NO. 40
MARCH 1986

U.S.A. $3.00
CANADA $4.00

THE #1 MAGAZINE FOR ATARI® COMPUTER OWNERS

ANALOG COMPUTING™

BOWMAN

APPLICATIONS ISSUE

WITH: Atari Can Mean Business
Micro-Mail • VisiPlot
Mouser • IRA
and *more!*

0 74470 12385 03

HIPPO DIGITIZERS AND MORE...

HIPPO ST SOUND DIGITIZER

Digitally sample, modify, and play back high-quality sounds. Experiment with adjustable real-time echos, phase shifts, voice synthesis and recognition, and fourier analysis. Adjustable sample and playback rates. Software includes many fun demonstrations including a real-time graphic oscilloscope. Useful for analog lab measurement and control too. Includes software (executable and source), hardware, microphone, power supply, and instructions. **$139.95**

THE SPECS

8 bit A/D and D/A. ■ 2.5µs flash A/D conversion rate allows sampling up to 400,000 samples per second. ■ 1 volt peak-to-peak signal and source. ■ Hardware plugs into printer port for high speed data transfer rate.

HIPPOVISION VIDEO DIGITIZER

Digitize from any standard composite video source (e.g. TV camera, VCR, TV tuner, etc...). Frame grabber "snaps" color picture in a single video frame. Creates Neo files. Perfect for game-designers and artists. Send video pictures over the phone (requires a modem)! Includes hardware and software. Contact us for high-quality color printers and ST color printer software drivers too. **Call for price.**

THE SPECS

256 × 256 × 9 bit resolution (3 bits per color). ■ Standard NTSC 1 volt peak-to-peak composite sync signal. ■ Software color "voting" system picks optimum 16-color ST palette according to image. ■ Hardware plugs into printer port for high speed data transfer rate.

OUR OTHER ST PRODUCTS

- *Professional Series* **HIPPOWORD™** — Advanced mouse-based word processor with powerful features: true multiple fonts, column editing, rulers, left/right justification, boldface, etc... **$89.95**
- *Professional Series* **HIPPOCONCEPT™** — The original ST idea processor leads you from vague ideas through drafts to clear, well-organized business plans, term papers, proposals, etc... **$89.95**
- **HIPPOSIMPLE™** — Powerful, flexible and easy-to-use database. Mouse-based screen editor, 16-level sort, merge, sum, data compression, 10+ programmable commands. **$49.95**
- **HIPPO DISK UTILITIES™** — Recovers deleted or lost files, reconstructs damaged disks; find files, edit RAM, files, and track & sectors. Floppies and hard disks. Disk and memory map. **$49.95**
- **HIPPOBACKGAMMON™** — Sophisticated A.I. algorithm for challenging play. Play yourself or create robot opponents and watch them play. Learn A.I. theory. Full-color or B&W animation. **$39.95**
- **HIPPOSPELL™** — Finds misspellings and suggests corrections. Word frequency stats, 30,000-word dictionary. User-defined dictionary. Works with most wordprocessors. **$39.95**
- **HIPPO RAMDISK™** — Runs programs 10 to 100 times faster. You choose 1K to 4 megabytes to serve as ultra-fast drive. RAMdisk automatically appears on boot-up. **$34.95**
- **HIPPO COMPUTER ALMANAC™** — It understands English, uses an A.I. parser and knows over 35,000 useful, intriguing facts. Many diverse topics from area codes to sports trivia, etc. **$34.95**
- **HIPPO JOKES & QUOTES™** — Insulting jokes, dirty jokes, one-liners, puns and quotations. Search by keyword/author. Select rating of PG, R or X. May not be suitable for children. **$34.95**
- **HIPPOART I™** — A rare collection of full-color masterpiece art. More than 30 detailed pictures in over 10 screens. Slide show program. Compatible with the ST's Neochrome. **$39.95**
- **HIPPO EPROM BURNER™** — Programs, reads, and verifies most EPROMs, including 27256 and 27512. Connects to printer port. EPROM cartridge boards available separately. **$139.95**
- **HIPPOCLEAN™** — Disk cleaning kit prevents head wear and protects your data. Kit includes head-cleaning disk, bottle of cleaning fluid and instructions. **$29.95**
- **HIPPOPIXEL™** — Create your own sprites and fonts. Select size and data format. Resize and pixel scroll command. Includes sample fonts and animation sequences. Color or B&W. **$39.95**

CIRCLE #101 ON READER SERVICE CARD

Dealer inquiries invited.
See your local dealer or contact Hippopotamus. VISA and Mastercard accepted. California residents add local sales tax. Please include $3 for shipping. Allow 1–2 weeks for delivery. Price, availability and specifications subject to change without notice.

HIPPOPOTAMUS SOFTWARE, INC.
985 University Avenue, Suite #12
Los Gatos, CA 95030
Phone: 408/395-3190 Telex: 650-284-0701

NO. 40 MARCH 1986

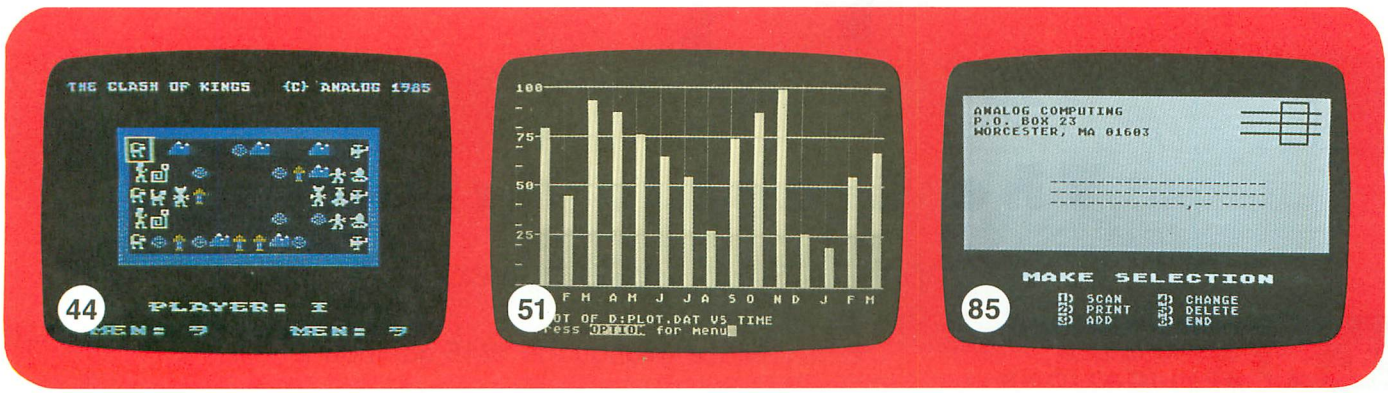
THE #1 MAGAZINE FOR ATARI® COMPUTER OWNERS™

ANALOG COMPUTING

FEATURES

REVIEWS

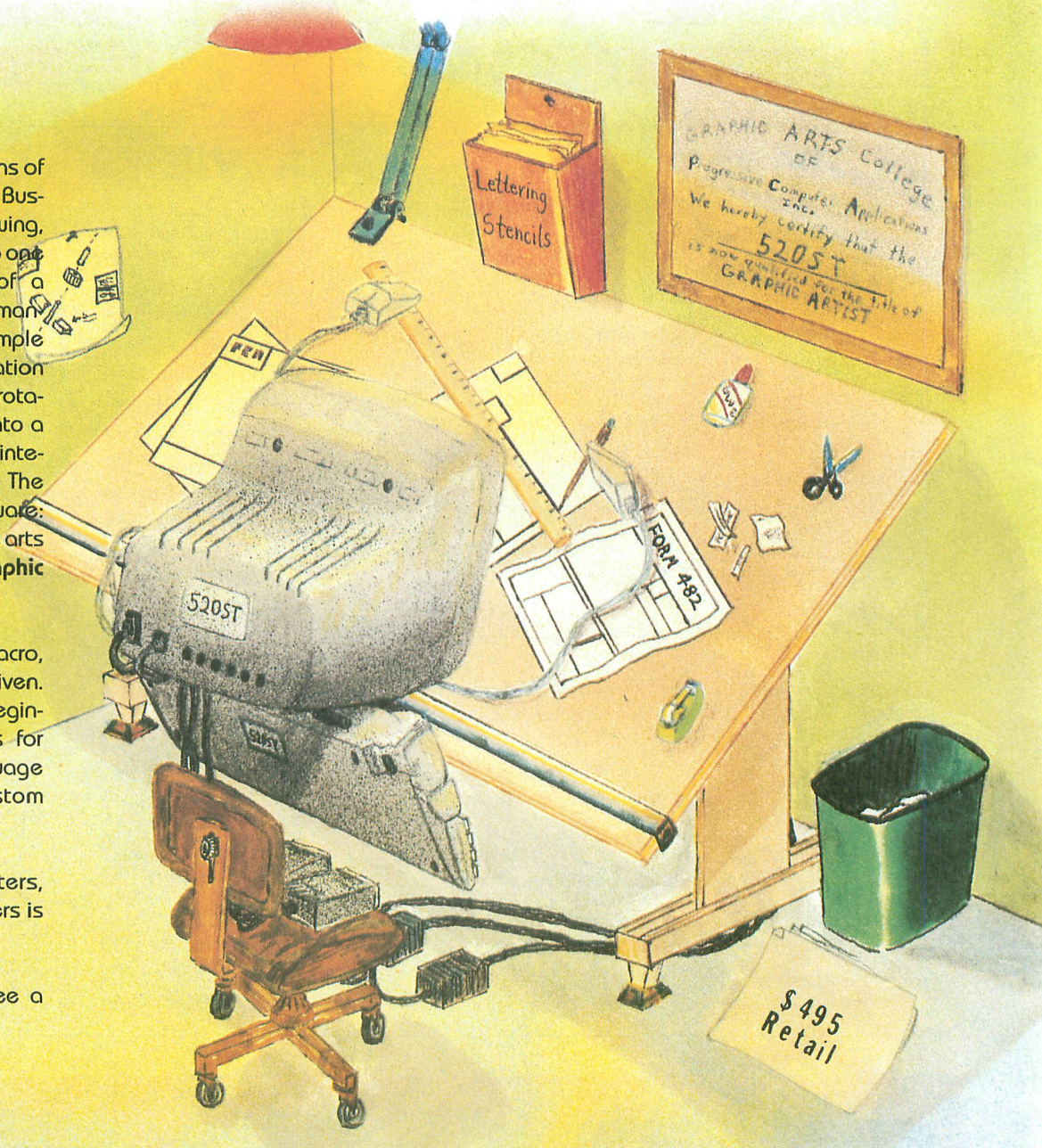
COLUMNS



ANALOG Computing (ISSN 0744-9917) is published monthly for $28 ($36 in Canada, $39 foreign) per year by ANALOG 400/800 Corp., 565 Main St., Cherry Valley, MA 01611. Second class postage paid at Worcester, MA and additional mailing offices. POSTMASTER: Send address changes to **ANALOG Computing**, P.O. Box 625, Holmes, PA 19043.

Graphic Arts

The Next Step in the Evolution of Software



Imagine combining the functions of Computer Aided Design (CAD), Business Graphics, free-hand drawing, and typesetting programs into one package. Include functions of a simple spreadsheet for data manipulation. Add abilities of a simple word processor for text annotation in various fonts, sizes, and rotations. Mix all these features into a single package instead of "integrating" separate packages. The result is a new breed of software: **Graphic Arts.** The only graphic arts program available: **The Graphic Artist.**

The Graphic Artist is menu, macro, command, and language driven. You read correctly. Menus for beginners, commands and macros for experts. And an optional language interpreter for creating custom applications.

Support for dot matrix printers, color plotters, and laser printers is standard, of course.

Impressed? Wait until you see a demo at a dealer near you!

The Graphic Artist

Graphic Arts has finally arrived.

PROGRESSIVE COMPUTER APPLICATIONS
2002 McAuliffe Drive
Rockville, Maryland 20851
(301) 340-8398

★Language $245 additional

CIRCLE #102 ON READER SERVICE CARD

ANALOG COMPUTING STAFF

U.S. newstand distribution by Eastern News Distributors, Inc., 1130 Cleveland Rd., Sandusky, OH 44870

WHERE TO WRITE

All editorial material (programs, articles, letters and press releases) should be sent to: Editor, **ANALOG Computing**, P.O. Box 23, Worcester, MA 01603.

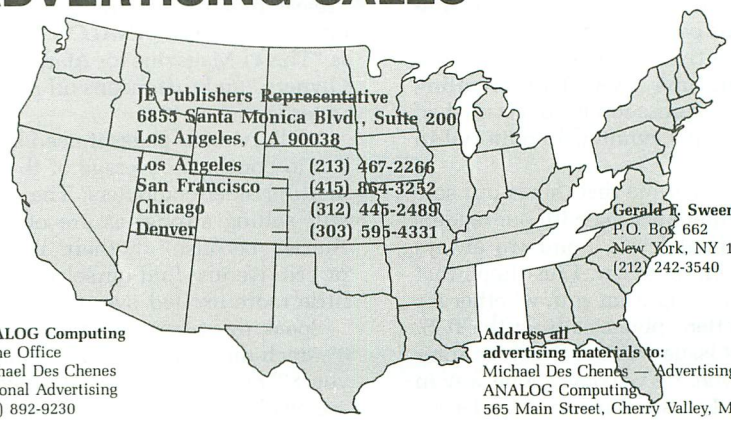
Correspondence regarding subscriptions, including problems and changes of address, should be sent to: **ANALOG Computing**, 100 Pine Street, Holmes, PA 19043, or call 1-800-345-8112 (in Pennsylvania, call 1-800-662-2444).

Correspondence concerning a regular column should be sent to our editorial address, with the name of the column included in the address.

We cannot reply to all letters in these pages, so if you would like an answer, please enclose a self-addressed, stamped envelope.

An incorrectly addressed letter can be delayed as long as two weeks before reaching the proper destination.

ADVERTISING SALES



JE Publishers Representative
6855 Santa Monica Blvd., Suite 200
Los Angeles, CA 90038

Los Angeles	(213) 467-2266
San Francisco	(415) 864-3252
Chicago	(312) 445-2489
Denver	(303) 595-4331

Gerald E. Sweeney & Assoc.
P.O. Box 662
New York, NY 10113
(212) 242-3540

ANALOG Computing
Home Office
Michael Des Chenes
National Advertising
(617) 892-9230

Address all advertising materials to:
Michael Des Chenes — Advertising Production
ANALOG Computing
565 Main Street, Cherry Valley, MA 01611

SUBSCRIPTIONS

All subscriptions should be addressed to:

ANALOG Computing
P.O. Box 625, Holmes, PA 19043
or call our toll-free number:
1-800-345-8112
(in PA 1-800-662-2444)

Foreign subscriptions must be made payable in U.S. funds.

Subscription prices in the U.S.: $28 for 1 year; $52 for 2 years; $79 for 3 years.

Subscription prices in Canada: $36 for 1 year; $62 for 2 years; $89 for 3 years.

Foreign subscriptions: $39 for 1 year; $72 for 2 years; $99 for 3 years.

Airmail is available for foreign subscriptions at an additional $50 per year.

Please allow four to six weeks for delivery of your first issue.

PERMISSIONS

AUTHORS

When submitting articles and programs, program listings should be provided in printed *and* magnetic form, if possible. Articles should be furnished as typed or printed copy in upper and lower case with double spacing. If a submission is to be returned, please send a self-addressed, stamped envelope.

EDITORIAL

It's been quite some time since I've had my thoughts and comments on the editorial page. I never realized how important this page was until Clayton Walnum, a one-time reader and contributor turned full-time **ANALOG Computing** employee, expressed his concern when I suggested not running an editorial in this issue.

He reminded me that this is our sole way of communicating with our readers on a personal level. . . and we always have *something* to say. I just hope that we keep hearing from you, whether it's through letters, phone calls, or the TCS.

Our last issue was our five-year anniversary issue. We've come a long way in that time. I was talking to a reader recently who made me understand just how important **ANALOG Computing** has been over the past five years.

We started publishing **ANALOG Computing** shortly after the 400/800 computers were put on the market. At that time, there was very little in the way of software and support, so we decided to make our own. I'm not trying to take credit for the success and survival of the Atari computer. If they weren't good machines to begin with, we wouldn't have made a difference.

However, as the reader I spoke with made me realize, Atari computer owners are among the most knowledgeable when it comes to the working and programming of their machines. And, of course, they're one of the biggest groups of hackers, anywhere. This reader credited us with making those statements fact. It would please me just to know that we may have helped save your Atari from the popular "closet death" that befell so many computers over the years.

As you've probably noticed recently, we've been expanding our coverage of the 520ST. Many readers have expressed their concern over the possibility of a decrease in coverage for the 8-bit machines. Never fear, **ANALOG Computing** is "The #1 Magazine for Atari Computer Owners," and that means *all* Atari computers.

We'll increase pages if need be, in order to continue coverage of the 520ST *and* the older computers. The 130XE is still selling strong, as are other 8-bit Ataris, because of their unbeatable prices. We just find ourselves getting a little more excited over the newer ST.

Speaking of excitement, we now have 3½-inch disk subscriptions available for you ST owners, and our 5¼-inch disks are available from many of the dealers who carry our magazine. If your representative doesn't carry the disk versions, have him or her give us a call. Remember, you can also get a disk subscription directly from us.

Last, I'd like to mention our TCS. It's been a big success, but we've been trying to keep it a little low-key, because of the lack of available units which we could use simultaneously. At the present, we have four. We once hoped to upgrade that number to seven, but revised our plans, as you'll see further on.

We were, believe it or not, afraid to publicize the fact that we have over 800 download files available for the 8- and 16-bit Atari computers. With only the occasional ads we've run for the TCS, we've had around 2000 people sign on.

The single continuing complaint that we hear is about the long distance phone call, especially from those TCS users with 300 baud. Still, we've had people as far away as Australia call and use the TCS.

We thought of many different ways to curb costs, but none were feasible—without raising the TCS subscription fee. We've also been looking at some of the larger services, thinking of transferring the complete TCS over to one of these. Our demands were: speed, capacity, ease of use, low cost to the user, and no lag time between the time we upload to the service and the time the information was actually available to the user.

I know, you're saying, "There's no such animal." Surprise! Have you taken a look at Delphi lately? It fits our needs perfectly, and we're already in the process of getting the **ANALOG Computing** TCS switched.

Don't worry; if you're already a TCS subscriber, your time will be transferred over—and now your phone bills won't put you in the poorhouse.

Our resident SYSOP, Charles Bachand, will finally be able to sleep nights. On more than one occasion he's called the TCS after midnight and found the system down. That means a 15-mile drive for him to fix the problem. He deserves a lot of thanks for his long hours of work on the TCS.

Thanks must also go to Tom Hudson for the TCS setup and software design. Nice work, guys.

Michael J. Des Chenes
Editor/Publisher
ANALOG Computing

ARE YOU LOOKING FOR A DRIVE...

That is fast, quiet, reliable?
That has a built-in printer interface?
That holds 360 KBYTES of information?
That is compatible with Atari® 400, 800, 600XL, 800XL,1200XL and 130XE computers?

THEN, THIS IS THE "ONE".™

- Single or double density (software selectable).
- Single or double-sided (software selectable).
- Direct drive motor.
- Printer interface built-in.

FREE "TOPDOS."

130XE SUPPORT WITH:

- Standard 64K RAM disk.
- Expanded 78K RAM disk.
- Special 90K buffer for one pass disk duplication with only one drive.

WISE DENSITY

- Automatic selection between single density, double density, and double-sided double density.

AUTO RAM DISK INITIALIZATION FOR ATARI 130XE.

✳ASTRA SYSTEMS, INC.

2500 South Fairview/Unit L
Santa Ana, California 92704

Call **(714) 549-2141**

*Atari is a reg. trademark of Atari Corporation.

CIRCLE #103 ON READER SERVICE CARD

READER COMMENT

Bonanza—a hot strike?

I am an Atari owner—XL/XE computers. I find these products to be very well suited to my needs.

In issue 35 (October, 1985) of **ANALOG Computing**, I came across an Atari Corp. ad offering, among other things, **AT Speller/AtariWriter** for $9.99. This was under the overall title of "Atari Bonanza."

To make a long story short, I sent a bank money order for an even $10.00 drawn on October 10, 1985. Since this was made out and sent, I have heard nothing from Atari.

I find two months enough time for most "things."

Sincerely yours,
Kenneth T. Buck
Andover, MA

If you have any problems, or questions about Atari Bonanza products and service, contact Neil Harris at Atari Corp., (408) 745-2160. — *Ed.*

ST programs sought.

I am a computer programmer by trade. When I get home from a hard day's work, I don't want to work just as hard at home. I want to relax and unwind.

With your type-in programs, all the hard work (writing, testing, debugging) is already done. All I have to do is type them in and run them. And with my TCS account, I don't even have to type! Which brings me to my first point.

Lately, some really good programs have appeared in **ANALOG Computing**, written in Action! I do not own Action! —and I have no plans to purchase it in the future. But those programs look *sooo* inviting. . .The solution? On TCS, the Action! programs are also stored as .OBJ files. Just download the .OBJ file and run it as a binary program, no Action! cartridge necessary.

From time to time, you help Atari users locate specific software. I am looking for two programs (for the ST). The first is a floor plan designing program, the second is **Chopper Command** (by Activision). Anyone out there have a home-grown package of the former or know the whereabouts of the latter?

Thank you for your time reading this letter, and keep up the good work on your fine magazine.

James E. Caple
Columbus, OH

More on Graph E's.

Here is a modification to the **Graph E's** article of August, 1984 (issue 21). This will double the size of the image generated on the line printer.

```
80 DIM AD$(4),H5(20),A(3),
DUM$(3),DUM1$(3),A$(384),B
$(384)
3625 RESTORE 3710:FOR B=1
TO 113:READ N:POKE 1535+B,
N:NEXT B:DM=PEEK(88)+PEEK(
89)*256:DM=DM+40*191
3670 A$=CHR$(0):A$(384)=CH
R$(0):A$(2)=A$
3671 B$=A$
3680 W=USR(1536,X,ADR(A$),
ADR(B$))
3690 LPRINT CHR$(27);"K";C
HR$(128);CHR$(1);B$
3691 LPRINT CHR$(27);"K";C
HR$(128);CHR$(1);A$
3700 NEXT X
3710 DATA 104,104,133,203,
104,133,202,104,133,205,10
4,133,204,104,133,207,104,
133,206,160,193,162,0,173,
255,255
3715 DATA 136,240,83,141,2
55,255,169,8,133,201,161,2
02,10,38,199,38,200,6,199,
38,200,198,201,208,243,165
3720 DATA 199,129,204,165,
200,129,206,230,206,240,24
,230,206,240,25,230,204,24
0,31,230,204,240,32,165,20
2,56
3725 DATA 233,40,133,202,1
44,13,76,23,6,230,207,76,6
3,6,230,207,76,67,6,198,20
3,76,23,6,230,205
3726 DATA 76,71,6,230,205,
76,75,6,96
```

Our thanks to the anonymous donor of this modification. — *Ed.*

LOGO Demo correction.

Listing 2 of issue 36's ***LOGO Demos*** *was missing some copy—our error. For ease in correcting this, we're printing the initial lines (the procedure section) of the listing here:*

```
TO RCIRCLE
COIN
IF :C = 0 [MAKE "X RANDOM!
 80] [MAKE "X RANDOM -80]
COIN
IF :C = 0 [MAKE "Y RANDOM!
 150] [MAKE "Y RANDOM -15!
0]
MAKE "R RANDOM 150
MAKE "FC 5E :X :Y
CIRCLE 5E :FC :R
END
```

The shaded section is that copy which was left out. Our apologies for this inadvertent omission. — *Ed.*

Catch up, Electronic Arts.

I recently read (on CompuServ) a letter from the president of Electronic Arts, stating that they would not be producing software for the Atari ST.

Was it not Electronic Arts which had originally promised **Marble Madness** and other titles for the ST? Why the sudden "cold shoulder" toward the ST?

After careful consideration, I can only arrive at one possible explanation for Electronic Arts behavior: Electronic Arts

(continued on next page)

ATARI

SUPERPRINTER PACKAGES

SG-10 Printer & U-Print A ... 279
Panasonic 1091 & U-Print A.... 309
Powertype & U-Print A 369
Legend 808 & U-Print A 229
Super printer packages have no extra charges added when shipped in Continental USA.

While Supplies last
ATARI 1027 PRINTER 99

MONITORS

Commodore 1802 .. 169
Taxan.....Call
Sakata 100-C 169
Amdek .....Call
Samsung Green .. 79.95
Samsung Amber .. 79.95
Monitor Shipping $10.00

Indus GT .. Call

Atari 130 XE Computer ..... Call For Current Prices
Atari 1050 Disk Drive ..... Prices

PRINTER BUFFERS

Microfazer ... Call
U-Print-16K Printer Buffer 79.95
U-Print-32K Printer Buffer 89.95
U-Print-64K Printer Buffer 99.95

star micronics inc
THE POWER BEHIND THE PRINTED WORD.

SG-10.....215
SG-15 369
SD-10 339
SD-15 449
SR-10 ..... Call
SR-15 ..... Call

MODEMS

Volksmodem 1200 189
Atari XM-301 ... 44.95
US Robotics 2400 469
Password 1200...209
Novation ..... Call
Prometheus 1200 319
Reverter 39.95

PRINTERS

Citoh 7500AP ... 219
Epson ..... Call
Toshiba 1340 559
Legend 808 169
Panasonic 1091 ... 245
Panasonic 1090 .. 199
Powertype..... 309

Buy THE PRINT SHOP for 27.95 with the purchase of any printer.

PRINTER INTERFACES

Aid Interfast I .. 99.95
U-Print A 54.95
P.R. Conn. 69.95
U-Print A/16K Buffer .. 79.95
U-Print A/32K Buffer .. 89.95
U-Print A/64K Buffer .. 99.95

HUNDREDS OF ITEMS AVAILABLE FOR THE ATARI PLEASE CALL

Infocom ST

Deadline 34.95
Starcross 34.95
Zork I, II, or III.....29.95
Witness 27.95
Suspended 34.95
Planetfall 27.95
Sorcerer 29.95
Seastalker 27.95
Cutthroats 27.95
Hitchhiker 27.95
Suspect 29.95
Wishbringer 27.95
Infidel 29.95
Enchanter 27.95
Spellbreaker 34.95

OSS ST

Personal Pascal 64.95
Personal Disk Kit 27.95
Personal Prolog 79.95

Hippopatamus

Hippopixel 27.95
Hippo Almanac.....23.95
Hippo Ramdisk.....23.95
Hippospell 27.95
Hippobackgammon 27.95
Hippo Disk Utilities.....34.95
Hipposimple 34.95
Hippoconcept 64.95
Hippoword 27.95
Hippoart 27.95

Casio CZ-101 Electronic Keyboard w/Midi Interface 369

ATARI 520 ST SOFTWARE & HARDWARE*

Haba

Hippo C 54.95
Checkminder 54.95
Business Letters 34.95
Wills 34.95
Haba Writer 54.95
Haba Calc.....54.95
Haba Graph 54.95
Haba File 54.95
Haba Com 54.95

Atari 520 ST RGB System.. Call

Atari 520 ST Monochrome System ..... Call

***Please call for stock availability on Atari ST products before ordering by mail**

Miscellaneous ST

PC/Intercom 89.95
Hex.....27.95
Sundog.....27.95
Softspool 24.95
VIP Professional 129.95
Ultima II 39.95
Perry Mason.....34.95
Degas 27.95
Farenheit 451 .....34.95
Amazon 34.95
ST Talk 17.95
H&D Base 69.95
Typesetter ST 27.95
Habadex Phone 39.95
General Ledger .....Call
Word Invaders 24.95
Hacker 29.95
Spellbreaker 34.95
The Final Word 94.95
9 Princes Amber 34.95
Dragonwood 34.95
Homeword Math.....34.95
Homeword Writer 34.95
Gato 34.95
King's Quest II.....34.95
Treasure Island.....27.95
Wizard of Oz.....27.95
Transylvania 27.95
Borrowed Time 34.95
Mi-Term 54.95
Regent Word.....34.95
Regent Spell 34.95
Zoomracks 74.95

SUPER SPECIALS ★ SUPER SPECIALS ★ SUPER SPECIALS ★ SUPER SPECIALS ★ SUPER SPECIALS ★ SUPER SPECIALS

BUY ANY INFOCOM TITLE AND RECEIVE THE CORRESPONDING INVISICLUE ABSOLUTELY FREE*

SUPER SPECIAL PRICES EFFECTIVE NOW THRU MARCH 31, 1986

★ SUPER SPECIALS ★ SUPER SPECIALS ★ SUPER SPECIALS ★ SUPER SPECIALS ★ SUPER SPECIALS ★

A T A R I S O F T W A R E

ACTIVISION
Call for items and prices

BRODERBUND

Karateka-D 20.95
Champ Loderunner-D ...23.95
Print Shop-D 28.95
Bank Street Writer-D ...34.95
Print Shop Graphics I, II or III-D 19.95
Print Shop Paper 16.95
Print Shop Comp. 27.95

INFOCOM

Deadline-D 29.95
Enchanter-D 24.95
Infidel-D 29.95
Planetfall-D.....24.95
Sorcerer-D 29.95
Starcross-D 29.95
Starcross-D 29.95
Suspended-D 29.95
Witness-D 29.95
Sea Stalker-D 24.95
Cutthroats-D.....24.95
Suspect-D 29.95
Hitchhiker-D 24.95
Zork I-D 24.95
Zork II or III-D 27.95
Wishbringer-D 27.95
Spellbreaker-D 29.95

XLENT SOFTWARE

Megafont-D.....19.95
Page Designer-D 23.95
Typesetter-D 27.95
Megafiler-D.....23.95
Rubber Stamp 23.95

GAMESTAR

Star League Baseball-D/T.....20.95
Starbowl Football-D/T 20.95
On Track Racing-D 20.95

MICROPROSE

Silent Service-D 23.95
Gunship-D 23.95
Accrojet-D 23.95
F-15 Strike Eagle-D 23.95
Decision in Eagle-D 27.95
Kennedy Approach-D ...23.95
Crusade in Europe-D....27.95

OPTIMIZED SYSTEMS

Basic XE-Cart 52.95
MAC 65 XL-Cart 49.95
Action-Cart 49.95
Basic XL 39.95
All Tool Kits 20.95

ELECTRONIC ARTS

Archon 19.95
Archon II 24.95
Mule 19.95
Realm/Impossibility 19.95
Murder/Zinderneuf 19.95
Music Construction 19.95
Pinball Construction 19.95
One on One 24.95
Seven Cities of Gold 24.95
Financial Cookbook 29.95

EPYX

Rescue on Fractalus-D... 19.95
The Eidolon-D 24.95
Koronis Rift-D 24.95
Ballblazer-D 24.95
Summer Games-D.....24.95

DATSOFT

Goonies 23.95
Zorro 23.95
Alternate Reality 27.95

D—DISK T—CASSETTE CART—CARTRIDGE

BATTERIES INCLUDED

Home Pak-D 34.95
Paper Clip-D 39.95
B-Graph-D.....34.95

MISCELLANEOUS

TAC III Joystick 12.95
TAC II Joystick 12.95
Starfighter Joystick 9.95
Silk Stik Joystick 7.95
Wico 3-Way Joystick ...23.95

EST. 1982

ComputAbility™

P.O. Box 17882, Milwaukee, WI 53217
ORDER LINES OPEN
Mon-Fri 11 a.m. - 7 p.m. CST • Sat. 12 p.m. - 5 p.m. CST

To Order Call Toll Free

800-558-0003

For Technical Info, Order Inquiries, or for Wisc. Orders

414-351-2007

ORDERING INFORMATION: Please **specify system.** For fast delivery send cashier's check or money order. Personal and company checks allow 14 business days to clear. School P.O.'s welcome. **C.O.D. charges are $3.00.** In Continental U.S.A. include $3.00 for software orders, 4% shipping for hardware, minimum $4.00. Master Card and Visa orders please include card #, expiration date and signature. WI residents please include 5% sales tax. HI, AK, FPO, APO, Puerto Rico and Canadian orders, please add 5% shipping, **minimum $5.00.** All other **foreign orders add 15% shipping, minimum $10.00.** All orders shipped outside the Continental U.S.A. are shipped first class insured U.S. mail. If foreign shipping charges exceed the minimum amount, you will be charged the additional amount to get your package to you quickly and safely. All goods are new and include factory warranty. Due to our low prices all sales are final. **All defective returns must have a return authorization number.** Please call (414) 351-2007 to obtain an R.A.# or your return will not be accepted. Prices and availability subject to change without notice.

MISCELLANEOUS

Panzer Grendier-D 24.95
Hacker-D 19.95
Mindshadow-D 19.95
Countdown/Shutdown-D 19.95
Cross Ctry. Road Race-D 19.95
Master of Lamps-D.....19.95
Mastertype-D 27.95
Flight Simulator-D 34.95
Sam-D 37.95
Castle Wolfenstein-D ...20.95
Compuserve Starter Kit 21.95
Home Accountant-D 49.95
Monkey Wrench-Cart ...23.95
Ultima III-D 37.95
Sargon III-D 34.95
Spy vs. Spy-D.....23.95
Odesta Chess-D 49.95
MMG Basic Compiler-D 69.95
Net Worth-D 49.95
Ramrod XL 69.95
Universe-D 69.95
Beachead-D 21.95
Letter Perfect-D 39.95
Data Perfect-D 39.95
Fleet System II-D 49.95
Strip Poker-D 23.95
Halley Project-D 29.95
Micro League Baseball-D 29.95
Harcourt/Brace S.A.T.-D 49.95
Ultima I-D 23.95
Ultima II-D 37.95
F-15 Strike Eagle-D.....23.95
Ultima IV-D.....41.95
Bounty Bob-Cart 29.95
Alien Voice Box 99.95
Spy Hunter-D 29.95
Tapper-D.....29.95
Ken Uston's Blackjack-D 49.95
Omnimon 69.95
Island Caper-D 23.95
General Manager/MLB 29.95

SSI

Carrier Force-D.....37.95
Combat Leader-D 24.95
Cosmic Balance II-D 24.95
Cosmic Balance-D 24.95
Broadsides-D 24.95
War in Russia-D 49.95
50 Mission Crush-D 24.95
Questron-D 32.95
Rails West-D.....24.95
Computer Ambush-D ...37.95
Galactic Adventures-D 37.95
Computer Baseball-D ...24.95
Reforger 88-D 37.95
Objective Kursk-D 24.95
Breakthru/Ardennes-D 37.95
Field of Fire-D.....24.95
Imperium Galatium-D ...24.95
Oper. Market Garden-D 32.95
Kampfgruppe-D 37.95
Computer Quarterback-D 24.95
Colonial Conquest-D 24.95
Gemstone Warrior-D 21.95
Six Gun Shootout-D 24.95
Battle of Antietnam 32.95
USAAF 37.95

SYNAPSE

Synfile 32.95
Syncalc 32.95
Syntrend 25.95
Synchron 25.95
Syncom 25.95
Synstock 25.95
Loderunner's Rescue... 20.95
Syncalc Templates 16.95
Essex 27.95
Brimstone.....27.95
Mindwheel 27.95

ATARI is a trademark of ATARI, INC.

No surcharge for MasterCard MasterCard **or Visa** VISA

CIRCLE #104 ON READER SERVICE CARD

must have been given a very good reason from Commodore itself to terminate production of their ST software, in a vain attempt to "kill" the ST (perhaps thinking that other companies would follow suit).

Electronic Arts should realize that there is a large market for ST software. Sure, write it for the Amiga, but at least port it over to the ST. There are plenty of programmers who could do the porting and do it well, not just cheap "translations" (as is done with some 8-bit software).

While Commodore has been busy advertising the Amiga, Atari has been busy selling STs. If Electronic Arts does not realize that they have made a mistake now, they will be playing "catch-up" with the other software companies that were intelligent enough to produce ST versions of their software, especially when the Amiga is not the big success they had hoped it would be.

Sincerely,
Blake Arnold
Dover, DE

Let's hear some *news*.

The home computer industry is once again in the doldrums, if we can believe articles in the trade magazines and daily newspapers. The reason is familiar: the manufacturers have already sold their product to the curious and to the high-tech junkies out there. But the other 95 percent of the market can find no practical use in the real world to justify spending $1000-$3000 on a sophisticated toy.

Renewed sales growth in home computers—and motivating those who already own them to continue using them—is totally dependent on finding day-to-day applications that make life easier for those who could care less about a slicker way to write a letter, or playing mindless games.

This truism has been said over and over, and still nothing of substance is done about it. Manufacturers continue to be hung up on internal technology, which results in more sophistication in a cheaper, smaller package (read: 520ST, Amiga), but no real change in practical application. The new machines just do what the old ones did, only faster. That's obviously not the kind of application that's going to appeal to the masses.

It seems to me that the focus of technological attention ought to be on external electro-mechanical interfacing (the 520ST's MIDI interface for musical instruments is a small step in that general direction).

Unless or until home computers can be programmed to turn things on and off in the house, in a way that is less of a bother than a light switch, the home computer industry will have to accept their current sales figures and live with them. Less bother than a light switch? Now that's a real challenge.

What's frustrating about all this is that some of the external electro-mechanical technology is already available. There's no need for computer makers to reinvent the wheel . . . just make what they have work with what *we* have.

One example is the line of AC line carrier transmitters and receivers by BSR (available at most Radio Shack stores). Without getting into the technical aspects of how it works, line carrier uses the existing AC voltage/wiring in a house to transmit and receive signals on various selectable channels. Thus, one can plug in a transmitter in a bedroom, and by pressing a button, turn on lights plugged into any number of receivers anywhere (or only in certain rooms) in the house. The transmitter can be programmed to do this automatically every day—lights on and off. Pretty basic stuff in today's world, but how many computers are equipped to do it? Not many, if any, yet this is precisely the kind of thing home computers must be capable of doing.

Another potential application suggested itself over the past holiday season. Outdoor (and indoor) Christmas light displays haven't changed much in thirty years. Miniature lights that randomly wink and flash are the best we can do. Pretty boring stuff.

But suppose my Atari 800XL had an interface (and a program) that would enable me to create all sorts of interesting, sequencing light effects.

It should be a piece of cake, with the right line carrier transmitters/receivers and seven or eight individually addressable strands of lights. And the computer should be able to handle both outside and inside displays, turning each on and off at various times of the day and night. Pretty basic stuff, but my computer can't do it, and neither can the 520ST or the Amiga.

When are computer manufacturers going to wake up and start focusing on the real world? When are computer magazine publishers going to begin helping the process by devoting space on a regular basis to developing practical uses for the home computer?

If you asked them, I'm sure your readers can develop a long list, along with programs. Then all we'll need is the interface hardware. (Sam Tramiel, are you listening?)

I apologize for the length of this diatribe, but I feel this is something that needs to be said—often. I yield in hopes of further discussion on this vital topic.

Sincerely,
Terry D. Eisenberg
York, PA

All right, readers. This is a challenge if we've ever heard one. Our June issue will spotlight home uses for the Atari and we'll be keeping this letter in mind. How about it, programmers? — Ed.

Turning off alpha keys.

How often have you made simple typing mistakes during numeric entry, like typing the letter *O* when you meant to press *0*, or hitting *I* or *L* when you wanted *1*? Did you know that the Atari Operating System (OS) can trap those errors? My technique fools the OS into rejecting the alphabet keys.

The secret in turning off the alphabet lies in how the Keyboard Handler Routine is structured. Since keyboard entry is permitted only in the lowercase, uppercase and control modes, it's illegal to press both the SHIFT and CTRL keys together, as if that combination were a fourth mode.

The first thing the OS does is check for that illegal CTRL/SHIFT combination. But, by putting the equivalent value of the CTRL/SHIFT key combination into the ShiftLock memory location (702 or $2BE), all the alphabet keys—and only the alphabet keys—are disabled.

Why? The Keyboard Handler Routine processes alphabet keys twice, while the other keys are processed only once. In the first processing loop, any key pressed along with the CTRL/SHIFT combination is rejected, because a keyboard code of 192 ($C0) or higher is impossible. The ATASCII character set reference table contains only 191 values.

What happens if the ShiftLock value is changed (POKE 702,192) to simulate that illegally high CTRL/SHIFT value? Since the ShiftLock is checked only in the second processing loop used for al-

(Reader Comment *continued on page 102)*

HIPPOSPELL
HIPPOPOTAMUS SOFTWARE, INC.
985 University Avenue, Suite 12
Los Gatos, CA 95030
520ST $39.95

by Clayton Walnum

I hate to start off a review sounding like a bad joke, but I've got good news and bad news. The good news is that **HippoSpell** works, and works fast. All of you who are absolutely desperate for a spelling checker and have excess cash you just have to get rid of, feel free to run right out and pick up a copy. For the rest of you, here's the bad news.

Yes, **HippoSpell** works. Unfortunately, it's so full of minor annoyances and outright bugs that there's no way I can recommend this product. But make up your own mind.

When you load **HippoSpell**, you're presented with a blank screen. At its top are three drop-down menu headings: *Desk*, *File*, and *Help*. The desk menu contains the usual "about Hippo," as well as any desk accessories that may be in memory.

The help menu allows you to view, on-screen, the brief program documentation. I might mention that this on-disk documentation is all you get in the way of instructions. Hippo doesn't seem to believe in manuals. Worse though, the instructions are so skimpy that they leave discovery of some functions to what I refer to as "the Biblical method" (you know—seek and ye shall find).

All of **HippoSpell**'s main functions are accessed from the file menu. Choices include: *Check File*, *Correct File*, *Show File*, *Save New Words*, *Use New Words*, *Save Corrections*, *Load Corrections*, *Word Statistics*, and *Quit*

To check a document, you select the check file option. A file selection dialog box will appear, and you make your choice with the mouse. Once the file is selected, you'll get your first shock. The dialog box goes away and the screen remains blank.

That's right, you can't view the document as it's being processed. This is certainly one of the reasons for **HippoSpell**'s speed (a 10K file with no errors will check in about 15 seconds), but, gee, I want to see what the program is doing.

When **HippoSpell** encounters a word it doesn't recognize, a dialog box pops up with the message: *Is this a correctly spelled word?* Below the message is the word in question, as well as YES and NO buttons. Remember, the document isn't on-screen. That means you can't see the context in which the word's being used. More often than not, it doesn't really matter, but sometimes...

If you click the YES button, the word is added to a list of new words. If you click the NO button, you're asked to type in the correct spelling. The process then continues.

Don't be surprised if, when you click the YES button, the same word appears again, then again! Ditto for other flagged words. It seems that, even though the program's creating a file of new words, it doesn't bother to check when it comes across something not in the dictionary. If the word *bamboozler* appears twenty times in your document, you're going to have to tell **HippoSpell** twenty times that it's spelled correctly.

There is a way around this. After checking the document, you may use the save new words option to add the words to a disk file. As long as the use new words option is activated while you are checking a file, these will be included with the regular dictionary. But take care—once you save new words to the file, that's it. You own them. I could find no way to remove mistakes.

I was amazed at some of the words **HippoSpell** did not recognize. I never considered the words *color* and *program* to be at all unusual. Hard to believe these words are missing, considering the folks at Hippo claim a 30,000-word dictionary.

Getting back to dialog boxes, whenever one pops up the YES or NO buttons are always there. It doesn't matter whether they make sense or not. What do you make of a message like *No Errors Detected*, that requests a yes or no answer? What's that supposed to mean? Well, I'll tell you. It means that the author of this program was too lazy to design the appropriate dialog box.

When **HippoSpell** is finished examining a file, it will ask if you wish to make corrections. If you reply YES, the program will scan through the document, replacing all occurrences of words in its corrections list with the new spelling you entered for the word. If you should select NO, you may still correct the file by using the correct file option in the file menu.

Another option you have after checking a file is that of saving frequently misspelled words to disk. The next time you

check some text, you may load the corrections, and any misspellings of the words in the list will be corrected automatically. This can be pretty handy.

Finally, if you're one of those people who love to have everything laid out in a statistical manner, there's the word statistics option. You can get a chart that tells you the number of words starting with each letter of the alphabet, as well as their lengths. A count of words based on their lengths is also available. And, if that sounds like fun, how about a complete list of every word in the document, including the number of times it was used?

I suppose that some of the above information could be useful, especially to a person trying to write at a specific grade level. But the alphabetic listing seems a complete waste to me. A more useful feature might have been a calculation of average sentence length.

As I mentioned before, there are a lot of things you're going to have to discover for yourself. One is how to abort the program once it starts checking a file. I discovered this one by randomly pressing keys until something happened. As it turns out, pressing the ESC key does the trick. This will give you a dialog box with the message *Really Abort?* But watch out—it's unpredictable.

Clicking the YES button aborts the function in progress, but if you click NO (actually, you have to click it twice; the first time it won't respond), the program will ask you to correct the last word flagged. Huh? Another time I attempted an abort, I found myself back at the desktop. I guess I annoyed the program. It walked out on me.

Since copy protection, especially of utility and application software, seems to be becoming a sore spot among users, I should add that the **HippoSpell** disk is copy protected. You *can* make a backup, but the Hippo disk must be in drive A when you first start up the program. After you've selected one function, you may replace the original with your backup. If something happens to the original, you're *still* out of luck.

I think Hippo Software should get on the ball and spend a lot more time developing their software, rather than rushing it out the door as fast as they can. Their customer support is also lacking.

When I first started to evaluate this program, I couldn't get it to function at all. I called Hippo twice, and both times I was told that someone would return my call. They never did. As it turned out, the software wasn't compatible with the version of TOS we've been using at **ANALOG Computing** (for the hard disk).

I've been hearing nothing but complaints about Hippo's software. In last month's review of **Hippo Ramdisk**, I tried—considering that it was the first of their products I'd looked at—to give them the benefit of the doubt.

I'm no longer willing to do that. Poor products will do nothing but give the ST a bad name. Hippopotamus Software managed to get a lot of software out at a time when ST owners had a real need. That doesn't excuse them for marketing inferior products, but it's certainly a consideration. They should now take a step back, re-evaluate their position in the industry and decide whether or not they plan to stay around.

Adventurous Programming

by Clayton Walnum

Bernard Leddenberry opened the package with trembling hands. How long had he waited for this? A year? Two years? He could hardly remember the first time the rumors of its development had reached him. There had been a lot of unsubstantiated stories at first. The words uncanny and extraordinary were frequently heard, whispered from informed sources into the ears of those privy to the "inside scoop." Over the last few months, the subject had been treated with a reverence that was downright eerie.

But that was past; the long wait was finally over. Here it was—Intercom's new adventure game ***Brainchild****. Taking a deep breath, he closed his eyes and booted it up.*

YOU ARE STANDING IN FRONT OF YOUR HOUSE. A SIDEWALK LEADS NORTH AND SOUTH. A DOOR IS TO THE WEST.

Hmmm. Doesn't look any different. In fact, it's a bit disappointing. Well, let's see what happens.

>GO IN THE DOOR
YOU ENTER YOUR HOUSE. THERE IS A COMPUTER HERE AND A TELEPHONE.
>LOOK AT THE COMPUTER
YOU SEE YOURSELF PLAYING BRAINCHILD
>TALK TO MYSELF
WHAT ARE YOU? SOME SORT OF LOONY?
>WHAT ARE YOU? A WISEGUY?
WHAT'S IT TO YOU?

Bernard sat back, stared at the screen. Boy, this game sure had a large vocabulary—he couldn't deny

that. But he really didn't like the idea of a rude computer. Nothing worse than a machine that doesn't know its place. He had half a mind to. . .

Wait a minute. What was he thinking? It's only a game, after all!

>LOOK AT THE PHONE
IT JUST STARTED RINGING.
>ANSWER THE PHONE
OKAY. IT'S YOUR MOTHER.

The phone in the kitchen rang, so loudly that Bernard almost fell out of his chair. Who had turned the darn thing up? What he should've done was take it off the hook. Was it too much to ask for a little peace, so he could get into the game? He crossed the room, cursing with each step.

A few minutes later he was back at the keyboard. Why had his mother decided to call him just then? It was just a coincidence, right? There was no way the computer could've known.

>OKAY, I'M BACK
THAT'S NICE. WHAT DID YOUR MOTHER WANT?

Bernard felt something cold slither up his spine. This was getting too weird. He certainly didn't expect to get a response to that last command, except maybe the standard I don't understand that.

Okay, okay. Just cool it. The program is obviously referring to the mother in the game, not yours. Let's not get carried away. He typed in his answer, confident that there was no way the program was going to decipher it.

>HOW DO I KNOW WHAT SHE WANTED. YOU DIDN'T TELL ME YET.
HOW COULD I TELL YOU? YOU'RE THE ONE WHO ANSWERED THE PHONE.

Good Grief! It did *understand! Was there no limit to this game's vocabulary?*

>ARE YOU GOING TO LET ME PLAY OR WHAT?
YOU KNOW, BUB, I DON'T CARE TOO MUCH FOR YOUR ATTITUDE. YOU BETTER STRAIGHTEN OUT OR I'LL SEE TO IT THAT THIS IS YOUR LAST ADVENTURE GAME.
>OKAY, COMPUTER, LISTEN UP.
YES?
>YOUR MOTHER WEARS ARMY CPU'S!
WHY YOU. . .! THERE'S A KNOCK AT YOUR DOOR.
>WHO IS IT?
IT'S ME, YOU BUM. NOBODY TALKS ABOUT MY MOM! YOU'RE DEAD. THIS ADVENTURE IS OVER!

Bernard froze. He listened in mounting panic for a few moments, then snatched the disk from his drive. No good. The sound continued.

The monitor! Turn off the monitor! He banged at the ON/OFF switch. There was a snap as the switch broke, and the picture went black.

The sound continued.

One last chance. He booted up DOS, crammed the game disk back into the drive, and jabbed a couple of keys. The chunk-chunk sound of the format operation filled the room. Please let this work, he thought. Please! The drive's busy light blinked off. The disk was erased, but it didn't help. The knocking at his door continued.

Back again.

Good grief! Somebody help that poor fool. No, don't answer the door. . .

Huh? Oh, hi. Didn't see you there. Welcome back to **Adventurous Programming**. Sorry about all that carrying on; things were getting a bit spooky.

In case it isn't already obvious from the above nastiness, this month's installment deals with adventure game intelligence. Of course, there's never been a game quite like **Brainchild** (and, if we're lucky, there never will be), but some pretty amazing things have been done. We're going to take a look at what makes these games so smart, then get to work on our own adventure.

Program intelligence.

One of the most impressive aspects of an adventure game is the program's ostensible intelligence. First-time players may find interaction with the game downright uncanny. This is especially true if the program has been designed carefully, incorporating a large vocabulary and allowing the player plenty of alternatives.

Do our computers actually understand English? No, not really. This interaction between player and computer is a carefully crafted illusion. And guess who's responsible for creating and sustaining that illusion? You are, my friend! Put on your thinking cap—here comes Lesson 2.

Lesson 2—the parser.

Every text adventure game has this precept: it must accept commands from the player, then translate these commands into a form the computer can interpret. The parser is the portion of the program that tackles this.

Parsers come in many forms, from those that handle only simple verb/noun commands to the sophisticated, full-sentence parsers that you've seen in games like **Zork**. Obviously, the more text the parser can handle in a single input, the more complicated the programming involved. For that reason, we're going to keep things simple and stick to two-word commands.

With a verb/noun parser (at least, with ours), the player's command is first read into a string by an "in-

put" statement. The string is then divided into its verb and noun portions, each of which is compared with a table of valid input. If a match for both verb and noun is found, the verb's location within the table is used to determine where program execution should continue. If no match is found for either the verb or the noun, a message of some type is given to the player, advising him that the program can't translate his input. See? Nothing to it.

Listing 1 is the parser we'll be using in our adventure. Type it in, then check it with **Unicheck**. When you have a good copy, SAVE it to disk or tape, then RUN it.

Now get out the command list from last month and try typing some of them in. Each time you enter a valid command, the program will use the verb's value to determine what to do next. In our example, the program will print the verb you used. Not too impressive, but our game is far from complete.

What happens if you type something that isn't on the command list? As long as you use a valid verb and noun, the program will have no trouble translating your input. Any combination of the words included in the command list will work. Try some weird ones like *MOVE WINDOW* or *OPEN KEY*. The parser doesn't care if these commands make sense or not. All it knows is that it checked the tables and —yep—those are valid verbs and nouns. It's up to us, the programmers, to decide if the command makes sense, or if we wish to allow the player a particular action.

Now try typing in a command with an invalid verb or noun (something not found on the command list). The program will respond by telling you that it doesn't understand what you've typed. What a clever machine!

Taking it apart.

Now that you've learned the theory behind a parser, let's take a close look at the program. Rather than go through it from top to bottom, we'll trace program statements in the order of execution. Line 1 should need no explanation. If you don't know what's happening here, then you're reading the wrong article.

Lines 5780 and 5785 initialize some variables. The variables *N1* through *N20* will be used throughout the program in the place of constants, saving great chunks of memory—6 bytes apiece, to be exact. That adds up fast. You'll find, when you start writing your own games, that you'll need all the memory you can get. Adventure games tend to be quite large, since they must store a lot of text.

The variables *NN*, *NV*, and *SZ* are part of a little trick that will save you a lot of time. *NN* is the number of nouns our program will recognize, *NV* is the number of verbs, and *SZ* is the maximum length of an item description. As an adventure game develops, new ideas will occur to you. This will necessitate the addition of words to the game's vocabulary.

By using variables in the program wherever you refer to the verb or noun count, the installation of new vocabulary becomes a simple process. In fact, all you have to do is add the new word, then change the appropriate word count in Line 5785. Substituting *SZ* wherever we refer to the size of each item description allows us to easily change the item data. You'll see how all this works when we get deeper into the program example.

Why did we jump to the end of the program to start initialization? One reason is that, since we're only going to execute these statements once in the game, we might as well get them out of the way. More importantly, getting as much of the "nongame" code as possible moved to the end will make our game run much faster.

Lines 5790 and 5800 dimension the strings and arrays we'll need. Notice how we're using variables instead of constants. Also take special note of the variables *NN* and *NV*. They're used five times in these two lines alone. If you should decide to add to the game's vocabulary, you don't have to touch any of them. Just change their value in Line 5785.

Here's a description of each dimensioned item:

VB$	Table of valid verbs
I$	Item descriptions
A$	Text to be displayed
IN$	Player's most recent command
V$	Verb portion of the command
N$	Noun portion of the command
CC$	Machine language subroutine
D$	Machine language subroutine
Z$	Utility variable
NN$	Table of valid nouns
I()	Each item's current location
V()	Jump table for verbs
L$	Machine language subroutine
INV()	Player's inventory

We won't be using some of these items (like the machine language subroutine in D$) right away, but rest assured that we'll cover all of them in detail, sooner or later.

Line 5810 fills the string *I$* with blanks.

Line 5820 initializes some game variables. The variables are used as follows: *R* = player's current location; *UL* = flag to update screen; and *OP* = flag

indicating if the book is open. Once again, not all these variables are used in this month's program segment.

Line 6000 creates the table of valid nouns.

Line 6010 creates the table of valid verbs. Notice that, in both the noun and verb table, we use only the first three letters of each word. Now you know why, with some games, you can use a command like *JUM CLI*, when what you really mean is *JUMP CLIFF*. We can use as many letters as we wish, but the more we use, the bigger the tables become.

Line 6700 sends us up to Line 900.

Line 900 loads the verb translation table. A table like this allows us to have several verbs with the same meaning, for instance, *LOOK* and *EXAMINE*. This makes the game less frustrating for the player, since he isn't always stuck with trying to figure out an exact word. (Alternatives, remember?)

We're loading the table from the data in Line 31000. Look at this data and compare it to the verbs in VB$. There's one value for each verb. Both *LOOK* and *EXAMINE* are given the same value, because, as far as our game is concerned, they do exactly the same thing. If we wanted, we could add the verb *INSPECT* and give it a value of 1. Then we'd have three *LOOK* verbs.

You may be wondering why we've jumped back to the top of the program to read in this data. The reason is that FOR...NEXT loops run much more quickly when they're placed here. The more line numbers that come before a loop, the slower it runs. We don't want our player to wait any longer than necessary for initialization.

Line 950 reads the item descriptions, as well as their initial locations, into a pseudo string array. Since Atari BASIC doesn't allow string arrays, we have to simulate one. What we've done here is put each element of our "array" into one long string. Each element is eighteen characters long (the value of our variable SZ). Item number one occupies positions 1-18 within the string, item two is found in positions 19-36, and so on. To find a particular item's description, all we have to know is its number, then use a formula like *ITEM$ = I$(I*SZ-(SZ-1),I*SZ)*, where *ITEM$* is the string that will hold the item's description, *I$* is the pseudoarray, *I* is the item number, and *SZ* is the number of characters in each element.

Not all of our items are eighteen characters long, you say. True. That's why we filled the string *I$* with blanks before reading in the data. Each element will contain the name of the item, plus any blanks necessary to fill the field. *¿Comprende?*

Now look at the numbers that follow each item in the data at Lines 32120 and 32130. These are the beginning locations for each item. A positive number means that the item may be picked up by the player. A negative number means that the item is a permanent fixture of the room and cannot be taken. A *0* means that the item isn't to be placed in any location at the start of the game. These are usually items that must be discovered by the player, like the key we've hidden under the mat.

The only exceptions to the above rules are values -1 to -4. These are reserved to indicate items within the player's inventory. An item with a value of -1 is an object the player is carrying. A value of -2 means the player is wearing the item. The values -3 and -4 can be used in any way you wish. You might want -3 to indicate that the item is in the player's pocket or in a backpack.

Now you know why we started numbering our rooms with 5. If we began with room 1, the item value of -1 would be a bit ambiguous. Does it mean that the item is a permanent fixture of room 1, or is it in the player's inventory?

Getting back to the data, the welcome mat is a permanent fixture of room 8. The key has no starting location. We'll place it in the game when the player finds it. The coat starts off in room 9, and can be picked up by the player. The batteries, like the key, must be found—and so on.

Line 1260 is the beginning of our parser. We print the prompt *WHAT NOW?* and beep the computer to get the player's attention.

Line 1265 sets up an input "trap," in case the player presses RETURN without entering a command. The TRAP 1260 statement will catch the error and return control to the parser, giving the player another chance. After setting up the trap, we input the player's command. We also check to see if the player entered the single-word command *LOOK*. This command will reprint the room's description and is the only command in our game that doesn't require a noun. For this reason, it needs special handling. If the player enters it, the program skips the rest of the parser and jumps to the portion of code (which isn't there yet) that redraws the screen. If we didn't jump over the rest of the parser, the program would look for the noun portion of the command and, not finding it, give the player a warning.

Line 1270 checks if a single-letter command, like a direction, was entered. If it was, the command is stored in V$ (verb), and the program jumps to the appropriate section of code.

Lines 1480 and 1490 locate the space character between verb and noun segments of the command.

Line 1495 prints a warning to the player and returns to the parser if the program couldn't find the space character. This means that the player only entered one word (or more words, without a space).

Line 1500 separates the command into its verb and noun segments, using the location of the space as the dividing point.

Lines 1520 and 1540 check for the special verbs *SAVE* and *LOAD*. If they were entered, the program jumps to the disk/cassette section of the program.

Line 1580 checks for a two-letter verb and increases it to three characters, by adding a space to the end. The verb *GO* is an example of this.

Line 1590 checks for the command *YELL*. Commands starting with the verbs *YELL*, *SAVE* and *LOAD* require special handling, because the noun portions that follow are not items found in the game —and they won't be found in the noun table. If we allowed these verbs to pass by, the program would check the tables for verb and noun matches. Not finding a match for the noun, a warning would be printed at Line 1720 and the player would be prompted for a new command. This isn't what we want.

The commands *SAVE GAME* and *LOAD GAME* could've been changed to single-word commands and handled in the same manner as *LOOK*. But what about *YELL*? We can't ignore the second half of this command, because it's what the player wants to yell. We want to allow him to scream his head off if he wishes, and anything he yells will almost certainly be missing from our table of valid nouns. So we must redirect the program before the parser starts checking the noun table.

Lines 1620 and 1630 compare the noun that was entered with the valid nouns stored in the table. If a match is found, the noun's number is stored in the variable *Y*.

Lines 1640 and 1650 do the same for the verb input, and store the verb number (if found) in the variable *Z*.

Lines 1720 and 1730 print a warning message and jump back to the parser if the verb or noun was invalid.

Line 1760 translates the verb number to the verb type. For instance, *LOOK* is verb 1 and *EXAMINE* is verb 2. The statement on Line 1760 will give them both a value of 1, since they're really the same verb. This may be a little confusing, but if you study the program it should become clear.

Line 1780 concludes our parser. The verb value is used to jump to the section of the program that handles the verb.

Lines 1820 through 2900 are where all our adventure's action will take place. Essentially, this area of the program will be a large block of IF. . .THEN statements. Because of the large number of commands we must process, if we didn't divide this section in some way, the program would run too slowly. In our case, we've set aside a special area for each verb. This way, we need only examine those statements that perform the actions of the verb input.

Faster!

You've probably noticed that, when you type in a command, it takes the program a few seconds to validate your input and jump to the appropriate section. And this is only a small adventure. When you write a full-length game, the tables will be much longer, and the program will take a proportionately longer time to execute your commands.

Now that you understand what's going on with the parser, it's time for some of those machine language subroutines I promised you last month.

Type in Listing 2 and check it with **Unicheck** (see

page 10). When you're sure there are no typing errors, list it out to disk or cassette. Now LOAD the original program (Listing 1) and ENTER the file you created with Listing 2. SAVE the new version of the program, and RUN it.

Type in a few commands. Wow! Much faster, huh?

The machine language routine stored in L$ takes over the task of finding the space between the verb and noun. The routine stored in CC$ assumes the responsibility of comparing your input with the verb and noun tables. They both do the same thing we did with the BASIC FOR . . . NEXT loops, only much faster.

End of lesson.

That's it. Now you should have a good understanding of how a parser works. Next month we'll finish our adventure by taking a look at how the computer keeps track of items and locations. We'll also study how the program implements the commands we input and performs the necessary actions. In the meantime, experiment with the parser (but save a copy of the original—you'll need it next month) by adding new vocabulary and getting the parser to recognize it. Don't forget to change the value of the variables *NN* and *NV* when you add new words.

See you next month in the special adventure issue.

Listing 1.
BASIC listing.

```
1 GOTO 5780
10 RETURN
900 RESTORE 31000:FOR X=N1 TO NV:READ
A:V(X)=A:NEXT X
950 FOR X=N1 TO NN:READ A$,A:Q=5Z-LEN(
A$):I$(X*5Z-5Z+N1,X*5Z-Q)=A$:I(X)=A:NE
XT X
960 REM
1260 ? :? "WHAT NOW";:SOUND N0,N20,N10
,N8:FOR X=N1 TO N10:NEXT X:SOUND N0,N0
,N0,N0
1265 TRAP 1260:INPUT IN$:IF IN$="LOOK"
 THEN 960
1270 IF LEN(IN$)=N1 THEN V$=IN$:GOTO 1
820
1480 A=N0:FOR X=N1 TO LEN(IN$):IF IN$(
X,X)=" " THEN A=X
1490 NEXT X
1495 IF A=N0 THEN ? :? "What?":GOTO 12
60
1500 V$=IN$(N1,A-N1):N$=IN$(A+N1)
1520 IF V$="SAVE" THEN 7060
1540 IF V$="LOAD" THEN 6840
1580 IF LEN(V$)=N2 THEN V$(N3)=" "
1590 IF IN$(N1,N3)="YEL" THEN 2900
1620 Y=N0:FOR X=N1 TO NN:IF NN$(X*N3-N
2,X*N3)=N$(N1,N3) THEN Y=X
1630 NEXT X
1640 Z=N0:FOR X=N1 TO NV:IF VB$(X*N3-N
2,X*N3)=V$(N1,N3) THEN Z=X
1650 NEXT X
1720 IF  NOT Y THEN ? :? "Don't unders
tand that noun.":GOTO 1260
1730 IF  NOT Z THEN ? :? "Don't unders
tand that verb.":GOTO 1260
1760 Z=V(Z)
1780 ON Z GOTO 2000,2100,2200,2300,240
0,2500,2600,2700,2800,2900
1820 ? "ONE LETTER COMMAND":GOTO 960
1999 REM ********** LOOK **********
2000 ? "VERB = LOOK":GOTO 960
2099 REM ********** DROP **********
2100 ? "VERB = DROP":GOTO 960
2199 REM ********** GET **********
2200 ? "VERB = GET":GOTO 960
2299 REM ********** MOVE **********
2300 ? "VERB = MOVE":GOTO 960
2399 REM ********** GO **********
2400 ? "VERB = GO":GOTO 960
2499 REM ********** UNLOCK **********
2500 ? "VERB = UNLOCK":GOTO 960
2599 REM ********** INSERT **********
2600 ? "VERB = INSERT":GOTO 960
2699 REM ********** OPEN **********
2700 ? "VERB = OPEN":GOTO 960
2799 REM ********** READ **********
2800 ? "VERB = READ":GOTO 960
2899 REM ********** YELL **********
2900 ? "VERB = YELL":GOTO 960
5780 N1=1:N2=2:N3=3:N4=4:N5=5:N6=6:N7=
7:N8=8:N9=9:N10=10
5785 N11=11:N12=12:N13=13:N14=14:N15=1
5:N16=16:N17=17:N18=18:N19=19:N20=20:N
N=N10:NV=N11:5Z=N18
5790 DIM VB$(NV*N3),I$(NN*5Z),A$(160),
IN$(26),V$(N10),N$(N15),CC$(77),D$(37)
,Z$(N1)
5800 DIM NN$(NN*N3),I(NN),V(NV),L$(35)
,INV(N6)
5810 I$(N1)=" ":I$(NN*5Z)=" ":I$(N2)=I
$
5820 R=N5:UL=N0:OP=N0
6000 NN$="MATKEYCOABATFLAMAGWINDOOLIGO
PE"
6010 VB$="LOOEXADROGETMOVGO UNLIN5OPER
EAYEL"
6700 GOTO 900
30999 REM ********** V(X) DATA *******
31000 DATA 1,1,2,3,4,5,6,7,8,9,10
32120 DATA WELCOME MAT,-8,KEY,0,COAT,9
,BATTERIES,0,FLASHLIGHT,11,MAGAZINE,10
,WINDOW,-12,FRONT DOOR,-8
32130 DATA LIGHTED FLASHLIGHT,0,OPEN W
INDOW,0
```

CHECKSUM DATA.

(see page 10)

```
1 DATA 443,752,130,122,108,337,904,7,8
72,550,365,897,278,294,280,6339
1590 DATA 274,324,550,374,552,819,782,
477,955,778,879,487,848,493,526,9118
2200 DATA 619,859,493,624,432,969,109,
953,129,770,513,732,465,783,509,8959
5780 DATA 757,286,34,744,375,5,680,163
,917,236,592,43,106,4938
```

Listing 2.
BASIC listing.

```
920 FOR X=N1 TO 72:READ A:CC$(X)=CHR$(
A):NEXT X
930 FOR X=N1 TO 35:READ A:L$(X)=CHR$(A
):NEXT X
1480 A=USR(ADR(L$),LEN(IN$),ADR(IN$)):
IF A=N0 THEN ? :? "What?":GOTO 1260
1490 REM
1500 V$=IN$(N1,A):N$=IN$(A+N2)
1620 Y=USR(ADR(CC$),ADR(N$),ADR(NN$),L
EN(NN$))
1630 REM
1640 Z=USR(ADR(CC$),ADR(V$),ADR(VB$),L
EN(VB$))
1650 REM
31999 REM **********CC$ DATA*********
32000 DATA 104,104,133,204,104,133,203
,104,133,206,104,133,205,104,104,133,2
07,162,0,142,255,6,134,213,232
32010 DATA 160,0,177,203,209,205,208,8
,200,192,3,208,245,134,212,96,173,255,
6,24,105,3,197,207,240
32020 DATA 16,141,255,6,165,205,24,105
,3,133,205,144,217,230,206,176,213,169
,0,133,212,96
32049 REM
32090 DATA 104,104,104,141,255,6,104,1
33,204,104,133,203,160,0,177,203,201,3
2,240,8,200,204,255,6,208
32100 DATA 244,160,0,132,212,169,0,133
,213,96
```

•

CHECKSUM DATA.
(see page 10)

```
920 DATA 894,137,259,296,589,808,296,7
95,298,711,854,952,52,584,197,7722
32100 DATA 841,841
```

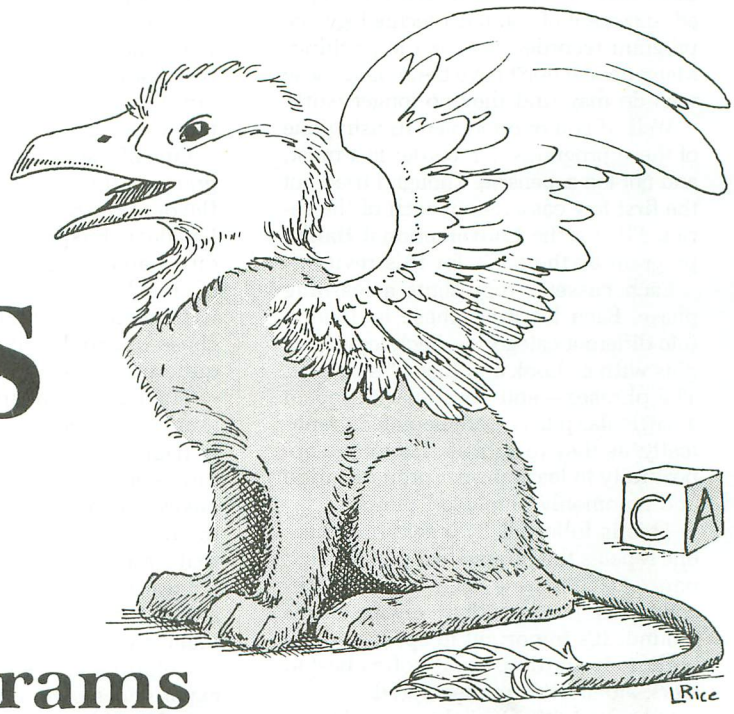
•

Attention Programmers!

ANALOG Computing is interested in programs, articles, and software review submissions dealing with the Atari home computers. If you feel that you can write as well as you can program, then submit those articles and reviews that have been floating around in your head, awaiting publication. This is your opportunity to share your knowledge with the growing family of Atari computer owners.

All submissions for publication must be typed, upper and lower case with double spacing. Program listings should be provided in printed form, and on cassette or disk. By submitting articles to **ANALOG Computing**, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of **ANALOG Computing**. If not accepted for publication, the articles and/or programs will remain the property of the author. If submissions are to be returned, please supply a self-addressed, stamped envelope. All submissions of any kind must be accompanied by the author's full address and telephone number.

Send programs to:
Editor, **ANALOG Computing**
P.O. Box 23, Worcester, MA 01603.



GRIFFIN'S LAIR

Educational Programs Review

by Braden E. Griffin, M.D.

It's been so long since I've written reviews for **ANALOG Computing** that **Griffin's Lair** may be introduced as a new column this month. The editorial staff here is extremely tolerant of my condition, *columnus interruptus*, an affliction separated by time and space. Some say my problem stems from apathy. Who cares?

I'm able to continue with my column, sporadic as it is, only because I've succeeded in addicting the entire staff of the magazine. Not an easy task, when one considers that, as a pediatrician, I had to accomplish this with lollipops and oral polio vaccine. I may need to introduce them to more hard-line drugs (like chewable vitamins) if they start to slip away.

This month's column will review the foreign language programs offered by Atari. I was able to assess the benefits of one of these, having just spent a couple of weeks in Italy. La-dee-dah. Anyway, my dearest mother, who writes a newspaper column, "Dot's Dashes" (gliberness must be hereditary), toured Europe for several weeks a few years ago and got three to four months of columns out of it. I figured I ought to get at least one.

Finally, before we settle down to reviewing, my apologies to Walter B. Gibson (a.k.a. Maxwell Grant), who passed away a few weeks back, are—unfortunately—too late. Now, *no one* knows what evil lurks, eh, Lamont?

CONVERSATIONAL ITALIAN, SPANISH, GERMAN, FRENCH
Atari Corp.
Sunnyvale, CA 94086
16K Cassette $59.95

There are two ways to approach the study of a foreign language. . .three, if one includes, "Let 'em speak English, like everybody else." There's the classic approach, fostered by traditional educational systems, which relies on the development of a basic knowledge of the language. One learns about declension of verbs, predicate nominatives, etc.

Vocabulary and pronunciation skills are stressed, but this highly structured technique does not quickly lead to conversational adequacy. Many students are "turned off" to foreign languages as being mere exercises in rote memory.

Knowledge of one's native tongue is certainly not acquired this way. First we learn to speak, then—much later—the rules governing proper use of the language are taught.

Things seem to have gotten a bit reversed. Maybe Latin is to blame. It lends itself to structured teaching and can't be used conversationally. Everything else just seemed to follow suit. There is a place for the detailed study of a foreign language. For most individuals, though, it doesn't matter *what* a gerundive is, but, more simply, how does one *say* it.

The second approach to learning foreign languages is the "quick and dirty" way. The tools to accomplish this usually include tapes (or records) and books. These methods must be somewhat effective; they continue to thrive.

Atari introduced conversational language programs several years ago, using computers to enhance learning. Each language program is composed of five cassettes and a course book. Already,

there may be a problem. With the rapid advancement of computer technology, the program recorder has been left behind. Many people don't have them, and those who do may find they no longer work.

Well, if you're interested in using one of these programs, a recorder is a must, and not too expensive. Though I tried out the first few cassettes in each of the series, I'll use the **Conversational Italian** program as the basis for this review.

Each cassette represents a learning phase. Each learning phase is divided into different categories. Each lesson begins with a "Look and Listen" segment. The phrases—and vocabulary used in a particular phase—are depicted graphically, as they're spoken. Here's the opportunity to learn proper pronunciation and commonly employed phrases.

This is followed by a session where one repeats the phrases used in the first part. At first, one feels a little self-conscious, particularly if anyone else is around. It's important to approach this with no inhibitions, so it's often best to work alone and uninterrupted.

Other activities include responding to questions and participating in a conversation. Using multiple choice techniques, the appropriate answer is selected. After each learning phase, one studies the course book relating to it.

The course book contains study, practice, and activity sections in each chapter. There is basic grammar instruction, along with written exercises to be completed at the end of each session.

There are ten units or learning phases presented. Unit One covers how to greet people and introduce oneself. Unit Two teaches one to ask how people are and respond in kind.

Other units include sessions on asking and giving directions, telling time, describing things, finding out what people are doing, and learning to ask for what one wants. In the back of the book is a grammar summary and a vocabulary list, as well as answers to the written exercises.

One of the best features of these programs is the way they help develop the flavor of the country while teaching the language. Frequent references to various cities and the country's geography are of great benefit when travelling. Practice with monetary exchange and the purchase of articles likely to be bought is quite effective.

The true test of any educational program, in particular language programs, is trial by fire. Does it really make a difference when travelling abroad? The answer is an unequivocal yes. It may be unfair to use Italy as an example, but that's all I have to go by.

I was told that, if one made an honest attempt to speak the language, the Italians would respond positively and be as helpful and friendly as possible. My experiences certainly justified that advice. We did not travel with a tour group and were frequently placed in situations where no one spoke English. Although my conversational Italian often elicited broad grins and the occasional hearty laugh, the bottom line is that I was able to communicate.

Being placed in a situation where one has no choice but to speak Italian is both frightening and enlightening. A train ride from Rome to Chiavari, near Genoa, was just such an experience. Caught in the middle of an impromptu train employee strike, we were unsure of the proper train to board, where to change trains, and where to get off.

Successfully managing to arrive at our destination without a hitch was quite an ego builder. I so enjoyed conversing only in Italian that, when we arrived in Firenze (Florence), I was a little disappointed to find so many natives speaking English; I wasn't forced to speak their language.

At one point, I ventured off alone to an area not infested with tourists, to seek out bicycle dealers. My intention was to purchase an Italian racing bike. I was pleased that I once again had the opportunity to speak Italian, since the bike shops cater primarily to Italians.

My joy was short lived when I realized that my mastery of the language was not sufficient to give me the confidence to fork over 1,000,000 lira without a better understanding of what I was actually getting. I was able to acquire numerous biking accessories for Christmas gifts. The next time, I'll be ready to take the big plunge.

Another unique experience was playing golf with a caddy who spoke only Italian. It was certainly a quick way to brush up on one's use of numbers. The way I played, I got to use every number in my bag, over and over.

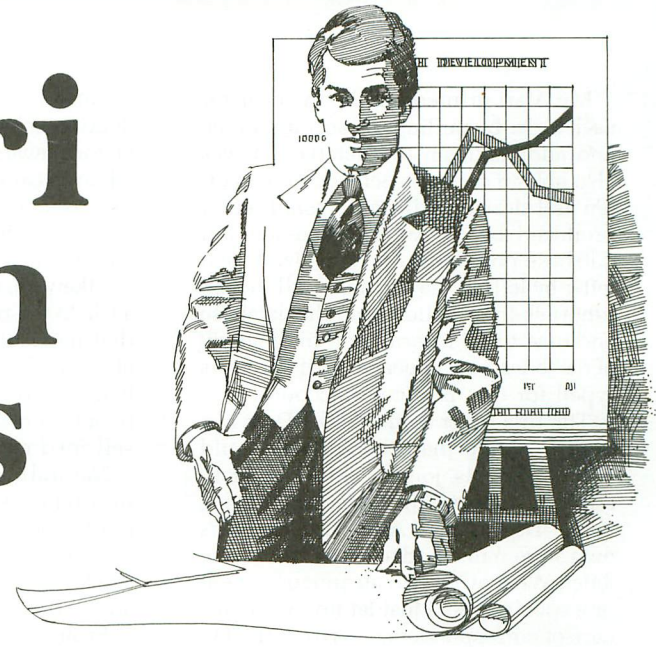
Interestingly, the Italian word for a "Mulligan" is *Mulligan*. I found that out on the very first hole. Though my golf game was less than sparkling, the experience further enhanced the flavor of the country.

My trip to Italy was the most enjoyable vacation I've ever had. This was in no small way due to having used the **Conversational Italian** program and being prepared. It was of such benefit that, soon after I arrived home, I began to review tapes in preparation for my next trip. There's no doubt the other language programs offered would be as helpful.

If anyone is planning a trip to a foreign country, I recommend the development of some basic conversational skills. There are many ways to do this. I found that using Atari's conversational language program was more than adequate.

I'm sure that the fall of the Italian government on the day we left was mere coincidence. Craxi and I only talked for a short while and it was mostly about artichokes.

Atari Can Mean Business



by Daniel A. Silvestri

Atari computers have long been acclaimed as terrific game machines, producing fantastic sound and graphics. Kids love Atari, and software manufacturers have been catering to the needs of the most hungry consumers. Even big kids love Atari in light of the tremendous adventure and strategy games available; some of the simulations of historical battles are as challenging to us swivel-chair generals as the real battles must have been to real generals.

Yet, as the smoke clears on these computer battle simulations, genuine business software can be seen lurking on the horizon—software that can make your business life easier, more efficient and certainly more productive. Can an Atari produce such results? The machine has always stood ready to deliver; now the software is available to bring this efficiency to reality.

Atari, in business.

How can a computer help you manage a small business? Well, if you run a small business of your own, or are thinking about running one, you'll quickly discover the many tedious tasks involved in the process.

A business requires that you have either a product or a service to sell, customers to sell them to, records to help organize your business, some means of acquiring and maintaining customers, and a whole lot more! When you launch your new product or service, you'll immediately become enmeshed in all of the details of record-keeping, billing, contacting customers, planning profit and loss sheets and cash-flow analysis projections. Just these few tasks could keep you busy forever!

In general, if you have to do a job only once, you probably don't need a computer to help you. However, every business I can think of requires you to do many tasks over and over again. For example, perhaps you'd like to contact your customers by mail at least once a month. This is a repetitive task that a computer doesn't mind doing over and over again, while you're doing something else!

To accomplish most of the things that can be done by a computer in business, you'll need a word processor, a database management system and a spreadsheet. Telecommunications software and a modem could help you too, depending on your needs. Ideally, you want the database program and word processor working together, to exchange and use information in one that you've assembled in the other. A stand-alone spreadsheet can serve you well, doing all of your mathematical calculations, and so on, as mentioned above.

Atari, in my business.

For almost two years, I ran a small business doing about $150,000 a year in gross sales. I was selling a product that needed to be assembled: children's jog suits. I bought fabric, contracted with a remote factory in Texas (900 miles from Chicago) to manufacture the finished product and ship it, assembled a sales force in various parts of the country, and did trade shows where I would display my products and take orders for later delivery.

My product was fairly well received, and I found myself with customers all over the country. The business now required that I keep track of what styles had to be made, when they had to be delivered and to whom, how much it would cost to do it, what was in stock, when I would need the most money based on sales needs and purchases required to complete the orders, and so on! My Atari 800 computer was the best partner I could have.

I bought a database and a word processor that could work together, as I mentioned. Then I bought a spreadsheet. I immediately created a database with each account's name, address, order amount and date of delivery. I created a second database which registered all of the various styles, colors and sizes that had to be made per delivery period. Then I loaded in my spreadsheet and, on a month-by-month basis, figured out how much cash I had on hand, how much would be received from my customers in a given month, and how much I would need in order to produce the amount of the product I needed for delivery in a given period.

Once this initial stage of inputting the information was complete, maintenance became simple and sweet. I was now armed with the kind of information retrieval system that could make my work easier and more efficient!

My Atari printed out the invoices for each order by pulling the account name and address from the database. I was also able to contact each account as often as I desired with a *personal* letter, produced by my word processor and database programs! I personalized each letter by letting the program pull the appropriate information from the database and insert that information in the body of the letter, so it appeared as if each was typed for that particular account.

The first letter could read, "Dear Mr. Smith," while the second letter could read, "Dear Ms. Jones." This is far more effective than a "form" letter that just reads "Dear Buyer." Also, the customer's own store name and address was in the letter! Whenever I had an announcement or a special sale, I just let my Atari take care of correspondence, writing the letters and doing the envelope labels!

I also needed to keep up on the production schedule and make my remote factory aware of what had to be made, when. Again, my Atari went to work for me, as I used my modem to transmit special production orders to the factory's computer.

Likewise, the factory would transmit a file to me reporting what was in stock that particular day. This made it more efficient for me to sell what we had on hand, or to put the word out to the sales people across the country as to what to sell for immediate delivery.

Naturally, all of the sales people were in a database, so I could quickly do a mail merge to them concerning these matters and include a printout of in-stock items! This helped increase sales and reduce waste.

From billing to past due notices, production scheduling to delivery orders, mail merges to shipping dates, my Atari helped out tremendously.

I even created an entire business plan of about 50 pages with the word processor, basing totals on numbers derived in a spreadsheet analysis of my business needs. You can use these business plans as a road map for running your business, or as a vehicle for summarizing the current state of affairs of your business and its plans for the future—so you can raise additional capital, should you need to expand.

Banks require profit and loss statements, cash-flow projections, and so on, before considering the possibility of a loan for your business. Your electronic spreadsheet can make this project easier for you, allowing you to do some "what if" analysis for your projected needs. As you enter new, experimental numbers, rows and columns are automatically recalculated, making this task far less oppressive.

It's a good idea to do this whether or not you're seeking investment or loan capital, just so you have a better handle on what your business potential can be. It's just good business sense to anticipate potential problems or cash needs down the road *before* you get there.

Once you get involved with using the Atari to help your existing business or to start a new one, you'll discover the wide variety of ways that it can help you be more successful.

Selecting the right software.

Now that you have a better idea of how your Atari can help you in business, you need to know how to choose the software that will meet and exceed your needs.

I say "meet *and* exceed" your needs, because the software you buy must allow your business or usage to grow. After all, you're in business to make a profit and to attain a successful growth rate as you continue in business.

I believe that reviews of software (and hardware) are your first level of help in the selection process. I have *always* found reviews in reliable magazines invaluable in assisting me to make an intelligent choice.

Business Tools

A. Database
An organized system for storing or cataloging information on disk for later retrieval in various forms.
1. Names and addresses
2. Information on a collection

B. Word Processor
A program that lets you create and edit letters, documents and reports on-screen before printing, and also lets you store that information on disk.
1. A business letter to accounts
2. An article for a magazine

C. Spreadsheet
A program that lets you create and store rows and columns of names and numbers to use for mathematical calculations.
1. A monthly cash-flow projection
2. A home budget

Figure 1.

Second, I firmly believe in the tried-and-true method. If you know someone who has used a piece of software (or hardware) for the same or similar purpose as you intend to, ask for a personal review of it. See if the user likes the product. Does it do what it says it will do? How much effort does it take to actually learn and use the product? And, most importantly, does it meet and exceed the user's needs? These word-of-mouth recommendations can save you lots of legwork and reduce your headaches as well. Look to local Atari user groups and clubs for assistance on this level.

Last, if at all possible, try the software yourself before you buy it. There's no substitute for getting your own hands-on experience to see if it will answer all of the above questions to your satisfaction.

With the right insight and the proper combination of products, you can turn your Atari into a lean, mean business machine that can help you become more profitable and more successful!

PORTFOLIO MANAGER
BASIC BYTE, INC.
P.O. Box 924
Southfield, MI 48037
(313) 540-0655
48K Disk $39.95

by Daniel A. Silvestri

Atari users have always had an abundance of entertainment software for the would-be adventurer, sports fanatic or arcade junkie. But when it comes to practical application software packages, how does Atari fare? With their new **Portfolio Manager**, Basic Byte, Inc. helps balance the scales.

Portfolio Manager is a stock management system that adds to the personal software available for Atari. The program was released first for IBM, Apple and Commodore computers, and, finally, the Atari version is here!

The designers at Basic Byte tell me that this is the first of three related programs to be released for Atari. **Options Management** and **Graphics Analysis** are promised to follow shortly. **Graphics Analysis** will enhance **Portfolio Manager**, by providing a tool with which to evaluate the performance of your stock portfolio.

Basically, **Portfolio Manager** is a stock *management* program and makes no real claims to be an evaluator. You'll therefore get no buy-or-sell advice, no method of tracking the performance of a specific stock and no graphs. There are other programs on the market that will do a lot more, but they also cost a lot more. If you want a program that will keep a record of your stock transactions (with no fancy frills and no fancy price), then this one may interest you.

The program is written in BASIC, so execution is a bit slow. However, you'll forget all that, as the wait primes you for viewing what your portfolio is worth! **Portfolio Manager** is menu-driven, so there's almost no need for the terse documentation that accompanies it. Within minutes of opening the attractive book-like package, you'll be entering pertinent statistics of your personal stocks. So what *can* you do with this package?

You have nine selections from the main menu. The first thing to do is select option 1, "Create New Portfolio," which allows you to enter an account name and number, the stock name, current quote, number of shares, total cost of the transaction (including broker's fees) and, finally, the purchase date.

One by one, you enter up to a total of 70 different stocks and 300 total transactions. If these are stocks you've owned for a while, you may want to enter any dividends that you've received by using option 4, "Dividends Payment and Report." In any event, it's probably time to move to option 2, the "Disk I/O Menu."

From this menu, you can load data (a previously SAVEd file), save data. display the disk directory (in case you forget a filename) and format a new data disk. You must SAVE your data after every update. It's also a good idea to SAVE it under different filenames each time you update (Stock1, Stock2, and so on).

By doing this, you can maintain your own performance record of each stock. The program will retain your adjusted purchase price (adjusted for broker's fees) and the current market price. It *won't* retain a performance schedule, so you won't know if the current price is higher or lower than the price a month ago. For this reason, you can't track a stock to look for trends in the market; you can't see peaks and valleys—or overall performance—at all.

Perhaps the **Graphics Analysis** program promised to follow will alleviate this problem; for now, you're on your own, unless you SAVE as I suggested. You can print out these reports each time you update and create your own track record.

From option 3, the "Change Portfolio Menu," you can add stock recently purchased, or record the transaction if you happened to sell stock. Stock splits will adjust the number of shares, and review/change portfolio allows you to look at all the information in the portfolio and to make any needed changes. File housekeeping lets you purge the buy, sell or dividend records for individual stocks. Option 4, "Dividend Payments and Report," records any dividend payments and gives a year-to-date total.

Okay, it's time for an update. You've owned your stocks for some time now, and you want to get a clear idea as to the portfolio's value. Well, first select option 5, "Enter Current Quotes." Here, you'll be prompted to enter the latest quote for each stock owned. One by one, they'll be brought to the screen, and you'll be allowed to change the per-share price (or leave it the same, if it's unchanged). After you're finished, move to option 6, "Portfolio Value Display."

From the option 6 menu, you can get what you've been waiting for: the answer to the question, "How much is my portfolio now worth?" When I first received the program to review, I spent a theoretical $2,350 on eight of my favorite picks from the New York Stock Exchange. I updated for about a month or so, and, at this writing, my investment is worth $2,187.70, down a total percentage change of 7%. In dollars and cents, I'm down $162.50!

I had made printouts after each up-

date, so that I could see where I went wrong—what I should have sold and when. You can display or print information on your entire portfolio, listing the stock, number of shares, cost/share at purchase, current quote, present value, percentage change (up or down) and dollar gain/loss. Totals for all categories will also be displayed or printed.

Your "Capital Gain/Loss Display" is accessed through option 7, which will display or print long- and short-term capital gains and losses. Option 8, "Print Reports," will let you get a hard copy of the dividends payment report, capital gain/loss report, and portfolio value report—if you have a printer. Finally, option 9 from the main menu exits you from the program.

Portfolio Manager will tidy up your desk and, maybe, thin out your file folder. It will store whatever you tell it when prompted for information. It's easy to use and automatically totals shares, cost, value, and so on. Though the back jacket cover claims "you'll be able to react faster to changing market conditions, spot trends to buy and sell," I think the spotting is up to you, and you'll have to shuffle through your printouts to do it.

I'm looking forward to Basic Byte's upcoming additions, **Options Management** and **Graphics Analysis**, to see if they'll increase the productivity of this program.

All in all, **Portfolio Manager** is a dedicated database that can handle an adequate amount of information and can also be useful as a practice tool for those not yet involved in the roller coaster world of the real stock market.

Daniel A. Silvestri taught at a university before turning to sales. Now Retail Account Manager in Illinois and Wisconsin for Ashton-Tate, a major manufacturer of business software, he enjoys adventure games, personal management and business software.

Winter CES

A first look

by Arthur Leyenberger

This report is being filed directly from the floor of the 1986 Consumer Electronics Show in Las Vegas, Nevada. In fact, I'm standing in front of an Atari 130XE computer at the crowded Atari booth, using an XM 301 modem and **XE-Term**.

Enough about the particulars; on to the exciting information that Atari has announced here. The big news of the 1986 Winter CES is Atari's success with both its 8- and 16-bit lines. The XL and XE computers have sold well over the past Christmas season, and products for these machines continue to be forthcoming.

Atari's 16-bit ST computers have done well since their introduction, too, and a new model was announced here at CES. . . Enter the new high-end Atari, the 1040ST.

This model should please a lot of people, since it has—as built-in features—many functions attractive to both "power" users and casual users.

Specifically, the 1040ST contains 1 megabyte of RAM, a built-in double-sided disk drive, an RF modulator and video outputs—as well as all the ports, jacks and features of the previous model, the 520ST. The price for all this remarkable technology is $999 for the monochrome system and $1200 for the RGB system.

The 1040ST arrives with its TOS on ROM, and has several applications programs. Packed with the system, these include the **NEO-Chrome** paint program, the **1st Word** word processor, ST BASIC and ST LOGO. The computer has the power supply for its keyboard and disk drive built in, rather than as separate external boxes. The 1040ST will only be sold at computer retail stores, which will also handle some of the new peripherals.

One new peripheral announced is the SHD-204, a 5¼-inch hard disk for the ST computers. The SHD-204, with its 20 megabytes, will retail for $699 and is to be available by March.

More interesting news from Atari: the latest 520STs—which will also have the TOS on ROM, and an RF and composite video monitor output—will now be sold at mass merchandising establishments. Stores like Toys 'R' Us, Sears and perhaps K-Mart will also get to sell the single-sided disk drive.

All ST components have been "unbundled" for the mass merchandisers. This assures maximum flexibility for the individual wanting to get involved in using the Atari STs.

Still more news: the company showed a new, improved 2600 video game that will sell for $50. The almost-forgotten 7800 ProSystem game machine has been resurrected and will sell for under $80.

Atari claims that the continuing purchases of the 2600 led them to believe that the video game market isn't as dead as some would have imagined. Well over a million 2600 VCS systems were sold in 1985, according to Atari. Amazingly, more than 25 million 2600s have been sold since the mid-seventies.

The 7800 ProSystem is an advaced video game that boasts the best color graphics of any currently available game machine. Not only does the 7800 offer better graphics and color than other systems, but the number of video objects that can appear on-screen at one time has been increased to 100. Also, the size and shape of the objects can be manipulated more easily. This 7800 is identical to the one introduced by the "old" Atari in 1984. For a complete description, see issue 21.

Software for the 8-bit computers was also shown and is supposed to be coming out in a few weeks. This includes programs which had been demonstrated previously: **Silent Butler**, a bookkeeping program for paying bills; **Star Raiders II**; **Music Painter**; and **Atari Planetarium**. **Star Raiders II** is a combination of the never-released **Last Starfighter** game and the original, excellent **Star Raiders**.

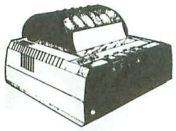
That completes my initial, brief CES report. I must thank Atari Corp., whose cooperation in allowing me to use their equipment—in the midst of a very hectic booth—shows that they're serious about providing the best computers and support to the Atari end user.

ATARI LIQUIDATION

ATARI 800XL

$59.99

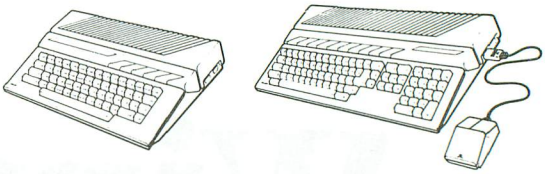
ATARI PRINTERS



1020 Printer **$29.99**

1027 Printer.....**$119.00**

NEW MODELS



ATARI 520 ST.....CALL
ATARI 130XE.....CALL

ATARI 1050

DISK DRIVE
DOS 3.0

$CALL

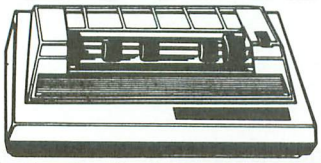
INDUS

ATARI GT DISK DRIVE
$219.00

COMREX

DOT MATRIX PRINTER

- 50 cps
- Automatic Pin Feed
- Cable included



Direct connect, no interface needed **$89.99**

Atari ROM Specials

CX4005	Video Easel
CX4008	Space Invaders
CX4011	Star Raiders
CX4012	Missile Command
CX4013	Asteroids
CX4024	Galaxian
CX4025	Defender
CX8030	E.T. PHone Home
CX8039	Eastern Front

YOUR CHOICE
$2.99 ea.
ROM ONLY

MODEMS

ATARI 1030.....**$59.99**
ATARI 835.....**$39.99**

Digital Devices
U-Call Pocket Modem.....**$99.99**
Anchor Volksmodem.....**$59.99**
Supra 1000E.....**$74.99**
Supra 1200AT.....**$189.00**
Supra 1200ST.....**$189.00**

ATARI Arcade Champ

INCLUDES:

- Pac Man
- Qix
- ROM Holder

$9.99
Order No. AA7102

INTERFACES

Digital Devices

DDA01.....**$49.99**
DDA02.....**$59.99**
DDA03.....**$69.99**
ICD PR Connection.....**$59.99**
Micro Print.....**$39.99**
MMP-1150.....**$49.99**

ATARI 850 INTERFACE

We found 200! Maybe the last available in the World!

$149.00
Order No. AA850

ROKLAN Software

Deluxe Invaders	$3.99
Anti Sub Patrol	$3.99
Gorf	$4.99
Wizard of Wor	$4.99
Space Journey	$4.99
Journey to Planet	$4.99
Rack-Em-Up	$4.99
Diamond Mind	$4.99

Avalon Hill Software

YOUR CHOICE **$3.99 each**

- Conflict 2500
- Stocks & Bonds
- Guns of Fort Defiance
- Football Strategy
- Flying Ace
- Panzer Jagd
- Viet Cong
- GFS Sorcress
- Market Forces
- Facts in Five
- Space Station Zula
- Paris in Danger
- Vorrak
- Gypsy
- TGIF
- Divex
- Legionnaire

DISKETTES

Nashua 5¼" SS/DS.....**$8.99**
Maxell 3½" SS/SD.....**$29.99**
Maxell 5¼" SS/SD.....**$13.99**

••• NOTE •••

All items except "new models" are sold "as is" with a 15 day exchange policy. Units available for sale for the most part do not have instruction manuals and/or original packaging. Some units may have minor cosmetic damage. All units are in new working condition. All items are in limited quantities.

COMPUTER MAIL ORDER

CALL TOLL-FREE 1-800-233-8950

477 East Third Street, **Dept. B703,** Williamsport, PA 17701

SHIPPING: Add 3%, minimum $7.00 shipping and handling on all orders. Larger shipments may require additional charges. All items subject to availability and price change. Returned shipments may be subject to restocking fee.

VISA · MasterCard · AMERICAN EXPRESS · DINERS CLUB

1-800-268-3974 Ontario/Quebec

1-800-268-4559 Other Provinces

CANADIAN ORDERS
All prices shown are for U.S.A. orders. Call the Canadian Office for Can. prices.

1-416-828-0866 In Toronto

Telex:06-218960
2505 Dunwin Drive, Mississauga, Ontario Canada L5L1T1

CIRCLE #110 ON READER SERVICE CARD

Boot Camp

by Tom Hudson

Are you ready to dive further into the world of 8-bit assembly language? I hope so, because this issue we'll look at two programs that further illustrate the use of the Atari Central Input-Output (CIO) system.

Program number one.

The first program we'll look at, shown in Listing 1, is a modification of the last **Boot Camp** program. As you recall, we wrote a program which copied any input file (keyboard, disk, etc.) to the screen. This time, we'll have one that does just the opposite. It will copy whatever you type on your computer screen to a disk file, the printer, or any other device.

If you take a look at issue 38's **Boot Camp**, you'll notice that this issue's Listing 1 is a relatively minor modification of that program. The filename input section (Lines 290-480 in both programs) is identical, since the entry of a filename through the keyboard remains the same, regardless of the program's function.

Lines 520-720 make an attempt to open the file indicated as output. IOCB number 1 is used for the output file (Line 520).

Lines 520-630 perform the open operation. If the file was opened successfully, Line 640 branches to FILEOK, where actual processing begins.

If the file couldn't be opened due to an invalid filename or device specifier, Lines 650-670 print an error message (OPNERR) to the user, informing them that the file can't be opened and they must try again. After printing the message, IOCB number 1 is closed (Lines 680-710), and the program jumps back to the GETFN routine, to accept another filename.

If the file was successfully opened, the FILEOK routine prints a message to the user that it's ready to accept text. The Atari screen editor (opened by the system as IOCB number 0) will accept lines of text (text records) up to three screen lines long (120 characters). You can type lines of text as long as you like, up to the 120-character limit, terminating each with the RETURN key. As you enter each line, the program will write the text to the specified device.

If you indicated D:TEXTFILE as your output filename, the text will be written to a disk file named TEXTFILE. Entering *P:* as your filename will cause the text to be written to the printer. If you don't have the device specified in the filename connected to your computer, the open operation will fail; you'll be asked to enter another filename.

To tell the computer you're done entering text, press CTRL-3 (the CTRL and 3 keys simultaneously). The output file will be closed. As mentioned in earlier installments of **Boot Camp**, the CTRL-3 sequence on the keyboard indicates an end-of-file, the code that tells the system there's no more input available.

Lines 860-980 handle text entry. The text editor is used to receive text (IOCB number 0), which is placed into a block of memory labeled BUFFER (defined in Line 1970 as 128 bytes in length). Lines 930-960 tell the system to limit the input text length to 128 bytes. In this case, the system automatically limits text to 120 characters, because of the text editor's built-in limitation. It's a good idea to set up the buffer length values anyway, to prevent possible trouble.

If there's a problem getting a text record, Line 980 branches to the BADRD (BAD READ) routine, which checks the type of error that occurred.

If the line of text was read successfully, Lines 1020-1130 write the text to the output device. Lines 1030-1060 point to the text buffer; Lines 1090-1110 tell the maximum number of bytes to write ($FFFF, or 65535); and Line 1120 writes the text.

If the write operation fails for some reason (printer not ready or disk full, for example), Line 1130 branches to BADWRT (BAD WRITE) to report the error. If the write was successful, the program branches back to the READIT routine, where another line of text will be accepted from the screen editor.

Lines 1180-1260 check the error condition when an error occurs on text input. If the error code (found in the 6502 Y-register) is a *136*, the end-of-file indicator has been read, and the program jumps to the QUIT routine to finish processing.

Any other error on input is handled by Lines 1230-1260. These print an error message and branch back to get another line of text at READIT.

Lines 1270-1290, which are labeled BADWRT, are executed when an error occurs when writing to a file. The program performs a jump to subroutine (JSR) to WEPRNT (WRITE ERROR PRINT), which is a short subroutine to print the write error message. This message may be printed elsewhere in the program, so, in order to save memory, it's made into a short subroutine that may be called from anywhere in the program.

After printing the message at the WEPRNT subroutine, the program jumps to the QUIT routine. The JMP instruction isn't necessary in this case, because the QUIT code starts with the next instruction. It's good programming practice, however, to put the JMP in—you (or someone else) may later add a routine at this point and forget that the BADWRT routine falls through to the QUIT code. A JMP only takes a few microseconds longer to execute and doesn't noticeably affect the program's speed.

Lines 1330-1450, labeled QUIT, perform the final processing necessary to finish the program. First, in Lines 1340-1380, the program closes IOCB number 1, the output file. Because of the way the disk operates, final data may not actually be written to the disk until this point in the program. If the disk is full, a write error may be encountered. For this reason, we do a check, to see if there was an error on the write operation and branch to BADEND if something went wrong. If we didn't check for an error at this point, one could occur on the close operation, and the user would never notice it.

If the output file was closed properly, the program prints the *DONE* message and exits with the BRK instruction.

The BADEND routine, as mentioned above, is executed if an error occurs on the close operation. It calls the WEPRNT subroutine in order to print an error message, then executes a BRK to exit.

The WEPRNT (WRITE ERROR PRINT) subroutine, Lines 1490-1530, is a simple subroutine which prints the *WRTERR* (WRITE ERROR) message on-screen, using the PRINT subroutine. It then returns to the calling code via the RTS instruction.

This is a good example of the use of short subroutines to shorten your programs—rather than duplicate this print code twice in the program (in the BADWRT and BADEND routines), we make it a subroutine. This saves space and also makes it easier to change the program if we need to alter the write error message (the change only has to be made in one place).

Lines 1540-1770 make up the PRINT subroutine, which we wrote in issue 38's installment of **Boot Camp**. This routine simply prints the specified string to the screen.

Lines 1780-1950 are the various text messages used by the program. All of these must be terminated by an ATASCII End-Of-Line (EOL) character (*$9B*) to print properly.

Line 1960 is the block of memory (20 bytes) reserved for the output filename.

Line 1970 is a 128-byte block of memory reserved for the text input buffer.

Comparing this program to the one in issue 38, you can see that functions are easily changed by modify-

ing a few lines and using prewritten "building blocks," like the PRINT subroutine. You may want to save useful subroutines to disk for later—you can merge them with your code and write programs more quickly. Be sure to document the operation of the subroutine in comments (as in Lines 1540-1600), so that you can remember how the subroutine works!

Program number two.

The second program this time is a further modification of the first, so it can use any input file (disk, screen editor, etc.) and copy the output to any other file. You may, for instance, copy from one disk file to another, from the disk to the screen, from the screen editor to the printer, and so forth. The device independence of CIO allows you almost unlimited flexibility. Let's take a look at the program.

Listing 2 is the universal copy program. Just looking at it, you can see its similarity to Listing 1—it has roughly the same filename entry (two filenames this time, though, instead of just one), a copy loop reading the first file and writing to the second, error message handling, and the PRINT subroutine. It's just an extension of the principles we've used up to this point, scaled up slightly.

In this program, we'll accept two filenames, one specifying the input file, the other the output file. Because we don't know beforehand what devices will be used, we must open both files to separate IOCBs.

In the earlier examples, one file was usually the screen editor, opened by the operating system as IOCB number 0. We have eight IOCB's available (actually seven, since IOCB 0 is used for the screen editor), so we'll pick IOCB number 1 for the input file and IOCB number 2 for the output file (we could use any of the seven available IOCBs for the program; I just picked 1 and 2 arbitrarily). Simple enough.

Lines 290-480 are quite similar to their counterparts in Listing 1, but you'll notice that the labels have changed slightly. We must be able to differentiate between the two file input sections and the two filenames, so the files are called FNAME1 (the input file) and FNAME2 (the output file). The labels for getting the files are correspondingly labeled GETFN1 and GETFN2.

After accepting the first filename and placing it in FNAME1, we must try opening the file for input—to be sure it exists—before moving on and accepting the output filename. If, for instance, the user types *P:* (the printer) as the input device, the system won't want to open it as input. The printer is an output-only device, so CIO will return an error. If the input file is a disk file, opening it for input makes sure it's actually on the disk before proceeding. If there's any error in opening the file, the program asks for another filename.

Once we're sure the input file is ready to go, we move on to the GETFN2 routine, Lines 760-1160. This routine is virtually identical to the GETFN1 section, except that the specified file is opened as output, using IOCB number 2 (Line 960).

If an error occurs opening for output (this could happen if the printer wasn't ready or the disk drive power was off), the program prints a message and asks for another filename. You've seen all this code before, so I won't go into a painful, line-by-line analysis.

Now that both files are open, the input file using IOCB number 1 and the output file using IOCB number 2, we're ready to start copying. The easiest copy, which we'll employ, is a byte-by-byte copy. That is, we'll read 1 byte from the input file and immediately write it to the second file, repeating this process until the end-of-file on the input file is reached. There are other, faster ways to get the job done, but, for the time being, let's keep it simple.

Lines 1210-1230 print a message which informs the user that the program is beginning to copy the files. If you've specified a file other than the screen editor as input, sit back and relax as the computer takes care of the "dirty work." If you're using the screen editor as input, type the text you want, ending each line with the RETURN key. When finished, press CTRL-3 to tell the system you're done.

Lines 1270-1350, labeled COPYIT, are the section of the program which reads 1 byte from the input file. The main point of interest here is that Lines 1310-1330, which set the number of bytes to read, set the byte count to 0.

As explained in an earlier **Boot Camp**, this tells CIO to read 1 byte only, and to place it into the 6502 accumulator. We'll have to take certain precautions to be sure the contents of the accumulator aren't altered, because this is the data we're copying! If the read operation resulted in an error condition, Line 1350 branches to BADRD, to determine what kind of error occurred and handle it properly.

If the read was successful, we have a byte from the input file sitting in the accumulator and are ready to write it to the output file. The only problem is that we need to use the accumulator to set up the command bytes in IOCB number 2 for the write operation. If we alter the accumulator, we'll clobber the data.

No problem—we'll simply save the data from the accumulator somewhere else, putting it back in the accumulator when we're ready to use it. The most convenient place to stick the byte in this case is the Y-register. We aren't using that for anything and we won't be changing it to set up IOCB number 2, so a simple TAY (Transfer Accumulator to Y-register) in Line 1390 moves the byte to a safe place.

We now set up IOCB number 2 in the normal fashion for a 1-byte write from the accumulator (Lines 1400-1450), put the byte back in the accumulator from the Y-register with the TYA (Transfer Y-register to Accumulator) instruction, and write it to the file in Line 1470.

If an error occurred on the write, Line 1480 will branch to the error routine, BADWRT. Otherwise, the program branches back to COPYIT to continue copying the next byte. This process continues until the EOF on the input file is reached.

This operation illustrates one of the most difficult tasks facing the assembly-language programmer: you must be aware of what's in each register and keep track of what happens to each of them through the execution of your program. With one mistake in coding, you can blow away important data.

The moral: always be aware of the 6502 register contents—*never* assume that a piece of data is safe. If in doubt, save it somewhere. The few microseconds you waste saving a value will never be noticed, and they could save your reputation, as well as a great deal of time!

Lines 1530-1640 are the error-handling routines for the copy program. They're similar to those of Listing 1, but, instead of going back for another line of input if an error occurs, this program will print an error message and quit, because lost data of any kind means the file is unusable when not the screen editor.

Lines 1680-1840 close the two files, printing an error if the output file close produces an error.

The remaining lines of the program are explained in Listing 1, except for the two filenames, FNAME1 and FNAME2. These are the 20-byte areas of memory reserved for the two filenames. Note that there's no buffer used in this program for data, because we're using the accumulator to hold each byte as it's read and written.

Play around with this program and modify it so that the two filename input operations (Lines 360-480 and 830-950) are reduced to a single subroutine. It's not too hard—I'll show a way to do it next time. We'll also start looking at how to work with the computer's graphics through CIO.

Listing 1.

```
0100      .OPT NO LIST
0110 COLOR4 = $02C8
0120 ICCMD = $0342
0130 ICSTA = $0343
0140 ICBAL = $0344
0150 ICBAH = $0345
0160 ICBLL = $0348
0170 ICBLH = $0349
0180 ICAX1 = $034A
0190 ICAX2 = $034B
0200 CIOV =  $E456
0210 ;
0220 ;SET STARTING ADDRESS
0230 ;
0240      *=  $6000
0250 ;
0260 ;PRINT FILENAME ENTRY PROMPT
0270 ;
0280      CLD          ;BINARY MODE!
0290 GETFN
0300      LDA #PROMPT/256 ;HI PART IN A
0310      LDY #PROMPT&255 ;LO PART IN Y
0320      JSR PRINT    ;PRINT PROMPT!
0330 ;
0340 ;ACCEPT FILENAME FROM EDITOR
0350 ;
0360      LDX #$00     ;EDITOR: IOCB #0
0370      LDA #$05     ;GET RECORD...
0380      STA ICCMD,X  ;COMMAND
0390      LDA #FNAME/256 ;POINT...
0400      STA ICBAH,X  ;TO...
0410      LDA #FNAME&255 ;FILENAME...
0420      STA ICBAL,X  ;BUFFER
0430      LDA #20      ;MAXIMUM NAME...
0440      STA ICBLL,X  ;= 20 CHARS
0450      LDA #0       ;(20 IN LO,
0460      STA ICBLH,X  ;0 IN HI)
0470      JSR CIOV     ;GET RECORD!
0480      BMI GETFN    ;RETRY IF ERROR
0490 ;
0500 ;TRY OPENING FILE FOR OUTPUT
0510 ;
0520      LDX #$10     ;USE IOCB #1
0530      LDA #$03     ;SET UP...
0540      STA ICCMD,X  ;OPEN COMMAND
0550      LDA #FNAME/256 ;POINT...
0560      STA ICBAH,X  ;TO...
0570      LDA #FNAME&255 ;USER'S...
0580      STA ICBAL,X  ;FILENAME
0590      LDA #$08     ;OPEN FILE...
0600      STA ICAX1,X  ;FOR OUTPUT
0610      LDA #$00     ;AUX2...
0620      STA ICAX2,X  ;NOT USED
0630      JSR CIOV     ;OPEN IT!
0640      BPL FILEOK   ;OPENED OK!
0650      LDA #OPNERR/256 ;UH-OH, PRINT
0660      LDY #OPNERR&255 ;ERR MESSAGE
0670      JSR PRINT    ;USING SUBROUTINE
0680      LDX #$10     ;BETTER...
0690      LDA #$0C     ;CLOSE...
0700      STA ICCMD,X  ;IOCB #1...
0710      JSR CIOV     ;TO PLAY IT SAFE
0720      JMP GETFN    ;TRY AGAIN!
0730 ;
0740 ;PROMPT THE USER TO ENTER TEXT
0750 ;
0760 FILEOK
0770      LDA #PROMP2/256 ;POINT TO...
0780      LDY #PROMP2&255 ;TEXT PROMPT
0790      JSR PRINT    ;PRINT IT!
0800 ;
0810 ;NOW READ TEXT RECORDS FROM
0820 ;SCREEN EDITOR AND WRITE THEM
0830 ;TO THE DISK FILE.
0840 ;
0850 READIT
```

```
0860      LDX #$00     ;IOCB #0
0870      LDA #$05     ;SET TO...
0880      STA ICCMD,X  ;GET RECORD
0890      LDA #BUFFER/256 ;POINT...
0900      STA ICBAH,X  ;TO...
0910      LDA #BUFFER&255 ;INPUT...
0920      STA ICBAL,X  ;BUFFER
0930      LDA #128/256 ;MAXIMUM...
0940      STA ICBLH,X  ;READ...
0950      LDA #128&255 ;128...
0960      STA ICBLL,X  ;CHARACTERS
0970      JSR CIOV     ;READ IT!
0980      BMI BADRD    ;READ ERROR!
0990 ;
1000 ;RECORD'S OK, WRITE IT!
1010 ;
1020      LDX #$10     ;IOCB #1
1030      LDA #BUFFER/256 ;POINT TO...
1040      STA ICBAH,X  ;THE RECORD...
1050      LDA #BUFFER&255 ;BUFFER WITH
1060      STA ICBAL,X  ;OUR TEXT
1070      LDA #$09     ;PUT RECORD...
1080      STA ICCMD,X  ;COMMAND
1090      LDA #$FF     ;SET UP...
1100      STA ICBLL,X  ;MAXIMUM...
1110      STA ICBLH,X  ;RECORD SIZE
1120      JSR CIOV     ;WRITE IT!
1130      BMI BADWRT   ;WRITE ERROR!
1140      JMP READIT   ;LOOP FOR MORE!
1150 ;
1160 ;CHECK ON ERROR CONDITION
1170 ;
1180 BADRD
1190      CPY #136     ;EOF?
1200      BNE NOTEOF   ;OTHER ERROR!
1210      JMP QUIT     ;AND QUIT!
1220 NOTEOF
1230      LDA #RDERR/256 ;GOT AN ERROR,
1240      LDY #RDERR&255 ;PRINT ERROR..
1250      JSR PRINT    ;MESSAGE
1260      JMP READIT   ;AND TRY AGAIN!
1270 BADWRT
1280      JSR WEPRNT   ;PRINT ERROR MSG
1290      JMP QUIT     ;AND QUIT!
1300 ;
1310 ;CLOSE FILE, END!
1320 ;
1330 QUIT
1340      LDX #$10     ;IOCB #1
1350      LDA #$0C     ;SET CLOSE...
1360      STA ICCMD,X  ;COMMAND
1370      JSR CIOV     ;CLOSE IT!
1380      BMI BADEND   ;BAD CLOSE!
1390      LDA #DONE/256 ;EOF, WE'RE...
1400      LDY #DONE&255 ;ALL DONE...
1410      JSR PRINT    ;PRINT MESSAGE
1420      BRK          ;AND EXIT
1430 BADEND
1440      JSR WEPRNT   ;PRINT ERR MSG
1450      BRK          ;AND EXIT!
1460 ;
1470 ;WRITE ERROR PRINT SUBROUTINE
1480 ;
1490 WEPRNT
1500      LDA #WRTERR/256 ;POINT TO...
1510      LDY #WRTERR&255 ;ERR MESSAGE
1520      JSR PRINT    ;PRINT IT
1530      RTS
1540 ;
1550 ;PRINT SUBROUTINE:
1560 ;
1570 ;INPUT:
1580 ;ACCUMULATOR: HI ADDR OF STRING
1590 ;Y REGISTER:  LO ADDR OF STRING
1600 ;
1610 PRINT
1620      LDX #$00     ;USE EDITOR
1630      STA ICBAH,X  ;MOVE A TO HI
1640      TYA          ;PUT Y IN A REG.
1650      STA ICBAL,X  ;MOVE IT TO LO!
1660      LDA #$09     ;SET UP...
1670      STA ICCMD,X  ;PUT RECORD
1680      LDA #$FF     ;SET BUFFER...
1690      STA ICBLL,X  ;LENGTH...
1700      STA ICBLH,X  ;TO MAXIMUM
1710      JSR CIOV     ;PRINT IT!
1720      BMI FATAL    ;ERROR!
1730      RTS          ;OK, RETURN
1740 FATAL
1750      LDA #$34     ;CHANGE...
1760      STA COLOR4   ;BORDER COLOR
1770      BRK          ;AND EXIT
1780 PROMPT
1790      .BYTE "ENTER OUTPUT "
1795      .BYTE "FILENAME "
1800      .BYTE "(INCLUDE D:)",$9B
1810 PROMP2
1820      .BYTE "ENTER TEXT, "
1825      .BYTE "TYPE CTRL-3 "
1830      .BYTE "TO QUIT.",$9B
1840 OPNERR
1850      .BYTE "CAN'T OPEN FILE! "
1860      .BYTE "-- TRY AGAIN",$9B
1870 RDERR
1880      .BYTE "*** ERROR GETTING "
1890      .BYTE "RECORD! ***",$9B
1900 WRTERR
1910      .BYTE "*** ERROR WRITING "
1920      .BYTE "FILE! ***",$9B
1930 DONE
1940      .BYTE "*** FILE WRITE "
1950      .BYTE "COMPLETE! ***",$9B
1960 FNAME *= *+20
1970 BUFFER *= *+128
1980      .END
```

Listing 2.

```
0100      .OPT NO LIST
0110 COLOR4 = $02C8
0120 ICCMD = $0342
0130 ICSTA = $0343
0140 ICBAL = $0344
0150 ICBAH = $0345
0160 ICBLL = $0348
0170 ICBLH = $0349
0180 ICAX1 = $034A
0190 ICAX2 = $034B
0200 CIOV =  $E456
0210 ;
0220 ;SET STARTING ADDRESS
0230 ;
0240      *=  $6000
0250 ;
0260 ;PRINT FILENAME 1 ENTRY PROMPT
0270 ;
0280      CLD          ;BINARY MODE!
0290 GETFN1
0300      LDA #PFILE1/256 ;HI PART IN A
0310      LDY #PFILE1&255 ;LO PART IN Y
0320      JSR PRINT    ;PRINT PROMPT!
0330 ;
0340 ;ACCEPT FILENAME #1 FROM EDITOR
0350 ;
0360      LDX #$00     ;EDITOR: IOCB #0
0370      LDA #$05     ;GET RECORD...
0380      STA ICCMD,X  ;COMMAND
0390      LDA #FNAME1/256 ;POINT...
0400      STA ICBAH,X  ;TO...
0410      LDA #FNAME1&255 ;FILENAME...
0420      STA ICBAL,X  ;BUFFER
0430      LDA #20      ;MAXIMUM NAME...
0440      STA ICBLL,X  ;= 20 CHARS
```

(Assembly listing continued on next page)

```
0450      LDA #0        ;(20 IN LO,
0460      STA ICBLH,X   ;0 IN HI)
0470      JSR CIOV      ;GET RECORD!
0480      BMI GETFN1    ;RETRY IF ERROR
0490 ;
0500 ;TRY OPENING FILE FOR INPUT
0510 ;
0520      LDX #$10      ;USE IOCB #1
0530      LDA #$03      ;SET UP...
0540      STA ICCMD,X   ;OPEN COMMAND
0550      LDA #FNAME1/256 ;POINT...
0560      STA ICBAH,X   ;TO...
0570      LDA #FNAME1&255 ;USER'S...
0580      STA ICBAL,X   ;FILENAME
0590      LDA #$04      ;OPEN FILE...
0600      STA ICAX1,X   ;FOR INPUT
0610      LDA #$00      ;AUX2...
0620      STA ICAX2,X   ;NOT USED
0630      JSR CIOV      ;OPEN IT!
0640      BPL GETFN2    ;OPENED OK!
0650      LDA #BADIN/256 ;UH-OH, PRINT
0660      LDY #BADIN&255 ;ERROR MESSAGE
0670      JSR PRINT     ;USING SUBROUTINE
0680      LDX #$10      ;BETTER...
0690      LDA #$0C      ;CLOSE...
0700      STA ICCMD,X   ;IOCB #1...
0710      JSR CIOV      ;TO PLAY IT SAFE
0720      JMP GETFN1    ;TRY AGAIN!
0730 ;
0740 ;PRINT FILENAME #2 PROMPT
0750 ;
0760 GETFN2
0770      LDA #PFILE2/256 ;HI PART IN A
0780      LDY #PFILE2&255 ;LO PART IN Y
0790      JSR PRINT     ;PRINT PROMPT!
0800 ;
0810 ;ACCEPT FILENAME #2 FROM EDITOR
0820 ;
0830      LDX #$00      ;EDITOR: IOCB #0
0840      LDA #$05      ;GET RECORD...
0850      STA ICCMD,X   ;COMMAND
0860      LDA #FNAME2/256 ;POINT...
0870      STA ICBAH,X   ;TO...
0880      LDA #FNAME2&255 ;FILENAME...
0890      STA ICBAL,X   ;BUFFER
0900      LDA #20       ;MAXIMUM NAME...
0910      STA ICBLL,X   ;= 20 CHARS
0920      LDA #0        ;(20 IN LO,
0930      STA ICBLH,X   ;0 IN HI)
0940      JSR CIOV      ;GET RECORD!
0950      BMI GETFN2    ;RETRY IF ERROR
0960      LDX #$20      ;USE IOCB #2
0970      LDA #$03      ;SET UP...
0980      STA ICCMD,X   ;OPEN COMMAND
0990      LDA #FNAME2/256 ;POINT...
1000      STA ICBAH,X   ;TO...
1010      LDA #FNAME2&255 ;USER'S...
1020      STA ICBAL,X   ;FILENAME
1030      LDA #$08      ;OPEN FILE...
1040      STA ICAX1,X   ;FOR OUTPUT
1050      LDA #$00      ;AUX2...
1060      STA ICAX2,X   ;NOT USED
1070      JSR CIOV      ;OPEN IT!
1080      BPL DOCOPY    ;OPENED OK!
1090      LDA #BADOUT/256 ;UH-OH, PRINT
1100      LDY #BADOUT&255 ;ERR MESSAGE
1110      JSR PRINT     ;USING SUBROUTINE
1120      LDX #$20      ;BETTER...
1130      LDA #$0C      ;CLOSE...
1140      STA ICCMD,X   ;IOCB #1...
1150      JSR CIOV      ;TO PLAY IT SAFE
1160      JMP GETFN2    ;TRY AGAIN!
1170 ;
1180 ;PRINT COPYING MESSAGE
1190 ;
1200 DOCOPY
1210      LDA #CPYMSG/256 ;POINT TO...
1220      LDY #CPYMSG&255 ;COPY MESSAGE
1230      JSR PRINT     ;PRINT IT!
1240 ;
1250 ;COPY THE FILE!
1260 ;
1270 COPYIT
1280      LDX #$10      ;IOCB #1
1290      LDA #$07      ;SET TO...
1300      STA ICCMD,X   ;GET CHARS
1310      LDA #0        ;READ 1 CHAR...
1320      STA ICBLL,X   ;AND PUT IN...
1330      STA ICBLH,X   ;ACCUMULATOR
1340      JSR CIOV      ;READ IT!
1350      BMI BADRD     ;READ ERROR!
1360 ;
1370 ;GOT A BYTE, WRITE IT!
1380 ;
1390      TAY           ;PUT CHAR IN Y
1400      LDX #$20      ;IOCB #2
1410      LDA #$0B      ;PUT CHARS...
1420      STA ICCMD,X   ;COMMAND
1430      LDA #0        ;TELL CIO TO...
1440      STA ICBLL,X   ;WRITE 1 BYTE...
1450      STA ICBLH,X   ;FROM ACCUMULATOR
1460      TYA           ;GET CHAR IN ACC.
1470      JSR CIOV      ;WRITE IT!
1480      BMI BADWRT    ;WRITE ERROR!
1490      JMP COPYIT    ;LOOP FOR MORE!
1500 ;
1510 ;CHECK ON ERROR CONDITION
1520 ;
1530 BADRD
1540      CPY #136      ;EOF?
1550      BNE NOTEOF    ;OTHER ERROR!
1560      JMP QUIT      ;AND QUIT!
1570 NOTEOF
1580      LDA #RDERR/256 ;GOT AN ERROR,
1590      LDY #RDERR&255 ;PRINT ERROR
1600      JSR PRINT     ;MESSAGE
1610      JMP QUIT      ;AND ABORT!
1620 BADWRT
1630      JSR WEPRNT    ;PRINT ERROR MSG
1640      JMP QUIT      ;AND QUIT!
1650 ;
1660 ;CLOSE FILES, END!
1670 ;
1680 QUIT
1690      LDX #$10      ;IOCB #1
1700      LDA #$0C      ;SET CLOSE...
1710      STA ICCMD,X   ;COMMAND
1720      JSR CIOV      ;CLOSE IT!
1730      LDX #$20      ;IOCB #2
1740      LDA #$0C      ;SET CLOSE...
1750      STA ICCMD,X   ;COMMAND
1760      JSR CIOV      ;CLOSE IT!
1770      BMI BADEND    ;BAD CLOSE!
1780      LDA #DONE/256 ;EOF, WE'RE...
1790      LDY #DONE&255 ;ALL DONE...
1800      JSR PRINT     ;PRINT MESSAGE
1810      BRK           ;AND EXIT
1820 BADEND
1830      JSR WEPRNT    ;PRINT ERR MSG
1840      BRK           ;AND EXIT!
1850 ;
1860 ;WRITE ERROR PRINT SUBROUTINE
1870 ;
1880 WEPRNT
1890      LDA #WRTERR/256 ;POINT TO...
1900      LDY #WRTERR&255 ;ERR MESSAGE
1910      JSR PRINT     ;PRINT IT
1920      RTS
1930 ;
1940 ;PRINT SUBROUTINE:
1950 ;
1960 ;INPUT:
```

(Assembly listing continued on page 102)

TYPESETTER
by Len Dorfman and Dennis Young
XLENT SOFTWARE
P.O. Box 5225, Dept. B
Springfield, VA 22510
(703) 644-8881
48K Disk $34.95

by Jonathan Buckheit

Typesetter by XLent Software is a new "printware" graphics program for your Atari 400, 600XL, 800, 800XL, 1200XL, or 130XE computer. As the term *printware* implies, **Typesetter** is not a tool to create graphics on your monitor; it's a graphics editor for your Epson, Gemini, Panasonic, NEC, or Prowriter printer (and printers compatible with these).

Typesetter is similar in purpose to the popular Broderbund program **The Print Shop**. However, unlike **The Print Shop**, **Typesetter** gives you full creative control over the printed page. There are no predefined page layouts, icons, and text fonts (although the program disk comes with several icons and fonts to get you started).

Typesetter can be used to create just about anything: letterheads, posters, title pages, signs, invitations and even a newsletter! That's right. On the last page of this review is a sample page from a newsletter created regularly by Ira Brickman using **Typesetter**, and it's quite impressive.

Though the highest resolution graphics mode on 8-bit Atari computers is 320x192, **Typesetter** squeezes 704x624 resolution to the printed page on 48K machines like the 800 and 800XL, and 768x640 (that's 491,520 separate pixels) on the 130XE. **Typesetter** is one of the few programs to take advantage of the 128K in the 130XE. A 130XE coupled with **Typesetter** can produce printouts that rival the $2500.00 Apple Macintosh. In fact, **Typesetter** gives you double the resolution. This high resolution is used to create pages that look almost professionally typeset.

Of course, you can't edit a 768x640 page all at once, even on the highest resolution Atari screen. **Typesetter** uses a special Atari programming technique known as fine scrolling to make your monitor's display a window on the printed page. Multiple graphics 6 screens are used.

You edit these pages in one color, since the aforementioned printers are monochrome devices (I'd really like to see a **Typesetter** for the new color printers. . .).

The only bug I've found in the program concerns scrolling. The screen address bytes of a load memory scan ANTIC instruction cannot cross a 4K boundary (i.e., $3FFF to $4000). **Typesetter** doesn't take this into account, and some lines are scrambled on the scrolling display. The lines print out normally, though, and you can edit them as you would other lines. The problem is purely cosmetic.

The **Typesetter** program is 100% machine language and comes on an autoboot protected disk. There are separate versions for the 130XE (130 version) and the 400/800 (65 version, so named for the ill-fated 65XE).

When you boot up **Typesetter**, a menu pops up, giving you several options: text editor, sketch pad, and print options. To give you the maximum resolution possible with limited memory, **Typesetter** keeps its text and graphics editors separate. These editors are fully integrated, so it isn't too onerous a restriction. The text editor portion scrolls over the main screen; you add text by simply typing.

On the sketch pad you design your own graphics icons. Once in this mode, you can switch to the main screen, then scroll around it and overlay the icon anywhere you want to put it on the main screen.

The main screen is kept resident in memory throughout, so you don't have to load and save to disk when you're going from text editor to the main menu to the sketch pad, and vice versa.

I'd like to be able to go directly from the text editor to graphics editor, without the wait for loading into the main menu (and, conversely, from the graphics editor to the text editor). Perhaps some options will be added for these functions in future releases.

The graphics editor is like many of the drawing programs for the Atari, plus its overlay options to add your icons to the main screen (you can also save the icon to disk). You can plot points, draw lines, circles and spheres, and do six pattern fills (these aren't of the seed-fill or the diamond-fill variety, so you may have to fill a complex object two or three times).

You can also invert the screen.. The program is joystick operated and very easy to use, since there's a menu line on top (similar to the desktop on the ST).

You use the powerful text editor to add text to your main screen. It will support thirty-two sizes of characters, and you

PRINTERS

EPSON

LX90	$234.95
HS80 Letterjet	318.95
JX80	448.95
FX185	454.95
LQ1500 (PAR)	949.95
FX85	331.95
LX80	210.95

STAR MICRONICS

SG10	$210.95
SG10C	234.95
SG15	368.95
SD10	321.95
SD15	440.95
SR10	468.95
SR15	581.95
Power type	299.95

CITIZEN

MSP10	$274.95
MSP15	438.95
MSP20	438.95
MSP25	548.95

PANASONIC

1091	$231.95
1092	368.95
1093	424.95
3151	424.95
3131	259.95

OKIDATA

Okimate 10	$170.95
Okimate 20	209.95
182	219.95
192	349.95
193	514.95
84	640.95

LEGEND

808/NLQ	$149.95
1080/NLQ	205.95
1380	259.95
13805	294.95

JUKI

6000	$189.95
6100	379.95
6300	769.95

TOSHIBA

P351	$1289.95
1340	599.95

DAISYWRITER

2000	$739.95

N.E.C.

2030	$699.95
2050	699.95
3530	1309.95
3550	1389.95

MODEMS

ANCHOR

Volkmodem 12	$179.95
Signalman Express	299.95
Lightning 2400	399.95
Westridge	59.95

DIGITAL DEVICES

300 Baud (Atari)	$98.95

HAYES

300	$137.95
300/1200	386.95
1200 B w/software	378.95
1200B	349.95

NOVATION

Smart Cat Plus (1200)	$309.95
Professional 2400	629.95

SUPRA

MPP1000E (Atari)	$54.95
MPP1064(C-64)	54.95

CAL-ABCO

Smart Team 1200	$199.95

C.D.I.

Mitey Mo (64)	$59.95

DISKETTES

BONUS	5¼	3½
SS/DD	$ 9.50	—
DS/DD	13.50	—

NO LABEL (with Pen & Flip-n-file)

SS/DD	$10.50	—
DS/DD	14.50	—

MAXELL

MD1	$15.95	32.95
MD2	19.95	38.95

MEMOREX

SS/DD	$12.50	29.95
DS/DD	16.50	34.95
FF30/20SS/DD	—	59.95
FF50/20SS/DD	36.95	—
FF50/20DS/DD	42.95	—

All Diskettes Carry a Lifetime Warranty

MONITORS

PRINCETON

Max 12E	$189.95
HX-9	459.95
HX-9E	519.95
HX-12	469.95
HX-12E	555.95
SR-12	589.95
SR-12E	649.95

NEC

1201 (Green)	$99.95
1205 (Amber)	99.95
1280 (TTL Green)	148.95
1285 (TTL Ameber)	148.95
1460	269.95
1401 (R.G.B.)	659.95

ZENITH

ZVM 122/123	$ 74.95
ZVM 124 (TTL Amber)	149.95
ZVM 131 (RGB)	279.95
ZVM 135 (RGB/Color)	459.95

TEKNIKA

MJ-10	$178.95
MJ-22	254.95

AMDEK

300G	$116.95
300A	126.95
310A	144.95
Color 300	175.95
Color 500	339.95
Color 600	398.95
Color 700	468.95
Color 710	538.95
Color 722	579.95

INTERFACES

INDUS

GT (Atari)	$198.95
GT (Commodore)	218.95

TANDON

TM100-2	$109.95

TEAC

55B	$104.95

COMPUTERS

ATARI

520 ST/RGB	CALL
520 ST/Mono	CALL
800XL	$ 79.95
130XE	139.95
1050 Drive	144.95
Happy 1050 Drive	299.95
Happy Enhancer	139.95
1030 Modem	59.95

DISK DRIVES

SUPRA

MTP1150	$45.95

DIGITAL DEVICES

U Print/Port	$49.95
U Print/16K	69.95
U Print/64K	79.95

CARDCO

G-Wiz	$48.95
AT-1 (Atari)	48.95

TYMAC

Epson	$58.95
PAR (Centronics Std.)	54.95

SOFTWARE

CONTINENTAL

Home Accountant	$27.95
Tax Advantage	27.95

BRODERBUND

Printshop	$27.50
Graphics Library I	17.50
Graphics Library II	17.50

SYNAPSE

Synfile	$31.95
Syncalc	31.95
Syntrend	27.95

BATTERIES INCLUDED

Paper Clip	$36.95
Homepak	31.95
Paper Clip w/spell	57.95

ORDER TOLL FREE

WESTERN REGION 1-800-351-3455

EASTERN REGION 1-800-351-3442

POLICY

No deposit on C.O.D. orders. Free freight on all prepaid cash orders over $300 in the Continental U.S.A. APO and FPO orders add $5.00 per hundred. For Priority Mail add $10.00 per hundred. Free shipping for PA residents. PA residents add 6% sales tax. All defective products must have a prior RA. number.

"Where Prices are Born, Not Raised."

WHITE HOUSE COMPUTER

WESTERN REGION

11327 Trade Center Drive
Suite 335
Rancho Cordova, CA 95670
Customer Service: 916-635-3455
Call Toll Free: 1-800-351-3422

Hours: Monday thru Friday 9 a.m. - 6 p.m.

EASTERN REGION

P.O. Box 4025
Williamsport, PA 17701
Customer Service: PA residents 717-322-7700

MASTER CARD 4% **VISA 4%** **AMERICAN EXPRESS 5%**

CIRCLE #114 ON READER SERVICE CARD

IRA

A Tax-Exempt Savings Evaluation

by David L. Clark

Do you currently have an Individual Retirement Account (I.R.A.), Keogh Account, or other tax-exempt savings program? If not, have you considered opening one? If you do, have you compared the expected value of your tax-exempt account at retirement with the expected future value of another type of savings program?

This user-friendly program (we'll call it **IRA**) helps you evaluate the potential benefits of an I.R.A. or other tax-exempt savings program. When run, **IRA** shows the future value at retirement that a regular savings program would have, the future value of the tax-exempt savings program, and the future value of the tax-exempt savings program after withdrawal at a lower tax rate. It also shows the total cash paid in, the purchasing power in today's dollars of the savings plans, and the amount by which spendable income will be reduced if the cash is deposited annually in an I.R.A.

Required inputs.

When you run the program after typing it in and saving it, **IRA** will ask for five inputs. These are:

(1) Number of years to retirement — Just enter the number of years until you plan to retire. For example, if you're now 32 and you plan on retiring at 65, enter *33* (=65-32).

(2) Compounded annual interest rate — The interest rate should be entered as the compounded annual interest rate. Other rates need to be converted to this one.

When entering the interest rate, remember to enter it as a decimal value. For example, you would enter 0.1025 if the interest rate for a certificate of deposit for one year, or for an I.R.A., was quoted at 10.25% compounded annually.

If you don't have a tax-exempt savings plan now, the current rate for a certificate of deposit is normally the best for comparison.

(3) Annual deposit amount — Enter the amount that you'll be adding to your account or investment each year. For example, if you have $1,800.00 to save each year, enter *1800* (**IRA** ignores commas).

(4) Current tax rate — Enter your current marginal tax rate, or the amount of additional income that would be paid as income taxes. For example, if you were to make an additional $1,000 and $350 would go to taxes, your marginal tax rate would be 350/1000 or 35%.

The tax rate should be entered as a decimal. For example, if the tax rate is 35%, enter *0.35*.

If you don't know or cannot calculate your marginal tax rate, simply divide the amount you pay in taxes by your total income. This will give your total tax rate, which is lower and not as accurate as the marginal rate, but will still give a good indication of the value of a tax-exempt savings plan.

IRA *continued*

```
740 PK=PEEK(BTT):IF PK/A3<>INT(PK/A3)
OR PK=A0 THEN 740
750 IF PK=A3 THEN ? CHR$(126);:POS=POS
+(POS<=21)-A2*(POS>21):GOTO 730
760 IF POS=21 THEN GOTO ABB
770 POSITION A2,19:IF POS>21 THEN PC=C
1:? ERR$:? :? "NOW PRINTING":PRT$="P:"
:GOSUB 400:GOSUB PING:GOTO ABB*A2
780 ? ERR$(A1,A4);CHR$(157);CHR$(157):
? "CHANGE WHICH VALUE?":POS=A4
790 POSITION A2,POS:GOSUB PAUS
800 PK=PEEK(BTT):IF PK/A3<>INT(PK/A3)
OR PK=A0 THEN 800
810 IF PK=A3 THEN ? CHR$(126);:POS=POS
+(POS<=7)-A4*(POS>7):GOTO 790
820 GOTO 150
900 REM INITIALIZE VALUES
910 G51=500:G52=560:G53=580:G54=590:A0
=0:A1=1:A2=2:A3=3:A4=4:A5=5:AB=10:AC=2
0:AD=30:ABB=100:TR1=290:C1=177:C2=114
920 C3=65:BKC=710:CBA=510:GSINPUT=600:
BTT=53279:PING=650:PAUS=660:TOOBG=4000
0
930 DIM B$(20),NY$(26),IN$(27),PMT$(21
),YT$(8),DEC$(14),EN$(10),ER$(2),TT1$(
28),TT2$(33),PRT$(2),BL$(9),L1$(36)
940 DIM ARR$(2),ERR$(6),INFL$(21),PP$(
16),FV$(5):ARR$=CHR$(27):ARR$(A2)=CHR$
(31):ER$=CHR$(28):ER$(A2)=CHR$(156)
950 NY$="NO. OF YEARS TO RETIREMENT":I
N$="COMPOUNDED ANNUAL INT. RATE":PMT$=
"ANNUAL DEPOSIT AMOUNT"
960 YT$="TAX RATE":DEC$="(AS A DECIMAL
)":EN$="ENTER THE ":TT1$=" TO I.R.A. O
R NOT TO I.R.A. "
970 TT2$=" A TAX-EXEMPT SAVINGS EVALUA
TION ":BL$="                ":L1$="******
**********************************"
980 INFL$="ANNUAL INFLATION RATE":PP$=
"PURCHASING POWER":FV$=" FV- "
990 FOR I=A1 TO 6:ERR$(I)=CHR$(156):NE
XT I:RETURN
```

●

CHECKSUM DATA.

(see page 10)

```
10 DATA 267,215,465,758,93,277,764,843
,95,67,107,150,873,666,954,6594
290 DATA 902,81,741,551,452,803,891,28
3,61,308,345,911,580,804,483,8196
540 DATA 958,234,678,283,280,747,984,2
28,552,312,992,987,96,993,662,8986
720 DATA 96,330,81,741,659,280,224,347
,70,676,718,863,226,731,888,6930
940 DATA 649,575,317,165,59,253,2018
```

●

TEMPLE OF APSHAI TRILOGY
EPYX
1043 Kiel Court
Sunnyvale, CA 94089
48K Disk $39.95

by Patrick J. Kelley

It is dark in the ancient temple. Outside of your own rasping breath in your ears, the only sound is the faint rush of wind down a nearby chamber. Even though you've been through many such adventures, you cannot help but feel a tinge of apprehension. Your intuition, honed to a fine edge, tells you that something is around the next bend. In this near darkness, your eyes are of little help to you, so you must rely on your finer senses.

Adjusting the grip of your small shield, you edge forward. The hefty shortsword feels good now, its leather wrapping clenched in your sweaty palm. With a slight mental shrug, you curse yourself for not purchasing plate armor; the chain mail covering your body will not block the sting of poison barbs or claws. As you edge forward, step by slow step, you hear the rustling of the unseen something. Here it comes — confrontation.

Steeling your nerves, you whirl with the sword, ready to slash out at the now-visible form. Suddenly, out of the shadows, something lunges. It is, amazingly, a man-sized ant! No time for your amazement now; you raise your shield against the antman's pike. The battle for the ***Temple of Apshai*** *has begun.*

If the introduction to this review is a bit wordy, you must pardon me. However, as any of you who have ever played fantasy games know, words are the key to a successful game—or, if not words themselves, then the images thus conjured up. In this arena of the mind, the more elaborate the image, the better.

In the Epyx compilation of the Apshai games, **Temple of Apshai Trilogy**, these images flow like water from a wellspring. This latest adventure takes the best of three games, ties them together with a central theme and lets you loose into the universe of Apshai.

In the first, you—the hardy freebooter—must venture into the crumbling **Temple of Apshai** in search of gems and gold. During your exploration of the ruins, you must evade various traps, monsters, ghouls and thingamabobs, to survive and emerge victorious.

Still greedy? Don your armor again and explore the **Upper Reaches of Apshai**. This is where the fun really starts, and the dangers you must face increase. In this round, you're wandering about the deserted citadel of the Apshaians, again looking for things to wear, eat and bank.

If you make it through these funfests, the **Curse of Ra** awaits you. Oooh! Wanna hint? **Curse of Ra** takes you into the desert this time, wandering the parched sands for the ultimate goal of financial fulfillment (or, for the less mercenary among us), the satisfaction of solving the **Curse of Ra**.

If all of this sounds like your goblet of ale, come on in. The adventure of a lifetime awaits.

Beginnings and other shenanigans.

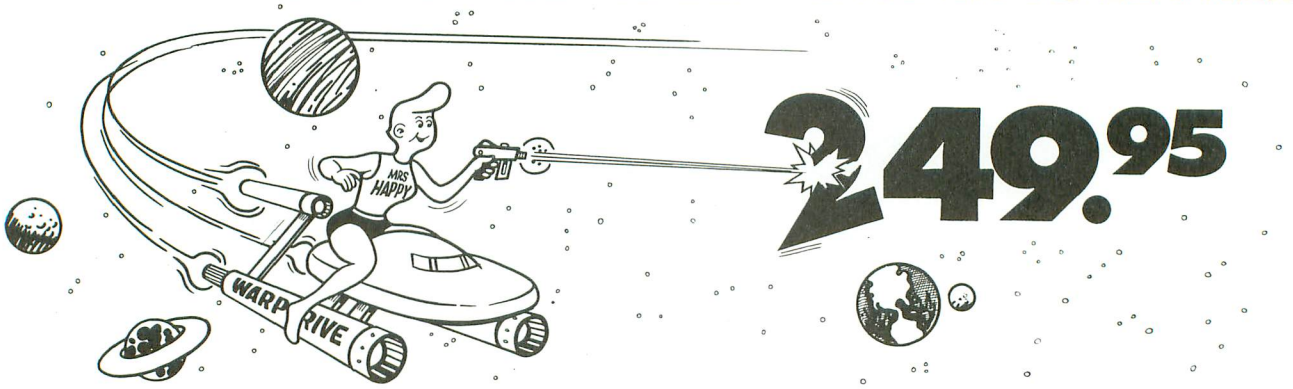
You start your career as an adventurer in the inn, choosing your character's name, strength, armory, healing prowess and intelligence. You can let the computer set your character for you, and save it to disk.

About this time, you'll be called upon to make decisions. First, you choose the attributes of your character: intelligence levels, cunning, and so forth. You can also set such particulars as ego, which may aid your character in later scrapes. You'll be called upon to choose your weapons and their size. Human nature will make you choose the biggest sword, bow, etc., but this will be no help to your energy levels later—fighting will be inhibited by using an unwieldy weapon or shield.

Your friendly author set forth into the dank reaches of Apshai as the fearless Rolf Weimann, teutonic freebooter and roustabout. Armed to the hilt with arrows, sword and such, Rolf and I set off to bash some monsters. What we learned in our time together was that adventuring can be fun, and that monsters can

ATARI DISK DRIVE OWNERS . . .

HAPPY BLASTS RETAIL PRICE—ORDER TOLL FREE!



THE FAMOUS HAPPY ENHANCEMENT NOW ONLY $149.95
for 1050 order number HC1G, for 810 order number HC8G

Makes your disk drive read and write faster, and allows you to execute the HAPPY WARP SPEED SOFTWARE. Available only for ATARI 1050 and 810 disk drives. 1050 version allows true double density plus the original single and enhanced density. PRICE INCLUDES WARP SPEED SOFTWARE BELOW, installation required.

HAPPY WARP SPEED SOFTWARE REV 7 (not sold separately)

Includes the famous HAPPY BACKUP and COMPACTOR which are the most powerful disk backup utilities available for your ATARI computer, plus MULTI DRIVE which allows high speed simultaneous reading and writing with up to 4 HAPPY ENHANCED drives, plus SECTOR COPIER which is the fastest disk copier that supports the 130XE RAMDISK, plus the WARP SPEED DOS which improves ATARI DOS 2.0s to allow fastest speed, plus HAPPY'S DIAGNOSTIC which allows comprehensive disk drive testing.

HAPPY 1050 CONTROLLER $64.95 order number HC2G

For use with HAPPY ENHANCED 1050 disk drives only. Allows easy access to HAPPY 1050 slow and fast speeds and ultimate control of disk drive write protect, including writing to disk back side and protecting valuable data disks. Printed circuit board has switches and write protect indicator LED, installation required.

GET YOUR FAVORITE HIGH SPEED DOUBLE DENSITY DOS

Both of these disk operating systems support the fastest speed with both HAPPY 810* and 1050, and with HAPPY 1050 you get true double density. WARP SPEED DOS XL is HAPPY's own version of OSS DOS XL, and includes under cartridge, under ROM and AXLON RAM disk version, and is order number HC4G at $29.95. TOP DOS version 1.5 from ECLIPSE SOFTWARE has more menu driven features, operates in all three densities, supports the 130XE RAMDISK, and is order number HC6G at $39.95. *Note: 810 requires upgrade below.

810 VERSION 7 UPGRADE $49.95 order number HU3G -XXXX

Allows older 810 HAPPIES to use newer software. Includes custom plug in IC and rev 7 WARP SPEED SOFTWARE. Same price for all HAPPY 810s registered or not. When ordering replace XXXX in part number with the serial number of your HAPPY COMPUTERS manufactured 810 board, or with a 2732 or 2532 which corresponds to the EPROM part number in your HAPPY 810 socket A102 of your side board modified HAPPY (not made by HAPPY COMPUTERS), installation required. Upgrade not needed for new 810 HAPPYS and serial number over 8000.

SUPER PACKAGE SPECIALS

Get a HAPPY 1050 ENHANCEMENT and CONTROLLER and WARP SPEED DOS XL for just $199.95 order number HS5G, or get the same with TOP DOS 1.5 instead of DOS XL for just $214.95 order number HS7G. If you already have the 1050 ENHANCEMENT you can get the HAPPY 1050 CONTROLLER and WARP SPEED DOS XL for $74.95 order number HXL9G, or get the HAPPY 1050 CONTROLLER and TOP DOS 1.5 for just $84.95 order number HTD9G. For other specials and dealer pricing call (408) 779-3830.

All prices include UPS shipping in USA, add $10.00 for shipment outside USA. California residents add sales tax. No extra charge for credit cards or COD, VISA or MASTERCARD accepted. Our toll free number is an order taking service, not our line. To ORDER ONLY call (800) 538-8157 outside California, or (800) 672-3470 inside California, ask for extension 817 and have your credit card, part number and quantities ready. Toll free hours 6 am to 12 pm Mon.–Fri., 8 am to 8 pm Sat. & Sun., Pacific Time. For answers to questions call HAPPY COMPUTERS at our number below. Office hours 9–5 Mon.–Fri. Pacific Time.

HAPPY COMPUTERS, INC. * P.O. Box 1268 * Morgan Hill, CA 95037 * (408) 779-3830

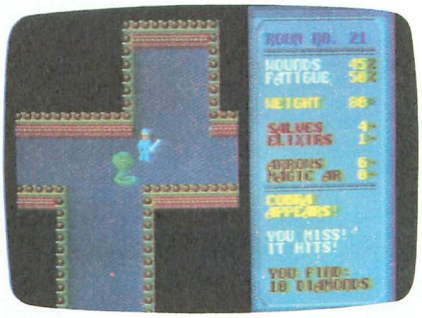
CIRCLE #117 ON READER SERVICE CARD

bash back sometimes. (I won't spoil the game for you by giving you a blow-by-blow description of the various things you'll see and do, but a few words to the wise will help.)

Proceed cautiously through the many rooms of each level. The variety of rooms is staggering. In your travels, you will venture into dank caves, smooth hallways, stagnant monster lairs and catacombs. Computer-generated beasties are counting on your blundering around, and will jump on your frame at the drop of a helmet.

Also, keep an eye on the fatigue and health readings determined by the computer. Ignoring these can be fatal, especially if you stumble across a howling, pike-swinging Antman. As in real adventure gaming, you have hit points and such, so know your strengths and guard well your weaknesses.

Keep your treasure hoarding to a minimum, making frequent trips back to the inn to deposit your finds. While you're at it, watch out for traps. Whoever programmed this game must have taken lessons from de Sade, because the traps that befall the unwary are grisly: flame trap (roasted alive), ceiling trap (splat), crossbow trap (sproing) and, my personal favorite, creeping crud (yeah).



Temple of Apshai.

If these don't do you in, the various monsters (such as Giant Ticks, Ghouls, Jackals, Scorpions, the dreaded Antman, and the Gryphon) will.

If all of this gives you the idea that I had a ball playing this game, you're right. I always have been a sucker for any game that lets you shoot arrows into people or things, so this was right up my twisted little alley. Rolf and I are good friends now, seasoned in the fire of monster combat.

So it is, too, with the **Temple of Apshai Trilogy** and me. Fast friends and good ones. So, if you're the armchair-barbarian type, come on in. You have nothing to lose but. . .

32K Cassette or Disk

The Clash of Kings

by Bryan Schappel and Barry Kolbe

Welcome, one and all. Welcome to the 48th Intergalactic **Clash of Kings**. Today, we have with us the champions, humanoids from the planet Zorn, of the Ribos system. Their worthy opponents have just arrived from planet Mechtron, of the Aldeberon system. (Mechtron is inhabited only by a race of robots.)

As is well known, the challengers have their choice in the form of combat to be used. This year, the game will be Terri-wars (terrain wars, for those with no imagination). The champions will return to their home planet with the Trophy of Kings.

I can see in the distance that the teams (each team is composed of nine players) are taking their positions on the field now.

Allow me to quickly explain the rules of Terri-wars, for

those of you who are attending for the first time. Turn up your interpreters and listen. The goal is to destroy the opponent's king. Simple?

As the game begins, players are positioned on a maplike field. The players move alternately, one square at a time. Diagonal moves are not permitted, and if an illegal move is attempted the team is alerted either by a tone or a verbal explanation. Combat is initiated by moving on an opponent. The two players involved are then whisked away to their Heli-Battle Globes (HBGs). At this point, the viewing monitor shifts to the Terri-wars battlefield.

The HBGs are equipped with Plasmaton Disruptors (also known as highly high-tech, extremely dangerous, very nasty Zappo-guns). These are traditionally used to ionize your opponent in some way. But this year, when an HBG is hit, the instant before impact a supremely intelligent super-computer teleports the lucky occupant to safety, or so the theory goes. You see, this year the teleportation systems were provided by Defectron, whose slogan is: "If it doesn't work it needed fixing, anyway."

Each player begins with a certain number of strength points, displayed as a bar graph at the bottom of your viewing screen. Each hit an HBG receives reduces the player's strength by one. When a player's strength reaches zero, that person is removed from play. (This is actually quite true. The teleportation systems have one minor flaw: each time someone is teleported, some body mass is lost, and the player's strength is decreased. When the points reach zero, the remainder of the player is collected and transported to the hospital for the severely impaired.)

The players, board positions, names and strength points are given below.

Board positions.

5										E
6	2								B	F
7	3	1						A	C	G
8	4								D	H
9										I

Humanoid — Planet: Zorn.

Player #	Name	Points
1	King	9
2, 4	Vipers	7
3	Canine	4
6, 8	Bipeds	3
5, 7, 9	Scorpions	5

Robots — Planet: Mechtron.

Player #	Name	Points
A	Cyborg	9
C, E	Serv Droids	7
B	Land Rover	4
F, H	Warbots	3
D, G, I	Avions	5

This year, to make Terri-wars more challenging, the battlefield has a randomly changing maze of pylon stars. Players aren't allowed to move through these, but bumping into them doesn't damage the battle globe.

The field also has randomly generated terrain features at the beginning of each game. There's at least one path through the terrain composed of rocks, trees and mountains. Players aren't allowed to move over these terrain features.

Notes on the program.

The impressive introduction screen for **Clash of Kings** was achieved by a Display List Interrupt (DLI) and a little fine vertical scrolling. The DLI will shade any object that's 16 scan lines high (equal to a graphics 2 character in height) in all the shades of one color. This makes the text look as if it were cut from a polished metal bar.

The terrain board border was accomplished by using a character that was defined with every other bit turned on. This technique was also used to create the multi-colored terrain features. The players' pieces are composed of four redefined characters, forming a 2x2 grid. The cursor was done with player 0 in double width.

The battlefield was done in graphics mode 2, and the HBGs are players 0 and 1, while the missiles are players 2 and 3. The battlefield portion runs almost completely in the vertical blank. Explosion sound effects were borrowed from Tom Hudson and Kyle Peacock's game, **Fire Bug** (issue 23).

Typing it in.

Listing 1 is the BASIC data and data checking routines. This listing is used to create both cassette and disk versions of **Clash of Kings**. The data statements are listed in hexadecimal (base 16), so the program will fit in 16K systems.

The assembly language source code for the game was written with MAC/65 from OSS. Those readers interested in how the game works can obtain the listing on the disk version or on the **ANALOG Computing** TCS.

Cassette instructions.

Type Listing 1 into your computer, using the BA-

SIC cartridge, and verify your typing with **Unicheck** (see page 10).

Type *RUN* and press RETURN. The program will ask:

MAKE CASSETTE (0), OR DISK (1)?

Type *0* and press RETURN. The program will begin checking the data statements, printing the line numbers as it goes. It will alert you to any problems. Fix any incorrect lines and re-RUN the program until all errors are eliminated.

When all your data lines are correct, the computer will beep twice and prompt you to *READY CASSETTE AND PRESS RETURN*. Now, insert a blank cassette in your recorder, press the RECORD and PLAY buttons simultaneously, and hit RETURN. The message *WRITING FILE* will appear, and the program will create a machine language boot tape version of **Clash of Kings**, printing each data line number as it goes. When the *READY* prompt appears, the game is recorded and ready to play. CSAVE the BASIC program on a separate tape before continuing.

To play the game, rewind the tape created with the BASIC program to the beginning. Turn the computer off and remove all cartridges. Press the PLAY button on your recorder and turn on your computer, while holding down the START key. If you have an XL or XE series computer, you must hold the START and OPTION keys when you turn on the power. The computer will beep once. Hit RETURN, and **Clash of Kings** will load and run automatically.

Disk instructions.

Type Listing 1 into your computer, using the BASIC cartridge, and verify your typing with **Unicheck**. Type *RUN* and press RETURN. The program will ask:

MAKE CASSETTE (0), OR DISK (1)?

Type *1* and press RETURN. The program will begin checking the data lines, printing each line number as it goes. It will alert you to any problems. Fix incorrect lines and re-RUN the program, if necessary, until all errors are eliminated.

When all data lines are correct, you will be asked to *INSERT DISK WITH DOS, PRESS RETURN*. Put a disk containing DOS Version 2 in drive 1 and press RETURN. The message *WRITING FILE* will appear, and the program will create an AUTORUN.SYS file on the disk, displaying the line numbers as it goes. When the *READY* prompt appears, the game is ready to play. Be sure to SAVE the BASIC program before continuing.

To play the game, insert the disk containing the AUTORUN.SYS file in drive 1. Turn off your computer, remove all cartridges, and turn the computer back on. **Clash of Kings** will load and run automatically.

Playing the game.

The Clash of Kings is a two-player game that requires two joysticks plugged into ports 1 and 2. The joystick in port 1 controls the player on the left, and stick 2 controls the player on the right. To move a player on the terrain board, move the cursor over the player you want and press the fire button. A short tone is given when a player is chosen.

Once you choose a player, it *must* be moved; illegal moves are not permitted, so choose with caution and skill. If you choose a player which can't be moved, you'll be told, and you may choose another piece. Move the cursor in one of the four cardinal directions and press the fire button again to make the choice final.

The cursor is on a rubber band. If you try to move it more than once in one direction, it snaps back to its original position for another choice. **Clash of Kings** will wait one-half second before snapping back the cursor. During this time, you may press the trigger to select a piece destination. Move onto a square held by your opponent to do battle. You're not permitted to move onto your own pieces.

On the battlefield, control the Battle Globes by moving your stick in the four cardinal directions. To fire a Plasmaton Disruptor, press the stick button and move the stick in the direction you want to fire. You can only fire one shot at a time, and you can't fire again until your Disruptor hits an object.

Pressing RESET at any time will return you to the title screen, ready to play again.

Well, that's all. We hope you enjoy winning the Trophy of Kings and taking it back to your home planet. Now, let the games begin.

Barry Kolbe is a mathematics teacher in Madison, WI. He uses the Atari to demonstrate graphing in his classroom. His former student Bryan Schappel studies Computer Science at the University of Wisconsin. This is their first major project as a team.

Listing 1.
BASIC listing.

```
10 REM *** CLASH ***
20 TRAP 20:? "MAKE CASSETTE (0), OR DI
SK (1)";:INPUT DSK:IF DSK>1 THEN 20
30 TRAP 40000:DATA 0,1,2,3,4,5,6,7,8,9
,0,0,0,0,0,0,0,10,11,12,13,14,15
40 DIM DAT$(91),HEX(22):FOR X=0 TO 22:
READ N:HEX(X)=N:NEXT X:LINE=990:RESTOR
E 1000:TRAP 120:? "CHECKING DATA"
```

```
50 LINE=LINE+10:? "LINE:";LINE:READ DA
T$:IF LEN(DAT$)<>90 THEN 220
60 DATLIN=PEEK(183)+PEEK(184)*256:IF D
ATLIN<>LINE THEN ? "LINE ";LINE;" MISS
ING!":END
70 FOR X=1 TO 89 STEP 2:D1=ASC(DAT$(X,
X))-48:D2=ASC(DAT$(X+1,X+1))-48:BYTE=H
EX(D1)*16+HEX(D2)
80 IF PASS=2 THEN PUT #1,BYTE:NEXT X:R
EAD CHKSUM:GOTO 50
90 TOTAL=TOTAL+BYTE:IF TOTAL>999 THEN
TOTAL=TOTAL-1000
100 NEXT X:READ CHKSUM:IF TOTAL=CHKSUM
 THEN 50
110 GOTO 220
120 IF PEEK(195)<>6 THEN 220
130 IF PASS=0 THEN 170
140 IF  NOT DSK THEN 160
150 PUT #1,224:PUT #1,2:PUT #1,225:PUT
 #1,2:PUT #1,236:PUT #1,42:CLOSE #1:EN
D
160 FOR X=1 TO 71:PUT #1,0:NEXT X:CLOS
E #1:END
170 IF  NOT DSK THEN 200
180 ? "INSERT DISK WITH DOS, PRESS RET
URN";:DIM IN$(1):INPUT IN$:OPEN #1,8,0
,"D:AUTORUN.SYS"
190 PUT #1,255:PUT #1,255:PUT #1,0:PUT
 #1,32:PUT #1,144:PUT #1,52:GOTO 210
200 ? "READY CASSETTE AND PRESS RETURN
";:OPEN #1,8,0,"C:"
210 ? :? "WRITING FILE":PASS=2:LINE=99
0:RESTORE 1000:TRAP 120:GOTO 50
220 ? "BAD DATA: LINE ";LINE:END
230 DATA 0,42,216,31,255,31,169,0,141,
47,2,169,60,141,2,211,169,0,141,231,2,
133,14,169,56,141,232,2
240 DATA 133,15,169,0,133,10,169,32,13
3,11,24,96
1000 DATA 00000000000000000DBDB18E7E718
DBDB0000000707FFFF00084A34DC3B2C5210FF
7E3C187E183CFF0000000000,395
1010 DATA 00000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000,395
1020 DATA 00000000000000000000001C1C38
00000000000000000000003C3C00000000
000000000000FE66EE76667E,567
1030 DATA 000018787818187E00007C664C18
367E00007ECE180C663E00000C1C3C6CFE0C00
00FE60FC06667C00007CE07C,823
1040 DATA E6667E0000FE660C18303000003C
663CE666FE0000FE66FE060C380000003C3C00
3C3C000000000000000000006,507
1050 DATA 0C0C3C0C0C0600000000000000000
006030303C303060000000000000000000000
00000000000000F83C666667E,703
1060 DATA E70000FC66FC66667C0000FE6660
F0667E0000F86C66E66C7C0000FE607CE0667E
0000FE63FC6C606000000FE66,895
1070 DATA 60EE667E000076367E3636760000
7E183C18187E00000E0606F6663C0000E66EF8
786E660000F060E060767E00,619
1080 DATA 00E377FF6B6377000066767E7E6E
760000FE66E666667E0000FC66EE7C60700000
FE66E6666C760000FC66E67C,707
1090 DATA 6C760000FE607E06667E00007ED8
1818181C0000E666E6667E7C0000E666E6667E
180000E363EB7F77630000E6,949
1100 DATA 663C3C66760000E6663C18383800
00FE6C1830667E0000010319 3F3D1F00000000
0000000000080C098FCBCF8,925
1110 DATA 0000000003070C18301830600C080
000000003030307070700100E0E0E0C0CCCCF0
0000001F707070700000000,529
1120 DATA F000F8F8000000000000007F0000
7E66667E1CCC00000C0F0F070301000018F8F8
F0E0C0000000007060607000,725
1130 DATA 00000000F8F8F800000000010307
7F00000000C0E0F0F0000000001F03637F0000
0000C00606FE000000000307,231
1140 DATA 0707000000000008080980018 1F0F
0F0307670030F0E0C08080D8000007010103070
70000F0C0C0E0F0F0AAAAAA,891
1150 DATA AAAAAAAAAA000F030606070E0000
86C76767E67700006F66E6EEE667000080C060
60C0C0000000000020A0A28,652
1160 DATA 0000000080200A0880000000010505
1515000000004040505000000307 0A0A2A2A00
0000848E8AEAAA0000000000,379
1170 DATA 0000000101010707060700C0C0F0
F030303C007F7F303038000000C0C0C0C0E000
0000606C6C6C6F607F00CCCC,989
1180 DATA CCCCCC0CFC001B0F1B030F0F0C0F
E6FCE6E0F030303C717F7F3F30303800C0F0F0
C0C0C0E00007030F30361F00,538
1190 DATA 00F0E0F80E6EF8000007F7F63030F
000000E080000000000000007FC7070F0C1C3C00
F09880C0E060E0007F63070F,297
1200 DATA 0C0C3C00F0D8C0C0E060E0000303
0F3F3F0F0600E0E0F87E7EF830005555555555
5555550000000000000000000,857
1210 DATA 00000000000000000000000000000
00000000000000000000022280A020000000A888
208000000000020202020205,422
1220 DATA 0000808080808080400000AAAAAA00
00000000AAAAAA0000000000000000000000000
00707070425724F070704200,201
1230 DATA 1002020202020202020202827002
70467F2470469324410024707070704 2A62470
70F0C7C624077070F0E20010,618
1240 DATA 2222222222222202412424707070
4700100707070707070707074143240000
34282500232C213328002F26,80
1250 DATA 002B292E27330000001C231E0021
2E212C2F27001119181500000000000000706C
617965725A0069000000000,570
1260 DATA 0000006D656E5A40590000000006D
656E5A4059008080080A1AEA1ACAFA780A3AFAD
80B5B4A9AEA780B0B2A5B3A5,767
1270 DATA AEB4B39A8080080C0C0C0F4E8E5FE
FFE3ECE1F3E8C0C0C0C0FBFDC0EFE6C0EBE9EE
E7F3C0FBFDC0000000004041,113
1280 DATA 6061404160614243626342436263
4243626344456465444564654849686946476 6
674A4B6A6B4A4B6A6B4C4D6C,979
1290 DATA 6D4C4D6C6D4C4D6C6D4E4F6E6F4E
4F6E6F525372735051707159 5A797A5B5C7B7C
5D5E7D7E0003808080808080,425
1300 DATA 8080808000C0000010680808080 80
8080800F0A0000040809080808080808012110D
0000020780808080808080,875
1310 DATA 100B0000058080808080808080800
800E003332242322302F2E3B3C3D3E30313233
2E2F30313F40414200FC8484,952
1320 DATA 8484848484848484848484848484
FC00FF01F20E050605066629812 99C29BB2981
2966 29BB299C29FFFFFFFF03,85
1330 DATA 0305050507070409030305050507
070409C60F020000212C21330C00392F350023
212E2E2F34002D2F36258E30,355
1340 DATA 2C2139253200000C00392F350037
292E00342825000 48E00000E26322F2D1AFF00
000580A2A2AB8000252E3425,565
1350 DATA 3230322933253300111918 15FF00
00000A233221263425240000223 91AFF00000A
2221323239000 02B2F2C2225,337
1360 DATA FF000E555657 58FF0009223239 21
2E003323282130302 52CFF0000000003404146
4700003032253333 0080B3B4,24
1370 DATA A1B2B4800000505 14A4BFF036061
6667000000000342F00222527292E0000000000
70716A6BFE08111A08E67446,100
1380 DATA 0000005600000C1A296888142261
52860A6418000000000000F1030E8081224D8
A24C3000000000000000000,881
```

```
1390 DATA 9E6A1D649EE34000000000000000
00004020100916E8E70000000000000000000
00000000000000FDF9F5F1ED,554
1400 DATA EAE6E2DEDAD6D3CFCBC7C3BFBCB8
B4B0ACA8A5A18A86827E7B77736F6B6764605C
5854504D4945413936322E2A,777
1410 DATA 26221F1B17130F0B0804978B8074
685D51463A2E23170C00834383C3070B0E0D01
FFEC14161700008C786E6478,759
1420 DATA 645A14152728 3B3C4F5063645176
77788B8C9FA018FF18242442 91A1B5A3D74E3C
187E18003C187F183C42FFFF,71
1430 DATA 81A5A581FF7E1818181824244291
A1B5A3D74E3C18FF18007F18FE183C42FFFF81
939381FF7E18187E18242442,914
1440 DATA 89919D93D74E3C18181800FE183C
183C42FFFF81C9C981FF7E1800246618186624
00003C425A5A423C00000000,744
1450 DATA 00000000000000000000000000A0
00989900149900159900169900 17C8D0F160A2
03A0008485A9108586989185,971
1460 DATA C8D0FBE686CA10F660A027A90099
E3118810FA60858B848C201F28C8B18B100229
7F99E811B18B3003C8D0F0A0,743
1470 DATA 772007298810FA60A205204228CA
D0FA60A5921869508D9A24A5931869508DA424
60A900854D4C5FE4A919D002,492
1480 DATA A9648D00D2A9AA8D01D2208C28A9
008D00D28D02D28D01D28D03D260A003200729
8810FA60A2008EC8028EC502,279
1490 DATA A20E8EC502A2508E0002A22E8E01
02A2C08E0ED4A23E8E2F0260A2008E2F024C07
29A2008EC2028EC302E88E6F,569
1500 DATA 028E08D020F627A9FC8DC002A910
8D07D4A9038D0AD08D0BD08D1DD0A9FFA03E99
0016990017C8C0A0D0F5A92C,31
1510 DATA 8D02D0A9B48D03D060A58D490185
8D60A9008514A514F0FC60858E3018F018C913
B014A58DD008A58EC90AB008,275
1520 DATA 9008A58EC90AB002186038 60A583
8D00D0A484A2008A9900148 8D0FAA484BD9C25
990014C8E8E014D0F4990014,555
1530 DATA C8D0FA60A48AB9B8258D6429B9B9
258D65294CFFFFA482B9FF05F013A0078488C6
832007292030 29A4888810F1,231
1540 DATA C68260A482B90106F0F8A0078488
E6832007292030 29A4888810F1E68260A58238
E90EA8B90006F0328482A00F,459
1550 DATA 8488C6842007292030 29A4888810
F160A58218690EA8B90006F0138482A00F8488
E6842007292030 29A4888810,823
1560 DATA F160A68DA90F9D7802A9019D8402
A58A8591A9FF858ABD7802C90FF022C90BD004
A900F016C907D004A902D00E,136
1570 DATA C90ED004A904D006C90DD0DDA906
858A3860BD8402D0D21860A20BA9008DC6028D
C802A9078585A9108586A019,157
1580 DATA A95491858810FBA5851869288585
A58669008586CA10E760A014AD0AD2C90F90F9
C951B0F5AAAD0AD2C90190F9,83
1590 DATA C904B0F5858EBD000610E2A58E18
69129D000688 10D7AD0AD22903C903F0F70A0A
0AA8A207B9842584 88A8A980,987
1600 DATA 990006A488C8CA10EF60A000A900
8D3002A9248D3102B90E06F03F1002A9000A00A
8488AA980AA8B900078580B9,237
1610 DATA 01078581A000BDE6249180E8C8C0
02D0F5A000A5801869288580A581690085 81BD
E6249180E8C8C002D0F5A488,125
1620 DATA C8C046D0B760D8A9018509A9EC85
0CA92A850DA9038D0FD2A9008D44028D08D2A2
809500E8D0FBA8990006C8D0,651
1630 DATA FAA9098592859 3A045B93E25990E
068810F7A011B9CC2599B0068810F7A9008580
A907858 1A200A92C858BA910,760
1640 DATA 858CA000A58B186902858B9180A5
8C6900C89180858CC8C01CD0E9A58018691C85
80A5816900858 1A58B186934,47
1650 DATA 858BA58C6900858CE8E005D0C820
412D204A2A20B728A065A228A906205CE42009
28201E2A20962A20BF282095,259
1660 DATA 28A94E8583A9468584A90F8582A9
008D8D24A58DF00DA9A68583A91A8582A9698D
8D2420542820962A203029A6,123
1670 DATA 8D20EA2990062055294CC32BA003
A9FF99C8258810FAA482848FB900068590 30DB
C913B0D7A68DD006C90A9006,369
1680 DATA B0CDC90A90C9A003B9B4258D0E2C
B9B0258D0D2C8488A482B9FFFF201029B007A4
88980A99C825A4888810DCA0,879
1690 DATA 03B9C8251010 8810F8A9E3A02520
2A28201F284CC32B206C2820DA29A58A30F9A0
03D9C825F0098810F8207028,381
1700 DATA 4C3A2C20552920DA299028A68DA0
1DA9008514BD8402F01BA514F0F78810F0A491
B9C0258D7C2CB9C1258D7D2C,421
1710 DATA 20FFFF4C3A2C206C28A482B90006
1012A5909900 06A48FA9809900062000294C9B
2BA8C90A900A849A88B9B006,740
1720 DATA 85C3D00884998 8B9B00685C2A490
C00A900A849A88B9B00685C3D00884 9988B9B0
0685C220B728A9408D0ED420,398
1730 DATA B02EA201B499B5C28899B006CA10
F5A58DAAC598D00A4901AAA590A48299 0006A9
80A48F9900062000 29D692AD,843
1740 DATA B806F008ADC106F0034C802B8A49
011869118D00262009282054 28201E2A20962A
20BF28209528A9F9A0252 02A,418
1750 DATA 28204B2820F627A9408D0ED420B7
284CEC2AA9008D2F02A9248D3002A9248D3102
AD0AD229F08597A9208DF402,829
1760 DATA 20F627200928A9218D0002A92E8D
0102A9C08D0ED4A004B9DE2599C4028810F7A9
008585A9118586A200BD1126,311
1770 DATA F012A8E8BD1126C9FFF009C9FEF0
169185C8D0EFA585186920858 5A58669008586
E84C862DA93D8D2F02A9108D,189
1780 DATA 3724A9008D3624854D8594A90085
87200729E687A587C908F00C8D05D4AD1FD0C9
06F03FD0E9A00084878C05D4,319
1790 DATA AD3624186920 8D3624AD37246900
8D3724E694A594C925F0B2A002D9A726F00588
10F830B9A9008514A514C978,212
1800 DATA F0AFAD1FD0C906D0F3A9408D0ED4
4CB728488696A5978595A207A5958D0AD48D19
D0186902 8595CA10F0A207A5,975
1810 DATA 9538E9028D0AD48D19D08595CA10
F068A6964048A9028D0AD48D18D0A9658D0002
A92E8D0102684048A9008D0A,687
1820 DATA D48D18D0A9508D0002A92E8D0102
6840A013A9019900 1099B4108810F7A9085A5
A91085A6A20820A22EA901A0,148
1830 DATA 0091A5A01391A5CA10F060A5A518
691485A5A5A6690085A660AD8402F0FBAD8502
F0F6A232A058A907205CE4A9,858
1840 DATA 0185A1200928A9008DC4028DC502
8DC6028DC702A9438D3002A9248D3102200931
207A2E20603320362F20422F,336
1850 DATA 20F62720902F20BA308D1ED02016
33204C33A5A0F0F6207833C6A0A6A020C533A6
A0D6C2B5C2F0062085334CC7,609
1860 DATA 2E8A4901859820783320853 3A907
A2E4A062205CE44CB728A004B9AA2699C40288
10F760A93E8D2F02A9018D6F,292
1870 DATA 02A93A8DC0028DC202A9AA8DC102
8DC302A9108D07D4A9038D1DD0A900A2039D00
D09D08D0CA10F7A90085B985,458
1880 DATA BA60A900A0FF99001688D0FA60A9
00A0FF99001788D0FA60A06FA20FBD76279900
14BD86279101588CA10F0A9,921
1890 DATA 388D00D085A9A9C08D01D085AAA9
6085ADA97085AE60A9008D1ED0B5C4F04FB5B5
C907D025B5C4C908D0152026,841
1900 DATA 31B5B3F007201834A5CBD005A900
95C460F6B1F6A9B5A99D00D0D6C460C90BD020
B5C4C908D01020263 1B5B3F0,836
```

```
1910 DATA DE201834A5CBF0D7D6B1D6A9B5A9
9D00D0D6C460B5C6F074B5B5C90ED036B5C6C9
10D01A202631B5B3F0072018,170
1920 DATA 34A5CBD005A90095C660B5B138E9
1495B120AD308884A3A000B1A591A3C8C011D0
F7D6ADD6C660C90DD034B5C6,514
1930 DATA C910D015202631B5B3F0CD201834
A5CBF0C6B5B118691495B120AD30C884A3A011
B1A591A38810F9C8A90091A5,286
1940 DATA F6ADD6C660BD780295B5C907D005
A90895C460C90BD002F0F5C90ED005A91095C6
60C90DF0F760BD642785A685,413
1950 DATA A4B5ADA884A560A95185B1A97685
B2A90385A2859EA91E85CEA90F85CFA90085BC
85BD85C485C585C685C785A0,897
1960 DATA 85BB85C985A160A00F20F33099E6
2788D0F760AD0AD2C91690F9C9B2B0F5A20FDD
6627F0EECA10F86020E730A2,409
1970 DATA 0FBDE627A886B7AD0AD22903AABD
4D27A6B7990010CAD0E96086A7A90095B3B5B1
85A8B5B5A203DD5127F003CA,306
1980 DATA 10F8BD5527185A8A884A8A6A7B9
0010D004A90195B3608D1ED0B5BCD02ABD8402
F00160B5B586A8A203DD5127,812
1990 DATA F00ACA10F8A900A6A895BC60A6A8
95BCA90785C9A90985CA20F83186A7B5BCC907
D00BB5AB18690295AB9D02D0,754
2000 DATA 60C90BD00BB5AB38E90295AB9D02
D060C90ED01EA90285C820EC318884C0A000B1
BE91C0C8C008D0F7D6AFC6C8,947
2010 DATA A5C8D0E760C90DF00160A90285C8
20EC31C884C0A007B1BE91C08810F9A0009891
BEF6AFC6C8A5C8D0E460BD59,656
2020 DATA 2785BF85C1B4AF84BE60B5A995AB
B5AD18690495AF86A7E000D01B207A2FB4AFA2
00BDD627990016C8E8E008D0,566
2030 DATA F4A6A7B5AB9D02D06020852FB4AF
A200BDDE27990017C8E8E008D0F4A6A74C1E32
60A978859CA900859BC69BA5,567
2040 DATA 9BD0FAC69CA59CD0F060A9468 59C
D0E9A5BBF0062003344C62E4A5A1F006207D28
4C62E4E69DA59DC909F0034C,864
2050 DATA F532A900859DE69FA59FC9039002
A900859FA59FD017A9768DD632A9278DD732A9
868DE632A9278DE7324CD132,692
2060 DATA C901D017A9968DD632A9278DD732
A9A68DE632A9278DE7324CD132A9B68DD632A9
278DD732A9C68DE632A9278D,691
2070 DATA E732A4ADA200BD9627990014C8E8
E010D0F4A4AEA200BDA627990015C8E8E010D0
F4A900854D8D1ED0A2002051,371
2080 DATA 31E8205131A20020BB2FE820BB2F
208E3320B0332039344C62E4AD06D0F007A900
85BC8D02D0AD07D0F007A900,46
2090 DATA 85BD8D03D0AD0ED02902F006A901
85BA85A1AD0FD02901F006A90185B985A18D1E
D060A20086A0A001B5B9F003,354
2100 DATA 84A060E8C8B5B9D0F760A200B4C2
88A90299C9108810FAE8B4C28899D3108810FA
60A90085BC85BD8D02D08D03,633
2110 DATA D060A9008D00D08D01D060C69EA5
9EF00160A902859EA6A2F00EA9848D01D2BDAF
268D00D2C6A260A20286A260,379
2120 DATA A5C9D00160A6C9A9AA8D03D2BD5B
278D02D2C6C960B4ADA94385BB84A3A90585A8
A92685A6A9B285A5BD642785,568
2130 DATA A4A000B1A591A3C8C010D0F7203D
3218A5A5691085A5A5A669008 5A6C6A8A5A8D0
DF203D3260A6BBD00160A6BB,436
2140 DATA A98A8D03D2BD02278D02D2C6BB60
86CCA90085CBE000D00EE8A5A8D5B1D002F002
E6CBA6CC60CAA5A8D5B1D0F4,133
2150 DATA F0F4C6CEA5CEF00160A91E85CEA6
CFBDE627A8A90099001020F330A6CF9DE627A8
AD0AD22903AABD4D27990010,975
2160 DATA C6CFA5CFD004A90F85CF60000000
00000000000000000000000000000000000000
00000000000000000000000,584
2170 REM * 5265 BYTES
```

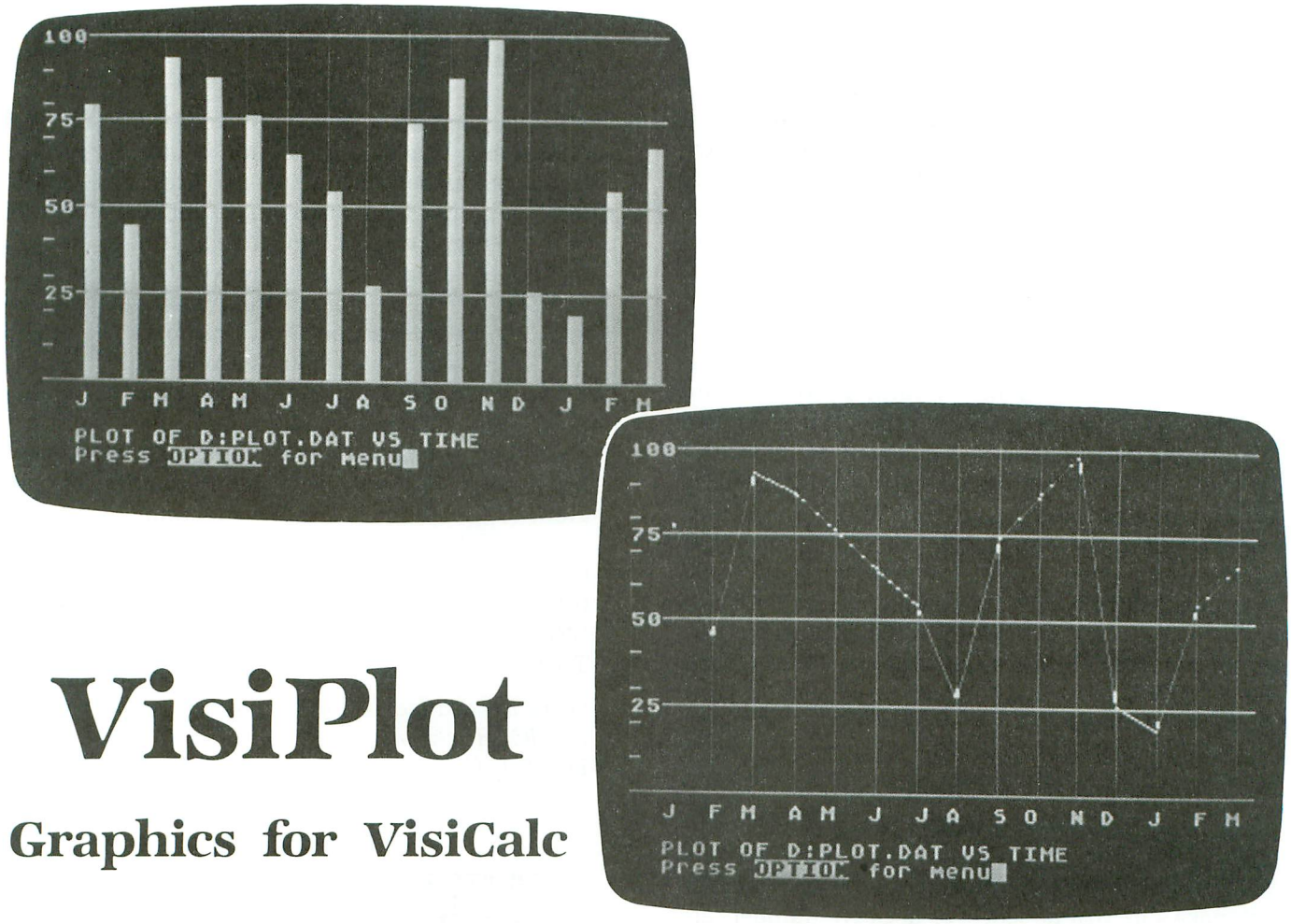
●

CHECKSUM DATA.

(see page 10)

```
10 DATA 278,351,496,811,423,729,200,60
3,555,573,694,613,29,205,203,6763
160 DATA 747,198,962,621,288,30,155,10
4,169,631,313,784,658,362,796,6818
1060 DATA 841,572,838,582,401,963,252,
938,330,577,937,378,988,632,377,9606
1210 DATA 506,956,67,24,998,827,388,45
6,965,986,262,59,155,348,65,7062
1360 DATA 314,274,982,125,263,602,59,9
18,607,43,820,645,644,903,265,7464
1510 DATA 767,716,832,649,395,923,833,
976,905,672,694,955,811,826,542,11496
1660 DATA 783,809,80,493,958,755,673,8
49,550,731,639,144,785,703,17,8969
1810 DATA 949,650,839,540,722,779,89,9
04,940,793,949,872,812,58,183,10079
1960 DATA 125,103,884,103,44,982,29,14
5,77,991,42,839,774,812,991,6941
2110 DATA 32,27,821,129,198,876,644,27
27
```

●



VisiPlot

Graphics for VisiCalc

by Larry M. Bevard

Before I became an unemployment statistic, I was a quality control supervisor. Part of my job was to graph and analyze data generated in the plant and the lab to see how to improve quality, productivity and procedures. The graphing was all done by hand, and the analysis was done with the help of a calculator—all very laborious.

In 1982 I got my first computer, a Timex-Sinclair 1000, and set up a graphic plotting/data analysis program. But, because of the limited graphics, there was no data storage (the Timex only allowed data to be saved with a copy of the program) and it had other shortcomings, including no printer; it was easier to do the graphics and analysis by hand.

I soon upgraded to an Atari 400, then an 800, and now an 800XL with disk drive and printer. I also bought a copy of **VisiCalc**. By this time, I was unemployed; all of my equipment was of little use — or so I thought. I started using **VisiCalc** to keep track of my utility bills, my charge cards and to analyze budget and money matters. I realized that using the graphics package to analyze my bills would be as valuable at home as on the job. **VisiPlot** was born.

Working with VisiPlot.

After entering Listing 1, SAVEing it to disk and RUNning it, you must decide to use either a single variable or a double variable. Entering a *1* will allow you to graph one set of data to be plotted on the Y-axis (the vertical axis). The X-axis will be some even unit, such as months or other units of time. This will be useful for plotting out how your heating or electric bill has changed over time. Entering a *2* will use two sets of data and graph them out in data pairs. This is useful for tracking changes in the electric rates. On the graphs, you're offered the option of labeling the X-axis as months or as consecutive num-

bers. Using two sets of data, both axes are labeled with the actual value.

The program then asks you to enter the name of the data file(s) to be used. Just enter the name with its extension; you don't have to enter *D:*. If you have two variables to analyze, you'll be prompted twice for filenames. The first will be graphed on the X-axis, and the second on the Y-axis. These filenames are saved by the program to use in labeling the graph. Scaling factors are then set up on both sets of data, and the main menu is displayed.

Your options are: "Bargraph," which prints a bargraph of the data, useful for utility bills; "Point Graph," which prints a scatter graph (unconnected dots) of the data (useful for looking at double variable data, also interfaces with the statistical routine to plot a best fit line through the data on the double variable option); "Line Graph," which prints graph as connecting points; "Statistical," to find mean (average) standard deviation, standard error and variance (all three measures of how much the data varies from the average), also finds a best fit line equation to the data; "New Data," used to clear variables and enter a new set of data.

After graphing out the data, press OPTION to go back to the menu and try another graph.

Creating data files.

To create the data files needed to use with VisiPlot.bas, enter your data as usual with **VisiCalc**, but make sure that no labels or negative numbers are in the row of data generated. This is very important, or **VisiPlot** will complain.

After getting your data arranged (either in rows or columns) put the cursor at the first data cell. Then use the command */p* and *f* for file. You'll be prompted for a filename. (Use a very descriptive name, as this will also be used to label the graph.) Then press RETURN and move the cursor to the end of your data. Pressing RETURN will SAVE your data in suitable form with the extension of *.PFR*.

Creating data with PLOTDATA.BAS.

If you don't have **VisiCalc**, but would like to use this program, then ENTER, SAVE and RUN Listing 2, PLOTDATA.BAS. This program will allow you to enter and/or append data to a number file, so you can use it with **VisiPlot**.

Very simply, this program allows you to enter up to 320 data points into a file or to append data onto the end of an existing file. You won't have to enter all 320 points at once. On your last entry, press RETURN at the prompt. Your data will be saved to disk. This file is just like one created using **VisiCalc**; the difference is that you worked harder to get this one.

Postscripts.

Using graphics 8, instead of 7 or 7+ like **B/Graph** or **Graphit**, I gave up the option of plotting in three colors, but gained finer points and lines, and a better-looking graph. Lack of color shouldn't be a problem with this type of program, though it does enhance the usage. Because of the high-resolution screen, you can plot up to 310 separate data points (as opposed to 100 with **B/Graph**, or 32 on **Graphit**, and you can't plot line graphs from data with **Graphit**).

I said before that negative numbers weren't allowed in **VisiPlot**. I didn't allow them because most real world data is in positive numbers (except my checking account balance). Feel free to modify this program to include negatives if you want.

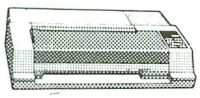
Most of the graphs should be in good form to dump to a printer if you desire. But if you want to change the wording on the bottom two lines, just press BREAK and use the cursor to make any changes. That's why I used graphics 8 instead of 8+16.

Larry Bevard learned to program in FORTRAN and BASIC. He's now using BASIC, BASIC XL, Action!, FORTH, C and assembly on an 800 and 800XL.

Listing 1.
BASIC listing.

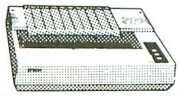
```
1 REM VISIPLOT.BAS BY Larry M. Bevard
10 DIM X(320),X$(35),Y(320),Y$(35),F$(
35),R$(35),B$(35),RE$(1),TXT$(64),MONT
H$(12)
20 XCOUNT=0:YCOUNT=0:Z=1:SCAT=1:STAT=0
:LABEL=1:MONTH$="JFMAMJJASOND"
30 ? "KIS THIS A;":? "    (1) SINGLE VA
RIABLE OR":? "    (2) DOUBLE VARIABLE":
? "GRAPH"
40 INPUT I:IF I=1 THEN LET R$="TIME":G
OTO 140
50 GOSUB 9000
60 LET R$=F$
70 Z=2:TRAP 120
80 FOR I=1 TO 320
90 GOSUB 10000
100 LET X(I)=D1
110 NEXT I
120 XMIN=DMIN:XMAX=DMAX
130 CLOSE #3:XCOUNT=I-1:POP :GOTO 300
140 ? "USE MONTH OR NUMBER LABEL ON X-
AXIS"
150 INPUT RE$:IF RE$="M" THEN LET LABE
L=2:? "STARTING MONTH #";:INPUT MONTH
300 GOSUB 9000
310 TRAP 380
320 FOR I=1 TO 320
330 GOSUB 10000
340 LET Y(I)=D1
350 NEXT I
380 YMIN=DMIN:YMAX=DMAX
390 CLOSE #3:YCOUNT=I-1:POP
400 IF Z=1 THEN LET XCOUNT=YCOUNT
410 REM SET Y AXIS SCALE
420 LET YSCALE=YMAX/150
```

Lyco Computer Marketing & Consultants



1091.....$233

SAVE ON THESE IN STOCK PRINTERS



SG-10.....$208

AXIOM SEIKOSHA

GP550AT (Atari)	222
GP550CD (C-64)	222
GP700AT (Atari)	439
GP700AP (Apple)	439
ELITE5CD (C-64)	229

C. ITOH

ProWriter 8510Sp+	349
1550Sp+	489
StarWriter	769
PrintMaster	929

TOSHIBA

P1340	469
P351+	1149
P341P	969
P341S	999
351 Sheet Feeder	529

CARDCO

32K BUFFER (C-64)	59

CORONA

LP300 Laser Printer	2686
200361 Toner Cartridge	89

EPSON

FX85 (New)	333
LX80	212
FX185 (New)	464
LX90 (New)	226
SQ2000 (New)	1555
JX80	467
Homewriter 10	193
CR-20-Atari	153
CR-220-C-64	153
DX-10 (New)	207
DX-20 (New)	297
HS-80 (New)	288
LQ1500P	975
LQ1500S	1039
RX-100	356
FX-100+	CALL

CITIZEN

MSP-10	269
MSP-15	358
MSP-20	337
MSP-25	495
Sheetfdr (10/20)	189
Sheetfdr (15/25)	199

OKIDATA

Okimate 10	179
Okimate 20	CALL
182	214
192	348
193	563
92P	349
93P	565
84P	645
92 Imagewriter	349

*IBM versions also

BROTHER

HR-15XL-P	359
HR-15XL-S	359
HR-35P	839
HR-35S	839
2024L-P	949
M1009-P	189

JUKI

Juki 6100	347
RS232 Serial Board	55
6100 Tractor	119
6100 Sheet Feeder	209
Juki 6300	757

LEGEND

880	188
1080	222
1380	262
1385	296
LEGEND 808	159

DIGITAL DEVICES

16K BUFFER	75
32K BUFFER	89
64K BUFFER	125

DIABLO

D25	549
630 API	1599
630 ECS	1799
D 80 1F	2395
P 32 CQ1	699
P 38	1749
C 150	999
DX-35 (NEW)	CALL
AP-80	CALL

PANASONIC

1091	*233*
3131 (NEW)	269
1092	373
1093	426
3151 Letter	426
4K Buffer	65

SILVER REED

EXP400	249
EXP500	295
EXP550	399
EXP770	749

STAR MICRONICS

SG 10	208
SG-15	373
SD-10	336
SD-15	442
SR 10	483
SR-15	583
SB 10	595
Power Type	303
SG-10 C-64 (NEW)	CALL

MONITORS

AMDEK

300 Green	118
300 Amber	128
310 Amber IBM	155
Color 300 Audio	234
Color 500 Composite	369
Color 600	397
Color 700	495
Color 710	569

TEKNIKA

MJ-10 Composite	179
MJ-22 RGB	255

PANASONIC

DT1300D	13"	RGB/Composite 247
DTM140	14"	RGB/Composite 329
DTH103	10"	RGB Hi Res 395
DTS101	10"	Composite 175
DT1000G	10"	RGB 166
TX12H3P	12"	Color 419
TR120M1PA	12"	Green 109
TR120MBPA	12"	Amber 109
TR122M9P	12"	Green IBM 148
TR122MYP	12"	Amber IBM 148

X-TRON

Comcolor I Composite Green	177

520 ST SOFTWARE

INFOCOM (520ST)

Wishbringer	28.95
Zork I	28.95
Zork II	31.95
Zork III	31.95
Deadline	33.95
Starcross	33.95
Suspect	31.95
Suspended	33.95
Cutthroats	28.95
Sea Stalker	28.95
Infidel	31.95
Sorcerer	31.95
Enchanter	28.95
Witness	28.95

GLENTOP

ST Logo Book	15.95

MARK of UNICORN (520ST)

HEX	29.95
MINCE	129.95
PC/InterComm	99.95

FTL GAMES

Sundog	29.95

SIERRA

ULTIMA II	39.95

HABA

10 Megabyte	CALL
Hippo C Compiler	49.95
Check Minder	49.95
Business Letters	36.95
Wills	36.95

SOFTECHNICS

Timelink	75.00
Rhythm	39.95

MIRAGE CONCEPTS

EXPRESS	34.95

MODEMS

HAYES

SMARTMODEM 300	133
SMARTMODEM 1200	377
SMARTMODEM 1200B	347
SMARTMODEM 2400	598
MICROMODEM IIE	135

ANCHOR

Volksmodem	55
Volksmodem 12	186
Mark 12	229

DRIVES

INDUS
Atari
215

INTERFACING

AXIOM

AT846 (Atari)	65

ATARI

850 Interface	109

DIGITAL DEVICES

ApeFace XLP (Atari)	49
U-Print A (Atari)	54
U-CALL RS232 (Atari)	37

CARDCO

C/?AT (Atari)	49

DISKETTES

DENNISON

ELEPHANT 5¼" SSSD	11.99
ELEPHANT 5¼" SSDD	12.99
ELEPHANT 5¼" DSDD	14.99
PREMIUM 5¼" SSDD	13.99
PREMIUM 5¼" DSDD	15.99

VERBATIM

5¼" SSDD	13.99
5¼" DSDD	19.99

BONUS

5¼" SSDD	9.99
5¼" DSDD	12.99

ATARI

800XL	CALL
130XE (NEW)	CALL
520ST (NEW)	CALL
1050 Drive	165
1010 Recorder	42
1020 Printer	55
1025 Printer	159
1027 Printer	179
850 Interface	109

ATARI SOFTWARE (NEW)

Codewriter	35.75
Filewriter	20.75
Reportwriter	20.75
Menuwriter	20.75
Home Integrator	19.75
Small Bus. Inventory	11.75
Salesman Expenses	11.75
Accs Rec/Pay	11.75
Retail Invoice	11.75
Final Legacy	15.75
Adventure Writer	18.75

MICROPROSE (Atari)

Kennedy Approach	21.75
Crusade in Europe	24.75
Decision in Desert	24.75
Solo Flight	20.75
Nato Commander	20.75
Spitfire Ace	18.75
F-15 Strike Eagle	20.75

SUB LOGIC (Atari)

Flight Simulator II	*32.75*
Night Mission Pinball	20.75

BRODERBUND (Atari)

The Print Shop	28.95
Graphics Library	18.95
Graphics Library II	19.50
Bank St. Writer	42.75
Whistler's Brother	18.95
Spelunker	18.95
Stealth	18.95
Serpent's Star	24.95
Mask of the Sun	24.95

TOLL FREE 1-800-233-8760

MasterCard

TO ORDER

CALL TOLL FREE
800-233-8760
Customer Service 1-717-327-1825

VISA

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

RISK FREE POLICY

In-stock items shipped within 24 hours of order. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. Volume discounts available. PA residents add sales tax. APO, FPO, and international orders add $5.00 plus 3% for priority mail service. Advertised prices show 4% discount for cash, add 4% for MasterCard or Visa. Personal checks require 4 weeks' clearance before shipping. Ask about UPS Blue and Red label shipping. All merchandise carried under manufacturer's warranty. Free catalog with order. All items subject to change without notice.

CIRCLE #119 ON READER SERVICE CARD

```
430 REM SET X AXIS SCALE
440 IF Z=2 THEN XSCALE=XMAX/310:GOTO 4
90
450 XSCALE=YCOUNT/310
460 FOR I=1 TO YCOUNT
470 X(I)=I
480 NEXT I
490 GOTO 1000
500 GRAPHICS 8:SETCOLOR 2,0,0:SETCOLOR
 1,0,14:COLOR 1
520 GOSUB GRAPH
590 IF Z=2 THEN GOSUB 6000:GOTO 610
600 GOSUB 7000
610 GOSUB 8000
980 POKE 656,2:POKE 657,2
998 ? "PLOT OF ";F$;" VS ";R$:? "Press
 OPTION for menu";
999 IF PEEK(53279)<>3 THEN 999
1000 REM MAIN MENU
1010 GRAPHICS 1+16
1020 POSITION 3,2:? #6;"VisiPlot - Men
U"
1030 POSITION 0,7:? #6;"  1. bARGRAPH"
1040 ? #6;"  2. pOINT GRAPH"
1050 ? #6;"  3. lINE GRAPH"
1060 ? #6;"  4. sTATISTICS"
1070 ? #6;"  5. nEW DATA"
1300 POSITION 4,22:? #6;"enter number"
1790 SCAT=1:PZY=0
1800 OPEN #4,4,0,"K:"
1930 GET #4,CHOICE:CLOSE #4:IF CHOICE=
49 THEN LET GRAPH=3000:GOTO 500
1940 GRAPH=4000
1950 IF CHOICE=50 THEN SCAT=2:PZY=150:
GOTO 500
1960 IF CHOICE=52 THEN 12000
1970 IF CHOICE=53 THEN RUN
1980 IF CHOICE=51 THEN 500
1990 GOTO 1930
2000 REM TEXT CONVERTION SUBR
2010 SA=PEEK(89)*256+PEEK(88)
2020 IF Y>192 OR X>40 THEN RETURN
2025 ROWS=192:COLS=40
2030 START=SA+Y*COLS+X
2070 FOR E1=1 TO LEN(TXT$)
2080 GOSUB 2200
2090 CHARSET=PEEK(756)*256
2100 CHARSET=CHARSET+E3*8
2110 FOR E2=6 TO 1 STEP -1
2120 POKE START+E2*COLS,PEEK(CHARSET+E
2)
2130 NEXT E2
2140 X=X+1:IF X>=COLS THEN START=START
+COLS*8:X=0
2150 START=START+1
2160 NEXT E1
2165 X=0
2170 RETURN
2195 REM ATASCII CONVERTION ROUTINE
2200 E3=ASC(TXT$(E1))
2210 IF E3<32 OR (E3>127 AND E3<160) T
HEN E3=E3+64:RETURN
2220 IF E3>31 AND E3<96 THEN E3=E3-32
2230 RETURN
3000 REM PLOT BAR GRAPH ROUTINE
3005 PLOT 0,0:DRAWTO 0,155:DRAWTO 319,
155
3020 IF Z=1 THEN 3050
3030 IF YCOUNT>XCOUNT THEN LET YCOUNT=
XCOUNT
3035 IF YCOUNT<XCOUNT THEN LET XCOUNT=
YCOUNT
3050 FOR I=1 TO YCOUNT
3053 PLOT X(I)/XSCALE,155-Y(I)/YSCALE:
DRAWTO X(I)/XSCALE,155:DRAWTO X(I)/XSC
ALE+7,155
3058 DRAWTO X(I)/XSCALE+7,155-Y(I)/YSC
ALE:DRAWTO X(I)/XSCALE,155-Y(I)/YSCALE
3060 POSITION X(I)/XSCALE,155
3063 POKE 765,1:XIO 18,#6,0,0,"S:"
3070 NEXT I
3080 RETURN
4000 REM PLOT AND DRAWTO ROUTINE
4005 PLOT 0,0:DRAWTO 0,155:DRAWTO 319,
155
4010 PLOT X(1)/XSCALE,155-Y(1)/YSCALE
4020 IF Z=1 THEN 4040
4030 IF YCOUNT>XCOUNT THEN YCOUNT=XCOU
NT
4035 IF YCOUNT<XCOUNT THEN XCOUNT=YCOU
NT
4040 IF SCAT=2 THEN 4090
4050 FOR I=2 TO YCOUNT
4060 DRAWTO X(I)/XSCALE,155-Y(I)/YSCAL
E
4070 NEXT I
4080 RETURN
4090 FOR I=2 TO YCOUNT
4100 PLOT X(I)/XSCALE,155-Y(I)/YSCALE
4110 NEXT I
4120 IF STAT=0 OR Z=1 THEN RETURN
4125 YPLS0=0:YPLS1=YMAX
4130 IF XPLS0<0 THEN XPLS0=0:YPLS0=-B/
M
4140 PLOT XPLS0/XSCALE,155-YPLS0/YSCAL
E:DRAWTO XPLS1/XSCALE,155-YPLS1/YSCALE
4150 RETURN
6000 REM PLOT X AXIS SCALE
6010 TRAP 6130
6020 XLINE=XMAX/100:XLINESCALE=10:IF X
LINE<10 THEN 6040
6030 XLINE=XLINE/10:XLINESCALE=XLINESC
ALE*10:IF XLINE>10 THEN 6030
6040 XLINE=INT(XLINE+0.5)*XLINESCALE*1
0
6050 LET U=2.5:IF XLINE/XLINESCALE/U>2
0 THEN U=5
6060 IF XLINE/XLINESCALE/U>20 THEN U=1
0
6070 POKE 656,1:POKE 657,2
6080 ? "XSCALE = (VALUE)*";XLINESCALE*
U;
6090 IF SCAT=2 THEN 6125
6095 TRAP 6125
6100 FOR I=0 TO XLINE STEP XLINESCALE*
U
6110 PLOT I/XSCALE,0:DRAWTO I/XSCALE,1
55
6120 NEXT I
6125 TRAP 6155
6130 FOR I=0 TO XLINE STEP XLINESCALE
6135 TRAP 6155
6140 PLOT I/XSCALE,150:DRAWTO I/XSCALE
,155
6150 NEXT I
6155 PZ=0
6160 FOR I=XLINESCALE*U*2 TO XLINE STE
P XLINESCALE*U*2
6165 TRAP 6250:PZ=PZ+2
6220 POKE 657,INT(I/XSCALE/8):POKE 656
,0
6230 ? PZ;
6240 NEXT I
6250 RETURN
7000 FOR I=0 TO YCOUNT
7010 PLOT I/XSCALE,155:DRAWTO I/XSCALE
,PZY
7020 NEXT I
7030 MONTH1=MONTH
7140 PZ=0
7150 IF LABEL=1 THEN 7220
7160 FOR I=0 TO YCOUNT
7165 TRAP 7270
7180 POKE 657,INT(I/XSCALE)/8+2:POKE 6
56,0
7190 ? MONTH$(MONTH1,MONTH1);
```

```
7200 MONTH1=MONTH1+1:IF MONTH1)12 THEN
 MONTH1=1
7210 NEXT I:RETURN
7220 FOR I=2 TO YCOUNT STEP 2
7230 TRAP 7270:PZ=PZ+2
7240 POKE 657,INT(I/XSCALE)/8:POKE 656
,0
7250 ? PZ;
7260 NEXT I
7270 RETURN
8000 REM PLOT Y AXIS SCALE
8010 TRAP 8260
8020 YLINE=YMAX/100:LINESCALE=10:IF YL
INE<10 THEN 8040
8030 YLINE=YLINE/10:LINESCALE=LINESCAL
E*10:IF YLINE)10 THEN 8030
8040 YLINE=INT(YLINE+0.5)*LINESCALE*10
8050 LET U=2.5:IF YLINE/LINESCALE/U)15
 THEN U=5
8060 IF YLINE/LINESCALE/U)15 THEN U=10
8110 FOR I=0 TO YLINE STEP LINESCALE
8120 TRAP 8200
8130 PLOT 0,155-I/YSCALE:DRAWTO 5,155-
I/YSCALE
8140 NEXT I
8200 FOR I=0 TO YLINE STEP LINESCALE*U
8205 IF SCAT=2 THEN 8230
8210 TRAP 8260
8220 PLOT 0,155-I/YSCALE:DRAWTO 319,15
5-I/YSCALE
8230 IF I=0 THEN NEXT I
8240 TXT$=STR$(I):X=0:Y=INT(152-I/YSCA
LE):GOSUB 2000
8250 NEXT I
8260 RETURN
9000 ? "ENTER NAME OF FILE"
9010 DMIN=9999999:DMAX=-9999999
9020 TRAP 9020
9030 INPUT F$
9040 LET B$=""
9050 LET B$="D:":LET B$(3)=F$
9060 CLOSE #3:OPEN #3,4,0,B$
9070 RETURN
10000 INPUT #3,D1
10010 IF DMIN)D1 THEN DMIN=D1
10020 IF DMAX<D1 THEN DMAX=D1
10030 RETURN
12000 SX=0:SY=0:SX2=0:SY2=0:SXY=0:XVAR
=0:YVAR=0
12005 FOR I=1 TO XCOUNT
12010 SX=SX+X(I)
12020 SY=SY+Y(I)
12030 SX2=SX2+X(I)*X(I)
12040 SY2=SY2+Y(I)*Y(I)
12050 SXY=SXY+X(I)*Y(I)
12060 NEXT I
12070 XMEAN=SX/XCOUNT
12080 YMEAN=SY/XCOUNT
12090 FOR I=1 TO XCOUNT
12100 XVAR=XVAR+(X(I)-XMEAN)*(X(I)-XME
AN)
12110 YVAR=YVAR+(Y(I)-YMEAN)*(Y(I)-YME
AN)
12120 NEXT I
12130 XVAR=XVAR/(XCOUNT-1)
12140 YVAR=YVAR/(XCOUNT-1)
12150 XSTDD=SQR(XVAR)
12160 YSTDD=SQR(YVAR)
12170 XSTDERR=XSTDD/XMEAN*100
12180 YSTDERR=YSTDD/YMEAN*100
12190 M=(XCOUNT*SXY-SX*SY)/(XCOUNT*SY2
-SY*SY)
12200 B=(SY2*SX-SY*SXY)/(XCOUNT*SY2-SY
*SY)
12210 XPLSO=B:XPLS1=M*YMAX+B
12220 PRINT ,"X","Y"
12230 ? :? "MEAN",INT(XMEAN*1000)/1000
,INT(YMEAN*1000)/1000
12240 ? "STDDIV",INT(XSTDD*1000)/1000,
INT(YSTDD*1000)/1000
12250 ? "STDERR",INT(XSTDERR*1000)/100
0,INT(YSTDERR*1000)/1000
12260 ? "C.V.",INT(XVAR),INT(YVAR)
12270 ? :? :? "LEAST SQU. EQUATION;":?
 "X = ";INT(M*100)/100;" Y + ";INT(B*1
00)/100
12280 ? :? "PRESS RETURN FOR MENU"
12300 LET STAT=1
12310 INPUT RE$
12320 GOTO 490
```

●

CHECKSUM DATA.

(see page 10)

```
1 DATA 920,118,300,603,978,11,751,157,
73,837,28,731,389,94,942,6932
150 DATA 979,815,725,334,3,45,747,415,
24,645,367,113,371,391,55,6029
460 DATA 994,626,758,967,867,258,365,8
15,821,359,310,903,483,331,412,9269
1030 DATA 864,73,829,970,589,292,275,4
69,746,0,62,642,572,187,747,7317
2000 DATA 99,407,433,852,246,229,935,2
76,629,133,571,685,11,555,687,6748
2165 DATA 12,791,994,665,973,500,790,5
89,55,557,339,345,292,134,70,7106
3060 DATA 783,107,495,791,513,57,935,5
57,74,83,983,295,214,497,793,7177
4090 DATA 299,57,494,748,318,155,653,7
93,514,692,401,880,260,504,145,6913
6070 DATA 138,264,980,723,420,648,499,
725,367,726,640,502,311,320,147,7410
6220 DATA 213,541,504,800,294,855,498,
410,302,331,303,724,274,819,327,7195
7210 DATA 884,842,146,238,545,508,804,
519,709,54,720,15,244,888,81,7197
8120 DATA 701,132,505,267,993,715,288,
892,934,509,805,462,152,705,926,8986
9040 DATA 626,341,451,802,181,733,755,
46,384,25,450,460,982,992,106,7334
12060 DATA 678,214,221,28,210,222,671,
545,550,137,144,955,961,369,937,6842
12210 DATA 120,440,812,268,588,503,443
,209,417,204,757,4761
```

●

Listing 2.

```
10 DIM X(320),F$(20),B$(20):F$="":B$="
"
20 ? "PLOT DATA ENTRY"
30 ? :? :? "1) ENTER NEW DATA FILE"
40 ? "2) APPEND DATA TO FILE"
50 ? "3) GO TO BASIC"
60 ? :? :? "Which one";:INPUT R
70 ON R GOTO 1000,2000,90
80 GOTO 50
90 NEW
100 ? " Enter data (upto 320 points)":
? "Press RETURN to end data entry"
110 FOR I=1 TO 320
120 TRAP 200
130 ? "DATA FOR ";I;" ";
140 INPUT X:LET X(I)=X
```

VisiPlot *continued*

```
150 NEXT I
200 ? "{CLEAR}Saving file....."
210 FOR Z=1 TO I-1
220 PRINT #3;X(Z)
230 NEXT Z
240 ? "DONE":CLOSE #3:RUN
1000 TRAP 1000:? "{CLEAR}Enter file name for
 new file"
1010 INPUT B$:F$="D:":F$(3)=B$
1020 CLOSE #3:OPEN #3,8,0,F$
1030 GOTO 100
2000 TRAP 2000:? "{CLEAR}Enter file name to
append"
2010 INPUT B$:F$="D:":F$(3)=B$
2020 CLOSE #3:OPEN #3,9,0,F$
2030 GOTO 100
```

●

CHECKSUM DATA.

(see page 10)

```
10 DATA 266,163,227,676,744,732,211,61
9,269,418,327,698,892,330,743,7315
200 DATA 139,376,248,773,280,656,618,4
47,865,39,620,452,867,6380
```

●

THE ELECTRONIC CHECKBOOK
and
THE MONEY MANAGER
TIMEWORKS, INC.
444 Lake Cook Road
Deerfield, IL 60015
Both 48K Disk $24.95 each

by Karl E. Wiegers

Money. We never have enough of it. And, at the end of the month, we never know just where it's gone. Timeworks can't do much about the first problem, but they do offer some hope for the second. **The Electronic Checkbook** and **The Money Manager** are designed to help you keep track of your cash flow and to compare your financial reality with budgeted expectations.

With **The Electronic Checkbook** (**TEC**) you record financial transactions, including checks, deposits and cash payments. **The Money Manager** (**TMM**) lets you enter budget amounts for sixteen expense categories, record actual expenses and make comparisons. An interface procedure permits data from **TEC** to be read by **TMM**, so you only have to enter each item once.

TEC and **TMM** are similar in behavior and user interface. The programs come packaged in plastic boxes containing the program disk and user manual. The disks are not copy protected.

The manual for **TEC** covers both Atari and Commodore 64 versions of the program. Strangely, the manual for **TMM** is only for the Commodore, although the package and disk are clearly labeled Atari. However, the program conforms to the manual well enough to prevent a problem. The package says that the program is for the Atari 600/600XL/800 computers (whatever a 600 is). Both of these programs require 48K of RAM and one disk drive.

TEC and **TMM** use a basic black and white text screen for all displays, although you can choose whether you prefer black characters or black background. The displays are plain, but functional and easy to understand. No sound is used, except an occasional "brrraack" of the buzzer for illegal entries.

Each program has several menus that are clear and easy to use. Numerous additional prompts for choices usually demand just a single keypress. The programs do a very nice job of permitting the input of only legal characters.

For example, when numeric inputs are required, all keys besides a number or the period (decimal point) are ignored. Screen changes are very fast.

The user manuals are excellent. They contain thorough descriptions of all aspects of program use and give many examples. Step by step instructions for interacting with the various menus and prompts are presented in the most logical possible order. Many helpful notes are found throughout the manuals.

A tutorial in **TMM** walks the new user through a sample treatment of four months of budget data supplied on the program disk. Sections pertaining to the Atari, Commodore disk and Commodore cassette incarnations of the programs are clearly separated and marked. There are a few minor terminology discrepancies between screen displays and the manual (e.g., *search* versus *sort*), but overall the manuals are complete, well organized and very helpful.

Checkbook balancing was long ago suggested as a useful task for a home computer. Now, however, we all know that it's much more easily performed with a ten-dollar calculator. Fortunately, **TEC** is much more than just a checkbook balancing program. It really functions more like a database management system dedicated to a single task, with just one unalterable record type.

At the main menu for **TEC**, you first enter some general information about your checking account and the records you're storing on the data disk you provide. The next step is to enter your "transactions," which can be viewed and modified using a third menu option. An option to analyze transaction amounts shows the sum and average of a selected group of transactions. The "monthly reconciliation" choice is really the checkbook balancing step. Finally, you can print all of your stored records or a selected subset.

A transaction is a record of some expense or credit item, such as a check or deposit, or a cash transaction. Each record contains eight items or fields: (1) a transaction sequence number; (2) a description up to thirty characters long; (3) a four-character check or deposit number or the word *cash*; (4) an X signifying that the transaction has cleared the bank (or a blank showing that it has not); (5) the date of the transaction in the form MM/DD; (6) an "index number" or transaction type code; (7) the amount of the transaction in dollars and cents; and (8) your checkbook balance after the transaction.

One data disk must be dedicated to each account you wish to monitor. Up

to 1295 transactions can be stored on a single disk.

Identifying check transactions is simple; just enter the check number. Entering items such as cash withdrawals, check reorder fees and the like, seems to require that you assign these transactions fake check numbers. Deposits are identified with a *D* as the first character of the transaction number. I found it difficult to remember sequence numbers for deposits (needed for the checkbook reconciliation step), so record the number right in your checkbook for future reference.

To enter transactions, you reply to a series of prompts for each portion of the transaction record. You have a chance to correct any errors before the transaction is stored, and you can always change them later, using the "view/modify" menu option.

Transactions are saved to disk after every three or four entries. Your current checkbook balance is displayed after each transaction is entered, except that cash transactions don't affect the balance. The data entry process is about as fast and painless as you can expect for such a tedious task.

Both **TEC** and **TMM** provide sixteen categories for budgeted amounts and transaction entries. The "index number" for each category is used to link transactions from **TEC** to budget categories in **TMM** when using the data transfer interface feature. The titles of thirteen of these categories are provided by the programs, with the last three being wildcard "other" categories.

A menu option in **TMM** lets you alter category names to suit your needs, but no corresponding feature exists in **TEC**. Using **TEC**, you can assign additional index numbers from 17 to 99 to transactions, in order to classify entries as you wish. However, transactions with index numbers greater than 16 cannot be transferred to **TMM**. The inconsistent way that categories are handled in the two programs hampers the transfer of data between them, as you may need different classification schemes for the two programs.

An extremely useful feature of **TEC** is its ability to locate stored entries according to several criteria: transaction number, description, amount, date, index, check or deposit number. This procedure is called a "cross-search." After the matching records are retrieved from the complete data file, you can perform as many additional cross-searches as you like, to further subset the retrieved data into just those records you wish to see.

Character substrings within the transaction description can be used as search criteria. Also, ranges can be specified for numeric fields. For example, you can easily ask to see all the checks written to Sears for clothing in amounts between $10 and $45. The searching process is quite rapid, but it depends on the number of transactions in the file, of course.

In one test case, it took six seconds to locate the last three items in a small file of thirty entries when searching by index number. For all the "hits" thus obtained, you can opt to list the entire transaction record on the screen, then change it if necessary. A limited analysis procedure shows the total and average amounts of the transactions in the hit list. You can also print the records located as the result of your most recent cross-search procedure.

Printing records gives you a very simple dump of the data. Neither **TEC** nor **TMM** gives any options for different printer types, but the "plain vanilla" output should work fine on any flavor printer using the default character set.

Ironically, I found the **TEC** reconciliation of the checkbook balance tedious and cumbersome. You won't want to use **TEC** to help you actually balance your checkbook. Stick with your calculator.

TEC should have broad appeal, since almost everyone uses a paper checkbook. **TMM** is only useful to individuals who maintain a household or personal expense budget, although it may also be powerful enough to handle the budgeting needs of some small businesses. Its greatest limitation is the restriction to

sixteen budget categories, but with creativity you can get by. I found I had a rather large "miscellaneous" category.

I had some trouble initializing (or "initiating," as the program calls it) a data disk with **TMM**. The instructions tell you to "use a good quality 5¼-inch floppy disk," and they aren't kidding. My third disk, a brand new Kodak double-sided, double-density disk, finally did work properly.

TMM can handle a budget for twelve months. To create a budget, first set your titles for the sixteen categories. Then select the month you will begin with. The program steps you through each category, prompting you to enter your budgeted amount for each item. Only whole dollar amounts can be entered.

When you enter budget or actual amounts, all amounts already stored for the month selected are set to zero. You can use another menu option to change a specific category entry, if you just need to alter a single stored value. Pressing RETURN at a prompt in this step leaves the value at zero. If this isn't what you had in mind, you'll have a chance to make changes after you complete the list of sixteen entries.

There's no provision for copying the budget entries for one month into the corresponding fields for another month. Instead, the numbers for all sixteen categories must be entered manually for each month. A copy/edit feature would really accelerate the budget setup process.

Of course, a budget doesn't do your finances any good unless you stick to it. **TMM** lets you enter actual expenditures and make comparisons with previously budgeted amounts. You can enter the actual amounts either by typing them in (along with category codes from 1 to 16), or by automatically transferring transactions entered in **TEC**.

You can't combine entry methods in a single session. If you select the **TEC** interface input method, then the menu items for manual entry of your actual amounts in **TMM** are rendered inoperative. This could be a problem if all of your expenditures aren't logged in **TEC**.

Now for the analysis. You can compare budgeted amounts with actual expenditures in either tabular or bar graph form. Two kinds of table presentations are available. The first shows budgeted and actual amounts for all sixteen categories, one screen per month. The second table shows data for twelve months, with one screen per category. I found the tabular comparisons a little confusing, although they do provide a complete display of your data.

As with almost any analysis of numeric data, a graph is a far better display mechanism than a table. The bar chart option in **TMM** plots dollars on the vertical axis and months on the horizontal axis, with a separate screen for each budget category. For every month, the budgeted amount is shown as a column of inverse letter *B*s, and the actual expenditures are shown in an adjacent column of inverse *A*s. Plot is scaled for whichever is larger, budget or actual.

Horizontal lines are drawn at quartile intervals (25%, 50%, 75%, 100%) and labeled with the dollar amount. The bar charts are effective and easy to read, despite the use of simple monochrome character graphics.

Don't be misled by the nifty illustration of multicolored bar charts in the Timeworks catalog; this display must belong to the Commodore version of **TMM**.

Some further numeric information, such as year-to-date totals and the average amounts per month, is shown with each bar chart. The charts are easily sent to your printer, although the form taken by the inverse video characters which make up the bars depends on your printer. On Epsons, for example, you'll see italic *A*s and *B*s, rather than the inverse letters.

Numerous checkbook and home budgeting programs are commercially available for Atari computers. These Timeworks programs are among the more useful ones in the low-price range.

TEC is actually a simple database system for recording a variety of transactions in a single checking account. The flexible cross-search capability makes this program especially useful for retrieving the specific records needed at tax time, for example.

TMM is handy for small budgeting applications. While the quality of data presentations is acceptable, the program is somewhat cumbersome. The ability to read data already entered into **TEC** can save a lot of typing. But the interface between these programs is hampered by the sixteen-category limit.

These Timeworks programs are well designed and easy to use, with excellent user guides. As with any software, their utility depends upon your own needs. Remember, you can still do a lot with a cheap pocket calculator.

Karl E. Wiegers is a Senior Research Chemist at Eastman Kodak Company in Rochester, New York. He's worked with mainframe and micro computers (Atari, IBM, Apple) for 14 years and has had a number of applications programs published.

CALL 312-382-5050 ATARI Software Sale CALL 312-382-5050

(T) Tape, (C) Cartridge, (D) Disk.

GAMES

Electronic Arts

A0684 DR. J & LARRY BIRD GO 1 ON 1 (D) $24.95
A0685 MOVIE MAKER (D) $23.95
A0686 SEVEN CITIES OF GOLD (D) $23.95
A0687 PINBALL CONSTRUCTION SET (D) $16.95
A0688 MUSIC CONSTRUCTION SET (D) $16.95
A0689 FINANCIAL COOKBOOK (D) $27.95
A0690 M.U.L.E. (D) $16.95
A0691 MURDER ON THE ZINDERNUEF (D) $16.95

Atari

A0544 STAR RAIDERS (C) $14.95
A0545 MISSILE COMMAND (C) $14.95
A0546 GALAXIAN (C) $14.95
A0547 DEFENDER (C) $14.95
A0548 DIG DUG (C) $16.95
A0549 DONKEY KONG (C) $16.95
A0555 PENGO (C) $16.95
A0556 MILLIPEDE (C) $16.95
A0557 JUNGLE HUNT (C) $16.95
A0559 MOON PATROL (C) $16.95
A0560 FINAL LEGACY (C) $16.95
A0561 FOOTBALL (C) $14.95
A0562 TENNIS (C) $14.95
A0563 TRACK & FIELD (C) $29.95
A0564 EASTERN FRONT (C) $14.95
A0566 ADVENTURE WRITER (D) $19.95
A0567 STAR VOYAGER (D) $14.95
A0568 SARGON II (D) $16.95
A0569 MS PACMAN (C) $16.95
A0570 DONKEY KONG JR (C) $16.95
A0571 POLE POSITION (D) $18.95

Broderbund

A0514 MASK OF THE SUN (D) $24.95
A0515 OPERATION WHIRLWIND (D) $24.95
A0516 SPELUNKER (D) $18.95
A0517 LODE RUNNER (D) $20.95
A0518 WHISTLERS BROTHER (D) $18.95
A0502 STEALTH (D) $18.95
A0670 CHAMPIONSHIP LODE RUNNER (D) $26.95
A0671 KARATEKA (D) $23.95

Activision

A0597 PAST FINDER (D) $20.95
A0598 SPACE SHUTTLE (D) $19.95
A0599 GHOSTBUSTERS (D) $18.95
A0665 HACKER (D) $19.95
A0666 MIND SHADOW (D) $18.95
A0667 MASTER OF THE LAMPS (D) $19.95
A0668 GR AMER CROSS CNTRY RD RACE (D) $18.95

Suncom

A0190 PARTY QUIZ (D) $14.95
A0193 GENL EDITION 2 (D) $14.95
A0194 GENL EDITION 3 (D) $14.95
A0195 SPORTS EDITION (D) $14.95
A0196 EDUCATION EDITION (D) $14.95
A0197 BIBLE EDITION (D) $14.95
A0198 ENTERTAINMENT EDITION (D) $14.95

Avalon Hill

A0573 TGIF (D) $16.95
A0574 FLYING ACE (D) $22.95
A0575 MIDWAY CAMPAIGN (T) $12.95
A0576 B-1 NUCLEAR BOMBER (T) $18.95
A0577 LEGIONNAIRE (D) $20.95
A0578 TAC (D) $26.95
A0579 MARKET FORCES (D) $14.95
A0580 PANZER JAGD (D) $20.95
A0604 FREE TRADER (D) $19.95
A0605 EMPIRE OF THE OVERMIND (D) $26.95
A0606 QUEST OF THE SPACE BEAGLE (D) $22.95
A0607 CLEAR FOR ACTION (D) $26.95
A0608 PARIS IN DANGER (D) $26.95
A0609 GULF STRIKE (D) $22.95
A0504 GALAXY (D) $16.95
A0505 ANDROMEDA CONQUEST (D) $16.95
A0506 COMPUTER STOCKS & BONDS (D) $18.95

Add $3.00 for shipping handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail!

VISA MASTER CARD C.O.D.

No C.O.D. to Canada, APO-FPO.

EDUCATION

Xerox

A0412 STICKYBEAR BOP (D) $19.95
A0413 STICKYBEAR NUMBERS (D) $19.95
A0414 STICKYBEAR BKST BOUNCE (D) $19.95
A0415 STICKYBEAR OPPOSITES (D) $19.95
A0416 STICKYBEAR ABC (D) $19.95
A0417 STICKYBEAR SHAPES (D) $19.95

BUSINESS

A0201 ATARI WRITER (C) $39.95
A0203 VISICALC (D) $29.95
A0204 HOME FILING MANAGER (D) $19.95
A0206 FILEWRITER (D) $20.95
A0207 REPORT WRITER (D) $20.95
A0208 MENU WRITER (D) $19.95
A0209 FAMILY FINANCE (D) $19.95
A0210 HOME INTEGRATOR (D) $19.95
A0211 SMALL BUS INVENTORY (D) $11.95
A0212 SALESMAN'S EXPENSES (D) $11.95
A0214 RETAIL INVOICE (D) $11.95
A0215 TIMEWISE (D) $14.95
A0216 PEACHTREE G/L (D) $49.00
A0217 PEACHTREE A/R (D) $49.00
A0218 PEACHTREE A/P (D) $49.00
A0717 SYN CALC (D) $32.95
A0718 SYN CALC TEMPLATES (D) $14.95
A0672 APPT PLNR/WKLY SCHEDULE (D) $12.95
A0673 ACCOUNTS RECEIVABLE (D) $11.95
A0674 ACCOUNTS PAYABLE (D) $11.95

Synapse

A0534 ENCOUNTER (D) $14.95
A0535 BLUE MAX 2001 (D) $18.95
A0536 QUASIMODO/AIR SUPPORT (D) $16.95
A0537 NEW YORK CITY/ELECTRICIAN (D) $16.95
A0538 RAINBOW WALKER/COUNTDOWN (D) $16.95
A0539 FORT APOCALYPSE (D) $20.95
A0540 BLUE MAX (D) $20.95
A0715 MIND WHEEL (D) $25.95
A0716 ESSEX (D) $25.95

Epyx

A0520 JUMPMAN (D) $15.95
A0521 DRAGON RIDERS OF PERN (D) $18.95
A0522 SUMMER OLY GAMES (D) $24.95
A0523 PITSTOP II (D) $24.95
A0524 BALL BLAZER (D) $24.95
A0525 RESCUE ON FRACTULUS (D) $24.95
A0693 KORONIS RIFT (D) $24.95
A0692 THE EIDOLON (D) $24.95

Strategic Simulations, Inc.

A0601 SHOOTOUT AT OK GALAXY (D) $17.95
A0602 DNIEPER RIVER LINE (D) $24.95
A0603 SPACE COWBOY (D) $18.95
A0526 KNIGHTS OF THE DESERT (D) $24.95
A0527 FIELD OF FIRE (D) $24.95
A0528 FORTRESS (D) $22.95
A0529 COSMIC BALANCE (D) $24.95
A0530 IMPERIUM GALATUM (D) $24.95
A0531 RAILS WEST (D) $24.95
A0532 TIGERS IN THE SNOW (D) $24.95
A0533 50 MISSION CRUSH (D) $24.95
A0590 BROADSIDES (D) $24.95
A0591 COMPUTER QUARTERBACK (D) $24.95
A0592 COMPUTER AMBUSH (D) $34.95
A0593 COMPUTER BASEBALL (D) $24.95
A0712 COLONIAL CONQUEST (D) $24.95
A0713 COMBAT LEADER (D) $23.95
A0714 KAMPFGRUPPE (D) $34.95

Atari

A0420 ATARI MUSIC I (D) $19.95
A0421 ATARI MUSIC II (D) $19.95
A0422 INTRO PROG I (T) $14.95
A0423 INTRO PROG II (T) $14.95
A0424 INTRO PROG III (T) $14.95
A0425 ATARI LAB STARTER (C) $44.95
A0426 ATARI LAB LIGHT MOD (C) $33.95
A0428 SKYWRITER (C) $16.95
A0429 CONVERSATIONAL FRENCH (T) $16.95
A0430 CONVERSATIONAL SPANISH (T) $16.95
A0431 MY FIRST ALPHABET (D) $16.95
A0432 SPEED READING (T) $19.95
A0433 TYPO ATTACK (C) $16.95
A0435 VERBAL MODULE SAT (D) $29.95
A0436 SAT SAMPLE PRETEST (D) $17.95
A0437 MATH MODULE SAT (D) $29.95
A0438 TOUCH TYPING (T) $14.95
A0439 JUGGLES RAINBOW (D) $16.95
A0440 JUGGLES HOUSE (D) $16.95
A0442 TOUCH TABLET/SOFTWARE $49.00
A0443 PAINT (D) $19.95
A0315 PILOT/TURTLE GRAPHICS (C) $29.95
A0316 LOGO (C) $39.95
A0318 ASSEMBLER/EDITOR (C) $19.95
A0319 MACRO ASSEMBLER (C) $19.95

Spinnaker

A0444 LINKING LOGIC (C) $16.95
A0445 DANCE FANTASY (C) $16.95
A0446 MEMORY MANOR (C) $16.95
A0447 LOGIC LEVELS (C) $16.95
A0448 KINDERCOMP (D) $16.95
A0449 FACEMAKER (D) $16.95
A0450 KIDS ON KEYS (D) $16.95
A0451 GRANDMAS HOUSE (D) $16.95
A0452 KIDWRITER (D) $16.95
A0453 FRACTION FEVER (D) $18.95
A0454 IN SEARCH AMAZ THING (D) $22.95
A0455 TRAINS (D) $18.95
A0456 ALPHABET ZOO (D) $16.95
A0457 AEROBICS (D) $22.95
A0710 DELTA DRAWING (C) $16.95
A0711 ADVENTURE CREATOR (C) $16.95

American Educational Computer

A0459 VOCABULARY WORD BLDR (D) $16.95
A0460 GRAMMAR WRK USE SKILLS (D) $16.95
A0461 WORLD GEOGRAPHY FACTS (D) $16.95
A0462 SPANISH VOCAB SKILLS (D) $16.95
A0463 FRENCH VOCAB SKILLS (D) $16.95
A0464 WORLD HISTORY FACTS (D) $16.95
A0465 US HISTORY FACTS (D) $16.95
A0466 US GEOGRAPHY FACTS (D) $16.95
A0467 US GOVERNMENT FACTS (D) $16.95
A0468 A PLUS LEARN TO READ (D) $24.95
A0470 A PLUS READING COMPREHENSION (D) $24.95
A0471 COMPUTER LEARNING PAD $37.95
A0418 BIOLOGY FACTS (D) $16.95
A0493 ELEM SCIENCE 3 & 4 (D) $16.95
A0494 ELEM SCIENCE 5 & 6 (D) $16.95
A0495 ELEM SCIENCE 7 & 8 (D) $16.95

DLM

A0680 SPELLING WIZ (D) $19.95
A0681 ALIEN ADDITION (D) $19.95
A0682 METEOR MULTIPLICATION (D) $19.95
A0683 ALLIGATOR MIX (D) $19.95

Artworx

A0738 LINKWORD LANGUAGE-SPANISH (D) $17.95
A0739 LINKWORD LANG-FRENCH (D) $17.95
A0740 LINKWORD LANG-GERMAN (D) $17.95
A0741 LINKWORD LANG-ITALIAN (D) $17.95
A0663 MONKEYMATH (D) $15.95
A0664 MONKEY NEWS (D) $15.95

ORDER TODAY

COMPUTER DIRECT

We Love Our Customers

22292 N. Pepper Rd., Barrington, Ill. 60010

312/382-5050 to order

152K *Lowest Price In The USA!* 152K

ATARI®

Computer System Sale

• Students • Word Processing • Home • Business

LOOK AT ALL YOU GET FOR ONLY **$399**
LIMITED QUANTITIES **SYSTEM PRICE**

	LIST PRICE	INDIVIDUAL SALE PRICE
① Atari 130XE 152K Computer	$249.00	$134.95
② Atari 1050 127K Disk Drive	299.00	179.95
③ Atari 1027 Letter Quality 20 CPS Printer	299.00	179.95
Atari Writer Word Processer	59.95	49.95
Atari BASIC Tutorial Manual	16.95	12.95
TOTALS	**$923.90**	**$547.75**

All connecting cables & T.V. interface included.
☆ Monitors sold separetly.

SAVE OVER $100
All 5 ONLY
$399.00
SYSTEM SALE PRICE

CALL FOR 1027 PRINTER REPLACEMENT OPTIONS

Other Accessories	List	Sale	
☆ 12" Hi Resolution Green Screen Monitor	$199.00	$79.95	Add $9.95 for Connection Cables
☆ 13" Hi Resolution Color Monitor	$399.00	$159.95	Add $10 for UPS

15 DAY FREE TRIAL. We give you 15 days to try out this ATARI COMPUTER SYSTEM!! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!! **90 DAY IMMEDIATE REPLACEMENT WARRANTY**. If any of the ATARI COMPUTER SYSTEM equipment or programs fail due to faulty workmanship or material within 90 days of purchase we will replace it IMMEDIATELY with no service charge!!

Best Prices • Over 1000 Programs and 500 Accessories Available • Best Service • One Day Express Mail • Programming Knowledge • Technical Support

Add $25.00 for shipping and handling!!
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only. Add $10 more if C.O.D., add $25 if Air Mail.

COMPUTER DIRECT

We Love Our Customers
22292 N. Pepper Rd., Barrington, Ill. 60010
312/382-5050 to order

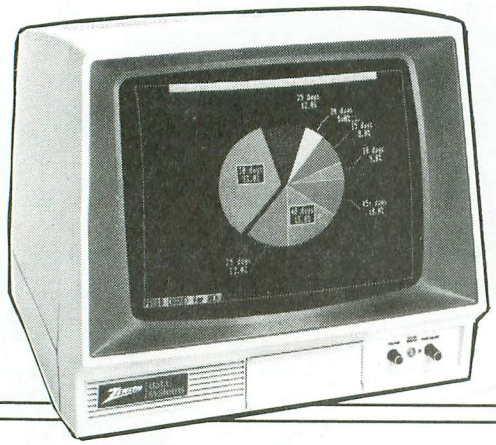
CIRCLE #125 ON READER SERVICE CARD

Computer Cleaners

Your Choice Reg. $19.95 **Sale $9.95**

½ PRICE SALE

Your Choice Reg. $19.95 **Sale $9.95**



TV/MONITOR SCREEN RESTORER & CLEANING KIT

Sale $9.95

Reduce eye fatigue by increasing the clarity of your TV or monitor screen. This kit contains a hard wax formulation to cover surface imperfections on TV screens and monitors. This restores maximum optical clarity, making what you see more distinct. Plus high absorbency cloths and an auto static spray cleaner allows you to clean your screen on a regular basis to keep your screen looking better than new. (This is a must for those who watch monitors or TVs for extended lengths of time.) List $19.95. **Sale $9.95.**

DISK DRIVE CLEANER

Reg. $19.95.
Sale $9.95.

- 60% of all drive downtime is directly related to poorly maintained drives.
- Drives should be cleaned each week regardless os use.
- Drives are sensitive to smoke, dust & all micro patricles.
- Systematic operator performed maintenance is the best way of ensuring error free use of your computer system.

This unique twin slot jacket design gives twice as many "wet-dry" cleanings per rotation as other leading brands. Non abrasive, 100% lint free, random fiber cleaners capture dust, smoke particles and disk oxide build up which insures you against disk and data loss from dirty disk drive heads just like you must clean your albums and tape players you must clean your disk drive heads to keep your disk drive working well. (24 cleanings per kit.) List $19.95. **Sale $9.95.**

ANTI-STATIC KEYBOARD CLEANER

Sale $9.95

Now you can clean your computer keyboard fast, efficiently, and safely. The keyboard cleaning solution is exclusively formulated to remove skin oils, dust, and dirt that can destroy your equipment. Plus this non residue solution with anto-static properties will not build up like ordinary household cleaners so you can clean as much as yo like without worry. Plus the lint free, high absorbency, non-abrasive cloths will not scratch or mar your equipment as the pick up dirt and grime in a matter of seconds. List $19.95. **Sale $9.95.**

Add $17.50 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $35.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO.

COMPUTER DIRECT

We Love Our Customers

22292 N. Pepper Rd., Barrington, Ill. 60010

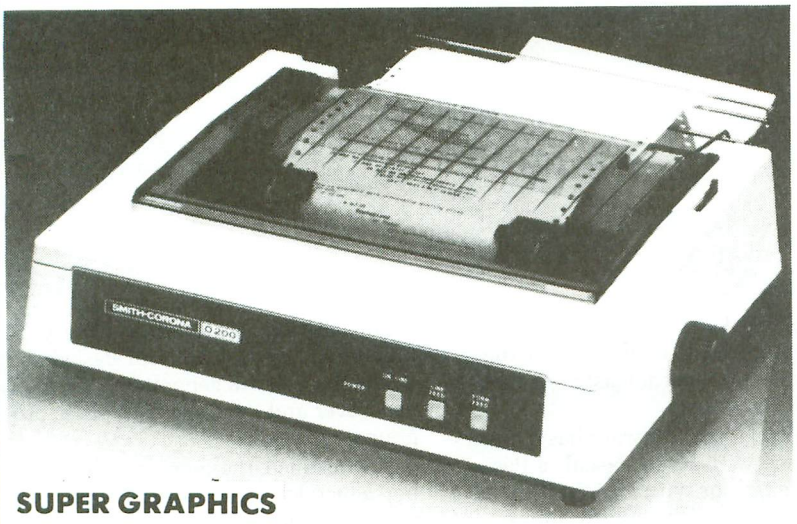
312/382-5050 to order

Famous Smith Corona National Brand

10" PRINTER SALE

Below Wholesale Cost Prices!!!

- **ONE YEAR IMMEDIATE REPLACEMENT WARRANTY**
- **Speed: 120 or 160 characters per second**
- **Friction Feed/Tractor Feed — Standard**
- **80 character print line at 10 CPI**
- **1 Line Buffer, 2K Buffer on 120/160 CPS Plus LQM**
- **Six pitches**
- **Graphics capability**
- **Centronics compatible parallel interface**
- **Features Bidirectional Print, Shortline Seek, Vertical And Horizontal Tabs**



SUPER GRAPHICS

This is a sample of our emphasized near-letter-quality print.

italic print. There is standard data processing quality print

Check these features & prices

120 CPS 10'' Printer

List $429.00 **SALE $159**

120 CPS + Letter Quality Mode 10'' Printer

List $449.00 **SALE $179**

160 CPS + Letter Quality Mode 10'' Printer

List $499.00 **SALE $199**

SPECIFICATIONS

(IBM — Commodore)

Size/Weight
Height 5.04" Width 16.7"
Depth 13.4" Weight 18.7 lbs.
Internal Char. Coding
ASCII Plus ISO
Print Buffer Size
120 CPS: 132 Bytes (1 line)
120/160 CPS Plus LQM: 2K
No. of Char. in Char. Set
96 ASCII Plus International
Graphics Capability
Standard 60, 72, 120 DPI
Horizontal 72 DPI Vertical
Pitch
10, 12, 16.7, 5, 6, 8.3, Proportional Spacing
Printing Method
Impact Dot Matrix

Char. Matrix Size
9H x 9V (Standard) to 10H x 9V
(Emphasized & Elongate)
Printing Features
Bi-directional, Short line seeking, Vertical Tabs, Horizontal Tabs
Forms Type
Fanfold, Cut Sheet, Roll (optional)
Max Paper Width
11"
Feeding Method
Friction Feed Std.; Tractor Feed Std.
Ribbon
Cassette — Fabric inked ribbon
Ribbon Life
4 million characters

(Apple — Atari — Etc.)

Interfaces
Parallel 8 bit Centronics compatible
120/160 CPS Plus NLQ: RS232 Serial inc.
Character Mode
10 x 8 Emphasized; 9 x 8 Standard; 10 x 8 Elongated; 9 x 8 Super/Sub Script (1 pass)
Character Set
96 ASCII
11 x 7 International Char.
Line Spacing
6/8/12/72/144 LPI
Character Spacing
10 cpi normal; 5 cpi elongated normal; 12 cpi compressed; 6 cpi elongated compressed; 16.7 cpi condensed; 8.3 cpi elongated condensed; 5.12.5 cpi elongated proportional

Cartridge Ribbon — List $19.95. **Sale $12.95.**

Interfaces

IBM $89.00 **Apple $59.00** **Atari $59.00** **Commodore $39.00**

Add $14.50 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $29.00 for CANADA, PUERTO RICO, HAWAII, ALASKA. APO-FPO orders. Canadian orders must be in U.S. dollars.

WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days delivery. 2 to 7 days for phone orders. 1 day express mail!

VISA — MASTERCARD — C.O.D. No C.O.D. to Canada or APO-FPO

COMPUTER DIRECT

We Love Our Customers

22292 N. Pepper Rd., Barrington, Ill. 60010

312/382-5050 to order

CIRCLE #125 ON READER SERVICE CARD

KYAN PASCAL
KYAN SOFTWARE
1850 Union St., #183
San Francisco, CA 94123
(415) 775-2923
64K Disk (Enhanced)
800XL/130XE $69.95

by Bob Curtin

I started fooling around with computers late in life, and, like most Atari computer owners, BASIC was the first language I learned. Actually, it's still the language I do most of my programming in. BASIC is easy to learn, easy to use, interactive and powerful. But it's also slow, bulky and notorious for propagating hideously bad programming habits.

Times have changed; "bulky" doesn't really mean much anymore. Most home computers are being sold with at least 64K, and many sport much more.

"Slow," however, is another story. It means a lot when we're messing around with graphics animation, interrupt routines, and floating point math. Yes, I know: assembly language routines pair up nicely with BASIC to speed things up, but they're a colossal chore to implement, even with a nifty programmer's aid like Tom Hudson's **BOFFO!** (issue 24). Besides, if you're anything like me, you find assembly language painful to use in any kind of volume.

What then? The C programming language? An excellent choice if you own anything but an 8-bit Atari. C is fast, compact and, compared to assembler, easy to learn. In my opinion, it's the best all-around language for many reasons, not the least of which is its portability (computerese for "being able to run on many different machines"). Unfortunately, since the demise of APX and the withdrawal of the C/65 compiler from OSS, there are no C compilers on the market for us 8-bitters, at least none that I know of.

Fortunately, Kyan Software has come to the rescue with **Kyan Pascal**, a Pascal compiler/editor with which, I'll tell you right up front, I'm highly impressed. Its similarity to C is obvious. Pascal is a "structured" language, one of the "ALGOL family" of languages (including PL/I, C, and Ada).

Just what is "structured programming"? Well, to tell you the truth, I couldn't dig up a hard and fast definition. At this point, it's more of a disciplined approach to computer programming than the "shooting from the hip" style most of us use in BASIC.

The very structure of Pascal requires a more systematic approach to the art of programming. The result is code of infinitely greater clarity and efficiency. In fact, Pascal was developed to teach systematic programming techniques.

Who cares, you say? So what if my programs look like spaghetti, as long as the results are the same, you ask?

How about speed? Pascal is a compiled language, running an average of five times faster than BASIC—and, at times, a lot faster than that.

Assembly routines can be written as unaltered 6502 source code and included in the Pascal source code. The compiler simply assembles the assembly language source code as is when found.

I'm already building a library of procedures written in both Pascal and assembler that are general enough to be used over and over again in other programs.

For instance, there's no random number generator function in Pascal, so I wrote a short assembly language subroutine to get a random number from $D20A, check to see that it resides within limitations (which I can quickly alter from program to program), and then return the value to a specified variable. I define the subroutine as a procedure to be called at any time in the program.

The combination of Pascal and assembler is incredibly powerful indeed, retaining the ease of use of a high-level language while giving the programmer the speed and control of pure assembler when it's needed. Pascal is at its best in long programs where the time involved in using pure assembly language would be prohibitive.

Another advantage of Pascal is its portability. As long as one refrains from packing code with machine-specific procedures and functions, Pascal will compile on different machines with virtually no change in the source code. This is really handy for those of us who're constantly programming a number of different computers.

Enough of the virtues of Pascal; what about the virtues of **Kyan Pascal**? When Charlie Bachand gave me the package to review, I was most interested to see if

Kyan Pascal was a full-bore version of the language, or just a limited subset. It's definitely the former. **Kyan Pascal** is a powerful implementation, including a few graphics procedures (Plot, Drawto, Position, etc.) just to make things easier in taking advantage of the Atari graphics capabilities.

The package comes with a disk containing the compiler, an editor, a library of functions, a number of procedures and a few programs—with both the source and the compiled object code.

The documentation is in the form of a well-written and comprehensive "tutorial" manual with, amazingly, a full index. It includes a few fliers, one of which is an apology to 800 owners, explaining that this package won't run on their machines. The tutorial manual, combined with a good Pascal language manual, will get you programming quickly and fairly painlessly.

A minimum of 64K is needed to run the compiler at this time, although Kyan did say they expect to offer a 48K version in about six months.

Incidentally, you might want to know that I dived into this review using a borrowed 800XL (I own an ancient 800). The editor and compiler worked like a charm. Shortly after I started, I received another package in the mail which required a 130XE, so I broke down and bought one.

The difference is incredible. It takes 45 seconds to load in the Pascal compiler from disk, but only 2.5 seconds from the 130XE ramdisk. What a savings in programming time! Debugging a compiled program is time-consuming, with switching from the editor to compiler to DOS and back again. Using the ramdisk makes the task a lot easier.

The editor supplied with the package is straightforward and natural, retaining all of the cursor controls you're used to in BASIC, plus a bunch more available to you through an assortment of CTRL/keystroke combinations. In addition, full search and replace, cut and paste (block operations), file insertion and file manipulation are yours at the touch of a button or two. DOS is available on command, and you can reload the editor or compiler from DOS using the DOS binary load (option L).

Kyan Pascal uses Atari DOS 2.5, allowing 1½ density if you own a 1050 drive. It includes the ramdisk option (for 130XE owners), to speed up compilation and assembly time considerably.

The compiler takes its sweet time to do the work, but, in all fairness, I have nothing to compare it to except a compiler written for the Epson QX-10 (whose MPU is much faster than the 6502). In one respect, this slowness is a blessing.

The compiler has full error-tracking routines and flags syntax errors, some garbage errors and a half-dozen assembler errors. There's also a list of run-time error messages when you fire up that new Pascal program you just compiled.

Unfortunately, there are a few messages for errors not listed in the manual, but, with some experience, you'll get used to those in a hurry. One really nice feature is the fact that—even though the actual compilation stops—error tracking continues and the results are displayed. This allows you to repair all of those errors before trying another compilation. Error listings, as well as the assembly code, can be directed to the screen, the printer, or both.

The Kyan compiler has a built-in assembler and linking/chaining functions—a glorious blessing. If you've ever been saddled with a compiler that only spits out source code to be run through an additional step of assembly (and linking) you can appreciate this feature.

All in all, I liked the package very much, and I get the feeling that I'd appreciate it even more if I were an expert Pascal programmer. But I do recognize software at a reasonable price; you won't be disappointed here.

Kyan Pascal isn't a half-baked version of Pascal. It's a serious, no-nonsense, full-blown Pascal compiler, and I recommend it without reservation.

Bob Curtin is a machinist who got into computing in 1982, when he bought an Atari 800. He uses it for writing, programming and telecommunications. He prefers more cerebral computer games and hopes to write "definitive" computer games.

Software Discounters of America

For Orders Only—1-800-225-7638
PA Orders—1-800-223-7784
Customer Service 412-361-5291

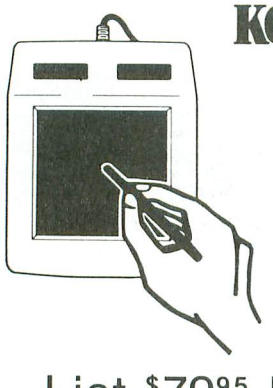
Open Saturday

- Free shipping on orders over $100 in continental USA
- No surcharge for VISA/MasterCard
- Your card is not charged until we ship

ACCESS
Beach Head (D) $21
ACCOLADE
Sundog 520ST $25
ACTIVISION
Gary Kitchen's Gamemaker (D) ..... Call
Great American Cross Country Road Race (D) . $16
Hacker (D) $16
Hacker 520ST $29
Master of the Lamps(D) . $16
Mindshadow (D) $16
Space Shuttle(D) $16
AMERICAN EDUCATIONAL COMPUTER
Biology (D) $16
French (D) $16
Grammar (D) $16
Science: Grades 3/4 (D) . $16
Science: Grades 5/6 (D) . $16
Science: Grades 7/8 (D) . $16
Spanish (D) $16
U.S. Geography (D) $16
U.S. History (D) $16
World Geography (D) ... $16
ARTWORX
Bridge 4.0 (D) $16
Monkeymath (D) $16
Strip Poker (D) $21
Female Data Disk 1 ..... $16
Male Data Disk 2 ..... $16
Female Data Disk 3 ..... $16
AVALON HILL
Computer Title Bout (D) . $19
Gulf Strike (D) $19
Jupiter Mission 1999 (D) $33
Legionnaire (D) $19
Maxwell Manor (D) $17
Panzer Jagd (D) $19
Quest of the Space Beagle (D) $23
Superbowl Sunday (D) .. Call
T.A.C. (D) $26
Telenguard (D) $19
BATTERIES INCLUDED
B-Graph (D) $25
Home Pak (D) $33
Paperclip (D) $39
Degas 520ST $27
BRODERBUND
Bank St. Writer (D) $33
Championship Loderunner (D) $19
Karateka (D) $19
Print Shop (D) $26
Print Shop Graphics Library #1 (D) $16
Print Shop Graphics Library #2 (D) $16
Print Shop Graphics Library #3 (D) $16
PS Paper Refill $14
Spelunker (D) $19
Stealth (D) $19
Whistler's Brothers (D) . $19
CBS
Astro Grover (R) $7
Big Bird's Funhouse (R) . $7
Big Bird's Spc. Delivery (R) $7
Dr. Seuss Puzzler (D) $7
Ernie's Magic Shapes (R) . $7
Math Mileage (R) $7
Sesame St. Letter Go Round (R) $7
Timebound (R) $7
Webster Word Game (D) . $7
CONTINENTAL/ARRAYS
Home Accountant (D) .. $44
Tax Advantage (D) $44
Book of Adventure Games II $16

DATAMOST
Aztec Challenge (D) $16
My Chess II (D) $19
DATASOFT
Alternate Reality (D) $25
Elevator Action (D) $19
The Goonies $19
Zorro (D) $19
DAVIDSON
Math Blaster (D) $33
Spell It (D) $33
Word Attack (D) $33
DESIGNWARE
All Titles Available ..... Call
ELECTRONIC ARTS
Archon (D)
Archon II (D)
Hard Hat Mack (D)
Movie Maker (D)
Murder Zinderneuf (D)
Music Const. Set (D)
Phone Book 520ST $39
HAYDEN
Sargon III (D) $33
HBJ
Computer SAT (D) $49
INFOCOM
Cutthroats (D) $23
Deadline (D) $29
Enchanter (D) $23
Hitchhiker's Guide to the Galaxy (D) $23
Infidel (D) $25
Planetfall (D) $23
Seastalker (D) $23
Sorcerer (D) $25
Spellbreaker (D) $29
Starcross (D) $29
Suspect (D) $25
Suspended (D) $29
Wishbringer (D) $23
Witness (D) $23

MICROPROSE
Acro Jet: Advanced Flight Simulator (D) ... Call
Crusade in Europe (D) ... Call
Decision in Desert (D) ... Call
F-15 Strike Eagle (D) ..... Call
Kennedy Approach (D) ... Call
Nato Commander (D) ..... Call
Silent Service: The Submarine Simulation (D) ..... Call
MINDSCAPE
Bank St. Music Writer (D) $26
Crossword Magic(D) ... Call
Halley Project—A Mission in Our Solar System (D) $26
Tinka's Mazes (D) $19
Tink's Adventure (D) $19
Tink's Subt. Fair (D) $19
Tonk in the Land of Buddy-Bots (D) $19
Personal Pascal 520ST . Call
Writer's Tool w/ Spell Checker (R) $44
ORIGIN
Ultima III (D) $34
Ultima III 520ST ..... Call
PENGUIN
Crimson Crown 520ST .. $25
Oo-Topos 520ST $25
Transylvania 520ST $25
PRECISION
Superscript (D) $49
PROFESSIONAL SOFTWARE
Fleet System 2 WP w/70,000 Word Spell Checker (D) $39
Trivia Fever (D) $19
PRYORITY
Forbidden Quest 520ST . $25
Gateway 520ST ..... Call
Snooper Troops 1 or 2 (D) $17
Story Machine (R) $15
Trains (D) $17
SPRINGBOARD
Early Games (D) $23
Fraction Factory (D) $19
Make A Match (D) $19
Music Maestro (D) $23
Piece of Cake Math (D) .. $23
SSI
Battalion Commander (D) . $25
Battle of Antietam (D) .. $33
Breakthrough in the Ardennes (D) $37
Broadsides (D) $25
Carrier Force (D) $37
Colonial Conquest(D) ... $25
Combat Leader (D) $25
Computer Ambush (D) ... $37
Computer Baseball(D) ... $25
Computer QB (D) $25
Field of Fire (D) $25
Gemstone Warrior (D) ... $23
Imperium Galactum (D) ... $25
Kampfgruppe (D) $37
Questron (D) $33
NAM (D) ..... Call
Panzer Grenadier (D) ... $25
Rails West (D) $25
Reforger '88 (D) $37
Six-Gun Shootout (D) ... $25
U.S.A.A.F. (D) $37
War In Russia (D) $49
SUBLOGIC
Flight Simulator II (D) ... $32
SYNAPSE
Essex (D) $25
Lode Runner's Rescue (D) $19
Mindwheel (D) $25
Syn-Calc (D) $33
Syn-Chron (D) $26
Syn-Comm (D) $26
Syn-File (D) $33
Syn-Stock (D) $26
Syn-Trend (D) $26
TELLARIUM
Amazon 520ST ..... Call
Fahrenheit 451 520ST .. Call
Nine Princes in Amber 520ST ..... Call
Perry Mason: Case of Mandarin Murder 520ST Call
TRONIX
S.A.M. (D) $39
XLENT
Megafont (D) $19
Page Designer (D) $23
Rubber Stamp (D) $23
Typesetter (D) $26
Typesetter 520ST ..... Call
ACCESSORIES
Ape Face Printer Interface $47
Astra Disk Drive ..... Call
Bonus Disks ..... Cheap
Compuserve Starter Kit (5 hrs.) $19
Digital Devices U-Print A $54
Disk Case (Holds 50) $9
Dows Jones News Retrieval Membership Kit (5 hrs.) $16
ICD P:R: Connection ... Call
Kraft Joysticks ..... Call
MPP1000E Modem ... Cheap
MPP1150 Printer Int. ... $47
Microprint Printer Int. ... $37
Sakata 13" Color Monitor $149
Wico Joysticks ..... Call



KOALAPAD™ TOUCH TABLET.

- Video drawing tool.
- Comes with Koala Painter® —full feature computer graphics program.
- Use the KoalaPad for cursor control, graphics, or custom keyboard.
- Educational, entertainment, business software available.
- Atari® on Disk.

Koala Technologies Corporation

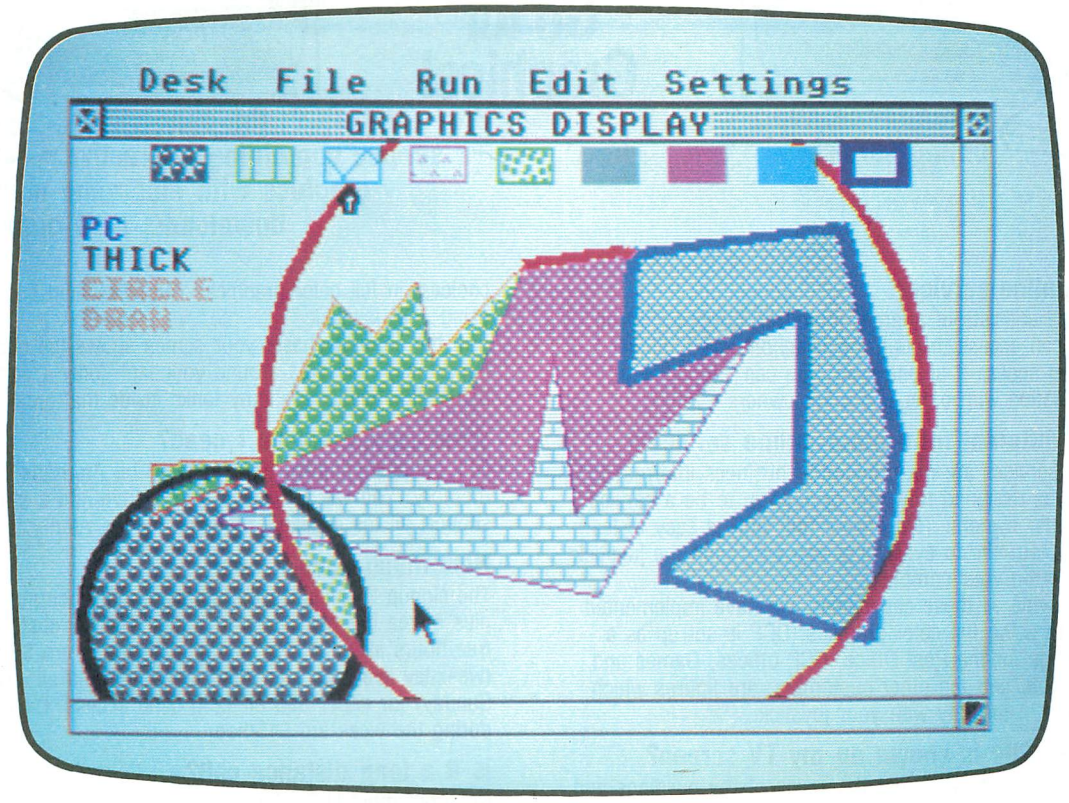
List $79.95 **Madness Price $29.95**

Sold to the first 135 customers

One-on-One (D)
Pinball Construction Set (D)
Realm of Impossibility (D)
Seven Cities of Gold (D)
Prices too low to advertise!! ..... Call
EPYX
Ballblazer (D) $25
Gateway Apshai (R) $9
Jumpman Jr. (R) $9
Koronis Rift (D) $25
Pitstop (R) $9
Pitstop II (D) $25
Puzzle Panic (D) $9
Rescue on Fractalus (D) $25
Temple of Apshai Trilogy (D) $25
The Eidolon (D) $25
FIRST STAR
Spy vs. Spy (D) $19
GAMESTAR
Baseball (D) $16
Football (D) $16
On Track Racing (D) $16
HABA SYSTEMS
Check Minder 520ST ... Call
Haba Word 520ST ..... Call
Hippo C Compiler 520ST . $47
Zork I (D) $23
Zork II or III (D) $25
* All titles in stock for 520 ST—**Call for prices**
LJK
Data Perfect (D) $39
Letter Perfect (D) $39
Spell Perfect (D) $29
*All three titles $89
LEARNING COMPANY
Bumble Games (D) $25
Colorasaurus (D) $19
Magic Spells (D) $23
Reader Rabbit (D) $23
Word Spinner (D) $23
MARK OF THE UNICORN
Hex 520ST ..... Call
PC Intercomm 520ST ... Call
The Final Word 520ST .. Call
MICHTRON
Flip Side 520ST $25
M-Disk 520ST $25
Mudpies 520ST $25
Soft Spool 520ST $25
Time Bandit 520ST $33
MICROLEAGUE
Baseball (D) $25
General Manager (D) ... $25
1984 Team Data Disk (D) $16
1985 Team Data Disk (D) $16
MIRAGE CONCEPTS
Atari ST Tool Box Vol. 1 ..... Call
Forth 520ST ..... Call
H & D Base ..... Call
MISC.
Diskey (D) $33
Gorf (R) $7
Hard Hat Mack (D) $9
Hulk (D) $21
Hole In One Golf (D) $9
Miner 2049'er (R) $7
Monster Maze (R) $7
Omnitrend Universe $59
Popeye (R) $9
Spiderman (D) $21
Star Warrior (D) $7
Ultra-Disassembler (D) .. $33
Wizard of Wor (D) $7
OSS
Action (R) $49
Action Tool Kit (D) $19
Basic XE (R) $49
Basic XL (R) $39
Basic XL Tool Kit (D) ... $19
DOS XL (D) $19
MAC 65 (R) $49
MAC 65 Tool Kit (D) $19
Personal Disk Kit 520ST ..... Call
REGENT
Regent Word 520ST $35
Regent Spell 520ST $35
SCARBOROUGH
Mastertype (D) $21
Net Worth (D) $44
SIERRA ON LINE
Dark Crystal (D) $25
Homeword (D) $33
Kings Quest II 520ST ... $33
Ultima I (D) $23
Ultima II (D) $37
Ultima II 520ST $39
Winnie the Pooh 520ST . $19
SIMON & SCHUSTER
N.Y. Times Crossword Puzzles Vol. 1 or 2 (D) . $16
Spy Hunter (R) $29
SPINNAKER
Adventure Creator (R) ... $15
Aerobics (D) $21
Alphabet Zoo (R) $15
Delta Drawing (R) $15
Facemaker (R) $15
Fraction Fever (R) $15
Homework Helper 520ST . Call
Kids on Keys (R) $17
Kindercomp (R) $17
Kung Fu-Exploding Fist 520ST ..... Call
Math Busters (D) $17

P.O. BOX 278—DEPT. AN—WILDWOOD, PA 15091

***Ordering and Terms:** Orders with cashier check or money order shipped immediately. Personal/company checks, allow 3 weeks clearance. No C.O.D.'s. **Shipping: Continental U.S.A.**—Orders under $100 add $3; **free shipping on orders over $100.** PA residents add 6% sales tax. **AK, HI, FPO-APO**—add $5 on all orders. Sorry—no International orders. Defective merchandise will be replaced with same merchandise. Other returns subject to a 15% restocking charge—NO CREDITS! Return must have authorization number (412) 361-5291. Prices subject to change without notice. **Modem Owners:** Save A Fortune! Join Our Discount Disk Club! call for details.



Doodler

A simple LOGO drawing program

by Sol Guber

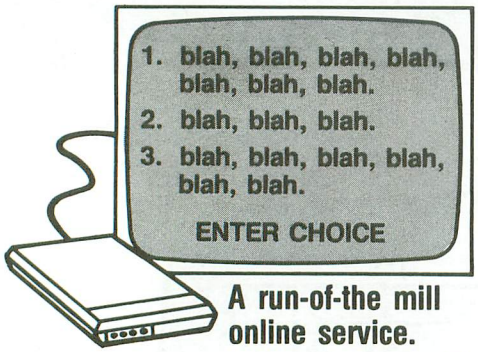
Having just gotten a 520ST, the first thing that my daughter wanted was a doodle program. It was no use explaining to her that all I had was LOGO and that BASIC was soon to be on its way. She still wanted some sort of program, so that she could put colors and shapes on the screen. Since she possesses a shrill voice and a long attention span, plus a feeling that her father can do everything (an illusion that I'm having a hard time breaking her of), I had to write **Doodler**, using the lovable mouse (which turned out to be a real pain).

This program was written in LOGO and runs very, very slowly. Also, it's impossible to have more than one graphics window in this version of LOGO. Therefore, the graphics choices at any one time are limited, and **Doodler** doesn't show off the 520ST nearly as well as is possible.

Using it.

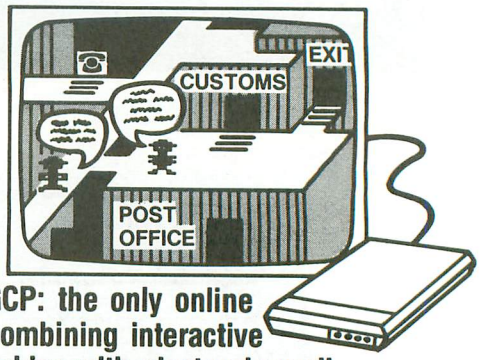
Let me describe **Doodler**. Wherever the mouse moves, a line will follow. The system monitors the mouse very closely. Mouse movement has the highest priority, so that if the mouse moves fast, then the system won't finish the line until the mouse stops. Thus, for very fine movements, the mouse has to be moved extremely slowly—or the ALT (Alternate) ARROW keys must be used. The system was set up so that, when the ALT keys are held down and the arrows are pressed, the cursor moves about ten steps in the direction of the arrow.

(continued on next page)



A run-of-the mill online service.

MODEM OWNERS: Tired of Text? Go for Graphics!



GCP: the only online service combining interactive graphics with electronic mail, downloads, AND games!

What is an online service?

An online service consists of a large, multiuser computer which your computer can access through the telephone lines. Since many people can access it at one time, you can interact and exchange information with other computer owners.

How is GCP different from other online services?

All other online services are out-growths of business information services. GCP was designed from the ground up to be a service for home computer owners. This means that GCP is easy to learn and fun to use. You can do everything you want with the joystick and function keys on your computer.

In order to provide all these capabilities, we have implemented the entire system using full color graphics. GCP is set up as a City, with buildings for the Post Office, GCP offices, Games and other services. You, and the other customers, are figures which you move around in the City with your joystick.

You mean the City is shown on my TV screen?

Yes, indeed. Not only the City, but the inside of the buildings and the games are shown on your screen in full color graphics. Additionally, the other customers are shown on your screen as they move around the City and buildings.

Isn't it slow downloading the graphics?

No, because we do not download the graphics. All the pictures of the City, buildings and games are supplied on disk. When you go from one building to another, the graphics are accessed at disk drive speeds.

Do I need special software?

Yes. very special. But don't worry, we provide it with your signup.

Can I download public domain programs?

Yes, GCP has a public domain archive in its Post Office with about a Megabyte of Atari programs you can download.

What games do you have?

At the moment, we have BioWar, CyberTank and CyberShip. Lords of Space is under development and may be done by the time you read this. All the games are played online against other customers, so you are matching wits with humans from all over the country.

BioWar is a multi-player adaptation of Conway's game of Life. Each player has a cell colony which he tries to expand, often at the expense of the other players, while contending with the problems of under- and over-population.

CyberTank and CyberShip are tactical design and combat games set on the CyberWorld, an artificial battleground for cybernetic machines. You design your own tank or ship and battle it out with up to 15 other players on a scrolling map.

Do the games use graphics also?

Extensively. For example, in CyberTank, when you design your tank, the hull is shown on the screen, as are all the equipment choices, in full color graphics. You select and place the items by using your joystick.

In the combat phase, your screen shows the status of your tank, the 1 mile area around your tank (only a part of the larger battlefield), and any enemy tanks inside that area.

What equipment do I need?

GCP supports any member of the 8 bit Atari line with 48K of memory. You will also need a disk drive and a modem. We support all the available modems for the Atari.

How much does it cost?

The signup kit includes the software and documentation you need, plus 5 free hours at standard rates. This kit costs $30. After the free hours are used up, the standard rates are $6 per hour (weekday evenings after 6pm local time and all day Saturday and Sunday) for either 300 or 1200 baud access. Daytime hours during the week are $15/hour.

Is it a long distance call?

Not from most U.S. cities. GCP is accessed through Tymnet, a national data service with over 500 locations in the U.S. The Tymnet charges are included in the standard rates, so you don't have to worry about add-on charges.

How do I sign up?

Just fill out the information requested below and send it to us with your credit card number or check/money order. For faster response, call us at:

(717) 848-2660 (VOICE)

and give us your logon name, password and credit card number over the phone.

Logon name (letters/numbers only)
[hint: keep it short]

Choices:

1st _____

2nd _____

3rd _____

Password (must be 6-10 chars):

_____

GCP GAMES COMPUTERS PLAY, INC.

112 East Market Street, York, PA 17401
717-848-2660 (VOICE)

CIRCLE #112 ON READER SERVICE CARD

There are many options which can be done in combination with the mouse. Since even a two-button mouse is very cumbersome, there are several selected keys used to make choices. The various things that the mouse can do are: draw in different colored lines, draw in different line thicknesses, make circles, and fill in various shapes with any of eight different patterns or colors.

Having played with the drop-down menus on the ST, I felt that the easiest method of seeing what was active was to vary the fonts and colors used to write the various information. I also used the first letter of the word as a help in remembering the command. Thus *PC* stands for Pen Color, and every time *P* is pressed, the color of the letters *PC* will change.

Something similar is done for pen thickness, at the command Thick. When the *T* is pressed, the word *THICK* on the left side of the screen is changed from faint characters to the normal characters, to the bold characters. These stand for the three different thicknesses of the pen.

The details.

Let me go into detail on how **Doodler** is run. After it's typed in, the word *START* initializes the program. There will be four words seen on the left side of the screen. They will be written in different colors and different font styles. On the top of the screen will be nine boxes, filled with various patterns and colors. The ninth box will be empty, and there'll be an arrow under the third box.

To start, move the cursor to any spot on the screen and press *D*. The *DRAW* on the side will become darker. When you move the mouse and stop, a line will be drawn from the spot where you were to the spot where the arrow's pointing. As you continue to move the mouse, lines will continue to be drawn.

This is a very coarse movement. To have the mouse move in finer increments, hold down the ALT key and press any of the arrow keys in the keypad. To stop the system from drawing, or to move without drawing, press *D* again, and the bold *DRAW* will disappear.

At any time except when Circle is active, you can press *F* to have the system fill in that shape with the color or pattern indicated by the arrow at the top of the screen. To change a color, press any number and the arrow will move. When *9* is pressed, the boxes will be randomly filled with different patterns and colors. It is recommended that you shut off "draw" (press *D*) before you use the fill function. This will allow you to move the cursor where you wish without leaving an unwanted line.

You can press *P* at any time, and the pen color will become the color of PC. As you continue to press *P*, the colors will go through all of those on the color palette. Pressing *T* will change the thickness of the word *THICK* and of the line that's drawn.

To use the circle function, place the cursor where you wish the circle's center to be, then press *C*. The word *CIRCLE* will become highlighted. Now place the cursor where you want the outside of the circle drawn and hold down the left mouse button. When the drawing is complete, turn off the CIRCLE function by holding down the right button.

Describing LOGO programs is difficult. They're usually so simple, logical and straightforward that it seems dull. They seldom have tricky algorithms, and things usually move slowly. However, it's good to see how some of these procedures were written.

LOGO is divided into three kinds of words. Those that start with a double quote (") are the names of variables. Those that start with a colon (:) are the values in the variables. Finally, words that start with a blank are action words. They do something, either return a value or make something happen.

The first word is *WHERE__AT*. It uses a word called *MOUSE*. This returns a list containing the position of the mouse—as if the buttons had been pressed. It tells if the mouse is in the window. The first thing is to make the variable "A equal to the list. Make R__BUTTON either true or false, depending on whether or not the right button has been pressed. The same is done for the left button. Finally WHERE__AT returns a list containing the X- and Y-coordinates of the mouse.

ETCH is the procedure that does the majority of the work. First, it makes OLDSPOT equal to WHERE__AT. Then it checks to see if a key has been pressed (whether "KEYP is true or false). If it's true, then "K is the value of that key. Make "KEYP false and go to CHANGE. Next, the turtle is told to go to position OLDSPOT—drawing a line to that spot, if the pen is down. ETCH calls the word again.

CHANGE is a series of IF tests, to determine what key has been pressed. If it's a number, then perform ARROW using that number, and return (STOP). If the letter was a *D*, then DRAW__IT. If it was a *"P*, then CHANGE__PEN. If it was a *"C*, then CIRCLE__IT and DRAW__CIRCLE. If it was a *"T*, then CHANGE__THICK. Finally, if it was an *"F*, then move slightly and perform a FILL. CHECK the pen and return.

START initiates everything. It calls all of the functions to put the boxes and the words on the screen. It then calls ETCH.

DRAW__CIRCLE is one of the action words. First, it makes the center where the cursor is found, lifts up the pen and makes "GFILL "FALSE, so that the circle won't be automatically filled in when it's drawn. Next comes a unique feature of this LOGO, called *LABEL*. You're allowed to have loops in this LOGO by going to a label.

Next it makes "A WHERE__AT, and checks the right mouse button. If this has been pressed, it makes the circle with a list of the center and uses CAL__R to calculate the radius. It then makes CENTER the present location. Then the word *CIRCLE* on the left side of the screen is changed, and the function is exited.

CAL__R determines the distance between two points, using the Pythagorean theorem. This is the radius of the circle that will be drawn.

CHANGE__PEN changes the color of the line being used to draw. The value is incremented and checked to see if it's equal to *16*. Only sixteen colors can be used. The old word *"PC* is erased with ERASE__IT, and a new word "PC" is written with WRITE__IT.

The function SETLINE is given a list containing the color and width of the line. The first number sets the style for the line. You can draw with dotted lines, banded lines, or even patterns that you make up yourself. The second number in the list for SETLINE is the thickness, and the final number is the color for the line.

CHANGE__THICK uses several new words. The first is *SETTEXT*. There are six different font types in the system. These include normal, bold, faint, italics, underlined, inverse and shadowed. Any and all of these options can be used simultaneously.

The number used for the SETTEXT option determines which of these various options are to be used. ERASE__IT is then performed, to erase the word *THICK*. PENT and PENTH are incremented and checked, to see if they are within the limits.

Depending on the value of PENT, TXT (the font to be used in writing *THICK*) is modified. WRITE__IT puts a new *THICK* on the screen, depending on the proper font. SETLINE is changed, so that the new line corresponds to the proper thickness.

The next five words are used to make the boxes on the top of the screen and fills them with the various colors and patterns. ARROW uses the number :A to determine where to place the arrow.

First, it must make certain that the position of the arrow is not 0. Then, it will erase the old arrow, using the function ERASE_IT, and calculate where the old position was. Next, the turtle will be moved to the new position, where it will insert a new arrow, in pen color 1.

The turtle is then moved back to OLDSPOT, and the pen color is returned to the old pen color. If the spot was 9, then the function MAKE_TOP is used to make a new series of patterns and colors. If it was not 9, then a new fill pattern is determined.

The variable "FILL_LIST contains a list of all the patterns. Each pattern contains three parts: the style, the pattern number and the color. The first three numbers in FILL_LIST are for the first box, the next three for the second box, and so on. SETFILL takes a PIECE of this list and uses it for the next fill operation.

FILL_BOX determines where the lower corner of the box is to be. BOX draws a box and needs a list containing the lower corner coordinate, as well as the length and width. If the "GFILL is true, then the box is filled with the last pattern from SETFILL.

MAKE_LIST first makes "GFILL true. Then it makes FILL_LIST empty. It calls PATFILL five times and puts nine more values in FILL_LIST, using random colors.

PATFILL first makes "A a random number, either 2 or 3. It then adds this value to the end of the FILL_LIST. If the number is 2, then increase FILL_LIST with a random number less than 24; otherwise, increase FILL_LIST using a random number less than 12.

Finally, pick a random color for the fill. SETFILL expects three numbers for the fill pattern. IF the number is 1, then a solid color is used. If the style number is 2, then any of fifty-two different crosshatch patterns are used. If the number is 3, then other plaid colors are used. Thus, thirty-six different patterns can be used for the fill operation.

MAKE_TOP first calls MAKE_LIST, then makes "A 1. It repeats eight times, a SETFILL, then FILL_BOX, and finally increments "A. It then makes "GFILL false, draws another box at spot 9 and moves the arrow to spot 3. It makes "GFILL true and does CHECK.

CIRCLE_IT and DRAW_IT toggle the styles for the words *CIRCLE* and *DRAW*. Each time the function is called, the style of color changes, to show whether or not the function is being done.

CHECK toggles the pen up or down, depending if "TDRAW is true or false. This is a general function used to see if the system is in the draw mode.

ERASE_IT is the general erase routine. It expects :A to contain a list of the coordinates where the turtle is to be moved to. :C is to contain the word to be erased. The 520ST uses bit-mapped graphics. This means that the letters are drawn on the screen and ORed to do so correctly. Thus, if you try to erase something by writing blanks to that spot, it will do nothing. If you write another letter over the first, all you get is the combination of the two letters.

To erase the letters completely, you must write over them using the same letters with the background color. First the pen is up, and the pen color is 0 (background). The turtle moves to the proper position, and the pen is put down. The text is set to 1 (bold) and the word is written (erased) on the screen.

WRITE_IT expects :A to contain the position, :B to contain the font style and :C to contain the word. It's very similar to ERASE_IT in action. It also moves everything back to the original positions and resets the font style. Finally, it calls CHECK to see if the pen is up or down.

Is **Doodler** a usable game? Yes and no. It's awkward and slow. It's cumbersome to use, because the cursor is read only when it's not moving fast. The good part of the program is that it's readily expandable. All you need to do is make a slight modification in CHANGE, so that more keys are recognized.

Writing the functions along the side is very easy, and implementing them is also simple.

Can an eight-year-old understand this program? Yes. Will it make pretty pictures? Yes, it will—extremely easily. Will this game show off LOGO and the ST? I certainly hope so. It makes lots of patterns and shapes on the screen. The fill is very fast, and there are so many patterns and colors available that the machine's a real joy to use.

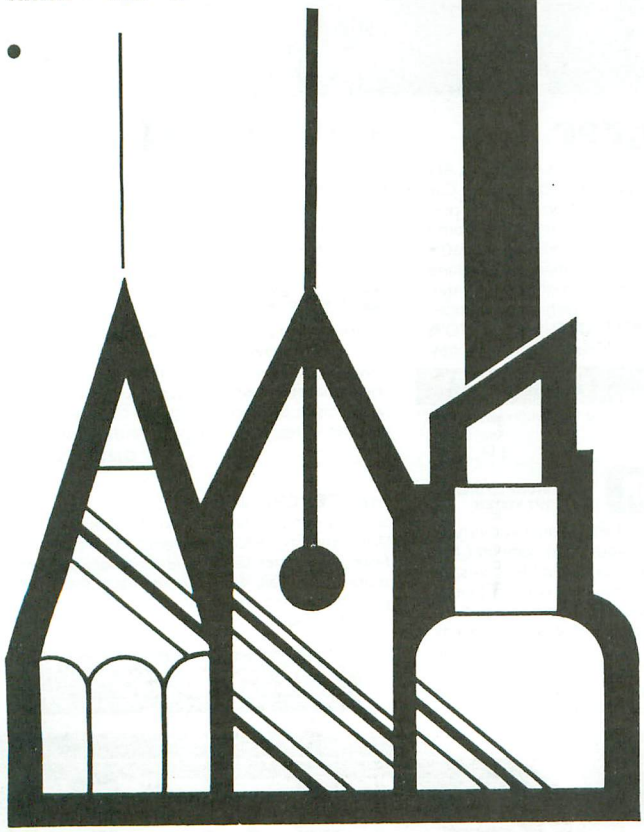
Sol Guber is a Chemical Engineer with a petroleum servicing company in the Midwest. His seven-year-old daughter corrects both his articles and games—now that her writing skills have improved. He's been programming the Atari 800 for four years and now has an 520 ST, which his daughter lets him use once in a while.

(Listings start on next page)

Listing 1.
LOGO listing.

```
TO WHERE_AT
MAKE "A MOUSE
MAKE "R_BUTTON ITEM 4 :A
MAKE "L_BUTTON ITEM 3 :A
OUTPUT SE ITEM 1 :A ITEM 2 :A
END

TO ETCH
MAKE "OLDSPOT WHERE_AT
IF KEYP [MAKE "K RC MAKE "KEYP "FALSE!
 CHANGE] []
SETPOS :OLDSPOT
ETCH
END

TO CHANGE
IF NUMBERP :K [ARROW :K STOP] []
IF :K = "D [DRAW_IT STOP] []
IF :K = "P [CHANGE_PEN STOP] []
IF :K = "C [CIRCLE_IT DRAW_CIRCLE STO!
P] []
IF :K = "T [CHANGE_THICK STOP] []
IF :K = "F [PU FD 2 FILL BK 2 CHECK S!
TOP]
END

TO START
CS HT
MAKE "PENTH -1
MAKE "PENT 0
MAKE "PENC 5
CHANGE_THICK CHANGE_PEN
MAKE "TCIRCLE "TRUE
MAKE "TDRAW "TRUE
DRAW_IT CIRCLE_IT
MAKE_TOP
ETCH
END

TO DRAW_CIRCLE
MAKE "CENTER WHERE_AT
PU MAKE "GFILL "FALSE
LABEL "DC
MAKE "A WHERE_AT
IF :L_BUTTON [CIRCLE SE :CENTER (CAL_!
R :CENTER :A) MAKE "CENTER WHERE_AT] !
[]
IF :R_BUTTON [CIRCLE_IT MAKE "GFILL "!
TRUE STOP] []
GO "DC
END

TO CAL_R :A :B
MAKE "X (ITEM 1 :A) - (ITEM 1 :B)
MAKE "X :X * :X
MAKE "Y (ITEM 2 :A) - (ITEM 2 :B)
MAKE "X :X + (:Y * :Y)
OUTPUT SQRT :X
END

TO CHANGE_PEN
MAKE "PENC :PENC + 1
IF :PENC = 16 [MAKE "PENC 1] []
ERASE_IT [-150 60] [PC]
WRITE_IT [-150 60] 0 [PC]
SETLINE SE (SE 1 :PENTH) (:PENC)
END

TO CHANGE_THICK
SETTEXT :TXT
ERASE_IT [-150 50] [THICK]
MAKE "PENT :PENT + 1
MAKE "PENTH :PENTH + 2
IF :PENTH > 5 [MAKE "PENTH 1] []
IF :PENT = 4 [MAKE "PENT 1] []
IF :PENT = 1 [MAKE "TXT 3] [IF :PENT !
= 2 [MAKE "TXT 0] [MAKE "TXT 1]]
WRITE_IT [-150 50] :TXT [THICK]
SETLINE SE (SE 1 :PENTH) (:PENC)
END

TO ARROW :A
IF :A = 0 [STOP] []
ERASE_IT SE ((:OLDA * 30) - 150) 70 C!
HAR 1
PU SETPOS SE ((:A * 30) - 150) 70
PD SETPC 1 TT CHAR 1
PU SETPOS :OLDSPOT
CHECK SETPC :PENC
MAKE "OLDA :A
IF :A = 9 [MAKE_TOP STOP] []
SETFILL PIECE (:A * 3 - 2) (:A * 3) :!
FILL_LIST
END

TO FILL_BOX :A
MAKE "B ((:A * 30) - 155)
BOX SE :B [79 20 13]
END

TO MAKE_LIST
MAKE "GFILL "TRUE
MAKE "FILL_LIST []
REPEAT 5 [PATFILL]
REPEAT 3 [MAKE "FILL_LIST SE :FILL_LI!
ST [1 1] MAKE "FILL_LIST SE :FILL_LIS!
T (2 + RANDOM 14)]
END

TO PATFILL
MAKE "A 2 + RANDOM 2
MAKE "FILL_LIST SE :FILL_LIST :A
IF :A = 2 [MAKE "FILL_LIST SE :FILL_L!
IST (1 + RANDOM 24)] [MAKE "FILL_LIST!
 SE :FILL_LIST (1 + RANDOM 11)]
MAKE "FILL_LIST SE :FILL_LIST (2 + RA!
NDOM 14)
END

TO MAKE_TOP
MAKE_LIST
MAKE "A 1
REPEAT 8 [SETFILL PIECE (:A * 3 - 2) !
(:A * 3) :FILL_LIST FILL_BOX :A MAKE !
"A :A + 1]
MAKE "GFILL "FALSE
FILL_BOX 9
ARROW 3
MAKE "GFILL "TRUE
CHECK
END

TO CIRCLE_IT
PU
IF :TCIRCLE [MAKE "TCIRCLE "FALSE MAK!
E "W 3] [MAKE "TCIRCLE "TRUE MAKE "W !
1]
ERASE_IT [-150 40] [CIRCLE]
WRITE_IT [-150 40] :W [CIRCLE]
END

TO DRAW_IT
IF :TDRAW [MAKE "TDRAW "FALSE MAKE "W!
 3] [MAKE "TDRAW "TRUE MAKE "W 1]
ERASE_IT [-150 30] [DRAW]
WRITE_IT [-150 30] :W [DRAW]
END

TO CHECK
IF :TDRAW [PD] [PU]
END
```

```
TO ERASE_IT :A :C
PU SETPC 0 SETPOS :A
PD SETTEXT 1 TT :C
END

TO WRITE_IT :A :B :C
PU SETPC :PENC SETPOS :A
PD SETTEXT :B TT :C PU
SETPOS :OLDSPOT SETTEXT 0
CHECK
END

MAKE "PENTH 1
MAKE "OLDSPSOT [24 64.934816]
MAKE "GFILL "TRUE
MAKE "Y -2.201183
MAKE "X 1028.845191
MAKE "W 1
MAKE "K "D
MAKE "C [82 -59.431762 FALSE FALSE TR!
UE]
MAKE "B 115
MAKE "FILL_LIST [2 3 12 2 18 15 3 6 1!
2 2 5 13 3 7 2 1 1 3 1 1 10 1 1 4]
MAKE "A [-91 26.414198 FALSE FALSE TR!
UE]
MAKE "CENTER [65 -17.609466]
MAKE "TRUE "FALSE
MAKE "KEYP "FALSE
MAKE "PENT 1
MAKE "PENC 11
MAKE "OLDSPOT [-91 26.414198]
MAKE "OLDA "4
MAKE "R_BUTTON "FALSE
MAKE "L_BUTTON "FALSE
MAKE "TCIRCLE "FALSE
MAKE "CTRUE "FALSE
MAKE "TDRAW "TRUE
MAKE "TXT 3
```



Digitize yourself

Digitize your picture with the perfect hardware/software combination: COMPUTER EYES and MAGNIPRINT II+

COMPUTER EYES, capture software and MAGNIPRINT II+
Only $114.95

COMPUTER EYES/MAGNIPRINT Camera System
A complete ready to run system for those without access to video equipment. This system includes Computer Eyes, Magniprint II+, a high quality B/W video camera, and a 10 ft coaxial cable with appropriate connectors. Only **$299.95**

COMPUTER EYES alone (with capture and display software only) **$99.95**

Computer Eyes lets you take any form of video input and save it as a high-resolution graphics screen. You can use a video camera, VCR, TV output, video disk, other computers, etc. Now you can capture your picture, your friends or any video image and show it on an Atari computer. Computer Eyes is an innovative slow scan device that connects between any standard video source and your Atari computer (see the review in A.N.A.L.O.G. magazine).

- Do a complete Hi-Res scan in under 6 seconds
- Unique multi-scan mode provides realistic grey scale images in 24 seconds, and up for more detail scans.
- Full one-year warranty on parts and labor
- Plugs into your Atari joystick ports and uses a standard video phono plug

Now anyone can create the kind of graphics seen in this ad. When Computer Eyes is combined with Magniprint II+, you get unique capabilities that no other system can offer.

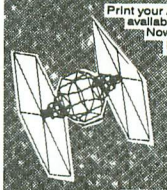
- Print your pictures in up to 19 different sizes, including poster sizes several feet across
- Print pictures with full shading for a level of realism even better than your TV screen
- Take your Computer Eyes images and modify them with your Koala Pad, Atari Touch Tablet, Micro Illustrator program, or Magniprint's special touch-up feature
- Works with Epson, Gemini, Panasonic, NEC, Citoh, and other compatible dot-matrix printers (850 interface or equivalent required)
- Supports all Magniprint II+ features

MAGNIPRINT II+

"Magniprint II is a versatile and powerful program which will meet most, if not all your print out needs."
ANALOG MAGAZINE

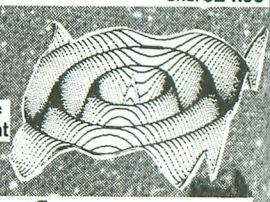
The plus is in the shading — now Magniprint lets you print the highest quality reproductions of your screens available anywhere. It lets you choose the shading you want for the different colors and shades on the screen. It allows you to adjust the shades while your picture is displayed on the screen, so what you see is what you get. Magniprint prints the shades on your standard printer and paper by using specially designed dot patterns, yielding better looking output than you get on your screen.

This new feature is perfect for printing pictures from Koala Pad, Touch Tablet, Micro Illustrator, and others. It will amaze you when you print pictures from Computer Eyes or Strip Poker.



Print your Atari® graphics like you've never seen before! Magniprint II is the most powerful print utility available for your computer! It is the only software that can accurately print GTIA modes with 16 shades. Now accepts pictures from PRINTSHOP*. Prints 19 different sizes from ½ page to giant wall sized posters. Prints across or down the page, regular or inverse, with a centering option. Works with printers compatible with NEC, CITOH, EPSON, & Gemini (850 interface or equivalent required). Accepts your own graphics creations or those created with Graphics Master*, Micropainter*, Fun With Art*, T SYN-Graph*, Atari Graphics Light Pen*, B/Graph*, Koala Pad*, Atari Touch Tablet*, Paint*, Strip Poker*, Graphics Magician*, and others • Includes many beautiful sample pictures • Does amazing graphics with your standard printer and paper • Special feature lets you modify pictures with your joystick • Add text to your pictures • Can convert a graphics mode 9 picture to a graphics mode 8 one (and vice versa) • Allows you to print the whole screen or any selected portion of the screen in multiple sizes • Includes HELP screen for easy use • Prints GTIA modes 9, 10, and 11 with 16 different shades. Posters print along continuous sheets of paper which are attached together when done.
40% Faster
ONLY $24.95

FREE: With Magniprint II-PRINTALL. Allows you to print your programs or files just as they appear on the screen. Prints all graphics symbols, even INVERSE and control characters.



Actually done with Magniprint

Digitize your voice with Parrot.

Tired of low-quality mechanical sounding voice output? Now you can make any Atari speak in your own voice. Tired of four tone sound? Now any Atari can play a whole orchestra complete with a singing choir. "The Parrot" digital sound synthesizer system lets you do all this and much more.

How it works — "The Parrot" system plugs into your joystick port and lets you record pure digital sound from your stereo, TV, microphone, or any other sound source. The special Parrot software lets you play back this high quality sound on any Atari system with no special hardware needed. It even lets you put this unbelievable sound right into your own programs, that will run on anyone's Atari. It also includes digital sequencer software that lets you turn your Atari into a synthesizer comparable to those costing thousands of dollars. Turn any natural sound into a musical instrument, or design your own custom sounds. Imagine playing a song with the sounds of a dog's bark, a chinese gong, a car's honk, your own voice, or anything your imagination can come up with. It turns your keyboard into an organ and lets you instantly switch between up to nine different digital sounds, each with three full octaves of notes. Recording time varies depending on available memory and quality level desired. You've got to hear it to believe it!!!!

THE PARROT digital input hardware and playback/synthesizer software with sample sounds and demos

All For Only **$39.95**

BOOK I + DISK: (The Original) Thoroughly explains the techniques used by advanced software pirates, and the copy protection methods used to stop them. It offers clear and understandable explanations sophisticated enough for software writers of any scale yet easy enough for a beginner just wanting to learn more about Atari® computers. A MUST READ FOR ALL ATARI® OWNERS.

BOOK INCLUDES: • Duplicate sectoring • Custom disk formatting • Creating "BAD" sectors • Hardware data keys • Legal protection like copyrights, trade secrets, patents • Protecting BASIC programs • Self-modifying Code • ROM + EPROM cartridges • Hidden serial numbers • Self-destructing programs • Freeware • Misassigned sectoring • Much, much more.

DISK INCLUDES: • Directory mover • VTOC scanner • Duplicate sector finder • Sector mover • Bad sector writer • Sector data displayer • Autorun builder • Other useful programs.

This comprehensive book and disk package should not be confused with low quality imitations offered elsewhere.

BOOK II + DISK II: Advanced Software Protection. This all new sequel starts where the highly acclaimed Book I leaves off. Book II is the most up-to-date resource available for the Atari® owner. Includes reviews and explanations of products such as: The Happy Enhancement,* The Impossible,* The Scanalyzer,* The Chip,* The Pill* and Super Pill* & many others.

Book II: Tells you specifically what they copy, what they won't, how they are used, and the details of how they work. Book II also includes such topics as: • Transmitting protected programs • Copying disks with more than 19 sectors/track. Includes the newest protection methods by companies like Synapse* AND Electronic Arts* • Data encryption • Phreaking methods • Program worms • Logic bombs • Bank-select cartridges • Random access codes • New trends in software law • Sample BASIC + Assembler programs • On-line security • And much more.

DISK II INCLUDES: • Automatic program protector • Custom format detector • Newest protection demos • Forced password appender • Data encrypter • And much more.

Book + Disk Packages only $24.95 each or Special Offer both for only $39.95

The Scanalyzer

NEW AND IMPROVED FEATURES!

AT LAST A UTILITY THAT DOES IT ALL! Scans & Analyses ALL Atari programs. Works on programs stored on: Disk, Cartridge, or directly from memory • Converts complex machine language into readable assembler • Transforms ANY Atari BASIC program into listable, modifyable BASIC • Changes a 4, 8, or 16K cartridge into a binary load file and source file that you can view and change using regular Atari assembler. • Clearly shows protection techniques such as: BAD SECTORS, BAD DATA MARKS, DUPLICATE SECTORS and FORCED CRC ERRORS. Even finds hidden directories.

NEW FEATURES: Now fully supports DOS 2.5 and includes one pass sector copier for 130XE owners

No other program can do all this! Complete with instructions on theory and use. **$29.95**

Impersonator

CARTRIDGE TO DISK COPY SYSTEM

CARTRIDGE TO DISK COPY SYSTEM Yes, for only $29.95, you can make working copies of all your Atari computer cartridges (16K or less). Our special package will let you save your cartridges to ordinary disk files. They will run exactly like the originals when used with the Impersonator. Each disk holds up to 12 cartridge programs. Now you can put all your real cartridges away for safe keeping and use the Impersonator for everything. YES, IT REALLY WORKS. The Impersonator does everything the high-priced cartridge back-up systems do...and more. **ONLY $29.95**

Letterman

Exciting, Educational, Fun. Letterman's like a computerized Hangman Game with multiple skill levels, hints on request, optional time limits, 400 built-in words, and the ability to add your own. Automatically tracks up to 9 players. Lively animation, colorful graphics, amusing sound effects. Second Prize Winner in the Atari Star Awards... **ONLY $24.95**

D•O•T•S

Learning the alphabet can be fun with DOTS, the electronic version of "follow-the-dot pictures". Joystick controlled follow-the-dot drawings unravel hidden pictures as you learn the alphabet, or learn to count. Built-in catalog of pictures, with musical accompaniment...or create your own pictures. Great for kids, fun to watch and listen to

ONLY $19.95

MASTERCRUNCH

Shrink down your programs to save disk space and memory. Will compress a BASIC program by 15% to 40%, so it loads faster, runs faster, takes up less memory and uses less disk space. Best of all, it is fully automatic, fast, and works on BASIC programs and binary load files (machine language programming).

ONLY $24.95

All for your Atari Computers. Disk drive and 48K required. Atari® is a registered trademark of Atari Corporation.
*Denotes products not related to Alpha Systems

TO ORDER:
Call 216-374-7469 to charge to MasterCard or VISA, or MAIL TO: Alpha Systems
4435 Maplepark Rd.
Stow, OH 44224

Send check or money order. Include $2.00 shp. & hdlg. ch.

Ohio residents add 5½% sales tax.

ALPHA SYSTEMS

Order any 3 programs and get FREE your choice
BONUS: ☐ Deluxe Space Games (3 games on a disk) or ☐ Disk Pak 1000 (Utility Package)

CIRCLE #129 ON READER SERVICE CARD

C-MANSHIP

Part 2.

by Clayton Walnum

Last month, we took a brief look at the way a simple C program is constructed. Along the way, we also learned a bit about the basic data types, and got an introduction to the C functions printf(), scanf(), and getchar(). Now that you've got all that mastered, it's time to learn how to handle strings. We'll also take a closer look at the powerful printf() and scanf() functions.

First, the typing.

Of course, you have a job to do first. . . namely, typing in Listing 1 and compiling it. Go to it! (If you need help, see the sidebar accompanying this article.)

Got it? When you RUN the program, you will be prompted to enter your first name. Type in your name, terminating the input with the SPACE BAR. You'll get a personal hello and be asked for your last name.

When you enter it, the program will print some important information (your name), after which it'll wait for you to press a key, to end the program. Gee, this is just like the old days when you first learned BASIC. For those of you who don't have your C compilers yet, a program run will look something like this:

```
Enter your first name: Clay
Hi, Clay! Enter your last name: Walnum
Your full name is Clay Walnum.
```

A look at the program.

Line 1 instructs the compiler to add the contents of the stdio.h file to our program.

Line 2 introduces you to the #define statement. The format of this statement is the word *#define*, followed by a symbolic name and the value you wish placed in the name. In our example, the symbolic name *TEXT* will contain the string constant *Your full name is*.

Since C doesn't provide the programmer with a special data type for strings, they're stored as an array of characters. The last character in this array will be the null character *\0*. You don't have to worry about supplying the null character, though. It's added automatically. Here's a graphic representation of how the string constant *TEXT* looks in memory:

Y	O	U	R		F	U	L	L		N	A	M	E		I	S	\0

Your C compiler contains a program known as the "preprocessor." When you compile a program, the preprocessor searches for any occurrence of items that were defined by the #define statement. Wherever it finds a match, it replaces the symbolic name with the value the name contains. In other words, in our program, every place the word *TEXT* appears,

Here are a few other examples of the #define:

```
# define ZERO 0
# define PI 3.14159
# define PLUS +
```

Why do we bother with the #define statement? Why not just use a regular variable? After all, aren't the statements *pi = 3.14* and *#define PI 3.14* really equivalent?

No! The difference has to do with the preprocessor we discussed earlier. By using #define, your program will actually run a bit faster than if you used a variable name. The reason is that each time a variable is encountered in the program, its storage location must be "peeked" to get its value. This is called "run-time substitution." With #define, the substitution is accomplished during compilation (compile-time substitution), so that when the program is run, the values are already in place.

You're probably beginning to realize just how powerful the #define statement is. Listing 2 shows you an extreme use of #define. The program hardly looks like C anymore.

Notice that the constants defined in the #define statement are written in upper case. This is standard practice in C, and makes it easy to distinguish your variables from your constants.

We can clean up Listing 2 by putting all those # define statements in a separate file called newc.h—then we delete them from the main program and substitute the statement # *include <newc.h>* in their place. When we compile the program, the file newc.h will be added to the main program.

Line 3 is our function name. Remember, all C programs must contain the function main().

Line 4 marks the beginning of the function.

Line 5 declares a variable of type character.

Line 6 should look a bit strange to you. Here we're declaring two arrays of type character. Remember, C doesn't have a data type for strings. We declare character arrays whenever we need a string variable.

In this example, we're declaring two character arrays, one to hold a first name and another to hold a last name. We've set aside 20 bytes for each, which means the longest name each array can hold is 19 letters (you need 1 byte for the null character).

The syntax for declaring an array is the name of the array followed by the number of bytes (within brackets) you wish reserved for it. As with any declaration, you may declare as many arrays as you like on one line, as long as they're separated by commas. Remember, the line must end with a semi-colon.

Line 7 brings us back to familiar territory. Here we're printing our first prompt.

Line 8 shows a new use for the scanf() function. The conversion specification %*s* tells scanf() to expect the input of a string. The corresponding argument is the address where we wish the string stored.

Notice something a bit different about this pointer? Last month, when we were using this function to get integer values, each variable name was preceded by an ampersand, telling the compiler that we wanted the address of the variable *not* its value. There's no ampersand here, though, is there? That's because array names *are* pointers. The value of fname contains the address of the first byte of the array.

Line 9 leaves a blank line after the first prompt.

Line 10 prints our second prompt, but with an extra something special. If you take a close look at this line, you'll see that %*s* again. In the argument list, you'll also note the array fname. The %*s* works just like %*d*, except it tells printf() to substitute a string rather than an integer.

Line 11 utilizes scanf() again to allow input of the last name.

Line 12 prints another blank line.

Line 13 prints out our program's final message. Notice that there's no text in the control string—only conversion specifications and newline characters. So where's all the text coming from?

Look at the control string. The function is being instructed to print three strings, followed by two newlines. The strings that'll be substituted for the conversion specifications are in the argument list. The first string will be the string constant *TEXT*, which we defined at the beginning of the program. The second and third strings are the first and second names we previously input with scanf().

Line 14 and 15 wait for a key press.

Line 16 marks the end of the program.

Some fancy stuff.

Now that you know how to use printf() and scanf() in their most basic form, it's time to take a look at some of the tricks you can do with them.

All along, you've probably been wondering what the *f* in each of these function names stood for. Well, your wondering is over. It stands for "formatted." Both printf() and scanf() allow you to format input and output in various ways, as well as to do "type conversions." First, let's take a closer look at printf().

Last month, I gave you a list of conversion specifications that could be used with printf() and scanf(). The output of printf() can be edited by adding conversion specification modifiers. Here are some examples: *%3d, %03d, %-5d, %ld, %5.3f.*

The first of these sets the minimum field width to 3. If the number or string is smaller than the minimum length, the field will be padded with spaces. If the data to be printed is larger than or equal to the minimum length, it'll be printed normally.

In the second, the leading *0* in the conversion specification causes the field to be padded with *0*s rather than spaces.

The *-5* in the third causes the data to be left justified (-) in a minimum field length of 5.

Number four tells printf() that the matching data should be interpreted as *long* rather than *int*.

The final example shows how to edit floating point numbers. Here, the data will be printed in a minimum field of ten with three decimal places following the whole number portion.

Listing 3 is a program example that utilizes the above editing techniques. The output looks like this:

```
>5555<
>        5555<
>0000005555<
>5555<
>5555          <
>3.141590<
>3.142<
>      3.1416<
>3.1416        <
```

The arrows in the output mark the beginning and ending of each field. Notice that floating point numbers are automatically rounded when you limit the size of the decimal portion. Also, take a look at the way the variable *num* is defined in this listing. This form of the declaration allows you to assign a value to the variable immediately.

Listing 4 uses a similar technique to format strings. The output looks like this:

```
>strings<
>      strings<
>strings      <
>        strin<
>strin        <
```

The formatting features work much the same way with strings as with numerical data. As a matter of fact, the only real difference between the two is that, when you use the precision modifier (the number af-

ter the decimal point), it refers to the number of characters you wish printed, rather than the number of decimal places.

The scanf() function has a specification modifier you may add, to limit the number of characters entered into a string. The s may be preceded by a number which indicates the maximum size of the input. The input of the string terminates when the maximum size is reached, or when the first white-space character is received, whichever comes first. Listing 5 is an example of this form of editing. When you run this program, you'll not be allowed to enter more than five characters.

Type conversions.

One of C's handy—and dangerous—features is the ability to convert from one data type to another. I say "dangerous" because C doesn't check for type mismatch, and will allow you to do all sorts of strange things without complaining in the least. If you're not careful, this can lead to some hard-to-locate problems.

The printf() function won't complain either, as long as you have the right number of arguments. You can print your data out in just about any form you want. The trick is the proper use of the conversion specifications. If you have a decimal number you'd like printed in an octal form, just use the %o conversion specification. You can even convert between decimal and character.

Listing 6 is an example of using printf() for type conversions. The output of the program looks like this:

```
Decimal:     100
Hexadecimal: 64
Octal:       144
Character:   d
```

Odds and ends.

Well, that just about does it for this month. I did want to mention a couple of things before closing, though.

First of all, we've been taking advantage of a lot of I/O functions, such as printf() and scanf(). These are handy for general use, but as we learn more about C, we're liable to find that we outgrow them.

For instance, I'm sure you've wondered why we've had to terminate our inputs with a space rather than with a return. In your work with the 8-bit machines, you've grown accustomed to thinking of the return key as the way to finalize input. But what you may not realize is that the RETURN key on the ST supplies only a \r character (carriage return), and not a \n (newline or linefeed). And guess what a function like scanf() is looking for? Yep, the newline. Luckily for us, it'll accept *any* white space character to terminate input, thus our use of the SPACE BAR. The BACKSPACE key will also work. In the future, we'll get back to using the RETURN key, but there are a few things you must learn first.

Also, you should note that the C language really doesn't support I/O routines at all. The functions we've been using have been added for the programmer's convenience, and shouldn't be considered an integral part of the language.

Finally, I suggest that you run to your nearest bookstore and pick up a copy of *The C Programming Language* by Kernighan and Ritchie. It's published by Prentice-Hall, Inc., and is *the* book by which all the others are judged—in other words: our Bible.

Listing 1.

```
# include <stdio.h>
# define TEXT "Your full name is"

main()
{
char ch;
char fname [20], lname [20];

printf ( "Enter your first name: " );
scanf ( "%s", fname );
printf ( "\n\n" );
printf ( "Hi, %s!  Enter your last name: ", fname );
scanf ( "%s", lname );
printf ( "\n\n" );
printf ( "%s %s %s.\n\n", TEXT, fname, lname );
ch = getchar();
ch = getchar();
}
```

Listing 2.

```
# include <stdio.h>
# define START {
# define STOP }
# define INPUT scanf
# define OUTPUT printf
# define TIMES *
# define EQUALS =
# define SQUARES main()
# define WAIT ch=getchar()
int _isconio;

SQUARES
START
int num,ans;
char ch;
_isconio = 1;

OUTPUT ( "Enter a number: " );
INPUT ( "%d", &num );
OUTPUT ( "\n\n" );
ans EQUALS num TIMES num;
OUTPUT ( "The square of %d is %d.", num, ans );
WAIT;
WAIT;
STOP
```

Listing 3.

```
# include <stdio.h>

main()
{
```

COMPUTER CREATIONS

YOUR ATARI 520ST SOFTWARE SUPPORT CENTER

ACTIVISION
- Borrowed Time
- Hacker

ACADEMY
- Typing Tutor
- Word Invaders
- PCA
- Graphic Artist

ATARI
- DB Master
- Megaroids
- ST Writer
- Basic
- Neochrome

BATTERIES INCLUDED
- Degas
- Portfolio
- Homepak

DRAGON GROUP
- 4X Forth
- 4X Forth Accelerator

ELECTRIC SOFTWARE
- GST Compiler
- GST Macro Assembler
- GST Linker
- GEM Screen Editor

ELECTRONIC ARTS
- Marble Madness
- Financial Cookbook

HABBA
- Business Letters
- Wills
- Hippo 'C' Compiler
- Checkminder
- HABA Writer
- Phone Book
- Habadisk (10 Meg)

HIPPOPOTAMUS SOFTWARE
- Hippo Computer Almanac
- Hippo Jokes & Quotes
- Hippo ST Disk Utilities
- Hippo ST Ramdisk
- Hippospell
- Hipposimple
- Hippoart I
- Hippobackgammon
- Hippo - Lock
- Hippo Eprom Burner
- Hippoword

HIPPOPOTAMUS cont.
- Hippoconcept
- Hippopixel
- Hippoclean

HOLMES & DUCKWORTH
- Toolbox
- Forth
- HED Base

INFOCOM
- Zork I
- Zork II
- Zork III
- Cutthroats
- Deadline
- Enchanter
- Hitchhiker's Guide
- Seastalker
- Sorcerer
- Suspect
- Witness
- Wishbringer
- Infidel
- Mind Forever
- Planetfall
- Sorceror
- Spellbreaker
- Starcross
- Suspended

MARK OF THE UNICORN
- Mince
- PC Intercom
- Final Word
- Hex

METACOMCO
- Macroassembler

MICHTRON
- M-Disk
- Mudpies
- Soft Spool
- Flip Side
- Calendar
- Mi-Term
- Gold Runner
- Time Bandit

MICROPROSE
- Silent Service

MIGRAPH
- Easy Draw

MIRAGE
- Express (Word Processor)

OASIS
- Sundog

OMNITREND
- Universe II

OSS
- Personal Diskit
- Personal Pascal
- Personal Prolog

OXXI
- Whiz Word

PENGUIN
- Transylvania

PHILON
- Compiled Basic
- Fortran
- Pascal

PRYORITY
- Forbidden Quest

QMI
- ST Talk

QUEST
- Cash Trader

QUICKVIEW SYSTEMS
- Zoomracks

REGENT
- Regent Word
- Regent Spell
- Regent Base

SIERRA
- Kings Quest II
- Ultima II
- Winnie the Pooh

SPINNAKER/TELLARIUM
- Homework Helper - Math
- Homework Helper - Writing
- Farenheit 451
- Perry Mason
- Treasure Island

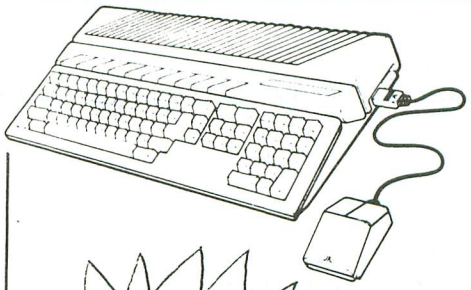
SST SYSTEMS
- Chat

SUBLOGIC
- Flight Simulator
- Jet

TDI SOFTWARE
- Modula - 2/ST
- Andra/ST
- UCSD Pascal

XLENT
- Typesetter
- Rubber Stamp

CALL FOR PRICES AND AVAILABILITY



CALL FOR FREE CATALOG

ATARI HARDWARE
CALL FOR PRICES

ATARI®
Power Without the Price™

ATARIWRITER PLUS™
Wordprocessing Program with Proofreader and Mail Merge.

THE LEARNING PHONE™
Access Software for PLATO®
Homelink™ Service

Programs Available Now!

BEST BUY ON SMALL QUANTITIES

COLORED DISKS AS LOW AS 79¢ EA. - FLOPPY DISKS AS LOW AS 59¢ EA.

Fully guaranteed. Includes sleeves and hub rings. DISKETTES (2 box minimum) 10 per box

5¼"	Black Generic Bulk		Colored Generic Bulk	
BULK	SS/DD	DS/DD	SS/DD	DS/DD
20-69	.74 ea.	.99 ea.	.89 ea.	1.09 ea.
70+	.59 ea.	.85 ea.	.79 ea.	.99 ea.

5¼"	Black Generic	Color. Generic	BASF
Boxes (10)	SS/DD	SS/DD	SS/DD
2 - 6	8.90	10.90	10.90
7+	7.40	9.90	9.90

3½"	Nashua
Box (5)	SS/DD
2-6	11.90
7+	11.29

To order call **TOLL FREE**

1-800-824-7506

ORDER LINE ONLY

No extra charge for MasterCard or Visa

COMPUTER CREATIONS, Inc.

P.O. BOX 493 - DAYTON, OHIO 45459

For information, order inquiries, or for Ohio orders (513) 435-6868

Order Lines Open 9:00 a.m. to 8:00 p.m. Mon.-Fri.; 10 a.m. to 4:00 p.m. Sat. (Eastern Standard Time). Minimum $15 per order. C.O.D. (add $3.00). Please specify computer system. Call toll free number to verify prices and availability of product. Prices and availability are subject to change without notice. We ship C.O.D. to Continental U.S. addresses only! Please include 4% shipping on all Hardware orders (min. $4.00). Software and accessories add $3.00 shipping and handling in Continental U.S. Actual freight will be charged outside U.S. to include Canada, Alaska, Hawaii, Puerto Rico and APO. Ohio residents add 6½% sales tax. Canadian orders add 5% shipping, (min. $5.00). All other foreign orders, please add 15% shipping, (min. $10). For immediate delivery send cashier's check, money order or direct bank transfers. Personal and company checks allow 3 weeks to clear. School purchase orders welcome. Due to our low prices, all sales are final. Canada and foreign orders must be in U.S. dollars. NO CREDITS. All defective returns must have a return authorization number. Please call (513) 435-6868 to obtain an RA# or your return will not be accepted for replacement or repair. **FOR YOUR PROTECTION WE CHECK FOR CREDIT CARD FRAUD.**

CIRCLE #132 ON READER SERVICE CARD

```
int num=5555;
char ch;

printf ( "%d\n", num );
printf ( "%10d\n", num );
printf ( "%010d\n", num );
printf ( "%3d\n", num );
printf ( "%-10d\n", num );
printf ( "%f\n", 3.14159 );
printf ( "%2.3f\n", 3.14159 );
printf ( "%10.4f\n", 3.14159 );
printf ( "%-10.4f\n", 3.14159 );
ch = getchar();
}
```

Listing 4.

```
# include <stdio.h>
# define TEXT "strings"

main()
{
char ch;

printf ( "%s\n", "strings" );
printf ( "%10s\n", TEXT );
printf ( "%-10s\n", "strings" );
printf ( "%10.5s\n", TEXT );
printf ( "%-10.5s\n", "strings" );
ch = getchar;
}
```

Listing 5.

```
# include <stdio.h>

main()
{
char ch;
char in[6];

printf ( "Enter some letters: ");
scanf ( "%5s", in );
printf ( "\n\n" );
printf ( "%s", in );
ch = getchar();
ch = getchar();
}
```

Listing 6.

```
# include <stdio.h>

main()
{
char ch;

printf ( "    Decimal: %d\n", 100 );
printf ( "Hexadecimal: %x\n", 100 );
printf ( "      Octal: %o\n", 100 );
printf ( "  Character: %c\n", 100 );
ch = getchar;
}
```

All the program listings in **C-manship** were written using the ST Developers Kit from Atari. Many of you who've recently received this package may be a little confused as to how to compile and run the programs (I know I was). For those nodding their heads in agreement, I've put together this quickie tutorial.

The first thing you must do is create the proper batch files for both the compiler and linker. LOAD your text editor and type the following exactly as it appears here:

```
cp68 %1.c %1.i
c068 %1.i %1.1 %1.2 %1.3 -f
rm %1.i
c168 %1.1 %1.2 %1.s
rm %1.1
rm %1.2
as68 -f -l -u %1.s
rm %1.s
wait.prg
```

When you're sure you've typed it correctly, SAVE it to your compiler disk under the name CC.BAT.

Now clear the previous text from memory and type in this batch file:

```
link68 [u] %1.68k=gemstart,%1,gemlib,libf,osbind
relmod %1 %1.tos
rm %1.68K
wait
```

Check your typing well, then SAVE it to your linker disk under the name LINK.BAT.

Now you're ready to compile any of the listings from **C-manship**. We'll use Listing 1 from this installment as an example.

Single-drive compilation.

(1) Use your text editor to type in Listing 1, then SAVE a copy under the name LIST1.C to both your compiler disk and a back-up disk.

(2) Place the compiler disk in your drive and double click the drive A icon.

(3) Double click the BATCH.TTP program, and enter *CC LIST1* into the parameter window, concluding the entry by pressing RETURN.

(4) After the compiler has finished, there should be a file named LIST1.0 on your compiler disk. Copy this file to your linker disk.

(5) Place the linker disk in your drive and double click the drive A icon.

(6) Double click the file BATCH.TTP, and enter *LINK LIST1* into the parameter window.

(7) When the linker has finished, the file LIST1.TOS should be on the disk. This is the executable version of the program. To RUN it, simply give it a double click.

Two-drive compilation.

(1) Use your text editor to type in Listing 1, then SAVE it to disk under the name LIST1.C.

(2) Place your compiler disk in drive A and your source disk (the one you saved the program to) in drive B.

(3) Double click the drive A icon.

(4) Double click the BATCH.TTP program, and enter *CC B:LIST1* into the parameter window, concluding the entry by pressing RETURN.

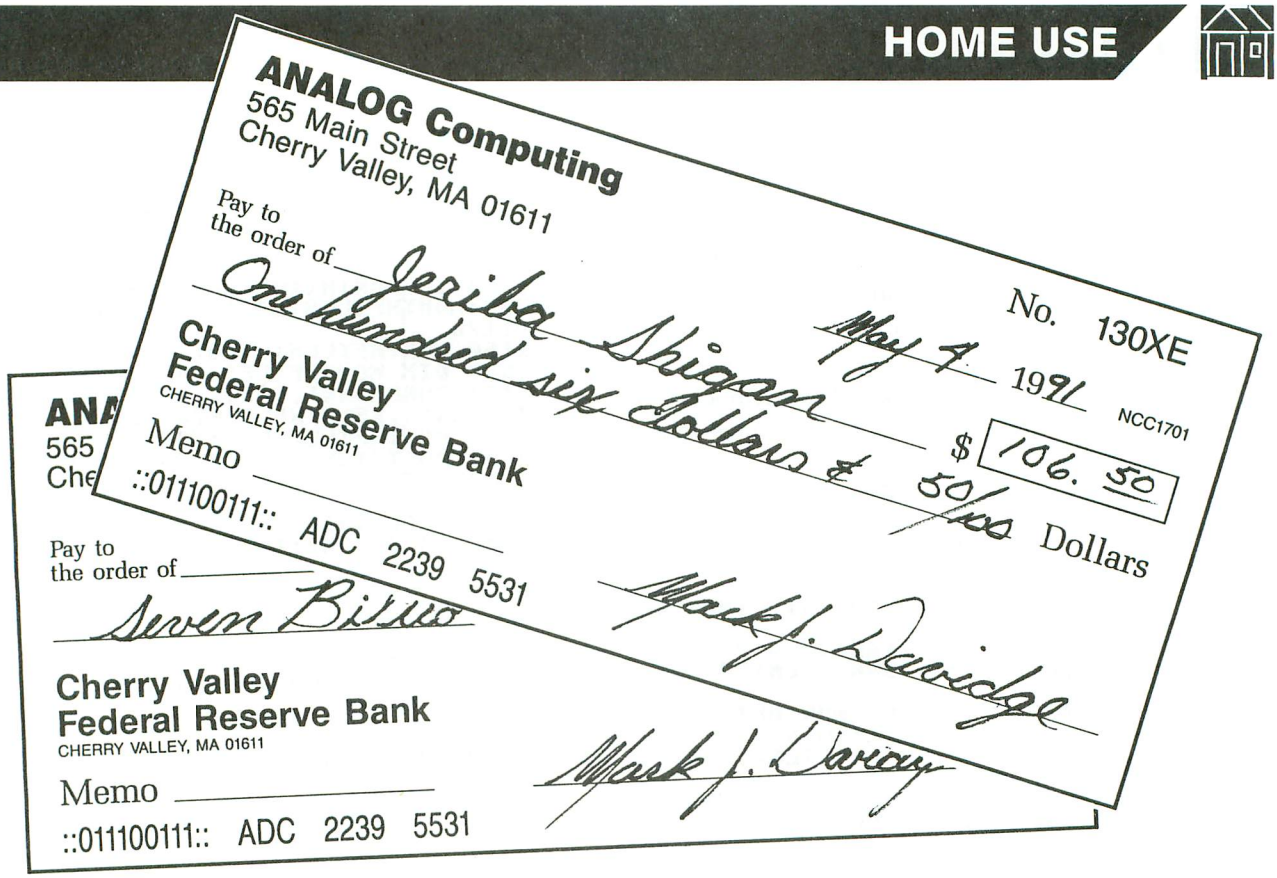
(5) When the compiler has finished, replace the disk in drive A with your linker disk.

(6) Double click the drive A icon.

(7) Double click the BATCH.TTP program and enter *LINK B:LIST1* into the parameter window.

(8) When the linking is complete, your source disk will contain the file LIST1.TOS. This is the executable version of the program. RUN it by giving it a double click.

The above instructions will work with all the C program listings presented thus far in **C-manship**. Only the filenames you use must be changed.



MicroCheck

For two-drive systems

by Clayton Walnum

It's been just about a year since **MicroCheck** saw print (issue 27), and the letters are still coming in. I can't tell you how flattered and pleased I am that the program turned out so successfully.

When I first sat down to write **MicroCheck**, I was simply trying to put together something tailored to my needs, something small and simple with the functions I felt were important. I never even considered the possibility of publication.

Needless to say, the program grew and grew over the course of the six months I spent writing it. Had I realized at the beginning what I was getting into, I don't think I would have had the ambition to even start it. Life plays those little tricks on us—for our own good, of course.

Anyway, one of the most frequently asked questions about **MicroCheck** is "How can I get it to run with two drives?" For those who've asked this question, and for those who were planning to, I've come up with some quick and dirty enhancements.

I say "quick and dirty," because, under the pressure of time (other projects always in the works), I took the easy way out—a direct conversion to a two-drive system.

When you make the following additions, **MicroCheck** will run *exclusively* with two drives. It doesn't give you a choice. For this reason, you should retain a copy of the original program.

The conversion.

To convert **MicroCheck** to a two-drive system, type in the four listings, LIST each to disk, check them with **Unicheck**(see page 10), then merge them with

the appropriate **Microcheck** program module, as follows:

Listing 1 . . . . . add to CHECKBOO.
Listing 2 . . . . . add to CHECKPRT.
Listing 3 . . . . . add to CHECKBAL.
Listing 4 . . . . . add to UTILITY.

And that's it. Hope this makes **MicroCheck** that much more usable for you. As for me, I have to put the shoulder back to the wheel. These articles don't write themselves, you know!

Listing 1.
CHECKBOO.

```
440 IF MONTHFILE$="D2:AUTO.DAT" THEN G
OSUB AMOUNT:GOTO 310
1180 IF MONTHFILE$()"D2:AUTO.DAT" THEN
 POSITION N7,N17:? "RETURN = ENTER NEX
T CHECK"
1210 IF PEEK(764)=N12 AND MONTHFILE$()
"D2:AUTO.DAT" THEN SVE=N1:GOTO 1260
1870 XIO 33,#N1,N0,N0,MONTHFILE$:MONTH
FILE$=MONTHFILE$(N4)
1900 POKE 842,N12:CLOSE #N2:OPEN #N2,N
8,N0,"D2:BALANCE.DAT":? #N2;BALANCE$:?
 #N2;YEAR:CLOSE #N2
1930 OPEN #N2,N8,N0,"D2:AUTO.TMP"
1960 XIO 33,#N1,N0,N0,"D2:AUTO.DAT":XI
O 32,#N1,N0,N0,"D2:AUTO.TMP,AUTO.DAT"
2090 DIM PAYEE$(21),AMOUNT$(N7),MEMO$(
22),MONTHFILE$(N14),MONTHDAT$(6300),CH
NUM$(N4),CHECK$(63),B$(37),T$(N8)
2100 DIM TEMP$(22),CH$(40),NAME$(N20),
ADDRESS$(N20),CITY$(N20),BALANCE$(N8),
DATE$(N8),TEMPFILE$(N14),A$(22)
2110 DIM AUTO$(315),AUTOFILE$(N14),FUL
L$(N5)
2180 POSITION 3,12:? #6;"into drive #2
":POSITION 3,22:? #6;"PRESS ANY KEY":G
OSUB IN2
2190 TRAP 2730:OPEN #N1,N4,N0,"D2:NAME
.DAT":INPUT #N1;NAME$:INPUT #N1;ADDRES
S$:INPUT #N1;CITY$:CLOSE #N1
2200 OPEN #N1,N4,N0,"D2:BALANCE.DAT":I
NPUT #N1;BALANCE$,YEAR:CLOSE #N1
2220 OPEN #N1,N4,N0,"D2:AUTO.DAT"
2410 MONTHFILE$="D2:MONTH  .DAT":GOSUB
 SND1:POSITION N20,N16:? "     enter mo
nth  "
2440 IF TEMP$="A" THEN MONTHFILE$="D2:
AUTO.DAT":POSITION 30,N5:? "AUTO":GOTO
 2540
2530 MONTHFILE$(N9,N10)=DATE$
2590 IF COUNT=N0 AND MONTHFILE$="D2:AU
TO.DAT" THEN 2760
2640 CHNUM$="0000":OPEN #N2,N4,N0,"D2:
AUTO.DAT"
```

CHECKSUM DATA.
(see page 10)

```
440 DATA 917,293,513,934,655,108,275,2
24,482,310,810,191,819,50,676,7257
2440 DATA 264,715,373,712,2064
```

Listing 2.
CHECKPRT.

```
1150 IF MONTH<10 THEN MONTHFILE$(9,9)=
"0":MONTHFILE$(10,10)=STR$(MONTH):GOTO
 1170
1160 MONTHFILE$(9,10)=STR$(MONTH)
1580 DIM MONTHFILE$(14),TEMP$(63),DATE
$(8),CHNUM$(4),C$(1),A$(1):K1=1:K2=2:K
3=3:K4=4:K5=5:K6=6:K7=7:K8=8
1610 MONTHFILE$="D2:MONTH  .DAT"
1650 POSITION 3,12:? #6;"into drive #2
":POSITION 3,22:? #6;"PRESS ANY KEY":G
OSUB IN2
1660 TRAP 1910:OPEN #2,4,0,"D2:MONTH01
.DAT":CLOSE #2:TRAP 40000
```

CHECKSUM DATA.
(see page 10)

```
1150 DATA 984,415,203,53,820,556,3031
```

Listing 3.
CHECKBAL.

```
210 DIM MONTHFILE$(N14),CHECK$(63),CHE
CKNUM$(N4),DATE$(N8)
270 POSITION N3,N12:? #N6;"into drive
#2":POSITION N3,22:? #N6;"PRESS ANY KE
Y":GOSUB 80
280 TRAP 1550:OPEN #N2,N4,N0,"D2:MONTH
01.DAT":CLOSE #N2
330 MONTHFILE$="D2:MONTH  .DAT"
340 IF MONTH<N10 THEN MONTHFILE$(N9,N9
)="0":MONTHFILE$(N10,N10)=STR$(MONTH):
GOTO 360
350 MONTHFILE$(N9,N10)=STR$(MONTH)
810 TEMPFILE$=MONTHFILE$:TEMPFILE$(N12
,N14)="TMP":CLOSE #N1:OPEN #N1,N8,N0,T
EMPFILE$
840 XIO 33,#N1,N0,N0,MONTHFILE$:MONTHF
ILE$=MONTHFILE$(N4)
990 MONTHFILE$="D2:MONTH  .DAT"
1020 IF MONTH<N10 THEN MONTHFILE$(N9,N
9)="0":MONTHFILE$(N10,N10)=STR$(MONTH)
:GOTO 1040
1030 MONTHFILE$(N9,N10)=STR$(MONTH)
1110 CLOSE #N2:OPEN #N2,N4,N0,"D2:BALA
NCE.DAT":INPUT #N2;BALANCE$:CLOSE #N2
```

CHECKSUM DATA.
(see page 10)

```
210 DATA 510,173,999,281,904,821,92,60
9,311,826,549,498,6573
```

Listing 4.
UTILITY.

```
150 POSITION 3,12:? #6;"into drive #2"
:POSITION 3,22:? #6;"PRESS ANY KEY":GO
SUB 20:RETURN
200 DIM NAME$(20),ADDRESS$(20),CITY$(2
0),BALANCE$(8),MONTHFILE$(14),B$(20),C
HECK$(63)
390 MONTHFILE$="D2:MONTH   .DAT":B$="
                "
430 POSITION 6,12:? #6;"drive #2":POSI
TION 3,22:? #6;"PRESS ANY KEY":GOSUB 2
0
440 TRAP 450:OPEN #1,4,0,"D2:MENU":CLO
SE #1:GOTO 760
450 GRAPHICS 18:GOSUB BRKDIS:POSITION
2,5:? #6;"FORMATTING DISK":XIO 254,#1,
0,0,"D2:"
550 OPEN #1,8,0,"D2:NAME.DAT":? #1;NAM
E$:? #1;ADDRESS$:? #1;CITY$:CLOSE #1
690 OPEN #1,8,0,"D2:BALANCE.DAT":? #1;
BALANCE$:? #1;YEAR:CLOSE #1
720 FOR X=0 TO 12:POSITION 13,10:? #6;
X:IF X<10 THEN MONTHFILE$(9,9)="0":MON
THFILE$(10,10)=STR$(X):GOTO 740
730 MONTHFILE$(9,10)=STR$(X)
750 OPEN #1,8,0,"D2:AUTO.DAT":? #1;"EN
D":CLOSE #1:GOTO INSPRO
760 GRAPHICS 17:GOSUB BRKDIS:POSITION
4,8:? #6;"PROGRAM DISK":POSITION 9,10:
? #6;"IN"
770 POSITION 6,12:? #6;"DRIVE #2":GOSU
B SND1:GOSUB DELAY:GOTO 420
810 GOSUB 1110:MONTHFILE$="D2:MONTH   .
DAT"
820 GOSUB INSDAT:TRAP 980:OPEN #2,4,0,
"D2:MONTH01.DAT":TRAP 40000
830 CLOSE #2:OPEN #2,8,0,"D2:MONTH00.T
MP"
850 FOR X=0 TO 12:POSITION 13,10:? #6;
X:IF X<10 THEN MONTHFILE$(9,9)="0":MON
THFILE$(10,10)=STR$(X):GOTO 870
860 MONTHFILE$(9,10)=STR$(X)
930 XIO 33,#1,0,0,"D2:MONTH00.DAT":XIO
 32,#1,0,0,"D2:MONTH00.TMP,MONTH00.DAT
"
940 OPEN #1,4,0,"D2:BALANCE.DAT":INPUT
 #1;BALANCE$:INPUT #1;YEAR:YEAR=YEAR+1
950 CLOSE #1:OPEN #1,8,0,"D2:BALANCE.T
MP":? #1;BALANCE$:? #1;YEAR:CLOSE #1
960 XIO 33,#1,0,0,"D2:BALANCE.DAT":XIO
 32,#1,0,0,"D2:BALANCE.TMP,BALANCE.DAT
"
1040 GOSUB INSDAT:TRAP 1090:OPEN #2,4,
0,"D2:AUTO.DAT":CLOSE #2
1060 OPEN #2,8,0,"D2:AUTO.DAT":? #2;"E
ND":CLOSE #2
```

CHECKSUM DATA.
(see page 10)

```
150 DATA 140,146,19,412,593,710,842,88
1,131,788,802,739,615,590,176,7584
830 DATA 1,149,799,160,34,14,220,755,8
72,3004
```

COMPUTER PALACE WE KNOW ATARI!

CALL US AT (503)683-5361 FOR INFORMATION

INVENTORY MASTER

Atari 400/800,XL/XE

$89.95 ST version **$179.95**

REPORTS GENERATED:
- Purchase Order
- Recommended Orders Report
- Inventory Control Report
- Product History Report

- Create your own invoice format
- Configures for any printer
- Adds more power to Inventory Master or use by itself.
- Can use 1 to 4 disk drives

Requirements: 48K ram • 1 Disk drive • 1 Printer

INVOICE MASTER

Atari 400/800,XL/XE

$89.95 ST version **$179.95**

INVENTORY MASTER AND A 48K ATARI COMPUTER GIVES SMALL BUSINESS THE ABILITY TO BECOME A MASTER OF INVENTORY CONTROL AT A VERY LOW PRICE. INVENTORY MASTER has all the features of programs costing many times more. Here are just a few: •Use 1 - 4 disk drives • Over 1700 records per disk • Retrieve any item within 5 seconds • Fast edit capability, plus many more features.

INVOICE MASTER is a very versatile program that allows small business owners to keep track of orders and then print them to an invoice of your own creation. Invoice Master can be used by itself or with Inventory Master to better manage your inventory. Over 200 records with up to 20 data fields per disk each can be stored, sorted, retrieved and finally printed to invoices saving hours of time.

Help calc Let HelpCalc Help You

SPREADSHEET TEMPLATES READY TO LOAD-N-GO FOR USE WITH SynCalc FEATURES:

Only **$19.95**

- Eight ready-to-use Templates.
- Easy to use...Just load them from Syncalc and enter your data.
- Save hours of tedious spreadsheet setup.
- All formats are pre-tested for accuracy.
- Templates included:
  - Loan Amorization Schedule
  - Net Present Value
  - Future Value
  - Personal Financial Statement
  - Alternative Investment Analysis
  - Check Register
  - Personal Register (Roster)
  - Depreciation Schedule

Super Mailer PLUS

One of the most versatile data-base programs available.

Features Includes:
- Lighting fast retrieval
- Fast Sorts on any field
- Supports up to 4 drives
- Single or double density
- Store about 1100 records per disk side in double density
- On-screen prompts
- Help-screens
- State abbreviation table
- Redefinable fields
- Print labels 1, 2, or 3-up
- View records on screen
- Search on any field (Fast sort on name field - 1 sec. to find a name out of 1000)
- Much more!

800, XL, XE 39.95
ST ver. 69.95

TRIVIA QUEST 48K Disk

- Program Covers 4 Disk Sides
- Outsmart Your Friends
- Outwit The Dragon
- Join The Quest

Only $39.95
For 800, XL,XE models with at least 48K and one disk drive.

A new concept in computer gaming. Intellectual challenge, strategy and arcade action. Each player assumes the role of a lord with a questing party of three characters. Complete the quest, earn the most gold by answering questions and battling the dragon. Win the favor of the king and thus, the game. There is also a Trivia Quest Utility disk available.

Utility disk: 1000 Additional questions plus create your own...**$24.95**

SPECIALS

DONKEY KONG	9.95 C
SUMMER GAMES	19.95 D
CHESS PARKER BRO'S	29.95 C
INV. TO PROG. #3	7.95 T
SANDS OF EGYPT	14.95 D
PACMAN/QIX	14.95 C
METEOR STORM*	4.95 D
DIGGER BONK	7.95 D
MUSIC MAJOR	7.95 D
LETTER WIZARD	29.95 D
GERMAN ATARIWRITER	39.95 D
BASEBALL STRATEGY	9.95 T
VISICALC	29.95 D

*** Note: any product with a star means not XL compatible or requires translator disk.**

No Documintation Tapes Only

BIG 99¢ SALE

FANCY FONTS
MATHS FOR FUN
SPACE GAMES
INSTEDIT
BRAIN BOGGLERT
MINI WORD PROC
MUSIC MAJOR
DIGGERBONK
DATA BASE DIALER
GUESS WHO'S COMING TO DINNER
MARATHON
BOB'S BUSINESS
PROTO'S GAME
GRADE BOOK
FONETONE
MEMORY MAP
DISPLAY LISTS

ONLY 7.95 Each Disk

#1 DISPLAY LISTS
#2 SCROLLING
#3 PAGE FLIPPING
#4 BASICS OF ANIMATION
#5 PLAYER MISSILE GRAPHICS
#6 SOUND & MUSIC
#7 DISK UTILITIES
#8 CHARACTER GRAPHICS
#9 GTIA,GRAPHICS 9-11
#10 SOUND EFFECTS
#11 MEMORY MAP TUTORIAL
#12 S.A.M. TUTORIAL
INSTEDIT
MARATHON
THE GRAPHICS MACHINE
MUSIC MAJOR
SPACE GAMES
MINI DATABASE/DIALER
MINI WORDPROCESSER
KID'S GAMES #1
KID'S GAMES #2
DOG DAZE
BOWLERS DATABASE
FONETONE
ATARI GRADE BOOK
THE BEAN MACHINE
GUESS WHAT'S COMING TO DINNER
MATHS FOR FUN
BRAIN BOGGLER
THE ADVENTURES OF PROTO
PROTO'S FAVORITE GAMES
PROTO'S FUN DAY
MASTER MEMORY MAP
GRAPHIC LABELS . . **ONLY $2.95**
TRICKY TUTORIALS #1-#6 . . . . . **$24.95**
TRICKY TUTORIALS #7-12 . . . . . **$24.95**

OUR LIBRARY SPECIALS!!!

DISK DISK DISK

$7.95 EA. $7.95 EA. $7.95 EA.

GAMES DISK #1	EDUC. DISK #4	BEST OF ACE #2
GAMES DISK #2	EDUC. DISK #5	BEST OF ACE #3
GAMES DISK #3	EDUC. DISK #6	BEST OF ACE #4
GAMES DISK #4	EDUC. DISK #7	BEST OF ACE #5
DEMOS DISK #1	EDUC. DISK #8	BEST OF ACE #6
DEMOS DISK #2	EDUC. DISK #9	BEST OF ACE #7
DEMOS DISK #3	EDUC. DISK #10	BEST OF ACE #8
UTILITIES #1	SPELLING BEE	BEST OF ACE #9
UTILITIES #2	BEST OF ACE #1	BEST OF ACE #10
UTILITIES #3		BEST OF ACE #11
ACTION DISK #1		BEST OF ACE #12
ACTION DISK #2		BEST OF ACE #13
ACTION DISK #3		BEST OF ACE #14
EDUC. DISK #1		
EDUC. DISK #2		
EDUC. DISK #3		

$5.95 each
WIZ OF WOR C
DELUXE INVADERS C
GORF (Not for XL) C
MINER 2049ER C

PROTECT YOUR EQUIPMENT

DELUXE DUST COVERS

Custom fitted, attractive leather brown color:
• ATARI 400, 800, 600/800/1200XL, New XE & ST, 410, 810, 1050, 1025, 1027, CX85 • EPSON, GEMINI, PROWRITER printers • INDUS, RANA, PERCOM, TRAK disk drives.

Additional covers ordered at same time.

ONLY **$8.95** EACH | ONLY **$7.95** EACH

DISK NOTCHER **9.95**

HARDWARE

LEGEND 808	199.00
PANASONIC 1091	299.95
PANASONIC 1092	399.50
OKIMATE 10	179.50
EPSON LX-80	249.95
1050 DISK DRIVE	169.95
INDUS GT	249.95
10 MEG.H.D.DRIVE(XL)	849.95
SWIVEL BASE STAND	29.95
OAK MONITOR STAND	39.95
XM 301 MODEM	44.90
MPP 1000E MODEM	59.95
UCALL POCKET MODEM	129.95
MPP1150 PRINTER INTF	59.95
16K RAM (800 only)	17.95

MISC

JOYSTICK HANDLE	1.49
JOYSTICK BOARD	2.49
JOYSTICK CORD	2.98
400/800 POWER SUPPLY	24.95
XL/XE POWER SUPPLY	24.95
XL 6'PRINTER CABLE	29.95
XL 12'PRINTER CABLE	36.95
ST 6'PRINTER CABLE	24.95
ST 12'PRINTER CABLE	28.90
ST LOGO BK	17.95
ST COMPANION BK	17.95

ST PROGRAMS

DEGAS	34.90
SOLUTIONS LETTERS	44.90
HABAHIPPO C COMPILER	69.50
HABA WRITER	59.90
HIPPO SIMPLE	44.90
HIPPO SPELL	33.90
PC/INTERCOMM	99.95
FINAL WORD	109.50
TRANSFER IBM/ATARI	39.95
FORTRAN	79.95
PASCAL	79.95
REGENT WORD 1	44.90
ST-TALK	17.95
ST WRITER/NEO	14.95
4 X FORTH 1	99.50
VIP PROFESSIONAL	CALL
FORBIDDEN QUEST	34.90
MUDPIES	26.90
KINGS QUEST II	44.90
WINNIE	26.90
ULTIMA II	44.90
FLIP SIDE	34.95
THE CRIMSON CROWN	34.90
TRANSYLVANIA	34.90
TREASURE IS	34.90
SPELLBREAKER	39.50

ST STAND

Only 29.95
+ $5.00 min. shipping charge
Beautifully finished stand to hold your ST monitor, 2 disk drives, a modem, ect . . .

COMPUTER PALACE

OPEN M-F, 9-6 Sat. 10-4 (Pacific Time)
2160 W. 11th Avenue Eugene, Oregon 97402

MasterCard VISA AMERICAN EXPRESS

USE YOUR CREDIT CARD & CALL
Toll Free 1-800-452-8013
★ ORDERS ONLY, PLEASE ★
There's never a penalty for using your credit card!
For Information, Call (503) 683-5361
Prices subject to change without notice.

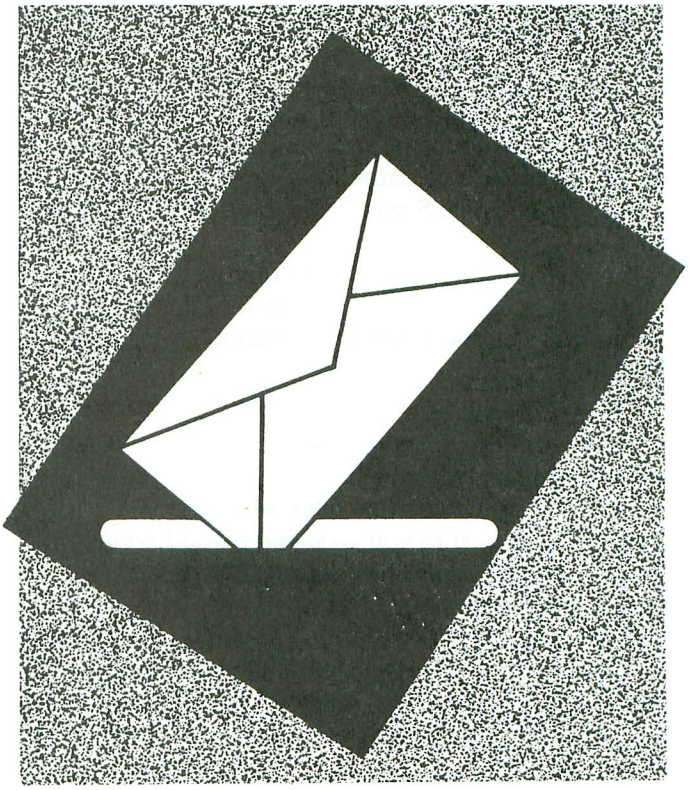
SHIPPING INFO: Minimum $2.90 Ground, $4.75 Air. Actual Cost depends on weight. Call (503) 683-5361 for information.
WARRANTY INFO: Everything that we sell is warrantied by the manufacturer. If any item purchased from us fails to perform properly when you receive it, call us at (503) 683-5361 so that we can assist you. No returned merchandise accepted without authorization. Defective software will be replaced with another copy of the same program, otherwise, no software is returnable.

• 2 Day Air Shipping **AVAILABLE** •

CIRCLE #135 ON READER SERVICE CARD

Micro-Mail

Update addresses with ease



by Clayton Walnum

Remember last December, when it was time to send out all those Christmas cards, and you tore the whole house apart looking for that little memo book with all your friends' and relatives' addresses in it? And remember, when you finally found it what a mess it was, with all the old addresses crossed out and all the new ones scribbled in the margins?

And remember your next door neighbors? You know, the ones who always let their dog run free, and, though you sent them a card last year, you got tired of scraping off the bottoms of your shoes every time you mowed the lawn, so you *sure* weren't going to send them a card this year. So you scribbled out their address, too. The book was *really* messy now!

You spent three hours copying all the addresses into a new book (except for Uncle Bernie's, since you couldn't read it). When you were all done, though, you realized that Cousin Bob had moved last year, but you hadn't changed the entry in the old book, and now the *new* book had the wrong address. That meant you could either scribble it out (making the new book look a lot like the old one), or you could start all over with *another* book.

Rather than spend an additional three hours trying to copy everything over again, you crossed it out (mumbling things about people who couldn't stay put).

When you were all done, there wasn't time to do the Christmas cards. So you went to bed, then the next day tore the house apart looking for that little memo book . . .

Deja vu.

Sounds a bit familiar, huh? It's time for you to get all those addresses organized, and **Micro-Mail** is just the thing you need. **Micro-Mail** will take the place of that memo book, allowing you to add, change or delete addresses without all that ugly scratching out. You'll be able to find any address in the blink of an eye, or print out a listing of any (or all) of the addresses you've entered. You can even make mailing labels!

And you're always going to know where your "book" is. It's going to be stored on that miracle of modern technology known as the floppy disk (or on a cassette, for those of you without disk drives). Interested? Thought you might be.

Typing it in.

Those of you with disk drives should type in Listing 1, then SAVE a copy. Put a disk containing DOS in your drive, then RUN the program. An AUTO-

RUN.SYS file will be written to your disk, making **Micro-Mail** self-booting.

Now you can type in Listing 2, the main program. When you've finished and made sure there are no typos, SAVE it to the disk under the filename MICRO-MAI. Make sure you spell the filename correctly, or the AUTORUN.SYS won't be able to run your program.

Cassette users should ignore Listing 1. Instead, go ahead and type in Listing 2, then add the changes in Listing 3. Now CSAVE the resulting program to a fresh cassette.

As you're typing in Listing 2, make sure you put your own name and address in the data statement on Line 2050.

Getting started.

When you're sure you've saved a good copy of the program, type *RUN* and press RETURN.

The first time you run **Micro-Mail**, you will have no data file. If you're using the cassette version, you will be asked if you have a data cassette. Answer by pressing *N*. Once you've created an address file, you will always answer this prompt with a *Y*. The only exception will be if you wish to start a new file.

The disk version will automatically check your disk for a data file.

You should now see the main screen. (You don't? Sure you got all those typos?) At the top of the screen, you'll see an envelope with your return address in the upper left and some dotted lines in the middle. Each of the dotted lines represents a field in the address data. They also show you how many characters a particular field is allowed. At the bottom of the screen is the options menu. To choose a function, simply press the number that corresponds with your choice.

Adding addresses.

Press the number *3* to add addresses. The cursor will appear at the beginning of the NAME field. Now get that old memo book. We're going to be entering all those addresses into **Micro-Mail**.

Micro-Mail will prompt you as you enter the name, street, city, state, and zip code for each address. When you finish entering a field, hit RETURN. The cursor will jump to the beginning of the next field. If you wish to return to the menu, press ESC. If you press ESC, the address you were working on will *not* be stored.

When the complete address has been entered, you will be asked if the information is correct. Check the address displayed on the screen. If it's what you want, then press *Y*. If you've made an error, press *N*. You will then be able to try again.

There's one thing you should be aware of as you enter information. Each time you add a new address, the addresses in memory are sorted into alphabetical order. They're sorted on the "last name." In other words, whatever you enter as the last word in the name field is the key the program uses to place the address into its proper position. Normally, this would be a person's last name. However, if you should enter *Robert Finnegan, M.D.*, the address will be sorted as if the last name were M.D. For this reason, you should avoid adding titles to entries in the name field.

Finding an address.

To find a specific address, use option 2, "scan." You'll be prompted to enter a name. You don't have to enter a complete name. If you're looking for Randy Smith's address, you could just enter *SMI*. All addresses with names that start with *SMI* will then appear, one by one, on the screen. When you see the address you need, press RETURN, and the computer will pause.

If you press RETURN again, the computer will continue displaying addresses until it either reaches the end of the list or you press RETURN to pause again. You may press ESC at any time to return to the options menu.

If you wish to scan the entire list from the beginning, simply enter * when prompted for a name.

If the computer can find no entry that matches the name you entered, you'll receive the message *ENTRY NOT FOUND*.

Changing addresses.

Remember Cousin Bob? Well, next time he moves, you can change his entry in **Micro-Mail** with very little fuss. Just press the number *4* for "change." You'll then be asked to enter a name. The same rules apply here as when you entered a name in the scan mode. The computer will find the first address that matches the name you've entered. You'll then be asked if you wish to change this address. If you answer *N*, the next address will be displayed.

When you find the address you wish to change, press *Y*. You'll then be prompted to enter the required information for each field. If the field on which the cursor rests doesn't need a change, simply press RETURN. You may press ESC at any time to return to the menu. However, if you do, the address will remain unchanged.

Deleting addresses.

This is choice 5 on the option menu, and it works

just like the change address function does. After you've entered a name, you'll be shown the first address that fits your search criteria and asked if you want it deleted. If you press *N*, you'll be shown the next address on the list. If you press *Y*, you will be asked *ARE YOU SURE?* Press *Y* to delete the address; press *N* to see the next. You may press ESC at any time to return to the menu.

Printing addresses.

Option 1, "print," is the most complicated function of **Micro-Mail**. There are two possible formats for printing. The first is in the form of a label. **Micro-Mail** is designed to print on standard $3\frac{1}{2} \times \frac{15}{16}$-inch mailing labels. The second way to print your addresses is in a horizontal format on an $8\frac{1}{2} \times 11$-inch sheet.

You'll also be given the choice of printing a single address or creating a list. If you choose to print a single address, label format is assumed.

To create a list, first press the number *2* for the print option, then answer *Y* to the question *IS THIS A LIST?* You'll then be asked to enter a name. **Micro-Mail** will start displaying addresses at the first entry that matches the name you enter. Each time a new address is displayed, you'll be asked if you want it included in your list. Answer *Y* or *N*. The next address will then appear. This will continue until either the program runs out of addresses or you press ESC. ESC has a slightly different function in the list mode. Pressing ESC tells **Micro-Mail** that you're done with your list and are ready to print it out.

If you want a listing of all the addresses you've stored, just enter * in response to the name prompt. This saves you from choosing each address individually.

You'll be asked if you want labels. If you answer *N*, the addresses will be printed in the horizontal format. If you answer *Y*, you'll be asked the number of labels you wish printed. This number of labels will be printed for each address you have selected. Type in a number and press RETURN. The printing will begin.

If you choose not to create a list, you'll be prompted to enter a name just as above, but will instead be asked *PRINT THIS ADDRESS?* If you answer *N*, the next address will be displayed. If you answer *Y*, you'll be asked how many labels you wish printed. Type a number and press RETURN.

You may press ESC at any time to abort a printout in progress.

Ending a session.

You must *always* end a **Micro-Mail** session with option 6, "end." This is when **Micro-Mail** saves your data. If you're using the cassette version, you'll be prompted to cue your cassette.

If you're using the disk version, the data will be saved to the disk in drive 1. This should be your **Micro-Mail** program disk.

If you've made no changes in your address data, pressing 6 simply returns you to BASIC.

Due to the unreliability of cassettes, you should never record new data on top of the previous session. Each time you have to save updated data, flip the cassette over. This way, the previous session will remain intact and be available to you should the new data not load properly.

More room!

Micro-Mail, as printed here, can hold up to 100 addresses. If you have more than the minimum needed memory, then you can boost the capacity by changing the statement *NA=100* in Line 1720 to a higher value, as long as it doesn't exceed 256.

The machine language routines.

There are two machine language subroutines that you may find useful in other applications. The first is the search routine found in Lines 2100 through 2120 and stored in the string FND$. To use this routine in your own programs, you should store the data you wish to search in a string, in this case A$.

Each record in the string must be the same length, no more than 256 bytes. You then call the routine, using *A = USR (ADR1,ADR2,NR,RL,ADR3,L)*, where *ADR1* is the address of the routine, *ADR2* is the address of your data string, *NR* is the number of records stored in the string (not to be greater than 256), *RL* is the length of each record, *ADR3* is the address of the string you wish to search for, and *L* is the length of that string. When called, the routine will find the first data entry that's equal to the search string.

Another routine you may find handy is the sort routine. This is found in Lines 2130 to 2180 and stored in SORT$. To use this routine, the data to be sorted must be stored in a string. All records must be the same length, no longer than 256 bytes. The number of records to be sorted shouldn't exceed 256. Call the routine using *A = USR (ADR1,ADR2,RL,D, NR)*, where *ADR1* is the address of the routine, *ADR2* is the address of your data string, *RL* is the length of each record in the string, *D* is the depth of the sort, and *NR* is the number of records in the data string.

The sort will work just as well with multiple field data as with single field data. In fact, there's a little trick you can use that will allow you to sort your data on more than one field. Simply store each field of each record in the order of significance, then set *D* equal to the length of the key fields combined.

For example, if you wanted address data sorted on name, state, and then finally on city, each record within your data might look something like this:

SMITHJOHN CTHARTFORD 22 SOUTH STREET06108

You would then set *D* in the USR call to the length of all three key fields or, in this case, *22*.

Listing 1.
BASIC listing.

```
10 OPEN #1,8,0,"D:AUTORUN.SYS"
20 FOR X=1 TO 155:READ A:PUT #1,A:NEXT
 X
30 CLOSE #1:? :? "■ALL DONE!"
40 DATA 255,255,0,6,142,6,162,0,189,26
,3,201,69,240,5,232,232,232,208,244,23
2,142,105,6,189
50 DATA 26,3,133,205,169,107,157,26,3,
232,189,26,3,133,206,169,6,157,26,3,16
0,0,162,16,177
60 DATA 205,153,107,6,200,202,208,247,
169,67,141,111,6,169,6,141,112,6,169,1
4,141,106,6,96,172
70 DATA 106,6,240,9,185,128,6,206,106,
6,160,1,96,138,72,174,105,6,165,205,15
7,26,3,232,165
80 DATA 206,157,26,3,104,170,169,155,1
60,1,96,7,0,251,243,51,246,67,6,163,24
6,51,246,60,246
90 DATA 76,228,243,0,32,32,32,32,32,32
,73,65,77,79,82,67,73,77,58,68,34,78,8
5,82,226,2,227,2,0,6
```

●

CHECKSUM DATA.

(see page 10)

```
10 DATA 26,316,812,536,325,812,297,549
,106,3779
```

●

Listing 2.

```
10 REM ┌──────────────────┐
20 REM │ MICRO-MAIL       │
30 REM │ by Clayton Walnum│
35 REM REVISED 12/16/85
40 GOTO 1700
50 RESTORE 2100:FOR X=N1 TO 75:READ A:
FND$(X)=CHR$(A):NEXT X:FOR X=N1 TO 152
:READ A:SORT$(X)=CHR$(A):NEXT X
60 RETURN
70 POKE N16,112:POKE 53774,112:RETURN
80 FOR X=LEN(T$) TO N1 STEP -N1:IF T$(
X,X)=" " THEN NEXT X
90 T$=T$(N1,X):RETURN
100 IF CNT THEN RETURN
110 POKE 708,54:POSITION N1,N16:? "ADD
RESS BOOK EMPTY":GOSUB 120:FOR X=N1 TO
 800:NEXT X:POP :GOTO 760
120 SOUND N0,90,N12,N8:FOR X=N1 TO 50:
NEXT X:SOUND N0,N0,N0,N0:RETURN
130 SOUND N0,N10,N12,N8:FOR X=N1 TO N2
0:NEXT X:SOUND N0,N0,N0,N0:RETURN
140 POKE 708,54:POSITION N1,N16:? " EN
TRY NOT FOUND  ":GOSUB 120:FOR X=N1 TO
 800:NEXT X:GOSUB 380:GOTO 760
150 REM IF SCAN AND PEEK(764)=N12 THEN
 P=P-N1:RETURN
160 NA$=A$(P*70-69,P*70):FOR X=N1 TO N
20:IF NA$(X,X)<>" " THEN NEXT X
170 LN$=NA$(N1,X-N1):FN$=NA$(X+N1,24):
FOR X=LEN(FN$) TO N1 STEP -N1:IF FN$(X
,X)=" " THEN NEXT X
180 FN$=FN$(N1,X):AD$=NA$(N25,48):FOR
X=N24 TO N1 STEP -N1:IF AD$(X,X)=" " T
HEN NEXT X
190 AD$=AD$(N1,X):C$=NA$(49,63):FOR X=
N15 TO N1 STEP -N1:IF C$(X,X)=" " THEN
 NEXT X
200 C$=C$(N1,X):GOSUB 420:RETURN
210 OPEN #N1,N4,N0,"K:":POKE 764,255:P
OKE 702,64:POKE 694,N0
220 IF PEEK(764)=255 THEN GOTO 220
230 IF PEEK(764)=39 OR PEEK(764)=60 OR
 PEEK(764)=124 OR PEEK(764)=103 THEN P
OKE 764,255:GOTO 220
240 GET #N1,A:CLOSE #N1:RETURN
```

```
250 L=N0:T$="":POKE 764,255:LOCATE COL
,ROW,B:POSITION COL,ROW:IF B>128 THEN
? CHR$(B):GOTO 270
260 ? CHR$(B+128)
270 GOSUB 210:IF A=RETRN THEN POKE 752
,N1:RETURN
280 IF A=ESCAPE THEN 760
290 POKE 752,N0:IF A=BACKSP THEN 320
300 L=L+N1:IF L>L1 THEN POKE 752,N1:RE
TURN
310 POSITION COL+L-N1,ROW:? CHR$(A);:T
$(L,L)=CHR$(A):GOTO 270
320 IF L>N0 THEN ? CHR$(126);:L=L-N1:I
F L=N0 THEN T$=""
330 IF L>N0 THEN T$=T$(N1,L)
340 GOTO 270
350 FLAG=N1:GOSUB 460:IF T$="*" THEN F
LAG=N0:P=N1:RETURN
360 P=USR(ADR(FND$),ADR(A$),CNT,70,ADR
(T$),LEN(T$)):FLAG=N0:RETURN
370 FOR X=N17 TO N21:POSITION N1,X:? B
$:NEXT X:RETURN
380 POKE 752,N1:POSITION N9,N8:? "----
--------------------":POSITION N9,N9:?
 "------------------------"
390 POSITION N9,N10:? "---------------
,-- -----":RETURN
400 POSITION N10,N17:? "1) SCAN    4)
CHANGE":POSITION N10,N18:? "2) PRINT
 5) DELETE"
410 POSITION N10,N19:? "3) ADD     6)
END":RETURN
420 IF SCAN AND PEEK(764)=N12 THEN P=P
-N1:RETURN
430 GOSUB 380:POSITION N9,N8:? FN$;" "
;LN$:POSITION N9,N9:? AD$
440 POSITION N9,N10:? C$:POSITION N25,
N10:? NA$(64,65):POSITION N28,N10:? NA
$(66)
450 RETURN
460 POSITION N1,N16:? "    enter name
   "
470 COL=N9:ROW=N8:L1=N24:GOSUB 250
480 IF EDIT AND T$="" THEN POSITION CO
L,ROW:? CHR$(B):T$=A$(P*70-69,P*70-46)
:GOTO 530
490 IF T$="" THEN GOSUB 120:GOTO 470
500 GOSUB 80:IF FLAG THEN RETURN
510 FOR X=LEN(T$) TO N1 STEP -N1:IF T$
(X,X)<>" " THEN NEXT X:GOSUB 120:GOTO
460
520 FN$=T$(N1,X-N1):LN$=T$(X+N1,LEN(T$
)):T$(N1)=LN$:T$(LEN(T$)+N1)=" ":T$(LE
N(T$)+N1)=FN$
530 NA$(N1,LEN(T$))=T$:RETURN
540 POSITION N1,N16:? "   enter addres
s  "
550 COL=N9:ROW=N9:L1=N24:GOSUB 250
560 IF EDIT AND T$="" THEN POSITION CO
L,ROW:? CHR$(B):T$=A$(P*70-45,P*70-N22
):GOTO 580
570 IF T$="" THEN GOSUB 120:GOTO 550
580 NA$(N25,N24+LEN(T$))=T$:RETURN
590 POSITION N1,N16:? "    enter city
   "
600 COL=N9:ROW=N10:L1=N15:GOSUB 250
610 IF EDIT AND T$="" THEN POSITION CO
L,ROW:? CHR$(B):T$=A$(P*70-N21,P*70-N7
):GOTO 630
620 IF T$="" THEN GOSUB 120:GOTO 600
630 NA$(49,48+LEN(T$))=T$:RETURN
640 POSITION N1,N16:? "    enter state
   "
650 COL=N25:ROW=N10:L1=N2:GOSUB 250
660 IF EDIT AND T$="" THEN POSITION CO
L,ROW:? CHR$(B):T$=A$(P*70-N6,P*70-N5)
:GOTO 680
670 IF LEN(T$)<>N2 THEN POSITION N25,N
10:? T$;"-":GOSUB 120:GOTO 650
680 NA$(64,65)=T$:RETURN
690 POSITION N1,N16:? "    enter zip
   "
700 COL=N28:ROW=N10:L1=N5:GOSUB 250
710 IF EDIT AND T$="" THEN POSITION CO
L,ROW:? CHR$(B):T$=A$(P*70-N4):GOTO 75
0
720 IF LEN(T$)<N5 THEN POSITION N28,N1
0:? T$;:FOR X=LEN(T$)+N1 TO N5:? "-";:
NEXT X:GOSUB 120:GOTO 700
730 FOR X=N1 TO N5:IF ASC(T$(X))<48 OR
 ASC(T$(X))>57 THEN GOSUB 120:GOTO 690
740 NEXT X
750 NA$(66,70)=T$:RETURN
760 POP :POP :POP :POP :GOSUB 380:GOSU
B 400:POSITION N1,N16:? "  make select
ion    ":POKE 708,40:EDIT=N0:SCAN=N0
770 TRAP 40000:GOSUB 210:IF A<ASC("1")
 OR A>ASC("6") THEN 770
780 POSITION N0,N16:? B$(N19):GOSUB 37
0:A=A-48:ON A GOTO 1160,790,1290,1530,
1380,1660
790 GOSUB 100:POSITION N0,N16:? "  IS
THIS A LIST?  "
800 GOSUB 210:IF A=ASC("Y") THEN LST=N
1:LST$=" ":LST$(CNT)=" ":LST$(N2)=LST$
:GOTO 830
810 IF A=ASC("N") THEN LST=N0:GOTO 830
820 GOTO 800
830 GOSUB 350:IF  NOT P THEN 140
840 IF T$(N1,N1)="*" AND LST THEN LST$
(N1)="*":LST$(CNT)="*":LST$(N2)=LST$:G
OTO 930
850 GOSUB 150:POSITION N1,N16:? "PRINT
 THIS ADDRESS?"
860 GOSUB 210:IF LST AND A=ESCAPE THEN
 930
870 IF A=ASC("N") THEN 910
880 IF A<>ASC("Y") THEN 860
890 PRT=N1:IF  NOT LST THEN 970
900 LST$(P,P)="*"
910 P=P+N1:IF P>CNT THEN 925
920 GOTO 850
925 IF  NOT PRT THEN 1120
930 GOSUB 380:GOSUB 130:POSITION N1,N1
6:? " ARE THESE LABELS? "
940 GOSUB 210:IF A=ASC("Y") THEN LABEL
S=N1:GOTO 970
950 IF A=ASC("N") THEN LABELS=N0:GOTO
990
960 GOTO 940
970 TRAP 970:POSITION N1,N16:? "    HO
W MANY       ":POSITION N13,N16:INPUT
Q
980 IF Q=N0 THEN 760
990 TRAP 1130:POSITION N0,N16:? B$(N19
):POKE 559,N0:POKE 764,255:IF  NOT LST
 THEN 1030
1000 IF  NOT LABELS THEN FOR X=N1 TO N
4:LPRINT :NEXT X:LINE=N0
1010 FOR P=N1 TO CNT:IF LST$(P,P)<>"*"
 THEN NEXT P:POKE 559,34:GOTO 760
1020 GOSUB 150:IF  NOT LABELS THEN 109
0
1030 FOR LABEL=N1 TO Q:IF PEEK(764)=N2
8 THEN POP :POKE 559,34:GOTO 760
1040 LPRINT "  ";FN$;" ";LN$:LPRINT "
 ";AD$
1050 LPRINT "  ";C$;", ";NA$(64,65);"
";NA$(66)
1060 LPRINT :LPRINT :LPRINT :NEXT LABE
L
1070 IF LST THEN NEXT P
1080 POKE 559,34:GOTO 760
```

```
1090 LPRINT LN$;", ";FN$;"   ";AD$;"
 ";C$;", ";NA$(64,65);" ";NA$(66):LPRI
NT
1095 IF PEEK(764)=ESCAPE+N1 THEN POKE
559,34:GOTO 760
1100 LINE=LINE+N2:IF LINE=60 THEN FOR
X=N1 TO N6:LPRINT :NEXT X:LINE=N0
1110 NEXT P
1120 PRT=N0:POKE 559,34:GOTO 760
1130 POKE 708,54:POKE 559,34:POSITION
N0,N16:? "   CHECK PRINTER!   ":GOSUB
120
1140 GOSUB 210:IF A=ESCAPE THEN 760
1150 POSITION N1,N16:? B$(N19):GOTO 99
0
1160 SCAN=N1:GOSUB 100:GOSUB 350:IF  N
OT P THEN 140
1170 POSITION N1,N16:? "  RETURN = STO
P   ":POSITION N23,N16:? "ESCAPE = MEN
U"
1180 POKE 764,255:GOSUB 150:FOR X=N1 T
O 100:NEXT X
1190 IF PEEK(764)=ESCAPE+N1 THEN 1280
1200 IF PEEK(764)<>N12 THEN 1260
1210 POSITION N12,N16:? "START":POKE 7
64,255
1220 IF PEEK(764)=N12 THEN 1250
1230 IF PEEK(764)=ESCAPE+N1 THEN 1280
1240 GOTO 1220
1250 POSITION N12,N16:? "STOP "
1260 P=P+N1:IF P>CNT THEN 1280
1270 GOTO 1180
1280 POSITION N1,N16:? B$(N19):POSITIO
N N21,N16:? B$(N19):SCAN=N0:GOTO 760
1290 IF CNT=NA THEN 1370
1300 NA$=" ":NA$(70)=" ":NA$(N2)=NA$:G
OSUB 460:GOSUB 540:GOSUB 590:GOSUB 640
:GOSUB 690
1310 POSITION N3,N16:? "EVERYTHING OK?
"
1320 GOSUB 210:IF A=ASC("N") THEN GOSU
B 380:GOTO 1290
1330 IF A<>ASC("Y") THEN 1320
1340 CNT=CNT+N1:A$(LEN(A$)+N1)=NA$
1350 IF CNT>N1 THEN A=USR(ADR(SORT$),A
DR(A$),70,24,CNT)
1360 SVE=N1:GOTO 760
1370 POKE 708,54:POSITION N1,N16:? "AD
DRESS BOOK FULL!":GOSUB 120:FOR X=N1 T
O 200:NEXT X:GOTO 760
1380 GOSUB 100:GOSUB 350:IF  NOT P THE
N 140
1390 GOSUB 150:POSITION N1,N16:? "DELE
TE THIS ENTRY?"
1400 GOSUB 210:IF A<>ASC("N") AND A<>A
SC("Y") AND A<>ESCAPE THEN 1400
1410 IF A=ESCAPE THEN 760
1420 IF A=ASC("Y") THEN 1450
1430 P=P+N1:IF P>CNT THEN GOTO 140
1440 GOTO 1390
1450 POSITION N1,N16:? "  ARE YOU SURE
?  "
1460 GOSUB 210:IF A<>ASC("N") AND A<>A
SC("Y") AND A<>ESCAPE THEN 1460
1470 IF A=ASC("Y") THEN SVE=N1:GOTO 15
00
1480 IF A=ASC("N") THEN 1430
1490 GOTO 760
1500 IF CNT=N1 THEN A$="":CNT=N0:GOSUB
 380:GOTO 760
1510 IF P=CNT THEN A$=A$(N1,(P-N1)*70)
:CNT=CNT-N1:GOTO 760
1520 A$(P*70-69)=A$((P+N1)*70-69):SVE=
N1:CNT=CNT-N1:GOTO 760
1530 GOSUB 100:GOSUB 350:IF  NOT P THE
N GOTO 140
1540 EDIT=N1:GOSUB 150:POSITION N1,N16
:? "CHANGE THIS ENTRY?"
1550 GOSUB 210:IF A<>ASC("N") AND A<>A
SC("Y") AND A<>ESCAPE THEN 1550
1560 IF A=ESCAPE THEN 760
1570 IF A=ASC("Y") THEN 1600
1580 P=P+N1:IF P>CNT THEN GOTO 140
1590 GOTO 1540
1600 NA$=" ":NA$(70)=" ":NA$(N2)=NA$
1610 GOSUB 460:GOSUB 540:GOSUB 590:GOS
UB 640:GOSUB 690:POSITION N3,N16:? "EV
ERYTHING OK?"
1620 GOSUB 210:IF A=ASC("N") THEN 1610
1630 IF A<>ASC("Y") THEN 1620
1640 A$(P*70-69,P*70)=NA$:SVE=N1:IF CN
T>N1 THEN A=USR(ADR(SORT$),ADR(A$),70,
24,CNT)
1650 GOTO 760
1660 IF A$="" OR  NOT SVE THEN 1690
1670 POKE 559,N0:OPEN #N1,N8,N0,"D:ADD
RESS.DAT"
1680 FOR X=N1 TO CNT:? #N1;A$(X*70-69,
X*70):NEXT X:CLOSE #N1
1690 GRAPHICS N0:END
1700 DIM R$(1),R(5):RA=ADR(R$)+1:RSW=1
1710 N1=1:N2=2:N3=3:N4=4:N5=5:N6=6:N7=
7:N8=8:N9=9:N10=10:N11=11:N12=12:N13=1
3:N14=14:N15=15:N16=16
1720 N17=17:N18=18:N19=19:N20=20:N21=2
1:N22=22:N23=23:N24=24:N25=25:N26=26:N
27=27:N28=28:N29=29:N30=30:NA=100
1730 FOR I=N0 TO N27:READ A:POKE RA+I,
A:NEXT I
1740 DATA 104,104,104,162,255,160,58,2
02,208,9,136,208,6,170,202,138,208,241
,96,142,10
1750 DATA 212,142,22,208,24,144,235
1760 GRAPHICS N17:GOSUB 70:POKE 559,N0
:DL=PEEK(560)+256*PEEK(561)+N4
1770 POKE DL-N1,71:POKE DL+N2,N7:POKE
DL+N3,N7
1780 POKE DL+N11,N2:POKE DL+N12,N2:POK
E DL+N18,65
1790 POKE DL+N19,PEEK(560):POKE DL+N20
,PEEK(561)
1800 POKE 708,N12:POKE 752,N1
1810 POKE 87,N0:POSITION N25,N0:? "MIC
RO-MAIL":POSITION N28,N1:? "THE"
1820 POSITION N25,N2:? "ELECTRONIC":PO
SITION N24,N3:? "ADDRESS BOOK"
1830 POSITION N13,N5:? "Copyright 1985
":POSITION N19,N6:? "by":POSITION N3,N
8:? "CLAYTON WALNUM"
1840 POKE 559,34:POKE RA+N23,N22:A=USR
(RA,N4)
1850 DIM A$(NA*70),NA$(70),T$(N25),B$(
38),N$(N25),AD$(N25),C$(N15),S$(N2),Z$
(N5),FND$(75),SORT$(152)
1860 DIM FN$(N20),LN$(N20),LST$(NA)
1870 B$=" ":B$(38)=" ":B$(N2)=B$
1880 SVE=N0:CNT=N0:RETRN=155:BACKSP=12
6:ESCAPE=N27
1890 TRAP 1910:OPEN #N1,N4,N0,"D:ADDRE
SS.DAT"
1900 INPUT #N1;NA$:CNT=CNT+N1:A$(CNT*7
0-69,CNT*70)=NA$:GOTO 1900
1910 CLOSE #N1:GOSUB 50
1920 GRAPHICS N0:GOSUB 70:POKE 559,N0:
DL=PEEK(560)+256*PEEK(561)+N4
1930 POKE DL+N15,130:POKE DL+N16,N6:PO
KE DL+N17,N6:POKE DL+N18,N6:POKE DL+N1
9,N6
1940 POKE DL+N24,65:POKE DL+N25,PEEK(5
60):POKE DL+N26,PEEK(561)
1990 RESTORE 2000:FOR X=1664 TO 1692:R
EAD A:POKE X,A:NEXT X
```

```
2000 DATA 72,138,72,152,72,169,10,162,
0,160,56,141,10,212
2010 DATA 141,24,208,142,24,208,140,23
,208,104,168,104,170,104,64
2020 POKE 709,N0:POKE 710,N8:POKE 711,
N12
2030 POKE 512,128:POKE 513,N6:POKE 542
86,192:POKE 752,N1:POKE 82,N1
2040 READ N$,AD$,C$,S$,Z$
2050 DATA YOUR NAME,YOUR ADDRESS,YOUR
CITY,ST,ZIP00
2060 POSITION N1,N1:? N$:? AD$:? C$;",
 ";S$;" ";Z$
2070 POSITION 34,N0:? "┌──┐":POSITION
N30,N1:? "──────┼──┼──"
2080 POSITION N30,N2:? "──────┼──┼──":POS
ITION N30,N3:? "──────┼──┼──":POSITION 34
,N4:? "└──┘"
2090 POKE 559,34:GOTO 760
2100 DATA 104,104,133,204,104,133,203,
104,104,141,0,6,104,104,141,1,6,104,13
3,206,104,133,205,104,104
2110 DATA 141,2,6,162,1,160,0,177,203,
209,205,208,8,200,204,2,6,208,244,240,
22,236,0,6,240
2120 DATA 15,232,165,203,24,109,1,6,13
3,203,144,224,230,204,176,220,162,0,13
4,212,169,0,133,213,96
2130 DATA 104,104,141,1,6,104,141,0,6,
104,104,141,2,6,104,104,141,3,6,104,10
4,141,5,6,173,1,6
2140 DATA 133,204,133,206,173,0,6,133,
203,24,109,2,6,133,205,144,2,230,206,1
69,0,141,6,6,162,1,160
2150 DATA 0,177,203,209,205,144,8,208,
38,200,204,3,6,208,242,232,236,5,6,240
,71,165,203,24,109,2,6
2160 DATA 133,203,144,2,230,204,165,20
5,24,109,2,6,133,205,144,212,230,206,1
76,208,169,1,141,6,6,160,0
2170 DATA 177,203,153,7,6,200,204,2,6,
208,245,160,0,177,205,145,203,200,204,
2,6,208,246,160,0,185,7
2180 DATA 6,145,205,200,204,2,6,208,24
5,240,179,173,6,6,208,129,96
```

●

CHECKSUM DATA.

(see page 10)

```
10 DATA 702,132,855,996,658,449,762,75
5,283,125,54,885,276,432,624,7988
150 DATA 887,193,299,35,960,440,364,45
3,356,247,984,958,684,304,780,7944
300 DATA 756,896,860,593,722,402,628,3
07,683,382,218,281,617,118,38,7501
450 DATA 602,539,747,699,708,553,967,8
6,867,215,746,570,700,348,378,8725
600 DATA 582,536,674,152,992,507,800,4
65,325,988,509,596,340,550,782,8798
750 DATA 313,234,522,232,417,599,701,7
17,7,900,912,595,86,435,239,6909
900 DATA 9,738,734,488,179,618,108,745
,563,550,412,4,753,946,991,7838
1040 DATA 83,979,512,52,277,780,526,27
9,509,977,655,398,828,473,625,7953
1180 DATA 213,809,667,455,893,806,712,
199,961,726,947,104,39,155,454,8140
1330 DATA 367,457,16,560,475,486,847,2
86,281,539,673,733,422,310,285,6737
```

```
1480 DATA 521,907,527,187,211,68,232,3
11,289,548,681,733,881,449,505,7050
1630 DATA 379,212,909,5,625,401,830,56
0,526,869,142,773,738,246,41,7256
1780 DATA 281,859,462,806,608,133,706,
963,238,196,651,691,178,946,321,8039
1930 DATA 715,687,538,964,663,812,65,4
5,209,679,113,973,280,76,202,7021
2120 DATA 885,424,852,881,385,76,956,4
459
```

●

Listing 3.

```
35 REM CASSETTE VERSION
1670 GRAPHICS N0:GOSUB 70:POSITION N1,
N8:? "CUE CASSETTE, PRESS RETURN TWICE
"
1675 GOSUB 210:OPEN #N1,N8,N0,"C:"
1890 GRAPHICS N0:GOSUB 70:POKE 752,N1:
POSITION N1,N8:? "DO YOU HAVE A DATA C
ASSETTE?";
1892 GOSUB 210:IF A=ASC("N") THEN 1915
1894 IF A<>ASC("Y") THEN 1892
1896 POSITION N1,N12:? "CUE CASSETTE,
PRESS RETURN TWICE":GOSUB 210:OPEN #N1
,N4,N0,"C:":TRAP 1910
1910 CLOSE #N1
1915 POSITION N1,N16:? "ONE MOMENT..."
:GOSUB 50
```

●

CHECKSUM DATA.

(see page 10)

```
35 DATA 468,47,331,771,540,421,514,776
,903,4771
```

●

Assembly listing.

```
; STRFIND.SRC
;
; ********************************
; *        SEARCH ROUTINE        *
; *                              *
; *      A=USR(A,B,C,D,E,F)      *
; *                              *
; *       A=ADR OF ROUTINE       *
; *       B=ADR OF TABLE         *
; *       C=# OF RECORDS         *
; *       D=RECORD LENGTH        *
; *       E=ADR SEARCH STRING    *
; *       F=SEARCH DEPTH         *
; *                              *
; ********************************
;
        *=  $5000
;
; EQUATES
; -------
;
SADR    =   $CB     ; ADR DATA STRING
KADR    =   $CD     ; ADR SEARCH STRING
NREC    =   $0600   ; # OF RECORDS
RLEN    =   $0601   ; LENGTH OF RECORDS
DEEP    =   $0602   ; DEPTH OF SEARCH
BRET    =   $D4     ; VALUE TO BASIC
;
; INITIALIZE
; ----------
;
        PLA         ; # OF ARGUMENTS
        PLA         ; DATA STRING ADR HI
        STA SADR+1
        PLA         ; DATA STRING ADR LO
        STA SADR
        PLA         ; IGNORE HI BYTE
```

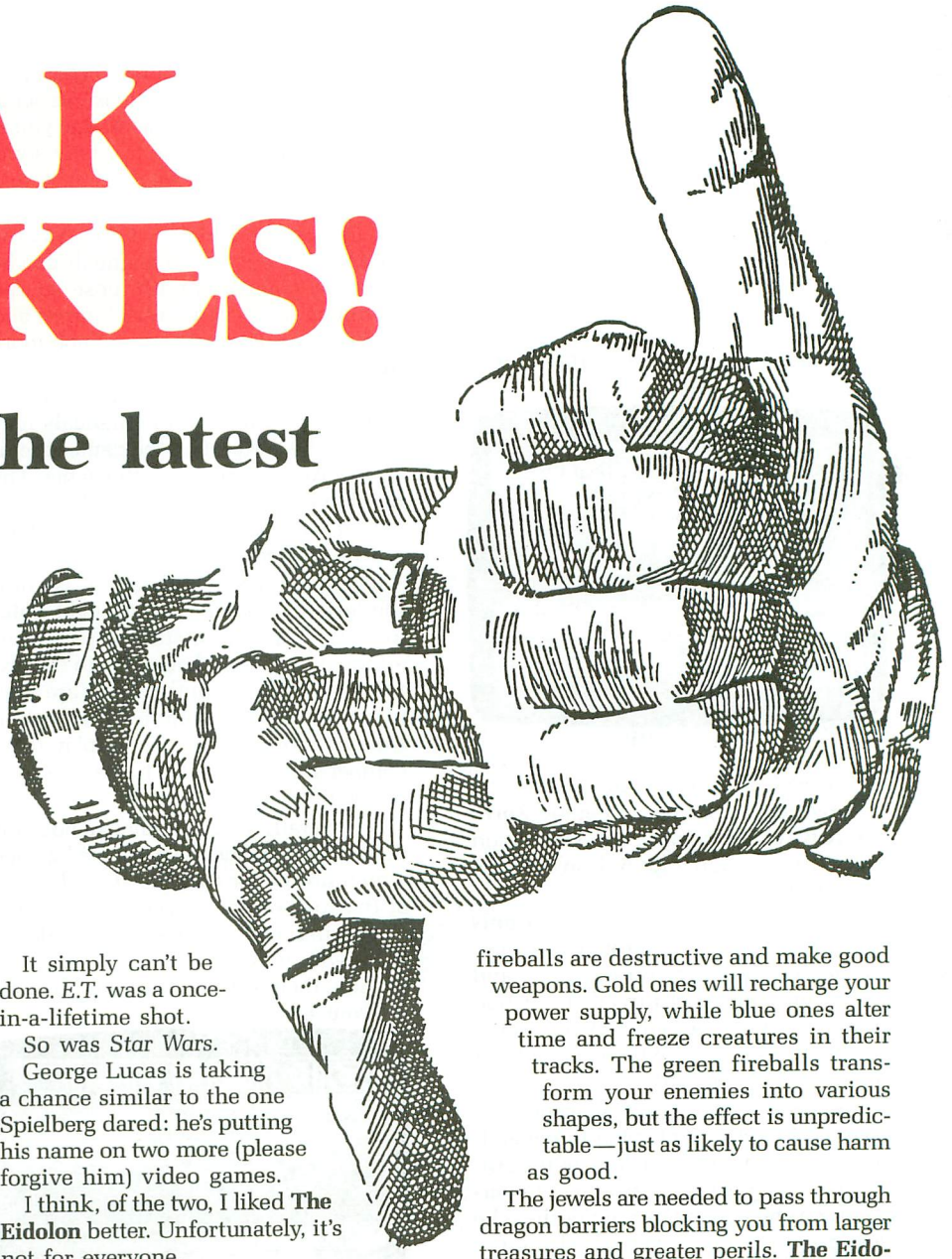
```
            PLA             ; GET # OF RECORDS
            STA NREC
            PLA             ; IGNORE HI BYTE
            PLA             ; GET RECORD LENGTH
            STA RLEN
            PLA             ; SEARCH STRING ADR HI
            STA KADR+1
            PLA             ; SEARCH STRING ADR LO
            STA KADR
            PLA             ; IGNORE HI BYTE
            PLA             ; DEPTH OF SEARCH
            STA DEEP
;
; MAIN PROGRAM
; ------------
;
            LDX #$01        ; X IS RECORD COUNTER
START
            LDY #$00        ; COMPARE CHARACTER OF
CHECK
            LDA (SADR),Y    ; SEARCH STRING TO
            CMP (KADR),Y    ; CHARACTER OF DATA
            BNE NEXT        ; IF NOT= THEN NEXT
            INY             ; POINT TO NEXT CHAR
            CPY DEEP        ; REACHED DEPTH?
            BNE CHECK       ; NO, CHECK NEXT CHAR
            BEQ END         ; YES! FOUND IT!
NEXT
            CPX NREC        ; LAST RECORD?
            BEQ NOGOOD      ; YES, NO MATCH FOUND
            INX             ; NO, +1 TO REC COUNT
            LDA SADR        ; CALCULATE THE ADR
            CLC             ; OF THE NEXT RECORD
            ADC RLEN        ; THEN GO BACK TO
            STA SADR        ; TRY AGAIN
            BCC START
            INC SADR+1
            BCS START
NOGOOD
            LDX #$00        ; NO MATCH, 0 TO BASIC
END
            STX BRET        ; VALUE TO BASIC
            LDA #$00        ; HI BYTE TO BASIC
            STA BRET+1
            RTS             ; BACK TO BASIC
```


Assembly listing.

```
; STRSORT.SRC
;
; ********************************
; *          SORT ROUTINE        *
; *                              *
; *      A=USR(A,B,C,D,E)        *
; *                              *
; *     A=ADR OF ROUTINE         *
; *     B=ADR OF STRING          *
; *     C=RECORD LENGTH          *
; *     D=DEPTH OF SORT          *
; *     E=# OF RECORDS           *
; *                              *
; ********************************
;
            *=  $5000
;
; EQUATES
; -------
;
SADR        =   $CB         ; ADR OF RECORD
HSADR       =   $0600       ; ADR OF DATA STRING
RLEN        =   $0602       ; RECORD LENGTH
DEEP        =   $0603       ; DEPTH OF SEARCH
TEMP        =   $0607       ; TEMP DATA BUFFER
NREC        =   $0605       ; # OF RECORDS
SSADR       =   $CD         ; ADR OF 2ND RECORD
FLG         =   $0606       ; SWITCHED FLAG
;
; INITIALIZE
; ----------
;
            PLA             ; # OF ARGUMENTS
            PLA             ; GET STRING ADR HI
            STA HSADR+1
            PLA             ; GET STRING ADR LO
            STA HSADR
            PLA             ; IGNORE HI BYTE
            PLA             ; GET RECORD LENGTH
            STA RLEN
            PLA             ; IGNORE HI BYTE
            PLA             ; GET DEPTH OF SORT
            STA DEEP
            PLA             ; IGNORE HI BYTE
            PLA             ; GET # RECORDS
            STA NREC
;
; MAIN PROGRAM
; ------------
;
START
            LDA HSADR+1     ; BEGINNING
            STA SADR+1      ; OF DATA STRING
            STA SSADR+1
            LDA HSADR
            STA SADR
            CLC
            ADC RLEN        ; CALCULATE ADR OF
            STA SSADR       ; NEXT HIGHEST RECORD
            BCC START2
            INC SSADR+1
START2
            LDA #$00        ; INITIALIZE FLAG
            STA FLG
            LDX #$01        ; RECORD COUNT
NCMPR
            LDY #$00        ; POINT TO 1ST CHAR
CMPR
            LDA (SADR),Y    ; COMPARE CHARACTERS
            CMP (SSADR),Y   ; IN EACH RECORD
            BCC NEXT        ; IN RIGHT ORDER
            BNE SWITCH      ; IN WRONG ORDER
            INY             ; EQUAL, SO NEXT CHAR
            CPY DEEP        ; REACHED SORT DEPTH?
            BNE CMPR        ; NO, CHECK NEXT CHAR
NEXT
            INX             ; YES, INC REC COUNT
            CPX NREC        ; LAST RECORD?
            BEQ CHKFLG      ; YES - SWITCHED?
            LDA SADR        ; NO, FIND ADDRESSES
            CLC             ; OF NEXT TWO RECORDS
            ADC RLEN        ; TO COMPARE
            STA SADR
            BCC NEXT2
            INC SADR+1
NEXT2
            LDA SSADR
            CLC
            ADC RLEN
            STA SSADR
            BCC NCMPR
            INC SSADR+1
            BCS NCMPR
SWITCH
            LDA #$01        ; SET SWITCH FLAG
            STA FLG
            LDY #$00        ; MOVE 1ST RECORD
SWITCH2
            LDA (SADR),Y    ; TO TEMPORARY
            STA TEMP,Y      ; STORAGE
            INY
            CPY RLEN
            BNE SWITCH2
            LDY #$00
SWITCH3
            LDA (SSADR),Y   ; MOVE 2ND REC
            STA (SADR),Y    ; TO 1ST RECORD'S
            INY             ; POSITION
            CPY RLEN
            BNE SWITCH3
            LDY #$00
SWITCH4
            LDA TEMP,Y      ; MOVE 1ST RECORD
            STA (SSADR),Y   ; FROM TEMP STORAGE
            INY             ; TO 2ND RECORD'S
            CPY RLEN        ; POSITION
            BNE SWITCH4
            BEQ NEXT
CHKFLG
            LDA FLG         ; CHECK FLAG
            BNE START       ; SET, SO SORT AGAIN
END         RTS             ; NOT SET, SORT DONE
```

PANAK STRIKES!

Reviews of the latest software



by Steve Panak

I've been accused—and rightfully so, lately—of mellowing. But it's not my fault; it's been the quality of the games—they've forced me into a corner.

It's not that they've been particularly good (although, amazingly, quite a few have been).

It's that most of them have been mediocre. Just the usual shoddy, substandard, abominable atrocities. Still, it's been a long time since I've seen anything outrageous enough to snap me out of the dazed stupor I'd lapsed into.

Until today.

THE EIDOLON
64K Disk $29.95

KORONIS RIFT
48K Disk $29.95
by Lucasfilm Games Division
EPYX, INC.
1043 Kiel Court
Sunnyvale, CA 94089

Last fall, Steven Spielberg brought his "Amazing Stories" to TV. It, along with the "Twilight Zone," tempted me back in front of the colorful cathode tube. Once again, its bruise-purple glow gushed from my darkened windows.

The critics panned "Amazing Stories." They didn't like the show, which had, I thought, one of the best opening episodes of the TV season. Evidently, they wanted an *E.T.* every night.

It simply can't be done. *E.T.* was a once-in-a-lifetime shot.

So was *Star Wars.*

George Lucas is taking a chance similar to the one Spielberg dared: he's putting his name on two more (please forgive him) video games.

I think, of the two, I liked **The Eidolon** better. Unfortunately, it's not for everyone.

First, it's not for owners of the old 800 (me, for instance). The game, like so many on the market today, requires at least 64K. It also isn't for those people who might think they're getting *Star Wars.* They're not. What they *are* getting is a rather imaginative video game.

The Eidolon is a strange craft, which harnesses mystical powers and transports its occupant to another dimension. In this netherworld, you'll encounter such exotic creatures as Rotoflies, Biter Birds and dragons.

Your object here is to collect as many jewels as you can before time runs out. You must also collect fireballs of different colors, each with its own power. Red fireballs are destructive and make good weapons. Gold ones will recharge your power supply, while blue ones alter time and freeze creatures in their tracks. The green fireballs transform your enemies into various shapes, but the effect is unpredictable—just as likely to cause harm as good.

The jewels are needed to pass through dragon barriers blocking you from larger treasures and greater perils. **The Eidolon**'s instrument panel displays energy and time remaining, level of play, and the direction you're facing, as well as jewels and fireballs at your disposal.

The Eidolon.

The caves form a maze, and moving through it is maddening. Your proximity meter shows how close you are to a dragon, and this leads you from one deadly confrontation to another—from a dead end to your own end. The deeper you dive into the caverns, the stranger their creatures are. The levels get progressively more difficult, and each is guarded by a dragon with a different vulnerability. Discovering their Achilles heels is half the fun. The other half is exploiting this knowledge.

Koronis Rift.

Those of us with 800s aren't completely forgotten. Everyone can enjoy **Koronis Rift**. Unfortunately, the words *can* and *will* are mutually exclusive as applied to this game.

While it's visually attractive, **Koronis** doesn't cover any new ground and, unfortunately, stumbles on the worn and weary ground it does tread. I felt that even the massive amount of detail couldn't raise it above the level of "just another shoot-'em-up."

You're an outer space scavenger in search of anything of value in general, and the weapons technology of the Ancients (a long-extinct super race) in particular. When your alarm unleashes the loudest din you've ever heard, you're positive the treasures of the Ancients are near.

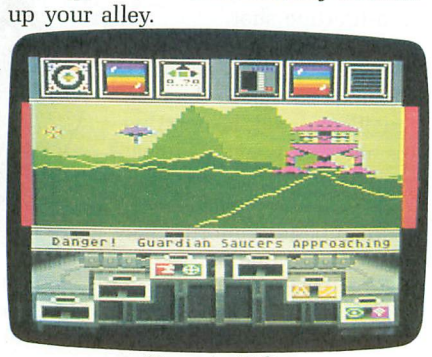
Using a remote-controlled surface rover, you loot eons-old space hulks for valuable weapons systems, such as lasers and radars. Thwarting you every step of the way are the Guardians, a race of genetically engineered warriors created by the Ancients to guard their stockpiles of invaluable technology.

Once you've collected a number of systems, you must analyze them using the Psytek 7500 Series Science Droid System Analyzer. The modules move past the droid, who may, at your whim, dismantle or analyze the system. You decide whether to sell it or place it in your ship, increasing your power. The modules provide help for the quest, as some are necessary to complete the game. This end may not come for quite some time, but, in an unusual display of mercy, a save game feature is included.

The documentation with each of the games is excellent. The manuals can be favorably compared with those offered by Infocom, due to their rich prose and background detail. They're much more than instruction manuals, and create a world which surrounds the ultimate game. Unfortunately, the manuals may actually be better than the games themselves. As for explaining the game, **The Eidolon**'s manual left more to the imagination, and, as a result, the game took a little longer to learn.

The graphics on both games is nothing less than excellent. I thought **Eidolon** was superior in this department, although this is probably due to that extra 16K of power. Both games are easy to control, though both require break-in time before you'll be able to play them competently. This is due to their incredible complexity.

So, overall, *Star Wars* they're not. But they're well planned and detailed games which are better than most of the rest of the stuff on the market. If you want something really different, **The Eidolon** is for you, while if you enjoy space exploring, then **Koronis Rift** may be more up your alley.



Koronis Rift.

MICRONOVELS
2808 S. 12th Street
Milwaukee, WI 53215
48K Disk $19.95

Well, this is it; you've all been waiting for it. I bear my fangs and go for the jugular, like some jungle beast—without remorse, shredding the baby gazelle.

It really pains me, though, to have to do this. That's because I have to denounce games which contain one of the few human qualities I hold in high regard: creativity, expressed as originality.

Well, a box full of these games fell into my hands a few weeks ago, and I finally got around to taking a good look at them. Unfortunately, my time could have been better spent.

MicroNovels is written using the AdventureWriter program from **Codewriter**. I've never been able to test this program, but it apparently allows you to write an adventure game, someone before you having done the difficult programming. Unfortunately, that programming wasn't done too well, and that's what makes **MicroNovels** so poor.

The first is *Star Voyage*. It's basically a parody of "Star Trek," changing the legendary characters' names to protect the innocent. Each disk contains two full adventures, and I tested two disks. I also looked at another series, *The Casebook of Hemlock Soames*, a take-off on poor old Sherlock.

All the **MicroNovels** games were entertaining and, while not up to par with Infocom games, they were quite a bit better than the old Scott Adams series. I liked *Star Voyages* a little more, since, being a trekkie, I caught all the inside jokes. The text is imaginative—not at all amateurish—and it draws you right in and drags you right through the story.

But the program holds you back. The vocabulary isn't very large, and I found myself spending a lot of time trying to think up the current synonym for the action I wanted to take. It was simply too hard to play to have fun. The puzzles were very difficult, and consisted of trying to find the correct word to use. It was extremely frustrating. There aren't many rooms to the adventures, and most seem to reside in memory.

The documentation was virtually nonexistent, just a sheet of paper telling how to boot and save the game, along with a few suggested commands. There's no background information upon which **MicroNovels** is built, nothing to make you care.

The price is certainly right. However, these games can only be obtained via mail order from the address above (and I'm not sure about *Hemlock*'s availability). So, while the price includes postage and handling, I presume it doesn't include a money-back guarantee. You're flying blind unless you know someone who's already ordered one.

If you feel the urge to adventure—and if you can afford to spend more on that

urge—I would recommend the following game.

A MIND FOREVER VOYAGING
by Steven Meretzky
INFOCOM
125 Cambridge Park Drive
Cambridge, MA 02140
128K ST Disk $44.94

Well, I have to admit it. I hate to admit it, but I have to. I made a mistake. You must understand; let me explain before you pull my plug.

Like a two-pack-a-day smoker who just heard Dan Rather say that cigarettes would be illegal tomorrow, my ravings were those of a man gone mad.

I thought that Infocom was cutting me off, cutting all 8-bit users off. I was wrong, as Brian Moriarty pointed out in last issue's **Reader Comment**. They've simply created two lines of games, Classic and Plus, the latter being larger and requiring at least 128K of memory.

This first Plus game is included in our evaluations for your perusal. I hope you'll try it and like it as much as I did.

Unlike the majority of Infocom's interactive fiction, **A Mind Forever Voyaging** (**AMFV**) threw me a curveball that kept me swinging for hours. They've forsaken their usual lighthearted approach to adventure and have substituted in its stead a more serious tone, one reminscent of the great science fiction stories of our time.

Imagine yourself living every day of your life exactly as you have: growing up, going to school, loving, losing—a normal life. Imagine suddenly being told you're not who you always thought you were—what's worse, you're not *what* you thought you were. Imagine being told you're a sophisticated computer program; your life has been a simulation. In **AMFV** you're told just that.

You're Prism, a supercomputer, one of the first utilizing true artificial intelligence, and you've spent the last few years instilling your sense of self awareness. You've broken out of your life-simulation years early to perform a much more important duty: to predict the future.

The arms race has degenerated into a miniaturization race, where the goal is to build weapons so small that they can be smuggled into enemy cities. This has forced the creation of a police state. The only solution seems to be Senator Richard Ryder's Plan for Renewed National Purpose. But it's untested. That's where you come in.

From this point, as always with Infocom, you're entirely on your own. However, this time, in addition to being in a different vein, the game is also on a much grander scale. It's the most complex game Infocom has yet unveiled. The statistics speak for themselves: several hundred locations, a vocabulary of 1800+ words, 128K minimum memory and a fast response time.

After testing the latest for the 800 (**Spellbreaker**, the final chapter in the **Enchanter** trilogy), **AMFV**'s reactions felt like greased lightning to me—no waiting, no noisy disk grinding.

Although as a computer life may be rather restricted, you have a number of modes you can enter. Library mode allows you to access information, much the way Whiz did in **Suspended**. Likewise, the interface mode lets you control the outside environment. Communication mode lets you see and hear in the complex, while sleep mode rests your circuits. Infocom regulars will be most at home in the simulation mode, which allows you to move around your environment exactly as you would in any other Infocom game.

AMFV itself is less puzzle oriented and more story oriented. This is not to say that you're merely a passive observer—just that your involvement is a little different. Keep your eyes and ears open, and don't bother to look over your shoulder for any evil grues.

With each new Infocom game, I wonder if I'll be writing the epitaph stored away in my mind: the one that says they've failed, they've reached their peak and are on the downside of the marketing curve. Well, with the addition of the Plus line of games, that time will be long in coming. **A Mind Forever Voyaging** takes the Infocom concept to the next logical step in its evolution—and takes you on a voyage you'll never forget.

Next month, we'll look at **Championship Lode Runner**, plus a few games available only through the mail from some small, individual creators.

The author would like to thank Magic One Computer Shop of Barberton, Ohio, for their invaluable assistance in the creation of this article.

Steve Panak is a banking computer operator and free-lance writer living in northeastern Ohio. He holds a B.S. and currently attends law school, where he develops software to teach complex legal concepts. In his spare time, he enjoys computer games.

XLent XPress

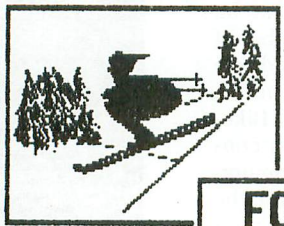
ISSUE 1 VOLUME 1 DEMONSTRATION AND ADVERTISEMENT BY XLent Software WINTER EDITION 86

CREATILITY PRINTWARE

XLENT PRESENTS RUBBER STAMP

FEATURES
EXPAND/SHRINK
ROTATE/MIRROR
REFLECT- FLIP
MULTI-FILLS
ELLIPSE/BOX
TRUE CIRCLE
ICONIZE PIX
16X16 FONTS
48 TEXT SIZES
CREATE FONTS
CONVERT ICONS
AND MUCH MORE

CREATE PIX, TEXT, &/OR ICONS! USE other ware pix. Add 16x16 HI-RES PROPORTIONAL text. CONVERT other ware icons into TYPESETTER icons.



← LEFT: ICON FROM PICTURE SHRINK

We can't list all of the features in RUBBER STAMP here. RUBBER STAMP is both a fast, fun graphics/text program AND a comprehensive program to integrate with TYPESETTER.

RUBBER STAMP HAS IT ALL!!

$29.95

+$2.00 SHIPPING

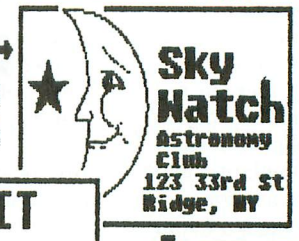
TYPESETTER

CREATIVITY & RESOLUTION

WHAT CAN TYPESETTER DO?

THIS NEWSLETTER AD IS JUST ONE EXAMPLE!! Create forms, labels, signs, letterhead, cards, OR HIGHLY DETAILED GRAPHICS FULL PAGE HARDCOPY! 48K controls over 400,000 pixels. DIFFERENT FROM ALL OTHER ATARI SOFTWARE!!

RIGHT: → HIGHEST RES. PIX AND TEXT TOGETHER



More resolution than most 8/16 bit computers!

130 XE Version on Same Disk MORE FEATURES!

ONLY AVAILABLE FOR ATARI COMPUTERS!!

$34.95

+$2.00 SHIPPING

FOR ALL 8-BIT ATARI COMPUTERS

EPSON, NEC, AND COMPATIBLES

INTEGRATED WARE

USE RUBBER STAMP SCREENS IN TYPESETTER AND PAGE DESIGNER

USE MEGAFONT II+ TO PRINT RUBBER STAMP SCREENS

USE CUSTOM FONTS IN ALL PROGRAMS

TYPESETTER ST

$39.95 for the 520ST + $2 shipping

PAGE DESIGNER

FULL PAGE, MEDIUM RES. DESIGN & PRINT PROGRAM

"....a pretty nifty package..." CURRENT NOTES

DESIGN full-page printouts ON-SCREEN. COMBINE text, borders, & pix from other ware. EDIT in 40 or 80 columns. Mix 40 column Fonts Graphics Editor puts final touches on layout.

PAGE DESIGNER allows anyone QUICK, EASY layout for ADS, REPORTS, NEWSLETTERS, ETC. Two Graphics 8 screens show you EXACTLY how your design will look when printed.

Not the resolution of TYPESETTER, but PERFECT when highest resolution is not essential.

$29.95 + $2.00 SHIPPING

MEGAFONT II+

TEXT FILE PRINT IN ANY FONT

SCREEN PRINT IN UP TO 4 SIZES

"...a good addition to anyone's library." -ANALOG COMPUTING

```
10 A="LVre
20 ? "help"
```

PRINT PROGRAM LISTING WITH SPECIAL CHARS.

If you are looking for a FAST PROGRAM LISTER that PRINTS ALL SPECIAL CHARACTERS, MEGAFONT is the program for you.

If you are looking for a VARIABLE SIZE PICTURE PRINTER, MEGAFONT II+ is for you.

COMING SOON For the ST

$24.95 + $2.00 SHIPPING

24 HOUR ORDER PHONE

(703) 644-8881

XLent Software

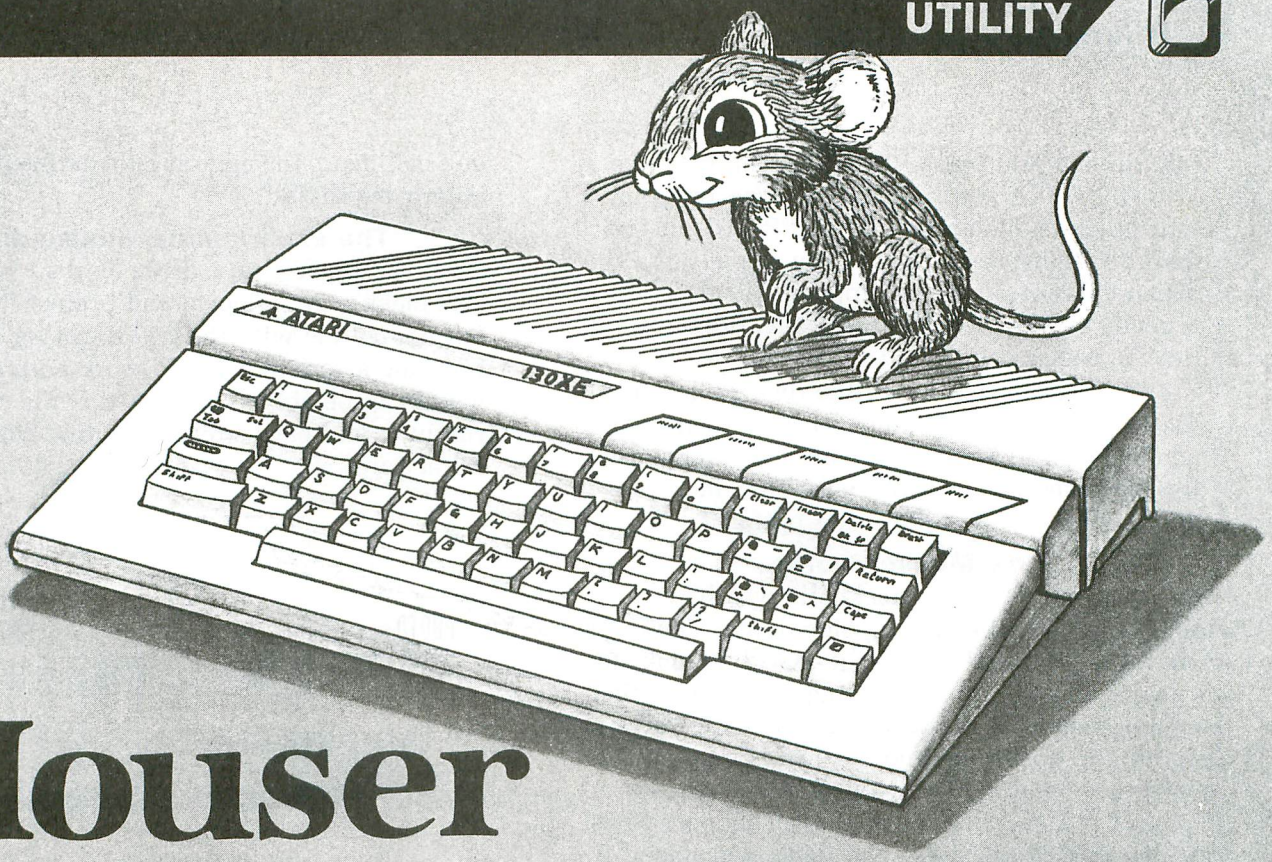
P.O. BOX 5228, DEPT. B
Springfield, VA 22150

DEALER INQUIRIES WELCOME

C.O.D.: Additional
VA RES.: Add 4% tax

This ad was created by Ira Brickman using TYPESETTER & White Lion Software's GRAPHICS LIBRARIES

CIRCLE #140 ON READER SERVICE CARD



Mouser

A mouse for your 8-bit Atari.

by Bernhard Engl

With the availability of the Atari 520ST, the 8-bit Ataris might be doomed to become dust-collecting relics of ancient times. Their fate will depend on the availability of new software. It now seems that this will be up to the users.

It will be a pity if the small Atari dies—the fun in computing is squeezing "impossible" things out of limited machines. That's the real art. After all, every below-average programmer could smash any 6502 problem with 68000 muscle power.

A project promising a new lease on life for the 8-bit Atari would be one that brought it the amazing world of mice and icons. This article illustrates the first step needed to fill such a tall order.

Typing it in.

Before typing anything, look at the listings accompanying this article.

Listing 1 is the BASIC data and data checking routine. This listing is used to create a disk version of **Mouser**. The data statements are listed in hexadecimal (base 16), so the program will fit in 16K disk systems.

Listing 2 is the assembly language source code for **Mouser**, created with the OSS MAC/65 assembler. You *don't* have to type this listing. It is included for those readers interested in assembly language.

Follow the instructions below to make your disk version of **Mouser**.

Instructions.

1. Type Listing 1 into your computer, using the BASIC cartridge and verify your typing with **Unicheck** (see page 10).

2. Type *RUN* and press RETURN. The program will ask:

```
MAKE CASSETTE (0), OR DISK (1)?
```

Type *1* and press RETURN. The program will begin checking the data lines, printing the line number of each statement as it goes. It will alert you if it finds any problems. Fix incorrect lines and re-RUN the program, if necessary, until all errors are eliminated.

3. When all the data lines are correct, you will be prompted to *INSERT DISK WITH DOS, PRESS RETURN*. Put a disk containing DOS 2.0S

into drive 1 and press RETURN. The message *WRITING FILE* will appear, and the program will create a file called MOUSER.OBJ on the disk, displaying each data line number as it goes. When the *READY* prompt appears, the program is ready to run. Be sure the BASIC program is SAVEd before continuing.

4. To run **Mouser**, insert the disk containing the MOUSER.OBJ file into drive 1. Turn your computer OFF, remove all cartridges and turn the computer back ON. When the DOS menu appears, load the program using option L.

How Mouser works.

In principle, mice are trackballs which were reduced in scale and turned upside down. The heart of the mouse is a steel ball, about an inch in diameter, covered with a thin rubber coating for better friction. Two tiny steel rollers make contact with the ball and separate the horizontal and vertical components of mouse movement. Conversion of roller rotation to electronic pulses is done by an optical chopper wheel and two phototransistors. The light is provided by one (or more) LEDs.

The friction ball can be removed from the mouse for cleaning, so you can look into the device without voiding your warranty. You can't see the chopper wheels and the optoelectronics. I don't recommend further disassembly—just look at Figure 1 to know what to expect.

Interface considerations.

The plug for the mouse matches the pinout for the Atari's controller jacks. This means you can plug the mouse into the Atari without any hardware modifications—no smoke will spill, and the mouse won't be hurt. The only incompatibility is with the right mouse button.

It pulls down the POTA line, which is normally used for the paddles. Unfortunately, this won't charge up the internal pot capacitor, so you can't sample the button. It would be possible to solder a pull-up resistor into the mouse, but I don't recommend this for mice under warranty, inexperienced users, or those without proper tools. Besides that, one button seems to be sufficient for any possible application.

The left button can be sampled in the same way as the joystick button, and the pulses for mouse movement can be sampled in the same way as the joystick direction. The only interfacing work that remains now is to interpret these pulses as up, down, left or right commands for a pointing arrow on the screen, keeping track of the X- and Y-coordinates, and calling up the appropriate routine if the joystick button is pressed.

The experimental mouse driver.

A full-fledged mouse driver for the Atari must: load in a clean way; relocate and protect itself as needed; not collide with the OS, BASIC or DOS; have a pointing arrow that would work with every graphics mode; be able to be moved while programs are running; and have a button routine with some useful action.

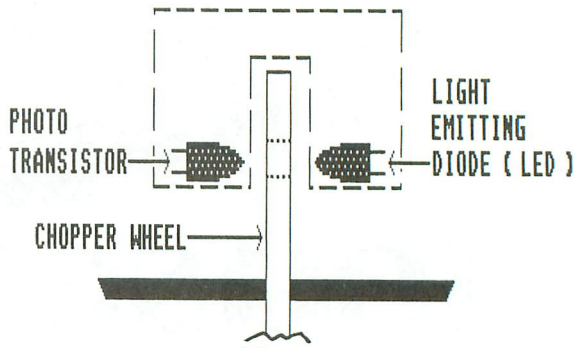


Figure 1.

My experimental **Mouser** (shown in Listing 2) does not meet all these specs. It was written under intense time pressure, most of the work was done without knowing what kind of pulses would come out of the mouse, and I had only one hour to do the final interfacing and testing. For these reasons, the program is neither optimum nor complete. Despite that, the basic mouse interface works quite well—you can move the arrow fast and accurately (even if a BASIC program is running), and it has the same "feel" as the one in the 520ST.

I believe I could write a perfected and polished version in a weekend or so, but I'm so overloaded with other work that it will have to wait until next year. I think there are many Atari users who'll be able to use my program as a guideline to turn out the real thing much faster and better than I could.

How the program works.

The pointing device, the arrow, is made from the missiles used as a fifth player. Since a big arrow looked very crude, I reduced it in size. Two missiles could be used for other tasks if the code were adjusted accordingly. But, as a general rule, it's not smart to pretty up working programs if you have no chance to test them afterwards.

Let's look at the listing now. The initialization routine (Lines 270 to 670) is standard. It sets up the player/missile graphics and installs the interrupt routines

that will keep track of the mouse movement for us.

These routines (Lines 860 to 1255) are mostly standard. You would write a primitive joystick driver in exactly the same way. In fact, the code at Lines 1000-1260 really *is* a joystick driver. It just looks for direction bits in MOMOVE in the order up/down/left/right, checks the coordinates in MOUSEX and MOUSEY for screen boundaries, updates them and moves the arrow accordingly. I used this approach because I had to test most of **Mouser** before I even saw the mouse. The conversion from mouse to joystick logic is done in Lines 900 to 980.

There are a few complications caused by the nature of the mouse. It produces pulses at a high rate, and, if the program misses only one pulse, the arrow will do funny things because the movement direction can't be determined anymore. The vertical blank interrupt occurs only every 20 milliseconds (PAL) or 16.6 milliseconds (NTSC), which won't take enough samples of the mouse port.

To get more samples per second, I've added lots of display list interrupts—one on every other display list instruction. In graphics 0, this translates to a sample per millisecond, which should be fast enough even for mouse wizards. The display list interrupts are set up in Lines 615-670, and you'll have to adapt this routine if you want to use any other graphics mode.

With the DLIs, another complication is invited: the arrow might be moved at the same time it's being drawn onto the screen. This will make it jerk and twist, so I've added a NOJERK routine, to delay movement if the arrow is drawn at critical times.

These routines would be enough to move the arrow around with the mouse, but something useful should happen if the button is pressed. So, once for every press, the routine BUTTON is called, which finds the coordinates of the arrow in MOUSEX (0. . . 159) and MOUSEY (0. . .95). If you'd like to write 10,000 lines of code, you could add parts of your own user interface here.

Since all interrupts are disabled, you have time to do complicated things. For compactness, I added only thirteen lines as an example. They move the cursor to the character pointed to by the arrow, but only after the next screen editor call, since it fakes a keyboard entry.

Now it's time to give you some hints about how the mouse-joystick converter works. It's the heart of the whole driver, and its operation looks rather obscure. I don't claim it's the optimum—perhaps you could do it in half of the lines.

For horizontal and vertical mouse movement, it generates a direction bit and a clock bit. This is done by comparing the state of the phototransistors with the previous state, in some appropriate fashion. I've illustrated the idea behind movement detection with a pair of phototransistors in Figure 2. In the beginning, both phototransistors (A and B) will be illuminated as shown. As the wheel turns clockwise, A will get dark first, followed by B. Then, A will be illuminated again, followed by B, and so on. Turning counterclockwise yields the reverse sequence.

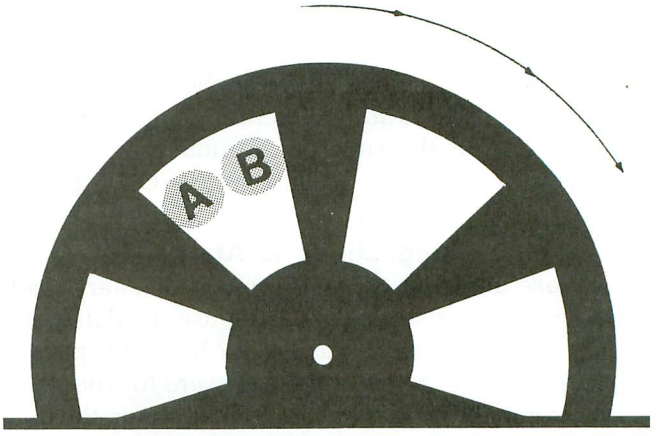


Figure 2.

The algorithm is as follows: let's assign the boolean variables A and B to phototransistors A and B, respectively, at sample n. The boolean variables A' and B' shall represent the previous sample *n-1*. Then, for sample *n*, the direction is given by the function D = A' EOR B, and the clock bit is given by C = (A EOR A') or (B EOR B'). After computing these bits, we can assign A' = A and B' = B in preparation of the next step.

The code at Lines 900 to 980 does this for both the horizontal and vertical components simultaneously. A lookup table is then used for the translation to the joystick logic.

Note that, for each step, a sample from PORTA is taken only once and kept in the temporary variable PRTEMP, because the port might change during the execution of this routine—with funny results if the samples were taken repeatedly.

Some notes on typing in assembly.

I've used OSS's MAC/65 to do the work—I think it's by far the best assembler/editor package ever sold for 6502 machines. If you want to do larger projects, it's a must. However, for the readers who have to use inferior assemblers, I didn't make use of any of MAC's special features, nor macro and equates libraries. You

can type in the program as is, even with the Atari Editor/Assembler cartridge.

After assembly, you should enter *DUP* and do a binary save MOUS.EXE,5000,516D,,5000. The program can be started by loading this file. Be aware that it is not linked to the OS initialization procedures in any way, so it will blow up if you leave DUP.

Suggestions for improvements.

You might want to add some code to get a clean initialization, in case of DUP entry/exit and SYSTEM RESET. Don't forget to reassemble the program to a good location and protect it by setting the OS RAM pointers.

You might want to free the unused missiles. If you've done this, you may want to replace the crude button routine by a doodle-type application for graphics 7. Don't forget to adjust the routine at 615-665 accordingly. And the superprogrammers among you might want to start writing a sort of GEM for the 6502.

Suggestions to Atari.

In case some responsible person at Atari reads this—how about making the mouse available at a reasonable price to all Atari users? It would spur the development of mouse-based software for the small Atari and make it the toughest competitor on the 8-bit market. Mice and icons should not be exclusively tied up to the 16-bit world. The ideas behind them are far too good to waste this way. Let's liberate mice and icons for all machines—regardless of their price class.

Bernhard Engl is currently studying Electrical Engineering at the Technical University in his native city of Munich, Germany. A computer enthusiast for ten years, he reports he currently owns 2½ Ataris.

Listing 1.
BASIC listing.

```
10 REM *** MOUSER ***
20 TRAP 20:? "MAKE CASSETTE (0), OR DI
SK (1)";:INPUT DSK:IF DSK>1 THEN 20
30 TRAP 40000:DATA 0,1,2,3,4,5,6,7,8,9
,0,0,0,0,0,0,0,10,11,12,13,14,15
40 DIM DAT$(91),HEX(22):FOR X=0 TO 22:
READ N:HEX(X)=N:NEXT X:LINE=990:RESTOR
E 1000:TRAP 120:? "CHECKING DATA"
50 LINE=LINE+10:? "LINE:";LINE:READ DA
T$:IF LEN(DAT$)<>90 THEN 220
60 DATLIN=PEEK(183)+PEEK(184)*256:IF D
ATLIN<>LINE THEN ? "LINE ";LINE;" MISS
ING!":END
70 FOR X=1 TO 89 STEP 2:D1=ASC(DAT$(X,
X))-48:D2=ASC(DAT$(X+1,X+1))-48:BYTE=H
EX(D1)*16+HEX(D2)
80 IF PASS=2 THEN PUT #1,BYTE:NEXT X:R
EAD CHKSUM:GOTO 50
90 TOTAL=TOTAL+BYTE:IF TOTAL>999 THEN
TOTAL=TOTAL-1000
100 NEXT X:READ CHKSUM:IF TOTAL=CHKSUM
 THEN 50
110 GOTO 220
120 IF PEEK(195)<>6 THEN 220
130 IF PASS=0 THEN 170
140 IF  NOT DSK THEN 160
150 PUT #1,224:PUT #1,2:PUT #1,225:PUT
 #1,2:PUT #1,0:PUT #1,80:CLOSE #1:END
160 FOR X=1 TO 67:PUT #1,0:NEXT X:CLOS
E #1:END
170 IF  NOT DSK THEN 200
180 ? "INSERT DISK WITH DOS, PRESS RET
URN";:DIM IN$(1):INPUT IN$:OPEN #1,8,0
,"D:MOUSER.OBJ"
190 PUT #1,255:PUT #1,255:PUT #1,0:PUT
 #1,80:PUT #1,148:PUT #1,81:GOTO 210
200 ? "READY CASSETTE AND PRESS RETURN
";:OPEN #1,8,128,"C:":RESTORE 230:FOR
X=1 TO 40:READ N:PUT #1,N:NEXT X
210 ? :? "WRITING FILE":PASS=2:LINE=99
0:RESTORE 1000:TRAP 120:GOTO 50
220 ? "BAD DATA: LINE ";LINE:END
230 DATA 0,4,216,79,255,79,169,0,141,4
7,2,169,60,141,2,211,169,0,141,231,2,1
33,14,169,56,141,232,2
240 DATA 133,15,169,0,133,10,169,80,13
3,11,24,96
1000 DATA A9518D07D4A27FA9009D8051CA10
FAE8A005B98B509D8F51E88810F6A9008DC702
A9118D6F02A000A2088C9250,361
1010 DATA 207950AD00D38D9350A998A2508D
00028E0102A907A09DA250205CE4A9268D2F02
8D00D4A9018D1DD0AD3002AE,979
1020 DATA 3102852C862DA9F0A001912CA9C2
A003912CA982A007912CC8C8C01D90F8608E91
508A186930A2039D04D06902,947
1030 DATA CA10F86000108A0C0E00000000000
0000004B8A489848A9008D0ED4AD00D38D9550
AD93500A4D9550290A8D9750,36
1040 DATA AD93504D95508D96504A0D965029
050D9750AABD52518D9750AD95508D93504E97
50900BAE9250F0064E9750CA,5
1050 DATA 100D4E9750901AAE9250E8E060B0
128E9250203F51A005B98B509D8F51E88810F6
4E9750900CAE9150F007CA20,128
1060 DATA 3F5120795O4E9750900EAE9150E0
9FB007E8203F51207950AD8402AA49012D9450
8E9450F003206251A9C08D0E,742
1070 DATA D44C62E4AD0BD438E916CD9250B0
076908CD9250B0EE60000400080105010900040
00080206020AAD91504A4AF0,577
1080 DATA 108555C655AD92504A4A8554A987
8DFC02600000000000000000000000000000000
0000000000000000000000,653
1090 REM * 405 BYTES
```

CHECKSUM DATA.

(see page 10)

```
10 DATA 685,351,496,811,423,729,200,60
3,555,573,694,613,29,205,218,7185
160 DATA 763,198,12,636,491,30,155,982
,173,846,775,664,591,716,667,7699
1060 DATA 645,576,895,518,2634
```

Listing 2.
Assembly listing.

```
0100      .OPT NO EJECT
0105      .TAB 13,17,29
0110 ;
0115 ;--------------------------------
0120 ;
0125 ; ATARI 800 XL  MOUSE DRIVER
0130 ;  written by Bernhard Engl
0135 ;
0140 ;--------------------------------
0145 ;
0150 PMBASE = $D407
0155 COLOR3 = $02C7
0160 GPRIOR = $026F
0165 PORTA = $D300
0170 SDMCTL = $022F
0175 GRACTL = $D01D
0180 HPOSM0 = $D004
0185 STRIG0 = $0284
0190 VDSLST = $0200
0195 NMIEN = $D40E
0200 SDLSTL = $0230
0205 SDLSTH = $0231
0210 ICSPRZ = $2C
0215 VCOUNT = $D40B
0220 DMACTL = $D400
0225 ;
0230 SETVBV = $E45C
0235 XITVBV = $E462
0240 ;
0245      *=  $5000
0250 MISSIL = *+$0180
0255 ARROW = MISSIL+15
0260 ;
0265 ;--------------------------------
0270 ;
0275 ;INITIALISATION ROUTINE: Set up
0280 ;PM Graphics, init coordinates,
0285 ;install VBLANK & DLI routine.
0290 ;
0295 INIT LDA #MISSIL/256
0300      STA PMBASE
0305 ;
0310 ;clear missiles and move
0315 ;arrow shape to them
0320 ;
0325      LDX #$7F
0330      LDA #0
0335 INI1 STA MISSIL,X
0340      DEX
0345      BPL INI1
0350      INX             zero X
0355      LDY #5
0360 INI2 LDA SHAPEA,Y
0365      STA ARROW,X
0370      INX
0375      DEY
0380      BPL INI2
0385 ;
0390 ;set missile color
0395 ;and fifth player/priority
0400 ;
0405      LDA #0
0410      STA COLOR3
0415      LDA #$11
0420      STA GPRIOR
0425 ;
0430 ;init (X,Y) coordinates
0435 ;
0440      LDY #0
0445      LDX #8
0450      STY MOUSEY
0455      JSR SETHPO
0460 ;
0465 ;get old input
0470 ;
0475      LDA PORTA
0480      STA OLPORT
0485 ;
0490 ;install DLI routine
0495 ;
0500      LDA #DLINT&$FF
0505      LDX #DLINT/256
0510      STA VDSLST
0515      STX VDSLST+1
0520 ;
0525 ;install VBLANK routine
0530 ;
0535      LDA #7        use VVBLKD
0540      LDY #VBLANK&$FF
0545      LDX #VBLANK/256
0550      JSR SETVBV
0555 ;
0560 ;missile DMA on
0565 ;
0570      LDA #$26
0575      STA SDMCTL
0580      STA DMACTL
0585      LDA #1
0590      STA GRACTL
0595 ;
0600 ;put DLI'S into display list
0605 ;WARNING: for graphics 0 only !
0610 ;
0615      LDA SDLSTL  get DLIST address
0620      LDX SDLSTH
0625      STA ICSPRZ
0630      STX ICSPRZ+1
0635      LDA #$F0
0640      LDY #1
0645      STA (ICSPRZ),Y
0650      LDA #$C2
0655      LDY #3
0660      STA (ICSPRZ),Y
0665      LDA #$82
0670      LDY #7
0675 INI3 STA (ICSPRZ),Y
0680      INY
0685      INY
0690      CPY #$1D
0695      BCC INI3
0700 ;
0705      RTS
0710 ;
0715 ;update missile HPOS registers
0720 ;
0725 SETHPO STX MOUSEX
0730      TXA
0735      CLC
0740      ADC #48
0745      LDX #3
0750 SETHP1 STA HPOSM0,X
0755      ADC #2
0760      DEX
0765      BPL SETHP1
0770      RTS
0775 ;
0780 ;arrow shape table
0785 ;
0790 SHAPEA .BYTE 0,$10,$A0,$C0,$E0,0
0795 ;
0800 ;variables
0805 ;
0810 MOUSEX *= *+1
0815 MOUSEY *= *+1
0820 OLPORT *= *+1
0825 OLBUTT *= *+1
0830 PRTEMP *= *+1
0835 MOTEMP *= *+1
0840 MOMOVE *= *+1
0845 ;
0850 ;--------------------------------
0855 ;
0860 ;VBLANK/DLI ROUTINE: track mouse
0865 ;movement and move arrow on
0870 ;screen accordingly.
0875 ;Call BUTTON routine if left
0880 ;mouse button pressed.
0885 ;
0890 DLINT PHA
0895      TXA
0900      PHA
0905      TYA
0910      PHA
0915 ;
0920 VBLANK LDA #0
0925      STA NMIEN
0930      LDA PORTA
0935      STA PRTEMP
0940      LDA OLPORT
0945      ASL A
0950      EOR PRTEMP
0955      AND #$0A
0960      STA MOMOVE
0965      LDA OLPORT
0970      EOR PRTEMP
0975      STA MOTEMP
0980      LSR A
0985      ORA MOTEMP
0990      AND #5
0995      ORA MOMOVE
1000      TAX
1005      LDA MOVTAB,X
1010      STA MOMOVE
1015      LDA PRTEMP
1020      STA OLPORT
1025 ;
1030      LSR MOMOVE  up?
1035      BCC VBLAN1  no
1040      LDX MOUSEY  upper boundary?
1045      BEQ VBLAN1  yes, ignore
1050      LSR MOMOVE  remove down bit
1055      DEX
1060      BPL VBLAN2  branch always
1065 ;
1070 VBLAN1 LSR MOMOVE down?
1075      BCC VBLAN4  no
1080      LDX MOUSEY  lower boundary?
1085      INX
1090      CPX #96
1095      BCS VBLAN4  yes, ignore
1100 ;
1105 VBLAN2 STX MOUSEY move up/down
1110      JSR NOJERK
1115      LDY #5
1120 VBLAN3 LDA SHAPEA,Y
1125      STA ARROW,X
1130      INX
1135      DEY
1140      BPL VBLAN3
1145 ;
1150 VBLAN4 LSR MOMOVE left?
1155      BCC VBLAN5  no
1160      LDX MOUSEX  left boundary?
1165      BEQ VBLAN5  yes, ignore
1170      DEX
1175      JSR NOJERK
1180      JSR SETHPO
1185 ;
1190 VBLAN5 LSR MOMOVE right?
1195      BCC VBLAN6  no
1200      LDX MOUSEX  right boundary?
1205      CPX #159
1210      BCS VBLAN6  yes, ignore
1215      INX
1220      JSR NOJERK
1225      JSR SETHPO
1230 ;
1235 VBLAN6 LDA STRIG0 button hit?
1240      TAX
1245      EOR #1
1250      AND OLBUTT
1255      STX OLBUTT
1260      BEQ VBLAN7  no
1265      JSR BUTTON
1270 ;
1275 VBLAN7 LDA #$C0 enable DLI & VBL
1280      STA NMIEN
1285      JMP XITVBV
1290 ;
1295 ;ANTI-JERK ROUTINE: don't move
1300 ;the arrow at the same time it
1305 ;is being written.
1310 ;
1315 NOJERK LDA VCOUNT
1320      SEC
1325      SBC #22
1330      CMP MOUSEY
1335      BCS NOJER1
1340      ADC #8
1345      CMP MOUSEY
1350      BCS NOJERK
1355 NOJER1 RTS
1360 ;
1365 ;movement translation table
1370 ;
1375 MOVTAB .BYTE 0,4,0,8,1,5,1,9
1380      .BYTE 0,4,0,8,2,6,2,10
1385 ;--------------------------------
1390 ;
1395 ;BUTTON ROUTINE: as an example,
1400 ;the cursor is positioned at
1405 ;the selected character.
1410 ;
1415 ROWCRS = $54
1420 COLCRS = $55
1425 CH   =   $02FC
1430 ;
1435 BUTTON LDA MOUSEX calc screen X
1440      LSR A
1445      LSR A
1450      BEQ BUTTO1  ignore column 0
1455      STA COLCRS  new cursor column
1460      DEC COLCRS  minus 1
1465 ;
1470      LDA MOUSEY  calc screen Y
1475      LSR A
1480      LSR A
1485      STA ROWCRS  new cursor row
1490 ;
1495      LDA #$87    fake cursor rt key
1500      STA CH      for click
1505 ;                and positioning
1510 BUTTO1 RTS
```

READER COMMENT

continued from page 8

phabet keycodes, it affects only alphabet keys. (Therefore, non-alphabet keys such as numbers and punctuation marks remain valid and unaffected.)

As the OS routine processes the alphabet key a second time, it combines the ShiftLock value at memory location 702 ($2BE) with the keyboard code, checks the value a second time, and aborts if the value is 192 ($C0) or above.

Therefore, to turn off the alphabet, place the value 192 ($C0) in memory location 702. In BASIC, simply POKE 702, 192.

To temporarily restore the alphabet, just use the SHIFT key. For instance, if you're entering in data statements and need to type the word *DATA*, simply type *DATA*, or its abbreviation, while pressing SHIFT. You press CAPSLOWR, SHIFT/CAPSLOWR, or CTRL/CAPSLOWR to exit from this "no-alphabet" lock. Also, SYSTEM RESET or POKE 702,64 restore the uppercase lock.

Alphabet Mode	Shiftlock Value	How Set
Lowercase	0	Press CAPSLOWR
Uppercase	64	Press SHIFT/ CAPSLOWR
Graphic Symbols	128	Press CTRL/ CAPSLOWR
None *(illegal)*	192	POKE 702,192

This method is most helpful when you're entering line after line of data statements, since most are often just numbers. Simply POKE 702,192, and you can be sure you won't enter a letter by mistake. This BASIC deomonstration program shows how POKE 702,192 catches almost as many numeric errors as TRAP:

```
10 TRAP 10
20 POSITION 2,8
30 ? CHR$(156);CHR$(157);:
REM CLEAR LINE
40 INPUT A
60 POKE 702,64:REM TURN ON
 UPPERCASE
70 TRAP 40000:REM TURN OFF
 TRAP
80 ? "NUMBER: ";A:REM PROG
RAM CONTINUES
```

Lines 10 to 80 can be replaced with the line shown here:

```
90 TRAP 90:POSITION 2,8:?
CHR$(156);CHR$(157);:POKE
702,192:INPUT A:POKE 702,6
4:TRAP 40000:? "YOU ENTERE
D ";A
```

By using the OS itself, my technique probably won't conflict with other subroutines. In fact, you may not want to bother with other machine-language numeric error-checking routines that may be more complicated. This principle also works in any machine language programs that use the Keyboard Handler Routine.

Mark Amanns
Cincinnati, OH

Wanted: 80-column board for *PaperClip*.

Even though I have only had **PaperClip** for a short time, I basically concur with Arthur Leyenberger's comments in issue 38 (January, 1986) on this new word processor from Batteries Included.

One of the major attractions for me was the future release of the 80-column board, which would make **PaperClip** nearly perfect for my purposes. It even prompted me to sell my comfortable 800 and buy an 800XL, since the 800 would not support the B.I.-80.

Imagine my chagrin when I called Batteries Included in Ontario, and was told that the B.I.-80 was cancelled. I was so disappointed, I even forgot to ask why.

I think Mr. Leyenberger failed to give us a complete review in omitting mention of the B.I.-80's demise. I hope it's not a matter of losing objectivity in the interest of advertising dollars. *(It's not—Ed.)* If anyone out there knows of a way to produce 80 columns using **PaperClip**, please share the wealth. Isn't the demand great enough for someone to finish the development for which Batteries Included must have spent a fortune?

Yours truly,
Fred Hart
Santa Cruz, CA

Boot Camp

continued from page 32

```
1970 ;ACCUMULATOR: HI ADDR OF STRING
1980 ;Y REGISTER:  LO ADDR OF STRING
1990 ;
2000 PRINT
2010     LDX #$00     ;USE EDITOR
2020     STA ICBAH,X  ;MOVE A TO HI
2030     TYA          ;PUT Y IN A REG.
2040     STA ICBAL,X  ;MOVE IT TO LO!
2050     LDA #$09     ;SET UP...
2060     STA ICCMD,X  ;PUT RECORD
2070     LDA #$FF     ;SET BUFFER...
2080     STA ICBLL,X  ;LENGTH...
2090     STA ICBLH,X  ;TO MAXIMUM
2100     JSR CIOV     ;PRINT IT!
2110     BMI FATAL    ;ERROR!
2120     RTS          ;OK, RETURN
2130 FATAL
2140     LDA #$34     ;CHANGE...
2150     STA COLOR4   ;BORDER COLOR
2160     BRK          ;AND EXIT
2170 PFILE1
2180     .BYTE "ENTER INPUT "
2190     .BYTE "(INCLUDE D:)",$9B
2200 PFILE2
2210     .BYTE "ENTER OUTPUT "
2215     .BYTE "FILENAME "
2220     .BYTE "(INCLUDE D:)",$9B
2230 BADIN
2240     .BYTE "CAN'T OPEN FILE! "
2250     .BYTE "-- TRY AGAIN",$9B
2260 BADOUT
2270     .BYTE "*** BAD INPUT "
2280     .BYTE "FILENAME ***",$9B
2290 CPYMSG
2300     .BYTE "COPYING FILE...",$9B
2310 RDERR
2320     .BYTE "*** ERROR READING "
2330     .BYTE "FILE! ***",$9B
2340 WRTERR
2350     .BYTE "*** ERROR WRITING "
2360     .BYTE "FILE! ***",$9B
2370 DONE
2380     .BYTE "*** FILE WRITE "
2390     .BYTE "COMPLETE! ***",$9B
2400 FNAME1 *= *+20
2410 FNAME2 *= *+20
2420     .END
```

THE END USER

THIS MONTH:

For the ST—TOS to go EPROM, digitizing to go, and... of calendars and kings (plus an ace)

by Arthur Leyenberger

By the time you read this, the Fall COMDEX (Computer/Dealer Exposition) will have come and gone, and you know what happened there (see issue 39). The show, held every November in Las Vegas and attended by ninety thousand or so dealers, distributors, buyers and vendors, was a great success for Atari.

The company took approximately forty software vendors under its wings by renting them space in the area of the main Atari booth. In doing this, Atari was able to show that it's here to stay—and there are dozens of software companies agreeing with that assessment. Atari said, in essence, "The 520ST computer is real; it has a promising future; and you need not take our word alone for it."

What was most interesting about this particular trade show was what Atari was *not* showing. At the previous COMDEX in Atlanta, and at the June Consumer Electronics Show, Atari was touting its CD-ROM player and technology. As reported at the time, Atari was to supply a CD player for "under $500," and Activenture was to provide the software enabling the 520ST to access data on the optical disc. **Grolier's Encyclopedia** was up and running on the CD player, and was very impressive.

At the Las Vegas COMDEX, the CD-ROM player was conspicuous by its absence. According to Jack Tramiel, it wasn't shown because Atari has not yet found a vendor to supply players at the price Atari wants. At first, you'd think that (as I said last month in this very column) Atari was crying wolf by announcing this product then not showing it. Well, if you think for a minute, you'll agree that Jack's attitude is to be commended.

The CD-ROM technology is very important to the computer industry. With it, we'll finally be approaching the amount of storage capacity that's necessary to get information right into the home. Encyclopedias, medical references, complete dictionaries, catalogs, and so forth previously required too much storage space to be practicably available to the home user.

Consumer grade audio CD players are now marketed at under $300. On the other hand, IBM and Apple have CD-ROM players available for their computers at close to $2000. There's no reason for a consumer to have to spend that much more for a CD-ROM player than for an audio player. Therefore, Atari's determination to wait until a player can be obtained to retail for the "correct" price is a wise move—one which will benefit us all.

TOS in ROMs on the ST.

There's a good chance that, by the time you read this, the ST computer operating system, TOS (including GEM) will be available in ROM (Read-Only Memory) at an Atari retailer near you. No, this is not blind faith in the word of Atari. This is based on the fact that, as I write these words TOS is snuggled happily beneath my fingers, in six chips deep inside the ST.

Thanks to the help of Gordon Monnier and Alex Leavens, I was able to obtain EPROMs (Electrically Programmable Read-Only Memory) chips containing GEM and TOS. EPROMs are like ROMs, except they can be "written over" again. So, in effect, I have a set of "prototype" chips.

What does all this electronic doubletalk mean to the average ST user? Speed and memory! No longer will you have to wait for TOS to load from disk; approximately 200,000 bytes of memory are now available for your programs and applications.

Here are some timing comparisons between loading TOS from disk and loading TOS from the EPROMs/ROMs:

Files	EPROMs	TOS on Disk
No files on disk	4 sec.	24 sec.
desktop.inf (only)	5 sec.	25 sec.
desk1.acc + desktop.inf	10 sec.	29 sec.
desk2.acc + desk1.acc + desktop.inf	14 sec.	32 sec.
No disk in drive A	42 sec.	No TOS

The timings were done with a wristwatch stopwatch and should be accurate to within a second or two. So, if you've saved the desktop to the disk and have two desk accessory files saved, it will only take 14 seconds to see those cute little icons, from the time you turn on the computer. You can boot up your ST computer about 20 seconds quicker with the operating system *not* on the disk. Here are some other interesting facts.

If you're fortunate enough to have both a monochrome and an RGB monitor for your Atari ST, you know that there are certain times when you want to switch from one to the other. You also know that, when you remove the monitor cable from the back of the ST, the computer locks up. Even if you only have one monitor, whenever you remove the monitor plug, the computer goes into a coma.

I'm happy to report that the version of TOS on ROM fixes this problem, by rebooting the computer (a "warm boot") whenever the monitor cable is removed. You still lose the screen, but it takes a maximum of 14 seconds to get it back.

Another change in this version of TOS is that, when you save the desktop using the "save desktop" option, the desktop.inf file is written to the disk in the currently active window. Previously, the desktop.inf file was always written to the disk in drive 1, regardless of what window was open. This change makes it a little easier to manage your disks, in that you can change your screen resolution preferences, view options and control panel settings, then save the desktop to different disks.

One other change seems to exist, although I haven't confirmed it with Atari—hard disk support is present in the new TOS. Occasionally, when booting my ST, I wind up with not only drive A and B icons, but also a "hard disk drive C" icon. With the coming of hard disks for the ST, from Atari and other sources, it makes sense to have this capability in the final version of TOS.

Unfortunately, there's one bug in TOS that hasn't been fixed in the new ROM version. When changing screen resolution modes while using an RGB monitor, the clock gets confused; date and time are lost.

It's annoying to have to continually reset the time and date if you have an RGB monitor. However, by the time you read this, there will probably be a real-time clock cartridge for the ST, to negate this problem.

I cannot disclose which company will market this product, but I do know that it's being tested as this is written. With the battery-operated cartridge, the time and date will always be correct, regardless of screen resolution mode changes or turning the ST off. When ready, the product will be reviewed here.

Of course, the biggest thrill of all is to see that extra 200K of memory become available. I typically use **M-Disk** by MichTron and have become used to having only about 90K of RAM memory available for my ramdisk. With TOS on the chips, I now have 290K of RAM available.

The extra memory freed up by having TOS on ROM lets you have larger files in **ST Writer** (or other application programs), and allows you to write longer BASIC and LOGO programs.

Telling It Like It Is Department.

I just finished talking with a software developer who had read the ST utilities review article in **ANALOG Computing**'s issue 39 (February). This person told me he was very surprised to see a "tell it like it is" article in a computer magazine.

He thought that all magazines would be more concerned with courting advertisers than with giving honest reviews and useful information to readers.

As you know, **ANALOG Computing** does "tell it like it is." You *expect* to read reviews that give you valid information to make purchasing decisions. You expect honest reviews, and that's what you get. The same is true for what's written in this column. I would rather alienate the manufacturer of an inferior product than lead you on or give you false information.

This software developer *is* now a believer. I just wanted you to know.

An alternative Operating System for the ST.

Although I cannot at this time divulge the name of the company who will market it, there will soon be another Operating System for use on the Atari ST. Called **OS/9**, this operating system is very Unix-like, but is supposedly faster and much smaller.

The people in the know about Operating Systems tell me that **OS/9** will take the ST community by storm. It offers multi-user features and relocatable code, and will permit a whole body of software to be easily ported over to the ST. Stay tuned for further announcements on this product.

Your name in lights.

Well, sort of—more like "your picture displayed in pixels." What I'm referring to is a service provided by a company called Bitmap. For a reasonable charge, they'll digitize any picture or photograph for the ST. They're currently doing only low-resolution **NEO-Chrome** files, but by the time you read this, they'll be able to provide you with low-, medium- or high-resolution **DEGAS** format files.

Bitmap requires that you send a disk, along with your material, to Bitmap, Box 52466, New Orleans, LA 70152 (504-891-4862 or BBS 504-394-6224). If you'd like to use their services, the cost is $10 for the first picture and $5 for each additional picture you'd like digitized. Their work is excellent, as you can see in the sample screen dump that follows. Of course, what you get is a black and white picture file that you can color as you like.

Bitmap also has what they call the **Electronic Coloring Book**. This is a disk of nine **NEO-Chrome** files that you can color, just as you would a coloring book. As juvenile as it may sound, it's really quite enjoyable. The pictures provided are primarily turn-of-the-century woodcut engravings, intricate in detail.

The 1050 DUPLICATOR IS HERE...

INCLUDES NEW ULTRA-SPEED SOFTWARE. READ & WRITE 3X FASTER IN SINGLE AND DOUBLE DENSITY!

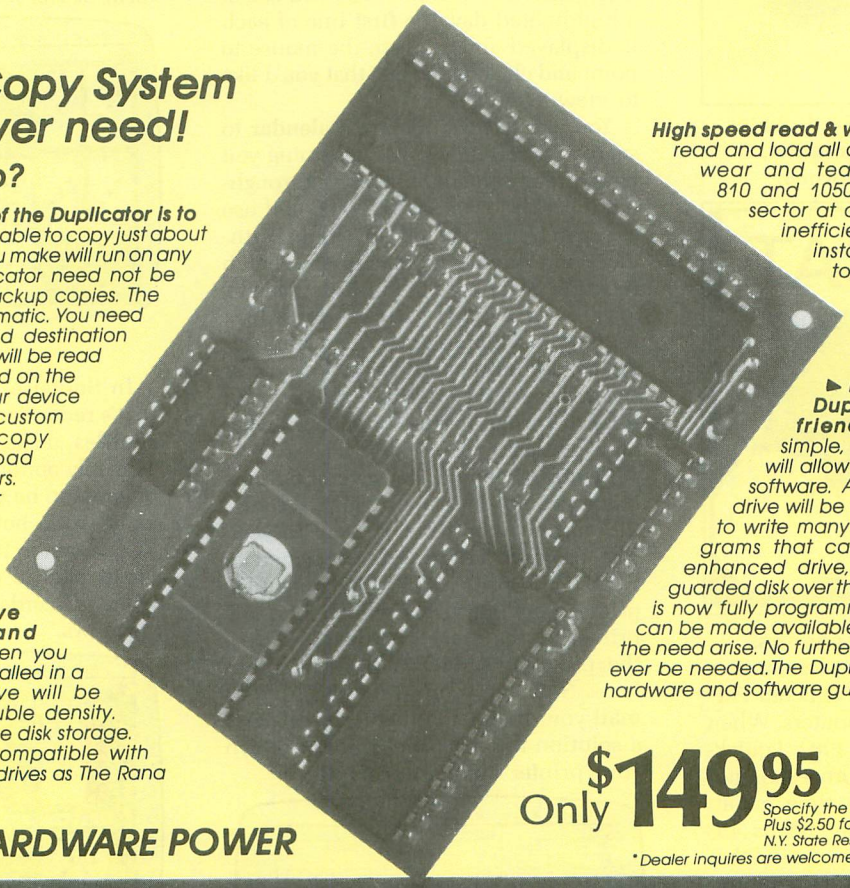
THE 1050 & 810 DUPLICATOR: The most powerful diskdrive copy system ever developed for the ATARI.

The only Copy System You will ever need!

What will it do?

►**The main purpose of the Duplicator is to copy disks!** You will be able to copy just about any disk! The copies you make will run on any Atari drive. The Duplicator need not be present to run your backup copies. The Duplicator is fully automatic. You need only insert source and destination disks. Custom formats will be read and in turn reproduced on the backup copy disk. Our device will reproduce any custom format or heavily copy guarded scheme, bad sectors, double sectors. 19 through 24 sector format will present no problem to the Duplicator.

►**You will still have single density, and double density.** When you have a Duplicator installed in a 1050 drive that drive will be turned into true double density. You will have twice the disk storage. Your drive will be compatible with other double density drives as The Rana Indus. Percom, etc.



HARDWARE POWER

High speed read & write. Your disk drive will read and load all of your software, saving wear and tear on your drive. The 810 and 1050 drives now read one sector at a time. This is slow and inefficient. With the duplicator installed you will be able to read eighteen sectors in the time it takes standard, unenhanced drives to read one.

►**Included with every Duplicator will be user friendly disk software.** A simple, menu driven program will allow you to copy all of your software. A Duplicator enhanced drive will be a SMART drive. We plan to write many new and exciting programs that can only be run on an enhanced drive, eg. sending a copy-guarded disk over the phone. Since the drive is now fully programmable, future upgrades can be made available to you on disks, should the need arise. No further hardware changes will ever be needed. The Duplicator comes with a full hardware and software guarantee.

Only $149.95

Specify the 810 or 1050 when ordering.
Plus $2.50 for shipping and handling.
N.Y. State Residents add 7½% Sales Tax.
*Dealer inquires are welcome call for quantity price quote.

The Duplicator for The New "ST" will be available March 1st.

EASY 5 MINUTE INSTALLATION

NO HARM TO YOUR DRIVE OR INCOMPATIBILITY PROBLEMS CAN EVER ARISE AS A RESULT OF THE INSTALLATION OF OUR DUPLICATOR.

IMPORTANT:

Only a hardware device like the DUPLICATOR can backup heavily copy-guarded disks. Don't be fooled by software programs that claim to do this.

Fully Compatible with the XL & New XE Series.

"Our competition promises tomorrow...
We Deliver Today!

DUPLICATING TECHNOLOGIES inc.
Formerly Gardner Computing

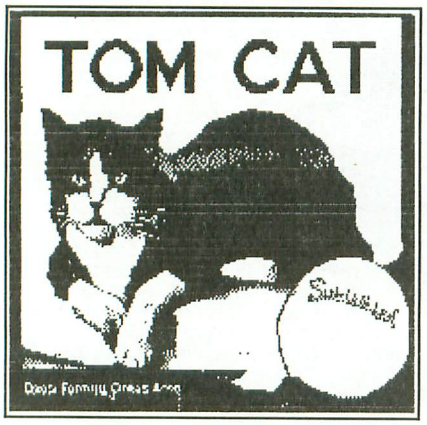
VISA MasterCard

99 Jericho Tpke., Suite 302A Jericho, N.Y. 11753 Order Business Hrs. **(516) 333-5808, 5805, 5807** Order Eve's. and Weekends **(516) 333-5950**

TERMS: We accept American Express, Visa, MasterCard and C.O.D. orders. Foreign orders must be in U.S. dollars. All personal checks allow 14 days to clear. We ship within 24 hours.

CIRCLE #113 ON READER SERVICE CARD

Screen dumps of digitized photos by Bitmap—low-resolution, NEO-Chrome file format.



A calendar for all seasons.

For a long time, I've been a supporter of LJK Enterprises, makers of **Letter Perfect** for the Atari 8-bit computers. When I first got my Atari in early 1982, I quickly found that this word processor not only suited my needs at the time, but allowed me to grow as my needs increased. Now one of the founders of that company has formed a new corporation: Soft Logik Corp.

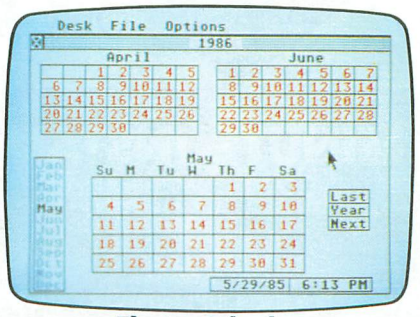
Soft Logik has two new programs for the Atari 520 ST: **Electro Calendar** and **Electro Solitaire & 21**. **Electro Calendar** is a computerized calendar system that spans the centuries from 1776 (did you know that July 4, 1776 was a Thursday?) to 3001. Not only does the program allow you to display and print any month during this period, it also lets you mark important dates with brief reminders.

Once the month you've selected is on-screen, you simply click the mouse button on the particular day that you want to remind yourself of something. Then type in your message at the top of the screen and save it, either for the currently displayed year or for all years. Once the message has been saved, the day of the month is highlighted right on the calendar.

When entering a message for a specific day of the month, **Electro Calendar** provides several editing commands for you to use, such as insertions, deletions, tab stops and cursor movement. Also, more than one memo can be entered for any particular day. If there's already more than one memo when you select a highlighted day, the first line of each is displayed and you use the mouse to point and click on the one that you'd like to view or edit.

You can use the **Electro Calendar** to search for certain events. Assuming you have already entered birthdays throughout the year for relatives, you could use the search function to find "Jean's Birthday," for example. Of course, the search function isn't limited to birthdays alone. Regardless of what's entered—birthdays, reminders, business meetings, anniversaries, appointments—you can enter whatever search criteria you want and have the program find a match on what you entered.

Electro Calendar has several options that let you print memos you've written, the current month, or the entire year. The program is currently designed to work with Epson and Epson-compatible printers, but Soft Logik will support any printer for the Atari ST. If you purchase the program and have printing problems, Soft Logik promises that, if you mail your manual to them, they'll send a solution to your problem (and return your printer manual) promptly.

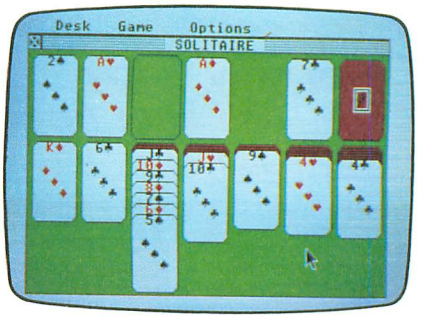


Electro Calendar.

Electro Calendar sells for $40 and is available now from Soft Logik, 4129 Old Baumgartner, St. Louis, MO 63129 (314-894-8608). Currently, **Electro Calendar** is an application program. Soft Logik plans to release the program as a desktop accessory. When this comes out, all previously registered owners will be notified, and there will be no charge for the exchange of programs.

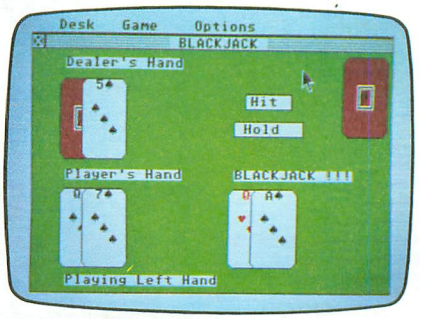
How about a game of cards?

Soft Logik also has card games for sale. Called **Electro Solitaire & 21**, this disk costs only $20 and includes both the Blackjack and Solitaire games. I won't spend time telling you how to play either, since the computer gameplay is exactly the same as the card version. Both games provide on-line help, in the form of rules.



Solitaire.

In Solitaire, you play alone. I guess that's redundant. Anyway, the program shuffles, deals and won't let you cheat. The only option is to change the amount of cards to be turned over at once. The game works both on a monochrome and RGB monitor. It is obviously a little easier to play in color, and I just love it when the cards are dealt using the GEM windows.



Blackjack.

Blackjack can also be played on either of the Atari monitors and provides a chance for you to play against the dealer. Splitting and doubling down are available as well as the ability to change the amount of your bet. The only criticism I have of this game is that when you and the dealer both show the same card count, the dealer wins, rather than it being a push (tie).

Since you get both games for $20, I recommend that you buy **Electro Solitaire & 21** if you like these card games. Also, the disk is not copy-protected, so please don't abuse Soft Logik's sane poli-

cy of not having copy protection. Don't give copies or accept copies of this program.

Almost the end.

There's so much more to talk about, though I've run out of room. I have 8-bit computer products, as well as more ST products to discuss.

Alas, next month will come before I know it. . . and I do want to mention one product very briefly before I sign off this month. This product really should have appeared in the **Stocking Stuffer** article in issue 38 (December, 1985), but its information didn't arrive in time to meet the press deadline.

If you have an Atari 800 computer and you want 80 columns on the screen at once, until recently you had only one choice: get a Bit-3 80-column board. This board is not only expensive, but requires that you have a 32K ROM board in your 800, as well. An 800XL or 130XE computer owner was out of luck completely.

This has changed with the release of the **Ace 80** cartridge from Amiable Computer Enhancements. The cartridge will work with any of the Atari 8-bit computers, to give you 80 columns when using either BASIC or the **Letter Perfect** word processor.

The cartridge sells for only $50 and comes in two models: one for the 800 computer and one for the XL/XE machines.

The **Ace-80** is not quite as good as the Bit-3 board, because Bit-3 is a hardware product, while **Ace-80** is software. Still, the **Ace-80** works on a TV or monitor and gives acceptable results, especially on a monitor.

If you use **Letter Perfect** or do a lot of BASIC programming, you should check out this product, from TNT Computing, P.O. Box 443, Holt, MI 48842 (517-394-2412). The **Ace-80** cartridge needs 48K of memory.

See you next month.

Koala Slideshow

by Charles Johnson

If you have either an Atari 130XE computer or an Atari 800 with an AXLON 128K board installed, this program will use your extra RAM to hold and display quite a few compressed Koala format picture files. And since the **KoalaPad** and the **Atari Touch Tablet** use the same picture compression technique, **Slideshow** will also display pictures created with the **Atari Artist** cartridge. (In fact, whenever I use the term "Koala file" in the following article, I will also be referring to **Atari Artist** files.) **Slideshow** automatically determines your system configuration when it starts and will not continue unless it's running on a 130XE or AXLON-equipped 800.

You can choose to show all the Koala pictures on a disk, or select only certain pictures to display. Once the pictures are loaded into the extra RAM, you can flip through them forward and backward by pressing the "+" and "–" keys or by using a joystick; or you can set **Koala Slideshow** to automatically flip through the pictures for you. The number of pictures your memory can hold depends on your equipment and the size of the pictures.

If you have an Atari 130XE computer, the amount of RAM available for picture storage is 80K. This is almost the storage capacity of a single-density disk. If you have an AXLON 128K board, the full 128K is available to store pictures, which gives you more RAM than a single-density disk. Since **Slideshow** works with any DOS, it can be used in double density as well. The 130XE version allows a maximum of 32 pictures; the AXLON version allows 48.

Using the program is very simple. **Slideshow** is in machine language and can be loaded with the *L* (Binary Load) command of Atari DOS 2.0 and 2.5, or you can name it AUTORUN.SYS to have it boot automatically from that disk. If you use the AUTORUN.SYS technique, there's no need to press OPTION when you turn on your 130XE.

When **Slideshow** begins, it will ask you for a drive number—1 or 2. Put a disk containing Koala format picture files in the drive you want to use and type the number of that drive. The Koala files should all have the extension *.PIC*. You'll be asked if you want to display all the pictures on the disk. If you type *Y*, **Slideshow** will load as many Koala pictures from the disk as it can.

If you type *N*, **Slideshow** will read the disk directory and display the names of all files with a *.PIC* extension. As each name is displayed, the program will wait for you to type *Y* or *N*. Type *Y* to select that picture for loading, or *N* to skip it. If you decide you want to start over with another disk at this point, hit ESC, and the program will restart.

Important: while **Slideshow** is reading the directory from a disk, *do not remove the disk* and insert another. This could lead to a scrambled directory and a crashed disk!

When you hit either the maximum number of pictures allowed or the end of the directory, **Slideshow** will begin loading the pictures. Each filename is printed as it loads. If your memory runs out before all the pictures can be loaded, you'll see the message *MEMORY FULL!* When this happens, or all your pictures have been loaded, the show starts automatically with the first picture.

To move forward and backward through the pic-

Program

tures, press the + and – keys. If you have a joystick plugged into port 1, you can do the same thing by pushing the stick forward or backward. The next picture will be displayed very quickly (since it's in RAM). **Slideshow** also prints the name of each picture and a reminder of the key commands on a separate line above the screen. If you want to get rid of this line of text at the top of the screen, press either the OPTION key or the joystick button. Another press restores the text line. To restart the program and show pictures from another disk, just press ESC.

The START key toggles the automatic show option. This will step forward automatically through the pictures, pausing for about five seconds on each one. While in automatic mode, the only keys that will work are START, OPTION and ESC.

Slideshow also has some other special capabilities. It can display pictures in graphics mode 8 and 9, in addition to the normal Koala mode of ANTIC +7. To do this you must alter the eighth (starting from zero) byte in the picture file. Every Koala format picture file has a header at the beginning that contains information about the picture; colors, compression type, etc. The eighth byte in is the graphics mode. This byte will usually be 14 ($0E). If you change it to 15 ($0F), **Slideshow** will show your picture in graphics 8. To display a graphics 9 picture, change the mode byte in the file header to 79 ($4F).

Here's a way to convert any standard 62-sector picture file to a Koala format file. First, use DOS to rename your 62-sector file to *PICTURE*. Then load either the **Micro-Illustrator** program or **Atari Artist**. Go to the drawing screen, put the disk containing *PICTURE* in drive 1, and press the CLEAR key (not SHIFT/CLEAR). This will load your picture file into MI or AA.

It may not look right, since it's not being displayed in its proper graphics mode, but don't worry about that. Just use the program's SAVE command to save the picture as a Koala format file. Then you can use the following short BASIC program to change the mode byte, so that **Koala Slideshow** will display it in the correct graphics mode.

```
10 DIM F$(15)
20 GRAPHICS 0:? :? "Name of Koala file
 to modify?"
30 ? ">";:INPUT #16;F$
40 ? :? "Which mode (8 or 9) ";
50 ? ">";:INPUT #16;MODE
60 IF MODE<>8 AND MODE<>9 THEN 50
70 IF MODE=8 THEN MODE=15
80 IF MODE=9 THEN MODE=79
90 OPEN #1,12,0,F$
100 FOR I=0 TO 7:GET #1,A:NEXT I
110 PUT #1,MODE
120 CLOSE #1
130 END
```

Note: I think I've discovered a bug in both the **Micro-Illustrator** and **Atari Artist** programs. The bug has to do with the very conversion process I've described above.

I have a disk full of digitized photographs in graphics mode 9, which I wanted to convert to Koala format to display with **Koala Slideshow**. The pictures all loaded into MI or AA by naming them PICTURE and pressing CLEAR. But, with a couple of them, when I tried to SAVE the pictures as Koala files, something about the picture totally derailed the compression process. The program wrote a disk file that was over 150 sectors long! When I then tried to load this file back in, only garbage would appear on-screen.

The bug seems to bite when you're trying to save a file that has a very large amount of detail. The digitized photos I was trying to convert definitely fit that description. So a final *caveat:* you may not be able to convert some pictures with the above method—but, honest, it's not my fault!

Charles Johnson is a professional musician, currently playing with the band Chicago. A self-taught programmer, he likes machine language best.

Listing 1.
BASIC listing.

```
10 REM *** SLIDESHOW ***
20 TRAP 20:? "MAKE CASSETTE (0), OR DI
SK (1)";:INPUT DSK:IF DSK>1 THEN 20
30 TRAP 40000:DATA 0,1,2,3,4,5,6,7,8,9
,0,0,0,0,0,0,0,0,10,11,12,13,14,15
40 DIM DAT$(91),HEX(22):FOR X=0 TO 22:
READ N:HEX(X)=N:NEXT X:LINE=990:RESTOR
E 1000:TRAP 120:? "CHECKING DATA"
50 LINE=LINE+10:? "LINE:";LINE:READ DA
T$:IF LEN(DAT$)<>90 THEN 220
60 DATLIN=PEEK(183)+PEEK(184)*256:IF D
ATLIN<>LINE THEN ? "LINE ";LINE;" MISS
ING!":END
70 FOR X=1 TO 89 STEP 2:D1=ASC(DAT$(X,
X))-48:D2=ASC(DAT$(X+1,X+1))-48:BYTE=H
EX(D1)*16+HEX(D2)
80 IF PASS=2 THEN PUT #1,BYTE:NEXT X:R
EAD CHKSUM:GOTO 50
90 TOTAL=TOTAL+BYTE:IF TOTAL>999 THEN
TOTAL=TOTAL-1000
100 NEXT X:READ CHKSUM:IF TOTAL=CHKSUM
 THEN 50
110 GOTO 220
120 IF PEEK(195)<>6 THEN 220
130 IF PASS=0 THEN 170
140 IF  NOT DSK THEN 160
150 PUT #1,224:PUT #1,2:PUT #1,225:PUT
 #1,2:PUT #1,0:PUT #1,38:CLOSE #1:END
160 FOR X=1 TO 31:PUT #1,0:NEXT X:CLOS
E #1:END
170 IF  NOT DSK THEN 200
180 ? "INSERT DISK WITH DOS, PRESS RET
URN";:DIM IN$(1):INPUT IN$:OPEN #1,8,0
,"D:AUTORUN.SYS"
190 PUT #1,255:PUT #1,255:PUT #1,0:PUT
 #1,38:PUT #1,184:PUT #1,48:GOTO 210
200 ? "READY CASSETTE AND PRESS RETURN
";:OPEN #1,8,128,"C:":RESTORE 230:FOR
X=1 TO 40:READ N:PUT #1,N:NEXT X
210 ? :? "WRITING FILE":PASS=2:LINE=99
0:RESTORE 1000:TRAP 120:GOTO 50
220 ? "BAD DATA: LINE ";LINE:END
230 DATA 0,22,216,37,255,37,169,0,141,
47,2,169,60,141,2,211,169,0,141,231,2,
133,14,169,56,141,232,2
240 DATA 133,15,169,0,133,10,169,38,13
3,11,24,96
1000 DATA D8A20020362BA9038D4203A92F8D
4403A92F8D4503A90C8D4A032056E4A9E38D01
D38D0040A9FF8D01D3AD0040,459
1010 DATA C9E3F01DA90185AEA9D385AFA920
85B0A90485B1A204BDAA2F9D742FCA10F7303D
A9078DC0CF8D0040A9008DC0,203
1020 DATA CFAD0040C907F01DA9C085AEA9CF
85AFA93085B0A90785B1A204BDAF2F9D742FCA
10F7300CA925A235A02F2023,605
1030 DATA 2B4C8826A9008DC502A90A8DC602
A9248DC80218A55869508D442EA55969008D45
2EA93A8D3002A92E8D3102A2,794
1040 DATA 2020362BA9039D4203A9319D4403
A92F9D4503A9049D4A032056E4A9008584A940
8585A9038554A913A2B4A02F,56
1050 DATA 20232BA220A9079D4203A9009D48
039D49032056E4C931F004C932D0F58DC72F8D
28308D3130A900859DA91CA2,612
1060 DATA C7A02F20232BA2202056E48DE32F
C959D006E69DA902D002A928A2E3A02F20232B
A900858D859F85A285888589,839
1070 DATA 85868587203E2BA230A9039D4203
A9279D4403A9309D4503A9069D4A032056E4A2
30A9059D4203A9419D4403A9,997
1080 DATA 309D4503A9119D4803A9009D4903
2056E4AD5030C953F06FA588C5B0F069A59DD0
4BA908A243A03020232BA905,898
1090 DATA A20BA03020232BA220A9079D4203
A9009D48039D49032056E4C959F010C94EF00C
C91BD0E2A23020362B4C0026,157
1100 DATA 8D1030A902A210A03020232BAD10
30C94EF086A007B943309180881OF818A58069
0885809002E681E6884C5B27,827
1110 DATA A22020362BA23020362BA588D003
4C0026203E2BA9258582A9328583A901A211A0
3020232BA000848CB180C920,841
1120 DATA F008993330C8C008D0F2A200BD2B
30993330C8E8E005D0F498186903858A18A580
690885809002E681A908A212,106
1130 DATA A03020232BA58AA230A03020232B
A210A9039D4203A9309D4403A9309D4503A904
9D4A032056E4A907858BA907,99
1140 DATA 9D4203A9009D48039D49032056E4
C68B10F9A4899915332056E4A4899945333A903
858B2056E4C68B10F9A90485,51
1150 DATA 8B2056E4A48B9182C68B10F518A5
8269058582902E683A908858B2056E4C68B10
F9A9079D420318A90065869D,295
1160 DATA 4403A94065879D450338A584E586
9D4803A585E5879D49032056E4A58CD011A489
A586997533A58799A533A58D,679
1170 DATA 99D53318A5867D48038586A5877D
49038587BD4303C988F03FA58DC5B1F0201869
01858DA4B0C030F00838E901,28
1180 DATA 0A0A1869E3A00091AE848684874C1428
A9808583A9008582859E85A98D2F028D00D48D
C8028D1AD0A240A89182C8D0,769
1210 DATA FBE683CAD0F6A00FB98130996630
8810F7A95E859885A0A92E859985A1A005B953
3091988810F8A907A22BA0AF,806
1220 DATA 205CE4A9C08D0ED4A900858B858D
A4B0C030F002A9FFA00091AEA5A2490185A218
A48BB97533690085BEB9A533,483
1230 DATA 6940858FB9D533858DA6B0E030F0
0A38E90130050A0A1869E3A00091AEA48BB915
338590D006200F2E18900320,60
1240 DATA D22CE69FA900854DA9FF8DFC02AD
1FD085ABC906D025AD1FD0C906F0F920472BA5
A9490185A9AABDA13085ACBD,95
1250 DATA A33085ADA00FB1AC9966308810F8
A5ABC903D0034C0F2BA5A9F027A5141869FFAA
AD1FD085ABC906F0BFC903D0,842
1260 DATA 034C0F2BADFC02C91CD0034CDA2A
E414D0E24CEA2AAD7802C90FF01648A514C514
F0FC68C90ED0034CEA2AC90D,51
1270 DATA D0034C032BAD8402D0034C0F2BAD
FC02C9FFD0034C452AC91CD008A9FF8DFC024C
0026C921F004C906D00EE68B,971
1280 DATA A58BC589F0034CF8294CE629C915
F007C90EF0034C402AC68B1005A689CA868B4C
F829AD8402F0FBAD1FD0C903,588
1290 DATA F0F420472BE69E4C402A8D48038E
44038C4503A90B8D4203A2004C56E4A90C9D42
034C56E4A9A58580A9308581,242
```

```
1300 DATA 60A97885AAA9A88D01D2A5AA8D00
D238E91485AA3008A514C514F0FCD0EC60488A
489848ADC80229F0A20EA000,905
1310 DATA 8D0AD48D18D08E17D08C1BD0A98F
8D0002A92B8D010268A868AA6840488A489848
ADC602AEC502AC6F028D0AD4,814
1320 DATA 8D18D08E17D08C1BD068A868AA68
40A9658D0002A92B8D0102A59FD0034CB22CC6
9FA6A2BD2B2F8D662EBD2D2F,785
1330 DATA 8DC62EA68BBD4533290F859ABD45
3329C08D6F028D1BD0A90085818A0A26810A26
810A2681186 9A58580A58169,353
1340 DATA 308581A007B18038E920995B3088
10F518A95E8D30028D02D4690685 98A92E8D31
028D03D469008599A9228D2F,919
1350 DATA 028D00D4A000A59AC90FD01848A9
008DC6028D18D08DC8028D1AD0A90E8DC5028D
17D06809409198A060919818,820
1360 DATA A5986903859890 02E699A05CA59A
91988810FB18A5986960859890 02E699A060A5
9A91988810FBC90FD0034C62,562
1370 DATA E4A9008583A58B0A26830A268318
658B9002E6831869258582A5836932858 3A004
A200B1829DC4029D16D0E888,543
1380 DATA 10F4301DA59EF019C69EA005A930
D1A0F00791A08810FB3008B9533091A08810F8
4C62E4A4A2A950859185A385,291
1390 DATA A585A7B92B2F859285A485A61869
1E85A8A000849484978493B18E0820DA2D2818
2A26934A8596D00EB18E8597,463
1400 DATA 20DA2DB18E859620DA2DA593F022
B18E20DA2D9191A690E002D00620C22D4C2C2D
205E2DC696D0E6A597F0BCC6,230
1410 DATA 9710DEB18E859520DA2DA5959191
A690E002D00620C22D4C522D205E2DC696D0E9
A597F096C69710E118A59169,38
1420 DATA 5085919002E692A591C5A7D02CA5
92C5A8D026A594D023E69418A5A56928859 1A5
A669008592 18A5A7692885A7,993
1430 DATA A5A8690085A8E6A5D002E6A660C6
94A5A5859185A7A5A6859218691E85A8A5A318
6928AAA5A46900C592D0E0E4,257
1440 DATA 91D0DC686860E691D002E692A5A4
18691EAAA5A3E492D0C7C591D0C3F0E548A58E
C9FFD026A58FC97FD020E68D,404
1450 DATA A58DA4B0C030F00838E9010A0A18
69E3A00091AEA900858EA940858FD006E68ED0
02E68F6860A4A2A9508591B9,43
1460 DATA 2B2F8592A900859BA91E859CA000
B18E9191E691D002E69220DA2DC69BD0EFC69C
10EB60707050425A2F000210,408
1470 DATA 42000002020202020202020202
0202020202020202020202413A2E70D0C2593010
4F50810F0F0F0F0F0F0F0F0F,775
1480 DATA 0F0F0F0F0F0F0F0F0F0F0F0F0F0F
0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F
0F0F0F0F0F0F0F0F0F0F0F0F,450
1490 DATA 0F0F0F0F0F0F0F0F0F0F0F0F0F0F
0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F
0F0F0F0F0F0F4F00900F0F0F,303
1500 DATA 0F0F0F0F0F0F0F0F0F0F0F0F0F0F
0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F
0F0F0F0F0F0F0F0F0F0F0F0F,978
1510 DATA 0F0F0F0F0F0F0F0F0F0F0F0F0F0F
0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F
0F0F0F0F0F0F0F0F0F0F0F0F,653
1520 DATA 0F0F0F0F415E2E81A190B0453A4B
3A533AA0CECFA0C5D8D4D2C1A0D2C1CDA0C6CF
D5CEC4A1A0D0D2CFC7D2C1CD,391
1530 DATA A0C1C2CFD2D4AEAEAE8080808080
808080ABAFA1ACA180B3ACA9A4A5B3A8AFB780
809C80808080809E80808080,357
1540 DATA 8080808080808080808080808080
E2F980A3E8E1F2ECE5F380AAEFE8EEF3EFEE80
8080808080808080808091,730
1550 DATA 9390B8A5A1B8ACAFAE5768696368
20647269766520 28B12FB2293F20209B9B5368
6F7720616C6C207069637475,600
```

```
1560 DATA 7265732028D92FCE293F20209B9B
D965732FCE6F3D53656C6563742070696374 75
7265732020C5D3C33D526573,353
1570 DATA 746172749B9B202D2D3E20209B4C
6F6164696E6720CDC5CDCFD2D9A0C6D5CCCCA1
9B44203A2A2E5049439B4420,500
1580 DATA 3A20202020202020202020202020
2020202020202020202020202020202020202020
70D0C259301000000000000,249
1590 DATA 00000000000000B1D266F727761
7264008D1D2261636B00A5B3A31D2578697400
008B1D266F7277617264008D,533
1600 DATA 1D2261636B000000D6A1B5B4AFAD
A1B4A9A3C20008191303000000000000000000
00000000000000000000000000,220
```

●

CHECKSUM DATA.

(see page 10)

```
10 DATA 128,351,496,811,423,729,200,60
3,555,573,694,613,29,205,232,6642
160 DATA 739,198,962,659,491,30,155,12
6,187,852,111,107,815,734,698,6864
1060 DATA 877,571,741,679,526,435,615,
735,734,680,699,881,780,744,728,10425
1210 DATA 768,954,796,542,134,354,17,1
58,805,933,968,79,780,788,770,8846
1360 DATA 774,692,662,754,972,25,737,8
97,125,81,850,108,290,200,318,7485
1510 DATA 297,353,14,759,694,654,943,6
03,362,34,4713
```

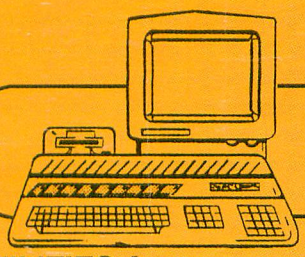
●

INDEX TO ADVERTISERS

READER SERVICE #	ADVERTISER	PAGE #
108	Allen Macroware	24
129	Alpha Systems	74
116	American TV	40
—	ANALOG Publishing	OBC
143	Applied Computers, Inc.	107
105	Applied Technologies	15
103	Astra Systems	5
—	Athena Software	58
111	BC Computers	18, 101
122	Beckemeyer Development Tools	58
131	CAL COM	77
145	Centurian Enterprises	112
120	Coast to Coast	56
106	Commnet Systems	20
121	Compucat	58
104	Computability	7
132	Computer Creations	79
146	Computer Eyes	112
147	Computer Games Plus	68
110	Computer Mail Order	26
135	Computer Palace	84
—	COVOX	17
113	Duplicating Technologies	35, 105
130	Electronic One	76
112	Games Computers Play	32

READER SERVICE #	ADVERTISER	PAGE #
117	Happy Computers	42
101	Hippopotomas Software	IFC
124	Integrated Computer Equipment	59
—	Jesse Jones	56
127	Kyan Software	70
138	Lionheart	95
119	Lyco Computers	53
136	Megamax	87
148	MichTron	IBC
128	Microtyme	73
133	New Horizons Software	83
102	Progressive Computer Applications	4
125	Protecto	60, 61, 62, 63
118	San Jose Computers	43
—	Senecom	107
123	Serious Software	59
126	Software Discounters	66
144	Southern Software	107
107	Sourceflow	22
137	T.D.I.	92
139	TNT Computing	95
134	Wedgwood Rental	83
114	White House Computers	36
115	Xentech	40
122	Xlent Software	96

This index is an additional service. While every effort is made to provide a complete and accurate listing, the publisher cannot be responsible for inadvertent errors.



for the ATARI ST

UTILITIES from our computer toolbox

MICHTRON UTILITIES *Utility by Timothy Purves*

Change file contents, attributes, file and volume names, or *any individual bytes on disk!* Just type the new data or click on selector buttons. To restore deleted files or lost data, simply inspect the data and click on a button to save or reject it. You can also format individual disk tracks, and copy individual sectors to repair damaged disks! A new 10 sector format utility lets you add 80K of storage capacity to your disks!

For the Atari ST $59.95

M-DISK *RAM-Disk by Timothy Purves*

M-Disk gives you the power of a super-fast, incredibly tough disk drive with-out the extra cost! *M-Disk* isn't a hardware device, but software utility. It makes a portion of your computer's memory resemble the data on a disk drive and makes the operating system recognize it as a hardware device, giving you the equivalent of an extra disk drive!

For the Atari ST $39.95

MI-TERM *Communications Utility by J. Weaver Jr.*

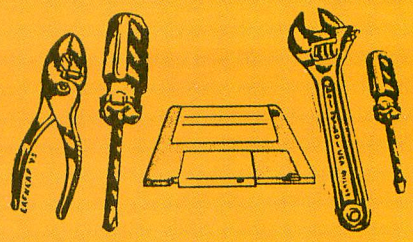
Far beyond the simple functions of other terminal software, *Mi-Term* provides DFT, XMODEM, and ASCII file transfers, 300/1200/9600 baud support, printer echo, automatic capture buffer, connect-time clock, automated logons and commands, and much more. With simple commands and the power of GEM, *Mi-Term* is as easy as "point and click".

For the Atari ST with modem $49.95

SOFT SPOOL *Printer Utility by Timothy Purves*

When printing files, your computer is dead weight: waiting for the printer to finish the job. With *Soft Spool*, your computer can think and print at the same time! Instead of going to the printer, printed data waits in *Soft Spool's* RAM buffer. You can then use your computer normally as the spooler feeds its data to your printer between tasks.

For the Atari ST $39.95



CALENDAR *by J. Weaver Jr.*

From almost anywhere in GEM, *Calendar* is ready to display or print calendars of any year and month and to store reminder messages for any date, any time: from 1980 to 2099. Set "alarms" on any or all messages. Alarms and hourly chimes appear even when *Calendar* isn't on the screen. Using *Calendar* is as easy at "point and click".

For the Atari ST $29.95

MI-DUPE *Copy Utility by Timothy Purves*

Copying your data files can be slow and tedious. Some programs are even copy-protected to stop you from making any duplicate at all. *Mi-Dupe* makes files duplications faster, easier, and helps you obtain archive copies of "protected" software.

For the Atari ST $29.95

BBS *Bulletin Board System by Timothy Purves*

MichTron presents a high quality Bulletin Board System for the ST. This many featured BBS has 16 special interest areas, supports XMODEM, DFT, and ASCII up- and downloading, bulletins, message base and much more. Call MichTron's BBS at (313)-332-5452 for a trial run.

For the Atari ST $49.95

GAMES from beyond imagination

TIME BANDIT *Arcade Game by Bill Dunlevy & Harry Lafnear*

Battle Evil Guardians as you collect the Treasures of Time. Two can play simultaneously, each with his own view! Each of 20 unique maps has over 15 levels and is a complete game in itself! Explore medieval dungeons, western frontiers, and future worlds all in one game. Full color graphics, unique creatures and 3D-style terrain for each Time, beautiful scrolling landscapes, detailed animation, thrilling sound, hundreds of screens: *the conquest of Time and Space awaits you!*

For the Atari ST with color montitor $39.95

MUDPIES *Arcade Game by Phil MacKenzie & Jeffrey Sorenson*

When the circus came to town, Arnold wanted in on the fun. He threw mudpies at the clowns but they didn't think it was so funny. They threw things back and tried to catch him. Soon the place was a *REAL* circus!

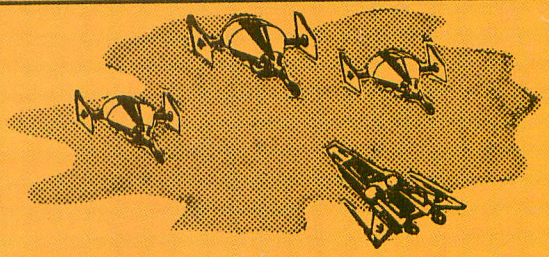
Keep Arnold from harm by using mudpies to ward off angry clowns. Challenge rounds, special prizes, mud-slinging rounds, amazing music and sound effects make this a great addition to your game collection!

For the Atari ST with color monitor $39.95

GOLD RUNNER *Arcade Game by Dave Dies*

As Commander of the Load-Runners, your must infiltrate underground mines in search of gold and adventure. Use wit and skill to escape with the loot. Dozens of screens with narrow paths, steep ladders, dangling ropes, and hidden traps will challenge your skill and test your logic!

For the Atari ST with color monitor $39.95



LANDS OF HAVOC *Arcade Game by Microdeal*

The dark power over the land is almost complete. The only hope lies in finding the hidden magic of the good sorcerer, High Vanish. As Sador, the reptile warrior with a human heart, you must free Haven from the Dark Lords. You have only your powerful body, your intelligence, and the secrets High Vanish left behind. 2,000 screens hold many surprises.

For the Atari ST with joystick $19.95

FLIP SIDE *Strategy Game by Ken Olson & Phil Hollyer*

You'll flip over this Reversi-type game! Animated graphics bring the board to life as captured pieces flip themselves. Play against another person, or against the computer in one of six skill-levels: from novice to expert. The computer can show all available moves, or suggest one for you. You can switch sides with the computer, and even edit the board!

For the Atari ST $39.95

MichTron

576 S. Telegraph, Pontiac, MI 48053
Orders and Information (313) 334-5700
Prices Do Not Include Shipping and Handling

Apple Mac 512™

IBM PCAT™

Commodore Amiga™

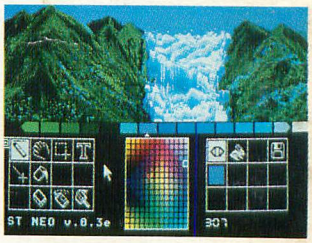
THERE'S ONLY ONE WORD FOR THESE PRICES: RIP-OFF.

Introducing the Atari 520ST personal computer system. $799.95 complete.*

Go ahead. Compare those other machines with the new Atari 520ST.™ They cost hundreds of dollars more, but you don't get much in return. That's what we call a rip-off.

For $799.95,* the 520ST comes complete with high-resolution monochrome monitor, 2-button mouse, 3.5" disk drive, TOS™ Operating System, including GEM™ Desktop, plus Logo™ and Atari BASIC programming languages. $200 more gives you an RGB color monitor with 512 glowing colors.

Choose innovative business, entertainment, education, systems management, and integrated package software. Expand your 520ST with industry standard parallel printers, modems, MIDI controlled synthesizers and keyboards, 1 megabyte floppies, 10 MB and larger hard disks, and more. All available now. At remarkably low prices.



So, go ahead. Compare the ST system to those other guys. Only Atari gives you so much. For so little.

For the dealer nearest you, write Atari Corp., Customer Services, 1196 Borregas Ave., Sunnyvale, CA 94086.

	ATARI™ 520ST	IBM™ PCAT™	APPLE™ Macintosh™	COMMODORE™ AMIGA™
Price	$799	$4675	$2795	$1795
CPU Speed MHz	68000 8.0	80286 6.0	68000 7.83	68000 7.16
Standard RAM	512K	256K	512K	256K
Number of Keys	95	95	59	89
Mouse	Yes	No	Yes	Yes
Screen Resolution (Non-Interlaced Mode) Color Monochrome	 640 x 200 640 x 400	 640 x 200 720 x 350**	 None 512 x 342	 640 x 200*** 640 x 200***
Color Output	Yes	Optional	None	Yes
Number of Colors	512	16	None	4096
Disk Drive	3.5"	5.25"	3.5"	3.5"
Built-in Hard Disk (DMA) Port	Yes	Yes	No	No
MIDI Interface	Yes	No	No	No
No. of Sound Voices	3	1	4	4

**With optional monochrome board (non bit-mapped)
***Interlace Mode – 640 x 400

*Plus applicable local taxes. $999.95 with color monitor.
All prices are manufacturer's suggested retail list.

ATARI®

Power without the price.