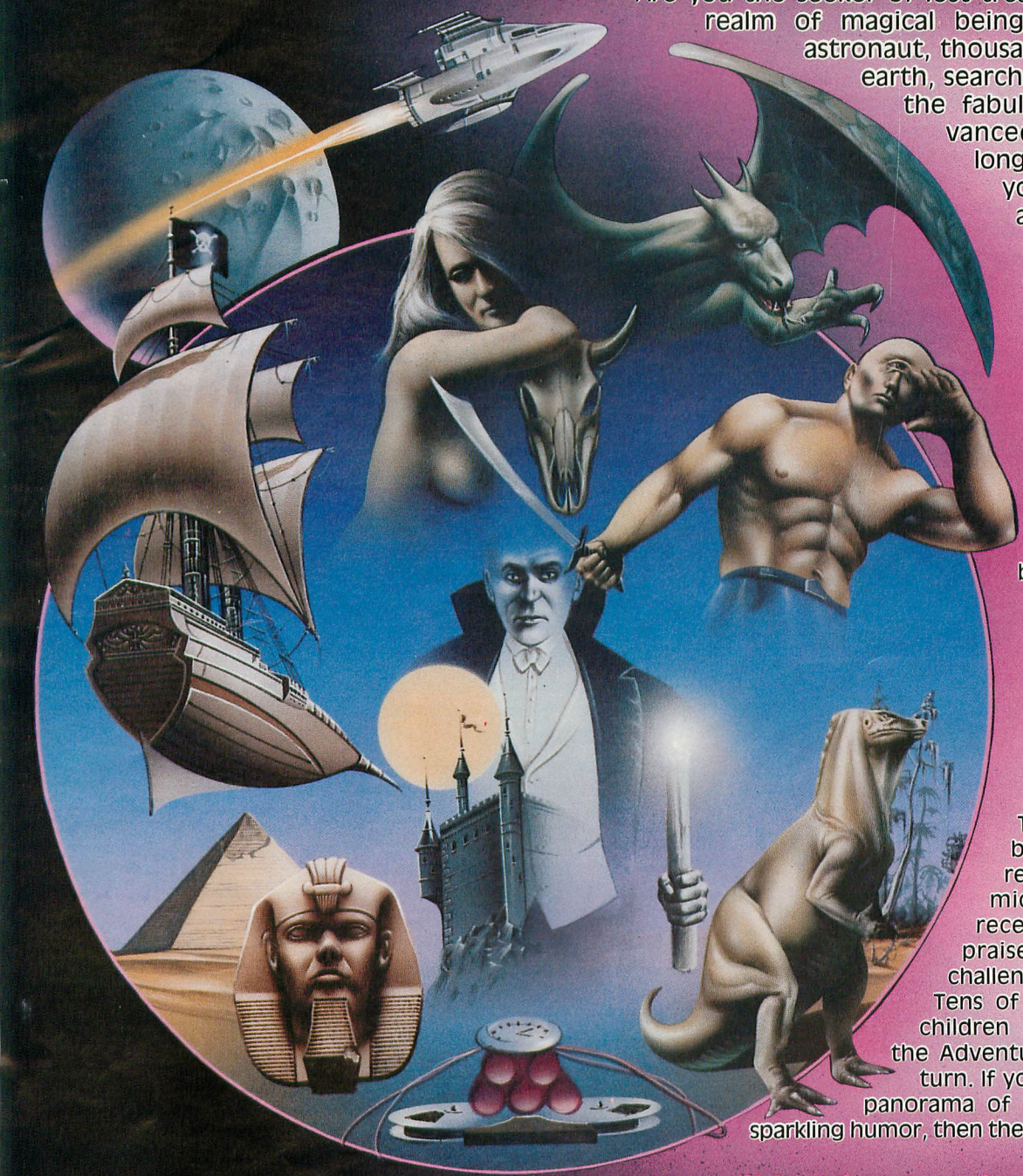
But enough for tonight. Tomorrow — another crack at the chasm. —by Ken Mazur

Reprinted with permission from
PERSONAL COMPUTING MAGAZINE, FEB. 1980

The 12 Scott Adams Adventures

Adventureland • Pirate Adventure • Mission Impossible • Voodoo Castle • The Count • Strange Odyssey • Mystery Fun House • Pyramid of Doom • Ghost Town • Savage Island-Part 1 • Savage Island-Part 2 • Golden Voyage

WHO ARE YOU THIS TIME?



Are you the seeker of lost treasures in an enchanted realm of magical beings? Perhaps you're an astronaut, thousands of light years from earth, searching the galaxy's rim for the fabulous treasures and advanced technologies of a long-dead civilization. Maybe you're the plunderer of ancient pyramids in a maddeningly dangerous land of crumbling ruins and trackless desert wastes.

Imagine.

You wish to soar to other worlds, to behold wonders never beheld by mortal eyes, to dream unrestrained to the furthest limits of your imagination.

Imagine now that you can.

The ADVENTURE SERIES by Scott Adams has been reviewed by **every** major microcomputing magazine, receiving only the highest praise for its mind-puzzling challenge & refreshing originality. Tens of thousands of adults and children have matched wits with the Adventure Master. Now, it's your turn. If you crave the challenge and panorama of the exotic, touched with sparkling humor, then the Adventure Series is for you.

With the incredible Adventure Series by Scott Adams, YOU ARE WHOMEVER YOU WANT TO BE, WHENEVER YOU WANT TO BE!

	SINGLE TAPE	12 PACK TAPE	SINGLE DISK	TRIPLE DISK	12 PACK DISK
APPLE	$19.95 24K	$129.95† 24K	$29.95* 48K	$69.95* 48K	—
ATARI	$19.95 24K	$129.95† 24K	—	$39.95 40K	$129.95† 48K
CP/M (Z-80)	—	—	—	—	$129.95†
TRS-80 MOD 1&3	$19.95 16K	$129.95† 16K	—	$39.95 32K	$129.95† 32K
TRS-80 MOD 2	—	—	—	—	$129.95

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

Adventure Hint Book (All 12)	$ 7.95
Adventure Hint Sheet (Single)	1.00
Commodore VIC (Cartridge)	39.95
T.I. 99/4 (Tape or Disk)	29.95

Requires Adventure Command Module

*Scott Adams' Graphic Adventure Series Apple Disks-Full Color Hi-Res Graphics.
†12 Adventure Hint Book Incl.

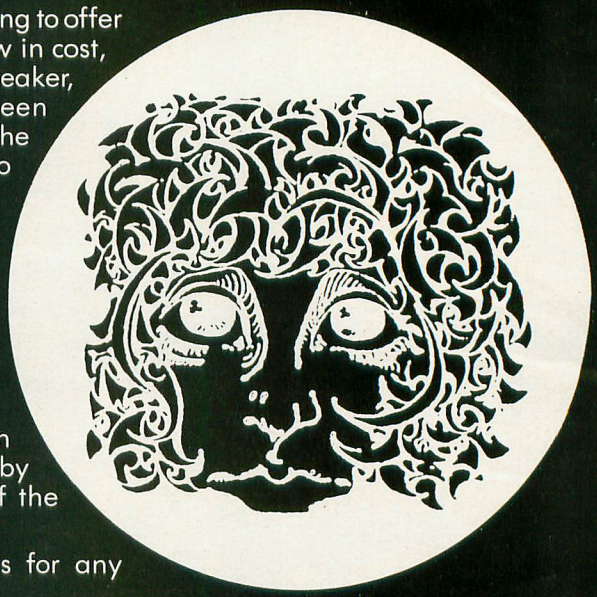
SHIPPING & HANDLING ARE EXTRA

Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.

BOX 3435, LONGWOOD FL 32750
(305) 862-6917 (QUESTIONS)
ORDER FROM YOUR FAVORITE DEALER
or CALL TOLL FREE (800) 327-7172
(ORDERS ONLY PLEASE A.I. #119)

ATARI SAYS ITS FIRST WORD

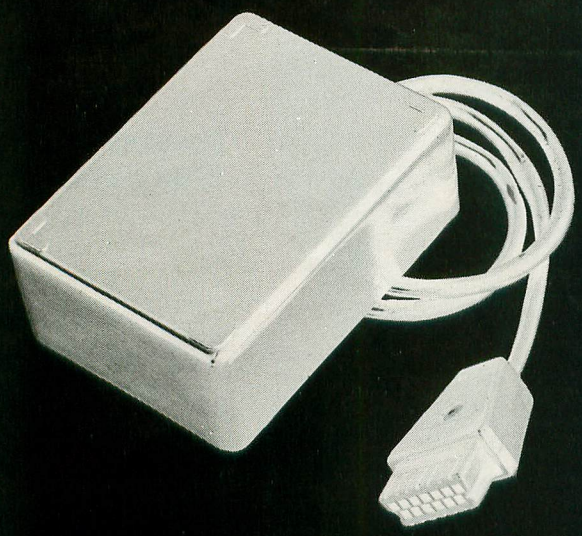
WITH A VOICE BOX BY THE ALIEN GROUP!



THE ALIEN GROUP has emerged from the underground, daring to offer a full-featured speech synthesis system that is flexible, low in cost, and needs no accessory devices. No Interface, Cables, Speaker, Amp, or External Power Required! VOICE BOX has been designed and programmed by Atari users to become the integral voice of a *400* or *800* computer. Simply plugged into the serial port, VOICE BOX automatically routes all speech into the speaker of your television monitor. With the menu-driven operating system supplied, you'll be creating original, intelligible speech within moments after loading disk or cassette. No lengthy or obscure instructions to wade through.

The system includes a dictionary which translates typed text into VOICE BOX's phonetic language. The dictionary can be expanded to include as many as *5,000* words of your own custom vocabulary. Unlimited speech can be produced by straightforward phonetic definiton at any time, even if the dictionary should be full.

The speech routines can be called from other programs for any purpose you can conceive. Here are a few suggested uses:



VOICE BOX
Speech Synthesizer

SOUND EFFECTS
- Access 64 phonemes at any of 4 pitch levels to add filtered, contoured sound to the Atari audio repertoire.

GAMES
- Program aliens to hiss threats, moan when destroyed.
- Devise weird, non-human tongues for dungeon dwellers.
- Insert cryptic spoken clues in maze games.

COMPUTER OPERATION
- Code verbal prompts and error messages that command attention and leave the current display intact.

EDUCATION
- Gain an introduction to the principles of phonetics.
- Learn touch typing through spoken feedback from the keyboard.

In addition, the *Random Sentence Generator* included in the operating system, which prints and speaks endlessly startling, amusing, even poetic combinations of words supplied by the user, helps teach school children to identify parts of speech and recognize a variety of sentence structures.

A minimum of 16K RAM is required by the operating system. Either disk or cassette includes both 16K and 32K versions. Try VOICE BOX for up to 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.

When ordering specify disk- or cassette-based operating system, and enclose check or money-order for $169, or state your VISA or MASTERCARD number. Send mail orders to: THE ALIEN GROUP
27 West 23rd Street
Dept. AL-1
New York, New York 10010

or telephone orders from
10 AM to 6 PM New York time
(212) 924-5546

ALSO AVAILABLE AT LEADING COMPUTER STORES THROUGHOUT THE WORLD.

Budget Worksheet

CASSETTE, 16K

by Aly Kahn

This program started with a rototiller. It seems that my tiller needed a set of knobs to hold the handle on securely. Upon arrival at the local power equipment store to my surprise I saw a sign that was to signal my return to the world of computers after a fifteen year absence.

The sign "COMING SOON — THE COMPUTER SHOPPE" caught both my attention and imagination, simultaneously. Perhaps I was using the wrong type of power tool; the new power tool was the personal computer that everyone seemed to be touting these days.

Two long weeks passed and finally the store opened. The following week I purchased an ATARI 400 and the other sundry items to feed the beast. That was the beginning of the end for going to bed early for the rest of the year and then some. Eventually, after some rather sleepless nights, the 400 began to make sense. My wife began to make sense. But, alas, and alack, my finances made no sense at all.

My wife Marion and I use a weekly budget that requires frequent updating. It has taken up 12 looseleaf notebooks since our marriage 8 years ago. It seemed to me that the 400 could do the budgeting and record storage more quickly and store the information in less room. The answer lay in learning how to use the screen and editor I/O.

There was also a third I/O unit that caught my attention. Not being rich, I did not buy a disk to go with the 400. Cassette I/O was the solution to my poverty.

The program is fairly straight-forward. The title page is displayed, followed by a prompt with two choices. If the user wishes to create a budget he types save and hits the return key. He will then be asked to input the month (alphabetically), day, and year. The screen then clears and asks the user to enter eight (or less) budget categories. Upon completion of each category the user must hit the return key for the next prompt. After the eighth category is entered and the return key depressed, the screen again clears and the user inputs the dollar and cents amounts of each category. At the final input he hits return and the complete budget is displayed in columnar structure with a date on the bottom and the total of the budget. If the user wishes he can save the budget to a fresh cassette by hitting return, getting two beeps, and hitting return again. The screen and cassette I/O routines (lines 217-240) then save the screen on the new cassette. Before or after the saving of the screen the user can return to the menu. If it is before the save and he wishes not to save the budget he hits "M" and the menu will be displayed. After the save the screen will automatically go to the menu.

An explanation of the I/O routines begins with line 217. The keyboard (device "K:") is opened for input only (4). I/O handler #3 is used to store the value of the key that is pushed. (155 is the value of the RETURN key.) If the value is 155 the program branches to the cassette I/O routine at line 226. It opens device #4 handling the cassette to write (8). The device is specified by "C:" (for cassette).

The for-next routine (228-230,234,236) is used to get all character positions possible on a Graphics O screen. (24 x 40). The locate routine takes each of these locations and finds the letter or blank stored at each location; the put statement puts the stored letter to device #4("C:") and then goes to the next value of Y and X until the for-next loop is completed (24 x 40=960) after 960 positions. The device handlers are then closed.

The loading routine uses the same principles but in a reverse fashion and with two additional lines used. I found the need for the screen editor "S:" and the Poke 82,0 (left margin) necessary to maintain the columnar structure of the budget on the screen after it is accessed. The for-next statement on the load routine uses the 960 positions from the save to write the stored information to the screen. (It gets (4) the values (A) from the cassette and puts the values (A) to the screen (2).

One more item needs to be explained. Upon loading the screen will go blank until the stored information is found; as it begins, the screen will come back on and the cursor will trace across all positions on the screen, leaving the budget behind it.

My thanks to John and Anne Smith of the Computer Shoppe for their technical help and for the use of their printer at the most unadvantageous times during their business day.

That about wraps the BUDGET WORKSHEET up. I wonder...if I use the program for a year will I enough money left over for a disk drive, a printer, a modem, and, of course, alimony payments? Happy hacking... □

```
0 GRAPHICS 2+16:SETCOLOR 2,4,4:CLR
1 POKE 764,255:REM BUDGET WORKSHEET
  REV.2.0 3/27/82
2 POSITION 5,1:? #6;"BUDGET"
3 POSITION 5,3:? #6;"WORKSHEET"
4 POSITION 5,6:? #6;"BY aly kahn"
5 POSITION 5,8:? #6;"  1982  "
6 FOR X=1 TO 200 STEP 0.1:NEXT X
7 DIM MENU$(4)
8 ? "TO LOAD DATA TYPE LOAD":? "TO BUD
GET TYPE SAVE"
9 TRAP 8:INPUT MENU$:IF MENU$="LOAD" T
HEN 1000:IF MENU$="SAVE" THEN 10
10 DIM A$(10),A(6),B$(10),B(6),C$(10),
C(6),D$(10),D(6),E$(10),E(6),F$(10),F(
6)
11 DIM MONTH$(9),DATE(2),YEAR(4)
12 DIM G$(10),G(6),H$(10),H(6)
13 TRAP 13:? "INPUT MONTH":INPUT MONTH
$:? "INPUT DAY":INPUT DATE:? "INPUT YE
AR":INPUT YEAR:TRAP 40000
14 ? CHR$(125)
15 ? "Type budget category- up to 9 le
tters"
16 ? "(UP TO 8 CATEGORIES,PLEASE)"
17 POKE 752,1
18 FOR L=1 TO 10:SOUND 2,INT(RND(1)*25
5)+1,10,10:NEXT L:SOUND 2,0,0,0
20 TRAP 20:INPUT A$
30 TRAP 30:INPUT B$
40 TRAP 40:INPUT C$
50 TRAP 50:INPUT D$
60 TRAP 60:INPUT E$
70 TRAP 70:INPUT F$
72 TRAP 72:INPUT G$
74 TRAP 74:INPUT H$
75 ? CHR$(125)
77 ? "Enter amount after category-up t
0 6 digits"
78 ?
79 ? "IF CATEGORY IS BLANK ,PLEASE ENT
ER 0"
80 TRAP 80:? A$;"=";:INPUT A
90 TRAP 90:? B$;"=";:INPUT B
100 TRAP 100:? C$;"=";:INPUT C
110 TRAP 110:? D$;"=";:INPUT D
120 TRAP 120:? E$;"=";:INPUT E
130 TRAP 130:? F$;"=";:INPUT F
132 TRAP 132:? G$;"=";:INPUT G
134 TRAP 134:? H$;"=";:INPUT H:TRAP 40
000
140 ? CHR$(125)
141 ? "EXIT TO MENU HIT  "M""
142 ? "To save data to tape,hit return
"
143 ? :? :?
145 ? A$,A
150 ? B$,B
160 ? C$,C
170 ? D$,D
180 ? E$,E
190 ? F$,F
192 ? G$,G
194 ? H$,H
200 LET TOTAL=(A+B+C+D+E+F+G+H)
201 FOR Z=1 TO 10:SOUND 0,TOTAL,10,10:
NEXT Z:SOUND 0,0,0,0
208 ? :?
210 ? "Total",TOTAL
213 ? :? MONTH$;" ";DATE;",";" ";YEAR
217 OPEN #3,4,0,"K:"
218 GET #3,A:IF A=155 THEN 226
219 IF A=77 THEN CLOSE #3:GOTO 259
220 GOTO 218
226 OPEN #4,8,0,"C:"
228 FOR Y=0 TO 23
230 FOR X=0 TO 39
232 LOCATE X,Y,Z
233 PUT #4,Z
234 NEXT X
236 NEXT Y
240 CLOSE #4:CLOSE #3
259 POKE 764,255:GRAPHICS 0:SETCOLOR 2
,4,4
260 ? "TO START A NEW WORKSHEET, HIT S
TART"
263 ?
264 ? "TO EXIT PROGRAM,HIT OPTION"
265 ? "TO LOAD PROGRAM PUSH SELECT"
270 IF PEEK(53279)=6 THEN 0
290 IF PEEK(53279)=3 THEN ? CHR$(125):
FOR R=1 TO 10:SOUND 1,INT(RND(1)*255)+
1,10,10:NEXT R:SOUND 1,0,0,0:END
291 IF PEEK(53279)=5 THEN GOSUB 1000
295 GOTO 270
1000 ? "R";:OPEN #4,4,0,"C:"
1001 POKE 82,0
1005 OPEN #5,12,0,"E:"
1010 OPEN #2,8,0,"S:"
1020 FOR I=1 TO 960
1030 GET #4,A:PUT #2,A
1040 NEXT I
1050 CLOSE #4:CLOSE #5:CLOSE #2
1052 ? "FOR MENU HIT ANY KEY"
1054 IF PEEK(764)<>255 THEN POKE 764,2
55:GOSUB 259
1055 GOTO 1054
```

Multiprocessing On Your ATARI

by Mark Chasin

No, this article will not enable you to set up a timesharing service on your ATARI home computer, but it will try to demonstrate how to implement a form of multiprocessing which has been used in a number of recently released programs for the ATARI. To understand the principles of this program, you will need some background on how the video display operates.

The beam of electrons generated in the cathode ray tube of your TV set is focused and directed at the phosphors on the screen. The beam begins scanning the screen at the upper left corner, and proceeds across the screen from left to right. At the right edge, it returns to the left side and drops down one scan line, and proceeds to the right again. This process is repeated 262 times until the whole screen has been scanned, and then the beam is turned off and returned to the upper left corner to repeat the process again, sixty times a second.

This seems like a great deal to handle in one-sixtieth of a second, but your ATARI has a machine cycle time of 560 nanoseconds, so in that time interval, the ATARI can execute approximately 30,000 cycles. The result of this is that when the beam returns to the upper left corner of the screen, there is a good deal of time to waste before it must start scanning again. At this point, the ATARI goes off on its own, performing a number of housekeeping functions, updating timers, and the like. Ultimately, it returns to the business of drawing on the screen.

The folks who built your ATARI designed the system so that it could be modified easily by anyone wanting to do so, and the remainder of this article will discuss such modification. The computer "knows" where to go during the wait described above because two memory locations contain a hexadecimal address telling it where to go, and every time it gets to the upper left corner of the display, it looks in these memory locations and goes to the indicated address, where the housekeeping routines are stored. This process is called vectoring. There are actually two independent routines performed during each interval, and separate vectors exist for each. The immediate vertical blank vector is found at hexidecimal address $0222 and $0223, and the second vector, called the deferred vertical blank vector, is found at $0224 and $0225. What we are about to do is change the address located at $0224, $0225 to point to our own routine, and then we'll jump back into the routine that the ATARI was originally pointing to. When this is accomplished, our routine will execute 60 times per second, and will continue to execute until we either turn off the computer, or hit SYSTEM RESET. This will be totally independent of anything else we may be doing at the time, such as programming, editing, or playing a game!

The BASIC program shown in Figure 2 is simply an implementation in BASIC of the Assembly language program shown in Figure 1, so I will describe the operation of the Assembly language program in detail. First, I will list the locations and their uses within the program.

COUNT 1... used to determine how many times we have gone through the routine, to calculate when to start and stop the notes to be played.
COUNT 2... used to remember which note the routine is playing.
VVBLKD... the location of the deferred vertical blank vector.
SETVBV..... an ATARI routine, described in more detail below.
MUSIC ..... the location where the list of notes to be played is stored.
RETURN... this is where we need to jump to return to the ATARI housekeeping routines.
SND ..... the frequency register for SOUND O
VOL ..... the distortion and volume register for SOUND O

Lines 130 and 170-190 are housekeeping functions of this routine. Line 130 provides the PLA instruction necessary for accessing this routine from BASIC, and lines 170-190 set both COUNTs to 0. Lines 230-270 repoint the delayed vector to our routine, as follows. Since the 6502A inside your ATARI is an 8 bit processor, we can only handle one byte at a time. It should be obvious that if the computer tries to access this vector we have changed one byte of the address but before we have changed the second, the computer will go on a wild goose chase looking for where it should be. To prevent this, those clever folks who wrote the operating system for your computer built in a routine, called SETVBV, which will change these vectors without the chance of fouling things up. To use it, we load the Y register with the new vector low byte, and the X register with the new vector high byte, $20 and $06,

respectively, in this case, since our routine is located at $0620. We then load the accumulator with a 7 if we are setting the deferred vector, or a 6 if the immediate vector, and then we jump to the subroutine SETVBV. Presto! Our vector is changed, and the routine starts operating.

This routine will play a little familiar background music while you slave away over a hot computer. Later on, I'll describe how to change the tune to your own selection. The routine starts on line 320. This is the first time through, so we increase COUNT 1 to one. If COUNT 1=12, we'll turn the note off, and when COUNT 1=15, we'll play the next note, and reset COUNT 1 to 0. Lines 360-370 shut off the note, lines 410-420 reset the count to 0, and lines 430-470 play the next note. The tune consists of eight notes repeated over and over, and COUNT 2 keeps track of which note is being played. When it gets up to 8, it's reset to 0 (lines 480-530), so the first note is played right after the eighth. If COUNT 1 is not equal to either 12 or 15, the routine ends and returns to the normal housekeeping functions performed by the ATARI during the vertical blank period (line 400). Also, after a new note is started, the same thing happens (line 540). The table of notes played in the tune is located in line 590.

The BASIC program in Figure 2 simply converts the instructions described above into decimal form, and POKEs the routines into the correct place in memory. The routine is then set in motion with the USR call in line 27000, and from that point on, can be ignored. It will continue by itself!

Changing the tune being played is very simple. Choose a song in which all the notes are the same length, e.g., quarter notes. In line 24000, change the 1639 to (1632+the number of notes in your tune-1), replace the data in lines 25000 and 26000 with the notes for your tune, and change the 8 in line 22000 with the number of notes in your tune. Remember, the tune will play over and over, so pick something which sounds good on repetition.

The routine presented here can be ended by a power-off, power-on sequence, or by a SYSTEM RESET. A third method, probably more useful for use in a program, is this:

```
POKE 1562,104:POKE 1544,98:POKE 1546,228:X=USR(
1542):SOUND 0,0,0,0
```

This is a simple demonstration of the use of the vertical blank interrupt routine. There are many other potential uses for this approach, such as background music for another program, checking for keyboard or joystick input during a program, or implementation of multitasking. It is perfectly feasible to have two completely separate programs running "simultaneously", but the programming for this gets fairly complicated. One program would run in real time, and the other during the vertical blank interrupt routines. Play around with the ideas presented here, and learn all about simultaneous processing. But most of all, HAVE FUN!! □

FIGURE ONE

```
0000          10          *=   $0600
00C0          20 COUNT1   =    $00C0
0224          30 VVBLKD   =    $0224
00C2          40 COUNT2   =    $00C2
E45C          50 SETVBV   =    $E45C
0660          60 MUSIC    =    $0660
E462          70 RETURN   =    $E462
D200          80 SND      =    $D200
D201          90 VOL      =    $D201
              0100 ;
              0110 ;   PLA FOR BASIC ACCESS
              0120 ;
0600 68       0130        PLA
              0140 ;
              0150 ;   INITIALIZE COUNTERS TO ZERO
              0160 ;
0601 A900     0170        LDA  #0
0603 85C0     0180        STA  COUNT1
0605 85C2     0190        STA  COUNT2
              0200 ;
              0210 ;   NOW RESET DEFERRED VECTOR
              0220 ;
0607 A020     0230        LDY  #$20
0609 A206     0240        LDX  #$06
060B A907     0250        LDA  #07
060D 205CE4   0260        JSR  SETVBV
0610 60       0270        RTS
              0280 ;
              0290 ;   MAIN INTERRUPT ROUTINE
              0300 ;
0611          0310        *=   $0620
0620 E6C0     0320        INC  COUNT1
0622 A6C0     0330        LDX  COUNT1
0624 E00C     0340        CPX  #12        ;TIME TO STOP NOTE?
0626 9005     0350        BCC  K1         ;NO
0628 A900     0360        LDA  #0         ;YES, SO STOP IT
062A 8D01D2   0370        STA  VOL
062D E00F     0380 K1     CPX  #15        ;15/60 SECONDS GONE?
062F B003     0390        BCS  PLAY       ;YES, PLAY NEXT NOTE
0631 4C62E4   0400        JMP  RETURN     ;NO, END INTERRUPT
0634 A900     0410 PLAY   LDA  #0
0636 85C0     0420        STA  COUNT1     ;RESET COUNT1 TO ZERO
0638 A6C2     0430        LDX  COUNT2     ;GET NOTE TO PLAY
063A BD6006   0440        LDA  MUSIC,X    ;LOOK IT UP
063D 8D00D2   0450        STA  SND        ;SET IT'S FREQUENCY
0640 A9A6     0460        LDA  #$A6
0642 8D01D2   0470        STA  VOL        ;SET PURE NOTE,VOLUME=6
0645 E6C2     0480        INC  COUNT2     ;SET UP NEXT NOTE
0647 A6C2     0490        LDX  COUNT2
0649 E008     0500        CPX  #8         ;ALL NOTES USED UP?
064B 9004     0510        BCC  DONE       ;NO
064D A900     0520        LDA  #0         ;YES, START OVER AGAIN
064F 85C2     0530        STA  COUNT2
0651 4C62E4   0540 DONE   JMP  RETURN     ;ALL DONE
              0550 ;
              0560 ;   TABLE OF MUSICAL NOTES
              0570 ;
0654          0580        *=   $0660
0660 F3       0590        .BYTE 243,243,217,243,204,243,217,243
0661 F3
0662 D9
0663 F3
0664 CC
0665 F3
0666 D9
0667 F3
```

FIGURE TWO

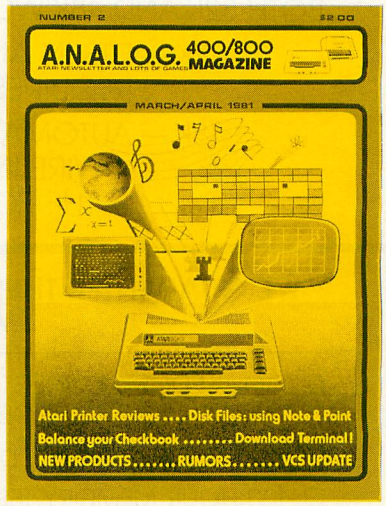
```
8000 REM ********************  FIRST,
 WE'LL POKE IN THE LINES FROM 130-270
OF THE ASSEMBLY LISTING **********
9000 RESTORE 10000:FOR I=1536 TO 1552:
READ A:POKE I,A:NEXT I
10000 DATA 104,169,0,133,192,133
11000 DATA 194,160,32,162,6,169
12000 DATA 7,32,92,228,96
13000 REM ****************************
 THEN WE'LL POKE IN THE MAIN ROUTINE
**************************************
14000 RESTORE 15000:FOR I=1568 TO 1619
:READ A:POKE I,A:NEXT I
15000 DATA 230,192,166,192
16000 DATA 224,12,144,5,169,0
17000 DATA 141,1,210,224,15,176
18000 DATA 3,76,98,228,169,0
19000 DATA 133,192,166,194,189,96
20000 DATA 6,141,0,210,169,166
21000 DATA 141,1,210,230,194,166
22000 DATA 194,224,8,144,4,169
23000 DATA 0,133,194,76,98,228
23500 REM ****************************
FINALLY, WE'LL POKE IN THE TABLE OF NO
TES **************************************
24000 RESTORE 25000:FOR I=1632 TO 1639
:READ A:POKE I,A:NEXT I
25000 DATA 243,243,217,243,204,243
26000 DATA 217,243
26500 REM ****************************
NOW WE'LL RUN THE ROUTINE!!**********
27000 X=USR(1536)
```

BACK ISSUES
of A.N.A.L.O.G. 400/800 Magazine are now available!

ISSUE NO. 1

Become a Star Commander
Five Star Treks
Music Composer
Maze Rider

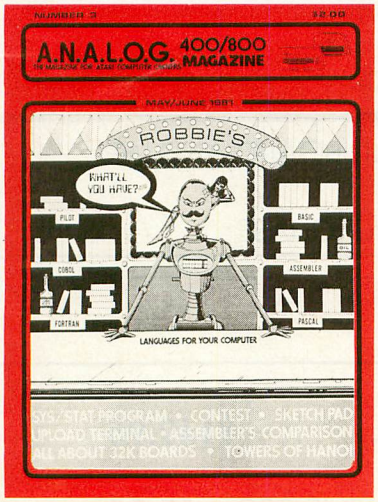
$3.00



ISSUE NO. 2

Printer Reviews
Disk Files
Download Terminal
Balance Your Checkbook

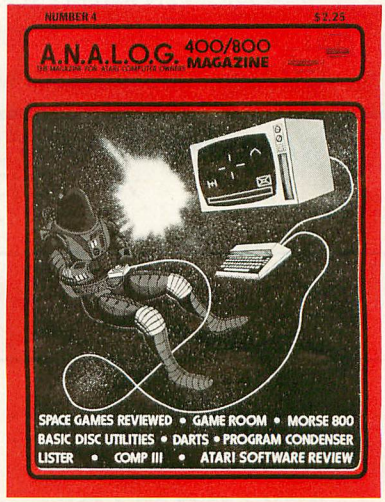
$3.00



ISSUE NO. 3

Sketch Pad
Upload Terminal
Assembler's Comparison
All About 32K Boards

$3.00



ISSUE NO. 4

Basic Disk Utilities
Comp III
Space Games Reviewed
Morse 800

$3.00

Each issue of A.N.A.L.O.G. includes VCS updates, software reviews, new products information and other articles. A.N.A.L.O.G. is the first magazine devoted solely to the ATARI home computers. We talk to our readers, not down to them.

Please note that the advertisements in back issues of A.N.A.L.O.G. are the originals, and the information contained therein may no longer be accurate (pricing, company addresses or phone numbers).

Mastercard and Visa accepted.

ANALOG

P.O. Box 23
Worcester MA
01603
(617) 892-9230

AUSTIN FRANKLIN ASSOCIATES ANNOUNCES

32K - ATARI® 400/800 - 48K

$94.95

2-YEAR FULL WARRANTY

HIGHEST QUALITY AND RELIABILITY POSSIBLE

$124.95

(WITH 16K IN TRADE)

BOTH 32K AND 48K WORK IN THE 400 AND 800 - FULLY ILLUSTRATED INSTRUCTIONS

32K — FITS IN ANY CONFIGURATION ANY OTHER BOARD WILL
REQUIRES NO MODIFICATIONS TO THE 400

48K — AUTOMATIC MEMORY DE-SELECTION WHEN CARTRIDGE USED
4-WIRE INSTALLATION TO 400 - NONE TO 800
¼ THE POWER CONSUMPTION OF OTHER BOARDS

WE GUARANTEE THESE TO BE THE FINEST 32K AND 48K BOARDS AVAILABLE AT ANY PRICE — AND HERE'S WHY:

- ☐ GOLD EDGE CONNECTOR — BEST CONNECTION TO BACK PLANE
  TAKES FULL ADVANTAGE OF ATARI'S BUSS STRUCTURE — INSURES TOTAL COMPATABILITY
- ☐ COMPUTER GENERATED PC BOARDS — MOST ACCURATE
- ☐ 100 HOUR MEMORY TEST BURN-IN — GUARANTEES PERFECTLY OPERATING BOARDS
- ☐ ALL IC'S FULLY DECOUPLED (MANY CAPACITORS) — INSURES GLITCH-FREE OPERATIONS AND NO SCREEN INTERFERENCE
- ☐ NOT SOCKETED — SOCKETS ARE UNRELIABLE AND COST MORE
- ☐ PARTS PURCHASED IN MASS QUANTITY — SAVINGS REFLECTED IN PRICE

WARRANTY: 2-YEAR FULL REPLACEMENT — JUST RETURN YOUR BOARD AND WE WILL SEND YOU A NEW ONE WITHIN 10 DAYS.

10-DAY FULL REFUND FROM DATE OF PURCHASE

THESE BOARDS DO EVERYTHING THE COMPETITORS WILL DO — BUT COST YOU MUCH LESS.

48K (WITHOUT 16K TRADE) **$149.95**

32K LOOPBACK — ALLOWS 32K TO BE USED ALONE IN THE 800 . . **$5.00**

48K LOOPBACK — ALLOWS 48K TO BE USED ALONE IN THE 800 . . **$5.00**

SEND CHECK OR MONEY ORDER TO:

AUSTIN FRANKLIN ASSOCIATES
43 GROVE STREET, AYER, MA 01432

FOR C.O.D., MASTERCARD OR VISA, CALL
(617) 772-0352

DEALER INQUIRIES AVAILABLE

CLUB DISCOUNTS AVAILABLE

STATIC DAMAGE IS NOT COVERED BY WARRANTY.

The Program Doctors At The C.E.S.

by The Program Doctors

The Summer International Consumer Electronics Show is now history, but the excitement is just beginning. This year there was greater emphasis on computers and related products than ever before. Come with us now as we re-live our journey and share with you many of the show's highlights.

Before even checking into our hotel room we had to start hitting the CES hot spots. First stop was the Knickerbocker Hotel to see what K-Byte has in store for us in the coming months and we were treated to a showing of their three new ROM cartridges. In K-RAZY KRITTERS (available in July) you must eliminate eight columns of alien Kritters who swoop at you unmercifully. Your reward for this is the next of ten levels, each one sporting ten colors simultaneously. K-STAR PATROL (available in fall) puts you in command of a fleet of ships powered with lasers and hydrogen fueled shields. You scroll across the planet's surface destroying attack ships, dodging a low-level avoidance system and trying to blow away the Inter-Galactic Leech that attempts to envelop your ship. K-RAZY ANTIKS (available in fall) is a computer game par excellance. Invading enemy ants, anteaters, torrential rains and enemy eggs will challenge everyone's skill. This game offers you a choice of six different playfields each with ninety-nine (!) difficulty levels. This one is definitely **not** "just another maze game".

K-Byte was also showing their smart communications package in a ROM pack, K-COM I. A new utility package K-DATE ORGANIZER (available in Sept.) is billed as the "calendar of the future". A seven year personal secretary, it will hold all important dates for up to four people.

K-Byte is demonstrating tremendous support for the ATARI in all user areas.

Leaving our car downtown, we grabbed a cab (or did it grab us?) and sped to McCormick Place. The excitement was intense as we joined the thousands of people exploring the wonders of this electronics playground.

Of course, we had to check out the ATARI booth first. It was massive, incorporating the computers, the VCS, and their new 5200 Super Game System. Their new software for the computer includes two self-improvement programs and two programs for children aged three to six. ATARI SPEED READING (available in fall) combines computer programs and the Speedway reading course. It will consist of five cassettes and a two-hundred page workbook. ATARI MUSIC TUTOR I teaches music fundamentals such as note reading, key signatures, and the C Major scale. It will incorporate tutorials, drills, tests, and games. It will be available late this year on cassette and disk. Two programs to teach essential pre-reading skills will also be released late this year. JUGGLES' HOUSE teaches the concepts of upper, lower, inside, and outside, while JUGGLES' RAINBOW will teach above, below, right, and left. Detachable keyboard strips will integrate the computer with screen.

ATARI is also coming out with a new communications kit. It will include the new ATARI 835 DIRECT CONNECT MODEM (DCM), the new TELELINK II Cartridge and information on the use of Compuserve, The Source, and the Dow Jones News Retrieval Service. COMMUNICATOR II also includes one free hour of use on each of these three services. The DCM will only be available in the kit, however TELELINK II may be purchased separately. The 830 Acoustic Modem and Telelink I will still be available.

Finally, the best news coming from ATARI was the price decreases. Asteroids, Chess, Missile Command, Space Invaders, Super Breakout, Star Raiders and Music Composer all will have new retail prices from 22% to 33% lower. The 400 was dropped in price not once but twice during the show to a new suggested retail of $299.95.

We then did a quick once over of the rest of the show. (Quick? It took us about two hours) and went to check in to our hotel. We met some friends for dinner and then spent the rest of the night mapping our strategy for the next day.

Monday morning, first thing, we **had** to make our annual trek through the adult films section. It was all there, folks, all the "art" films and their stars in the flesh (Boy, what flesh!!). What a way to dispel those Monday morning blues. But we had work to do, and Marcia led the way as she herded us through this wonderful paradise so we could get on with the tasks of the day.

Thorn EMI was our next stop. Thorn is fully supporting the ATARI 800/400 systems with programs on cartridge and cassette, and will soon have a library of over twenty pieces of software. If you are familiar with their Darts and Pool programs, you know the high standards this company has set

for itself. New cartridge releases include SUBMARINE COMMANDER, in which you must destroy all enemy merchant shipping in the Mediterranean. You are in full command of the sonar equipped submarine complete with a sophisticated instrument panel, diving ability, and torpedoes. This game is for one player with nine variations.

If submarines aren't your thing you can fly your own plane in JUMBO JET PILOT, a flight simulator for one player with ten variations. You have full flight instrumentation and views through the cockpit windows. Take off, navigate to your destination and land safely. The computer will then rate your flying performance.

Two sports games are also coming out on cartridge. SOCCER will feature a horizontally scrolling screen, multi-skill levels, the ability to play against the computer or a friend, or team up with a friend and battle the computer together. KICKBACK is a football style game for one or two players with ten variations and five skill levels.

Releases on cassette will include HOME FINANCIAL MANAGEMENT, SNOOKER & BILLIARDS, SUPER CUBES, a Rubik's Cube game with three hundred variations, and four jigsaw puzzle packages. These are BRITISH HERITAGE JIGSAW PUZZLES, Volume I and II, and EUROPEAN SCENE JIGSAW PUZZLES, Volume I and II, each volume containing two different pictures. Two programs based on nursery rhyme themes for pre-schoolers to adults (depending on which of the forty-eight variations you choose) will incorporate the old puzzle of dividing a picture into squares and jumbling it, which you must then slide back into place. FIGURE FUN teaches both multiplication and division in game form with over one thousand variations. OWARI AND BULL & COW and COMPUTE 4 & REVERSI are adaptations of the board games.

These new releases along with their vast financial empire should catapult Thorn EMI into a major software producer.

At one time or another in the course of your education, you probably had some contact with Milliken Publishing Co. They are one of the leading publishers of supplemental teaching aids for kindergarten through twelfth grade. In 1976 they set up a division to develop, publish and market educational software for schools. Now they have established *EDUFUN* to produce their high quality programs in game form to educate at home and in school. The first group of programs introduced here at the show is the MATHFUN! series, consisting of twelve different games in six packages covering a wide span of age groups, to be released in September. Each package will be available in both a home and a school version and will include instructions for children, parents, and teachers, reward stickers, and ideas for supplemental activities to reinforce the concepts taught.

Additional games in a variety of areas will be released periodically during the coming year (WORDFUN!, SPELLINGFUN!, HEALTHFUN! etc.) and we will be looking to Milliken to fill the void in educational software for the ATARI.

On we went to see what was brewing at Synapse Software. Ihor Wolosenko, Synapse's president, gave us a demonstration of three new arcade games, SLIME, SHAMUS, and NAUTILUS. NAUTILUS, written by Mike Potter, is a very unique submarine game with twin independent screen displays that scroll in opposite directions. Synapse is also releasing three utility packages and a tutorial. Look forward to FILE MANAGER 800+, an enhanced and expanded version of their popular FILE MANAGER 800, DISK MANAGER, to maintain an inventory of all the programs in your library and print labels for your disks, SYNASSEMBLER, a new assembler/editor, and PAGE 6, a hands-on machine language tutorial and BASIC enhancement disk.

As we left Synapse, we ran into a representative from Arcade Plus. While they did not have anything for us to see, we did get some information. To follow their successful Ghost Hunter, Arcade Plus is releasing three new games on cassette and disk, NIGHT RALLY, ARCADE BASEBALL, and ARCADE PRO FOOTBALL.

Up the aisle we went and discovered any game player's dream. A PAC-MAN PHONE!! It looks just like the little guy with speakers and buttons inside his mouth. American Telecommunications Corp. has really got a cutie with this one!

We stopped to say hello at the Computer Magic booth (KAYOS) and learned that they are adding three new games to their product line, the first of which will be POGOMAN.

You all know about the infamous joystick controller that ATARI makes (how many have you had to replace already?). Help is on the way, because several companies are producing new controller peripherals. Wico, the world's largest designer and manufacturer of commercial arcade control devices, has designed a whole line, called COMMAND CONTROL. They will have three different joysticks and a trackball on the market within the next few months. Game Tech has a joystick that is available now and the Roklan Corp. and ATARI are both coming out with trackballs.

By now we've both got aching feet (you wouldn't believe how big this place is!) but we've got to make one more stop today. On we pushed to the impressive EPYX display. Talking with Joyce Lane, their advertising manager, we learned that four new fantasy adventures will be out on disk in July and August: CRYPT OF THE UNDEAD, THE

NIGHTMARE, KING ARTHUR'S HEIR, and ESCAPE FROM VULCAN'S ISLE. In addition, EPYX has announced the release of six new game cartridges. ALIEN GARDEN (available in September) is a strategy action game that challenges you to survive in a world whose vegetation can build your stamina or wipe you out forever. You must decide whether to eat or diet and the faster you do the more points you get.

Remember on those TV variety programs that used to be on every other week there was invariably a juggler who spun plates on sticks or his nose or elbows or hands or feet? Well, you will get your chance to try it in October when PLATTERMANIA hits the stores. FISHES (available in December) is your chance to lead a school of fish through treacherous waters. Feed them and your fish will grow, but watch out because snakes, hooks, jellyfish and other undesirables want you for dinner. In FAR PROTECTOR (available in November) you must guard the earth from final annihilation. Orbiting in a satellite, you watch the skies from any of six cameras. Besides missiles from earth, you must be on the lookout for enemy missiles in space. SOLDIERS OF SORCERY (available in November) is a multi-player fantasy role playing game in the EPYX tradition that is different every time you play. And finally for the kiddies, is QWERTY BIRD (available in December), an action game designed to teach keyboard recognition. Releasing ten games in six months is a tremendous undertaking (not to mention their software for all the other systems that EPYX supports) and we wish them good luck.

"Get me out of this zoo!!", we screamed simultaneously, "Enough!!" Barely able to walk we made our way to a taxi, but our day was not over yet. Time to mix some business with pleasure. The business part is for our accountant, because having cocktails with Jerry White, the ATARI computer's most prolific author, was a totally pleasurable experience. This man has no less than ten companies that market his software, and he is extremely dedicated to producing quality software. He told us about his two newest games. TRIVIA TREK, just released from Swiftware, will test your knowledge in fifty different categories and if that's not enough, you can make more of your own. You can play 5-card draw against seven other players in POKER TOURNEY to be released soon by Artworx.

After dinner, we fell into our hotel room and collapsed with exhaustion. All too soon the phone rang, and the hotel operator informed us it was Tuesday morning and time to get up.

We hurried through breakfast because we had an appointment at the offices of one of the new powerhouses in ATARI software. The Roklan Co. is providing full support through games, utilities and hardware.

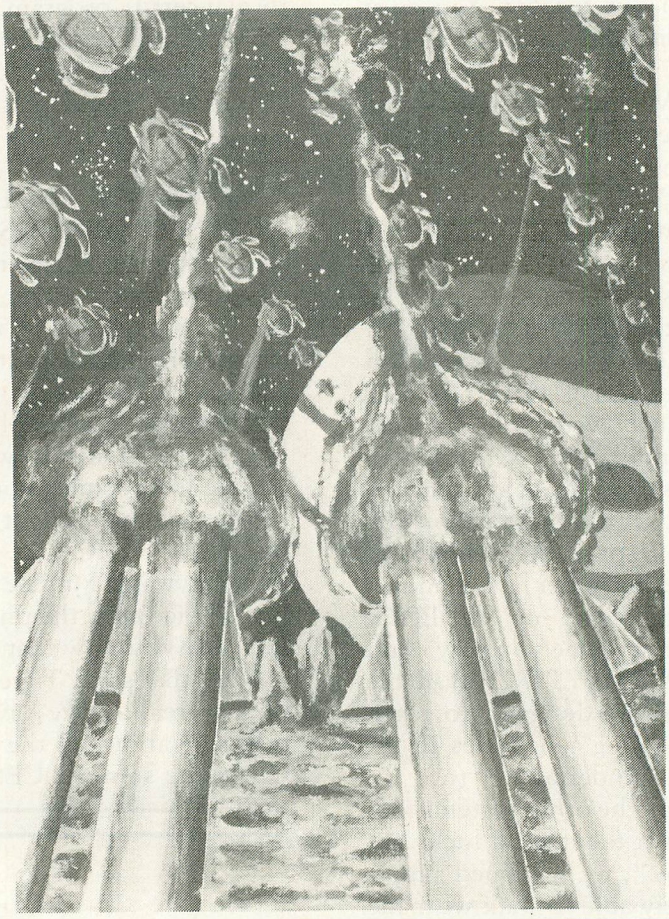
DELUXE INVADERS, WIZARD OF WOR, and GORF are the first of a series of arcade games developed with the cooperation of the original arcade manufacturers. AIDE is an absolute disk editor which permits inspection or modification of any sector on a disk. It also includes a Copy/verify-format certify program. Roklan is also releasing four business application programs. FINPAC, a package for all financial calculations, offers the flexibility to change any factor, such as interest rates, without having to input all the other factors again. MORTGAGE ANALYSIS, PROPERTY INVESTMENT ANALYSIS, and REAL ESTATE PROPERTY APPRAISING, will give you all types of information regarding payment formulas, resale values, income analysis, depreciation and changes in net income. The 6502 SIMULATOR is for programmers who want to understand the results of assembly language instructions in detail. TELE-COM, a smart phone modem package has just been released and as we mentioned their trackball will be forthcoming soon.

Oh, we almost forgot the most interesting thing we discovered at Roklan. Our prolific editor (one of them anyway), Mr. Lee Pappas! He will no longer be just a voice over the phone to us, and we can truthfully tell you now that he really does indeed exist.

Well that's it!! Time to start our journey home back to reality. We have to write an article for the magazine about all the new and exciting things we saw at the summer CES show... □

MOONBASE IO

• the battle for the moons of Jupiter •



Blast your way through the alien mine fields! Defend the moonbases from an attacking alien armada! It won't be easy. To win you will need to mount a heroic assault on the alien mother ship.

Moonbase Io combines three exciting arcade adventures in one exciting game. The machine-language program by John Konopa uses advanced graphics and sound effects. Action is fast and exciting—varying levels of skill required to go from one part of the game to the next.

BEYOND SOFTWARE's arcade-adventures are the next generation in computer games for the ATARI™ computer. A voice-activated program will help you meet and overcome the challenge—it may be an alien invasion, a fiendish murderer preying on a country village or a treasure trove buried deep in the shark-infested sea.

BEYOND SOFTWARE

Moonbase Io is available at fine computer dealers. Or, directly from PDI for $29.95 plus $2.00 shipping and handling.

Requires 24K ATARI™ computer with disk and cassette.
Cassette version available soon.

PDI

Program Design, Inc., 11 Idar Court, Greenwich, CT. 06830 203-661-8799

ATARI Basic Meets Complex Data Structures

by Raymond T. Tillman

One topic long neglected on the ATARI Computer is its way of handling strings. Everyone has been so busy with ATARI's graphics, they haven't noticed the mysteries hiding in ATARI BASIC's strange way of using strings.

Strings are just that in ATARI BASIC — strings — no arrays. That seeming limitation, however, pales once you learn that one string can be up to 32, 768 characters long. That means that you can define your own string arrays of whatever length you need. (Most versions of BASIC arbitrarily define a string to a maximum of up to 255 characters.)

There are no MID$, LEFT$, or RIGHT$ commands. ATARI BASIC works more directly: VA$(3,5) for MID$(VA$,3,5). Strings can be concatenated together in two ways: VA$(LEN(VA$)+1)=VB$ or VA$(8)=VB$ (if VA$ is 8 characters). Other BASICs use the "+" sign and add strings together: C$=A$+B$. There are other interesting features of ATARI's BASIC string handling, like the STR$, VAL$, and CHR$ functions used to convert between numbers and characters.

These variances from other versions of BASIC allow one to create with ATARI BASIC, some rather complex data structures in a very direct manner. The program which accompanies this article creates such a structure — a linked list.

What is a linked list? A linked list is a linear data structure which includes data records and addresses, or link pointers, which show where the next logically adjacent record is, whether it is physically adjacent to the current record or not. These pointers allow referencing of structures in a uniform way, regardless of the actual organization of the data.

The program included in this article builds a linked list consisting of names, social security numbers and phone numbers in three fields. A pointer to the next record is also included. The resulting record is 62 characters long. As written, up to 310 records can be created in a 48K ATARI (current program dimensions 200 62-character records.)

In order to keep track of the values of the pointers and to locate the position where any new records are to be placed in the data structure, an array of the same size as the number of records was created. This array, FRSTCK (a freestack) contains the INDEXES of the starting position of each record in the data string. It is accessed strictly as a stack, using a top index pointer. If a record is added, the INDEX value at the top of the stack is used to place the next record in the string. The stack is then popped. If a record is deleted, the stack is incremented and the record's pointer is restored to the new top of the stack. As a warning, each record must be of fixed length (here 62 characters), so the correct INDEX values can be computed. □

PROGRAM DESCRIPTION

Lines 20-40 initializes module calls (for named GOSUBs), and dimensions strings.

Lines 1000-2020 are the main program: Line 1001 opens a file, in this case the screen editor, but it could be opened to "P:" printer, or some other file. Lines 1010-1040 initializes the freestack pointer array. This could be modified to string form, reducing memory requirement. Line 1050 calls CREATE to initialize the data structure with the starting data. Line 1060 reads the command from data statements.

Lines 1070-1130 direct the program to the appropriate subroutine using a form of a case statement.

Lines 50-99 converts a numerical pointer value into character data. This is necessary because BASIC requires homogenious elements in a character string (only characters). Note method of overlaying the 4-character field.

Lines 100-190 performs searches be pointer. It converts the pointers to numerical indexes and points to that position in the string.

Lines 300-390 finds a record to be deleted, and if found, moves other pointers, returns associated pointer to a "free pointer stack", and deletes the record.

Lines 400-460 retrieves data, by calling the search subroutine. If the data is found, it is printed.

Lines 500-590 finds data, using the search subroutine, and if found, modifies it.

Lines 600-691 inserts new data into the linked list using the "freestack" and search subroutine.

Line 700-790 Prints the data in alpha-numeric order. This simulates storage to a file, such a disk.

Lines 800-899 read the data (lines 3000-4150) into the original list. These are read in sequentially. Program modifications could allow data to be input from disk or cassette files here. □

```
1 REM ************************************************************************************************************
2 REM *                                                                                                          *
3 REM *                Linked List Implementation in ATARI BASIC                                                 *
4 REM *                COPYRIGHT 1981 BY Raymond T. Tillman                                                      *
5 REM *                   all rights reserved                                                                    *
6 REM *                                                                                                          *
7 REM *     This program creates a linked list data structure, complete with pointers.  The file organization    *
8 REM * is an un-ordered character string list with embedded pointers.  The effect of the pointers is to access  *
9 REM * the records in the list in alphanumeric order.  The memory for the record structure is reserved at RUN   *
10 REM * but allocated as needed.  The garbage collection returns deleted records to the free stack.              *
11 REM *     The record format is as follows:                                                                    *
12 REM *          1.  INREC$                                                                                      *
13 REM *              2.  NAME$  CHAR(80)                                                                         *
14 REM *              2.  SSN$   CHAR(9)                                                                          *
15 REM *              2.  PHN$   CHAR(8)                                                                          *
16 REM *              2.  PNTR$  CHAR(4)                                                                          *
17 REM *      The freestack is simulated in an array built as a reverse order stack.                             *
18 REM ************************************************************************************************************
20 DIM LLST$(200*62),FRSTCK(200),PNTR$(4)
25 DIM INREC$(58),NAME$(40),SSN$(9),PHN$(8),FILL$(1),CMND$(1)
30 PNTBLD=50:STORE=700:SEARCH=100:CREATE=800:MAIN=1000:INSERT=600:MODIFY=500:RETRIEVE=400:DELETE=300:YES=1:NO=0
40 TOP=0:GOTO MAIN
50 REM PNTBLD MODULE
51 REM ------------------------------------------------------
52 REM This module converts the pointer to a 4-character string.
53 REM ------------------------------------------------------
60 LNPTR=LEN(STR$(FRSTCK(TOP))):PNTR$="0000"
70 PNTR$(5-LNPTR,4)=STR$(FRSTCK(TOP))
99 RETURN
100 REM SEARCH MODULE
101 REM ------------------------------------------------------------------
102 REM This module searches the list.  If an element is found,
103 REM a flag is set.  This module is used to find records and insertion points.
104 REM ------------------------------------------------------------------
110 CUR=HEAD
120 IF CUR=0 THEN FOUND=NO:GOTO 190
130 IF LLST$(CUR,CUR+39)<NAME$(1,40) THEN PREV=CUR:CUR=VAL(LLST$(CUR+57,CUR+61)):GOTO 120
140 IF LLST$(CUR,CUR+39)=NAME$ THEN FOUND=YES:GOTO 190
150 FOUND=NO
190 RETURN
300 REM DELETE MODULE
301 REM ------------------------------------------------------------------
302 REM This module deletes records based on the NAME given.  It calls SEARCH
303 REM to find the proper record.  If the record is not found, a message is
304 REM generated.
305 REM ------------------------------------------------------------------
310 CUR=HEAD
320 READ NAME$:NAME$(LEN(NAME$)+1)=BLNK$
330 GOSUB SEARCH
340 IF  NOT FOUND THEN ? #3;" We are sorry, we didn't find a record for ";NAME$:GOTO 390
350 TOP=TOP+1:? #3;" RECORD DELETED FOR ";NAME$
```

```
354 IF CUR=HEAD THEN HHEAD=HEAD
355 IF CUR=HEAD THEN HEAD=VAL(LLST$(CUR+58,CUR+61)):FRSTCK(TOP)=CUR:FOR I=1 TO 62:LLST$(CUR-1+I,CUR-1+I)=" ":NEXT I
356 IF CUR=HHEAD THEN 380
360 LLST$(PREV+58,PREV+61)=LLST$(CUR+58,CUR+61)
370 FOR I=1 TO 62:LLST$(CUR-1+I,CUR-1+I)=" ":NEXT I
375 FRSTCK(TOP)=CUR
390 RETURN
400 REM RETRIEVE MODULE
401 REM ------------------------------------------------------------------------
402 REM This module retrieves data based on an input name.
403 REM It uses the SEARCH module and prints a message for each record.
404 REM ------------------------------------------------------------------------
410 CUR=HEAD:READ NAME$
420 NAME$(LEN(NAME$)+1)=BLNK$
430 GOSUB SEARCH
440 IF  NOT FOUND THEN ? #3,,"We are sorry.  We could not find a record for ";NAME$:GOTO 460
445 ? #3,,"RECORD FOUND"
450 ? #3;"";LLST$(CUR,CUR+39);"";LLST$(CUR+40,CUR+42);"-";LLST$(CUR+43,CUR+44);"-";LLST$(CUR+45,CUR+48);"";
455 ? #3;LLST$(CUR+49,CUR+56)
460 RETURN
500 REM MODIFY MODULE
501 REM ------------------------------------------------------------------------
502 REM This module modifies specified elements in a record.  The input data
503 REM gives the name to search for.  When the name is located, the element
504 REM to be changed is identified and the appropriate new data is read in
505 REM and replaces former information.  If the record is not found, a
506 REM message is printed.
507 REM ------------------------------------------------------------------------
510 CUR=HEAD
520 READ NAME$:NAME$(LEN(NAME$)+1)=BLNK$
530 GOSUB SEARCH
540 IF LEN(SSN$)<>9 OR LEN(PHN$)<>8 THEN ? #3,"ERROR":STOP
545 ? #3;" RECORD MODIFIED "
550 READ NAME$
560 IF NAME$="NAME" THEN READ NAME$:NAME$(LEN(NAME$)+1)=BLNK$:LLST$(CUR,CUR+39)=NAME$
570 IF NAME$="SSN" THEN READ SSN$:LLST$(CUR+40,CUR+48)=SSN$
580 IF NAME$="PHN" THEN READ PHN$:LLST$(CUR+49,CUR+56)=PHN$
585 ? #3;"";LLST$(CUR,CUR+39);"";LLST$(CUR+40,CUR+42);"-";LLST$(CUR+43,CUR+44);"-";LLST$(CUR+45,CUR+48);"";
587 ? #3;LLST$(CUR+49,CUR+56)
590 RETURN
600 REM INSERT MODULE
601 REM ------------------------------------------------------------------------
602 REM This module inserts new data at the beginning, end, or in the middle
603 REM of the list.  The pointer is taken from the freestack and the record
604 REM is loaded into the list or tacked on to the end of it as directed by
605 REM the pointer value in the freestack.
606 REM ------------------------------------------------------------------------
609 ? #3;" RECORD TO BE INSERTED FOR ";
610 CUR=HEAD
620 READ NAME$,SSN$,PHN$
625 ? #3;NAME$
630 NAME$(LEN(NAME$)+1)=BLNK$
650 INREC$=NAME$:INREC$(41)=SSN$:INREC$(50)=PHN$:INREC$(58)=" "
660 GOSUB SEARCH
665 IF CUR=HEAD THEN HEAD=FRSTCK(TOP):GOSUB PNTBLD:LLST$(HEAD,HEAD+57)=INREC$:FRSTCK(TOP)=CUR:GOSUB PNTBLD:A=1
667 IF A=1 THEN LLST$(HEAD+58,HEAD+61)=PNTR$:A=0:TOP=TOP-1:GOTO 690
670 IF CUR=0 THEN CUR=FRSTCK(TOP):GOSUB PNTBLD:LLST$(PREV+58,PREV+61)=PNTR$:LLST$(CUR,CUR+57)=INREC$:A=1
671 IF A=1 THEN LLST$(CUR+58,CUR+61)="0000":TOP=TOP-1:A=0:GOTO 690
```

```
680 LLST$(FRSTCK(TOP),FRSTCK(TOP)+57)=INREC$:LLST$(FRSTCK(TOP)+58,FRSTCK(TOP)+61)=LLST$(PREV+58,PREV+61)
685 GOSUB PNTBLD:LLST$(PREV+58,PREV+61)=PNTR$:TOP=TOP-1
691 RETURN
700 REM STORE MODULE
701 REM ------------------------------------------------------------------------
702 REM This module simulates printing the file to storage device.  Actually,
703 REM it prints the data in ordered sequence to a device (printer).  Thus,
704 REM STORE is used to print the file at various points in the program.
705 REM The last time it is called, it is in simulation of printing to a
706 REM storage device.
707 REM ------------------------------------------------------------------------
710 CUR=HEAD
720 ? #3,,"DATA FILE":? #3
725 ? #3,"         NAME";"SOC.SEC.#";" PHONE":? #3
730 IF CUR()0 THEN ? #3;"";LLST$(CUR,CUR+39);"";LLST$(CUR+40,CUR+42);"-";LLST$(CUR+43,CUR+44);"-";:GOTO 740
735 IF CUR=0 THEN GOTO 790
740 ? #3;LLST$(CUR+45,CUR+48);"";LLST$(CUR+49,CUR+57):CUR=VAL(LLST$(CUR+58,CUR+61)):GOTO 730
750 ? #3:? #3,,"FILE DUMP":? #3:? #3;LLST$
770 ? #3;LLST$
790 RETURN
800 REM CREATE LIST MODULE
801 REM ------------------------------------------------------------------------
802 REM This module reads data from the data file in a simulation of reading
803 REM from an external device.  It loads the linked list sequentially, stor-
804 REM ing a pointer taken from the freestack for each record.  This file
805 REM serves as the base file for subsequent manipulations.  Only five
806 REM records are simulated, but the only real limitation is the size of the
807 REM data structure and the total memory of the machine.
808 REM ------------------------------------------------------------------------
810 HEAD=FRSTCK(TOP)
820 J=J+1:READ NAME$,SSN$,PHN$
830 NAME$(LEN(NAME$)+1)=BLNK$
840 INREC$=NAME$:INREC$(41)=SSN$:INREC$(50)=PHN$:INREC$(58)=" "
870 LLST$(FRSTCK(TOP))=INREC$:LLST$(LEN(LLST$)+1)="0000"
873 IF LEN(LLST$)=62 THEN 878
874 GOSUB PNTBLD
875 LLST$(FRSTCK(TOP)-4,FRSTCK(TOP)-1)=PNTR$
878 TOP=TOP-1
880 IF J<5 THEN GOTO 820
890 RETURN
990 REM ------------------------------------------------------------------------
991 REM Following is the main program module.  It acts as a control program.
992 REM First, it creates the file.  Then, while there is data to process, it
993 REM reads commands and calls subprograms for execution of the commands.
994 REM Finally, it simulates the storage to IO device of the data file.
999 REM ------------------------------------------------------------------------
1000 REM MAIN MODULE
1001 OPEN #3,8,0,"E":SETCOLOR 2,11,0
1002 DIM BLNK$(40):BLNK$="                                        "
1006 ? #3:? #3
1007 ? #3;CHR$(12)
1010 FOR I=200 TO 1 STEP -1
1020 TOP=TOP+1
1030 FRSTCK(I)=TOP*62-61
1040 NEXT I
1050 GOSUB CREATE
1060 TRAP 2000:? #3:READ CMND$
1070 NUM=ASC(CMND$)-64
```

SPHERE TODAY, GONE TOMORROW GRAPHICS DEMO TRY IT!

```
8 SIZE=90:REM ***RADIU
S***
9 CX=160:CY=96:REM **C
ENTER**
10 DEG :TIME=1
20 GRAPHICS 24:SETCOLO
R 2,0,8:SETCOLOR 1,0,0
:COLOR 1
25 PLOT CX+SIZE,CY:REM
 ***START***
30 FOR Y=90 TO 0 STEP
-12
40 FOR X=0 TO 360 STEP
 12
50 IF TIME=1 THEN X2=C
X+SIZE*COS(X):Y2=CY-(S
IZE*SIN(X)*SIN(Y)):GOT
O 60
55 X2=CX-(SIZE*SIN(X)*
SIN(Y)):Y2=CY+SIZE*COS
(X)
60 DRAWTO X2,Y2:NEXT X
:NEXT Y
90 TIME=TIME+1:IF TIME
=2 THEN PLOT CX,CY+SIZ
E:GOTO 30
100 SIZE=20+RND(1)*30:
CX=SIZE+1+(RND(1)*(318
-(SIZE*2))):CY=SIZE+1+
(RND(1)*(190-(SIZE*2))
):GOSUB 1000:TIME=1:GO
TO 25
910 REM *** ERASE HIDD
EN LINES ***
1000 COLOR 0:FOR X=0 T
O 90 STEP 0.5
1010 X2=SIZE*COS(X):Y2
=SIZE*SIN(X)
1020 PLOT CX+X2,CY+Y2:
DRAWTO CX-X2,CY+Y2:PLO
T CX+X2,CY-Y2:DRAWTO C
X-X2,CY-Y2:NEXT X:COLO
R 1:RETURN
```

●

```
1080 ON NUM GOTO 1100,1150,1110,1120,1150,1130
1100 GOSUB INSERT:GOTO 1160:REM INSERTS NEW NODE
1110 GOSUB MODIFY:GOTO 1160:REM MODIFIES NODE
1120 GOSUB DELETE:GOTO 1160:REM DELETES A NODE
1130 GOSUB RETRIEVE:GOTO 1160:REM RETRIEVES A NODE
1150 ? CMND$,"IS A BAD COMMAND":STOP
1160 ? #3:GOTO 1060
2000 GOSUB STORE
2010 ? #3:? #3:? #3,,,"Data File Dump":? #3:? #3:? #3;"";LLST$
2015 CLOSE #3
2020 END
3000 REM FOLLOWING IS THE DATA TO BE INPUT TO THE PROGRAM
3010 REM ******************************************************
3020 DATA ANDERSON MARSHALL H.,432393333,345-2645
3030 DATA CHAMBERLAIN CHARLES B.,213485420,628-8942
3040 DATA HARRISON PATRICIA N.,321467382,233-8523
3050 DATA SAMUELSONN PAULOS E.,356204060,345-3050
3060 DATA WILLIAMSON PETROS E.,502346049,621-7659
3070 DATA F,WILLIAMSON PETROS E.
3080 DATA C,CHAMBERLAIN CHARLES B.,NAME,CHAMBERLIN CHARLIE B.
3090 DATA F,ANDERSEN HAROLD H.
4000 DATA A,MARSHALL JAMES P.,323504932,450-8543
4010 DATA D,SAMUELSONN PAULOS E.
4020 DATA A,MITCHELL WILLIAM A.,430405067,430-3322
4030 DATA A,ADAMS SAMUEL A.,503235000,543-0560
4040 DATA A,YATES MARVIN H.,503005040,654-9009
4050 DATA A,PETERSEN DONALD T.,453403040,672-0011
4060 DATA C,PETERSEN DONALD T.,SSN,453043044
4070 DATA C,MORRISON WALTER A.,PHN,569-8904
4080 DATA C,YATES MARVIN H.,NAME,YATES MARTHA H.
4090 DATA D,ADAMS SAMUEL A.
4100 DATA D,YATES MARVIN H.
4110 DATA F,YATES MARTHA H.
4120 DATA D,YATES MARTHA H.
4130 DATA D,PETERSEN DONALD T.
4140 DATA F,SAMUELSONN PAULOS E.
4150 DATA F,MITCHELL WILLIAM A.
5000 CLOSE #3:STOP
```

●

USERS' GROUPS

Computer Workshop Atari User Grp.
1200 Haddonfield Rd.
Cherry Hill, NJ 08002
3rd Tuesday of Month @ 7 PM

LINCOLNLAND ATARI GROUP
c/o John E. Anderson
P.O. Box 91
Pleasant Plains, IL 62677

Index to Advertisers

VCS UPDATE...

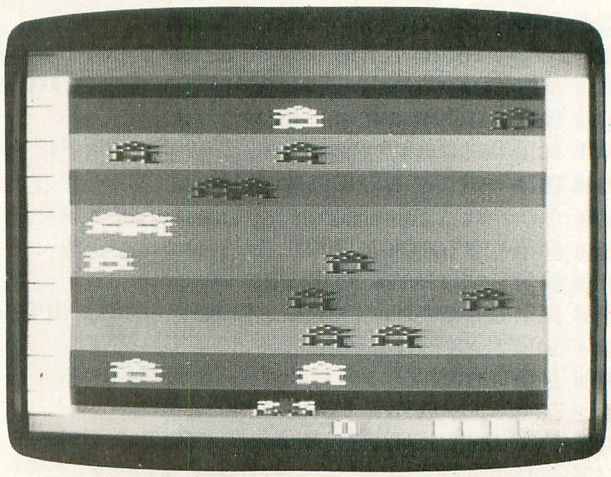
by Lee Pappas

Last issue several new titles for the ATARI™ VCS* were announced. I have officially lost track as to how many cartridges are now available, nor do I have any idea as to the exact number of games that will be available in the coming months, because there are so many being released. Here is a partial listing of the game programs announced at the summer Consumer Electronics Show in Chicago this past June. Even more astonishing than number of games in production are the myriad of new (and old) companies now producing all of these cartridges.

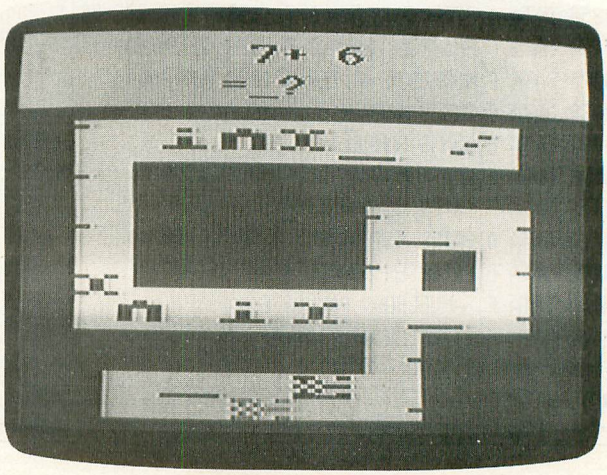
Soon to be available from ATARI is DEMONS TO DIAMONDS, a kids game designed to increase hand/eye coordination. MATH GRAND PRIX allows children to solve math problems to move their cars around a track. BERZERK is a very good representation of the arcade robot chase game. Transferred over from the home computers, STAR RAIDERS also works well on the VCS. What's really neat about this game is the inclusion of a small, angled control panel with dedicated "buttons" for the galactic map, computer, etc. List price on STAR RAIDERS and BERZERK is $39.95. Also from ATARI come enhanced versions of HOMERUN and FOOTBALL. HOMERUN is now called BASEBALL — and neither game resembles its previous version. The graphics are very impressive for the 5-year-old VCS. Other games previewed at CES were FROG POND, VOLLEYBALL, a new updated COMBAT, RAIDERS OF THE LOST ARK, and ADVENTURE I and II. I have to admit that the graphics on most of these new games are very well done!

A new company to the VCS market, ARCADIA, has a new approach to the ATARI home game system. A master cartridge plugs into the VCS console, and the software is loaded from cassette. The initial cost of the master cartridge is $69.95, while the cassettes are priced at $14.95. The first three titles announced are FIREBALL, COMMUNIST MUTANTS FROM SPACE, and PHASER PATROL. Plans for 1983 are for the release of three new games per quarter.

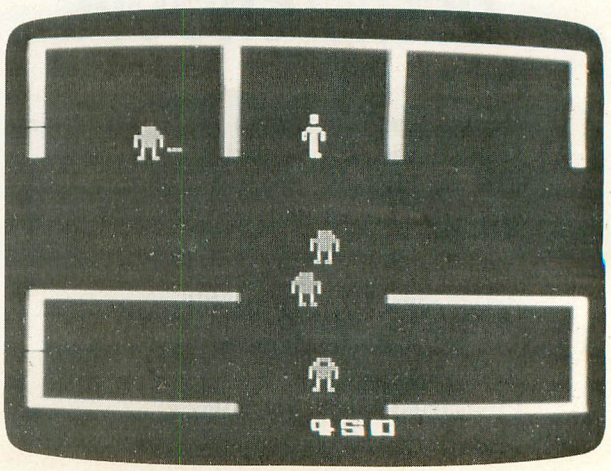
From the old standby, ACTIVISION, comes MEGAMANIA and PITFALL. In MEGAMANIA you fight off wave after wave of weird objects. Weird . . . like enemy hamburgers, rotating bow-ties, and cosmic dice. Seek out ancient civilizations in the jungle in PITFALL as you swing through trees, jump pits, and try to avoid cobras and other unfriendly types. These two games follow STARMASTER and CHOPPER COMMANDER, the latter one of my favorites on the VCS (Thanks, Activision).



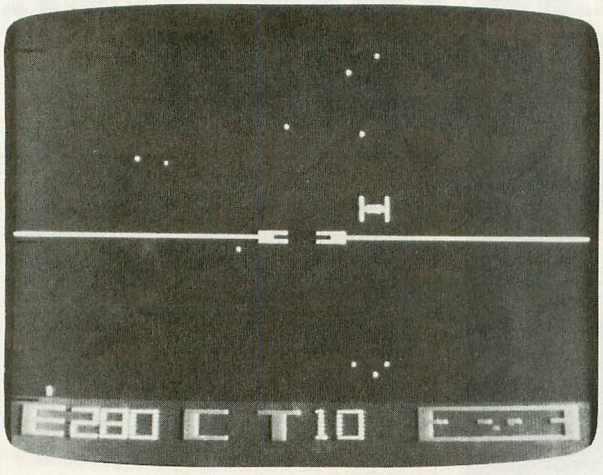
DEMONS TO DIAMONDS™



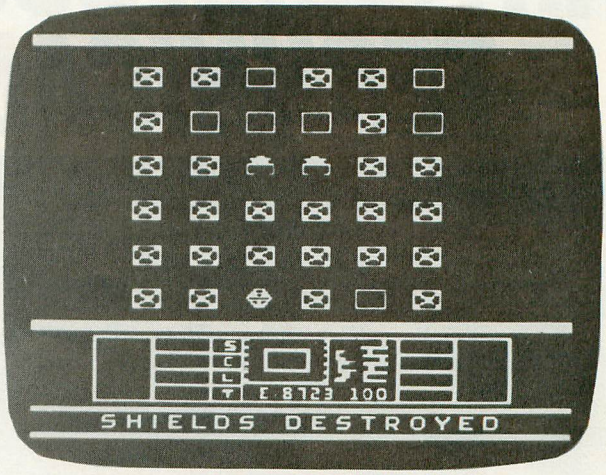
MATH GRAND PRIX™



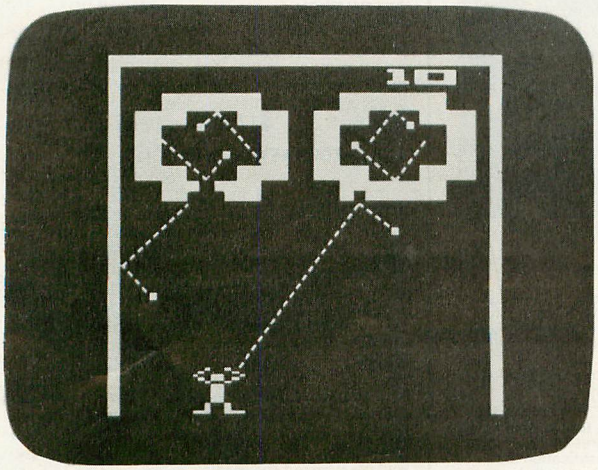
BERZERK™



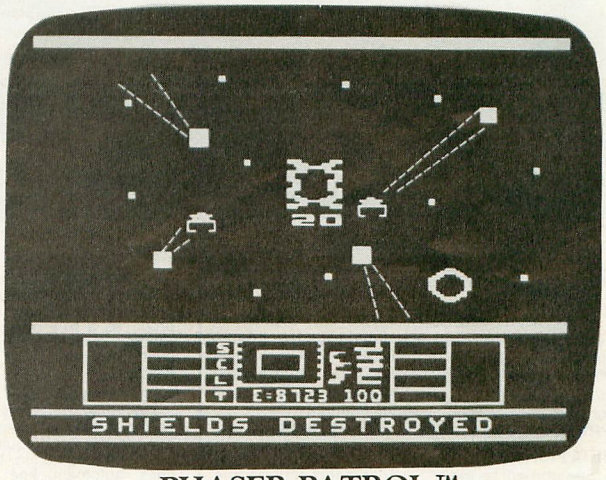
STAR RAIDERS™



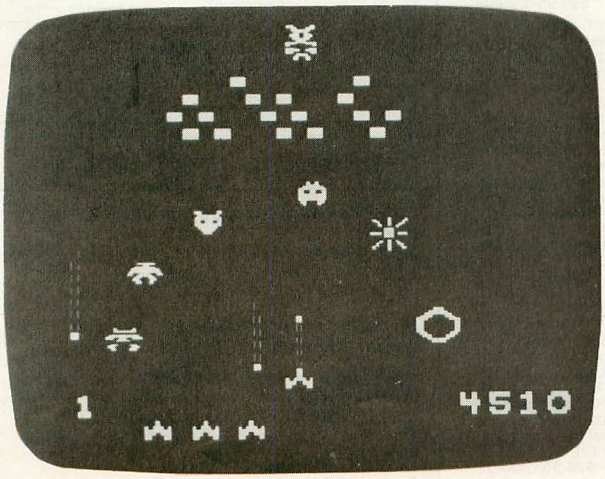
PHASER PATROL™



FIREBALL™



PHASER PATROL ™



COMMUNIST MUTANTS FROM SPACE™

SPECTRAVISION is a new company with several new games, including PLANET PATROL and GANGSTER ALLEY. Both games have interesting concepts and nice graphics and sound.

Even 20th CENTURY FOX is getting in on the act with DEADLY DUCK, BEANY BOPPER, FAST EDDIE, and WORM WAR I. The games ideas are good, although the graphics are not up to par with many of the other competitors now entering the field.

TIGER Electronic Toys has a new division, TIGERVISION, that (you guessed it) manufactures VCS games. Their new games are RIVER PATROL, MARAUDER, KING KONG, JAWBREAKER, and THRESHOLD. The latter two are licensed from On-Line Systems and have been adapted to the VCS. Intelligently, Tigervision didn't try and rewrite the games, but changed them completely while keeping the basic concepts of the games.

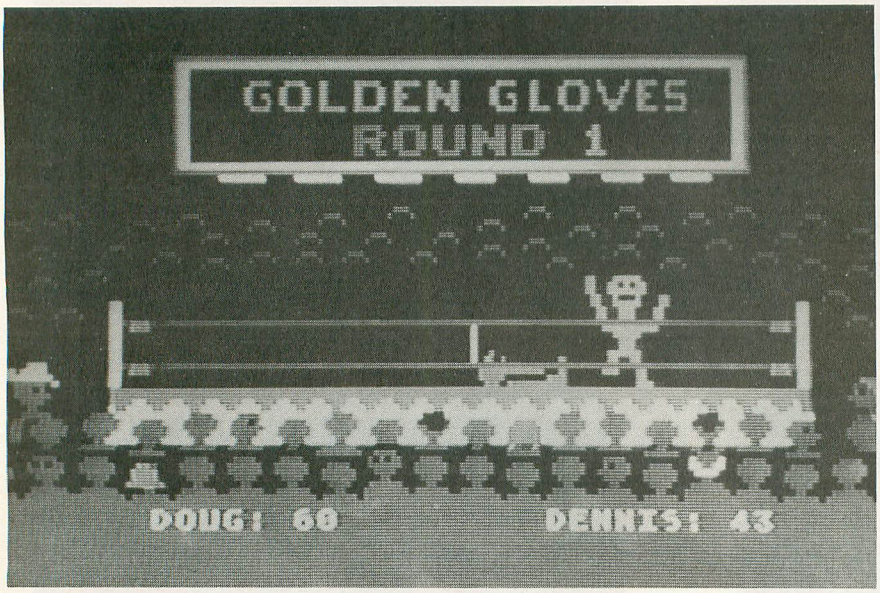
IMAGIC, hot on the heels of DEMON ATTACK, has announced SPHINX, COSMIC ARK, ATLANTIS, and FIRE FIGHTER; all with great graphics and sound. Incidentally, DEMON ATTACK will be available for the ATARI computers shortly.

US GAMES also has several new releases, and COLECO's games will soon hit the market, along with their new video game console. Mattel (makers of Intellivision) will shortly be releasing their new cartridges adapted to the ATARI unit. (Take that, George Plimpton!)

If these games aren't enough to satisfy your game habit, time will tell how many cartridges will be available this time next year! □

*TM of ATARI, Inc.

ARTWORX SCORES ANOTHER TECHNICAL KNOCKOUT.



Scene from GOLDEN GLOVES

HODGE PODGE: by Marsha Meredith (Atari and Apple)

NOW AVAILABLE FOR ATARI!!! This captivating program is a marvelous learning device for children from 18 months to 6 years. HODGE PODGE consists of many cartoons, animation and songs which appear when any key on the computer is depressed. A must for any family containing young children.
PRICE . . . . . .**$19.95** diskette

BETA FIGHTER: by Douglas McFarland (Atari, 16K)

See who will be the ace gunner in this action game set on a spectacular Martian landscape. BETA FIGHTER can be played with one or two players and uses player/missile graphics and delightful sound effects.
PRICE. . . . . .**$16.95** cassette **$20.95** diskette

DRAWPIC: by Dennis Zander (Atari 16K)

DRAWPIC provides the user with an unbelievably easy way to create screens in graphics modes 3-7. Just sit back with your joystick and use POINT PLOT, DRAW LINE, RUBBER BAND fill and COLOR SET to create beautiful images on your Atari. Full or partial screen images are saved as string data in the program and can be instantly recalled and combined into new images using machine language subroutines. These graphic images can be easily incorporated into your own programs. The images of HODGE PODGE and the landscape of BETA FIGHTER were made using DRAWPIC.
PRICE . . . . . .**$29.95** cassette **$33.95** diskette

☐ **ROCKET RAIDERS** by Richard Petersen (Atari 24K)

Defend your asteroid base against pulsar bombs, rockets, lasers, and the dreaded "stealth saucer" as aliens attempt to penetrate your protective force field. Precise target sighting allows you to fire at the enemy using magnetic impulse missiles to help protect your colony and its vital structures.
PRICE . . . . . **$19.95** cassette **$23.95** diskette

☐ **FOREST FIRE TWO:** by Richard Petersen (Atari 24K)

FOREST FIRE has been enhanced and now offers a two player mode for head to head competition to see who can survive, suffer the least damage and put their fire out first. User input now determines landscape, wind and weather conditions, offering limitless game variation. FOREST FIRE's excellent color graphics have been made even better, turning your computer into a super-detailed fire scanner.
PRICE . . . . . .$16.95 cassette $20.95 diskette

☐ **FORM LETTER SYSTEM:** (Atari, North Star and Apple)

This is the ideal program for creating personalized form letters! FLS employs a simple-to-use text editor for producing fully justified letters. Addresses are stored in a separate file and are automatically inserted into your form letter along with a personalized salutation. Both letter files and address files are compatible with **ARTWORX** MAIL LIST 3.0 and TEXT EDITOR programs.
PRICE . . . . . **$39.95** diskette

☐ **PILOT:** by Michael Piro (Atari, 16K)

Pilot your small airplane to a successful landing using both joysticks to control throttle and attack angle. PILOT produces a true perspective rendition of the runway, which is constantly changing. Select from two levels of pilot proficiency.
PRICE . . . . . **$16.95** cassette **$20.95** diskette

☐ **TEXT EDITOR:** (Atari and North Star)

This program is very "user friendly" yet employs all essential features needed for serious text editing with minimal memory requirements. Features include common sense operation, two different justification techniques, automatic line centering and straightforward text merging and manipulation. TEXT EDITOR files are compatible with **ARTWORX** FORM LETTER SYSTEM.
PRICE . . . . . **$39.95** diskette

☐ **MAIL LIST 3.0:** (Atari, Apple and North Star)

The very popular MAIL LIST 2.2 has now been upgraded. Version 3.0 offers enhanced editing capabilities to complement the many other features which have made this program so popular. MAIL LIST is unique in its ability to store a maximum number of addresses on one diskette (typically between 1200 and 2500 names!). Entries can be retrieved by name, keyword(s) or by zip codes. They can be written to a printer or to another file for complete file management. The program produces 1, 2 or 3-up address labels and will sort by zip code (5 or 9 digits) or alphabetically (by last name). Files are easily merged and MAIL LIST will even find and delete duplicate entries! The address files created with MAIL LIST are completely compatible with **ARTWORX** FORM LETTER SYSTEM.
PRICE . . . . . **$49.95** diskette

☐ **THE VAULTS OF ZURICH:** by Felix and Greg Herlihy (Atari, 24K, PET)

Zurich is the banking capital of the world. The rich and powerful deposit their wealth in its famed impregnable vaults. But you, as a master thief, have dared to undertake the boldest heist of the century. You will journey down a maze of corridors and vaults, eluding the most sophisticated security system in the world. Your goal is to reach the Chairman's Chamber to steal the most treasured possession of all: THE OPEC OIL DEEDS!
PRICE . . . . . **$21.95** cassette **$25.95** diskette

☐ **BRIDGE 2.0** by Arthur Walsh (Atari (24K), Apple TRS-80, PET, North Star and CP/M (MBASIC) systems)

Rated #1 by Creative Computing, BRIDGE 2.0 is the only program that allows you to both bid for the contract and play out the hand (on defense or offense!). Interesting hands may be replayed using the "duplicate" bridge feature. This is certainly an ideal way to finally learn to play bridge or to get into a game when no other (human) players are available.
PRICE . . . . . **$17.95** cassette **$21.95** diskette

☐ **ENCOUNTER AT QUESTAR IV:** by Douglas McFarland (Atari, 24K)

As helmsman of Rikar starship, you must defend Questar Sector IV from the dreaded Zentarians. Using your plasma beam, hyperspace engines and wits to avoid Zentarian mines and death phasers, you struggle to stay alive. This BASIC/Assembly level program has super sound, full player missile graphics and real time action.
PRICE . . . . . **$21.95** cassette **$25.95** diskette

NEW PROGRAMS!

GOLDEN GLOVES: by Douglas Evans (Atari 24K)

Use your joystick to jab, block and duck as each player attempts to land the knockout punch. This unique real-time program brings all of the excitement of ringside to your Atari. GOLDEN GLOVES is a one or two-player game, or you can be a spectator as the computer controls both fighters.
PRICE . . . . . .**$22.95** cassette **$26.95** diskette

CRAZITACK: by Peter Adams (Atari 16K)

The Crazies are attacking us and the only defenses are three MX bases. Missiles can be launched singly or in a salvo, but it is doomsday when you run out of missiles.
PRICE . . . . . .**$17.95** cassette **$21.95** diskette

DOMINATION: by Alan Newman (Atari 24K)

Between one and six players compete for power via economic, diplomatic and military means in this award-winning game. You must make decisions quickly, exercise skillful hand-eye coordination, outguess your opponents and cope with random events.
PRICE . . . . . .**$17.95** cassette **$21.95** diskette

POKER TOURNEY: by Edward Grau (Atari 32K, Northstar)

You are entered in a high stakes Draw Poker Tournament facing six opponents including Lakewood Louie, Shifty Pete and Dapper Dan. Each has his own style of play and of bluffing. POKER TOURNEY utilizes the Joker, has true table stakes play and each hand is played based on pot odds. The Atari version's graphics and sound are superb of course (programmed by Jerry White) making POKER TOURNEY the class program of its type.
PRICE . . . . . .**$18.95** cassette **$22.95** diskette

HAZARD RUN: by Dennis Zander (Atari, 16K)

The sheriff has spotted you and you must make the treacherous run through Crooked Canyon past Bryan's Pond to the jump at Hazard Creek and safety. You can even put the joystick-controlled GEE LEE car up on two wheels to make it through some tight spots. A lead foot is not always the answer as you dodge trees, rocks and chickens in this nerve-racking game. HAZARD RUN employs full use of player/missile graphics, re-defined characters and fine scrolling techniques to provide loads of fast action and visual excitement.
PRICE. . . . . .**$27.95** cassette **$31.95** diskette

ORDERING INFORMATION

Call ARTWORX toll-free number to order direct:
800-828-6573
In New York, Alaska, Hawaii call:
(716) 425-2833

All orders are processed and shipped within 48 hours.

Shipping and handling charges:
Within North America: Add $2.00
Outside North America: Add 10% (Air Mail)
New York State residents add 7% sales tax

Quantity Discounts:
Deduct 10% when ordering 3 or more programs

Ask for ARTWORX at your local computer store.

Write for FREE Catalogue listing more information about these and other quality ARTWORX programs.

Artworx

150 North Main Street Fairport, NY 14450

HARDWARE REVIEW: PERCOM DOUBLE DENSITY DISK DRIVE

PERCOM DATA CO.
211 N. Kirby
Garland, TX 75042
$799.00 - $399.00 Add-on drive

by Winston Lawrence

Double density 5¼ inch disk drives for the ATARI have finally arrived. PERCOM Data Company is now shipping ATARI compatible disk drives. I ordered one of the drives from Barry Cornfield at Stony Clove Computer Systems and within a week had one of the first drives in New York.

One of the biggest drawbacks with trying to use the ATARI computer system for business and serious programming has been the low storage capability of the single density 810 disk drives. The single density drive will hold about 88,000 bytes (characters) of storage which does not leave much room for data once programs have been stored on the disk. The new double density disk drive will allow approximately 176,000 bytes of storage. The increased storage will open up the ATARI to much more sophisticated software due to the larger disk storage capability.

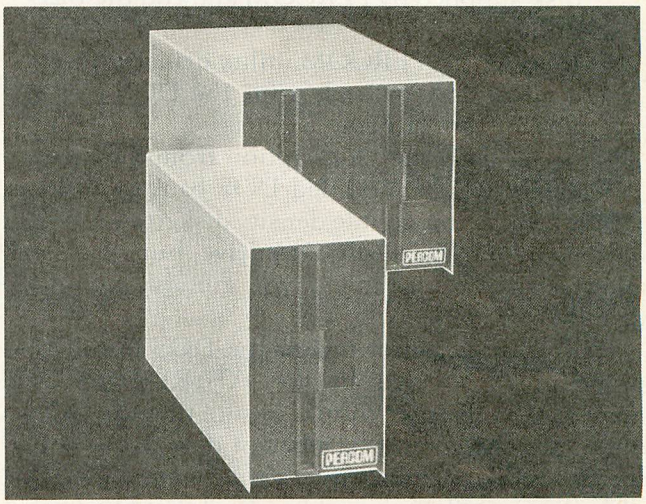
The single unit disk drive is about 50% heavier than the ATARI 810 drives and is encased in a solid metal enclosure. The drive stands about 6½ inches high, by 4½ inches wide and 11 inches deep. Two of these drives would take the same amount of desk space as one 810 drive due to the vertical mounting of the PERCOM disk drive as opposed to the horizontal mounting of the ATARI 810 disk drive.

The PERCOM drive comes with an installation manual which explains the installation and checkout of the drive system. The manual I received was marked "Preliminary Copy" and several sections were left blank, with a notation that the information would be included in a later version of the manual. Nevertheless, the actual installation of the drive was easy, and the manual did explain how to attach the drive to ATARI computers with no other drives or to ATARI systems with one or more 810 drives. The drive came with the familiar ATARI connector cable and plugged in like any other ATARI peripheral device. One point noted strongly in the manual is that, when the ATARI computer system does contain 810 drives, the PERCOM controller drive MUST be drive number one in the system.

The back of the PERCOM controller drive has two ATARI connectors and a white 34 pin connector. This connector is used when adding additional PERCOM drives and will be very important in the future. This connector can be used by the controller to control additional 40 track double-density drives, 80 track drives, and even 8 inch drives. Only the first PERCOM drive must have the controller card installed, and this controller can control three additional drives. If you add two additional controller drives (each of which can control three add-on drives), then you can have an eight drive system instead of the maximum of four drives when using ATARI 810's.

Additional PERCOM add-on drives do not require the controller cards. This allows the cost of an add-on drive at PERCOM's list price of $399.00 to be half the cost of the $799.00 double density drive with the controller card.

The only hardware problem I noticed since the installation is that the cassette cannot be used with the PERCOM unit attached. I do not use the cassette very often, so this does not provide a big problem for me. When I do need it, I boot the disk then plug the connector out of the computer console I/O port, and plug the cassette in. I unplug the cassette when finished, plug the drive connector back in and continue.



The PERCOM drive controller will detect whether a disk is single density or double density during the boot process and switch to the correct mode of operation, making it suitable for use in an ATARI system. Even when it is the only drive in the system, single density disks will cause the drive to operate in single density mode.

PERCOM supplies a BASIC demo program that changes the mode of operation of the PERCOM unit from double to single density, or from single to double density. One other utility supplied by PERCOM is named BLD and is used to convert the ATARI DOS 2.0S to work with double and single density drives. PERCOM does NOT supply a DOS with its drives, so the purchaser MUST have a copy of DOS 2.0S already before using the system. The DOS works the same as the unmodified ATARI

DOS 2.0S, however, the PERCOM installation manual notes that the duplicate functions (file or disk) will NOT work between double density and single density disks. The DOS menu's copy file function can be used to copy a file or an entire disk using the *.* wildcard as file names. The duplicate functions WILL work if the PERCOM disk is in single density mode AND the diskette was formatted in single density mode.

PERCOM has provided a utility called SDCOPY for users with only one disk drive. SDCOPY is used to copy (duplicate) disks or files of different densities.

As mentioned before, PERCOM does not supply DOS with the drives, so you must have a copy of DOS 2.0S available to create the double density DOS. ATARI's DOS 2.0S already has the code to handle double density, since it would have been used for the ill-fated ATARI 815 dual density drive.

The following steps must be taken to create a double density DOS.

1) Place the DOS 2.0S diskette into the PERCOM drive and boot the system.
2) Replace the DOS diskette with the utility disk supplied by PERCOM and run the BLD utility.
3) Remove the PERCOM utility diskette, insert a blank diskette and, using the DOS menu, format this diskette.
4) Select the write DOS option from the menu and write the modified DOS back to disk.

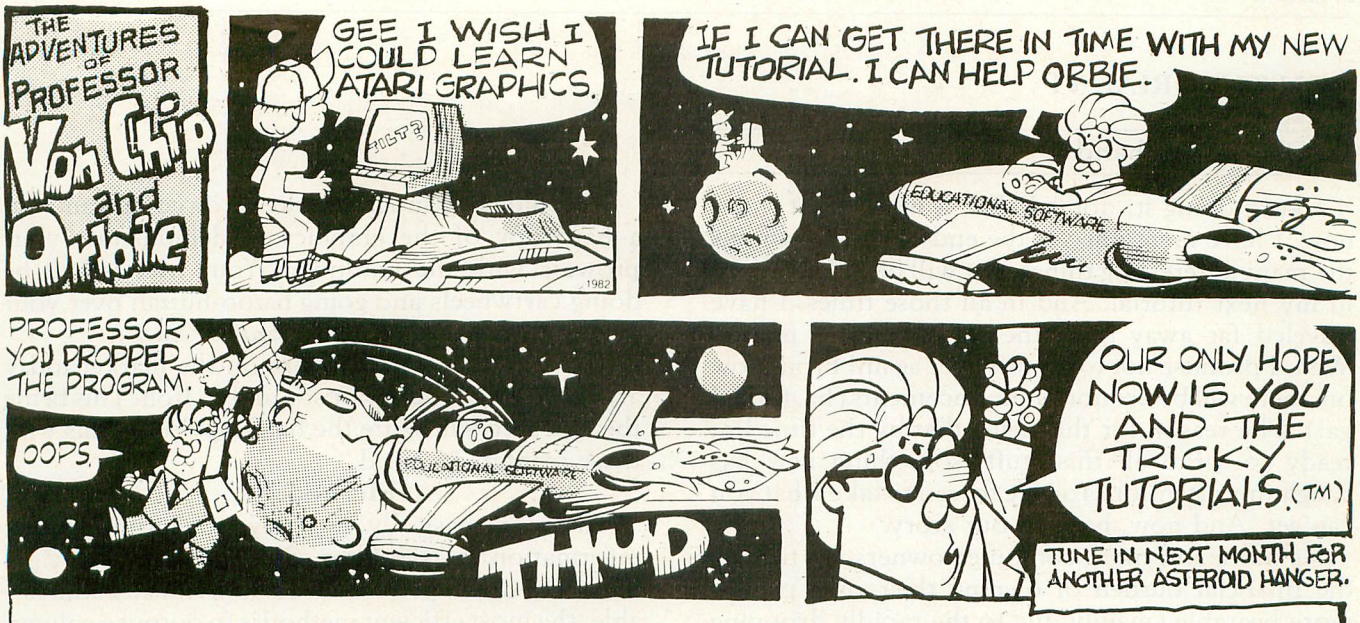
You now have a double density diskette which can only be used in the PERCOM drive, but which will allow normal operations if you are also using 810 disk drives.

PERCOM does not review the DOS installation procedures provided in the ATARI DOS 2.0S manual, and the user should ensure that the DOS is set up for the correct number of disk drives. I had several 160 errors attempting to use the PERCOM disk after installation until I realized that I had originally set up my DOS for single drive operation. It is a good idea to reread the ATARI DOS manual if any problems do occur after installing the unit.

The PERCOM disk drive worked flawlessly over more than six weeks of testing and — as a side benefit — is faster than the ATARI 810 drive. I/O operations to the drive take a noticeably shorter time. PERCOM has not released any information as to the internal operation of the drive, but at least one command has been added to the original ATARI disk commands. PERCOM's drive controller demonstration program shows a 78 (decimal) command which apparently causes the drive to return 12 bytes of information. This information indicates the number of tracks in the drive, the number of sectors in a track, and the number of bytes in a sector. This command does not return any useful information when sent to an 810 drive. The other disk commands (PUT SECTOR, PUT SECTOR WITH VERIFY, GET SECTOR, FORMAT DISK, and STATUS) work the same as with the ATARI 810 drive.

ATARI computer owners contemplating the purchase of a first or second disk drive should give serious consideration to the PERCOM unit. Although the $799.00 list price is nearly 50% higher than the ATARI 810 list price, double density disks and the capability to control 8 inch disk drives make this a tough act to follow. The biggest problem with adding these drives at present is a lack of software to control these new drives and the ATARI 810 drives at the same time.

Rumors are now surfacing that ATARI will announce new drives in the near future, but they will have to be much better than the 810 drives in storage, performance, and reliability to keep PERCOM from taking a large share of ATARI's disk drive market. The competition can only mean good things in the future for ATARI owners. □



TRICKY TUTORIALS (tm)

There are many things that the ATARI computers can do either better, or easier than other small computers. The following series of programs is designed for anyone who is at least familiar with BASIC programming. What each tutorial offers is similar to an extensive magazine article with all discussion in as simple language as possible, plus you get MANY examples already typed in and running. The instruction manuals range from 10 to 50 pages, and some tutorials fill up a complete tape or disk. There is little overlap in what is taught, so anyone wanting to know all they can should buy them all (my banker thanks you). ATARI buys these from us to use in training their own people! Rave reviews have been published in ANTIC, ANALOG, CREATIVE COMPUTING, and even INFOWORLD. You trust INFOWORLD, don't you?

TT #1: DISPLAY LISTS—This program teaches you how to alter the program in the ATARI that controls the format of the screen. Normally, when you say "Graphics 8", the machine responds with a large Graphics 8 area at the top of the screen and a small text area at the bottom. Now, you will be able to mix various Graphics modes on the screen at the same time. The program does all of the difficult things (like counting scan lines). You will quickly be able to use the subroutines included in your own programs.
16K Tape or 24K Disk. **$19.95**

TT #2: HORIZONTAL/VERTICAL SCROLLING—The information you put on the screen, either GRAPHICS or TEXT, can be moved up, down, sideways, or diagonally. We provide the basic methods and leave the rest up to your skill and imagination. Includes 18 examples to get you started, with several using a small machine language subroutine for smoothness.
16K Tape or 24K Disk. **$19.95**

TT #3: PAGE FLIPPING—Now you don't have to redraw the screen every time you change the picture or text. You will learn how to have the computer draw the next screen you want to see while you are still looking at the previous screen, then flip to it instantly. You won't see it being drawn, so a complicated picture can seem to just appear. Depending on your memory size and which graphics or text modes you are using, you can instantly look at up to 50 pages. The basic method takes only 9 lines and the usefulness is infinite.
16K Tape or 24K Disk. **$19.95**

TT #4: BASICS OF ANIMATION—This program shows you how to animate simple shapes (with some sound) using the PRINT and PLOT commands, and it also has a nice little PLAYER/MISSILE GRAPHICS game you can learn from. The P/M example is explained and will get you started on this complicated subject (more fully explained in TT #5). This would be an excellent way to start making your programs come alive on the screen with movement! Recommended for beginning users.
16K Tape or 24K Disk. **$19.95**

TT #5: PLAYER/MISSILE GRAPHICS—Learn to write your own games and other animated applications! The tutorial begins with many small examples that compliment the 50 page manual, then gradually builds up to a complete game where everything you need to know is fully explained. Also included are two machine language utilities that you can use to animate Players with from BASIC. Next we include two of the best editors currently available; one for editing playfield shapes (backgrounds); and one to edit your players, and all in glorious Technicolor!! Everything except the two editors run in 16K Tape or 32K Disk. **$29.95**

TT #6: SOUND AND MUSIC—Unless you have spent many years experimenting with the four voice channels, you will learn a lot from this one! Learn to play standard notes, chords, and whole songs using some simple "tricks". One of the nicest parts are the examples of special sound effects that you can refer to whenever you need a sound for a program or to impress a friend. This program will be of interest to all ages and levels of experience!
16K Tape or 24K Disk. **$19.95**

SPECIAL DISCOUNT

Order the first six tutorials in a 3-ring binder for $99.95, a $30.00 savings!

TT #7: DOS UTILITIES—We at Educational Software have been shocked by some of the prices others are charging to offer you small utilities to help in the use of your Disk Drive. We now offer you all of the following plus explanation as to how each was written, and how to use them: A UNIQUE MENU PROGRAM, AN AUTORUN.SYS BUILDER, DISK INSPECTOR (LOOK AT SECTORS), DISK JACKET PRINTER, AUTOMATIC FORMATTER, RECORD SAVE AND LOAD UTILITY.
32K Disk Only. **$29.95**

MASTER MEMORY MAP (tm)

This book is the most valuable source of information for your ATARI you can buy. It starts out by explaining how to PEEK and POKE values into memory, so that even new computer owners can use many of these "Tricks". Then you are given 32 pages of the memory locations that are the most useful, along with hints on how to use many of the locations. Finally, it includes hints on problems you may be having with the computer and discusses the new Graphics modes 9 to 11. Even ATARI buys this book from us! **$6.95**

CONTACT YOUR LOCAL DEALER or ORDER BELOW

We have other fine programs for ATARI computers.
Write for a catalog.
Send us your programs to sell too!

USER SUBMITTED PROGRAMS

MINI-DATABASE/DIALER—stores and edits up to 8 lines of information such as names & addresses, phone numbers, messages, inventories, or anything you want. It has the usual sort, search, and print options, but it also has an unusual feature: If your file includes phone numbers and your phone company allows touch-tone phone signals, the program will DIAL THE PHONE NUMBER FOR YOU! 16K Tape or 24K Disk. **$24.95**

THE GRAPHICS MACHINE—allows the ATARI to act like more expensive graphics computers using simple commands like line, box, circle, polygon, fill, and savescreen to get a high resolution picture you can save on disk in only five seconds! Many more features! 48K Disk Only. **$19.95**

BOB'S BUSINESS—14 small business type programs accessed from a common menu. 16K Tape or 32K disk. **$14.95**

MINI WORDPROCESSOR—A simple text editor to write, save, and print several pages at a time. 32K Tape or Disk. **$19.95**

KID'S #1—Includes a MATH QUIZ, a children's TREASURE HUNT, and a DIALOGUE program. 16K Tape or 24K Disk. 3 for . . . **$14.95**

KID'S #2—SPELLING BEE, WORD SCRAMBLE, and TOUCH. 16K Tape or 24K Disk. 3 Educational games for . . . **$14.95**

PLAYER PIANO—Turns your keyboard into a mini-piano and more. 24K Tape or 32K Disk. **$14.95**

GRAPHIC SYMBOL LABELS—for your keyboard to remind you of the built-in Graphics symbols. 2 complete sets for . . . **$2.95**

OUR NEWEST PROGRAMS

DOG DAZE—Two cute little doggies race for the fire hydrants, shoot their bones, and just have a lot of fun! A fast action program for all ages. 8K Tape or 16K Disk, in machine language. **$16.95**

*** OUR BEST GAME ***

SPACE GAMES—Our family is being attacked by ALIENS, and only you can save us. A comic book manual will guide you through three games that test your ability in space skills. Includes ALIENS, SURVIVE, and ROBOT ATTACK, and is for all ages. The first two games require 16K for Tape. The last game and all Disk users need 32K. **$24.95**

MATHS FOR FUN—Another ENGLISH import teaching basic math skills. Very colorful and enjoyable to use. For ages 5 to 16. 16K Tape or 24K Disk. **$19.95**

MARATHON—This is a unique math quiz for one or two players. You are in a race to move your runner across the screen first! There are four levels of play with five modes of operation for each. The game uses joysticks for all input, so play is easy for young children. This wonderful learning tool is imported from ENGLAND for your learning pleasure. Your kids will never even notice they are playing an EDUCATIONAL program. 16K Tape or 24K Disk. **$19.95**

GRADE BOOK—This nice record keeper will maintain a file of 35 students' test scores along with comments. Up to 7 subjects are allowed. 24K Tape or 32K Disk. A printer is optional. **$24.95**

To order COD, VISA or MasterCard call — **(408) 476-4901**

By mail include $3.00 postage (USA only) or $1.50 Memory Map only—
California residents add 6.5% TAX.—Specify Tape or disk.

EDUCATIONAL SOFTWARE inc.

5425 JIGGER DR.
SOQUEL, CA 95073

NON-TUTORIAL VI

by Charles Bachand

I keep doing it, don't I? For a number of times now, I have mentioned at the end of these tutorials the many interesting things that will be talked about in my next tutorial. And in all those times, I have traveled far away from the stated subject matter. Well, I promise not to do this ever again! From now on, there will be no more announcements (ha, fooled ya!). The reason for this being that by the time I'm ready to write all this stuff, something new has caught my attention. You are forced to take what you can get. And now, back to our story.

As more and more computer owners are finding the financial burden of owning their own printer more bearable (mainly due to the rapidly dropping prices, especially of those printers imported from Japan), they are asking for different ways to put their new-found toys to use. Many of the newer printers coming onto the market are microprocessor controlled, which allows them the ability to change not only character size and type style, but also the page length, line spacing, margin settings, underlining, and a host of other "neat" features. ("Yes, I use the Japanese character font every day. Doesn't everyone?")

Bernard Herrmann Music Starts Here...

"...but the feature that will concern us today is that of graphics, that unique and somewhat obscure ability to draw a picture on a screen and put it onto paper. For some, this is as easy as taking candy from a baby. For others, it is an impossibility, to be realized only in one's mind. Join us now as we meet a man, a computer, and a not so ordinary printer, all of which are about to connect, in the "Twilight Zone". "

Technical Stuff Starts Here!

or:

HATS OFF TO ATARI!!

This article contains a graphics program called "Archimedes Spiral". The program, although quite short, takes nearly three hours to run! This is definitely not a quick demo. (To produce the transparent version of the spiral, delete line 240.) *(It still looks like a hat to me. Ed.)*

```
100 REM ARCHIMEDES SPIRAL
110 REM
120 REM ANALOG MAGAZINE
130 REM
140 GRAPHICS 8+16:SETCOLOR 2,0,0
150 XP=144:XR=4.71238905:XF=XR/XP
160 FOR ZI=-64 TO 64
170 ZT=ZI*2.25:Z5=ZT*ZT
180 XL=INT(SQR(20736-Z5)+0.5)
190 FOR XI=0-XL TO XL
200 XT=SQR(XI*XI+Z5)*XF
210 YY=(SIN(XT)+SIN(XT*3)*0.4)*56
220 X1=XI+ZI+160:Y1=90-YY+ZI
230 TRAP 250:COLOR 1:PLOT X1,Y1
240 COLOR 0:PLOT X1,Y1+1:DRAWTO X1,191
250 NEXT XI:NEXT ZI
260 GOTO 260
```

It would be so much simpler if you could hand out a hardcopy of the graphics to demonstrate your prowess with the computer. Your friends will be doing cartwheels and going hazoo-huzzah over your printing expertise. *(Hazoo-huzzah?! Ed.)*

The most popular printers on the market today are the ones put out by Epson and C. Itoh. This being the case, we will write the hardcopy programs with these printers in mind.

Method #1

There are basically two ways to output graphic information to a printer. Since each of the eight hammers of the print head are individually addressable, the most efficient method is to output a column of eight pixels to the printer at a time. While this is the fastest way to do graphics, the code tends to be somewhat lengthy, and a somewhat obscure problem can crop up. This problem relates to the fact that a byte of the value 155 ($9B Hex) which is the ATARI's internal representation of a (RETURN), converts it to the decimal value 13 ($0D Hex), what the outside world understands as producing a carriage return signal. If the information for the eight pixels that you are working on adds up to 155, it will not look right in the final print out. This conversion is done in the 850 Interface module and I know of no way around it at this time.

Method #2

The second method is to use only one hammer of the print head. It will take eight times longer to produce a hardcopy of a picture, since you will only be working with one scan line at a time, but the code will be shorter and simpler. A routine to produce a printed copy of the Archimedes spiral follows.

```
260 REM EPSON GRAPHICS 8 DUMPER
270 REM ONE LINE AT A TIME!
280 OPEN #1,8,0,"P:"
290 PUT #1,27:PUT #1,65:PUT #1,1
300 REM ABOVE SETS PIXEL SPACING TO 1
310 FOR Y=0 TO 191
320 PUT #1,27:PUT #1,75
330 PUT #1,64:PUT #1,1
340 REM ABOVE 4 BYTE GRAPHIC HEADER
350 FOR X=0 TO 319:LOCATE X,Y,A
360 PUT #1,A:REM OUTPUT PIXEL
370 NEXT X:PRINT #1:NEXT Y
380 END
```

Line 290 sets the printer's line feed space of 1/72 of an inch. This the height of an individual print hammer. Lines 320 and 330 put the printer into graphics mode for one scan line. Line 330 specifies the number of bytes of graphic information to output to the printer. Since mode 8 graphics is 320 pixels across, we send this number to the printer in low byte, high byte format (64, 1). Line 350 sees if the pixel is a zero or a one, and line 360 sends it to be printed.

The next program listing is the same program written for the C.Itoh model 8510 printer. The main differences between this printer and the Epson is the method chosen to pass numeric values. A good example of this is how a 8510 is told that the next 320 bytes are to contain graphics. While the Epson people chose to specify number parameters as binary words (low byte, high byte) which is ideally suited to Assembly language programming, the manufacturer of the C.Itoh 8510 took an alternate route by passing numbers as character strings, something that BASIC can easily manipulate. Now the header of a 320 pixel line becomes: CHR$(27);"S0320".

```
260 REM C.ITOH GRAPHICS 8 DUMPER
270 REM
280 OPEN #1,8,0,"P:"
290 PRINT #1;CHR$(27);"T02"
300 REM ABOVE SETS PRINTER TO 1 PIXEL
310 FOR Y=0 TO 191
320 PRINT #1;CHR$(27);"S0320";
340 REM ABOVE OUTPUTS GRAPHICS HEADER
350 FOR X=0 TO 319:LOCATE X,Y,A
360 PUT #1,A:REM OUTPUT PIXEL
370 NEXT X:PRINT #1:NEXT Y
380 END
```

One thing to keep in mind if you are presently without a graphics printer but know someone who has one, is that all the information being sent to the printer does not have to go directly to it. The pixel data can be stored initially on a disk file, and later transferred to the printer, when one becomes available, using the copy file command in DOS.

The only negative effect in using the above programs with disk files is that the information will use up about 500 sectors of disk space. The next two programs will correct this problem by outputting six pixels per byte. The six pixel combination was decided upon as the largest number that divided evenly into 192 (the number of scan lines) and could not possibly equal 155 (the carriage return conversion problem mentioned earlier.)

```
260 REM EPSON GRAPHICS 8 DUMPER
270 REM SIX LINES AT A TIME!
280 OPEN #1,8,0,"P:":REM TRY DISK?
290 PUT #1,27:PUT #1,65:PUT #1,6
300 REM ABOVE SETS PIXEL SPACING TO 6
310 FOR Y=0 TO 191 STEP 6
320 PUT #1,27:PUT #1,75
330 PUT #1,64:PUT #1,1
340 REM ABOVE 4 BYTE GRAPHIC HEADER
350 FOR X=0 TO 319:A=0
352 FOR Y1=Y TO Y+5:LOCATE X,Y1,Z
354 A=A+A+Z:NEXT Y1
360 PUT #1,A:REM OUTPUT SIX PIXELS
370 NEXT X:PRINT #1:NEXT Y
380 END
```

```
260 REM C.ITOH GRAPHICS 8 DUMPER
270 REM SIX LINES AT A TIME!
280 OPEN #1,8,0,"P:":REM TRY DISK?
290 PRINT #1;CHR$(27);"T12"
300 REM ABOVE SETS PIXEL SPACING TO 6
310 FOR Y=0 TO 191 STEP 6
320 PRINT #1;CHR$(27);"S0320";
340 REM ABOVE OUTPUTS GRAPHICS HEADER
350 FOR X=0 TO 319:A=0
352 FOR Y1=Y+5 TO Y STEP -1
354 LOCATE X,Y1,Z:A=A+A+Z:NEXT Y1
360 PUT #1,A:REM OUTPUT PIXEL
370 NEXT X:PRINT #1:NEXT Y
380 END
```

These two programs will produce disk files that are now only 84 sectors long. A big improvement, don't you think? Anyway, this will allow you to put eight of these files on each and every disk you own. After all, "Eight Is Enough", don't you know? □

(Ed. Note: No one here at A.N.A.L.O.G is responsible for Charlie's state of mind when he writes these non-tutorials. Just thought you people would like to know.)

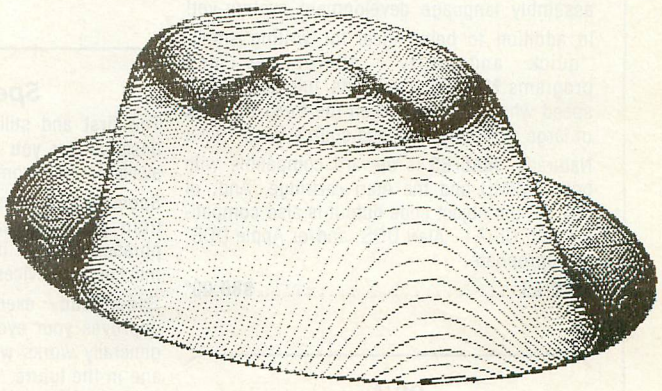


FIGURE ONE

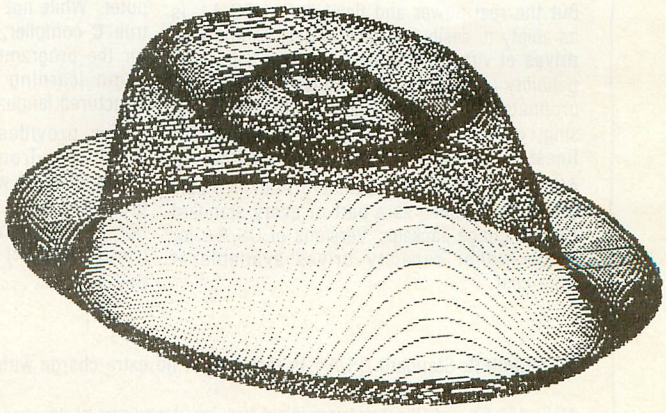


FIGURE TWO

FIRST and FINEST

In Systems Software for Atari and Apple

MAC/65

First we delivered Atari's Assembler/Editor (the cartridge).

Then we produced our enhanced "EASMD."

Now OSS is introducing the **finest** integrated assembly language development system yet!

In addition to being ideal for writing small, "quick and dirty" subroutines and programs, **MAC/65** shows its full power and speed when used with even the most complex of large assembly language source files.

Naturally, **MAC/65** is upward compatible with both EASMD and the Atari cartridge. And, of course, the object code output is also compatible with OS/A+, Atari DOS, and/or Apple DOS, as appropriate.

MAC/65 ..... **$80.00***

OS/A+

Optimized Systems Software — the group that produced **both** the **first** Apple DOS **and** the **first** Atari DOS — now brings you OS/A+, which combines the **finest** features of these and other successful personal computer operating systems.

OS/A+ is the **first** and **finest** operating system available for **both** Apple II and Atari computers and features a keyboard-driven, easy-to-use command processor. In addition to several simple resident commands, OS/A+ allows logical and readable requests for even the most sophisticated utility commands. In fact, the user can even add system commands as desired.

But the real power and flexibility of OS/A+ is its ability to easily interface to devices and **disk drives** of virtually any kind and size. File compatibility (with Apple DOS or Atari DOS, as appropriate), device independence, batch processing, easy of use — OS/A+ truly brings the **finest** in operating systems to **your** computer.

AND NOW OS/A+ (for standard Atari or Apple drives) is included as a part of **every** standard OSS language package. Versions of OS/A+ for some higher capacity drives available at extra cost.

Unless otherwise noted, all OSS products require 48K and at least one disk drive. We recommend 64K for the Apple version of OS/A+.

SpeedRead+

The **first** and still **finest** speed reading tutor designed for **you** to use on **your** computer is available **only** from OSS.

SpeedRead+ uses time-proven techniques to train **you** to instantly recognize words and phrases, and yet it goes far beyond what mere mechanical devices are capable of.

SpeedRead+ exercises your peripheral vision, improves your eye movement and timing, and generally works **with you** at your pace... now and in the future.

NOTE: The Atari version of SpeedRead+ needs only 16K of RAM.

SpeedRead+ **$59.95**

tiny C

As a product of Tiny C Associates, **tiny-c** was the **first** structured language interpreter for microcomputers. Now OSS brings this innovative interpretive language to **your** home computer. While not having the speed and power a true **C** compiler, **tiny-c** is an excellent choice for the programming student who is ready to begin learning the valuable techniques of structured languages.

tiny-c provides an easy-to-use, easy-to-modify environment that encourages experimentation while promoting proper programming style. The **tiny-c** package includes not only a comprehensive and instructional user manual but also **complete source.**

tiny-c **$99.95***

C/65

NOW AVAILABLE!

The **first** native mode **C** compiler ever produced for Atari **and** Apple computers.

C/65 supports a very usable subset of the extremely powerful and popular **C** language. Just as **C** is used by the most sophisticated programmers from the professional and academic communities, so shall **C/65** prove to be a powerful and much-needed tool for 6502 software developers.

C/65 supports integer and character types (and arrays), pointers, fully recursive functions, and much more.

NOTE: C/65 requires MAC/65 or an equivalent assembler. Two disk drives recommended but not required.

C/65 ..... **$80.00***

BASIC A+

"From the authors of Atari BASIC..."

It's a fact! OSS gave you that **first** and **most popular** language for Atari Home Computers. But why be content with the **first** when you can have the **finest**?

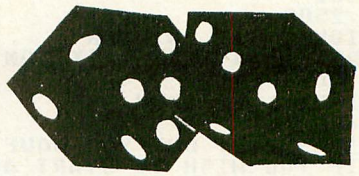
BASIC A+ is the **only** logical upgrade available to the Atari BASIC programmer. While retaining **all** the features which make Atari BASIC so easy to use, we've **also** given BASIC A+ features that place it at the forefront of modern interpretive languages. BASIC A+ will let **you** explore the worlds of structured programming, superior input/output, helpful programming aids, and even a very comprehensive PRINT USING command. **And**, exclusively for the Atari computer, an almost unbelievable array of PLAYER/MISSILE GRAPHICS commands and functions.

BASIC A+ **$80.00***

***REMEMBER:** Standard OS/A+ is **included** at no extra charge with BASIC A+, MAC/65, C/65, and tiny-c.

Optimized Systems Software, Inc., 10379 Lansdale Ave., Cupertino, CA 95014, (408) 446-3099

Triple Threat Dice



by Michael A. Ivins

Do you like to gamble but can't afford trips to Las Vegas or Atlantic City? If so then this program should be for you. By placing your bets carefully, you can be fairly sure of a high return, while impulse betting on the high odds might make you a big winner or it might make you go broke.

This game is modeled after a type of gambling machine found in Las Vegas casinos. These machines use three dice to play and give you several options to bet on. Unlike craps you are betting solely on the outcome of a single roll of the dice. You may bet up to five coins (normally quarters) on each of the betting options with no limit (other than your total cash) to how many of the options you choose to bet on.

You use your joystick to position the bet cursor next to the option you wish to bet. Pressing the trigger button will enter your bet one coin at a time up until you reach five coins after which it will not accept any more bets on that option. Moving the joystick to the left or right will move the cursor. After you have bet as many options as you wish, hold the joystick to the right until the pointer appears in the box marked "ROLL DICE". Press the trigger again and the computer will roll the dice.

After each roll of the dice the computer will display your win or say "SORRY" if you did not win. At this time you have an additional option. If you should wish to take your winnings and quit all you need do is pull the joystick toward to. A push on the trigger will return you to the betting routine.

Good luck and Happy Gambling! □

```
1 REM TRIPLE THREAT DICE
2 REM BY MICHAEL A. IVINS
10 DIM BET(31):COUNT=0
15 GRAPHICS 0:? "THIS IS A GAME PATTER
NED AFTER A":? "GAMBLING MACHINE IN LA
S VEGAS."
20 ? "YOU BET ON THE OUTCOME OF THE RO
LL OF":? "THREE DICE. YOU HAVE MANY OP
TIONS YOU"
25 ? "CAN BET ON.  TO SELECT THE OPTIO
N ON":? "WHICH YOU WISH TO BET, USE TH
E"
30 ? "JOYSTICK TO MOVE THE '>' UNTILL
IT":? "POINTS TO THE PROPER OPTION.  Y
OU"
35 ? "THEN ENTER YOUR BET BY PRESSING
THE":? "TRIGGER.  YOU MAY BET UP TO FI
VE"
40 ? "DOLLARS ON EACH OPTION."
45 ? :? "WHEN YOU HAVE FINISHED BETTIN
G, HOLD":? "THE JOYSTICK TO THE RIGHT
UNTILL A"
50 ? "POINTER APPEARS IN THE BOX MARKE
D":? "'ROLL DICE' AND PRESS TRIGGER."
60 ? :? "PRESS START TO BEGIN"
70 ? "GOOD LUCK!!!"
75 IF PEEK(53279)<>6 THEN 75
90 GOTO 1000:REM DRAW BETTING LAYOUT
100 M=100:POSITION 7,20:? M;
110 GOSUB 1200:REM CLEAR BETS RESET WI
N
120 IF COUNT=0 THEN M=100
130 IF STICK(0)=9 OR STICK(0)=10 OR ST
ICK(0)=11 THEN B=B-1:GOSUB 1500
132 IF STICK(0)=6 OR STICK(0)=7 OR STI
CK(0)=5 THEN B=B+1:GOSUB 1500
135 IF B<32 THEN IF BET(B)=5 OR M=0 TH
EN 160
140 IF B<32 AND STRIG(0)=0 THEN BET(B)
=BET(B)+1:POSITION X,Y:? BET(B);:SOUND
 0,150,10,15:M=M-1
141 IF B<32 THEN IF BET(B)=0 THEN POSI
TION X,Y:?  " ";
142 IF B<32 THEN IF BET(B)>0 THEN POSI
TION X,Y:? BET(B)
145 POSITION 7,20:? M;" ";
148 FOR DELAY=1 TO 20:NEXT DELAY
149 IF B>32 THEN B=32
150 IF B=32 AND STRIG(0)=0 THEN 200
155 FOR DELAY=1 TO 20:NEXT DELAY
160 SOUND 0,0,0,0:GOTO 130
200 COUNT=COUNT+1:REM ROLL AND DRAW DI
CE
205 GOSUB 2700
210 Y=0:A=INT(RND(0)*6+1):X=10:ON A GO
SUB 10000,10010,10020,10030,10040,1005
0
220 X=14:B=INT(RND(0)*6+1):ON B GOSUB
10000,10010,10020,10030,10040,10050
230 X=18:C=INT(RND(0)*6+1):ON C GOSUB
10000,10010,10020,10030,10040,10050
240 D=A+B+C:IF COUNT>1 THEN COUNT=1
250 REM PAY WINNING BETS
260 IF BET(0)=0 OR D<12 THEN 275
265 WIN=WIN+BET(0):POSITION 6,21:? WIN
;
270 POSITION 3,14:? "+";
275 IF BET(1)=0 OR D>9 THEN 290
280 WIN=WIN+BET(1):POSITION 6,21:? WIN
;
285 POSITION 3,15:? "+";
290 IF A<>B OR B<>C OR BET(2)=0 THEN 3
05
295 WIN=WIN+(BET(2)*36):POSITION 6,21:
? WIN;
300 POSITION 3,16:? "+";
305 IF (A<>B AND A<>C AND B<>C) OR BET
(3)=0 THEN 320
310 IF A=B OR B=C OR A=C THEN WIN=WIN+
(BET(3)*5):POSITION 6,21:? WIN;
315 POSITION 3,17:? "+";
320 IF A=B AND B=C THEN GOSUB 2000
330 IF A=B THEN G=A:GOSUB 2100
335 IF A=C THEN G=A:GOSUB 2100
340 IF B=C THEN G=B:GOSUB 2100
350 IF D<11 THEN GOSUB ((D-3)*10)+2200
352 IF D>10 THEN GOSUB (ABS(D-18)*10)+
2200
360 IF D=17 THEN GOSUB 2210:IF D=18 TH
EN GOSUB 2200
370 M=M+WIN:POSITION 7,20:? M;
372 IF M=0 THEN 420
373 IF M>=5000 THEN 5000
375 IF WIN>0 THEN 400
380 GOSUB 2600
382 POSITION 1,0:? "SORRY";
385 IF STRIG(0)=0 THEN POSITION 1,0:?
"     ";:GOTO 110
```

```
390 IF STICK(0)=13 THEN 500
395 GOTO 382
400 GOSUB 2500
402 POSITION 1,0:? "WINNER";
405 IF STRIG(0)=0 THEN POSITION 1,0:?
"      ";:GOTO 110
410 IF STICK(0)=13 THEN 500
415 GOTO 402
420 ? "I'M SORRY, BUT YOU HAVE GONE B
ROKE":? "IF YOU WISH TO START AGAIN WI
TH A"
430 ? "NEW BANKROLL PRESS START, TO QU
IT":? "PRESS SELECT"
440 IF PEEK(53279)<>6 AND PEEK(53279)<
>5 THEN 440
450 IF PEEK(53279)=6 THEN COUNT=0:GOTO
 90
460 IF PEEK(53279)=5 THEN ? "GOODBYE A
ND BETTER LUCK NEXT TIME":END
500 ? "KIT IS A WISE GAMBLER WHO KNOWS
 WHEN    TO QUIT."
510 ? :? "THANK YOU FOR PLAYING AND GO
OD LUCK    TO YOU THE NEXT TIME."
520 ? :? "GOODBYE.":END
1000 GRAPHICS 0:POKE 752,1:POKE 82,1:S
ETCOLOR 2,12,12:? "
       USE JOYSTICK";
1002 SETCOLOR 1,12,0:SETCOLOR 4,12,12
1005 ? "                                TO
MOVE BET"
1010 ? "                                POI
NTER"
1020 ? :? "PAYS 216-1  PAYS 18-1   TOT
AL   PAYS"
1025 ? "┌──────────┬──────────┬──────
──────┐"
1030 ? "|  3-ONES  ||  2-ONES  ||  3
  216-1|"
1035 ? "|  3-TWOS  ||  2-TWOS  ||  4
   72-1|"
1040 ? "|  3-THREES||  2-THREES||  5
   36-1|"
1045 ? "|  3-FOURS ||  2-FOURS ||  6
   21-1|"
1050 ? "|  3-FIVES ||  2-FIVES ||  7
   14-1|"
1055 ? "|  3-SIXES ||  2-SIXES ||  8
   10-1|"
1060 ? "└──────────┘└──────────┘|  9
    9-1|"
1065 ? "┌──────────────────────┐|  10
    8-1|"
1070 ? "|  HI (OVER 11)      1-1||  11
    3-1|"
1075 ? "|  LO (UNDER 10)     1-1||  12
    9-1|"
1080 ? "|  ANY 3 OF KIND    36-1||  13
   10-1|"
1085 ? "|  ANY 2 OF KIND     6-1||  14
   14-1|"
1090 ? "└──────────────────────┘|  15
   21-1|"
1095 ? "┌──────────┐┌──────┐|  16
   36-1|"
1100 ? "|CASH:          || ROLL||  17
   72-1|"
1105 ? "|WIN:           || DICE||  18
  216-1|"
1110 ? "└──────────┘└──────┘└──────
──────┘"
1120 GOTO 110
1200 POKE 752,1:FOR I=0 TO 31:BET(I)=0
:NEXT I
1210 FOR I=6 TO 11:POSITION 2,I:? "  "
;:POSITION 14,I:? "  ";:NEXT I
1220 FOR I=14 TO 17:POSITION 2,I:? "
";:NEXT I
1230 FOR I=6 TO 21:POSITION 26,I:? "
";:NEXT I
1240 POSITION 19,20:? " ↓← ";
1270 WIN=0:POSITION 6,21:? "        "
1280 B=0:GOSUB 1500
1290 RETURN
1500 IF B<0 THEN B=0:IF B>32 THEN B=32
1510 IF B=0 THEN POSITION 3,14:? ")↓←
";:X=2:Y=14
1511 IF B=1 THEN POSITION 3,14:? " ↓←)
↓← ";:X=2:Y=15
1512 IF B=2 THEN POSITION 3,15:? " ↓←)
↓← ";:X=2:Y=16
1513 IF B=3 THEN POSITION 3,16:? " ↓←)
";:POSITION 3,6:? " ";:X=2:Y=17
1514 IF B=4 THEN POSITION 3,17:? " ";:
POSITION 3,6:? ")↓← ";:X=2:Y=6
1515 IF B>4 AND B<9 THEN POSITION 3,B+
1:? " ↓←)↓← ";:X=2:Y=B+2
1516 IF B=9 THEN POSITION 3,10:? " ↓←)
";:POSITION 15,6:? " ";:X=2:Y=11
1517 IF B=10 THEN POSITION 3,11:? " ";
:POSITION 15,6:? ")↓← ";:X=14:Y=6
1518 IF B>10 AND B<15 THEN POSITION 15
,B-5:? " ↓←)↓← ";:X=14:Y=B-4
1519 IF B=15 THEN POSITION 15,10:? " ↓
←)";:POSITION 27,6:? " ";:X=14:Y=11
1520 IF B=16 THEN POSITION 15,11:? " "
;:POSITION 27,6:? ")↓← ";:X=26:Y=6
1521 IF B>16 AND B<31 THEN POSITION 27
,B-11:? " ↓←)↓← ";:X=26:Y=B-10
1522 IF B=31 THEN POSITION 27,20:? " ↓
←)";:POSITION 19,20:? " ↓← ";:X=26:Y=2
1
1523 IF B=32 THEN POSITION 27,21:? " "
;:POSITION 19,20:? "↓←";
1550 RETURN
2000 IF BET(A+3)=0 THEN RETURN
2010 WIN=WIN+(BET(A+3)*216):POSITION 6
,21:? WIN;
2020 POSITION 3,A+5:? "+";
2030 RETURN
2100 IF BET(G+9)=0 THEN RETURN
2110 WIN=WIN+BET(G+9)*18:POSITION 6,21
:? WIN;
2120 POSITION 15,G+5:? "+";
2130 RETURN
2200 IF BET(13+D)=0 THEN RETURN
2202 WIN=WIN+BET(13+D)*216:POSITION 6,
21:? WIN;
2204 POSITION 27,D+3:? "+";
2206 RETURN
2210 IF BET(13+D)=0 THEN RETURN
2212 WIN=WIN+BET(13+D)*72:POSITION 6,2
1:? WIN;
2214 POSITION 27,D+3:? "+";
2216 RETURN
2220 IF BET(13+D)=0 THEN RETURN
2222 WIN=WIN+BET(13+D)*36:POSITION 6,2
1:? WIN;
2224 POSITION 27,D+3:? "+";
2226 RETURN
2230 IF BET(13+D)=0 THEN RETURN
2232 WIN=WIN+BET(13+D)*21:POSITION 6,2
1:? WIN;
2234 POSITION 27,D+3:? "+";
2236 RETURN
2240 IF BET(13+D)=0 THEN RETURN
2242 WIN=WIN+BET(13+D)*14:POSITION 6,2
1:? WIN;
2244 POSITION 27,D+3:? "+";
2246 RETURN
2250 IF BET(13+D)=0 THEN RETURN
2252 WIN=WIN+BET(13+D)*10:POSITION 6,2
1:? WIN;
2254 POSITION 27,D+3:? "+";
2256 RETURN
2260 IF BET(13+D)=0 THEN RETURN
2262 WIN=WIN+BET(13+D)*9:POSITION 6,21
:? WIN;
2264 POSITION 27,D+3:? "+";
2266 RETURN
2270 IF BET(13+D)=0 THEN RETURN
2272 WIN=WIN+BET(13+D)*8:POSITION 6,21
:? WIN;
2274 POSITION 27,D+3:? "+";
2276 RETURN
2500 FOR I=1 TO 10
2505 FOR S=40 TO 90 STEP 5
2510 SOUND 0,S,10,10
2530 NEXT S
2540 FOR S=90 TO 40 STEP -5
2550 SOUND 0,S,10,10
2570 NEXT S
2580 NEXT I
```

```
2590 SOUND 0,0,0,0:RETURN
2600 SOUND 0,200,10,10
2620 FOR DELAY=1 TO 100:NEXT DELAY
2630 SOUND 0,241,10,10
2640 FOR DELAY=1 TO 150:NEXT DELAY
2650 SOUND 0,0,0,0:RETURN
2700 FOR I=1 TO 20
2710 FOR S=0 TO 50 STEP 20
2720 SOUND 0,S,8,15
2730 NEXT S:SOUND 0,0,0,0
2740 NEXT I
2750 POKE 77,0:RETURN
5000 ? CHR$(125);"THIS MACHINE HAS NO
MORE MONEY."
5010 ? :? "IF YOU WISH TO CASH IN YOUR
 BANKROLL";? "AND PLAY AGAIN AFTER THE
 MANAGE-"
5020 ? "MENT HAS REFILLED IT, THEN PRE
SS";? "START.  TO QUIT PRESS SELECT."
5030 IF PEEK(53279)<>6 AND PEEK(53279)
<>5 THEN 5030
5040 IF PEEK(53279)=5 THEN ? "THANK YO
U FOR PLAYING";? "GOODBYE!":END
5050 COUNT=0:GOTO 1000
10000 POSITION X,Y:? "  ";
10001 POSITION X,Y+1:? "  ";
10002 POSITION X,Y+2:? "  ";
10003 RETURN
10010 POSITION X,Y:? "  ";
10011 POSITION X,Y+1:? "  ";
10012 POSITION X,Y+2:? "  ";
10013 RETURN
10020 POSITION X,Y:? "  ";
10021 POSITION X,Y+1:? "  ";
10022 POSITION X,Y+2:? "  ";
10023 RETURN
10030 POSITION X,Y:? "  ";
10031 POSITION X,Y+1:? "  ";
10032 POSITION X,Y+2:? "  ";
10033 RETURN
10040 POSITION X,Y:? "  ";
10041 POSITION X,Y+1:? "  ";
10042 POSITION X,Y+2:? "  ";
10043 RETURN
10050 POSITION X,Y:? "  ";
10051 POSITION X,Y+1:? "  ";
10052 POSITION X,Y+2:? "  ";
10053 RETURN
```

SOFTWARE REVIEW: STEREO GRAPHICS PACKAGE

APX
ATARI, INC.
$22.95 32K Disk

by Brian Moriarty

Clyde Spencer's **Stereo 3-D Graphics** (APX-20087) is the first ATARI graphics package designed to generate true stereoscopic images. Used in conjunction with a regular still camera, this package lets you create high-resolution wire-frame stereograms that can be viewed with any standard stereoscope. It is also capable of producing side-by-side stereo image pairs on your TV screen that can, with practice, be viewed directly in 3-D.

The package contains six separate utility programs, all written in ATARI BASIC. The Graphics File Generator (GFILEGEN), Digitizer (DIGITIZE), Z-Coordinate Changer (ZCHANGE) and File Merger (FMERGE) programs let you build complex graphics models of practically any shape or size and store them as data files on a disk. The other two programs, MONO3D and STEREO3D, are used to display and photograph your 3-D models. Ten sample models are also included on the program disk. These are useful for learning how to manipulate the screen images.

Features

GFILEGEN is a general-purpose modelbuilding utility. Like the other programs in this package, it uses the center of your TV screen as the zero reference point for the X, Y and Z axes. To use it, you type in a starting coordinate for your wire-frame model, followed by the coordinates of each "draw-to" point you want the computer to remember. You can enlarge, reduce, move and/or rotate your models on the screen before storing the final version onto a disk. All of these editing functions can also be performed on a previously stored model.

The DIGITIZE utility provides a simpler method for building complicated models. You place a transparency with the desired outline over the TV screen and trace the X and Y coordinates of each draw-to point with a cursor controlled by a pair of game paddles. The Z-coordinate of each point is entered manually via the console. The models generated by DIGITIZE can then be editing with the GFILEGEN program.

ZCHANGE's sole function is to change every Z-coordinate in a stored graphics file to a single new value specified by you. It is used to vary the spatial depth of your models or to define the relationships between the elements of isoplethic (layered) models such as topographic maps.

FMERGE combines any number of sub-files into a new master file with its own filename. You can superimpose different incarnations of the same model or put several models onto one display screen.

The MONO3D program transforms an array of 3-dimensional coordinates into a single 2-dimensional screen display. It accepts coordinates either directly from the console or from a graphics file stored on a disk. You can specify the screen resolution (high or medium), field of view (1.2 to 179.9 degrees) and your viewing position relative to the center of the TV screen. Models are presented as white outlines on a green background and may be viewed from either side, above, below, behind or even inside. Off-screen lines are automatically clipped to avoid overrange errors, but there is no hidden line removal.

STEREO3D is the heart of the entire software package. This display utility incorporates all the important features of MONO3D plus the additional image processing needed to get left and right stereo pairs for true 3-D viewing. Two separate display modes are offered, high-res and medium-res.

In the high-res mode, STEREO3D draws the left-eye image as white lines on a green background. It then asks whether you want the screen cleared before drawing the right-eye image. This option gives you an opportunity to photograph the left and right images separately for later viewing with a stereoscope.

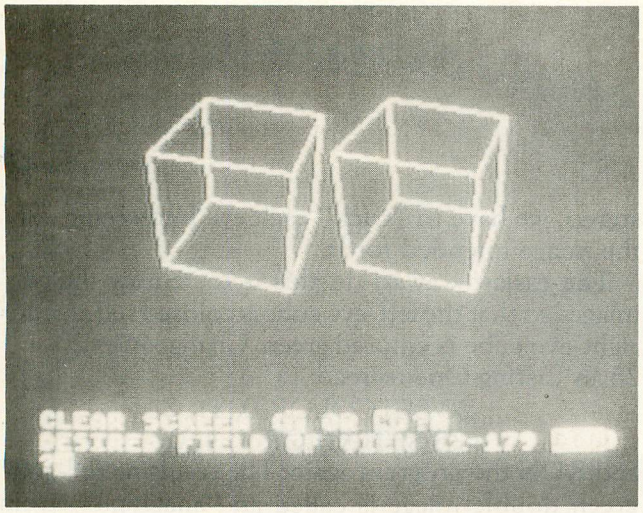
If you choose not to clear the screen, STEREO3D will draw the right-eye image directly over the left image. By specifying the correct separation and viewing distance, you can obtain side-by-side screen images that are directly viewable in 3-D. To accomplish this, you must learn how to converge your eyes so that the two outlines "fuse" together into a single, 3-dimensional whole. This crosseyed viewing takes practice; not everyone will be able to master it.

The medium-res display mode presents the left and right views as green and red outlines on a bright white background. If you look at this dual image through a pair of anaglyphic viewing glasses (with a red filter over the left eye and a green filter over the right), you can partially cancel the optical crosstalk between the two screen images. This makes it somewhat easier to fuse the double outline into a 3-D picture.

The STEREO3D medium-res mode will **not** produce a "3-D movie" effect on your TV screen! It's designed to help you acquire the special eye training you need to manually converge stereo image pairs. If you're interested in creating 3-D pictures that can be viewed with little or no training, read my tutorial on stereoscopic ATARI graphics elsewhere in this issue.

Using the package

The documentation supplied with **Stereo 3-D**



Graphics offers a good description of the stereo viewing process and the special problems involved with on-screen 3-D viewing. It includes directions for making your own anaglyphic glasses and helpful tips for photographing the TV screen, as well as an exhaustive bibliography of references. However, the sections dealing with the actual operation of the six programs are a little puzzling in some areas. The best way to learn the system is to boot in STEREO3D and start playing with the sample graphics files.

The drawing speed depends on the complexity of the graphics data. Simple isometric shapes (cubes, etc.) require only a few seconds to generate. Larger models with curved edges can take several minutes of processing time.

Although the individual utilities are easy to use, the overall structure of the package makes model building a relatively clumsy procedure. You have to save your current graphics file onto a disk before loading in each utility. It's a good idea to keep a record of the filenames you've used for each model segment, or you'll find yourself returning to DOS every few minutes for an update. A second disk drive would also be very convenient.

APX's **Stereo 3-D Graphics** is a great hands-on introduction to the mechanics of stereo vision, coordinate geometry and wire-frame modelbuilding. It doesn't have the features and speed of more sophisticated machine-language graphics programs, but for a list price of only $17.95 it's got to be one of the best software values around. You'll find it at selected ATARI dealers or you can order direct from the ATARI Program Exchange. □

Stereo Graphics Tutorial

by Brian Moriarty

3-D movies are back! The last several months have seen the release of several new stereoscopic films, and many more are in production as you read this article. Audiences across the country are screaming with delight as spears, flames and assorted animals leap off the theater screen.

Wouldn't it be great to enjoy the lifelike realism of 3-D in a computer game or adventure? Imagine stalking the Colossal Cave with your 3-D glasses, ducking to avoid the giant bats as they swoop off the TV screen into your face!

This tutorial will show you a low-cost way to experiment with real stereo graphics on your ATARI. Using the basic techniques described below, you'll be able to create floating 3-D outlines that seem to extend deep into your TV set or hang a foot in front of your nose. All you need is a good color TV set or monitor and an easy-to-make pair of 3-D viewing glasses.

How we see in 3-D

The illusion of depth is created by a number of perceptual factors. Size, shading and peripheral vision all help our brains to determine how far away an object is. For distances closer than 50 feet or so, one of the most important factors is the **binocular parallax** caused by the physical separation of our left and right eyes.

Each eye sees a slightly different perspective of the objects in front of us. Very close objects show a large difference in perspective, while more distant objects show less of a difference. Our brain analyzes the difference between our left- and right-eye views and builds up a single, 3-dimensional image of the world. Therefore, to give our computer graphics a 3-D movie effect, we must recreate the binocular parallax our brain needs to perceive stereo.

Suppose we want to put a 3-D cube on the screen. In order to produce the stereo effect, we'll have to present each eye with a slightly different perspective of the cube. The left eye should be able to see more of the left side of the cube, and the right eye should see more of the right.

Enter and RUN the program under **Listing 1** and the computer will draw a pair of cubes, one on top of the other. You'll notice that the edges of the two cubes don't line up exactly. This offset recreates the different perspectives your left and right eyes would see if the cube were truly 3-dimensional.

Unfortunately, you can't see the cube drawn by **Listing 1** in 3-D because your brain doesn't know how to interpret the confusing double image. To give your brain the information it needs to perceive stereo, we have to "trick" each eye into seeing only the image intended for it.

The easiest way to do this is to redraw the two images so that the left-eye cube is colored red and the right-eye cube is colored green. Change lines 20 and 25 in Listing One to read:

```
20 SETCOLOR 0, 11, 2
25 SETCOLOR 1, 4, 0
```

and RUN the program again. The resulting red-and-green double image is called an "anaglyph" (from the Greek word **anagluphein**, to carve in relief). If you look at this anaglyph through a pair of stereo viewing glasses (with a red filter over your left eye and a green filter over your right), you'll see the double cube fuse itself into a spectacular 3-D image!

Here's what happens when you put on the glasses: the red filter makes the red (left) outline appear much lighter to your left eye than the green (right) outline. The green filter does the same thing to the green outline. Since each eye is now seeing its own perspective-corrected cube, your brain interprets the double cube as a composite stereo image.

Important

Don't be disappointed if you can't see the 3-D effect right away. In order for the system to work properly, the colors on the screen have to be carefully adjusted to match the color filters in the 3-D glasses. Otherwise the left image will "leak" over into your right eye, and vice versa. The following procedures will help you to achieve the best stereo imaging with your TV set and glasses:

1. Turn the color intensity and contrast controls on your TV set below their normal settings. This will suppress the inevitable color fringes and ghosts around the outlines which can ruin the 3-D effect.
2. Close one eye and look at the screen through the **red** color filter. Adjust the tint control on your TV set so that the **green** lines appear as dark as possible. Then look through the green filter and try to darken the red lines. By switching back and forth between the filters, you should be able to find one position for the tint control that results in optimum cancellation of both colors.
3. The SETCOLOR values I've chosen for this demonstration should work well in the majority of cases. However, if you can't obtain satisfactory cancellation of the double image by adjusting your tint control, try "tuning" the SETCOLOR values yourself.
4. Don't sit close to your TV screen when viewing the 3-D cube. If you step back four or five feet, the stereo effect will appear much clearer and deeper.

Take off your 3-D glasses and study the structure of the double cube. Notice that the left and right edges of the two cubes are almost exactly superimposed. Because the difference in positioning is so small, your brain interprets these edges as existing on the plane of the screen. They do not seem to "go into" or "come out of" the TV set.

Now look at the parts of the cube that seem to float in front of the screen. The red (left) outline is placed well to the right of the green (right) outline. These lines are said to have **positive parallax** because they make your eyes converge inward. Similarly, the edges that float behind the screen have the red line on the left and the green line on the right. These lines have **negative parallax**. By changing the spacing and relative positioning of the red and green lines, you can define the kind of depth each part of the stereo image will exhibit.

Use the following rules to "build" your own left and right stereo views:

a. If you want an edge to appear on the plane of the TV screen, put the red and green lines right next to each other.
b. If you want an edge to float in front of the TV screen, put the green line on the left and the red line on the right.
c. If you want an edge to float behind the screen, put the red line on the left and the green line on the right.
d. The further apart the red and green lines are placed, the more an edge will seem to float inward or outward.

Stereo animation

Once you've successfully drawn a few anaglyphic outlines, you'll want to try animating your 3-D images. It isn't easy. You have to deal with two complete pictures at once, simultaneously handling both their relative positioning and motion.

Listing 2 is a primitive example of stereo animation. It uses color register rotation and data-line addressing to create a "spinning stick" on your TV screen. The bottom half of the stick seems to come off the screen while the top half recedes into the TV set. Note how the spacing of the red and green lines varies as the stick changes its orientation.

When entering **Listing 2**, be sure to type the line numbers exactly as shown or the program will almost certainly crash.

Lines 20-25 Main subroutine to fetch coordinate data and draw lines.

Lines 30-40 Initialize program variables. If you have to adjust the program colors to get better stereo imaging, change the values of RED and GREEN in line 40 by using this formula:

```
VALUE=16*(COLOR CODE)+(LUMINANCE VALUE).
```

Lines 45-55 Initialize animation routine and draw the first pair of lines.

Lines 55-70 Move the double-stick around the central axis and erase old lines.

Line 75 Complete the rotation and reset the coordinate address pointer for another round.

Lines 80-93 Coordinate data.

The information presented in this tutorial should be enough to get you started on your own stereo graphics applications. If you have any questions or problems, write to me in care of the magazine and I'll be glad to help out (include a SASE!). Who knows? Maybe you'll be one of the first to release an action-packed 3-D arcade game and send the software engineers at ATARI scurrying for their notebooks! □

```
5 REM * LISTING 1 *
10 REM * STEREO CUBE DEMO PROGRAM *
15 GRAPHICS 23
20 SETCOLOR 0,0,8
25 SETCOLOR 1,0,8
30 FOR CUBE=1 TO 2
35 COLOR CUBE
40 PLOT 40+3*CUBE,30
45 FOR EDGE=1 TO 15
50 READ X,Y
55 DRAWTO X,Y
60 NEXT EDGE
65 NEXT CUBE
70 REM * COORDINATE DATA FOR CUBES *
75 DATA 85,40,85,85,43,72,43,30,75,23
80 DATA 115,30,85,40,85,85,115,72,115
85 DATA 30,115,72,75,61,75,23,75,61
90 DATA 43,72,91,40,91,85,46,72,46,30
95 DATA 69,23,118,30,91,40,91,85,118
100 DATA 72,118,30,118,72,69,61,69,23
105 DATA 69,61,46,72
110 GOTO 110
```

```
5 REM * LISTING 2 *
10 REM * STEREO ANIMATION DEMO *
15 GRAPHICS 23:GOTO 30
20 START=START+Z1:RESTORE START
25 READ X1,Y1,X2,Y2:PLOT X1,Y1:DRAWTO
X2,Y2:RETURN
30 Z1=1:Z2=2:Z3=3:Z4=4:LINE=20:X=25
35 CR0=708:CR1=709:CR2=710
40 BLK=0:RED=64:GRN=178:BASE=80
45 START=BASE-Z1:POKE CR0,GRN:POKE CR1
,RED:POKE CR2,BLK
50 COLOR Z1:GOSUB LINE:COLOR BLK:RESTO
RE 93:GOSUB X:COLOR Z2:GOSUB LINE
55 FOR MOVE=Z1 TO Z4:Q=MOVE-Z1
60 COLOR Z3:GOSUB LINE:COLOR BLK:RESTO
RE BASE+Z3*Q:GOSUB X:POKE CR0,BLK:POKE
 CR1,GRN:POKE CR2,RED
65 COLOR Z1:GOSUB LINE:COLOR BLK:RESTO
RE 81+Z3*Q:GOSUB X:POKE CR0,RED:POKE C
R1,BLK:POKE CR2,GRN
70 COLOR Z2:GOSUB LINE:COLOR BLK:RESTO
RE 82+Z3*Q:GOSUB X:POKE CR0,GRN:POKE C
R1,RED:POKE CR2,BLK:NEXT MOVE
75 COLOR Z3:RESTORE BASE:GOSUB X:COLOR
 BLK:RESTORE 92:GOSUB X:GOTO 45
80 DATA 38,47,116,47
81 DATA 40,49,114,45
82 DATA 44,51,112,43
83 DATA 48,53,108,41
84 DATA 54,55,102,39
85 DATA 60,57,96,37
86 DATA 68,59,88,35
87 DATA 78,61,78,33
88 DATA 88,59,68,35
89 DATA 96,57,60,37
90 DATA 102,55,54,39
91 DATA 108,53,48,41
92 DATA 112,51,44,43
93 DATA 114,49,40,45
```

Make your own 3-D glasses

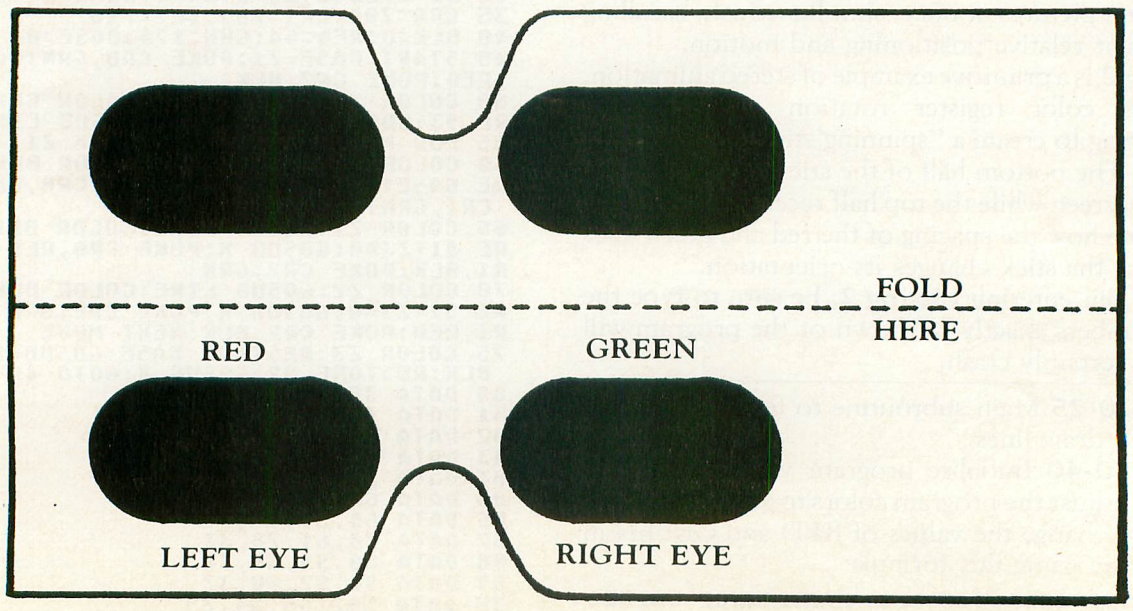
It's easy to make a pair of 3-D glasses for viewing the anaglyphs in this tutorial. You'll need to get two small pieces of plastic color filter material, one tinted red and the other tinted green or blue/green. These are available at many art, photographic and theatrical supply stores. Be sure the red and green filters are of approximately the same density.

Next, cut a piece of thin cardboard into the shape of a pair of glasses, using the pattern in the illustration as a guide. Sandwich the red and green filters inside the folded cardboard and staple the assembly together as shown. You may find it helpful to use **two** layers of filter material in each side of your 3-D glasses instead of just one. This will give you better separation betwen the left and right stereo views.

If you have a pair of anaglyphic glasses left over from a 3-D movie or comic book, you can probably use them to view your stereo ATARI graphics. *(Note: the polarized glasses from the movies Comin' At Ya! and Parasite will not work. —Ed.)* Be sure the filters are dense enough to properly cancel the cross-talk between the two images.

You must wear the 3-D glasses so that the red filter covers your left eye and the green filter covers your right eye. Otherwise you'll get an unpleasant inside-out stereo image. □

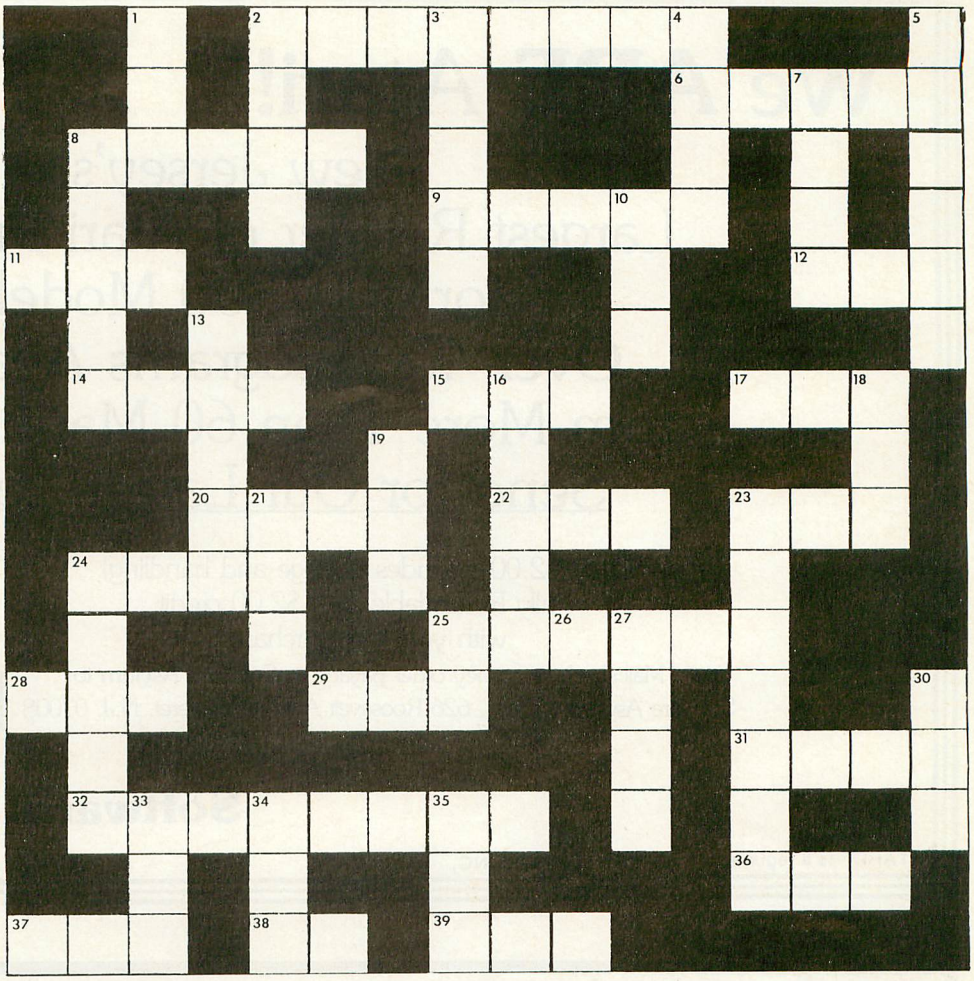
3-D GLASSES PATTERN



BATARI BASIC CROSSWORD PUZZLE

By Peggy Knoble

Being an avid crossword puzzle fan, I comprised a crossword that uses ATARI BASIC commands. You'll have to wait until the next issue for the answers. Good luck.



ACROSS:

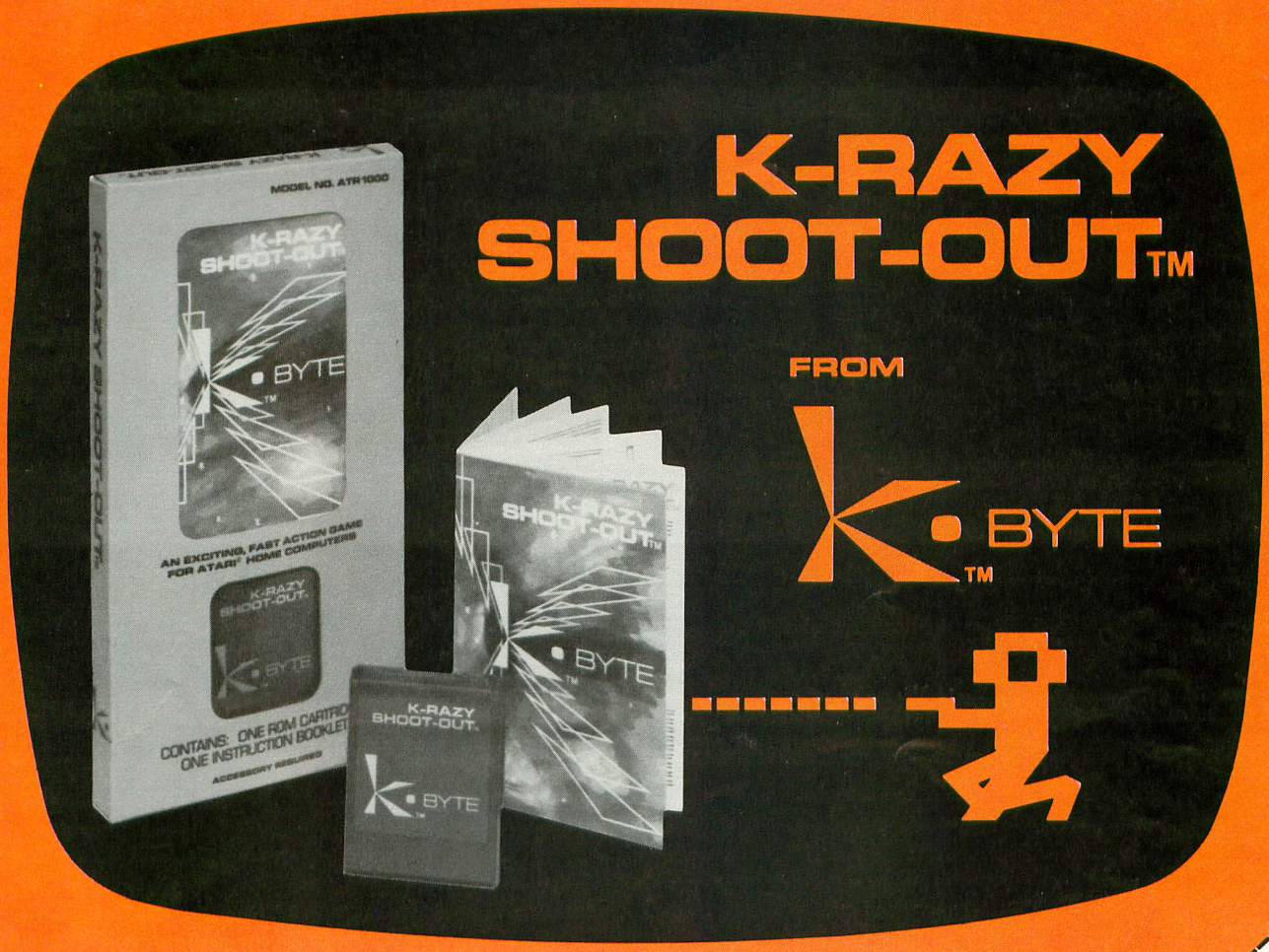
2 — Store hue and luminance data in a color register.
6 — Retrieve a program in untokenized format.
8 — Opposite of 8 DOWN.
9 — Has a variable for pitch.
11 — Disk Operating System.
12 — Returns a "1" if statement is untrue.
14 — Information (used with 4 DOWN).
15 — Reserves a specified amount of RAM for an array or string.
17 — Returns numeric equivalent of a string.
20 — Returns base 10 logarithm of an expression.
22 — Removes the loop variable from the GOSUB stack.
23 — Executes a program.
24 — Insert a byte into memory.
25 — Returns the condition for a specified device.
28 — Returns the amount of usable RAM remaining.
29 — Exits from BASIC.
31 — Readies a file for input.
32 — The ATARI computer has basically 8 of these modes.
36 — Stops program execution.
37 — See 15 ACROSS.
38 — Used in the FOR/NEXT loop.
39 — This function returns a random number.

DOWN

1 — Returns the exact value of an expression.
2 — Causes a program to be recorded on a diskette.
3 — Opposite of 31 ACROSS.
4 — Used in conjunction with 14 ACROSS.
5 — Statement creates a line between two points.
7 — Used with IF.
8 — Retrieves a program from cassette.
10 — Erases user RAM.
13 — Returns the position of a joystick.
16 — Causes the computer to ask for keyboard data.
18 — Returns the size of a string.
19 — Returns -1 or zero or +1.
21 — Assigns a value to a specified variable.
23 — Allows data to be read more than once.
24 — Returns paddle trigger status.
25 — 2 ACROSS abbreviated.
27 — Takes control of a program in case of error.
30 — Logical operator.
33 — A statement that has no effect on program execution.
34 — Causes output of a single byte of data.
35 — Opposite of 15 across.

ANALOG

READY FOR A REAL CHALLENGE?



It's a battle of wits and fast action, this challenging game in ROM form for *Atari® 400™ or Atari® 800™ Personal Computer Systems. You're trapped in an Alien Fortress. Your mission is to clear your sector of the Alien Droids and progress to the next sector. The first sector seems easy enough...BUT–WATCH-OUT...as you progress from sector to sector, the Droids become more numerous and aggressive. A great challenge and fun too!

K-RAZY SHOOT-OUT is destined to be a favorite of Atari enthusiasts across the country with features like:

- Solid-state ROM pack
- Seven progressive levels of play
- Every game different
- Bonuses and penalties
- Full-color T.V. graphics and space-age sound
- Full-color instruction booklet
- Millions of variations
- A constant challenge

[WARNING: Playing the game could be HABIT forming]

Now at your Atari Dealer!

Coming soon:

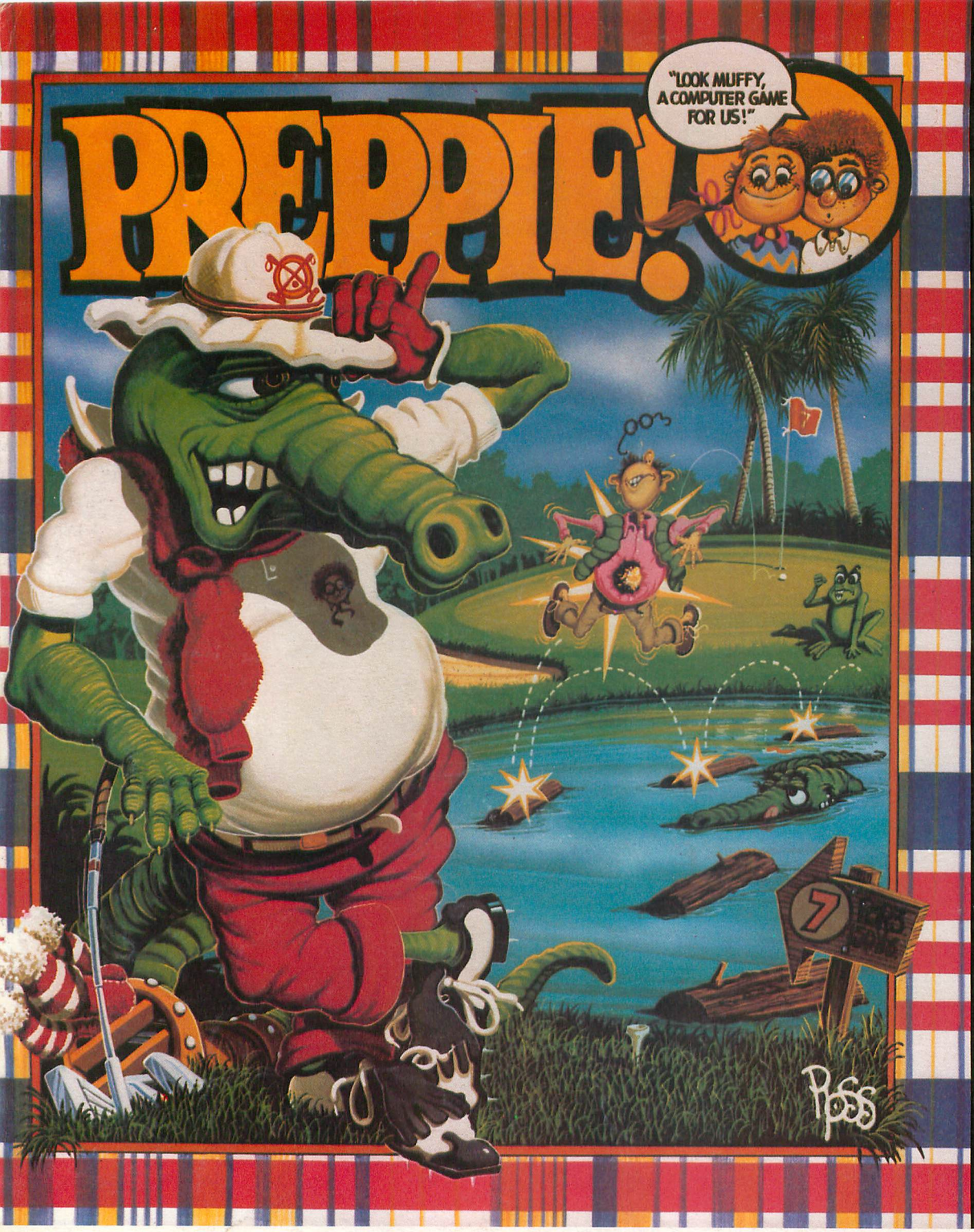
- K-DOS™
- K-STAR PATROL™
- K-RAZY KRITTERS™

*Atari is the Registered Trademark of Atari, Inc.

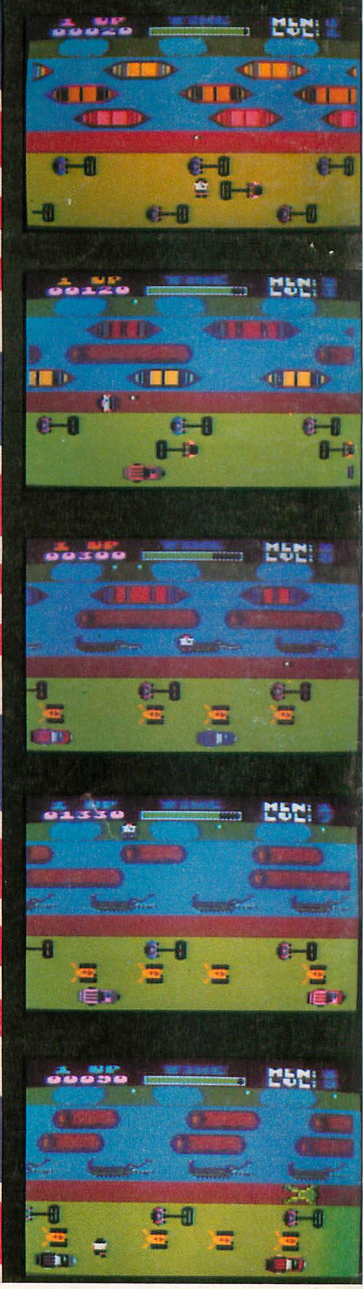
FOLD & TEAR

If not available in your area, send a check or money order directly to:
K-BYTE • P.O. Box 456 • 1705 Austin Street • Troy, Michigan 48099
Toll Free Phone Orders: **1-800-521-0560** • In Michigan: **(313) 524-9878**
☐ $49.95, plus $1.50 shipping and handling, Michigan residents add 4% sales tax.
☐ $1.00 for instruction book only.

NAME _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
Please allow 3 weeks for delivery.
☐ MasterCard ☐ VISA
CHARGE CARD NO. _____
EXPIRATION DATE _____



by Russ Wetmore



Good Golly! What A GAME!

The original arcade game **PREPPIE!** will give you hours of toe-tickling fun. **You** take the controls, moving your young prepster about on the golf course. And what a golf course! It's unlike any you have ever played on before. There are dangers **everywhere,** and only you can keep the little Ivy Leaguer from a fate worse then Penn State!

PREPPIE! is written in state-of-the-art machine language, is joystick compatible and fully utilizes the Atari's sound and graphics capabilities. Quite frankly, it will give you the most fun you can have on an Atari microcomputer.

PREPPIE! is available at fine computer, book, and hobby stores everywhere.

ATARI 16K TAPE — $29.95
ATARI 32K DISK — $29.95

An exclusive game from

ai Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.

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

Published by ADVENTURE INTERNATIONAL
a division of Scott Adams, Inc.
BOX 3435 • LONGWOOD, FL 32750 • (305) 862-6917

"My Atari never did things like this before!"
—Holister Townsend Wolfe

"I had so much fun I almost blew my doughnuts."
—Theodore Boston III

"I haven't had this much fun since Buffy and I went to Princeton for the weekend."
—Martha Vineyard