NO. 59
APRIL 1988

DISK VERSION $12.95

THE #1 MAGAZINE FOR ATARI® COMPUTER OWNERS

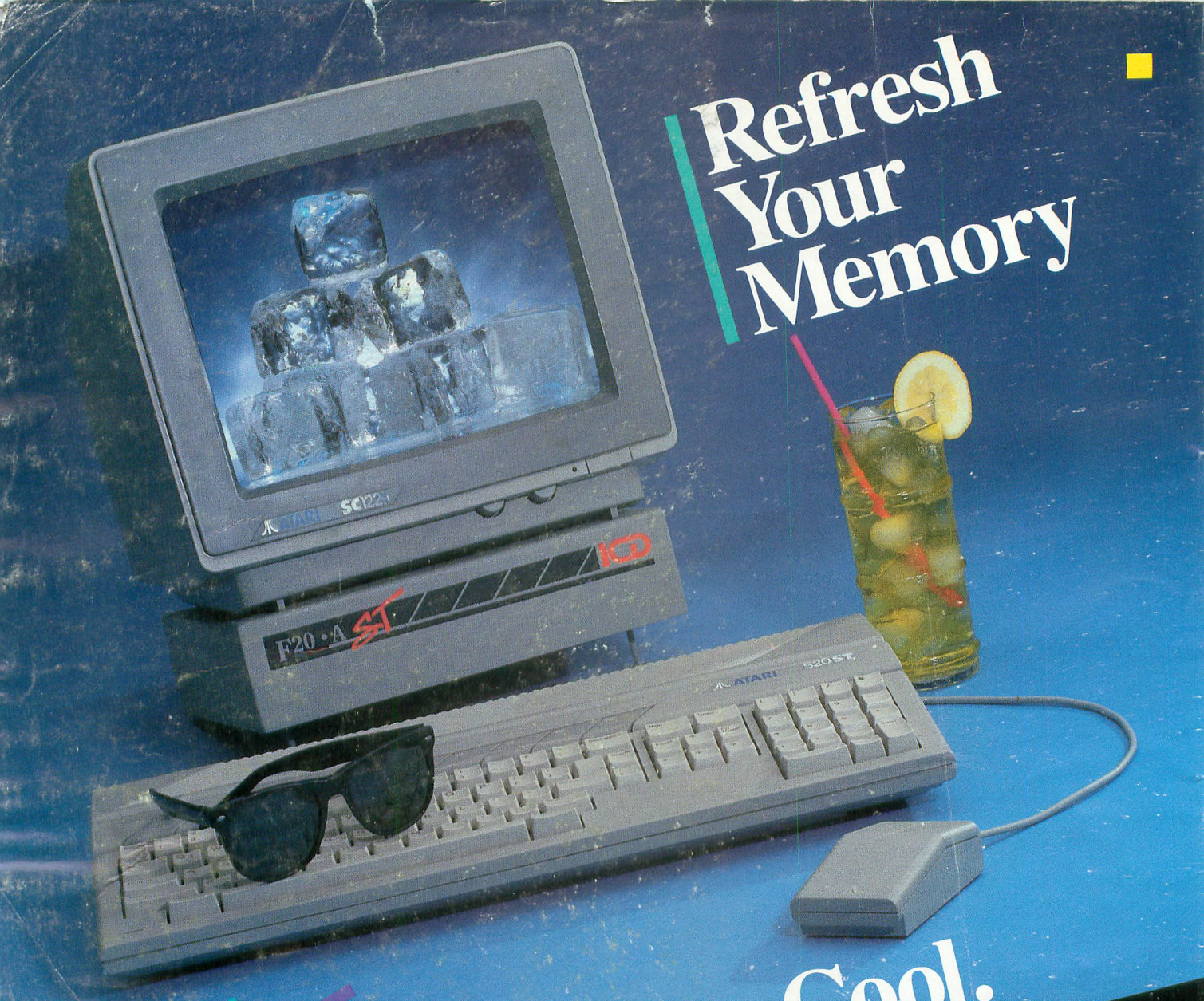
ANALOG COMPUTING™

BUYER'S GUIDE:
Shopping for the computer user

PLUS:
Upward
Presentation Graphics
The Wizard, Part 2

AND OUR SPECIAL SECTION:
VIDEO GAME DIGEST

Refresh Your Memory



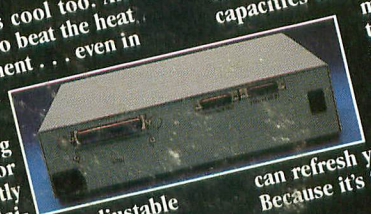
And Keep Your Cool.

Introducing the ST Hard Drive System from ICD that refreshes your memory better than any other ST hard drive around. *No problem.*

It's the drive that not only looks cool, but stays cool too. All because of a built-in fan that knows exactly how to beat the heat and maintain a calm, cool and collected environment . . . even in *your* most heated situations. *No sweat.*

And, it's the hard drive that takes a refreshing approach to aesthetic case design as well. See for yourself. It's easy on the space, fitting perfectly under the monitor. And it's easy on the eyes, tailored to look great in the company of your Atari ST. With adjustable front legs, your monitor gets the lift it needs for comfortable viewing. *No strain.*

Despite a sleek and compact exterior, the ICD ST Hard Drive System is packed full of overwhelming enhancements. Like an internal clock that tags each file with up-to-the-minute time and date information. Not to mention expansion capabilities that welcome the connection of up to six SCSI devices and daisy-chaining Atari's DMA Bus (ACSI). It's available in more memory capacities than you can imagine. With storage ranging from 20 megabyte systems up to 280 megabytes. And, there's dual drives too, that double your protection and double your confidence. *No stress.*



So, the next time you think about a hard drive for your Atari ST, think about the countless ways we can refresh your memory. It's the only drive worth remembering. Because it's from ICD. *No wonder.*

For further product information, please call or write for our catalog today.

ICD

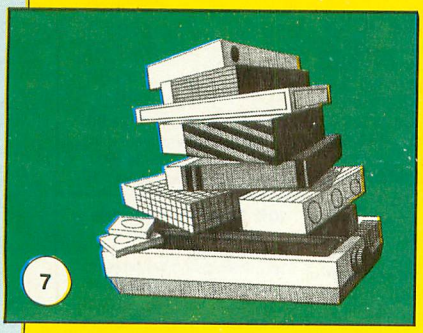
1220 Rock Street
Rockford, IL 61101-1437
(815)968-2228
MODEM: (815)968-2229
FAX: (815)968-6888

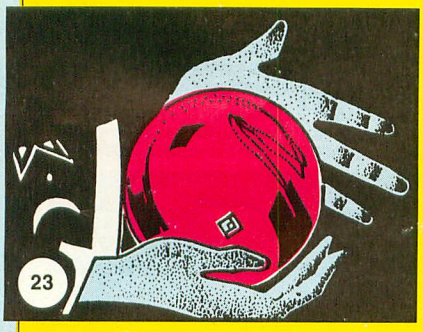
Atari ST is a trademark of Atari Corporation.

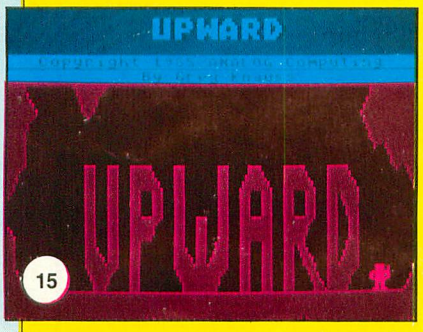
ISSUE 59
APRIL 1988

CONTENTS

FEATURES







REVIEWS

COLUMNS

ANALOG Computing (ISSN 0744-9917) is published monthly by L.F.P., Inc., 9171 Wilshire Blvd., Suite 300, Beverly Hills, CA 90210. U.S. subscription: $28 for one year (12 issues), $52 for two years (24 issues), $76 for three years (36 issues). Foreign subscription: Add $7 per year. Single copy $3.50 (add $1 for postage). Change of address: six weeks advance notice, and both old and new addresses are needed. POSTMASTER: Send change of address to: **ANALOG Computing** Magazine, P.O. Box 16927, North Hollywood, CA 91615. Second-class postage paid at Beverly Hills, CA, and additional mailing offices.

CIRCLE #102 ON READER SERVICE CARD

CIRCLE #103 ON READER SERVICE CARD

ANALOG COMPUTING STAFF

Editors/Publishers
MICHAEL J. DES CHENES
LEE H. PAPPAS

Managing Editor
KATHY WIESNER

Technical Editors
CHARLES BACHAND
CLAYTON WALNUM

East Coast Editor
ARTHUR LEYENBERGER

Midwest Editor
MATTHEW J. W. RATCLIFF

West Coast Editor
CHARLES F. JOHNSON

Contributing Editors
LEE S. BRILLIANT, M.D.
ARNIE KATZ
BILL KUNKEL
STEVE PANAK
D.F. SCOTT
JOYCE WORLEY

Contributing Artist
LINDA RICE

Graphics
SHERRY MORIN

Type Production
EDYTHE STODDARD

Advertising Manager
JE Publishers Representative
(213) 467-2266
(For regional numbers, see map)

Accounting/Circulation
ROBIN LEVITSKY

Production
DONNA HAHNER
VICE PRESIDENT, PRODUCTIONS

Contributors
GREG KNAUSS
MACK McLEOD
JEFFREY A. SUMMERS, M.D.

U.S. newsstand 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.

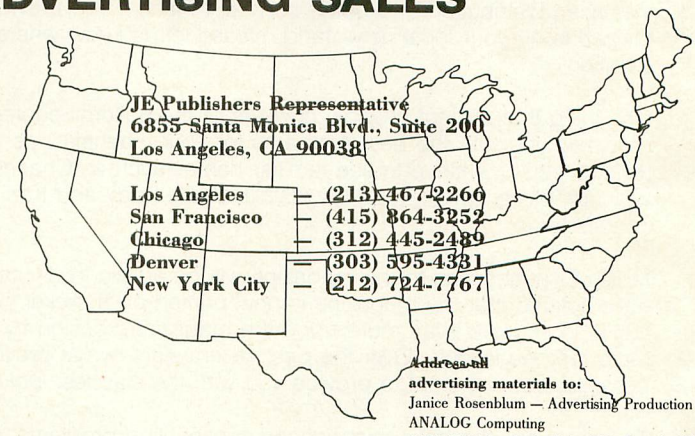
Correspondence regarding subscriptions, including problems and changes of address, should be sent to: **ANALOG Computing**, P.O. Box 16927, North Hollywood, CA 91615, or call 1-818-760-8983.

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
New York City	(212) 724-7767

Address all advertising materials to:
Janice Rosenblum — Advertising Production
ANALOG Computing
9171 Wilshire Blvd., Suite 300, Beverly Hills, CA 90210.

PERMISSIONS

SUBSCRIPTIONS

ANALOG Computing, P.O. Box 16927, North Hollywood, CA 91615, (818) 760-8983. Payable in U.S. funds only. U.S.: $28-one year, $54-two years, $76-three years. Foreign: Add $7 per year. For disk subscriptions, see the cards at the back of this issue.

AUTHORS

When submitting articles and programs, both program listings and text should be provided in printed and magnetic form, if possible. Typed or printed text copy is mandatory, and should be in upper- and lowercase, with double spacing. If a submission is to be returned, please send a self-addressed, stamped envelope.

In addition, any programs used must state that they are taken from **ANALOG Computing** Magazine. For further information, write to **ANALOG, P.O. Box 23, Worcester, MA 01603.**

Editorial

Hey, long time, no see.

That's what you're thinking, right?

It *has* been a long time, no doubt about that—too long for anyone's comfort, least of all for you, our faithful readers. But the wait is over. You can relax now: we come bearing good news.

For those of you who aren't aware of what's been happening, an explanation is owed. **ANALOG Computing** has been in transition for the past several months. The magazine is now under new ownership, and based in Beverly Hills, California, rather than Worcester, Massachusetts, where it has been for the past seven years. The increased financial backing provided by the new owners will bring many new and exciting improvements to the magazine, not the least of which are increased distribution and better service. In fact, if you previously had trouble finding copies of the magazine on your local newsstand, please let us know where you live, so we can remedy the situation.

One thing that will *not* change, however, is the editorial content of the magazine. We'll still be providing you with the best programs for your machines, as well as up-to-the-minute news and reviews. You'll continue to see familiar names such as Charles Johnson, Art Leyenberger, Clayton Walnum, Steve Panak, Matthew Ratcliff, Andy Eddy and Karl Wiegers, as well as our other contributors.

Most of the changes to the magazine will be artistic in direction. For instance, I'm sure all of you have noticed that the magazine is now printed on a slicker paper. We hope, in the months to come, to improve even more upon the magazine's design by adding more color and incorporating more creative layouts. Over the past seven years, we've provided you with the classiest magazine possible; now we want to provide you with the classiest *looking* magazine as well.

Those of you who have experienced subscription problems will be delighted to know that a new subscription fulfillment service has been selected for the magazine. All the subscription mix-ups that plagued us (and you) in the past will soon be corrected—just another example of the improved service you will experience under the new owners.

Your patience over the past few months has been more greatly appreciated than you will ever know, and we at **ANALOG Computing** are looking forward to many more years of offering you the kind of Atari coverage you've come to expect from us.

Yes, it's been a long wait. But the new **ANALOG** is what we've *all* been waiting for.

Lee H. Pappas
Publisher
ANALOG

ANALOG's new customer service number is (818) 760-8983. You may also write to:

ANALOG Computing
P.O. Box 16927
North Hollywood, CA 91615

ST-Log
P.O. Box 16928
North Hollywood, CA 91615

The above phone number and addresses are for subscription matters only.

Reader comment

Streamlining *Streamliner.*

I recently discovered a minor bug in my "Streamliner" program, in issue 56 of **ANALOG Computing**. Specifically, if you try to Streamline a file that won't fit in memory, the program kind of, uh, dies. The logic to handle this is already in the program, but a dumb mistake on my part causes it not to work. To fix it, place a disk with an unlocked copy of D:STRLINER.OBJ in drive 1 and *carefully* type in and run the following:

```
10 REM STREAMLINER V1.0 PA
TCH
20 REM by James Hague
30 OPEN #1,12,0,"D:STRLINE
R.OBJ"
40 FOR A=1 TO 908:GET #1,B
:NEXT A
50 PUT #1,160:CLOSE #1
```

When you see the *READY* prompt, the bug will have been corrected. To change the assembly language source code to reflect this fix (not necessary, unless you want to fool with the assembly version), change the LDY $FF in the PUTBUF routine (page 39) to LDY #$FF. Assembly programmers will recognize my mistake right away—I wish I had, sooner.

James Hague
Richardson, TX

Three cheers for Miner 2049er.

I've been an avid reader of **ANALOG Computing** since 1982 (issue 14) and have most of the back issues. Like many other Atarians, I started with an 800, went on to the 130XE, and now own a 1040ST. I sold my trusty 800, but still have the 130XE and now enjoy the best of both worlds. I like **ANALOG** and **ST-Log**, but your Four-Star Software issue prompted this, my first "letter to the editor."

Over the years, I've seen many articles featuring "the all-time greats" for the 8-bit Atari. I've tended to agree with the selections made by various writers. But I feel the time has come to stand up for truth, justice and the "American Way." Why has no one bothered to include Miner 2049er by Bill Hogue? This game is, without a doubt, one of the finest ever written for *any* computer! It has all of the elements that make a computer game great, yet it almost always gets overlooked. And the sequel, Bounty Bob Strikes Back, is every bit as good as the original. I've seen this game selling for as little as $2.99, and would say to anyone who's ever held a joystick: go out and get Miner 2049er—you won't be sorry!

Jasper Wilson
Tacoma, WA

A *Four-Star* salute and an update.

I loved your "Four-Star Software" article in the July/August issue, but would like to contest your omission of one program in the "Languages" section.

On the basis of your review of the Advan BASIC Compiler (issue 45), I closed my eyes and sent away for it and a couple of extras. I was not disappointed. No other BASIC for the Atari comes close. Advan is such a broad interpretation of BASIC that it really isn't BASIC. Who ever heard of BASIC commands for display list interrupts, PM graphics, scrolling, etc?

Unfortunately, Advan entered the market rather late in the game, so it hasn't taken off as it should have. If Advan would only get more exposure, who knows what could happen?

Mike Loader
Richelain, Quebec

Glad to get another opinion; thanks.

We'd like to inform readers that the Indus Disk Drive praised in our "Four-Star" article is, indeed, still manufactured—by Future Systems, 21634 Lassen Street, Chatsworth, CA 91311. Carried by major distributors, it now sells for only $279, not $449 as we mentioned. —*Ed.*

Just one more.

Can you stand to print yet another correction to Jeff Killeen's "CheckWriter" enhancement (issue 53) to Clayton Walnum's "MicroCheck" (issues 27 and 28)?

I've discovered that, if you're using the AUTOmatic monthly deduction feature of "MicroCheck," the "CheckWriter" modification hangs up when it encounters this.

The cure is very simple: just add the following line to "CheckWriter's" Listing 7:

```
212 IF FILE$(X+1,X+4)="AUT
O" THEN 245
```

When "CheckWriter" encounters this line, it skips over the AUTOmatic entry and goes to the next check.

Charles A. Cole
Sierra Vista, AR

Going to bat for the little guy.

I enjoy "Panak strikes!" and was glad to see a review of Autoduel in the September issue. In the review, Steve Panak included Origin Systems, Inc. as an affiliated label of Electronic Arts. As of August,

we are an affiliated label of Broderbund Software. This announcement may not have arrived at **ANALOG Computing** before press time.

I had some additional thoughts on the column. Consolidation of publishers helps balance the power of concentrated distributors like Ingram, Softsel, MicroD, etc. This shouldn't affect opportunities for small companies or individuals, though. Most of the best products to come out the last few years have come from small development organizations. Concentrating on development instead of marketing may be a large part of their advantage.

We at Origin participate in an affiliated label program because it allows us to concentrate on product development efforts. However, we are always actively searching for new authors, often in competition with companies like EA or Broderbund. Competition for authors should help a lot of good products make it to market—and help unknown authors make it in the industry.

Kathryn A. Roy, Director of Marketing
Origin Systems, Inc.

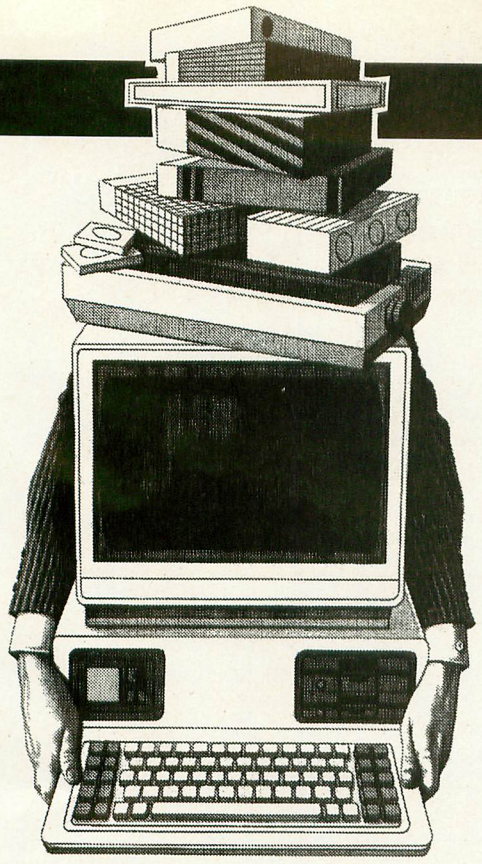
Wrestling wrong wrighted.

In issue 52's "ST notes" column, D.F. Scott wrote a review of Championship Wrestling from Epyx. Hulk Hogan, as stated in the review, did *not* assault ABC News correspondent John Stassel. Professional wrestler Dave Shultz was the one who hit Mr. Stassel. Perhaps you were thinking of the incident where Hulk Hogan knocked out Richard Belzer.

You owe Hulk Hogan, the World Wrestling Federation and your readers an apology.

Sincerely yours,
Eric Shaffer
Concerned reader and
professional wrestling fan

We know we don't want to offend the World Wrestling Federation or Hulk Hogan. Therefore, we ask (beg?) that they accept our apologies for the mix-up.—Ed.



Shopping for the computer user.

by Matthew J.W. Ratcliff

While reflecting one day on what I have and don't have in the way of hardware and software for my 8-bit system, I decided to make a list. This list consisted of many old favorites of mine, a number of new 8-bit products, and even a few ideas for future 8-bit applications. To make a short story long, this list grew, and grew, and grew. . .until, finally, it was magically transformed into a 5-page article.

Although you will find a few mindless, blasting video games on this list—a necessity in any Atari shopper's guide—you'll also find listed a number of games that stimulate the mind and actually teach the user—some more subtly than others.

So without further ado, here's the Matt Ratcliff guide to "Shopping for the computer user."

High-tech hardware.

The neatest technical gizmo ever to come out for the 8-bit Atari has to be ICD's **Multiple Input Output Board** (or MIO Board). This device gives you either 256K ($199) or 1 meg ($349) of RAM, to use as super high speed and extremely reliable RAMdisks and/or a printer buffer. This box also provides a standard printer and modem interface. (Most 64K printer buffers cost as much or *more* than the 256K version of the MIO, and look what else it does!) Finally, there's the hard drive interface, which allows you to attach over 100 megabytes of hard disk storage to your little old 8-bit Atari (an external hard drive, controller, case and power supply are required). ICD had also planned an 80-column add-in board as well, but at this writing, that project is on hold indefinitely.

If you need a simple printer interface, ICD's **Printer Connection** for $60 will do the job. Their high-level **SpartaDOS** is a great disk operating system for the power user ($35), and the **R-Time 8 cartridge** ($60) is just the card for keeping perfect time stamps on all your SpartaDOS files.

Atari received most of its new hardware from its Taiwan factories. The **XEP80**, an 80-column board, will sell for under $100. It gives the 8-bit Atari true 80-column text capability, with the added bonus of a printer interface.

The new **XE Game System** has also arrived. A 64K "game computer," the **XE Game System** is virtually identical in functionality to the 65XE, with the added bonuses of a detachable keyboard, Missile Command, Flight Simulator II and Blast 'Em video games. The remote keyboard should be nice for laptop typing during marathon editing sessions (like the one I'm having right now—at 4 a.m.)

The Atari **SX212**, a 1200-baud modem that connects directly to the ST or 8-bit with no extra interface required, sells for less than $100. It won't be very useful to 8-bit users, however, without Keith Ledbetter's latest version of **Express**, a telecommunications software package. This program should be on sale by the time you read this.

Atari's double-sided, double-density **XF551** floppy disk should also be available now. This 360K floppy disk, operating under the OSS developed ADOS, will transfer data between drive and computer nearly three times faster than the old 1050 disk drive. With the added storage, new versions of popular games can be stored on one disk (instead of two, three or four), eliminating the need for distracting disk swaps.

Announcements about the new Atari hardware should be taken with a *large* grain of salt. Getting hardware from Sunnyvale, California to the distributors and dealers will take a while. If you have a reputable local Atari dealer, check with him about final prices and delivery. I wouldn't recommend ordering any of the new hardware by mail—just yet. (Some mail order houses will accept your orders (and money) weeks in advance of actual product availability.)

High-level hacking.

If you're tired of BASIC and losing interest in hacking away at assembly language, you now have several options to move up to a high-level language.

Lightspeed C ($40) from Clearstar Softechnologies is the most complete C compiler package you'll find for the Atari 8-bit. You can comfortably develop your own C code with its support for BASIC-like functions. This package also follows the Kernighan and Ritchie C standard closely, allowing you to learn C as it was meant to be. Writing programs on the

(continued on page 11)

S & S Wholesalers, Inc.

FREE FREIGHT ON ORDERS OVER $100.00.

ALL PRICES REFLECT A 3% CASH DISCOUNT. ADD 3% FOR CREDIT CARD PURCHASES. COD ORDERS ADD $4.00 PER BOX. ORDERS UNDER $100.00 ADD $4.00 SHIPPING AND HANDLING. QUANTITIES ON SOME ITEMS MAY BE LIMITED. PRICES ARE SUBJECT TO CHANGE WITHOUT NOTICE. APO. FPO ADD 6% ADDITIONAL SHIPPING. FOREIGN ORDERS ACTUAL FREIGHT CHARGED. ALL SALES ARE FINAL. RETURN OF DEFFECTIVE MERCHANDISE FOR REPLACEMENT ONLY WITH AUTHORIZATION. AUTHORIZED RETURNS ARE SUBJECT TO A RESTOCKING FEE. COMPLETE LISTING OF ALL SALES POLICIES AVAILABLE UPON REQUEST.

*EXCEPT ON ITEMS MARKED BY AN ASTERIKS.

ATARI 8 BIT WORD PROCESSING SYSTEM

130XE COMPUTER
1050 DISK DRIVE
1027 LETTER-QUALITY PRINTER
TEXT PRO WORD PROCESSOR
DATA PRO DATA BASE
10 5¼" BLANK DISKETTES

487.77

XM 301 MODEM

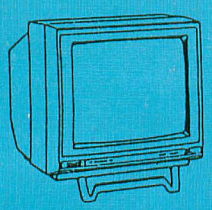
37.77

ST HARDDRIVE

SHD 20MB HARDDRIV

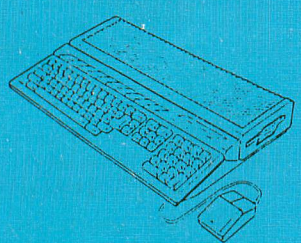
549.90

12" COLOR MONITOR



149.90*

ATARI 1040ST



CALL...WE BEAT ALL PRICES

SOFTWARE

ABACUS

ASSEMPRO	35.97
DATATRIEVE	29.97
PAINT PRO LIBRARY	17.97
PAINTPRO	29.97
PCB DESIGNER	237.00
POWER PLAN COLOR	47.97
POWER PLAN MONOCHROME	47.97
ST GEM	8.97
ST GRAPHICS & SOUND	8.97
TEXTPRO	29.97

ABSOFT

FORTRAN	119.97

ACCESS

EXEC. DISK-LEADER BOARD	11.97
LEADER BOARD	23.97
TOURN. DISK-LEADER BOARD	11.97
TRIPLE PACK	11.97
10TH FRAME BOWLING	23.97

ACCOLADE

ACE OF ACES	17.97
FAMOUS COURSE DISK	11.97
FIGHT NIGHT	17.97
HARDBALL	17.97
MEAN 18	26.97
SPY VS SPY I & II	8.97

ACTIVISION

BORROWED TIME	29.97
GHOSTBUSTERS	8.97
GREAT AMERICAN CC RACE	8.97
HACKER	8.97
HACKER II	23.97
LITTLE COMPUTER PEOPLE	14.97
MINDSHADOW	29.97
MUSIC STUDIO	29.97
PAINTWORKS	23.97
PITFALL	4.00
PORTAL	29.97
SHANGHAI	23.97
TASS TIMES	23.97

AEGIS

AEGIS ANIMATOR ST	47.97
ART PACK #1 FOR ANIMATOR	20.97

ARTWORK

BRIDGE 4.0	17.97
BRIDGE 5.0	20.97
COMPU-BRIDGE	17.97
HOLE IN ONE GOLF	11.97
STRIP POKER	23.97
STRIP POKER DATA 1 FEMALE	11.97
THAI BOXING	8.97

ATARI

ARCADE CHAMP KIT	19.17
ASTEROIDS	10.17
ATARILAB LIGHT MODULE	29.37
ATARILAB STARTER KIT	44.97
CAVERNS OF MARS	12.57
COMPU-BRIDGE	17.97
CONVERSATIONAL FRENCH	10.00
CONVERSATIONAL SPANISH	10.00
CONVERSATIONAL ITALIAN	10.00
DEFENDER	12.57
DIG DUG	12.57
E.T.	7.75
EASTERN FRONT	10.17
FAMILY FINANCE	6.00
GALAXIAN	10.17
HOME FILE MANAGER	16.77
JUGGLES RAINBOW	12.57
LEARNING PHONE	17.97
LOGO KIT	29.97
MICKEY & GREAT OUTDOORS	19.77
MISSILE COMMAND	10.17
MUSIC COMPOSER	16.77
PAINT	16.77
PROOFREADER	7.75
SILENT BUTLER	17.97
SKYWRITER	12.57
SPACE INVADERS	10.17
STAR RAIDERS	10.17
STAR RAIDERS II	12.57
THAI BOXING	8.97
TIMEWISE	2.99
VISICALC	17.77

AVALON

COMBAT CHESS	15.00
GULF STRIKE	18.00
JUPITER MISSION 1999	30.00
MISSION ON THUNDERHEAD	15.00
QUEST SPACE BEAGLE	21.00
T.A.C.	24.00

BATTERIES INCL.

B-GRAPH	23.97
DEGAS	23.97
HOME PAK	29.97
I*S DEGAS ELITE	47.97
I*S PAPERCLIP ELITE	59.97
I*S TALK	29.97
ISGUR PORTFOLIO SYSTEM	119.97
PAPERCLIP	35.97
PAPERCLIP W/SPELLPACK	35.97
THUNDER	23.97
TIME LINK	29.97

BRODERBUND

BANK STREET WRITER	29.97
CHAMP LODE RUNNER	17.97
KARATEKA	17.97
PRINT SHOP	26.97
PRINT SHOP COMPANION	20.97
PRINT SHOP GRAPHICS LIB 1	14.97
PRINT SHOP GRAPHICS LIB 2	14.97
PRINT SHOP GRAPHICS LIB 3	14.97
SYNCALC	29.97
SYNCALC TEMPLATE	11.97
SYNCHRON	23.97
SYNCOMM	23.97
SYNFILE	29.97
SYNSTOCK	23.97

DAVIDSON

MATH BLASTER	29.97
WORD ATTACK	29.97

MASTERTRONIC

NINJA	11.99
RENEGADE	11.99

INFOCOM

BALLYHOO	23.97
BUREAUCRACY	23.97
DEADLINE	29.97
ENCHANTER	17.97
FOOBLITZKY	23.97
HITCHHIKERS GUIDE GALAXY	17.97
HOLLYWOOD HIJINX	23.97
INFIDEL	26.97
INFOCOM SAMPLER	4.77
LEATHER GODDESSES PHOBOS	23.97
PLANETFALL	23.97
SORCERER	26.97
SUSPECT	26.97
SUSPENDED	29.97
TRINITY	23.97
WISHBRINGER	20.97
ZORK I	23.97
ZORK TRILOGY	35.97

ISD MARKETING

VIP PROFESSIONAL	94.90

MARK WILLIAMS

MARK WILLIAMS C	107.87

DITEK SOFTWARE

ST ACCOUNTS

DR. T's

CZ PATCH
KEYBOARD CTRL SEQUENCER
MIDI RECORDING STUDIO
THE COPYIST

ELECTRONICA ART

ADVENTURE CONST SET
ARCTIC FOX
AUTODUEL
CHESSMASTER 2000
FINANCIAL COOKBOOK
GOLDEN OLDIES
M.U.L.E.
MAIL ORDER MONSTERS
NEW TECH. COLOR BOOK
OGRE
PINBALL CONST. SET
QUIZAM!
RACING DESTRUCTION SET
SKYFOX
STAR FLEET I
ULTIMA III
ULTIMA IV

EPYX

CHAMP WRESTLING
ROGUE
SUB BATTLE
WINTER GAMES
WORLD GAMES
WORLD KARATE CHAMP

FIREBIRD

GOLDEN PATH
GUILD OF THIEVES
PAWN
STARGLIDER
TRACKER
UNIV. MILITARY SIMULATOR

GAMESTAR

CHAMP BASEBALL
GBA CHAMP BASKETBALL
GFL CHAMP FOOTBALL
ON TRACK RACING
STAR LEAGUE BASEBALL
STARBOWL FOOTBALL

HABA

HABA SPECIAL
HABA WRITER
HABA WRITER II
HABACOM
HABADEX PHONEBOOK
HABAMERGE
HABASPELL
HABAVIEW
HIPPO-C
HOME ACCOUNTANT
MAILROOM
PRO BUSINESS LETTERS

1-800-233-6345 1-800-331-7054

ARI SF354
19.90
WITH PURCHASE OF DRIVE CLEANER

PANASONIC 1080i
149.90*
*WITH PURCHASE OF INTERFACE

AVATEX 1200 MODEM
79.99
*WITH PURCHASE OF CABLE

ATARI 1050 DISK DRIVE
169.90*
*WITH PURCHASE OF CLEANER

D PROCESSING SYSTEM
COMPUTER
DISK DRIVE
MONITOR
OLUMN PRINTER
ER CABLE
" DISKETTES
PROCESSING PROGRAM
VSE
799.90

RADAR DETECTOR
SUPERHETRODYNE
69.90

IF WE DO NOT HAVE THE LOWEST PRICE ON AN ITEM, CALL AND ALLOW US TO GIVE YOU THE BEST BUY FOR YOUR $$$$.

METRCOMCO
BRIDGE LISP	119.97
ASCAL	59.97
CE C	89.97
RO ASSEMBLER	47.97
	17.97
COMCO BCPL	89.97
COMCO MAKE	41.97

MI GRAPH
DRAW	47.97
DRAW FONT PACK 1	23.97

MICHTRON
	17.97
IATOR	23.97
.0	29.97
NDER	17.97
OS	23.97
NERMAN	29.97
SHELL	23.97
O	23.97
BASIC COMPILER	47.97
BASIC INTERPRETER	47.97
O RUNNER	23.97
TE KIT II	23.97
ACHE	23.97
OPY	47.97
SK PLUS	23.97
JPE	17.97
ITRON BBS 2.0	47.97
CE SHUTTLE II	23.97
BANDITS	23.97

MICROPROSE
IFLICT IN VIETNAM	23.97
STRIKE EAGLE	20.97
SHIP	20.97
T SERVICE	20.97
O FLIGHT	14.97
IRE ACE	17.97
GUNNER SERIES	14.97

MICROLEAGUE
OLEAGUE DATA DISK	11.97
OLEAGUE BASEBALL	23.97
LD SERIES DISK	11.97
TEAMS DISK	11.97
WORLD SERIES	11.97

MINDSCAPE INC.
NCE OF POWER	29.97
N WRESTLE	17.97
ACCUS	29.97
NDER OF THE CROWN	29.97
ROLLER	29.97
RATOR	17.97
ERTYPE	23.97
	29.97
AD	29.97

MONOGRAM
ARS & SENSE	59.97

PEACHTREE
BACK TO BASIC ACCOUNTING	57.77

PECAN
UCSD PASCAL	47.97

PHILON, INC.
FAST BASIC-M COMPILER	77.40
FAST C COMPILER	89.40
FAST COBOL COMPILER	179.40
FAST FORTRAN COMPILER	179.40
HENRY'S FUNDAMENTAL BASIC	29.40

PROG. PERIPHERALS
LOGISTIX JR. 520 VESION	59.97
LOGISTIX 1040 1 MEG. REQ.	89.99
SUPERBASE GEM	89.97
SUPER SCRIPT	29.97

PROFESSIONAL SOFTWARE
FLEET SYUSTEM 2	41.97

REGENT
BUSINESS TEMPLATE I	29.97
INVENTORY MANAGER	47.97
MAIL MERGE DATA BASE	14.97
REGENT BASE	59.97
REGENT PAK	29.97
REGENT WORD	29.97
REGENT WORD II GEM VER.	47.97
SPELL CHECKER	29.97

ROYAL SOFTWARE
E-Z DATA	29.97
E-Z GRAPH	41.97
E-Z SCHEDULER	23.97
E-Z SPELL ENGLISH	23.97
E-Z SPELL FRENCH	23.97
E-Z SPELL GERMAN	23.97
E-Z THESAURUS	23.97
HELP CALC	14.97
HELP-MATE	11.97
NEW E-Z CALC	41.97
PAYROLL MASTER	47.97

SHELBOURNE S/W
ST POOL	20.97
ST SHUFFLEBOARD	17.97

SIERRA
ACCOUNTS RECEIVABLE	35.97
BLACK CAULDRON	23.97
CASH DISBURSEMENTS	35.97
DONALD DUCK'S PLAYGRD	14.97
GENERAL LEDGER	35.97
KING'S QUEST II	29.97
KING'S QUEST	29.97
KING'S QUEST III	29.97
ULTIMA II	4.50
WINNIE THE POOH	14.97
3-D HELICOPTER SIMULATOR	29.97

SPECTRUM HOLOBYTE
FALCON	23.97
GATO	23.97
PT BOAT	23.97

PRECISION SOFTWARE
ACTION	47.40
ACTION TOOL KIT	17.97
BASIC XE	47.40
BASIC XL	35.40
DOS XL WITH BUG 65	17.97
MAC/65	47.40
PERSONAL PASCAL	44.97
PERSONAL PROLOG	53.97

SPINNAKER
HOMEWORK HELPER MATH	29.97
HOMEWORK HELPER WRITING	29.97

SPRINGBOARD
CERTIFICATE MAKER	29.97
CERT. MAKER LIBRARY #1	20.97

STRATEGIC SIMULATIONS
BATTALION COMMANDER	23.97
BATTLE FOR NORMANDY	23.97
BATTLE OF SHILOH	23.97
BROADSIDES	23.97
COLONIAL CONQUEST	23.97
COMPUTER BASEBALL	23.97
COMPUTER QUATERBACK	23.97
FORTRESS	8.97
GEMSTONE WARRIOR	8.97
GETTYSBURG	35.97
KAMPFGRUPPE	35.97
NAM	23.97
PHANTASIE	23.97
PHANTASIE II	23.97
PHANTASIE III -WRATH/NIKADEMUS	23.97
RAILS WEST	23.97
REBEL CHARGE: CHICKAMAU	29.97
RINGS OF ZILFIN	23.97
ROADWAR 2000	23.97
SHARD OF SPRING	23.97
SIX GUN SHOOTOUT	23.97
USAAF	35.97
WAR IN RUSSIA	47.97
WARGAME CONST. SET	17.97
WIZARDS CROWN	23.97

SUBLOGIC
FLIGHT SIM. II COLOR	32.47
FLIGHT SIM II MONOCHROME	32.47
JET	32.47
NIGHT MISSION PINBALL	19.47
SCENERY DISK 1 TEXAS	12.97
SCENERY DISK 2 ARIZONA	12.97
SCENERY DISK 3 CALIFORNIA	12.97
SCENERY DISK 4 WASHINGTON	12.97
SCENERY DISK 5 UTAH, CO, WY	12.97
SCENERY DISK 6 KANSAS, NEB	12.97
SCENERY DISK 7 FL, NC, SC	12.97
STAR SCENERY-JAPAN	12.97
STAR SCENERY-SAN FRAN.	12.97
WESTERN SCENERY DISK PACK	64.97

CLOSEOUTS
POPEYE	6.77
Q'BERT	6.77
FROGGER	6.77
JAMES BOND	6.77
SUKPER COBRA	6.77
STAR WARS	6.77
CASTLE WOLFENSTEIN	6.77
BEYOND CASTLE WOLF.	6.77
RACK-EM	6.77
WIZARD OF WOR	6.77

THUNDER MOUNTAIN
ABC's W/ THE TINK TONKS	5.97
BEING A SMART THINKER	5.97
COUNT & ADD W/TINK TONKS	5.97
DEVELOP THINKING SKILLS	5.97
SONGWRITER	5.97
SPELLING WITH THE TINK TONKS	5.97
SUBTRACT W/ THE TINK TONKS	5.97
TINKA'S MAZES	5.97

TIMEWORKS
DATA MANAGER ST	47.97
SUPER GRAPHICS ST	29.97
SWIFTAX	47.97
SWIFTCALC ST	47.97
SYLVIA PORTER	47.97
SYLVIA PORTER VOL. 2	47.97
WORD WRITER ST	47.97

WEEKLY READER
STICKYBEAR ABC	17.97
STICKYBEAR BASKETBOUNCE	17.97
STICKYBEAR NUMBERS	17.97
STICKYBEAR SHAPES	17.97

UNISON WORLD INC.
ART GALLERY 1	17.97
ART GALLERY II	17.97
PRING MASTER PLUS	23.97

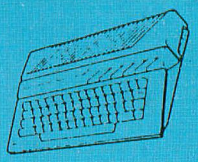
WORD PERFECT
WORD PERFECT 4.1	237.00

X-LENT SOFTWARE
HYPNOSIS W/BRNWAV. SYNCH	17.97
ICON COLOR DISK	11.97
ICON MONOCHROME DISK	23.97
MEGAFILER	11.97
MEGAFONT ST	23.97
MEGAFONT II+	14.97
MINIATURE GOLF PLUS	17.97
MUSIC BOX RGB	29.97
PAGE DESIGNER	17.97
PICTURE DISK	11.97
PRINT MASTER INTERFACE	17.97
PRINT SHOP INTERFACE	17.97
RUBBER STAMP	17.97
ST MUSIC BOX	17.97
TRIVIA MANIA	17.97
TYPESETTER ELITE	29.97
TYPESETTER-MONO & RGB	23.97
WRITE 90	17.97
1ST XLENT WORD PROCESSOR	17.97

8 BIT HARDWARE
800XL	89.90
1050 DISK DRIVE	229.90
INDUS GT	199.90
XM801 PRINTER	199.90
65XE	89.90
XM301 MODEM	37.77
ATARI LIGHT PEN	42.77
1027 PRINTER	124.77
20 MB HARD DRIVE	CALL

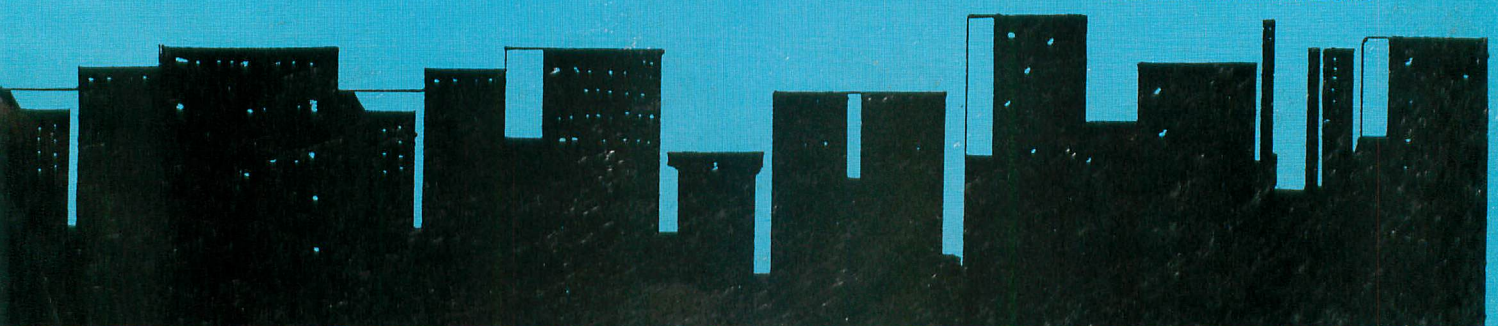
ST HARDWARE
520ST COMPUTER	359.90
1040ST COMPUTER	599.90
SM124 MONITOR	149.90
SC1224 MONITOR	329.90
SMM804 PRINTER	199.90
SF354 DISK DRIVE	CALL
SF314 DISK DRIVE	219.90

JOYSTICKS/ ACCESSORIES
WICO THE BOSS	10.77
WICO BAT	14.77
WKOS WAY	18.77
KRAFT	8.77
RECOTON BIG SHOT	11.77
5¼" DRIVE CLEANER	8.77
5½" DRIVE CLEANER	9.77
CRT CLEANER	14.77
800 XL COVER	7.77
520ST COVER	9.77
DRIVE COVER ST	7.77
1050 COVER	7.77
1025 COVER	7.77
1017 COVER	7.77
130 XE COVER	7.77
1224 COVER	11.77

130XE

99.99*
*WITH PURCHASE OF SOFTWARE PACKAGE

STAR NX10
179.90*
*WITH 2 RIBBON PURCHASE

CIRCLE #109 ON READER SERVICE CARD

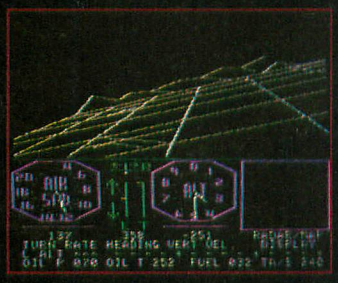


1987 - Expanding Scenery disk coverage; East Coast, Japan, & Europe

1985 - High-performance Jet flight simulator for the IBM, Commodore 64, and Apple II computers

1986 - Flight Simulator II for the 68000 computers

1982/1983 - Microsoft Flight Simulator & Flight Simulator II



1979 - 3D graphics applied to the original FS1 Flight Simulator for the new Apple II and TRS-80 computers

1977 - SubLOGIC's 3D graphics package in BASIC and M6800 Assembly Language

Ten Years of Technology and Dedication

Some say our technology has helped us define the state of the art in flight simulation. We believe our ten years of dedication have gotten us where we are today.

SubLOGIC's first black & white 3D graphics routines, developed in 1977, paved the way for our introduction to flight simulation and aerial combat gaming theory. Our second-generation Flight Simulator II was so well conceived that even we find it difficult to improve upon. Jet's spectacular land and sea battle scenarios set another classic milestone in state-of-the-art simulation gaming. Scenery concepts incorporated into SubLOGIC flight simulation products right from the start continue to evolve as we introduce new, more beautifully detailed areas of the world to explore. And coming in 1988 - a flight control yoke for even more realism!

We invite you to help us celebrate our first decade of success, and share in our anticipation of the next ten years to come. SubLOGIC tenth-anniversary promotional shirts and posters are available at your dealer or directly from SubLOGIC. See your dealer, or write or call us for more information.

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995
ORDER LINE: (800) 637-4983
(except in Illinois)

CIRCLE #120 ON READER SERVICE CARD

 continued from page 7

8-bit in C will give you a real head start if you move up to the ST, or into higher level programming classes in school, for example. It is a well integrated package, except for its limited line editor, and the fact that it doesn't take advantage of *any* Atari full screen editing capabilities. You can use other text editors, but then you lose the convenience of one integrated development environment.

If it's Pascal programming you want to learn, **Kyan Pascal** ($70) from Kyan Software, Inc. is a full implementation of "ISO Pascal." It has a full screen text editor, Pascal compiler, macro assembler, and a 300-page-plus manual. It's a great way to get into high-level language programming on a *low-level* machine like the 6502-based Atari.

Action! from OSS is another good high-level language implementation on the 8-bit. The only problem is that it's non-standard. Action! is not available for any other computers; however, the structured language programming concepts and techniques you learn in Action! will apply to any other structured language. Action! is probably the fastest, most tightly knit development environment of the languages listed here, since it is *supercartridge* based.

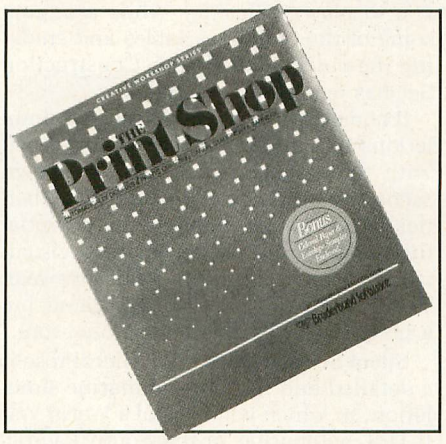
No programmer's library is complete without ***Mapping the Atari*** (revised edition) from *Compute!*, by Ian Chadwick. It is the most complete memory guide and tutorial manual you'll find for the 8-bit Atari. Selling for about $20, it can be found at most bookstores that carry Compute! Publications. The revised edition consists of expanded information for the XL and XE machines, including several new appendixes (an additional 60 pages or so).

The printworks, moneyworks, artworks and neatworks.

The Print Shop from Broderbund is one of the all-time, best-selling applications for the 8-bit graphics machines. This page designer lets you make greeting cards, posters and banners in a variety of ways. Custom fonts, icons and borders are supported.

The Print Shop Companion expands your font selections by twelve and border selections by fifty! It provides a new editor for creating not only your own custom icons, but fonts and borders too. The Companion can also generate calendars. If you're a regular Print Shop user, the Companion is a *must*.

High Tech's product list keeps growing and there's always a check mark in the Atari 8-bit column. It's one of the few companies left that consistently supports the 8-bit Atari across the board. Their specialty is low cost software that's graphic and printer oriented. They can help you create disk-based animated greeting cards with programs such as **Card Ware** and **Heart Ware**. Or you can design and print your own awards, coupons, checks or ribbons with **Award Ware**. A program quite competitive with Springboard's popular Certificate Maker (which is available for the ST only), Award Ware allows you to



The Print Shop.

"cut and paste" your awards from a large library of forms, borders, fonts and icons. A completed design can be printed on one of the many printers Award Ware supports (or you can use the built-in printer driver editor for odd machines).

If your interest in the Atari ST has been piqued by all the hoopla surrounding the buzzword *desktop publishing*, then **Word Magic** and **Graphic Magic** from The Catalog may be a good introduction for you. This pair of programs is an amazingly powerful word processor and graphics integration package for only $20. But, as is typical of products from The Catalog, documentation isn't all that terrific and you must print it yourself—a tedious process.

The Graphic Magic program can process MicroPainter, MicroIllustrator, KoalaPad, B/Graph and graphics mode 8 files. The Word Magic "view" function lets you preview scaled down pictorial representations of your text and graphics placement.

You can't read in the "view" mode, but can see general text format, page breaks and graphics placement. The processed graphics output files are *huge* (400 SD sectors), so you'll need lots of working disks and patience.

NetWorth by Mindscape is the complete home money manager. It lets you set up as many as 350 budget categories for tracking your actual income and expenses. You can keep track of banking and credit card transactions, with the ability to reconcile statements instantly. Up to ten bank accounts can be managed, and you can even print checks. It's being promoted as easy to use, with special "help" functions available on the screen at all times. The program "has been tested by Touche Ross & Co., Certified Public Accountants."

Now that the KoalaPad has virtually vanished from dealer's shelves, and the Atari Touch Tablet is scarcer than ever, a good graphics tablet for the 8-bit is hard to come by. Suncom, the well known joystick maker, offers the **Animation Station** ($90). This tablet is not unlike others for the Atari (it's even compatible with the MicroIllustrator software for the KoalaPad), but it comes with quite unique software; it's certainly a cut above the old MicroIllustrator standard. Its text handling features are phenomenal, featuring many different font styles and sizes, and even colored text. You'll be pleased to know that the software comes with a printer dump feature *built in*. You can cut and paste shapes, and portions of pictures, and even store them on disk. It comes with a full one-year warranty, and appears to be the best of the best in paint software for 8-bit Ataris.

Keep your computer, printer and disk drive neat and clean with **Classic Covers** ($8-12). These vinyl covers are labeled with the Atari logo and computer model they were designed for.

The arcade comes home.

The Great American Cross Country Road Race from Activision is an impressive conversion of the Sega coin-op game. Similar to Atari's Pole Position in some of its driving aspects, its horizons are much broader.

Your goal is to drive aross America within the allotted time. As you approach the big cities, the various skylines (such as the Gateway Arch in St. Louis) smoothly scroll up over the horizon. Displays are changed to reflect current weather conditions, and traffic congestion is accurately simulated, based on weather, location (near larger cities, traffic is worse) and time. You have an accurate odometer, radar detector, speedometer and gas gauge. In fact, if you run out of gas, you can even *push* your car to the next gas station.

Mastertronic game programs for the

Atari and Commodore generally occupy opposite sides of the same floppy disk. This means you can find Atari titles where only Commodore programs are generally carried. The average price for a Mastertronic game is about $10 list (typically $5-8 in the toy stores). They generally have good graphics, but not always top-notch playability. At these prices, however, you can't really miss with games like **Action Biker** and **Ninja Mission**. (Be sure the package says the Atari version is enclosed.)

Another arcade game worth looking into is **Battlezone**. I saw a superb beta test version at CES that certainly rivals the arcade original. John Skruch of Atari is making sure this one is done right—and it shows. In this "vector graphics" style game, you are looking out the viewport of a tank. The three-dimensional rocks, tanks and other obstacles are easily identified and attacked. You must act fast to avoid cracking up in this superb simulation. Battlezone is a must for any arcade addicts around the house.

Cool school.

Atari Planetarium from Atari Corp. is a great tool for helping astronomers locate and identify heavenly bodies. You can specify your location on earth and any part of the sky you wish to view, at any time in the past or future (over a span of 9999 years). An accurate representation of a clear sky is displayed. The 115-page manual helps you use the program *and* teaches about the universe. This is an excellent educational tool, even if you aren't an astronomer.

The most *fun* educational program I've ever used on the 8-bit has to be Randolph Constan's **Super 3D Plotter II** ($30) from Elfin Magic software. It lets you create three-dimensional images with a simple text or joystick entry editor. Once created, the image can be "spun" around on one or many axes. When viewing, "erase mode" may be turned off for an instant *spirograph*. Any object on the display may be dumped to the printer or a disk file. Using the proper creation methods, an object may be displayed with hidden line removal and even solid modeling (the sides shaded in one of three colors). It's all done in assembly language, so it's very fast. This program is easy to use, and comes with a good tutorial manual and complete keyboard reference card. Anyone taking a geometry class will learn a lot with this program—and have lots of fun, besides.

The Voice Master from COVOX ($90) is a sound digitizer and recognizer. It can be programmed to *learn*, *speak* and *recognize* new words (with special extensions to Atari BASIC). You can write very simple programs to drill your children on their spelling skills. Also, the computer can "hear" your children complete math problems that are displayed on-screen, then grade their work. Its potential as an educational tool is limited only by your imagination (and memory, of course).

Wargame Construction Set is not your average war game simulation. Most war games simulate specific battles, allowing you to play out history, while changing some of the battle's variables and studying the outcome. Wargame Construction Set has far more potential.

It comes with eight prefabricated simulations to battle out. Once you're bored with those, you may design your own scenarios, ranging from World War II battles to the Alamo. Imagine the potential this program has for helping you visualize your history lessons, year after year. This is not just another pretty game, but a potentially valuable educational tool.

Silent Service ($35) from MicroProse is a detailed and realistic submarine simulation, in which you control a World War II U.S. submarine in the South Pacific. You can navigate your sub with the aid of charts and maps, as well as a realistic periscope view. You can fire a deck gun or torpedoes. You make all the decisions, from where to patrol, to how to attack the enemy's forces.

Silent Service features player selected "reality levels" and time scaling for accelerated game play. You have a choice of a wide selection of historical scenarios, from individual attacks to patrol missions. This is a *fun* game to play, and you'll learn a lot about WWII in the process.

Chessmaster 2000 ($40) by Software Country (distributed by Electronic Arts) is the best of the best chess programs for the 8-bit. The chessboard is presented in a three-dimensional perspective, and can be rotated and viewed from four angles. This feature can be quite helpful when starting a game, and you're looking at a very crowded board. The view may be switched to a simple two-dimensional overhead shot, as well.

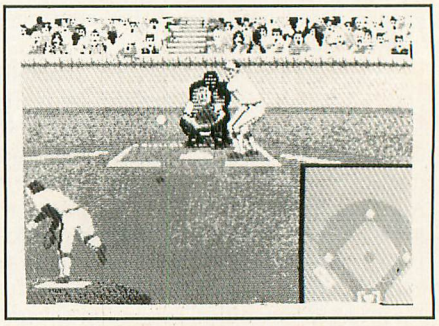
There are twenty levels of play, including an infinite level, where the program will continue to search for the perfect move until you stop it. Its features include: game save to disk; board setup; game progress output to the printer; hints to teach the novice; and complete control over the screen colors. Rounding out the package is a disk containing 100 classic games to replay and analyze. Before you join a chess club at school, let the Chessmaster whip you into shape.

A series of educational titles are available from Thunder Mountain, including several programs specifically designed for youngsters ages 4-8. With the help of Mercer Mayer's Tink Tonks™, your children can learn a variety of skills with programs such as **ABC's, Spelling, Develop Thinking Skills** (designed to help children develop reasoning skills) and **Being a Smart Thinker** (which helps develop concentration and memory skills).

For kids ages 5 and above, **Songwriter** by Learningways, Inc. composes music easily, with playback through the computer or stereo. It features a unique, easy to use "piano roll" graphics display of the music, with on-screen commands. All these titles are an affordable $10.

No sweat sports.

Around this time of year, sports buffs usually start getting a bit restless. A good sports simulation may be just the ticket to tide them over until spring.



Hardball!

The reviews of **Hardball!** from Accolade raved about superb graphics and realistic portrayals of pitcher and batter, then gave it a high ranking in the playability department, too. It's not a simulation of real baseball players like the "Wizard," Ozzie Smith, but one of forty-eight distinctly different, well defined computerized models. Each has a unique playing style, with differing strong and weak points. You'll soon develop an affection for your favorites. This is a good baseball simulation that will tide you over until next spring.

Accolade is a sports simulation specialist. Their $30 **Fight Night** pits you against any of six different boxers, or one of twenty-four possible customized pugilists you "construct" yourself. This is a good

implementation of a popular arcade game from a few years back, with the added bonus of custom "bodybuilding" your opponents (an electronic Dr. Frankenstein, if you will). The joystick response is not impressive, but if it's boxing you want, this is the best offering for the 8-bit.

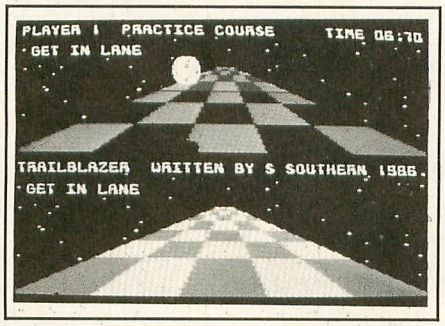
Epyx is also well known for its sports simulations. Their most recent release for the 8-bit is **World Championship Karate** for $35. In it, you participate in karate matches versus human or computer controlled opponents, against the backdrop of eight international settings. The graphics are great; the sound effects and musical score, superb. The motions of the players are smooth and fluid. Their figures are well defined, with some excellent expressions on the faces of the opponents as they battle it out. There are sixteen possible moves, based on the eight joystick positions with the fire button either up or down. It takes a little while to learn, but the controls are logical, and the joystick response and overall game playability rank as tops.

In **Racing Destruction Set** ($15) from Electronic Arts, you choose a vehicle, then pick the tires and engine size to match the road conditions. Drive on one of fifty built-in tracks, or design your own, with straightaways, curves, forks, crossovers and jumps. You can also select one of three different surface types: pavement, dirt or ice. With the split screen, you can race against an opponent or the computer. Add some land mines and oil slicks, and your rival will be spinning out and crashing, agent 007 style.

Star Raiders II ($20) from Atari portends to be the *boffo* sequel to the original smash hit Star Raiders, which helped sell thousands of Ataris all by itself. Actually, Star Raiders II was originally called The Last Starfighter, a space war, shoot-'em-up tribute to the movie of the same name. Atari bought it, spruced it up a bit and changed the name.

It's still a good arcade-style wrist jerker which will provide hours of challenging play time. In this renewed battle against the evil Zylons, you must warp between two solar systems. In one, you must protect the Atarian Federation cities from attack by the Zylon Master Force. In the other, you must destroy the Zylon Attack Bases by dropping Star Bursts. If you fail to destroy the Zylon Attack Bases, your foe will continue building new Zylon squadrons to attack your cities. You alone command the only ship capable of destroying all the Zylon forces.

TrailBlazer from Mindscape is a superb 3D race game—a sort of cross between Rainbow Walker and Ballblazer. The speed is terrific as you blast through TrailBlazer. Your opponent is left in a cloud of dust as you effortlessly leap over black holes, purple walls, blue bouncers and even the terrible "nasty zone." The game features high-speed animation, graphics



TrailBlazer.

and sound that push your computer to the limit. Each player gets a split screen view of the playfields. You can play any of the twenty fixed courses or the killer random option. It supports several play modes, including one- and two-player arcade, race and trial, plus an impossible computer opponent.

Wizards, princesses and magic.

Wizard's Crown ($40) from SSI is a fantasy role-playing game in the style of Dungeons and Dragons™. Your characters may be selected from eight standards, or custom built. Each character has life, dexterity, intelligence and strength attributes. You accumulate experience points in your quest. This game features good graphics and user-selectable battles. Its biggest drawback is the distracting disk swapping during game play.

Nearly all Infocom text adventures are available for the 8-bit Ataris. Recently, the price of **Wishbringer** was dropped to $15. An intermediate-level game, Wishbringer provides a good introduction to the Infocom works. Users will get a chance to heal the damaged tendons in their joystick wrists, while putting a lot more time into an Infocom game than into the average shoot-'em-up.

The Infocom programs are not merely games, but stories that you become a part of. The best thing about these stories is that *you* help develop the plot and determine the ending. Other recent releases from Infocom include **Stationfall**, a sequel to the hit Planetfall, and **The Lurking Horror**. These Infocom games are in the $25 to $40 price range.

Once you're hooked on these games, it's quite easy to get stumped, especially if you're a newcomer to the world of interactive fiction. Infocom's ***Invisiclues***™ make great gifts. Each booklet contains enough hints to help solve the game completely. Each clue may be revealed selectively with a special marker. *Invisiclues* are available for all Infocom games, and are priced at about $7 each. They can generally be found in B. Dalton and Waldenbook stores that sell software.

A good entry-level Infocom game is **Moonmist**, an interactive whodunit. Since it's relatively simple to solve, there are four different solutions to this mystery, to help maximize play time. This game is unique in that the *suspects* you must deal with in this murder mystery will actually give you hints when needed.

Ghostbusters ($15) from Activision is the video game that lets you star in a homegrown version of this whacky battle with poltergeists run amok. With limited funds, you cut and paste your plan of attack. Select one of four ectomobiles (cars) and equip it with a variety of ghostbusting gear. Then drive around town and catch ghosts until the Marshmallow Man appears. You must try to get past him to enter the Temple of Zuul. If you succeed, you'll win more credits—to build a better ectomobile the next time you play.

Firebird, of **The Pawn** fame, is well known for their superb graphics and near-genius command parsers. The Pawn, now available for the 8-bit ($40), has a virtually identical text scenario to that of the ST. The graphics are very well done. Rather than *smashing* all the pictures to fit, you can scroll the display over the larger views. This method retains much of the original work's superb graphics, with the minor trade-off of having to scroll the dis-

The Pawn.

play. Two more text and graphic adventures of lesser sophistication are available from Firebird: **Jewels of Darkness** ($25),

a rendition of the original classic simply called Adventure; and **Silicon Dreams** ($25), an adventure of robots, intergalactic travel and a lost city of Eden.

Do you have a *MAD* magazine fan in the house? If so, the **Spy Vs. Spy** video game series may be an ideal gift. In this parody of the Spy cartoons in *MAD*, you take control of one of the characters and play an interactive game of high adventure and intrigue through "simulvision." Each player has a window, a simulated video screen. Off to the side is a score and status window. Anytime one of the spies enters the other's view, vision is switched to that window, where a battle ensues.

Your players search rooms in the Embassy Espionage mission, to swipe a top secret briefcase and catch a getaway jet; or do some daring beach combing in the Island Caper mission, to assemble a missile and escape on the rescue submarine.

Both games come in one package called **Spy vs. Spy Volume I & II** ($15) from Accolade Inc., distributed under the Avantage Software label. Playing against the computer makes for short games, but Spy vs. Spy is well suited to two-player encounters.

Matt would like to thank Jeff Randall of Randall's Home Computers in St. Louis for his invaluable assistance in preparing this article.

Companies Mentioned within the Shopping Guide.

Accolade
20813 Stevens Creek Blvd.
Cupertino, CA 95014
(408) 446-5757

Activision
P.O. Box 7286
Mountain View, CA 94039

Atari Corp.
1196 Borregas Avenue
Sunnyvale, CA 94086

Broderbund Software, Inc.
17 Paul Drive
San Rafel, CA 94903
(415) 479-1170

Clearstar Softechnologies
1501 Wood Avenue, #36
Sumner, WA 98390
(206) 863-8523

COMPUTE! Books
P.O. Box 5406
Greensboro, NC 27403

COVOX Inc.
675-D Conger Street
Eugene, OR 97402
(503) 342-1271

Electronic Arts
2755 Campus Drive
San Mateo, CA 94403

Elfin Magic
23 Brook Place
E. Islip, NY 11730

Epyx
1043 Kiel Court
Sunnyvale, CA 94089

Firebird Inc.
P.O. Box 49
Ramsey, NJ 07446

High Tech Expressions, Inc.
2699 South Bayshore Dr.
Suite 1000A
Coconut Grove, FL 33133

ICD, Inc.
1220 Rock Street
Rockford, IL 61101-1437
(815) 968-2228

Infocom
125 CambridgePark Drive
Cambridge, MA 02140

Kyan Software, Inc.
1850 Union Street #183
San Francisco, CA 94123

MASTERTRONIC
7311B Grove Rd.
Frederick, MD 21701

MicroProse
120 Lakefront Drive
Hunt Valley, MD 21030

OSS
1221B Kentwood Avenue
San Jose, CA 95129
(408) 446-3099

Phoenix International Company
1090 South 350 East
Provo, Utah 84601

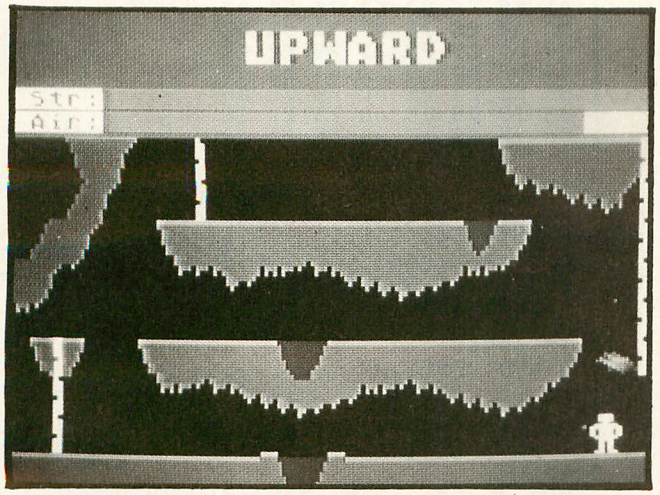
Software Country
9713 Santamonica Blvd.,
Suite 204
Beverly Hills, CA 90210

SSI
1046 N. Rengstorff Avenue
Mountian View, CA 94043

Suncom
260 Holbrook Dr.
Wheeling, IL 60090

The Catalog
524 Second St.
San Francisco, CA 94107
(800) 443-0100 ext. 133

Thunder Mountain
P.O. Box 1167
Northbrook, IL 60065-1167



Upward

Climb for your life in this Action! game.

by Greg Knauss

They left you! They actually *left* you! Sure, you had wandered off without telling anybody. Sure, the main transport was running short of air. Sure, nobody really likes you anyway. But still. . .

"Well," you think, glancing up into the dimly filtered and dirty light, "gotta get out somehow. What to do, what to do. . ."

For no very good reason, you think back fondly on your carefree and happy adventuring days, the years you spent exploring, discovering and fighting—all for fun and profit. You were brave, you were valiant. You were the type of guy who, on a bet, would walk into a Marine recruiting office and start signing papers without reading them. You would eat a plateful of beans and then attend an opera. Yes, you bordered on foolhardy. And then you signed up with the government. No, not the Marines; that test with the pegs is a lot harder than it looks. . .You joined the Solar Energy Commission, Subterranean Division.

And now you're trapped at the bottom of an unpleasantly deep cavern with only a full tank of oxygen, an ugly brute of a bat and the ever-present poison-tipped darts.

"Oh dear," blue-pencils in the editor after seeing what you would probably say in this situation. "Well, I might as well get started."

You plug your headphones into the government issue Ultimate Heavy-Duty Super-Compact Tape Player Complete with Ashtray and drop in a tape. Gunk music, just your luck. . .

As you start your long, long ascent, a thought flitters across your mind: "I hope the ropes hold."

Instructions.

To start this finger-mangling business of typing in **Upward**, first, select a disk with only DOS and DUP on it. Get the disk ready to store the thing on, and slam (or, for those of lesser enthusiasm, put) the Action! cartridge into your computer.

Now start typing Listing 1. A bit of advice: the numbers are easier to enter if you have a (soon to be ex-) friend read them to you.

Done? Okay, save the file (as UPWARD) through the editor before compiling! Also, make sure you check your typing with "D:CHECK in Action!" (issue 44). This is because, if you made a mistake, you don't want the whole program eaten. Half the game is machine language, and, if you mistyped something, the computer will acquire a bad case of schizophrenia. So save the silly thing!

Now, remove Action!, put in BASIC and type in Listing 2, the CREATOR, checking your typing with "BASIC Editor II" (issue 47). This program is only used once, since it creates a file on disk which **Upward** reads in as a set.

Now, after saving and running (in that order) CREATOR, reinsert the Action! cartridge, go to the monitor, and type *R"UPWARD"*. (You need to run it straight from the monitor, because the source and object code won't fit into the computer at once.)

The rules.

When the game is run, you'll see, low on the screen, the game title and a little man running around. Chasing our hero is an irritating bat named Skyler. Skyler's speed can be changed between its three settings by hitting the SELECT key. Choose one of his three accuracy levels with OPTION. Press START or tap on the joystick button to begin play.

Upward *continued*

You start the game at the bottom of a nauseatingly deep cavern, and you must reach the top while both your life and air hold out.

For you masochists, there are several ways to die:

(1) Lava has the predictable effect on your guy—anybody who's lost a marshmallow while toasting it knows what that's like.

(2) Skyler the bat takes huge joy in chomping large bites out of you, thus reducing your strength and killing you slowly. To indicate that Skyler's hurting you, your character will turn a bright pink, partly from anger, partly from rage—but, mostly, from embarrassment.

(3) Darts. Your good friend Illinois Smith told you there were lots of these sorts of things in big caves. They're here, *en masse*. As with Skyler, these don't kill you instantly, but reduce your strength.

And, of course, there's one way to lose air: breathe. Since you're assumed to be human (be it a correct assumption or not), you do this automatically.

When the game is in progress, there are two gray bars across the top of the screen. The top one, labeled *Str:*, is how much strength you have left. The second one, *Air:*, is the oxygen left in your tanks. Once either reaches zero, it's bye-bye!

Life, tenuous as it is, is apportioned to you at the rate of only one per game. Now, since one life in a cave where death can come at any instant is very aggravating (not to mention annoying and frustrating), I would advise you to look before you leap.

Move the joystick to have your man, er, person (sorry) move about. As might be predicted, you must be by a rope to move up and down.

Jumping is accomplished by pressing the button. And, yes, you can jump off of and onto ropes.

The SPACE BAR toggles the pause feature.

When the game ends, the obvious message will be displayed, along with your score and, if appropriate, an asterisk beside it to proclaim a high score. Hit the trigger to return to the title page.

And now, **Upward!**

Greg Knauss is: politically, liberal; physically, not too hot; scholastically, fairly bored; psychologically, very confused; cosmically, insignificant; and emotionally, quite happy, thank you. On a whim, he has been know to program an entire game just to write a new biography.

The two-letter checksum code preceding the line numbers in Listing 2 is *not* a part of the BASIC program. For further information, see the "BASIC Editor II," in issue 47.

Listing 1.
Action! listing.

```
; UPWARD by Greg Knauss

;       CHECKSUM DATA
;[6A 3D EB A8 39 AC 1E D9
; FF 74 FC 62 C0 0B 27 6C
; 92 E6 FF 87 E9 7E E6 9F
; AA A6 E8 78 A7 9A 95 62
; 80 D9 29 79 79 70 EF 7C
; C0 85 AA CD 1A B4 EF EB
; B8 E1 E9 F2 DD FB 7B D0
; 45 29 D6 58 AB 75 E8 7E
; 8F 16 A5 2E D9 6E 31 61 ]

BYTE
  P,PLY=1780,PIC,PL=1784,PD=1772,
  DRW=[1],SKX,SKY=1782,SPIC=[1],
  SSP=[3],SAC=[80],SCT=[0],DAX,
  DAY=1783,DART,Y,S,I,L,PM,PC=53253,
  PPC=53260,SLO,SHI,LO,HI,N,V,ES,STR,
  SN,LNS=[0]

BYTE ARRAY
  ST="D:UPSET.A"

INT
  X1,Y1,K,SX,SY,DX,DY

CARD
  DRB,DL,CH,SC=[0],HSC=[0],AIR,J


PROC DLIST()
  [112 112 112  71   0   0   6   2
     0   2 128   0 101   0   0  37
    37  37  37  37  37  37  37  37
     5  65   0   0]
RETURN


PROC DLI()
  [ 72 169   0 141  10 212 141  22
   208 169   0 141  23 208 169   0
   141  24 208 169   0 141  25 208
   169   0 141  10 212 169   0 141
    26 208 169   0 141   9 212 104
    64]
RETURN


PROC PLRVBI()
  [162   3 189 244   6 240  89  56
   221 240   6 240  83 141 254   6
   106 141 255   6 142 253   6  24
   169   0 109 253   6  24 109 252
     6 133 204 133 206 189 240   6
   133 203 173 254   6 133 205 189
   248   6 170 232  46 255   6 144
    16 168 177 203 145 205 169   0
   145 203 136 202 208 244  76  87
     6 160   0 177 203 145 205 169
     0 145 203 200 202 208 244 174
   253   6 173 254   6 157 240   6
   189 236   6 240  48 133 203  24
   138 141 253   6 109 235   6 133
   204  24 173 253   6 109 252   6
   133 206 189 240   6 133 205 189
   248   6 170 160   0 177 203 145
   205 200 202 208 248 174 253   6
   169   0 157 236   6 202  48   3
    76   2   6  76  98 228   0   0]
RETURN


PROC CHSTORE()
  [170 170 170 170 170 170 170 170
    85 170 170 170 170 170 170 170
     0  85 170 170 170 170 170 170
     0   0  85 170 170 170 170 170
     0   0   0  85 170 170 170 170
     0   0   0   0  85 170 170 170
     0   0   0   0   0  85 170 170
     0   0   0   0   0   0  85 170
     0   0   0   0   0   0   0  85
```

```
    168  32   0   0   0   0   0   0
    170  42  32   0   0   0   0   0
    170  42  34   0   0   0   0   0
    170 170 170 168  32   0   0   0
    170 170 170 170 168  32   0   0
    170 170 170 170 170 168  32   0
    170 170 170 170 170 170  40  32
      0   0   0   0   0   0   0   0
     85 170 170 170 170 170 170  32
     85 170  42  42  42  10  10   0
     85 170 170 168 168 160 128 128
      0   0  64 128 144 164 168 169
      0   0   0   0   0   0   0   0
      0   0   0   0   0   0   0   0
      1   2   2   6  10  26  42 106
    170  42  42  10  10   2   2   2
    170 168 160 128 128 128   0   0
    125 190 190 190 174 190 190 190
     48  60  60  60  12  60  60  60
    186 190 190 190 174 190 190 190
    255 255 255 255 255 255 255 255
    255 191 191 191 175 175 171 170
    255 254 254 250 250 234 170 170
    170 171 175 175 191 191 255 255
    170 234 234 250 250 250 254 254]
RETURN

PROC GPIX()
  [ 56  56  56  16 124 254 186 186
   186 124  56  40  40  40  40 108
    56 186 186 146 254 124  56  56
    56  56  56  40  40  40  40 108
   184 184 186 210 122  62  60  56
    56  60  60  44  40  44  32  96
    56 186 186 146 254 124  56  56
    56  56  56  40  40  40  40 108
    58  58 186 150 188 248 120  56
    56 120 120 104  40 104   8  12
    28  24  28   8  28  60  60  62
    60  30  31  17  50  34  32  48
    28  24  28   8  60 125  95  94
    28  28  28  54  98  66  67   0
    28  24  28   8  28  60  60  60
    62  28 108  60  20   4   4   6
    28  24  28   8  28  28  28  28
    28  28  30  30  28   8   8  12
    56  24  56  16  56  60  60 124
    60 120 248 136  76  68   4  12
    56  24  56  16  60 190 250 122
    56  56  56 108  70  66 194   0
    56  24  56  16  56  60  60  60
   120  56  54  60  24  16  16  48
    56  24  56  16  56  56  56  56
    56  56 120 120  56   8   8  24
    56  58  58  18 126 252 184 184
   184 120  56  40  40  40  40 108]
RETURN

PROC SPIX()
  [  0   0   0   0   0   0  40 186
   214 124  56  16  40  56 214 124
    56  16  40  56  84 254 186  16
    40  56  84 254  56  16]
RETURN

PROC MUS()
  [  0   0   0   0 150   0 100 100
     0   0  75   0 100 100   0  55
    75 100   0 100  75  55   0   0]

PROC MUS1()
  [250   0   0 250   0   0 250   0
     0 250   0   0 250   0   0 250
     0   0 250   0   0 250   0   0]
```

```
PROC SETUP()
  Graphics(0)
  Poke(82,0)
  Poke(752,1)
  PrintE(" ")
  CH=(Peek(106)-40)*256
  MoveBlock(CH+80,CHSTORE,272)
  Zero(CH,8)
  DL=PeekC(560)
  SHI=Peek(106)-36
  SLO=0
  MoveBlock(DL,DLIST,29)
  J=PeekC(88)

  PokeC(DL+4,J)
  PokeC(DL+13,SHI*256)
  PokeC(DL+26,DL)
  Zero(704,9)
  Poke(DLI+2,0)
  Poke(DLI+10,0)
  Poke(DLI+15,0)
  Poke(DLI+20,0)
  Poke(DLI+25,0)
  Poke(DLI+35,CH/256)
  PokeC(512,DLI)
  Poke(54286,192)
  MoveBlock(1536,PLRVBI,160)
  PM=Peek(106)-16
  DRB=PM*256+1
  Zero(DRB+1024,1024)
  Poke(623,4)
  Poke(559,62)
  Poke(1788,PM+4)
  Poke(53277,3)
  Poke(54279,PM)
  Poke(1771,PM)
  PL=16
  Poke(1785,3)
  PD=1
  Poke(1773,1)
  SKY=6
  SKX=0
  Poke(1786,6)
  Poke(1774,0)
  Poke(1787,1)
  Poke(1775,1)
  MoveBlock(DRB,GPIX,240)
  MoveBlock(DRB+511,SPIX,30)
  FOR J=DRB TO DRB+2 DO
    Poke(J+256,56)
    Poke(J+512+255,60)
  OD
  POKEC(548,PLRVBI)
RETURN

PROC OXY()
  Color=32
  Plot(38,2)
  DrawTo(AIR/200+5,2)
RETURN

PROC LIFE()
  Color=32
  Plot(38,1)
  DrawTo(STR/6-2,1)
RETURN

PROC PLAY()
  V==+2
  IF V>15 THEN
    N==+1 V=0
    Sound(0,0,0,0)
    Sound(1,0,0,0)
  FI
  IF N=24 THEN
    N=0
```

```
  FI
  J=Peek(MUS+N)
  IF J>0 THEN
    Sound(0,J,12,15-V)
  FI
  J=Peek(MUS1+N)
  IF J>0 THEN
    Sound(1,J,10,15-V)
  FI
  RETURN


PROC MOVE()
  PLY==+Y1
  P==+X1
  Poke(53248,P)
  Poke(53249,P)
  Poke(1781,PLY+15)
  Poke(704,15)
  IF PPC=6 THEN
    STR==-1
    Poke(704,70)
    LIFE()
  FI
  IF PPC=10 OR PPC=8 THEN
    STR==-25
    Poke(53251,0)
    DART=0
    LIFE()
  FI
RETURN


PROC SRTDRT()
  DART=0
  Poke(53251,0)
  DAX=Rand(50)+100
  DAY=0
RETURN


PROC BANG()
  DAX=DAX+DX
  Poke(53251,DAX)
  IF DAX=2 OR DAX=252 THEN
    DART=0
    SRTDRT()
  FI
RETURN


PROC SKYLER()
  IF SCT=SSP THEN
    SCT=0
    IF SKX MOD SAC=0 THEN
      SX=0
      SY=0
      IF SKX<P THEN
        SX=1
      FI
      IF SKX>P THEN
        SX=-1
      FI
      IF SKY<PLY THEN
        SY=1
      FI
      IF SKY>PLY THEN
        SY=-1
      FI
    FI
    SPIC==+1
    IF SPIC=5 THEN
      SPIC=1
    FI
    Poke(1774,SPIC*6)
    SKY==+SY
    SKX==+SX
    Poke(53250,SKX)
  FI
  SCT==+1
RETURN


PROC WAY()
  X1=0
  Y1=0
  IF S=14 THEN
    Y1=-1
    LO=1
    HI=4
  FI
  IF S=13 THEN
    Y1=1
    LO=1
    HI=4
  FI
  IF S=7 THEN
    X1=1
    LO=5
    HI=8
  FI
  IF S=11 THEN
    X1=-1
    LO=9
    HI=12
  FI
  IF X1<>0 THEN
    K=X1
  FI
RETURN


PROC LEGAL()
  IF PC=0 THEN
    Y1=1
    X1=0
  FI
  IF PC=1 OR PC=9 THEN
    Y1=0
  FI
  IF PC=2 THEN
    P==-K
    Y1=1
    X1=0
  FI
  IF PC=3 THEN
    Y1=-1
  FI
  IF PC=4 OR PC=6 THEN
    X1=0
  FI
  IF P<48 THEN
    P=48
  FI
  IF P>202 THEN
    P=202
  FI
RETURN


PROC ANIMATE()
  IF S=15 THEN
    IF PC=1 OR PC=5 OR PC=9 THEN
      PD=DRW
    FI
    IF PC=0 THEN
      PD=DRW+16
    FI
  FI
  IF PC=1 OR PC=9 THEN
    IF S=13 OR S=14 THEN
      PD=DRW+208
      S=15
    FI
  FI
  IF (S=7 OR S=11) AND
```

```
      (PC=4 OR PC=6) THEN
      S=15
    FI
    IF S<>15 THEN
      IF (X1<>0 AND P MOD 2=0) OR
         (PC<>0 AND Y1<>0 AND
          PLY MOD 3=0) OR
         (PC=0 AND (S=7 OR S=11)) THEN
        PIC==+1
      FI
      IF PIC<LO OR PIC>HI THEN
        PIC=LO
      FI
      PD=DRW+PIC*16
    FI
  RETURN


  PROC JUMP()
    IF X1=1 THEN
      PD=DRW+96
    FI
    IF X1=-1 THEN
      PD=DRW+160
    FI
    FOR I=1 TO 40 DO
      IF I>35 THEN
        Y1=2
      FI
      IF I<19 THEN
        Y1=1
      FI
      IF I<11 THEN
        Y1=0
      FI
      IF I<8 THEN
        Y1=-1
      FI
      IF I<4 THEN
        Y1=-2
      FI
      Poke(53278,0)
      FOR J=1 TO 750 DO OD
      IF (PC=4 OR PC=6) AND I>10 THEN
        PD=DRW+32
      FI
      IF PC>0 AND PC<>2 AND I>10 THEN
        RETURN
      FI
      IF PLY=189 OR STR<54 THEN
        RETURN
      FI
      IF P<48 OR P>202 THEN
        X1=-X1
        K=X1
      FI
      IF PC=2 THEN
        X1=-X1
      FI
      MOVE()
      SKYLER()
      PLAY()
      IF DART=1 THEN
        BANG()
      FI
    OD
  RETURN


  PROC DN()
    SndRst()
    FOR S=1 TO 6 DO
      FOR I=0 TO 15 DO
        Poke(54277,I)
        PLY==-1
        SKY==-1
        DAY==-1
        Poke(1781,PLY+15)
        FOR J=1 TO 100 DO OD
      OD
      Poke(54277,0)
      SLO==+40
      IF SLO<40 THEN
        SHI==+1
      FI
      Poke(DL+13,SLO)
      Poke(DL+14,SHI)
    OD
    SRTDRT()
    SC==-10
  RETURN


  PROC UP()
    SndRst()
    FOR S=1 TO 6 DO
      Poke(54277,15)
      SLO==-40
      IF SLO>215 THEN
        SHI==-1
      FI
      Poke(DL+13,SLO)
      Poke(DL+14,SHI)
      I=15
      FOR Y=1 TO 16 DO
        Poke(54277,I)
        I==-1
        PLY==+1
        SKY==+1
        DAY==+1
        Poke(1781,PLY+15)
        FOR J=1 TO 100 DO OD
      OD
    OD
    SRTDRT()
    SC==+10
  RETURN


  PROC FADEIN()
    FOR I=0 TO 15 DO
      Poke(704,I)
      Poke(707,I)
      Poke(710,I)
      IF I<5 THEN
        Poke(DLI+20,64+I)
      FI
      IF I<5 THEN
        Poke(712,144+I)
      FI
      IF I<7 THEN
        Poke(DLI+10,48+I)
        Poke(709,I)
      FI
      IF I<9 THEN
        Poke(DLI+2,48+I)
        Poke(706,I)
      FI
      IF I<11 THEN
        Poke(DLI+15,I)
      FI
      IF I<13 THEN
        Poke(708,32+I)
      FI
      FOR J=1 TO 2000 DO OD
      Sound(0,200,8,15-I)
    OD
  RETURN

  PROC PLRGO()
    FOR I=0 TO 3 DO
      Poke(704+I,0)
      Poke(53248+I,0)
    OD
    DN()
```

(continued on page 47)

MOVING?

DON'T MISS A SINGLE ISSUE

Let us know your new address right away. Attach an old mailing label in the space provided below and print your new address where indicated.

QUESTION ABOUT YOUR SUBSCRIPTION?

Check the appropriate boxes below:

- ☐ New subscription. Please allow 4 to 8 weeks for your first copy to be mailed.
- ☐ Renewal subscription. Please include a current address label to insure prompt and proper extension.
- ☐ 1 year — $28.00. This rate limited to the U.S. and its possessions.
- ☐ Payment enclosed.
- ☐ Bill me.

MAIL TO: **ANALOG COMPUTING**
P.O. BOX 16927 N. HOLLYWOOD, CA 91615

Name _____

Street Address _____

City _____ State ____ Zip _____

ATTACH LABEL HERE

(IF LABEL IS NOT HANDY, PRINT OLD ADDRESS IN THIS SPACE.)

RAM-INIT

RAMdisk initialization without MEM.SAV.

by Mack McLeod

Do you ever wish you had more RAMdisk space? If so, read on. The DOS 2.5 RAMdisk for the 130XE is initialized by RAMDISK.COM, placing DUP.SYS and MEM.SAV on D8:, and taking up 87 of the precious 499 sectors available. Some applications, though, do not need DUP and MEM. Autorun programs, or those loaded from the DOS menu, may not require memory to be saved. Also, there are several DOS modifications around that place these files under the OS ROM, making the D8: copies useless. The alternative is to manually delete the files after each boot.

For applications that need more RAMdisk space, **RAMINIT** is the answer. This program will properly initialize an empty RAMdisk. A basic program is used to create RAMINIT.COM. Delete or rename RAMDISK.COM, then rename RAMINIT.COM to install the function. If you write in assembly, the *RAMINIT* message may be eliminated by deleting Lines 240 to 410, 1140 to 1250 and 1280. Change the JSR in Line 1130 to a JMP.

The DOS 2.5 address of the RAMdisk initialization filename is *$14F1*. This contains D:RAMDISK.COM, followed by a *$9B*. The autorun filename address is *$170C*. It contains D1:AUTORUN.SYS, followed by a *$9B*. You may POKE any device prefixed name desired in these locations, as long as the lengths are not exceeded. You will need to use the SAVE DOS FILES option to make the change permanent.

Mack McLeod is an Analyst/Programmer for a major Canadian bank. Even though he works with the largest of mainframes, he's amazed at the power and flexibility of Atari 8-bit machines. He believes they truly provide "Power without the Price." He programs his 130XE primarily in assembly, occasionally in BASIC, and now in C.

The two-letter checksum code preceding the line numbers here is *not* a part of the BASIC program. For further information, see the "BASIC Editor II," in issue 47.

Listing 1.
BASIC listing.

```
OY 100 OPEN #1,8,0,"D:RAMINIT.COM"
ZQ 200 FOR I=1 TO 220
OR 300 READ A:PUT #1,A:NEXT I
NH 400 CLOSE #1:? "RAMINIT.COM CREATED"
SV 900 DATA 255,255,0,6,207,6,169,0,133,1
   9,133,20,169,14,133,85,169,8,133,84,14
   1,72,3,169,0
MS 901 DATA 141,73,3,170,169,200,141,68,3
   ,169,6,141,69,3,169,11,141,66,3,32,86,
   228,173,1,211
TH 902 DATA 133,206,169,235,141,1,211,169
   ,0,133,203,169,116,133,204,160,0,185,1
   95,6,145,203,200,192,5
CJ 903 DATA 208,246,169,0,145,203,200,192
   ,10,208,249,169,15,145,203,200,169,255
   ,145,203,200,192,55,208,249
XI 904 DATA 169,0,145,203,200,169,127,145
```

(continued on page 58)

COMPUTER GAMES +

(714) 639-8189

VIP PROFESSIONAL
$99.95 PRO
$119.95 GEM

520ST COMPLETE SYSTEM
Monitor, disk drive, computer
COLOR $679
B & W $549

COLOR MONITOR	299.95
MONO	169.95
SINGLE SIDED DRIVE	129.95
DOUBLE SIDED DRIVE	199.95
20 MEG HARD DISK	529.95
COMPUTER & DRIVE	399.95

ASTRA'S RAM EASY
$109.95
1 Meg ST upgrade

DB MAN
$89.95
(Best ST Data Base)

NEW!
XE DISK DRIVE XF551
$179.95

GUARANTEED LOWEST PRICES — CALL FOR TITLES NOT LISTED

RECENT AND SOON TO BE RELEASED ST PROGRAMS

BREACH	27.95
BARBARIAN	27.95
EMPIRE	37.95
DEFENDER OF THE CROWN	32.95
MIDI MAZE	27.95
VIDEO VEGAS	24.95
PAPER BOY	27.95
SENTRY	29.95
FAERY TALE ADVENTURE	34.95
ALIEN FIRES	27.95
JUPITER PROBE	19.95
F-15 STRIKE EAGLE	27.95
WIZARD'S CROWN	24.95
RINGS OF ZIFLIN	24.95

REMARKABLE EUROPEAN ST GAMES

GAUNTLET	27.95
ROAD RUNNER	27.95
ARKANOID	24.95
XEVIOUS	29.95
METROCROSS	29.95
MOUSE TRAP	24.95
INDIANA JONES	29.95
SOLOMON'S KEY	29.95
EAGLE'S NEST	27.95
PROHIBITION	24.95
EXTENSOR	24.95
HADES NEBULA	24.95
LIBERATOR	22.95
TYPHOON	24.95
ALTAIR	24.95
CRAFTON & XUNK	24.95
TRACKER	29.95
SILICON DREAMS	24.95
JEWELS OF DARKNESS	24.95
KARATE MASTER	22.95
PASSENGERS IN THE WIND	29.95
KNIGHT ORC	29.95
ANNULS OF ROME	29.95

ADVENTURES

S.D.I.	32.95
ALTERNATE REALITY-CITY	24.95
ULTIMA II	38.95
ULTIMA III	38.95
ULTIMA IV	38.95
PHANTASIE	24.95
PHANTASIE II	24.95
PHANTASIE III	24.95
MERCENARY	26.95
MERCENARY - 2nd CITY	14.95
THE PAWN	29.95
BORROWED TIME	29.95
MINDSHADOW	29.95
SPIDERMAN	9.95
UNIVERSE II	34.95
SUNDOG	24.95
GUILD OF THIEVES	29.95
BRATACUS	32.95

ARCADE / ACTION

WAR ZONE & FIRE BLASTER	27.95
SPACE STATION & PROTECTOR	27.95
WANDERER (3D!)	25.95
STAR GLIDER	29.95
DEEP SPACE	19.95
SUPER CYCLE	14.95
MAJOR MOTION	24.95
TIME BANDIT	24.95
STWARS	24.95
PLUTOS	19.95
Q BALL	19.95
TRAIL BLAZER	27.95
AIR BALL	24.95

STRATEGY / SIMULATION

HEX	29.95
BALANCE OF POWER	32.95
COLONIAL CONQUEST	24.95
ROADWAR 2000	24.95
AUTODUEL	32.95
SKYFOX	14.95
ARCTIC FOX	27.95
FLIGHT SIMULATOR II	32.95
SUPER HUEY	25.95
GATO	32.95
SILENT SERVICE	24.95
SUB BATTLE	24.95
HACKER	17.95
HACKER II	24.95
SHANGHAI	24.95
CHESSMASTER 2000	29.95
BRIDGE 5.0	22.95

SPORTS

GRIDIRON FOOTBALL	32.95
MICROLEAGUE II - BASEBALL	39.95
CHAMP. BASEBALL	29.95
CHAMP. WRESTLING	29.95
CHAMP. BASKETBALL	24.95
ST KARATE	19.95
WORLD CHAMP. KARATE	29.95
KARATE KID II	24.95
WORLD GAMES	24.95
WINTER GAMES	24.95
LEADER BOARD	24.95
TOURNAMENT DISK	14.95
10th FRAME BOWLING	24.95

PRODUCTIVITY

FLASH BAK / FLASH CACHE	49.95
PUBLISHING PARTNER	94.95
EASY DRAW	64.95
GRAPHIC ARTIST	124.95
AEGIS ANIMATOR	48.95
ST TALK 2.0	19.95
PERSONAL PASCAL	49.95
GFA BASIC	49.95
GFA COMPILER	49.95
TYPING TUTOR	24.95
WORD WRITER ST	44.95
DATA MANAGER ST	44.95
SWIFT CALC ST	44.95
REGENT WORD II	48.95
REGENT BASE 1.1	79.95
DEGAS ELITE	39.95
N-VISION	29.95
PAINTWORKS	24.95
MUSIC STUDIO	32.95
PRINT MASTER PLUS	24.95
ART GALLERY I	19.95
ART GALLERY II	19.95
DAC EASY ACCOUNTING	44.95
DAC EASY PAYROLL	32.95
MAGIC SAC	99.95
PC DITTO	74.95
DESK CART	69.95

XE / XL SOFTWARE

GAUNTLET	24.95
DEEPER DUNGEONS	14.95
ARKANOID	24.95
HEAD OVER HEELS	22.95
DRUID	22.95
LIVING DAYLIGHTS	24.95
ALT REALITY-CITY	24.95
ALT REALITY-DUNGEON	24.95
UNIVERSE	34.95
BOUNTY BOB STRIKES BACK	29.95
RETURN OF HERACLES	9.95
ULTIMA I	39.95
ULTIMA III	19.95
ULTIMA IV	39.95
THE PAWN	27.95
WOMBATS I	29.95
INFILTRATOR	22.95
BOP & WRESTLE	22.95
ACE OF ACES	22.95
HARDBALL	22.95
FIGHT NIGHT	22.95
LEADER BOARD	27.95
PHANTASIE	27.95
FLIGHT SIMULATOR II	37.95
AUTO DUEL	37.95
PARTY QUIZ	14.95
BASIC XL	36.95
BASIC XE	46.95
ACTION!	46.95
MAC 65	46.95
TOOL KITS	18.95
WRITER'S TOOL	38.95
ATARI WRITER PLUS	39.95
TYPING TUTOR	17.95
SAT COMPLETE	27.95
PRINT SHOP	29.95
GRAPHICS I	14.95
GRAPHICS II	14.95
GRAPHICS III	14.95
SYNCALC or SYNFILE	32.95
VIDEO TITLE SHOP	22.95
DELUXE JOYSTICK (was $35)	9.95
ATARI JOYSTICKS	2 / 9.95

DISK DRIVES

ASTRA THE ONE (XE/XL)	249.95
ASTRA 2001 (XE/XL)	279.95
ASTRA HD + (ST 20 meg & d/s drive)	849.95
INDUS GT	179.95
INDUS GT (ST d/s)	189.95

PRINTER INTERFACES (XE / XL)

MICROPRINT	29.95
MPP-1150	39.95
PR CONNECTION	59.95

PRINTERS

STAR NX-10	169.95
STAR NB24-10	449.95

MODEMS

AVATEX 1200	89.95
AVATEX 1200HC	119.95
PRACTICAL 2400	199.95

COMPUTEREYES ST 179.95
IMAGE SCAN . . . 79.95

Prices subject to change without notice.

ATARI

714-639-8189

ORDERS ONLY PLEASE
1-800-443-8189

SHIPPING: Software - free shipping on U.S. orders over $100, otherwise $2.50 U.S., $6.50 outside U.S. Hardware - depends on weight, call for quote. Charge cards + 3%. C.O.D. orders are welcome, add $1.90 for UPS + 3%.

MasterCard VISA

COMPUTER GAMES + • Box 6144 • ORANGE CA 92667 • (714) 639-8189

CIRCLE #110 ON READER SERVICE CARD

The Wizard
PART 2

The compiler module.

by Clayton Walnum

Last time we learned how to use **The Wizard**'s editor module to create our adventure database. You'll be happy to know that the hard part—understanding the editor—is now behind us. Using the compiler, believe it or not, is a simple process, requiring little more than the ability to type the word *RUN*.

The only error messages the compiler will give you are "missing item" messages. These occur when an item referenced in a command hasn't been entered into the item list yet. Why does **The Wizard**'s compiler handle only this one type of error? Is it stupid? No! Those of you who have done a lot of programming know that the first time you try to compile source code, you're sure to get syntax errors. This is because the data the compiler is trying to process was created in a "free form" manner, using a text editor. You can type anything you like with a text editor; it doesn't care if you're meeting the requirements of the compiler or not.

The Wizard's editor takes the information you give it and automatically stores it in the proper form for the compiler, guaranteeing that no syntax errors exist. This doesn't mean your adventure will run perfectly the first time you compile it. In fact, you'll be extremely lucky if it does. Like any other compiler, this one can't tell you when you've made a "logic" error. An adventure program is complicated, and you'll need to fine-tune the game, editing and recompiling several times before everything's just the way you want it.

You'll probably have to add commands and items, change some text here and there, and generally smooth over the game before it'll be ready to spring on your friends.

Typing instructions.

There are three listings at the end of this article, only two of which are actually required to run the compiler. Listing 1 is the adventure "frame," a series of BASIC instructions that are common to every adventure game created with **The Wizard**. The lines of code found in this listing are combined with the lines the compiler creates from your adventure database. Type in Listing 1 using "BASIC Editor II" to check your work, and save the file to disk under the filename WIZFRAME.BAS.

Listing 2 is the compiler itself. You should also type this in using "BASIC Editor II" to check your work. Save the program to disk under the filename WIZCOMP.BAS.

Listing 3 is a short adventure game that was created using **The Wizard**. To play the game, you must merge it with Listing 1 (the frame). As previously stated, Listing 3 is optional; it isn't required in order to use the compiler. But, since the sample adventure demonstrates many of the things you can do with **The Wizard**, I suggest you go the extra step, and play the game.

Note that in Listing 3 (which is, by the way, the code generated by the compiler), most of the text is encrypted. This keeps people from listing the program to find clues.

Compiling an adventure.

Last time, we created a database for an adventure called "Diamond For Sale." Now that you've typed in the compiler and the adventure frame, we can compile that example.

To compile an adventure, you must first make sure that a copy of the adventure frame (Listing 1) is on the same disk as your adventure data. Then simply load the compiler and run it, making sure that your adventure data disk is in drive

1. The compiler will read the data and create a BASIC listing, printing short messages to the screen as it works.

Once the compiler has completed the construction of the listing, it will automatically merge it with the adventure frame, then save a copy of the completed game to disk. When all the work has been completed, the screen cursor will be positioned over the word *RUN*. Simply press RETURN to test play your new adventure game.

Try the compile process on the database we created in Part 1, and you'll have a ready-to-run adventure.

All files written to your disk will have the filename you chose when you first created your game; only the extensions will vary. For example, in the case of the Diamond For Sale adventure, all the associated files will be named DIAMOND.ext, where "ext" will vary depending on the file. The lines created by the compiler will be found in the file DIAMOND.LNS, and the complete game (the file DIAMOND.LNS merged with the adventure frame) will be found in the file DIAMOND.BAS. The other files for the Diamond adventure on the disk were created by the editor and are as follows:

DIAMOND.NAM	Your name
DIAMOND.ITM	Item data
DIAMOND.RMS	Room data
DIAMOND.COM	Simple commands
DIAMOND.MOV	Movement commands
DIAMOND.CHG	Item commands
DIAMOND.FAT	Fatal commands
DIAMOND.FIN	Final commands
DIAMOND.INT	Intro text
FILENAME.DAT	Adventure filename

Odds and ends.

As mentioned previously, the compiler will give you only one type of error message, an "item missing" warning. When you get one of these, the compiler will stop, and you must then reload **The Wizard**'s editor to enter the missing item into the adventure database. Be forewarned, though, that the compiler catches only *some* missing items. If, while play-testing a game, you get the message *NO SUCH ITEM* for a command (assuming, of course, you expected the command to give you a different response), you'll have found an item missing from your database. You'll need to go back to the editor, enter that item into the database, then recompile the game.

During a game, there's room on the screen to display only five items each for the player's inventory and the room's visible items. Ordinarily this isn't a problem, because if a player tries to pick something up when his inventory is full, or tries to drop something in a room that has no space, he'll get a warning from the game, and the command will not be performed. But a problem can arise when a command (an item command) creates an item in the room or player's inventory. What if they're already full?

The item will be created anyway, and an asterisk will be added to the end of the item list (either the visible items or the player's inventory), warning the player that not all items are listed on the screen. To see all items, the player only has to pick up an item if the asterisk is on the room's list, or drop an item if the asterisk is on the inventory list.

Another problem (one I should have warned you about last time when describing the editor), is quote characters in the intro text. You must make sure to use single, not double quotes. This is because the intro text is stored in strings, and string declarations don't work very well if you stick double quotes in the middle of them.

There are four commands built into the game, commands the player will be able to use without you, the game designer, having to enter them into the editor. They are: GET item (the player supplies the "item"), DROP item, SAVE GAME and LOAD GAME. The SAVE and LOAD commands give the player the option of using a disk drive or cassette.

Finally, when typing commands during a game, the player should type complete words, not abbreviations. I know that some adventure games allow you to type *LOO TRE* for LOOK TREE, etc., but that won't work for most commands in a game created by **The Wizard**. Three-letter abbreviations will work for the GET and DROP commands, but I don't recommend using them, since you may use them somewhere else accidentally and get confusing results.

Listing 1.
BASIC listing.

```
EI 1 GOTO 10000
RO 3 A=USR(ADR(D$),ADR(A$),LEN(A$)):? A$:
   A$="":RETURN
EP 4 POSITION N9,N19:? "Okay":RETURN
TD 5 POKE N16,112:POKE 53774,112:RETURN
KO 45 N=N0:S=N0:E=N0:W=N0:U=N0:D=N0:GOSUB
    R*N10:POSITION N12,N2:GOSUB N3:? A$:A
   $="":RETURN
ZH 900 RESTORE 12001:FOR X=N1 TO NN-5I:RE
   AD A$,A:Q=N13-LEN(A$):I$(X*N13-N12,X*N
   13-Q)=A$:I(X)=A:NEXT X
PV 901 RESTORE 12205:FOR X=N1 TO 72:READ
   A:CC$(X)=CHR$(A):NEXT X
QM 902 FOR X=N1 TO 37:READ A:D$(X)=CHR$(A
   ):NEXT X:REM D$(23,23)=CHR$(N0)
AA 903 FOR X=N1 TO 79:READ A:E$(X)=CHR$(A
   ):NEXT X
ZO 904 FOR X=N1 TO 35:READ A:L$(X)=CHR$(A
   ):NEXT X:RETURN
YH 950 IF LEN(A$)>N0 THEN GOSUB N3
VH 953 IF  NOT UL THEN 1100
WI 955 A=USR(ADR(E$),N12,N2,24):GOSUB 45
LE 960 A=USR(ADR(E$),N12,N4,N23):POSITION
    N12,N4:IF N+S+E+W+D+U=N0 THEN ? "None
   ":GOTO 970
DG 962 IF N THEN ? "N ";
IJ 963 IF S THEN ? "S ";
UM 964 IF E THEN ? "E ";
MP 965 IF W THEN ? "W ";
KS 966 IF U THEN ? "U ";
YE 967 IF D THEN ? "D "
EU 970 A=USR(ADR(E$),N12,N6,N13):A=USR(AD
   R(E$),N12,N7,N13):A=USR(ADR(E$),N12,N8
   ,N13)
TN 971 A=USR(ADR(E$),N12,N9,N13):A=USR(AD
   R(E$),N11,N10,N14)
CN 972 ROW=N6:QZ=N0:FOR X=N1 TO NN:IF ABS
   (I(X))<>R THEN 974
CG 973 QZ=QZ+N1:IF ROW<N11 THEN A$=I$(X*N
   13-N12,X*N13):POSITION N12,ROW:GOSUB N
   3:IT=N1:ROW=ROW+N1:NEXT X
IH 974 NEXT X:IF QZ>N5 THEN POSITION N11,
   N10:? "*"
YH 976 IF  NOT IT THEN POSITION N12,ROW:?
    "Nothing"
QE 979 A=USR(ADR(E$),N12,N12,N13):A=USR(A
   DR(E$),N12,N13,N13):A=USR(ADR(E$),N12,
   N14,N13)
PO 980 A=USR(ADR(E$),N12,N15,N13):A=USR(A
```

```
   DR(E$),N12,N16,N13)
LE 981 ROW=N12:FOR X=N1 TO NN:IF I(X)<>-N
   1 THEN NEXT X:GOTO 984
CO 982 A$=I$(X*N13-N12,X*N13):POSITION N1
   2,ROW:GOSUB N3:ROW=ROW+N1:INV=N1:NEXT
   X
IO 984 IF  NOT INV THEN POSITION N12,ROW:
   ? "Nothing"
DA 1100 IT=N0:INV=N0:UL=N0
IL 1260 POKE 752,N0:POSITION N8,N21:SOUND
    N0,N20,N10,N8:FOR X=N1 TO N10:NEXT X:
   SOUND N0,N0,N0,N0
CW 1265 TRAP 1260:INPUT IN$:POKE 752,N1
VV 1280 A=USR(ADR(E$),N9,N18,N25):A=USR(A
   DR(E$),N9,N19,N25):A=USR(ADR(E$),N9,N2
   1,N25):POSITION N9,N18:? IN$
FO 1281 IF LEN(IN$)<N2 THEN 1285
EX 1282 B=USR(ADR(L$),LEN(IN$),ADR(IN$)):
   IF B=N0 THEN POSITION N9,N19:? "WHAT?"
   :GOTO N950
OV 1283 A=ADR(D$):POKE A+N17,N33:POKE A+N
   20,N24:POKE A+N21,105:POKE A+N32,61:A=
   USR(ADR(D$),ADR(IN$),LEN(IN$))
SA 1284 POKE A+N17,61:POKE A+N20,56:POKE
   A+N21,233:POKE A+N32,N33
VL 1285 POSITION N9,N19
ZA 1490 IF LEN(IN$)=N1 THEN V$=IN$:GOTO 1
   781
OB 1500 V$=IN$(N1,B):N$=IN$(B+N2,LEN(IN$)
   )
OQ 1510 IF LEN(V$)=N2 THEN V$(N3)="!"
SS 1520 IF IN$="TBWF!HBNF" THEN 9110
IM 1540 IF IN$="MPBE!HBNF" THEN 9010
SU 1560 IF LEN(V$)<N2 OR LEN(N$)<N3 THEN
   ? "WHAT?":GOTO N950
OE 1580 IF LEN(V$)=N2 THEN V$(N3)=" "
AJ 1620 Y=USR(ADR(CC$),ADR(N$),ADR(IT$),L
   EN(IT$))
SS 1640 Z=USR(ADR(CC$),ADR(V$),ADR(VB$),L
   EN(VB$))
LZ 1720 IF  NOT Y THEN ? "No such noun.":
   GOTO N950
PS 1730 IF  NOT Z THEN ? "No such verb.":
   GOTO N950
PP 1760 IF Z=N1 THEN 2560
QL 1763 IF Z=N2 THEN 2740
BK 1765 A=ASC(V$):A=A-65:GOTO 5902+A*100
JM 1781 REM
ES 1860 IF V$="N" AND N THEN R=N:GOTO 196
   0
RK 1865 IF V$="S" AND S THEN R=S:GOTO 196
   0
JJ 1870 IF V$="E" AND E THEN R=E:GOTO 196
   0
AK 1880 IF V$="W" AND W THEN R=W:GOTO 196
   0
UU 1900 IF V$="U" AND U THEN R=U:GOTO 196
   0
GM 1920 IF V$="D" AND D THEN R=D:GOTO 196
   0
XC 1940 ? "No such direction!":GOTO N950
FS 1960 IF  NOT F THEN GOSUB N4
JU 1970 F=N0:UL=N1:GOTO N950
ES 2000 REM ********* GET *********
EJ 2560 IF I(Y)=-N1 THEN ? "You already h
   ave it!":GOTO N950
PE 2580 IF I(Y)<N0 THEN ? "You can't get
   that!":GOTO N950
VO 2600 IF ABS(I(Y))<>R THEN ? "It's not
   here.":GOTO N950
TL 2620 LOCATE N12,N16,A:COLOR A:PLOT N12
   ,N16
YQ 2630 IF A<>32 THEN POSITION N9,N19:? "
   You can't carry anymore!":GOTO N950
HA 2660 GOSUB N4:I(Y)=-N1:UL=N1:GOTO N950
LL 2680 GOTO N6760
MD 2730 REM ********* DROP *********
MS 2740 IF I(Y)<>-N1 THEN ? "You don't ha
   ve it!":? :? :GOTO N950
FU 2750 LOCATE N12,N10,A:COLOR A:PLOT N12
   ,N10
WF 2755 POSITION N9,N19:IF A<>32 THEN ? "
   There's no more room!":GOTO N950
IP 2761 IF R<>N19 OR I(Y)<>-N1 OR Y<>N15
   OR FT THEN 2840
LI 2840 GOSUB N4
OI 2842 I(Y)=R:UL=N1:GOTO N950
FI 6401 GOTO N9000
FQ 6801 GOTO N9000
FD 7101 GOTO N9000
FF 7201 GOTO N9000
FN 7601 GOTO N9000
FC 8001 GOTO N9000
FO 8601 GOTO N9000
SX 9000 IF V$="MPPL" THEN ? "You see noth
   ing important":GOTO N950
VK 9005 ? "You can't do that!":GOTO N950
PD 9010 TRAP 9100
SB 9020 POSITION N9,N19:? "LOAD FROM TAPE
    OR DISK";:INPUT Z$:IF Z$="D" THEN 906
   0
GK 9030 IF Z$<>"T" THEN ? :GOTO 6820
PA 9040 POSITION N9,N19:? "CUE, PRESS RET
   URN TWICE.":OPEN #N1,N4,N0,"K:":GET #N
   1,A:CLOSE #N1
EI 9050 OPEN #N1,N4,N0,"C:":GOTO 9070
OO 9060 OPEN #N1,N4,N0,"D:ADVENTUR.DAT":A
   $=""
IX 9070 INPUT #N1,R
DP 9080 FOR X=N1 TO NN:INPUT #N1,A:I(X)=A
   :NEXT X
SQ 9090 CLOSE #N1:UL=N1:? :? :TRAP 1260:G
   OTO N950
AS 9100 POSITION N9,N19:? "NO GAME DATA S
   AVED!":CLOSE #N1:GOTO 1100
XA 9110 POSITION N9,N19:? "SAVE TO TAPE O
   R DISK";:INPUT Z$:IF Z$="D" THEN Z$=",
   ":GOTO 9150
VK 9120 IF Z$<>"T" THEN 7060
LO 9130 POSITION N9,N19:? "CUE, PRESS RET
   URN TWICE.":OPEN #N1,N4,N0,"K:":GET #N
   1,A:CLOSE #N1:Z$=","
GS 9140 OPEN #N1,N8,N0,"C:":GOTO 9160
ZB 9150 OPEN #N1,N8,N0,"D:ADVENTUR.DAT"
PQ 9160 PRINT #N1;R
ZW 9170 FOR X=N1 TO NN-SI:PRINT #N1;I(X):
   NEXT X
FW 9180 CLOSE #N1:? :? :A$="":GOTO 1100
HW 10000 N1=1:N2=2:N3=3:N4=4:N5=5:N6=6:N7
   =7:N8=8:N9=9:N10=10:N11=11:N12=12:N13=
   13:N14=14:N15=15:N16=16:N17=17:N18=18
LP 10002 N19=19:N20=20:N21=21:N22=22:N23=
   23:N24=24:N25=25:N26=26:N27=27:N28=28:
   N29=29:N30=30:N31=31:N32=32:N33=33
VB 10004 N34=34:N35=35:N36=36:N37=37:N38=
   38:N39=39:N40=40:N41=41:N42=42:N43=43:
   N44=44:N45=45:N46=46:N47=47
ZN 10006 N48=48:N49=49:N50=50:N950=950:N9
   000=9000
PN 10009 GRAPHICS N0:POKE 559,N0:POKE 82,
   N2:POKE 752,N1:DL=PEEK(560)+PEEK(561)*
   256+N4
OP 10010 POKE DL+N7,N7:POKE DL+N11,N6:POK
   E DL+N12,N6:POKE DL+N13,N6
LO 10020 POKE DL+N24,65:POKE DL+N25,PEEK(
   560):POKE DL+N26,PEEK(561):POKE 710,N0
ZP 10025 DIM NA$(N18),T$(N18)
YL 10030 RESTORE 12200:READ NA$,T$
KQ 10040 POSITION N9-INT(LEN(T$)/N2),N6:?
    T$:POSITION N28,N9:? "BY":POSITION N2
   9-INT(LEN(NA$)/N2),N10:? NA$
ES 10050 POSITION N7,N18:? "Programmed by
    THE WIZARD":POKE 559,N34
WO 10100 DIM VB$(NV*N3),I$(NN*N13),A$(160
   ),IN$(N26),V$(N10),N$(N10),CC$(77),D$(
   37),Z$(N1)
```

```
QF 10110 DIM IT$(NN*N3),I(NN),E$(85),L$(3
   5),FA$(N19),SA$(N15),VA$(N11),TA$(N10)
   ,INTRO$(L)
VZ 10113 R=N5:UL=N1
SB 10120 I$(N1)=" ":I$(NN*N13)=" ":I$(N2)
   =I$
YJ 10300 GOSUB 900:A$=""
WT 10400 GRAPHICS N0:POKE 710,N0:POKE 709
   ,N10:POKE 752,N1
GU 10410 POSITION N2,N4:? INTRO$:POSITION
    N13,N22:? "PRESS ANY KEY"
PD 10420 OPEN #N1,N4,N0,"K:":GET #N1,A:CL
   OSE #N1
HS 11000 GRAPHICS N0:GOSUB N5:POKE 559,N0
   :DL=PEEK(560)+256*PEEK(561)+N4:POKE DL
   -N1,71:POKE DL+N2,N6
HK 11010 POKE DL+N19,130:POKE DL+N24,65:P
   OKE DL+N25,PEEK(560):POKE DL+N26,PEEK(
   561)
AY 11020 POKE 82,N0:POKE 708,136:POKE 709
   ,N10:POKE 710,N0:POKE 712,112:POKE 752
   ,N1
LG 11022 RESTORE 11024:FOR X=N0 TO N19:RE
   AD A:POKE 1536+X,A:NEXT X
EI 11024 DATA 72,138,72,169,0,162,6,141,1
   0,212,141,23,208,142,24,208,104,170,10
   4,64
RV 11026 POKE 512,N0:POKE 513,N6:POKE 542
   86,192
FB 11030 POSITION N9-INT(LEN(T$)/N2),N0:?
    T$
WQ 11040 POSITION N1,N2:? "     PLACE:":PO
   SITION N1,N4:? "     EXITS:"
WS 11050 POSITION N1,N6:? "  YOU SEE:"
VT 11080 POSITION N1,N12:? "  YOU HAVE:"
YA 11090 POSITION N1,N21:? "COMMAND":POKE
    559,N34:GOTO N950
UF 12205 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
QF 12210 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
GA 12220 DATA 16,141,255,6,165,205,24,105
   ,3,133,205,144,217,230,206,176,213,169
   ,0,133,212,96
CP 12230 DATA 216,104,104,133,204,104,133
   ,203,104,104,133,205,160,0,177,203,201
   ,61,240,11,56,233,1,145,203
PV 12240 DATA 200,196,205,208,240,96,169,
   33,240,244,208,242
DL 12250 DATA 104,104,104,141,253,6,104,1
   04,141,254,6,104,104,141,255,6,165,88,
   133,203,165,89,133,204,174
SK 12260 DATA 254,6,224,0,240,14,165,203,
   24,105,40,133,203,144,2,230,204,202,20
   8,242,165,203,24,109,253,6,133
XN 12270 DATA 203,144,2,230,204,56,165,20
   3,233,1,133,203,176,2,198,204,172,255,
   6,169,0,145,203,136,208,251,96
IY 12280 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
AP 12290 DATA 244,160,0,132,212,169,0,133
   ,213,96
JY 13000 GRAPHICS N0:POKE 752,N1:POKE 710
   ,N0:POSITION N19-LEN(A$)/N2,N4:GOSUB N
   3
SY 13010 POSITION N8,N8:? "This adventure
    is over!":POSITION N7,N12:? "Do you w
   ant to play again?"
OX 13020 OPEN #N1,N4,N0,"K:":GET #N1,A:CL
   OSE #N1
LQ 13030 IF A=ASC("Y") THEN RUN
OP 13040 IF A=ASC("N") THEN GRAPHICS 0:EN
   D
YX 13050 GOTO 13020
```

Listing 2.
BASIC listing.

```
UA 10 REM ****************************
IT 20 REM *        THE WIZARD        *
HH 30 REM *      COMPILER MODULE     *
KV 40 REM *            BY            *
JS 50 REM *      CLAYTON WALNUM      *
UF 60 REM ****************************
VT 70 GRAPHICS 0:POKE 710,0
WT 80 N1=1:N2=2:N3=3:N4=4:N5=5:N6=6:N7=7:
   N8=8:N9=9:N10=10
LT 90 N11=11:N12=12:N13=13:N14=14:N15=15:
   N16=16:N17=17:N18=18:N19=19:N20=20:GOT
   O 3020
GY 100 RESTORE 3130:FOR X=N1 TO 72:READ A
   :CC$(X)=CHR$(A):NEXT X
MV 110 FOR X=N1 TO 26:READ A:TR$(X)=CHR$(
   A):NEXT X
JO 120 FOR X=N1 TO 100:READ A:55$(X)=CHR$
   (A):NEXT X
TD 130 FOR X=N1 TO 37:READ A:E$(X)=CHR$(A
   ):NEXT X:RETURN
PM 140 A=USR(ADR(TR$),ADR(N$),LEN(N$)):RE
   TURN
PS 150 A=USR(ADR(55$),ADR(C$),ADR(IT$),LE
   N(IT$),N13):IF  NOT A THEN 3270
ZJ 160 RETURN
WY 170 GOSUB 140:L$(LEN(L$)+N1)=Q$:N$=N$(
   N1,A+N1):GOSUB 180:L$(LEN(L$)+N1)=N$:L
   $(LEN(L$)+N1)=Q$:RETURN
AF 180 A=USR(ADR(E$),ADR(N$),LEN(N$)):RET
   URN
YU 190 A=USR(ADR(CC$),ADR(N$),ADR(IT$),LE
   N(IT$)):RETURN
RX 200 LV=LV-64:L(LV)=L(LV)+N2:L$(N1)=STR
   $(5900+LV*100+L(LV)):RETURN
IE 210 INPUT #N1;CM$:N$=CM$(N1,N3):GOSUB
   180:GOSUB 190:RETURN
KC 220 C$(CNT*N3-N2,CNT*N3)=N$:VCNT=VCNT+
   N1:CNT=CNT+N1:IF CNT>28 THEN GOSUB 255
   0
LO 230 IT$(VCNT*N3-N2,VCNT*N3)=N$:RETURN
WG 240 L$(N5)=" IF IN$=":N$=CM$(N1,N15):G
   OSUB 170:RETURN
MN 250 C$=CM$(N16,28):GOSUB 150
JK 260 IF CM$(81,81)="I" THEN L$(LEN(L$)+
   N1)=" AND I(N":L$(LEN(L$)+N1)=STR$(A):
   L$(LEN(L$)+N1)=")=-N1":GOTO 280
YC 270 L$(LEN(L$)+N1)=" AND AB$(I(N":L$(L
   EN(L$)+N1)=STR$(A):L$(LEN(L$)+N1)=")=
   R"
RF 280 L$(LEN(L$)+N1)=" THEN A$=":N$=CM$(
   30,54):GOSUB 170:RETURN
NM 290 L$(LEN(L$)+N1)=" THEN A$=":N$=CM$(
   55,79):GOSUB 170:RETURN
EC 300 REM ****************************
BP 310 REM *        ROOM LINES        *
EG 320 REM ****************************
QV 330 F$(LEN(F$)-N2)="RM5":OPEN #N1,N4,N
   0,F$
RM 340 FOR X=N1 TO N6:D(X)=N0:NEXT X
WM 350 TRAP 3230:INPUT #N1;RM5$
UO 360 ? "CREATING ROOM LINE"
HY 370 ROOM=ASC(RM5$):ROOM=ROOM+N4:L$(N1)
   =STR$(ROOM*N10)
NT 380 L$(LEN(L$)+N1)=" A$=":N$=RM5$(N2,2
   5):GOSUB 170
ZT 390 FOR X=N1 TO N6:IF ASC(RM5$(26+X,26
   +X))=N0 THEN NEXT X:GOTO 420
DZ 400 L$(LEN(L$)+N1)=":":R=ASC(RM5$(26+X
   ))+N4:IF R>34 THEN R=R-30:D(X)=-R
FR 410 L$(LEN(L$)+N1)=DIR$(X,X):L$(LEN(L$
   )+N1)="=N":L$(LEN(L$)+N1)=STR$(R):NEXT
    X
SJ 420 L$(LEN(L$)+N1)=":RETURN":? #N2;L$
EJ 430 REM ****************************
ZT 440 REM *   POS RESTRICTED EXITS   *
```

```
EN 450 REM ******************************
CO 460 FOR Y=N1 TO N6:IF D(Y)>-N1 THEN NE
   XT Y:GOTO 340
LU 470 ? "CREATING POS RESTRICTED LINE"
ME 480 L$(N1)=STR$(1780+ROOM*N2):L$(LEN(L
   $)+N1)=" IF R=N":L$(LEN(L$)+N1)=STR$(R
   OOM)
WP 490 L$(LEN(L$)+N1)=" AND IN$=":L$(LEN(
   L$)+N1)=Q$:L$(LEN(L$)+N1)=DIR$(Y,Y):L$
   (LEN(L$)+N1)=Q$
OJ 500 C$=RM5$(33,45):GOSUB 150
YE 510 IF RM5$(96,96)="R" THEN L$(LEN(L$)
   +N1)=" AND ABS(I(N":L$(LEN(L$)+N1)=STR
   $(A):L$(LEN(L$)+N1)="))=R":GOTO 530
TV 520 L$(LEN(L$)+N1)=" AND I(N":L$(LEN(L
   $)+N1)=STR$(A):L$(LEN(L$)+N1)=")=-N1"
LV 530 L$(LEN(L$)+N1)=" THEN A$=":N$=RM5$
   (46,70):GOSUB 170
SS 540 L$(LEN(L$)+N1)=":R=N":L$(LEN(L$)+N
   1)=STR$(-D(Y))
JT 550 L$(LEN(L$)+N1)=":UL=N1:G.N950":? #
   N2;L$
YM 560 REM ******************************
UA 570 REM *   NEG RESTRICTED EXITS   *
YQ 580 REM ******************************
RT 590 IF RM5$(71,71)=" " THEN 340
MK 600 L$(N1)=STR$(1781+ROOM*N2):L$(LEN(L
   $)+N1)=" IF R=N":L$(LEN(L$)+N1)=STR$(R
   OOM)
UK 610 ? "CREATING NEG RESTRICTED LINE"
WD 620 L$(LEN(L$)+N1)=" AND IN$=":L$(LEN(
   L$)+N1)=Q$:L$(LEN(L$)+N1)=DIR$(Y,Y):L$
   (LEN(L$)+N1)=Q$
UJ 630 L$(LEN(L$)+N1)=" THEN A$=":N$=RM5$
   (71,95):GOSUB 170
DJ 640 L$(LEN(L$)+N1)=":G.N950":? #N2;L$:
   GOTO 340
YL 650 REM ******************************
LM 660 REM *         ITEM DATA          *
YP 670 REM ******************************
VZ 680 CLOSE #N1:FI$(LEN(FI$)-N2)="ITM":O
   PEN #N1,N4,N0,FI$:L$="":I=N0
NS 690 I=I+N1:IF I>ICNT THEN 1150
WT 700 INPUT #N1;I$
XD 710 ? "CREATING ITEM DATA LINE"
XM 720 IF I$(N17,N17)="♥" THEN 5I=5I+N1:G
   OTO 690
WU 730 L$(N1)=STR$(12000+I):L$(LEN(L$)+N1
   )=" DATA ":N$=I$(N1,N13):GOSUB 140
WS 740 N$=N$(N1,A+N1):GOSUB 180:L$(LEN(L$
   )+N1)=N$
TV 750 L$(LEN(L$)+N1)=",":IF I$(N18,21)="
   NONE" THEN 770
HB 760 IF (I$(N18,N20)<>"GET" AND I$(33,3
   5)<>I$(N14,N16)) OR I$(N17,N17)="C" TH
   EN R=0:GOTO 780
AL 770 R=ASC(I$(N17))+N4:IF R>34 THEN R=-
   R+30
QW 780 L$(LEN(L$)+N1)=STR$(R):? #N2;L$
EY 790 REM ******************************
PN 800 REM *   ACCESS TO ITEMS LINES   *
EJ 810 REM ******************************
TQ 820 IF I$(N18,N20)="GET" OR I$(N18,21)
   ="NONE" OR I$(N17,N17)="C" THEN 690
SD 830 LV=ASC(I$(N18)):IF LV=32 THEN 690
QX 840 GOSUB 200:L$(N5)="IF IN$<>"
YE 850 ? "CREATING POS ACCESS LINE"
AU 860 N$=I$(N18,32):GOSUB 170
RS 870 L$(LEN(L$)+N1)=" OR I(N":L$(LEN(L$
   )+N1)=STR$(I-5I):L$(LEN(L$)+N1)=")"
BG 880 IF I$(N17,N17)="A" THEN 910
DC 890 L$(LEN(L$)+N1)=" OR R<>N":R=ASC(I$
   (N17))+N4:IF R>34 THEN R=R-30
GX 900 L$(LEN(L$)+N1)=STR$(R)
GW 910 IF I$(33,36)="NONE" THEN 950
HA 920 C$=I$(33,45):GOSUB 150
HF 930 IF I$(96,96)="I" THEN L$(LEN(L$)+N
   1)=" OR I(N":L$(LEN(L$)+N1)=STR$(A):L$
   (LEN(L$)+N1)=")<>-N1":GOTO 950
QZ 940 L$(LEN(L$)+N1)=" OR ABS(I(N":L$(LE
   N(L$)+N1)=STR$(A):L$(LEN(L$)+N1)="))<>
   R"
JF 950 L$(LEN(L$)+N1)=" THEN ":L$(LEN(L$)
   +N1)=STR$(VAL(L$)+N2):? #N2;L$
QH 960 L$(N1)=STR$(VAL(L$)+N1):L$(N5)=" A
   $=":N$=I$(46,70):GOSUB 170
BT 970 L$(LEN(L$)+N1)=":I(N":L$(LEN(L$)+N
   1)=STR$(I-5I):L$(LEN(L$)+N1)=")="
TS 980 IF I$(N17,N17)="A" THEN L$(LEN(L$)
   +N1)="R":GOTO 1010
AR 990 R=ASC(I$(N17))+N4:IF R>34 THEN R=-
   R+30
MT 1000 L$(LEN(L$)+N1)=STR$(R)
ZB 1010 L$(LEN(L$)+N1)=":UL=N1:G.N950":?
   #N2;L$
KP 1020 L$(N1)=STR$(VAL(L$)+N1):L$(N5)="
   REM":? #N2;L$
GO 1030 ? "CREATING NEG ACCESS LINE"
UR 1040 IF I$(71,71)=" " THEN 690
CJ 1050 LV=ASC(I$(N18)):GOSUB 200
SV 1060 L$(N5)="IF IN$=":N$=I$(N18,32):GO
   SUB 170
ND 1070 L$(LEN(L$)+N1)=" AND R=N":R=ASC(I
   $(N17))+N4:IF R>34 THEN R=R-30
NR 1080 L$(LEN(L$)+N1)=STR$(R)
WT 1090 L$(LEN(L$)+N1)=" AND NOT I(N":L$(
   LEN(L$)+N1)=STR$(I-5I):L$(LEN(L$)+N1)=
   ") THEN A$="
SU 1100 N$=I$(71,95):GOSUB 170
MM 1110 L$(LEN(L$)+N1)=":G.N950":? #N2;L$
   :GOTO 690
QE 1120 REM ******************************
OX 1130 REM *      COMMAND LINES       *
QK 1140 REM ******************************
LU 1150 CLOSE #N1:FI$(LEN(FI$)-N2)="COM":
   OPEN #N1,N4,N0,FI$:TRAP 3310
WM 1160 INPUT #N1;CM$:LV=ASC(CM$(N1)):GOS
   UB 200
PC 1170 ? "CREATING COMMAND LINE"
YD 1180 GOSUB 240
ZI 1190 IF ASC(CM$(29))=N0 THEN 1210
EA 1200 L$(LEN(L$)+N1)=" AND R=N":L$(LEN(
   L$)+N1)=STR$(ASC(CM$(29))+N4)
GG 1210 IF CM$(N16,N19)="NONE" THEN GOSUB
    280:GOTO 1230
YA 1220 GOSUB 250
AK 1230 L$(LEN(L$)+N1)=":G.N950":? #N2;L$
VS 1240 IF CM$(55,55)=" " OR CM$(N16,N19)
   ="NONE" THEN 1160
TQ 1250 LV=ASC(CM$(N1)):GOSUB 200:GOSUB 2
   40
AI 1260 IF ASC(CM$(29))=N0 THEN GOTO 1280
EV 1270 L$(LEN(L$)+N1)=" AND R=N":L$(LEN(
   L$)+N1)=STR$(ASC(CM$(29))+N4)
TU 1280 L$(LEN(L$)+N1)=" THEN A$=":N$=CM$
   (55,79):GOSUB 170
MD 1290 L$(LEN(L$)+N1)=":G.N950":? #N2;L$
   :GOTO 1160
II 1300 REM ******************************
BJ 1310 REM *   FATAL COMMAND LINES    *
IO 1320 REM ******************************
JU 1330 FI$(LEN(FI$)-N2)="FAT":CLOSE #N1:
   OPEN #N1,N4,N0,FI$:TRAP 3350
MR 1340 INPUT #N1;CM$:IF CM$="" THEN 2320
IY 1350 ? "CREATING 'FATAL' LINE"
KF 1360 LV=ASC(CM$(N1)):GOSUB 200
YE 1370 GOSUB 240
AN 1380 IF ASC(CM$(29))=N0 THEN 1400
SA 1390 L$(LEN(L$)+N1)=" AND R=":L$(LEN(L
   $)+N1)=STR$(ASC(CM$(29))+N4)
CZ 1400 IF CM$(N16,N19)="NONE" THEN 1400
YB 1410 GOSUB 250
IY 1420 L$(LEN(L$)+N1)=":G.13000":? #N2;L
   $
VR 1430 IF CM$(55,55)=" " OR CM$(N16,N19)
   ="NONE" THEN 1340
```

```
KB 1440 LV=ASC(CM$(N1)):GOSUB 200
YA 1450 GOSUB 240
RT 1460 L$(LEN(L$)+N1)=" AND R=":L$(LEN(L
   $)+N1)=STR$(ASC(CM$(29))+N4)
TV 1470 L$(LEN(L$)+N1)=" THEN A$=":N$=CM$
   (55,79):GOSUB 170
MC 1480 L$(LEN(L$)+N1)=":G.N950":? #N2;L$
   :GOTO 1340
DH 1490 REM ***************************
WY 1500 REM *       MOVE COMMANDS     *
CL 1510 REM ***************************
KT 1520 FI$(LEN(FI$)-N2)="MOV":CLOSE #N1:
   OPEN #N1,N4,N0,FI$:TRAP 3370
WL 1530 INPUT #N1;CM$:LV=ASC(CM$(N1)):GOS
   UB 200
YO 1540 ? "CREATING MOVEMENT LINE"
YC 1550 GOSUB 240
LM 1560 IF ASC(CM$(29))=N0 THEN 1580
FB 1570 L$(LEN(L$)+N1)=" AND R=N":L$(LEN(
   L$)+N1)=STR$(ASC(CM$(29))+N4)
JC 1580 IF CM$(N16,N19)="NONE" THEN GOSUB
    280:GOTO 1600
ZB 1590 GOSUB 250
TA 1600 L$(LEN(L$)+N1)=":R=N":L$(LEN(L$)+
   N1)=STR$(ASC(CM$(80))+N4):L$(LEN(L$)+N
   1)=":UL=N1:G.N950":? #N2;L$
XP 1610 IF CM$(55,55)=" " OR CM$(N16,N19)
   ="NONE" THEN 1530
JZ 1620 LV=ASC(CM$(N1)):GOSUB 200
XY 1630 GOSUB 240
KB 1640 IF ASC(CM$(29))=N0 THEN 1660
RU 1650 L$(LEN(L$)+N1)=" AND R=":L$(LEN(L
   $)+N1)=STR$(ASC(CM$(29))+N4)
AU 1660 GOSUB 290
NW 1670 L$(LEN(L$)+N1)=":G.N950":? #N2;L$
   :GOTO 1530
DI 1680 REM ***************************
SQ 1690 REM *       ITEM COMMANDS     *
CM 1700 REM ***************************
KV 1710 FI$(LEN(FI$)-N2)="CHG":CLOSE #N1:
   OPEN #N1,N4,N0,FI$:TRAP 3390
WM 1720 INPUT #N1;CM$:LV=ASC(CM$(N1)):GOS
   UB 200
VG 1730 ? "CREATING ITEM COMMAND LINE"
BA 1740 L$(N5)=" IF IN$<>":N$=CM$(N1,N15)
   :GOSUB 170
RN 1750 IF CM$(N16,N19)="NONE" THEN 1770
LS 1760 L$(LEN(L$)+N1)=" OR I(N":C$=CM$(N
   16,28):GOSUB 150:L$(LEN(L$)+N1)=STR$(A
   ):L$(LEN(L$)+N1)=")<>-N1"
XY 1770 IF CM$(122,125)="NONE" THEN 1800
OG 1780 L$(LEN(L$)+N1)=" OR I(N":C$=CM$(1
   22,134):GOSUB 150
DT 1790 L$(LEN(L$)+N1)=STR$(A):L$(LEN(L$)
   +N1)=")<>-N1"
MM 1800 IF CM$(81,84)="NONE" THEN 1830
VG 1810 L$(LEN(L$)+N1)=" OR I(N":C$=CM$(8
   1,93):GOSUB 150
PW 1820 L$(LEN(L$)+N1)=STR$(A):L$(LEN(L$)
   +N1)=")<>N0"
LG 1830 IF ASC(CM$(29))=N0 THEN 1850
OQ 1840 L$(LEN(L$)+N1)=" OR R<>N":L$(LEN(
   L$)+N1)=STR$(ASC(CM$(29))+N4)
BL 1850 L$(LEN(L$)+N1)=" THEN ":L$(LEN(L$
   )+N1)=STR$(VAL(L$)+N2):? #N2;L$
BK 1860 L$(N1)=STR$(VAL(L$)+N1)
NG 1870 IF CM$(81,84)="NONE" THEN 1920
QK 1880 L$(N5)=" I(N":C$=CM$(81,93):GOSUB
    150
JN 1890 L$(N8)=STR$(A):L$(LEN(L$)+N1)=")=
   ":IF CM$(95,95)="N" AND CM$(94,94)="R"
    THEN L$(LEN(L$)+N1)="-R:":GOTO 1920
MP 1900 IF CM$(94,94)="R" THEN L$(LEN(L$)
   +N1)="R":GOTO 1920
GE 1910 L$(LEN(L$)+N1)="-N1:"
BR 1920 IF CM$(96,99)="NONE" THEN 1980
LA 1930 C$=CM$(96,108):GOSUB 150
SP 1940 L$(LEN(L$)+N1)="I(N":L$(LEN(L$)+N
   1)=STR$(A):L$(LEN(L$)+N1)=")=N0:"
KJ 1950 IF CM$(109,112)="NONE" THEN 1980
VQ 1960 L$(LEN(L$)+N1)="I(N":C$=CM$(109,1
   21):GOSUB 150
KR 1970 L$(LEN(L$)+N1)=STR$(A):L$(LEN(L$)
   +N1)=")=N0:"
FW 1980 L$(LEN(L$)+N1)="UL=N1:A$=":N$=CM$
   (30,54):GOSUB 170
BQ 1990 L$(LEN(L$)+N1)=":G.N950":? #N2;L$
KK 2000 L$(N1)=STR$(VAL(L$)+N1):L$(N5)="
   REM":? #N2;L$
OQ 2010 IF CM$(55,55)=" " THEN 1720
BD 2020 LV=ASC(CM$(N1)):GOSUB 200:? "CREA
   TING NEG ITEM COMMAND LINE"
UW 2030 L$(N5)=" IF IN$=":N$=CM$(N1,N15):
   GOSUB 170
CJ 2040 L$(LEN(L$)+N1)=" AND ":IF CM$(122
   ,125)<>"NONE" THEN L$(LEN(L$)+N1)="("
XE 2050 L$(LEN(L$)+N1)=" I(N":C$=CM$(N16,
   28):GOSUB 150:L$(LEN(L$)+N1)=STR$(A)
HW 2060 L$(LEN(L$)+N1)=")<>-N1":IF CM$(12
   2,125)="NONE" THEN 2080
TH 2070 L$(LEN(L$)+N1)=" OR I(N":C$=CM$(1
   22,134):GOSUB 150:L$(LEN(L$)+N1)=STR$(
   A):L$(LEN(L$)+N1)=")<>-N1)"
TR 2080 L$(LEN(L$)+N1)=" THEN A$=":N$=CM$
   (55,79):GOSUB 170
PK 2090 L$(LEN(L$)+N1)=":G.N950":? #N2;L$
   :GOTO 1720
IF 2100 REM ***************************
UJ 2110 REM *       FINAL COMMANDS    *
IL 2120 REM ***************************
GJ 2130 FI$(LEN(FI$)-N2)="FIN":CLOSE #N1:
   OPEN #N1,N4,N0,FI$:TRAP 3330
WH 2140 INPUT #N1;CM$:LV=ASC(CM$(N1)):GOS
   UB 200
BQ 2150 ? "CREATING GAME END LINE"
XY 2160 GOSUB 240
IY 2170 IF ASC(CM$(29))=N0 THEN 2190
EX 2180 L$(LEN(L$)+N1)=" AND R=N":L$(LEN(
   L$)+N1)=STR$(ASC(CM$(29))+N4)
FC 2190 IF CM$(N16,N19)="NONE" THEN GOSUB
    280:GOTO 2210
XV 2200 GOSUB 250
ZF 2210 L$(LEN(L$)+N1)=":G.13000":? #2;L$
TJ 2220 IF CM$(55,55)=" " OR CM$(N16,N19)
   ="NONE" THEN 2140
JV 2230 LV=ASC(CM$(N1)):GOSUB 200
XU 2240 GOSUB 240
HM 2250 IF ASC(CM$(29))=N0 THEN 2270
RQ 2260 L$(LEN(L$)+N1)=" AND R=":L$(LEN(L
   $)+N1)=STR$(ASC(CM$(29))+N4)
AQ 2270 GOSUB 290
KE 2280 L$(LEN(L$)+N1)=":G.N950":? #N2;L$
   :GOTO 2140
JI 2290 REM ***************************
EO 2300 REM *  CREATE SEARCH STRINGS  *
IM 2310 REM ***************************
JA 2320 ? "CREATING VERB TABLE LINE"
PY 2330 FI$(LEN(FI$)-N2)="FAT":C$="":CNT=
   N1:VCNT=N2:IT$="HFUESP"
GM 2340 L$(N1)="10210 ":L$(N7)="VB$(N1)="
   :L$(N15)=Q$:L$(N16)=IT$
US 2350 CLOSE #N1:OPEN #N1,N4,N0,FI$:TRAP
    2380:LCNT=N0:? "FATAL"
UG 2360 GOSUB 210:IF A THEN 2360
AD 2370 GOSUB 220:GOTO 2360
WU 2380 FI$(LEN(FI$)-N2)="COM":CLOSE #N1:
   OPEN #N1,N4,N0,FI$:TRAP 2410:? "COMMAN
   D"
XV 2390 GOSUB 210:IF A THEN 2390
CB 2400 GOSUB 220:GOTO 2390
BJ 2410 FI$(LEN(FI$)-N2)="ITM":CLOSE #N1:
   OPEN #N1,N4,N0,FI$:TRAP 2450:? "ITEM"
SW 2420 INPUT #N1;I$:IF I$(N18,21)="NONE"
    THEN 2420
SH 2430 N$=I$(N18,N20):GOSUB 180:GOSUB 19
   0:IF A THEN 2420
```

```
XE 2440 GOSUB 220:GOTO 2420
NG 2450 FI$(LEN(FI$)-N2)="MOV":CLOSE #N1:
   OPEN #N1,N4,N0,FI$:TRAP 2480:? "MOVE"
VJ 2460 GOSUB 210:IF A THEN 2460
BB 2470 GOSUB 220:GOTO 2460
QD 2480 FI$(LEN(FI$)-N2)="FIN":CLOSE #N1:
   OPEN #N1,N4,N0,FI$:TRAP 2510:? "FINAL
   "
YY 2490 GOSUB 210:IF A THEN 2490
CZ 2500 GOSUB 220:GOTO 2490
RL 2510 FI$(LEN(FI$)-N2)="CHG":CLOSE #N1:
   OPEN #N1,N4,N0,FI$:TRAP 2540:? "CHANGE
   "
YF 2520 GOSUB 210:IF A THEN 2490
XZ 2530 GOSUB 220:GOTO 2520
LJ 2540 L$(LEN(L$)+N1)=C$:L$(LEN(L$)+N1)=
   Q$:? #N2;L$:GOTO 2580
PY 2550 L$(LEN(L$)+N1)=C$:L$(LEN(L$)+N1)=
   Q$:? #N2;L$
TS 2560 LCNT=LCNT+N1:L$(N1)=STR$(10210+LC
   NT):L$(N6)=" VB$("
QH 2570 L$(N11)=STR$(LCNT*84):L$(LEN(L$)+
   N1)=")=":L$(LEN(L$)+N1)=Q$:CNT=N1:C$="
   ":RETURN
TW 2580 FI$(LEN(FI$)-N2)="ITM":CLOSE #N1:
   OPEN #N1,4,0,FI$:L$(N1)="10200 IT$=":L
   $(LEN(L$)+N1)=Q$
KV 2590 ? "CREATING ITEM TABLE LINE"
WL 2600 FOR I=N1 TO ICNT:INPUT #N1;I$:N$=
   I$(N14,N16):GOSUB 180
EU 2610 IF I$(N17,N17)="♥" THEN SI$(LEN(S
   I$)+N1)=N$:NEXT I:GOTO 2660
UG 2620 IF INT(I/30)<>I/30 THEN 2650
UG 2630 L$(LEN(L$)+N1)=N$:L$(LEN(L$)+N1)=
   Q$:? #N2;L$:L$(N1)="10201 IT$(91)="
UG 2640 L$(LEN(L$)+N1)=Q$:NEXT I:GOTO 266
   0
ZD 2650 L$(LEN(L$)+N1)=N$:NEXT I
XG 2660 L$(LEN(L$)+N1)=SI$:L$(LEN(L$)+N1)
   =Q$:? #N2;L$:CLOSE #N1
DG 2670 REM ***************************
FG 2680 REM *       MISC LINES        *
DM 2690 REM ***************************
CV 2700 ? "CREATING MISC LINES"
KU 2710 TRAP 2720:FI$(LEN(FI$)-N2)="INT":
   CLOSE #N1:OPEN #N1,N4,N0,FI$:GET #N1,L
   H:GET #N1,LL:L=LH*78+LL:CLOSE #N1
YK 2720 L$(N1)="10060 NN=":L$(N10)=STR$(I
   CNT):L$(LEN(L$)+N1)=":NV=":L$(LEN(L$)+
   N1)=STR$(VCNT)
GT 2730 L$(LEN(L$)+N1)=":SI=":L$(LEN(L$)+
   N1)=STR$(SI):IF L=N0 THEN L=N1
KR 2740 L$(LEN(L$)+N1)=":L=":L$(LEN(L$)+N
   1)=STR$(L):? #N2;L$
WR 2750 FI$(LEN(FI$)-N2)="NAM":CLOSE #N1:
   OPEN #N1,N4,N0,FI$
DS 2760 INPUT #N1;N$:INPUT #N1;T$:CLOSE #
   N1
IM 2770 L$(N1)="12200 DATA ":L$(N12)=N$
TV 2780 FOR X=N1 TO LEN(T$):IF T$(X,X)>="
   A" AND T$(X,X)<="Z" THEN T$(X,X)=CHR$(
   ASC(T$(X,X))+32)
MO 2790 NEXT X
KE 2800 L$(LEN(L$)+N1)=",":L$(LEN(L$)+N1)
   =T$:? #N2;L$
EP 2810 L$(N1)="9060 OPEN#N1,N4,N0,":L$(N
   20)=Q$:L$(21)="D:":L$(23)=NA$:L$(LEN(L
   $)+N1)=".DAT"
IZ 2820 REM ***************************
XI 2830 REM *         INTRO           *
JF 2840 REM ***************************
EC 2850 TRAP 2900:FI$(LEN(FI$)-N2)="INT":
   OPEN #N1,N4,N0,FI$:GET #N1,LH:GET #N1,
   LL:L=LH*78+LL
XP 2860 FOR X=N1 TO LH+N1:INPUT #N1;N$:L$
   (N1)=STR$(10129+X):L$(N6)=" INTRO$("
RZ 2870 L$(N14)=STR$(X*78-77):L$(LEN(L$)+
   N1)=")=":L$(LEN(L$)+N1)=Q$
SP 2880 REM IF X=LH+N1 THEN GOSUB 200:N$=
   N$(N1,A+N1)
ZD 2890 L$(LEN(L$)+N1)=N$:L$(LEN(L$)+N1)=
   Q$:? #N2;L$:NEXT X
CN 2900 CLOSE #N2
CU 2910 REM ***************************
TN 2920 REM *     CREATE PROGRAM      *
DA 2930 REM ***************************
HH 2940 GRAPHICS N0:POKE 710,N0:? :? :? "
   LOAD ";CHR$(34);"D:WIZFRAME.BAS";CHR$(
   34)
LV 2950 FI$(LEN(FI$)-N2)="LNS":? :? :? "E
   NTER ";CHR$(34);FI$;CHR$(34)
DU 2960 FI$(LEN(FI$)-N2)="BAS":? :? :? "S
   AVE ";CHR$(34);FI$;CHR$(34):? :? :? "P
   OKE 842,12":? :? :? "RUN"
DE 2970 POSITION N0,N0:POKE 842,13
QC 2980 STOP
DS 2990 REM ***************************
OF 3000 REM *       INITIALIZE        *
CD 3010 REM ***************************
RF 3020 DIM F$(N15),FI$(N15),R$(N2),D(N6)
   ,RM$$(96),L$(120),DIR$(N6),C$(90),Q$(N
   1),I$(96),IT$(650),L(26)
CJ 3030 DIM CM$(134),N$(78),T$(N18),TR$(2
   6),VB$(N11),NN$(N12),CC$(72),SS$(100),
   SI$(650),NA$(N8),E$(37)
JB 3040 F$="D:":DIR$="NSEWUD":Q$=CHR$(34)
XZ 3050 OPEN #N1,N4,N0,"D:FILENAME.DAT":I
   NPUT #N1;NA$:CLOSE #N1:F$(N3)=NA$:F$(L
   EN(F$)+N1)=".LNS"
HD 3060 OPEN #N1,N8,N0,F$:CLOSE #N1:OPEN
   #N2,N9,N0,F$
JA 3070 FOR X=N1 TO 26:L(X)=N0:NEXT X:ICN
   T=N0:SI=N0
CD 3080 FI$=F$:FI$(LEN(FI$)-N2)="ITM":OPE
   N #N1,N4,N0,FI$
UC 3090 TRAP 3120:FOR X=N1 TO 50:INPUT #N
   1;I$:ICNT=ICNT+N1
IB 3100 IF I$(N17,N17)="♥" THEN SI$(LEN(S
   I$)+N1)=I$(N1,N13):SI=SI+N1:NEXT X:GOT
   O 3120
IX 3110 IT$((X-SI)*N13-N12,(X-SI)*N13)=I$
   (N1,N13):NEXT X
YA 3120 POP :CLOSE #N1:IT$(LEN(IT$)+N1)=S
   I$:SI$="":SI=N0:GOSUB 100:GOTO 330
AZ 3130 DATA 104,104,133,204,104,133,203,
   104,133,206,104,133,205,104,104,133,20
   7,162,0,142,255,6,134,213,232
KY 3140 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
SB 3150 DATA 16,141,255,6,165,205,24,105,
   3,133,205,144,217,230,206,176,213,169,
   0,133,212,96
VX 3160 DATA 104,104,133,209,104,133,208,
   104,104,168,136,240,6,177,208,201,32,2
   40,247,132,212,169,0,133,213,96
EB 3170 DATA 104,104,133,204,104,133,203,
   104,133,206,104,133,205,104,141,1,6,10
   4,141,0,6,104,104,133,207
KT 3180 DATA 162,0,142,2,6,142,3,6,134,21
   3,232,160,0,177,203,209,205,208,8,200,
   196,207,208,245,134
DR 3190 DATA 212,96,173,2,6,24,101,207,14
   1,2,6,144,3,238,3,6,173,2,6,205,0,6,20
   8,8,173
JP 3200 DATA 3,6,205,1,6,240,13,165,205,2
   4,101,207,133,205,144,200,230,206,176,
   196,169,0,133,212,96
XY 3210 DATA 216,104,104,133,204,104,133,
   203,104,104,133,205,160,0,177,203,201,
   33,240,11
RA 3220 DATA 24,105,1,145,203,200,196,205
   ,208,240,96,169,61,240,244,208,242
FH 3230 IF PEEK(195)=136 THEN GOTO 680
```

(continued on page 69)

We Won't Be UNDERSOLD*

* On items marked *"No One Sells This ----- For Less"*. Copy of valid ad required.

15 Day Free Trial • 90 Day Immediate Replacement Policy • Prices Expire 11-30-87

Low Sale Price

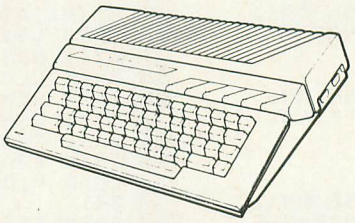
Atari 130XE 152K Computer

$129.95

List $249

(Add $3.00 shipping.*)

While Supplies Last



- 128K RAM - 24K ROM • 256 Colors
- 4 Independent Sound Voices
- 62 Typewriter Style Keys
- 4 Special Function Keys
- 29 Graphic Keys

300 Baud For Atari

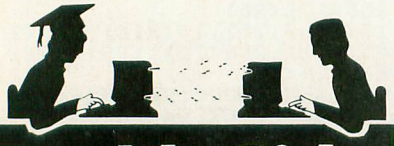
Complete Atari Auto Dial Telecommunications Package

Sale $29.95

(Add $3.00 shipping.*)

List $99

Cartridge Software Included With Modem!!!



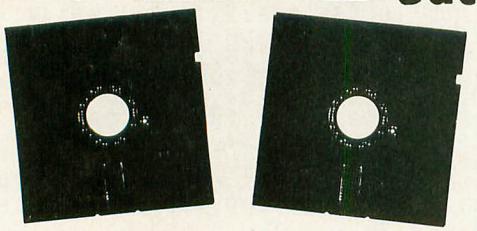
This Complete, Easy to Use Modem Package features...

- Auto Dial - Auto Answer
- Dialing From Keyboard
- X-Modem Protocol
- Smart Terminal Software
- Stores downloaded files on disk
- Uploads text, program or data files
- Downloads text, program or data files
- Plus Much More

Double Sided - Double Density Disk Sale

5¼" Floppy Disks

For As Low As 24¢ each



Each Disk 100% Certificatied (Add $3.00 shipping.*)

Never experience data or program loss during your lifetime!

Free Lifetime Replacement Warranty

Against failure due to faulty materials or workmanship .

Automatic Dust Remover

Constantly removes dust and dirt during operation.

1 Box of 10 ..... $ 3.40 (34¢ each)
(with sleeves)

1 Box of 50 ..... $14.50 (29¢ each)
(with sleeves)

1 Box of 100 ..... $24.00 (24¢ each)
(without sleeves)

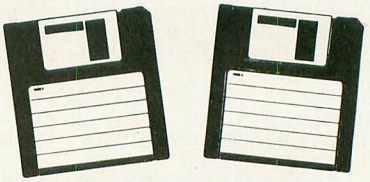
Paper Economy Sleeves..... 50¢ per quantity of 10

Disk Labels..... $3.95 per quantity of 100

3½" Micro Disks

On Sale For Only 99¢ each

(Add $3.00 shipping.*)



For use with all 3½" Disk Drives.

- Compact and easy to handle
- Same tough and reliable qualities of 5¼" floppies other disks in a compact, cassette-like format
- Hard-plastic shell provides maximum media protection and safe handling.

Holds more data than a conventional 5¼" floppy disks

Quantity of 10 ..... $9.90 (99¢ each)

CIRCLE #111 ON READER SERVICE CARD

Call

(312) 382-5050

We Love Our Customers!

Mail

COMPUTER DIRECT

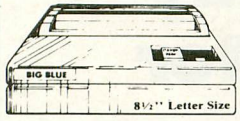
22292 N. Pepper Road
Barrington, IL. 60010

1st In Price, Support, & Warranty

15 Day Free Trial • 90 Day Immediate Replacement Policy • Prices Expire 11-30-87

8½" Letter Size 80 Column Printer

We Liked This Printer So Much,
We Bought Out The Factory!



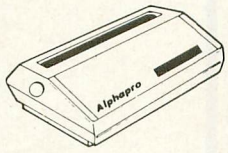
Sale $39.95

*Add $ 7.50 Shipping. List $199

- **Big Blue Dot Matrix Printer**
- **Heat Transfer • Enlarged**
- **Graphics (Apple, Atari, & CBM)**
- **Underline • Upper & Lower Case**
- **True Descenders • Adapters For:**
- **Commodore & Atari . . Sale $19.95**
- **Apple IIc & Laser 128 . Sale $12.95**
- **RS-232 Adapter . . . . . Sale $12.95**
  (Please Specify Male or Female)

Pro 20 Daisy Wheel Printer

Wide Carriage Letter Quality
Daisy Wheel Printer



Sale

Sale $99.95

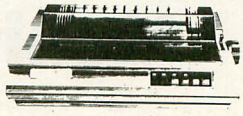
*Add $10.00 Shipping. List $499

- **Daisy Wheel Printer**
- **18 CPS Shannon & 22 CPS AAA Text**
- **Print Wheel (Courier 10) & Cartridge Compatible with Diablo® & Qume®**
- **Friction Feed • 13" Form Width**

Extra Daisy Wheels . . . . . Sale $ 9.95
Parallel Centronics Interface (IBM®PC, Apple®IIc, Laser 128, CBM . **Sale $19.95**

10" Comstar 1000 Printer

With Near Letter Quality
Best Quality In The U.S.A.



Sale $139.95

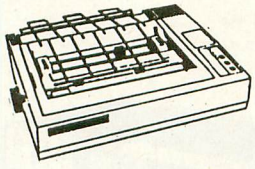
*Add $10.00 Shipping. List $349

- **100 CPS Draft & 20 CPS NLQ**
- **Serial Impact Dot Matrix • Bold**
- **Underline • Condensed • Italics**
- **Pica • Elite • Double Strike**
- **Dot Addressable Graphics**
- **Automatic Paper Loading**
- **Tractor & Friction Feed**
- **Superscript & Subscript**
- **Right & Left Margin Settings**

No One Sells These Printers For Less!

NLQ 180 (180 CPS) Printer

Near Letter Quality
Lifetime Warranty*



Sale $189.95

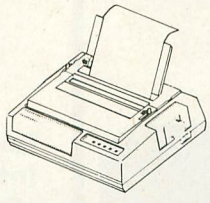
*Add $10.00 Shipping. List $499

- **Near Letter Quality Selectable From Front Panel • 8K Buffer**
- **High Speed Dot Matrix**
- **Letter Quality Modes • Italics**
- **Elite • Pica • Condensed**
- **Graphics • Tractor & Friction Feed**
- **Lifetime Warranty on Print Head***
- **6 Month Immediate Replacement Policy For Printer**

Hi-Speed LV-2010 (200 CPS) Printer

High Speed Printing (200 CPS Draft)
with Crisp Near Letter Quality

Save



Sale $209.95

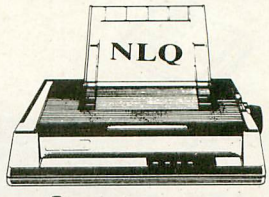
*Add $10.00 Shipping. List $499

- **High Speed 200 CPS Draft**
- **IBM® Compatible • Dot Matrix**
- **Near Letter Quality Printing**
- **Standard Pull Tractor & Automatic Single Sheet Loading**
- **Standard Parallel & Serial Interface Ports**
- **Ultra High Resolution Bit Image Graphics • Continous Underline**

300 CPS Printer

300 CPS Draft - 50 CPS NLQ

Comstar 1300 High Speed Printer
With Color Printing Capabilities



Sale $299.95

*Add $10.00 Shipping. List $599

- **300 CPS Draft - 50 CPS NLQ**
- **Superb Near Letter Quality**
- **Variety Of Characters/Graphics**
- **IBM & Epson Modes • 10K Buffer**
- **Auto Paper Loading & Ejection**
- **Parallel & Serial Interface Ports**
- **Download Character Setting**
- **Front Panel Margin Setting**
- **Optional 7-Color Printing Kit...$99.95**

COMPUTER DIRECT (A Division of **PROTECTO**)
22292 N. Pepper Road, Barrington, IL. 60010
Call (312) 382-5050 To Order
We Love Our Customers!

• Illinois residents add 6½% sales tax. All orders must be in U.S. Dollars. We ship to all points in the U.S., CANADA, PUERTO RICO & APO-FPO. Please call for charges outside continental U.S. or C.O.D. **MAIL ORDERS** enclose cashier check, money order or personal check. Allow 14 days delivery, 2 to 7 for phone orders and 1 day express mail. Prices and availability subject to change without notice. (Monitors only shipped in continental U.S.)

VISA — MASTERCARD — C.O.D.

CIRCLE #111 ON READER SERVICE CARD

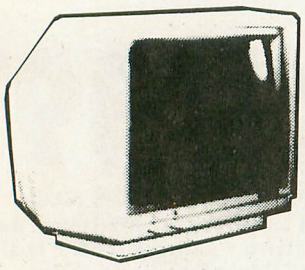
We Won't Be UNDERSOLD

* On items marked "*No One Sells This ----- For Less*". Copy of valid ad required.

15 Day Free Trial • 90 Day Immediate Replacement Policy • Prices Expire 11-30-87

Computer Monitor Sale

12" Hi-Res Monochrome Monitor

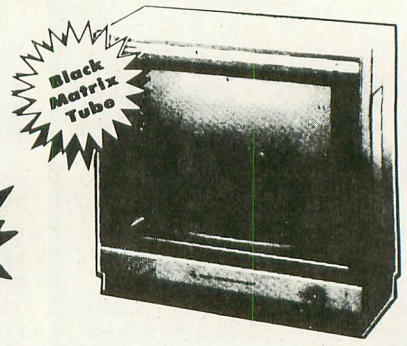


Sale $79.95

(Add $10.00 shipping.*) List $329

80 Column • 1000 Lines At Center • High Resolution • Non-glare Screen

13" Color Display Monitor



Sale $159.95

(Add $14.50 shipping.*) List $329

Anti-glare Screen • Volume Control • Built-in Audio Amplifier & Speaker

Remote Control TV Stereo Tuner

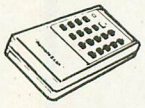
MTS Stereo Remote Control TV

Sale $99.95

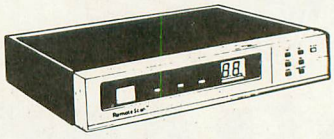
(Add $3.00 shipping.*) List $200

Converts Any Monitor Or TV To True MTS Stereo Sound & Cable Ready Remote Control • 139 VHF/UHF Cable Channels • Sleep Switch • Dbx Noise Reduction • Plus More

Remote Control TV Tuner



Turn Monitor Into Remote Control TV



Sale $79.95

(Add $3.00 shipping.*) List $180

Convert Your Monitor Or Tv Into A Remote Control TV • Direct Access To 139 VHF/UHF/Cable Channels • Sleep Timer • Quartz Frequency Synthesized Tuner • Individual Antenna Connections • And More

Call

(312) 382-5050

We Love Our Customers!

Mail

COMPUTER DIRECT
22292 N. Pepper Road
Barrington, IL. 60010

CIRCLE #111 ON READER SERVICE CARD

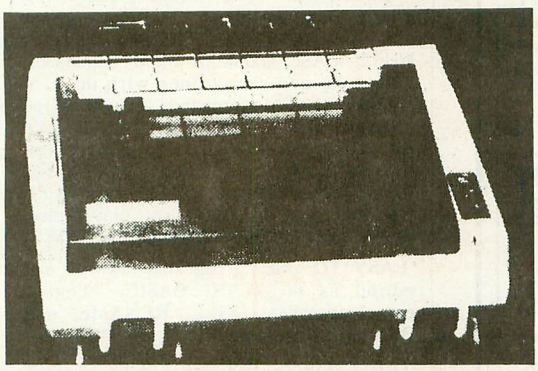
1st In Price, Support, & Warranty

15 Day Free Trial • 90 Day Immediate Replacement Policy • Prices Expire 11-30-87

Star Micronics LV-1215 Printer Sale

15" NLQ Printer Sale

Wide Carriage Business Printer For The Price Of a 10" Carriage Printer!



Price Break Thru

- 120 CPS Print Speed
- Near Letter Quality Print Mode
- IBM Graphics Printer Compatible
- 136 Column Wide Carriage
- Uses Inexpensive Spool Ribbons

No One Sells This Printer For Less!

Sale $199.95

List $499

SPECIFICATIONS

Printing Method
Impact Dot Matrix

Printing Speed
120 Characters per secon (at 10 CPI)

Print Buffer Size
816 Bytes

Paper Feed
10 Lines/second (at 1/5 inch line spacing);
Sprocket or Friction Feed

Character Sets
96 Std. ASCII; 96 NLQ ASCII
83 Special & 50 Block Graphic Characters

Character Matrix
9 x 9 Standard Character;
12 x 6 Block Graphics;
8 dot x 60 dot/in. : 8 dot x 120 dot/in. :
8 dot x 240 dot/in. Bit Image Modes.

Line Spacing
1/6, 1/8 inch or 7/72 inch standard:
n/72 or n/216 inch programmable

Characters Per Line
Pica — 136 CPL; Condensed — 233 CPL
Pica Expanded — 68 CPL:
Condensed Expanded — 116 CPL

Port
Centronics Parallel Port

Printing Width
Single Sheets — 5.5 inch to 14.5 inch:
Continuous Paper — 4 inch to 15.5 inch

Copies
Original Plus 2 Copies

Dimensions
H-5.80" W-15.2" D-12.4"

Replacement Ribbons
Black . . . . . Sale $4.95

Parallel Interfaces

IBM — $24.95 Apple — $44.95 Laser 128 — $19.95 Commodore — $29.95 Atari — $39.95

COMPUTER DIRECT (A Division of **PROTECTO**)
22292 N. Pepper Road, Barrington, IL. 60010
Call (312) 382-5050 To Order
We Love Our Customers!

* Illinois residents add 6½% sales tax. All orders must be in U.S. Dollars. We ship to all points in the U.S., CANADA, PUERTO RICO & APO-FPO. Please call for charges outside continental U.S. or C.O.D. **MAIL ORDERS** enclose cashier check, money order or personal check. Allow 14 days delivery, 2 to 7 for phone orders and 1 day express mail. Prices and availability subject to change without notice. (Monitors only shipped in continental U.S.)

VISA — MASTERCARD — C.O.D.

CIRCLE #111 ON READER SERVICE CARD

CIRCLE #113 ON READER SERVICE CARD

CIRCLE #124 ON READER SERVICE CARD

CIRCLE #114 ON READER SERVICE CARD

CIRCLE #130 ON READER SERVICE CARD

CIRCLE #115 ON READER SERVICE CARD

Bits & Pieces

Atari Zucchini.

by Lee S. Brilliant, M.D.

If you've ever had a garden, you probably grew zucchini—and you know how hard it is to get rid of the stuff! You *never* plant more than one plant. Even then, it will produce more green stuff than the U.S. mint. When you're ankle deep in it, you try to unload some of the veggies on the unsuspecting neighbors. So you make a trip to your neighbor's house with a twenty-pound beauty—only to find your neighbor waiting at his door with a zucchini of his own for you. Soon you're innundated with zucchini; if only there was another use for this prolific vegetable.

In 1979, Atari began selling its first home computer, the 400. Shortly thereafter, its bigger brother the 800 (remember when they came with a whole 16K?) appeared. Then they produced the ill-fated 1200XL, followed by the 600XL/800XLs. Finally, they introduced the 65XE and 130XE. What does that mean? It means most Atari users have upgraded. I paid $399 for my 800, $199 for my 800XL, and today you can get a 130XE for $129. But what happens to all those old 400, 800, 600XL, 800XL, 1200XL computers, and all the RAM expanders that were produced for them?

Sure, a lot are in the hands of users right this minute. Others are sitting around gathering dust. You can buy them used for peanuts (a legume, not a vegetable). Of course, selling them is like trying to give away zucchini—everyone has one by now.

What makes a peripheral.

In writing this series, I've always tried to illustrate theory with *simple* construction projects. When working on the serial port sections, I was trying to devise an easy-to-build peripheral that could communicate with the Atari. As I researched, I found it was technically much too difficult for the average reader. Then I saw an article in an electronics magazine on how to build a printer buffer. I never built it, because it would have cost as much to build as to buy one already put together. The memory chips alone cost $32, and there were custom ROM chips you had to buy from the author—for a tidy profit.

Now wait a minute! ROM chips in a memory buffer? It turns out that a buffer is nothing more than a small, dedicated microcomputer. This one used a Z-80 processor, 64K of RAM and a custom ROM, some address decoding, a serial input port, an 8-bit parallel output port, and so on. Add a keyboard, a video circuit and some sound, and you would have a full-fledged computer. I was surprised to find out the venerable 810 disk drive has a 6507 heart—that's a close cousin to the 6502—running the show in your Atari.

Putting all this together, I found out that one Atari could be used to communicate with another through the serial port. It only takes some modified cables and some software. If you're willing to follow this somewhat difficult series of three articles, you'll understand how the Atari communicates through the serial ports and how interrupts are used; then you

can build your own functional printer interface/buffer from a surplus 400/800 computer—for around $40.

What shall we call this creation? Why, the Atari Zucchini, what else!

Project outline.

A thorough description of how the Atari communicates serially would take several chapters in a book, but I'll try to condense it into three articles. First, we'll consider the hardware, then the Serial I/O (SIO), then finally, how to grow your Zucchini.

Let's start with the 6520 PIA and POKEY, and see how they handle the interrupt system—and how this relates to SIO. Next month, we'll check out the operating system, to see how serial I/O is accomplished. Our third episode will show you the software and hardware involved in turning an Atari into a printer interface/buffer.

The 6502 CPU has two pins which, when set to *0*, will cause it to stop operation and execute a special program called an "interrupt." The two pins are labeled *NMI* and *IRQ*, which stand for "Non-Maskable Interrupt" and "Interrupt ReQuest," respectively. When these pins are set, the processor will finish its current instruction, save its place in the current program onto the stack, and jump to the INTERRUPT PROCESSOR program. The processor will determine the source of the interrupt request and be routed to the specific interrupt handler for that source.

For example, pressing a key causes an IRQ interrupt, so the processor determines that the keyboard is requesting service, and then routes to the keyboard handler. The handler should end in a RTI instruction (NMI) or PLA,RTI (IRQ). The locations of the processors are stored in the last few bytes of memory; in this case, the Operating System (OS) ROM. The IRQ vector is at $FFFE and $FFFF, while the NMI vector is at $FFFA and $FFFB. NMI and IRQ interrupts differ in two major respects: (1) the CPU *must* respond to a signal on the NMI pin, but can be programmed to ignore one on the IRQ pin; and (2) IRQ interrupts affect the processor registers, while NMIs do not. IRQs set the I bit of the Process Status Register, which can be programmed to ignore the interrupt—that is to say, the interrupt can be "masked." To do this, you use the machine SEI and CLI commands. The CPU cannot be programmed to ignore an NMI, hence the "Non-Maskable" appellation.

The *only* NMIs in Atari are Display List, Vertical Blank and Reset, so these are generated by ANTIC. The interrupt processors are in ROM and cannot be altered, but the IRQ handler first travels through a RAM location (called a "vector") which can be changed and thus rerouted to a processor of your own design. The NMI processor does not have a vector in RAM which can be changed, so no modifications to the NMI handler are possible. (This does not apply to an XL or XE, where ROM can be replaced by RAM and the OS moved. There will be an addendum on this at end of third article.)

While an NMI is not maskable at the CPU level, it *is* maskable at the ANTIC level. This is done through a special pair of registers residing in ANTIC. NMIST (NMI STatus) is a Read-Only register at location 54287 ($D40F). Whenever an NMI is requested by ANTIC, this register will show which kind of interrupt is wanted. So if ANTIC wants to generate a DLI (called by adding a *128* to any line command in the Display List), it sets Bit 7 of NMIST.

BIT 7	BIT 6	BIT 5	4,3,2,1,0
Display List Interrupt	Vertical Blank Interrupt	Reset	Unused

Before ANTIC will drop its NMI line for a requesting interrupt, it checks the Write-Only register NMIEN ($D40E NMI ENable), to see if your program will allow this interrupt to occur. To ENABLE DLIs or VBIs, you must write a *1* to the proper bit of NMIEN. ANTIC then checks NMIEN against NMIST before allowing the NMI request to be passed along to the CPU. Since NMIEN is Write Only, you may want to keep a shadow register in RAM which contains NMIEN's current value.

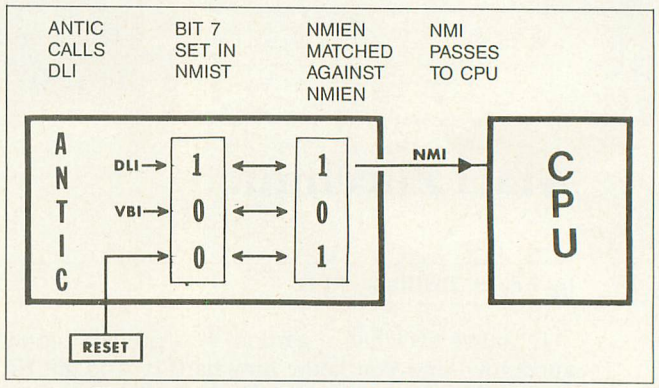


Figure 1. — Outline of interrupt process.

The CPU will finish its current command, read locations $FFFA and FFFB, and jump to the address stored in these bytes. The address held there contains the location in memory of the NMI processor. The processor reads NMIST and routes the program flow to either the VBLANK routine or the DLI program, or initiates the Reset chain. DLI and VBI programs next route through a vector address located in RAM, then back to the handler in ROM. The handlers can be intercepted at these vectors so you can modify or replace them, but the Reset chain has no vectoring until you reach the point where DOS is called for initialization. Bit 5 of NMIEN is always set so you cannot mask a RESET.

IRQ interrupts work the same way. POKEY contains two registers called IRQST (IRQ STatus) at address $D20E, and IRQEN (IRQ ENable), also at $D20E, that work just like NMIST and NMIEN, in that an IRQ is actually generated—but only if the bits match. If the proper bit in IRQEN was not set, then POKEY will not create the IRQ. The bit assignment of IRQST are as follows:

Bit	Decimal	Function
7	128	BREAK key enable
6	64	KEYBOARD interrupt enable
5	32	Serial data input ready (VSERIN)
4	16	Serial data output request (VSEROR)
3	8	Serial data output complete (VSEROC)
2	4	POKEY timer 4 interrupt enable
1	2	POKEY timer 2
0	1	POKEY timer 1

So, to disable the BREAK key, write a *0* to bit 7 in IRQEN,

plus its shadow. When you press the BREAK key, no interrupt is generated, because it's blocked at the POKEY level.

All of these interrupts (except the BREAK key in the 400/800) have RAM vectors which can be altered to point to your own routines. The TIMER vectors point to a dummy ROM routine, consisting of PLA and RTI, while the BREAK key has a routine to set the BREAK KEY FLAG at *17 ($11)*.

Note that three of the POKEY interrupts are involved with serial I/O. We'll look at these later. The timers are described in the Atari tech manuals and are not used in Zucchini. However, we will disable the BREAK key and alter the keyboard interrupt. IRQEN is a Write-Only register; once written to, you cannot retrieve that data. That's why there's a shadow register at location 16, called POKMSK (POKey MaSK), which must always be changed whenever IRQEN is varied, to keep track of which interrupts are enabled.

Once the IRQ is validated and the request generated, the main processor JMPs to the interrupt processor, which tests the various bits of IRQST—to determine the interrupt actually calling—then does a JSR to the appropriate interrupt routine. Since all IRQs have a RAM vector which can be changed, you can reroute any IRQ to your own routine in RAM, instead of to the built-in one in ROM. Try this program:

```
10 DATA 238,198,2,238,298,2,76
20 FOR S=1536 TO 1542:READ D:POKE S,D:
NEXT S
30 POKE 1543,PEEK(520):POKE 1544,PEEK(
521)
40 POKE 520,0:POKE 521,6
```

This program intercepts the keyboard vector to cause a change of screen color whenever a key is pressed, then goes on to the normal key process. The vector for the keyboard handler is in locations 520/521, but, if you use the same program with the global IRQ handler vector at 534/535, then the screen will change with any interrupt—such as serial I/O. You can use this simple system to create displays that change during I/O; this will keep you occupied during long loads. You can disable all POKEY interrupts with POKE 53774,0:POKE 16,0. You'll need the RESET key to recover.

By now, you should be familiar with the I/O functions and how to program the ports for input or output. In addition to the port pins, there are four other IRQs associated with the 6520 PIA, besides the POKEY IRQs. While the port pins go to the joystick plugs, there are four additional lines which go to the serial port. These four pins can generate interrupt requests directly, bypassing POKEY, and two of them are programmable for output.

PORTA has two interrupts, PA1 and PA2. Similarly, PORTB has PB1 and PB2. The functions of these interrupt bits are controlled by PACTL (Port A ConTroL) and PBCTL according to this pattern:

PACTL BITS

7	6	5	4	3	2	1	0
PA1 status	PA2 status	PA2 Interrupt control			Data direct register	PA1 interrupt control	

The same pattern applies to PBCTL($D303). You should recognize that bit 2 is the one you alter when reprogramming the joystick pins.

When the interrupt control bits are set for input and enabled, an IRQ is generated whenever a trigger pulse comes in over an interrupt line. The last 2 bits of the port control register show which interrupt line was triggered. The Port Control bit patterns to control the interrupts are as follows:

PA1/PB1

BIT 1	BIT 0
0=Responds to –transition 1=Responds to + transition	0=Disable interrupt 1=Enable interrupt

PA2/PB2

BIT 5	BIT 4	BIT 3
0=Interrupt from PA2/PB2 1=Output to PA2/PB2	0=Responds to – transition 1=Responds to + transition 0=* 1=Direct output	0=Disable 1=Enable * 0=Logic 0 1=Logic 1

* =Too complex and not pertinent.

Interrupt controls.

PB1 is connected to Pin 13 on the serial plug. If you trigger this line, an IRQ is created and bit 7 of PBCTL is set to *1*. These bits are tested by the IRQ Processor, which then vectors to the interrupt routine. The flag bits are reset by a READ operation to the control registers. Note that the OS Interrupt Processor does not recognize PA2/PB2 interrupts, only PA1 and PB1 because Atari only uses these lines for output. You would have to create your own handler for these.

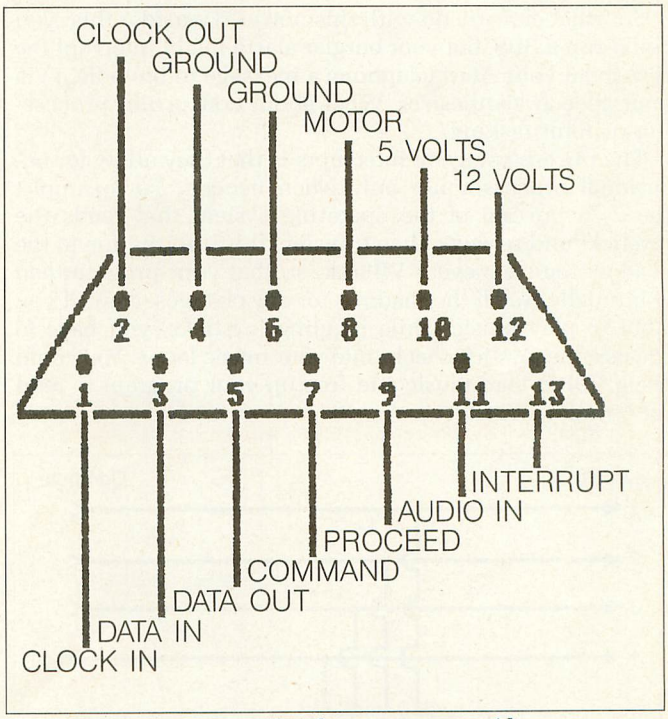


Figure 2. — Serial port, computer side.

Serial Plug.

The PA2 and PB2 interrupts can be reprogrammed as outputs. In the Atari, PA2 controls the cassette recorder, while PB2 is involved in SIO. PBCTL and PACTL usually hold the number *60*, which sets bits 2-5 to *1* or enables PA2/PB2 for output and sets the lines high. PA1 and PB1 are similarly

available on the serial bus, have vectors in RAM from $0202 to $0205, are not controlled by IRQEN, and are unused as far as I know. If you have the fortitude, you can try the following program to demonstrate.

```
10 DATA 238,198,2,238,198,2,104,64
20 FOR S=0 TO 7:READ D:POKE 1536+S,D:N
EXT S
30 POKE 516,0:517,0
40 POKE 54019,61
```

Line 40 sets PBCTL for transition and enables PB1, while Line 30 steals the PB1 interrupt vector to point to our routine. Now, remove the serial plug and attach a small insulated alligator clip to pin 13 of the port and another to pin 4 or 6. Touching the free ends together causes the screen to change color. Note that interrupt lines respond only to *transitions* in logic state, not the logic state itself. POKEing PBCTL with 63 causes the screen to change colors when the connection between the clip leads is opened.

Standard 6520 design puts ports and interrupts together to create two 8-bit I/O ports, each with 8 data bits, a strobe and a busy line. In the Commodore, all these lines are available in the user's port. In the Atari, the interrupt lines are available only on the serial data port, while the data lines are at the joysticks. (Apple doesn't even have a PIA!) We can use them together to create our own devices. All interrupt routines *must* end in PLA, RTI instructions, or the computer will get lost.

So, what can you do with this power? Consider this: you could run a BBS, but your burglar alarm might interrupt the BBS to let your Atari telephone a message to the police via your speech synthesizer. What about concurrent processing or multitasking?

The big advantage of interrupts is that they allow for occasional processor use *only when needed*. For example, there's a portion of the operating system that reads the joysticks and triggers, then transfers this information to the shadow registers every VBlank, so that your program can continually watch the shadows for any changes. Now, if you want to play music while juggling joysticks, you have to squeeze the joystick checks into your music loops. You could use a VBI to load music and free up your program to read the joysticks only.

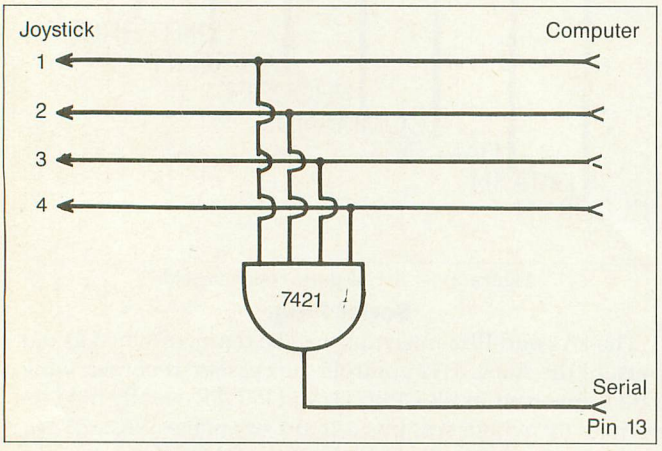


Figure 3. — Joystick port inputs.

Many programs use VBI routines to update player positions or make other changes, leaving main program lines to do other things. A better way, however, is to use an interrupt like the keyboard does. The OS doesn't spend its time scanning the keyboard, waiting for a keypress. Instead, POKEY scans the keyboard and signals the CPU to process a keypress via an IRQ. So a slightly different approach to joystick port inputs might look like Figure 3.

With the circuit in Figure 4, whenever any switch is pressed, it will generate an IRQ which can read the joystick and determine which actions to take. You could use this idea with an encoder chip to give multiple inputs:

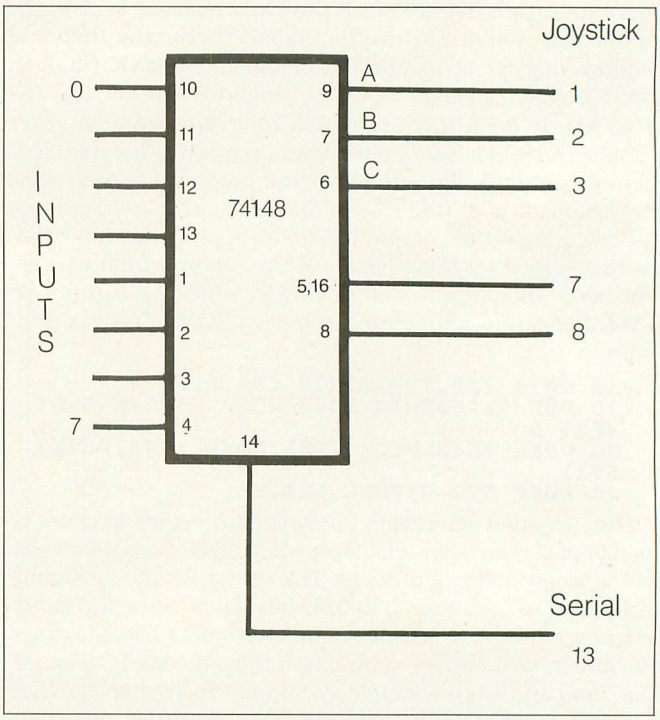


Figure 4. — Another approach using an encoder chip.

For the technically minded, this circuit works as follows:

When all of the input and reference lines to the comparator are equal, the equals output is high.

If any of the input lines changes state, the equals line drops, generating a pulse for the interrupt line. The resistor/capacitor combinations create a short time delay between a change of the input and the catch-up of the reference lead.

When the reference lead again equals the input, the equals line returns to high.

Whenever the mouse moves, it now generates an interrupt, so the CPU can read the position of the mouse.

Remember the 74LS14 will "invert" the value of the photo transistor.

The ultimate extension of this interrupt/joystick combination would be a remote keyboard run through the joysticks. Perhaps another article. . .

Computer designers often face a problem deciding if something is better done in the hardware or software. Such a dilemma is illustrated by "Mouser" (**ANALOG Comput-**

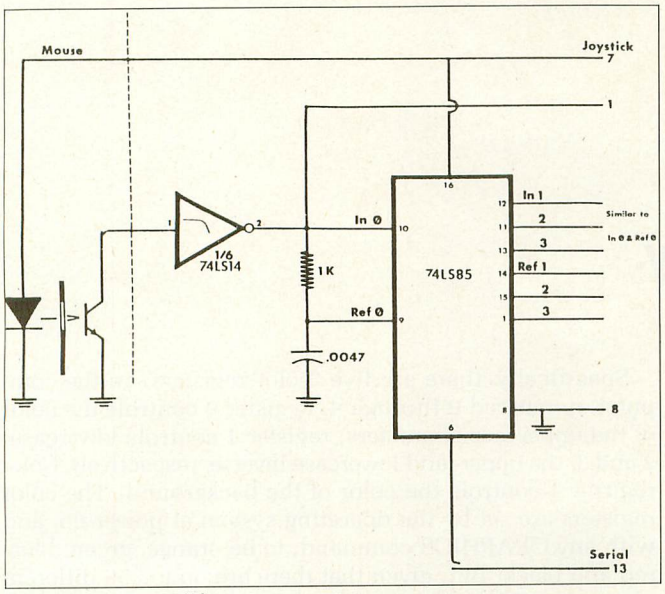


Figure 5. — Mouser circuit.

ing issue 40). The author had to use several software manipulations to obtain a high sample rate of the joystick ports to watch Mouser's position. While it simplifies the hardware, this consumes a lot of processor overhead and makes integration with other programs difficult because of the extensive DLI and VBI routines.

This simple circuit in Figure 5 generates a short pulse whenever the logic state of Mouser's sensors changes. This pulse in turn triggers the interrupt line on the serial port, so your interrupt routines could read the mouse position on the joystick port. No VBI or DLI changes are needed, making it easier to integrate into other software.

Next month we'll delve further into POKEY, the operating system, and how serial I/O is accomplished. Until then, keep on plugging—or is that unplugging? Oh, and don't eat any used zucchini!

Get your point across—visually.

by Jeffrey A. Summers, M.D.

Need to convey information visually to a lot of people? Is your local user's group having a table at a local computer show? Do you then need something to show off the Atari machines, while also giving information about the group? Are you giving a talk that needs to be spiced up a bit with a slideshow, including graphs and pictures?

If you're in any of these situations, **Presentation Graphics** can help. The program PRESENT is written for Atari 8-bits, compatible with models of at least 48K memory.

To use the program, type it in using "BASIC Editor II" (issue 47) to check your work. Listing 2 will create a file named D:LINES.LST, that contains program Lines 12000 to 12100. Run the program in Listing 2, then use ENTER to add the lines to the main program, Listing 1.

Also, though the version here works with all models of Atari 8-bits, if you have an XL or XE, you may delete Lines 5411, 10160 and 10170, and change Line 5410 to GRAPHICS 31 for slightly faster and cleaner picture loading. Save two copies, then run the program. After the title screen appears, the program offers the main menu.

How to use it.

A little digression is in order here, for optimal use of **Presentation Graphics**. The computer selects the size of type on the screen and the number of colors available by the "graphics mode" chosen. These modes are numbered 0-15, though only 0-8 are available on certain machines (CTIA-equipped 400s and 800s).

The normal type you see on the screen when the computer is turned on is mode 0. Mode 1 is twice as wide, and mode 2 is both twice as wide and twice as high. The letters in modes 1 and 2 may be any of four different colors.

Modes 0 through 2 are the most useful for this program. Modes 3 through 8 are all "map modes." In other words, they do not display text. Instead, they display colored dots of varying sizes. Any BASIC reference will give you the complete list of dot sizes and numbers of colors available.

The other bit of information (pun intended) is about the manner in which the color of the text is chosen in modes 1 and 2. If the text is uppercase, it is one color; lowercase is a second; "inverse uppercase" a third; and "inverse lowercase," a fourth. The letters themselves will always be uppercase (capitals); only the color changes by changing the letter to lowercase or using the inverse (Atari logo) key.

Specifically, there are five "color registers" in the computer, numbered 0 through 4. Register 0 controls the color of the uppercase characters; register 1 controls lowercase; 2 and 3, the upper- and lowercase inverse, respectively. Color register 4 controls the color of the background. The color registers are set by the operating system at power-up, and with any GRAPHICS command, to be orange, green, blue, red and black. But, given that there are up to 256 different colors available to choose from for each color register, a lot of variation is possible.

The colors fall into sixteen different classes, or hues, each with sixteen different lightnesses, or luminances, numbered 0 through 15. Some modes on some models may ignore odd-numbered luminances, displaying the next lowest even luminance. To find the value for the color register that corresponds to the color you wish, take the number of the hue, multiply by *16* and add the luminance. The hue values may be found in Table 1, below. For example, the value for a dark red would be *4* (for red) * *16+2* (a low luminance value), or *66*.

Hue number	Color
0	Black/White
1	Rust
2	Red/Orange
3	Dark Orange
4	Red
5	Dark Lavender
6	Cobalt Blue
7	Ultramarine Blue
8	Medium Blue
9	Dark Blue
10	Blue/Gray
11	Olive Green
12	Medium Green
13	Dark Green
14	Orange/Green
15	Orange

Table 1. — Hue values.

If that all left you in the dust with your head reeling, don't despair. Experimenting with the program is a great way to learn about these things, so let's get started.

Run the program and, when the main menu is displayed, press 3 to "Create Screen." You needn't press the RETURN key.

The program displays "Mode lines defined so far," followed by a blank line (you haven't defined any yet), and

Presentation Graphics

then a prompt for you to enter the mode number you desire. This number is one of the graphic mode numbers defined earlier. For demonstration purposes, type a *2* for a double-high, double-wide line, followed by a RETURN.

The program now asks for the text to be displayed on that line. Again, for demonstration purposes, hit a RETURN. A *2* is now displayed under the "Mode lines defined so far." The next line we enter will also be a *2*. If you're not changing modes, you can simply press RETURN when asked for the next mode number. Do so, then type *ANALOG* for the text.

Now, try mode 1 and put PrEsEnTaTiOn for the text. Another 1 and GrApHiCs will be all we'll be entering. Hit RETURN for the mode and RETURN for the text over and over, until the program returns you to the main menu. (The program asks for lines until you've defined enough mode lines to fill the screen.)

Let's look at what we've created. Choose option 6, "Display Screen." Pressing any key while the screen you've created is displayed returns you to the main menu. You should see an off-center, two-color message. Let's spice it up a bit.

Touch any key. When the main menu is displayed, choose option 4, "Edit Screen." The mode lines and text for those lines is displayed. If the line contains more characters than can fit on the line, an ellipsis mark (. . .) appears at the end, implying that there's more text to that line. You're asked whether you wish to change any text, the color of the text, or if you're done (exit). Press a *T* for Text, and you're asked which line you wish to change.

Let's change the first line on which we have text, Line 2. Type the *2*, followed by a RETURN. The old text is displayed, and you're asked for the new text. Type the same message, only eight characters further to the right, for better centering. Center the other lines, as well, and add a few inverse upper and inverse lower characters, to display those letters in the other colors. Press *E* when you're done to return to the main menu, then display the results with option 6. Like it? If not, edit again until you do.

If your screen contains more than sixteen mode lines, it is broken up into "pages." Each page is up to sixteen lines long, and you may move from page to page using the N (Next) and P (Prior) commands.

Now try a few different colors, by choosing C from the edit menu. By pressing C in edit mode, you're opting to change colors. The current values for each of the color registers is displayed, then you're asked which of the registers you wish to change. For now, choose register 4. Remember that this is the register that controls the background color.

The current value is *0*, for black. Try substituting the value for dark red, *66*. The program then asks if you want to change another register value. For now, answer no to return to the edit menu. Press *E* to exit from edit, then *6* at the main menu to display the screen—and voilà! We now have a red background for our display.

To save your screen, choose option 2 from the main menu. You're asked for the filename under which you want the screen saved. Type the name of the file completely, using the usual format of DEVICE:FILENAME.EXT. I find it convenient to name the files after the proposed application, and use the extension to hold the number corresponding to the order in which I'm planning to display the screens. For example, our local users' group is called ACORN, and the first in a series of screens I created for a local computer show was D:ACORN.1; the second, D:ACORN.2; and so on.

If the save is successful, you're returned to the main menu. If not, a message is displayed and, again, you're returned to the menu. Later, you can reload the screen by using main menu option 1, "Load Screen," for editing and display. If you can't remember the name of the file you saved the screen under, you can always get a directory of the disk in drive 1 by using option 7. The program won't let you load an incompatible file, so don't worry about mistakes.

Choosing your options.

That leaves presentation sequence options as the last topic to be covered. These options are very similar to the others. Option 5, or "Create a Presentation Sequence," allows you to enter a series of filenames containing screens to be displayed.

You can choose to have as many as twenty or as few as one file displayed. The files may be files created with this program, files created with MicroIllustrator or compatible programs, or B/Graph image files. This allows the conveyance of text information, graphic pictures, and business-style graphs and pie charts in a sequence.

You are offered the options of "Cued" or "Automatic" advancement, much like those on a slide projector.

If you choose Cued, touching any key will load the next screen in the sequence. If you choose Automatic, you're

asked for the delay before the program is to load the next slide. The number entered will be approximately the number of seconds before the next slide is shown, and decimals are allowed.

You're then asked if you wish the sequence repeated endlessly. If this is a presentation at a local computer show, you probably will want it to repeat over and over, so type *Y* for Yes. If you're presenting a series of screens as a "slideshow" for a business meeting, you probably don't want the sequence repeated, so type *N* for No. The first screen is then loaded, and the sequence is underway.

Option 8, or "Run a Prewritten Sequence," allows you to prepare a sequence ahead of time. Using most word processors, text editors or DOS, you may create a file containing a simple list of files to load. On choosing this option, you're asked for the name of the file containing the list. After this, the prompts regarding Cued or Automatic, time between loads, and whether to repeat endlessly are seen. The program then proceeds through your list.

Remember that the list may contain only up to twenty filenames. If your presentation requires more than that, you'll need to set up more than one sequence. You'll probably be running low on disk space by the time you hit twenty filenames, in any case, so this isn't much of a restriction.

In general . . .

Now for some tips on using **Presentation Graphics**. First, when using the sequence options 5 and 8 from the main menu, any illegal file will be simply passed over without an error message. This is to keep things moving smoothly, should it happen to be a presentation to the boss, and you messed up typing the name of a file.

This also means that it's a good idea, for critical sequences, to create them "off line." In other words, write the list of files with a word processor or the like, and run the sequence through once to make sure everything goes smoothly.

Next, if you plan to use the program for an endless sequence—all day long—a lot of wear and tear can be put on your disk drive. I'd suggest using a 130XE or other RAMdisk-capable machine, and putting the screens on the RAMdisk. The load speed for the pictures will be slightly faster, and there won't be any drive wear.

Third, don't be afraid to use one of the "map mode" graphic mode lines in your screens. A mode 3 line, where the "text" is a series of semicolons, makes a nice multicolor border. A mode 8 line with no text makes a thin line straight across the screen in the color of register 2. I usually use these in pairs.

And, finally, experiment to get the most out of **Presentation Graphics**.

Jeffrey A. Summers, a practicing Internist, received his B.S. in math prior to getting his M.D., and took a strong interest in computers in college. A four-year Atari owner, his interests lie in languages, application programming and medical usage. He teaches beginning BASIC at his local Atari users' group, ACORN, in Rochester, New York.

The two-letter checksum code preceding the line numbers here is *not* a part of the BASIC program. For further information, see the "BASIC Editor II," in issue 47.

Listing 1.
BASIC listing.

```
GH 10 REM PRESENT
TH 20 REM PRESENTATION GRAPHICS PROGRAM
BV 30 REM BY JEFFREY A SUMMERS
BB 40 REM
BC 50 REM
GU 99 GOTO 10000
KH 110 SCR$(LEN(SCREEN$))=" "
CZ 120 D=USR(ADR(CNV$),ADR(SCREEN$),ADR(S
   CR$),LEN(SCREEN$))
UT 170 DLI(C1)=112:DLI(C2)=112:DLI(C3)=11
   2:DLI(4)=64+ANTIC(DLIST(C1)):DLI(C6)=I
   NT(ADR(SCR$)/256)
MI 180 DLI(C5)=ADR(SCR$)-256*DLI(C6)
OL 190 FOR CNT=C2 TO DLEN:DLI(C5+CNT)=ANT
   IC(DLIST(CNT)):NEXT CNT
HX 200 DLI(CNT+C5)=65:DLI(CNT+C6)=0:DLI(C
   NT+7)=C6
AH 220 POKE 559,0:FOR PADR=C1 TO CNT+7:PO
   KE 1535+PADR,DLI(PADR):NEXT PADR
VJ 230 POKE 560,0:POKE 561,C6:POKE 559,34
PB 250 FOR CNT=C1 TO C5:POKE 707+CNT,COLR
   (CNT):NEXT CNT
ZK 260 RETURN
NL 500 ? LINE$(C1,NCHARS(MODE));:FOR Q=C1
    TO NCHARS(MODE):? "↑";:NEXT Q
OK 505 IF NCHARS(MODE)>38 THEN PRINT "↑";
TX 510 MARKER=C1
DZ 520 GET #C1,X:IF X=126 THEN MARKER=MAR
   KER-C1:IF MARKER=0 THEN MARKER=C1
EH 530 IF X=126 THEN GOTO 550
NE 535 IF X=155 THEN 560
GI 540 TEXT$(MARKER)=CHR$(X)
CB 545 MARKER=MARKER+C1:IF MARKER>NCHARS(
   MODE)+C1 THEN MARKER=MARKER-C1:PRINT "
   ↑";
FT 550 PUT #16,X:GOTO 520
QA 560 TEXT$(MARKER,80)=SP$
MN 570 START=C1:IF NL=C1 THEN 600
EY 580 FOR Q=C1 TO NL-C1:START=START+NCHA
   RS(DLIST(Q)):NEXT Q
YE 600 ENN=START+NCHARS(DLIST(NL))-C1
SR 610 SCREEN$(START,ENN)=TEXT$
ZC 620 RETURN
ZY 700 POKE 694,0:POKE 702,64:RETURN
AJ 710 CLOSE #C1:OPEN #C1,C4,0,"K:":RETUR
   N
TP 720 POKE 16,112:POKE 53774,112:RETURN
OS 730 PRINT CHR$(125);:POKE ADL+C3,PEEK(
   ADL+C3)-C5:POKE ADL+C6,C2:POKE ADL+26,
   C2:POKE ADL+27,C2:RETURN
GZ 1000 GOSUB 730:GRAPHICS 0:SETCOLOR C2,
   C1,C2:GOSUB 720
DD 1005 ADL=PEEK(560)+256*PEEK(561):POKE
   ADL+C3,PEEK(ADL+C3)+C5
TJ 1006 POKE ADL+C6,7
JH 1010 PRINT "      load                F
   ILE"
BZ 1011 POSITION C1,11:PRINT "Enter filen
   ame to load: ";
VN 1020 INPUT #16,FILENAME$
RQ 1022 TRAP 1025:IF FILENAME$(C2,C2)=":"
    OR FILENAME$(C3,C3)=":" THEN 1025
HC 1023 FN$=FILENAME$:FILENAME$="D:":FILE
   NAME$(C3)=FN$
WT 1025 CLOSE #C1
WC 1030 TRAP 1120:OPEN #C1,C4,0,FILENAME$
PA 1035 TRAP 1500:INPUT #C1,FILENAME$:IF
```

```
   FILENAME$<>"\SUMMERS\PRESENT\" THEN 15
   00
BC 1040 FOR I=C1 TO C5:INPUT #C1,C:COLR(I
   )=C:NEXT I
NN 1050 INPUT #C1,DLEN:FOR I=C1 TO DLEN:I
   NPUT #C1,C:DLIST(I)=C:NEXT I
QU 1060 SCREEN$(2048)=" "
TN 1070 HI=INT(ADR(SCREEN$)/256):LO=ADR(S
   CREEN$)-HI*256:POKE 850,7:POKE 852,LO:
   POKE 853,HI
UM 1071 POKE 856,0:POKE 857,8:D=USR(ADR(S
   HIN$),16):SCL=PEEK(856)+256*PEEK(857):
   SCREEN$(SCL)=""
QT 1080 SCRFLG=C1:GOSUB 710
QN 1090 PRINT "↰     load
   FILE";:POSITION 5,11:PRINT "Display sc
   reen(Y/N): ";:GET #C1,X
PA 1100 IF X<>89 THEN GOTO 11000
WJ 1101 PRINT CHR$(125);:POSITION 5,11:PR
   INT "Press any key for menu"
GU 1105 FOR I=C1 TO 200:NEXT I:GOSUB 110:
   GET #C1,X
OX 1110 GOTO 11000
AF 1120 PRINT "Error in file input/output
   .":PRINT "Please enter the filename us
   ing the"
HK 1130 PRINT "format (dev):(filename).(e
   xt)"
ZD 1135 GOSUB 710
FG 1140 PRINT "Also, check to be sure dev
   ice is":PRINT "turned on and able send
    or receive"
YS 1141 PRINT "the data."
HM 1145 PRINT :CLOSE #C5:GOTO 1520
IT 1500 CLOSE #C1:PRINT "File is not comp
   atable with":PRINT "this program.  Ple
   ase only load files"
ST 1510 PRINT "that were created with thi
   s program.":PRINT "B/Graph and Microil
   lustrator files"
VO 1515 PRINT "cannot be edited and must
   be loaded":PRINT "with the sequencing
   commands."
XI 1520 PRINT "Press any key for menu"
GM 1530 GOSUB 710:GET #C1,X:GOTO 11000
HQ 2000 IF SCRFLG=0 THEN PRINT :PRINT "No
    screen in memory":FOR I=C1 TO 500:NEX
   T I:GOTO 11000
JA 2003 GOSUB 730:GRAPHICS 0:SETCOLOR C2,
   C2,C2:GOSUB 720
ZE 2005 ADL=PEEK(560)+256*PEEK(561):POKE
   ADL+C3,PEEK(ADL+C3)+C5:POKE ADL+C6,7
VS 2006 PRINT "     save                 F
   ILE"
DN 2010 POSITION C1,11:PRINT "Enter filen
   ame for save: ";:INPUT #16,FILENAME$
GZ 2022 TRAP 1120:IF FILENAME$(C2,C2)=":"
    OR FILENAME$(C3,C3)=":" THEN 2030
HD 2023 FN$=FILENAME$:FILENAME$="D:":FILE
   NAME$(C3)=FN$
MH 2030 TRAP 1120:OPEN #C5,8,0,FILENAME$
XG 2035 PRINT #C5;"\SUMMERS\PRESENT\"
BD 2040 FOR I=C1 TO C5:PRINT #C5;COLR(I):
   NEXT I
GL 2050 PRINT #C5;DLEN:FOR I=C1 TO DLEN:P
   RINT #C5;DLIST(I):NEXT I
VA 2060 PRINT #C5;SCREEN$:CLOSE #C5
PO 2070 GOTO 11000
ET 3000 GOSUB 730:GRAPHICS 0:GOSUB 720
MF 3010 SETCOLOR C2,C3,C2:IF SCRFLG=0 THE
   N 3030
IX 3020 POSITION C5,11:PRINT "There is a
   screen in memory":POSITION C5,12
KL 3021 PRINT "Do you wish to save it fir
   st?";:GET #C1,X:IF X=89 THEN 2010
EA 3022 SCRFLG=0:PRINT CHR$(125);
TN 3030 CURLIN=0:NL=0:DLEN=C1
YY 3031 ADL=PEEK(560)+256*PEEK(561):POKE
   ADL+C3,PEEK(ADL+C3)+C5:POKE ADL+C6,7
SO 3035 PRINT "     create                SC
   REEN"
QZ 3040 ? :? "Mode line thus far":?
TA 3045 GOSUB 700:TRAP 3100
HM 3050 ? "Enter graphics mode for line:
   ";:INPUT #16,MODE
GJ 3055 IF MODE<0 OR MODE>9 THEN 3100
JM 3060 DLIST(DLEN)=MODE:DLEN=DLEN+C1
QZ 3070 CURLIN=CURLIN+NLINES(MODE):NL=NL+
   C1
BO 3080 ? "Enter text for line":PRINT SP$
   ;"↑↑↑↑↑↑";
XV 3082 GOSUB 500
KQ 3090 IF CURLIN>=192 THEN SCRFLG=C1:DLE
   N=DLEN-C1:GOTO 11000
KQ 3100 POSITION C1,C3:IF NL<>0 THEN FOR
   I=C1 TO DLEN-C1:? DLIST(I);" ";:NEXT I
OD 3105 ?
SJ 3110 GOTO 3045
TQ 4000 IF SCRFLG=0 THEN PRINT "↰";:POSIT
   ION C5,11:? "No Screen in Memory":FOR
   I=C1 TO 500:NEXT I:GOTO 11000
MM 4005 GOSUB 730:GRAPHICS 0:SETCOLOR C2,
   C4,C2:GOSUB 720
FA 4006 ADL=PEEK(560)+256*PEEK(561):POKE
   ADL+C3,PEEK(ADL+C3)+C4:POKE ADL+C6,C6
ZQ 4007 DFLAG=0
RK 4010 PRINT "↰     edit                 S
   CREEN"
NZ 4020 ? "Mode lines      Text"
RN 4025 PTR=C1:IF DFLAG=0 THEN SDISP=C1:E
   DISP=16:IF DLEN<16 THEN EDISP=DLEN
KQ 4026 IF DFLAG=C1 THEN SDISP=17:EDISP=3
   2:IF DLEN<32 THEN EDISP=DLEN
GV 4027 IF DFLAG=C2 THEN SDISP=33:EDISP=4
   8:IF DLEN<48 THEN EDISP=DLEN
RG 4028 IF SDISP<>C1 THEN FOR I=C1 TO SDI
   SP-C1:PTR=PTR+NCHARS(DLIST(I)):NEXT I
PN 4030 FOR I=SDISP TO EDISP:? I;" ";DLIS
   T(I),
HD 4031 IF NCHARS(DLIST(I))<40 THEN PRINT
    SCREEN$(PTR,PTR+NCHARS(DLIST(I))-C1):
   GOTO 4035
ZK 4032 PRINT SCREEN$(PTR,PTR+23);"..."
MB 4035 PTR=PTR+NCHARS(DLIST(I)):NEXT I
SV 4040 ? "Color, Text, Next Page Prior o
   r Exit"
YI 4042 GOSUB 700
XS 4045 GET #C1,X
OL 4050 IF X=ASC("E") THEN GOTO 11000
PK 4060 IF X=ASC("T") THEN 4120
PP 4065 IF X=ASC("N") THEN 4150
TB 4066 IF X=ASC("P") THEN 4170
XY 4070 IF X<>ASC("C") THEN 4045
ND 4080 ? "↰    change                COLORS
   ":? :? :? "Current Colors":FOR I=C1 TO
    C5
QM 4081 ? "Register ";I-C1;" - ";COLR(I):
   NEXT I
TB 4090 ? :? "Register # to change: ";:TR
   AP 4090:INPUT #16,C:IF C<0 OR C>C4 THE
   N 4090
CN 4100 ? "New value: ";:INPUT #16,X:IF X
   <0 OR X>255 THEN 4100
PJ 4102 COLR(C+C1)=X:? :? "Change another
    register?(Y/N): ";:GOSUB 700:GET #C1,
   Y:IF Y<>78 AND Y<>89 THEN 4102
IP 4105 IF Y=89 THEN 4080
QQ 4110 GOTO 4010
UX 4120 TRAP 4010:? "↑Enter number of lin
   e to edit text:  ←←";:INPUT #16,NL
UW 4121 PRINT "Old text":PTR=C1:IF NL=C1
   THEN 4124
JE 4122 FOR I=C1 TO NL-C1:PTR=PTR+NCHARS(
   DLIST(I)):NEXT I
JN 4123 IF NCHARS(DLIST(NL))>38 THEN PRIN
   T SCREEN$(PTR,PTR+36):GOTO 4125
```

```
CY 4124 PRINT SCREEN$(PTR,PTR+NCHARS(DLIS
   T(NL))-C1)
QW 4125 MODE=DLIST(NL)
ML 4126 PRINT "New text"
XB 4130 GOSUB 500
OZ 4140 GOTO 4010
NQ 4150 IF DLEN<17 THEN 4010
PS 4151 IF DLEN<33 THEN DFLAG=C1:GOTO 401
   0
MP 4152 IF DFLAG=C1 THEN DFLAG=C2
QC 4153 IF DFLAG=0 THEN DFLAG=C1
PS 4154 GOTO 4010
PF 4160 GOTO 4010
KF 4170 DFLAG=DFLAG-C1:IF DFLAG<0 THEN DF
   LAG=0
QC 4175 GOTO 4010
NH 5000 GOSUB 730:GRAPHICS 0:SETCOLOR C2,
   C5,C2:GOSUB 720
AN 5001 SCRFLG=0:ADL=PEEK(560)+256*PEEK(5
   61):POKE ADL+C3,PEEK(ADL+C3)+C5:POKE A
   DL+C6,7
MP 5005 FN=0:FILE$(C1)=" ":FILE$(400)=" "
   :FILE$(C2)=FILE$
CR 5010 PRINT "K     create                SE
   QUENCE"
RG 5011 POSITION C2,11:? "Enter filename:
    ";:INPUT #16,FN$
TU 5020 FILE$(FN*20+C1)=FN$
ZE 5030 ? " Any more files? (Y/N): ";
XA 5031 GET #C1,X
KG 5040 IF X=ASC("Y") THEN PRINT CHR$(X):
   FN=FN+C1:GOTO 5010
BB 5045 IF X<>ASC("N") THEN 5031
OL 5046 PRINT CHR$(X)
JE 5050 ? " Automatic or Cued? (A/C): ";
XW 5055 GET #C1,X
VV 5060 IF X=ASC("C") THEN PRINT CHR$(X):
   GOTO 5080
ZK 5065 IF X<>ASC("A") THEN 5055
OR 5066 PRINT CHR$(X)
UN 5070 TRAP 5070:? " Delay before starti
   ng load of":? " following screen: ";:I
   NPUT #16,J:J=J*111
RH 5080 ? " Repeat sequence endlessly? ";
YD 5081 GET #C1,Y
TR 5085 IF (Y<>89) AND (Y<>78) THEN 5081
PO 5086 PRINT CHR$(Y)
FC 5090 FOR F=C1 TO FN+C1
XS 5095 CLOSE #C1
RO 5096 TRAP 5330
DK 5100 FILENAME$=FILE$((F-C1)*20+C1):IF
   FILENAME$(C2,C2)=":" OR FILENAME$(C3,C
   3)=":" THEN 5104
GU 5101 FN$=FILENAME$:FILENAME$="D:":FILE
   NAME$(C3)=FN$
IQ 5104 OPEN #C1,C4,0,FILENAME$
QC 5105 INPUT #C1,FILENAME$:IF FILENAME$(
   )"\SUMMER$\PRESENT\" THEN 5400
AZ 5110 FOR I=C1 TO C5:INPUT #C1,C:COLR(I
   )=C:NEXT I
SP 5120 INPUT #C1,DLEN
NR 5125 FOR I=C1 TO DLEN:INPUT #C1,C:DLIS
   T(I)=C:NEXT I
QR 5130 SCREEN$(2048)=" "
LV 5140 HI=INT(ADR(SCREEN$)/256):LO=ADR(S
   CREEN$)-256*HI
ZP 5141 POKE 850,7:POKE 852,LO:POKE 853,H
   I:POKE 856,0:POKE 857,8:D=USR(ADR(SHIN
   $),16)
XE 5142 SCL=PEEK(856)+256*PEEK(857):SCREE
   N$(SCL)=""
YT 5150 GOSUB 710
XQ 5160 SCR1$(LEN(SCREEN$))=" "
UP 5170 D=USR(ADR(CNV$),ADR(SCREEN$),ADR(
   SCR1$),LEN(SCREEN$))
YD 5205 POKE 559,0:POKE 77,0:POKE 54272,0
RQ 5210 SCR$=SCR1$
YO 5300 GOSUB 170
WP 5310 IF X<>ASC("C") THEN FOR K=C1 TO J
   :NEXT K:GOTO 5330
PZ 5320 GET #C1,K
QS 5330 GOSUB 710:NEXT F
EZ 5340 IF Y=ASC("Y") THEN GOTO 5090
PR 5350 GOTO 11000
WQ 5400 GRAPHICS 24:POKE 710,0:FOR I=LEN(
   FILENAME$)+C1 TO 26:GET #C1,K:FILENAME
   $(I)=CHR$(K):NEXT I
DU 5401 IF FILENAME$(C1,C1)<>CHR$(255) OR
    FILENAME$(C2,C2)<>CHR$(128) THEN 5500
MR 5410 POKE 559,0:DLA=RAMTOP*256-257:FOR
    I=C1 TO 202:POKE DLA+I,ASC(DLA$(I)):N
   EXT I
OJ 5411 POKE 560,0:POKE 561,RAMTOP-C1:POK
   E 88,0:POKE 89,RAMTOP-31:POKE 559,34
JE 5420 FOR I=14 TO 18:POKE 694+I,ASC(FIL
   ENAME$(I,I)):NEXT I
YY 5430 DIR=ASC(FILENAME$(8,8))
OS 5435 IF DIR<>C1 THEN D=USR(ADR(HORZ$))
WP 5440 IF DIR=C1 THEN D=USR(ADR(VERT$))
ZX 5450 GOSUB 710:GOTO 5310
KY 5500 CLOSE #C1:FILENAME$=FILE$((F-C1)*
   20+C1):IF FILENAME$(C2,C2)=":" OR FILE
   NAME$(C3,C3)=":" THEN 5504
IE 5501 FN$=FILENAME$:FILENAME$="D:":FILE
   NAME$(3)=FN$
FD 5504 POKE 559,0:OPEN #C1,C4,0,FILENAME
   $
BM 5505 SETCOLOR C2,0,0:DL=PEEK(560)+256*
   PEEK(561):FOR I=203 TO 201 STEP -1:POK
   E DL+I,PEEK(DL+I-C2)
GU 5506 NEXT I:POKE DL+199,15:POKE DL+200
   ,15:GOSUB 720
TQ 5507 INPUT #C1,C:POKE 712,C:INPUT #C1,
   C:POKE 710,C:INPUT #C1,C:POKE 709,C
TS 5508 POKE 850,7:POKE 852,0:POKE 853,C6
   :POKE 856,202:POKE 857,0
BQ 5509 D=USR(ADR(SHIN$),16)
KT 5510 POKE 850,7:POKE 852,PEEK(88):POKE
    853,PEEK(89):POKE 856,0:POKE 857,31
YZ 5511 POKE 559,34
BD 5515 D=USR(ADR(SHIN$),16)
ZQ 5520 GOSUB 710:GOTO 5310
EQ 6000 GOSUB 730:GRAPHICS 0:SETCOLOR C2,
   10,C2:GOSUB 720
YS 6001 ADL=PEEK(560)+256*PEEK(561):POKE
   ADL+C3,PEEK(ADL+C3)+C5:POKE ADL+C6,7
IE 6002 PRINT "     display               SC
   REEN"
UN 6005 IF SCRFLG=0 THEN POSITION 8,11:PR
   INT "No screen in memory":FOR I=C1 TO
   200:NEXT I:GOTO 11000
AZ 6010 POSITION 7,11:PRINT "Press any ke
   y for menu"
IH 6020 FOR I=C1 TO 200:NEXT I:GOSUB 110:
   GET #C1,X:GOTO 11000
GF 7000 GOSUB 730:GRAPHICS 0:SETCOLOR C2,
   11,C2:GOSUB 720
YT 7001 ADL=PEEK(560)+256*PEEK(561):POKE
   ADL+C3,PEEK(ADL+C3)+C5:POKE ADL+C6,7
VB 7005 PRINT "     disk                  DIR
   ECTORY"
YU 7010 CLOSE #C5:TRAP 7200:OPEN #C5,C6,0
   ,"D:*.*"
FA 7020 TRAP 7100:GET #C5,X:IF X=155 THEN
    PRINT "  ";:GOTO 7020
SH 7025 PRINT CHR$(X);:GOTO 7020
ZL 7100 CLOSE #C5:PRINT :PRINT "Press any
    key for menu ";:GET #C1,X
PD 7110 GOTO 11000
CH 7200 PRINT "Cannot open drive one's di
   rectory":PRINT "Please check the drive
    to be sure it"
PS 7210 PRINT "is on and the disk inside
   is in the ":PRINT "format of the curre
   nt DOS"
QE 7220 GOTO 7100
```

```
KW 8000 GOSUB 730:GRAPHICS 0:SETCOLOR C2,
   14,C2:GOSUB 720
EK 8001 ADL=PEEK(560)+256*PEEK(561):POKE
   ADL+C3,PEEK(ADL+C3)+C4:POKE ADL+C6,C6
ZA 8010 PRINT "run a prewritten          SE
   QUENCE"
ZU 8011 POSITION C3,11:PRINT "Enter filen
   ame holding sequence"
MB 8012 POSITION 10,13:PRINT " ";:INPUT #
   16,FILENAME$
II 8013 IF FILENAME$(C2,C2)=":" OR FILENA
   ME$(C3,C3)=":" THEN 8020
HK 8014 FN$=FILENAME$:FILENAME$="D:":FILE
   NAME$(C3)=FN$
WE 8020 TRAP 1120:CLOSE #C1:OPEN #C1,C4,0
   ,FILENAME$
JE 8030 FN=0:TRAP 8050
CZ 8035 INPUT #C1,FILENAME$
PH 8040 FILE$(20*FN+C1)=FILENAME$:FN=FN+C
   1:GOTO 8035
AS 8050 GOSUB 710:GOTO 5050
TV 10000 CLR :DIM SCREEN$(2048),SCR$(2048
   ):A=ADR(SCR$):IF INT(A/4096)=INT((A+20
   48)/4096) THEN 10002
HL 10001 CLR :DIM SCR$(2048),SCREEN$(2048
   )
IH 10002 DIM FILENAME$(40),COLR(5),NLINES
   (8),NCHARS(8),TEXT$(82)
RV 10003 DIM DLIST(100),SP$(80),DLI(130),
   ANTIC(8),LINE$(80),SCR1$(2048)
KH 10004 DIM FN$(20),FILE$(400),CNV$(101)
   ,VERT$(374),SHIN$(7),HORZ$(289),DLA$(2
   02)
SK 10005 C1=1:C2=C1+C1:C3=C2+C1:C4=C2+C2:
   C5=C2+C3:C6=C5+C1
MG 10010 GRAPHICS 18:PRINT #C6:PRINT #C6
XI 10020 PRINT #C6;"     PreSEntATioN"
DO 10025 PRINT #C6;"       GraPHicS"
KF 10030 PRINT #C6;"       ProGRaM"
MW 10031 PRINT #C6
VT 10032 PRINT #C6;"     from analog    "
VT 10033 PRINT #C6;"     by j summerS"
VU 10039 FOR I=C1 TO 100:NEXT I
SK 10040 FOR I=C1 TO C5:READ C:COLR(I)=C:
   NEXT I
AQ 10041 DATA 40,202,148,70,0
TW 10050 FOR I=0 TO 8:READ C:NLINES(I)=C:
   NEXT I
VZ 10051 DATA 8,8,16,8,4,4,2,2,1
JS 10060 FOR I=0 TO 8:READ C:NCHARS(I)=C:
   NEXT I
JD 10061 DATA 40,20,20,10,10,20,20,40,40
BE 10070 OPEN #C1,C4,0,"K:"
BI 10090 SP$(C1)=" ":SP$(80)=" ":SP$(C2)=
   SP$(C1)
MJ 10091 LINE$(C1)="_":LINE$(80)="_":LINE
   $(C2)=LINE$(C1)
KL 10100 FOR I=0 TO 8:READ C:ANTIC(I)=C:N
   EXT I
QF 10110 DATA 2,6,7,8,9,10,11,13,15
LH 10130 GOSUB 12000
ZQ 10150 SHIN$="hhh▒LV▒"
AN 10160 DLA$(C1,C5)="PPPN♥":RAMTOP=PEEK(
   106):DLA$(C6)=CHR$(RAMTOP-31):FOR I=7
   TO 199:DLA$(I)=CHR$(14):NEXT I
UV 10170 DLA$(102,103)="N♥":DLA$(104,104)
   =CHR$(RAMTOP-16):DLA$(200,201)="A♥":DL
   A$(202,202)=CHR$(RAMTOP-C1)
ZM 11000 GRAPHICS 0:GOSUB 720
XE 11001 ADL=PEEK(560)+256*PEEK(561):POKE
    ADL+C3,PEEK(ADL+C3)+C5:POKE ADL+C6,C6
VD 11002 PRINT "  presentation           G
   RAPHICS"
NE 11003 POKE ADL+26,7:POKE ADL+27,C6:POK
   E ADL+28,7:POSITION C2,20
VN 11004 PRINT "  presentation           G
   RAPHICS"
FU 11009 POSITION C2,C1
BX 11010 PRINT "Presentation Graphics Pro
   gram":?
BH 11020 PRINT "Select option by number:"
MQ 11030 PRINT "0 - Exit"
FV 11040 PRINT "1 - Load screen"
PU 11050 PRINT "2 - Save Screen"
LQ 11060 PRINT "3 - Create screen"
OC 11070 PRINT "4 - Edit screen"
IK 11080 PRINT "5 - Create Presentation s
   equence"
UE 11081 PRINT "6 - Display current scree
   n"
DR 11082 PRINT "7 - Directory of drive 1"
DB 11083 PRINT "8 - Run a pre-written seq
   uence"
VS 11090 ? :PRINT "Select: ";:GET #C1,X
QE 11100 X=X-48:IF X<0 OR X>8 THEN GOSUB
   730:GOTO 11000
TB 11110 ON X GOTO 1000,2000,3000,4000,50
   00,6000,7000,8000
NV 11120 GOSUB 730:GRAPHICS 0:POKE 559,34
   :END
BK 13000 RETURN
```

Listing 2.
BASIC listing.

```
ER 10 OPEN #1,8,0,"D:LINES.LST"
VX 20 LINE=12000:LINEINC=10:STRPOS=1:SAVP
   OS=1
IG 30 DIM STRNAME$(10),STORE$(80)
CX 40 STRNAME$="CNV$"
AV 50 PRINT #1;LINE;" ";STRNAME$;"(";STRP
   OS;",";
RP 55 SAVPOS=STRPOS
TK 60 READ A:IF A=999 THEN 99
AH 65 CKSUM=CKSUM+A*STRPOS
TZ 70 STORE$(LEN(STORE$)+1)=CHR$(A):STRPO
   S=STRPOS+1:IF (STRPOS-1)/80<>INT((STRP
   OS-1)/80) THEN 60
GX 75 PRINT #1;SAVPOS-1+LEN(STORE$);")=";
   CHR$(34);STORE$;:STORE$=""
OL 80 LINE=LINE+LINEINC:? #1:GOTO 50
HH 99 PRINT #1;SAVPOS-1+LEN(STORE$);")=";
   CHR$(34);STORE$;:STORE$=""
XN 100 IF STRNAME$="HORZ$" THEN 1000
VF 110 IF STRNAME$="VERT$" THEN STRNAME$=
   "HORZ$"
SF 120 IF STRNAME$="CNV$" THEN STRNAME$="
   VERT$"
YN 125 STRPOS=1
EM 130 LINE=LINE+LINEINC:? #1
QY 140 GOTO 50
VA 1000 IF CKSUM<>14670727 THEN PRINT "ER
   ROR IN DATA!"
FX 1010 ? #1:CLOSE #1:END
FF 15000 DATA 104, 104, 133, 204, 104, 13
   3, 203, 104, 133, 206
SG 15010 DATA 104, 133, 205, 104, 133, 20
   8, 104, 133, 207, 160
HX 15020 DATA 0, 177, 203, 41, 128, 133,
   209, 177, 203, 41
UJ 15030 DATA 127, 201, 96, 16, 12, 201,
   32, 48, 6, 56
PU 15040 DATA 233, 32, 24, 144, 2, 9, 64,
    5, 209, 145
JD 15050 DATA 205, 165, 203, 24, 105, 1,
   133, 203, 165, 204
FT 15060 DATA 105, 0, 133, 204, 165, 205,
    24, 105, 1, 133
EE 15070 DATA 205, 165, 206, 105, 0, 133,
    206, 165, 207, 56
MM 15080 DATA 233, 1, 133, 207, 165, 208,
    233, 0, 133, 208
ZK 15090 DATA 201, 0, 208, 181, 165, 207,
    201, 0, 208, 175
LU 15100 DATA 96,999
```

```
WD 16000 DATA 104, 165, 88, 133, 216, 165
   , 89, 133, 217, 169
CS 16010 DATA 0, 133, 212, 133, 220, 133,
    221, 133, 222, 133
JA 16020 DATA 223, 169, 1, 133, 213, 165,
    216, 24, 101, 212
UJ 16030 DATA 133, 214, 165, 217, 105, 0,
    133, 215, 165, 214
QG 16040 DATA 24, 105, 0, 133, 218, 165,
    215, 105, 30, 133
PD 16050 DATA 219, 162, 16, 169, 7, 157,
    66, 3, 169, 1
MZ 16060 DATA 157, 72, 3, 169, 0, 157, 73
   , 3, 169, 220
TH 16070 DATA 157, 68, 3, 169, 0, 157, 69
   , 3, 32, 86
ZB 16080 DATA 228, 169, 0, 133, 222, 165,
    220, 16, 8, 41
XD 16090 DATA 127, 133, 220, 169, 1, 133,
    222, 165, 220, 208
DA 16100 DATA 47, 162, 16, 169, 7, 157, 6
   6, 3, 169, 2
MI 16110 DATA 157, 72, 3, 169, 0, 157, 73
   , 3, 169, 220
SQ 16120 DATA 157, 68, 3, 169, 0, 157, 69
   , 3, 32, 86
IQ 16130 DATA 228, 165, 220, 166, 221, 13
   4, 220, 133, 221, 24
QO 16140 DATA 144, 6, 24, 144, 162, 24, 1
   44, 193, 162, 16
JB 16150 DATA 169, 7, 157, 66, 3, 169, 1,
    157, 72, 3
WM 16160 DATA 169, 0, 157, 73, 3, 169, 22
   3, 157, 68, 3
PJ 16170 DATA 169, 0, 157, 69, 3, 32, 86,
    228, 160, 0
UZ 16180 DATA 165, 223, 145, 214, 165, 21
   4, 24, 105, 80, 133
VQ 16190 DATA 214, 165, 215, 105, 0, 133,
    215, 197, 219, 48
ZF 16200 DATA 83, 165, 214, 197, 218, 48,
    77, 165, 213, 201
OS 16210 DATA 1, 208, 42, 165, 216, 24, 1
   01, 212, 133, 214
PG 16220 DATA 165, 217, 105, 0, 133, 215,
    165, 214, 24, 105
CD 16230 DATA 40, 133, 214, 165, 215, 105
   , 0, 133, 215, 169
TX 16240 DATA 2, 133, 213, 24, 144, 38, 2
   4, 144, 149, 24
JX 16250 DATA 144, 149
EI 16260 DATA 24, 144, 179, 169, 1, 133,
    213, 165, 212, 24
EM 16270 DATA 105, 1, 133, 212, 201, 40,
    208, 1, 96, 165
CF 16280 DATA 216, 24, 101, 212, 133, 214
   , 165, 217, 105, 0
JE 16290 DATA 133, 215, 165, 222, 201, 1,
    208, 41, 240, 9
GN 16300 DATA 24, 144, 207, 24, 144, 207,
    24, 144, 207, 162
JK 16310 DATA 16, 169, 7, 157, 66, 3, 169
   , 1, 157, 72
LL 16320 DATA 3, 169, 0, 157, 73, 3, 169,
    223, 157, 68
SG 16330 DATA 3, 169, 0, 157, 69, 3, 32,
    86, 228, 165
CZ 16340 DATA 220, 56, 233, 1, 133, 220,
    165, 221, 233, 0
AD 16350 DATA 133, 221, 165, 220, 208, 20
   6, 165, 221, 208, 202
IE 16360 DATA 165, 222, 201, 1, 208, 13,
    165, 223, 133, 220
RQ 16370 DATA 169, 0, 133, 221, 133, 222,
    24, 144, 180, 24
LW 16380 DATA 144, 174, 999
WF 17000 DATA 104, 165, 88, 133, 216, 165
   , 89, 133, 217, 169
AJ 17010 DATA 0, 133, 220, 133, 221, 133,
    222, 133, 223, 169
RR 17020 DATA 1, 165, 216, 133, 214, 165,
    217, 133, 215, 165
NK 17030 DATA 214, 24, 105, 0, 133, 218,
    165, 215, 105, 30
QE 17040 DATA 133, 219, 162, 16, 169, 7,
    157, 66, 3, 169
SO 17050 DATA 1, 157, 72, 3, 169, 0, 157,
    73, 3, 169
SF 17060 DATA 220, 157, 68, 3, 169, 0, 15
   7, 69, 3, 32
JU 17070 DATA 86, 228, 169, 0, 133, 222,
    165, 220, 16, 8
SC 17080 DATA 41, 127, 133, 220, 169, 1,
    133, 222, 165, 220
AP 17090 DATA 208, 47, 162, 16, 169, 7, 1
   57, 66, 3, 169
SJ 17100 DATA 2, 157, 72, 3, 169, 0, 157,
    73, 3, 169
RO 17110 DATA 220, 157, 68, 3, 169, 0, 15
   7, 69, 3, 32
TQ 17120 DATA 86, 228, 165, 220, 166, 221
   , 134, 220, 133, 221
UL 17130 DATA 24, 144, 6, 24, 144, 162, 2
   4, 144, 193, 162
JS 17140 DATA 16, 169, 7, 157, 66, 3, 169
   , 1, 157, 72
LT 17150 DATA 3, 169, 0, 157, 73, 3, 169,
    223, 157, 68
IY 17160 DATA 3, 169, 0, 157, 69, 3, 32,
    86, 228, 160
OA 17170 DATA 0, 165, 223, 145, 214, 165,
    214, 24, 105, 1
NY 17180 DATA 133, 214, 165, 215, 105, 0,
    133, 215, 197, 219
EL 17190 DATA 48, 7, 165, 214, 197, 218,
    48, 1, 96, 165
QS 17200 DATA 222, 201, 1, 208, 41, 240,
    9, 24, 144, 179
DF 17210 DATA 24, 144, 179, 24, 144, 209,
    162, 16, 169, 7
CJ 17220 DATA 157, 66, 3, 169, 1, 157, 72
   , 3, 169, 0
PT 17230 DATA 157, 73, 3, 169, 223, 157,
    68, 3, 169, 0
HX 17240 DATA 157, 69, 3, 32, 86, 228, 16
   5, 220, 56, 233
LL 17250 DATA 1, 133
MK 17260 DATA 220, 165, 221, 233, 0, 133,
    221, 165, 220, 208
EK 17270 DATA 206, 165, 221, 208, 202, 16
   5, 222, 201, 1, 208
LF 17280 DATA 13, 165, 223, 133, 220, 169
   , 0, 133, 221, 133
OG 17290 DATA 222, 24, 144, 180, 24, 144,
    174
TL 17300 DATA 999
```

 continued from page 19

```
   PLY=200
   DN()
   Poke(559,0)
   Poke(712,0)
   UP()
   PLY=50
   UP()
RETURN


PROC PAUSE()
  Poke(764,255)
  SndRst()
  Position(27,0)
  Print("PAUSED")
  WHILE Peek(764)<>33 DO OD
  Position(27,0)
  Print("      ")
  Poke(764,255)
RETURN


PROC TITLE()
  SndRst()
  J=PeekC(88)
  Poke(77,0)
  Poke(88,SLO-40)
  Poke(89,SHI-1)
  Print("***>    B*C           ")
  Print("                A******")
  Print("***C     3            ")
  Print("                B******")
  Print("***                   ")
  Print("                 4B****")
  Print("**C      + + ++> +    ")
  Print("+ A+> ++> ++>    3B**")
  Print("*C       * * * * *    ")
  Print("* * * * * *B*       B*")
  Print("*>       * * **+C * + ")
  Print("* *** **C * *        *")
  Print("**       *>* *    *>*A")
  Print("* * * *B> *A*      A*")
  Print("**>      B*C *    B***")
  Print("C * * * B>**C      A**")
  Print("B*******************")
  Print("++++++++++++++++++*C")
  Print(" 3456789876543456789")
  Print("8987654345678987653 ")
  Zero(SHI*256+400,600)
  PokeC(88,J)
  Print("5          UPWARD")
  Position(4,1)
  Print("Copyright 1985 ")
  Print("ANALOG Computing")
  IF HSC=0 THEN
    Position(13,2)
    Print("By Greg Knauss")
  FI
  IF HSC>0 THEN
    Position(13,2)
    Print("High Score: ")
    PrintC(HSC)
  FI
  AIR=6600
  STR=246
  SN=1
  N=0
  V=0
  P=183
  PLY=187
  S=15
  PIC=0
  MOVE()
  FADEIN()
  FOR L=1 TO 5 DO
  PLY=187
    PD=DRW+208
    FOR J=1 TO 10000 DO OD
    PD=DRW
    FOR J=1 TO 10000 DO OD
  OD
  SRTDRT()
  S=11
  WHILE Peek(53279)<>6 AND
        STrig(0)<>0 DO
    AIR=6600
    STR=246
    IF P=190 THEN
      S=11
    FI
    IF P=58 THEN
      S=7
    FI
    PLY=187
    WAY()
    MOVE()
    ANIMATE()
    SKYLER()
    PLAY()
    Poke(53278,0)
    FOR J=1 TO 700 DO OD
    IF Peek(53279)=5 THEN
      SSP==-1
      SCT=0
      IF SSP=0 THEN
        SSP=3
      FI
      SndRst()
      FOR J=1 TO 250 DO
        Sound(0,J,10,15-J/16)
      OD
    FI
    IF Peek(53279)=3 THEN
      SAC==-20
      IF SAC=0 THEN
        SAC=80
      FI
      SndRst()
      FOR J=1 TO 250 DO
        Sound(0,250-J,10,15-J/16)
      OD
    FI
    IF P=DAX AND DART=0
             AND DAY=0 THEN
      DART=1
      DAY=Rand(100)+75
      DX=Rand(2)
      IF DX=0 THEN
        DX=-3
        DAX=245
      FI
      IF DX=1 THEN
        DX=3
        DAX=3
      FI
    FI
    IF DART=1 THEN
      BANG()
    FI
    IF DART=0 THEN
      SRTDRT()
    FI
  OD
  SC=110
  PLRGO()
  SHI=SHI+4
  Poke(DL+14,SHI)
RETURN


PROC ERR()
  Close(1)
  IF SN=1 THEN
    Position(0,1)
    Print(" Please insert a ")
    PrintE("disk with Upward sets ")
```

```
    Print("    present and press")
    Print(" the trigger .   ")
    Poke(712,149)
    Poke(559,62)
    WHILE STrig(0)=1 DO OD
    SSP==+1
    SAC==+20
    Poke(559,0)
    Poke(712,0)
  FI
  SN=1
  SSP==-1
  IF SSP=0 THEN
    SSP=1
    SAC==-20
    IF SAC=0 THEN
      SAC=20
      SSP=1
    FI
  FI
  Open(1,"D:UPSET.A",4,0)
RETURN


PROC READSET()
  IF LNS=0 THEN
    Close(1)
    LNS=1
    Poke(ST+9,SN+64)
    Error=ERR
    Open(1,ST,4,0)
    FOR J=0 TO 2999 DO
      S=GetD(1)
      Poke(SHI*256+J,S)
    OD
  FI
  Close(1)
  AIR=6600
  STR=246
  ES=0
  Print("K        UPWARD")
  Position(2,1)
  Print("Str:                ")
  Print("                    ")
  Position(2,2)
  Print("Air:                ")
  Print("                    ")
  FOR J=3000 TO 3079 DO
    Poke(SHI*256+J,10)
  OD
  Poke(559,62)
  FADEIN()
  FOR P=1 TO 11 DO
    DN()
    PLY=200
  OD
  PD=DRW
  Poke(704,0)
  P=185
  PLY=187
  MOVE()
  FOR I=0 TO 15 DO
    Poke(704,I)
    FOR J=1 TO 3000 DO OD
  OD
RETURN


PROC GOGOGO()
  DO
    SETUP()
    TITLE()
    DO
      READSET()
      WHILE ES=0 DO
        Poke(77,0)
        IF PC=8 OR PC=10 OR PC=12 OR
           STR<54 OR AIR<54 THEN
          ES=1
        FI
        IF PLY<110 AND SLO=0 THEN
          ES=2
        FI
        AIR==-1
        IF AIR MOD 200=0 THEN
          OXY()
        FI
        S=Stick(0)
        WAY()
        IF STrig(0)=0 AND PC<>0 AND
           PC<>2 AND PC<>3 AND
           PC<>8 THEN
          JUMP()
        FI
        LEGAL()
        MOVE()
        SKYLER()
        ANIMATE()
        PLAY()
        Poke(53278,0)
        IF P=DAX AND DART=0 AND
           DAY=0 THEN
          DART=1
          DAY=PLY+10
          DX=Rand(2)
          IF DX=0 THEN
            DX=-3
            DAX=245
          FI
          IF DX=1 THEN
            DX=3
            DAX=3
          FI
        FI
        IF DART=1 THEN
          BANG()
        FI
        IF PLY>189 THEN
          DN()
        FI
        IF PLY<85 THEN
          UP()
        FI
        FOR J=1 TO 700 DO OD
        IF Peek(764)=33 THEN
          PAUSE()
        FI
      OD
      IF ES=2 THEN
        SC==+(STR/3)+(AIR/100)
        Poke(706,0)
        Poke(707,0)
        Position(25,0)
        Print("SCORE: ")
        PrintCE(SC)
        Print("              ")
        Print("Congratulations!")
        PrintE("              ")
        Print("  For next cavern ")
        Print("press the trigger .")
        WHILE STrig(0)=1 DO
          PD=DRW+16
          X1=0
          SKY=200
          JUMP()
          FOR J=1 TO 800 DO OD
        OD
        Zero(SHI*256+400,600)
        PLRGO()
        SC==+110
        LNS=0
        SN==+1
      FI
      UNTIL ES=1
    OD
    PD=DRW+16
```

```
    FOR I=0 TO 15 DO
      PL==-1
      PLY==+1
      Poke(704,15-I)
      SKYLER()
      IF DART=1 THEN
        BANG()
      FI
      PLAY()
      FOR J=1 TO 1000 DO OD
    OD
    S=0
    WHILE STrig(0)=1 DO
      SKYLER()
      IF DART=1 THEN
        BANG()
      FI
      PLAY()
      S==+1
      IF S=1 THEN
        Position(21,0)
        Print("    game  over    ")
      FI
      IF S=100 THEN
        Position(21,0)
        Print("    SCORE: ")
        PrintC(SC)
        IF SC>HSC THEN
          Print("*")
        FI
        Print("   ")
      FI
      FOR DL=1 TO 800 DO OD
    OD
    IF SC>HSC THEN
      HSC=SC
    FI
    IF SN>1 THEN LNS=0 FI
  OD
RETURN
```

Listing 2.
BASIC listing.

```
IK 0 POKE 82,0:DIM A$(42),B$(42)
DZ 10 ? "↑↓↓↓  PLEASE PUT THE DISK TO CON
   TAIN THE      SET INTO DRIVE ONE AND P
   RESS RETURN.":INPUT A$
UE 20 OPEN #1,8,0,"D:UPSET.A"
QV 30 FOR J=1 TO 75:READ A$
BP 40 IF LEN(A$)=42 THEN 70
SR 50 A$(LEN(A$)+1)=","
NQ 60 READ B$:A$(LEN(A$)+1)=B$:GOTO 40
JN 70 B$=A$(2,41)
QJ 80 FOR I=1 TO 40:PUT #1,ASC(B$(I))-32
PH 90 NEXT I:NEXT J
FJ 100 CLOSE #1:? "  OK! SET'S DONE, YOU
   CAN PLAY NOW!"
SO 10000 DATA '
                  '
S5 10010 DATA '
                  '
FZ 10020 DATA '//012   2212   21000000000
   011100112      '
RV 10030 DATA '******D+*******++++++++++++
   **********+,,--'
OG 10040 DATA '954343EB89876543B******765
   43456789876E43'
GI 10050 DATA '     E          4B***C
               E  '
UU 10060 DATA '     E            BC3
               E  '
NV 10070 DATA '     E
               E  '
BY 10080 DATA '     E          <++=    .
    11         E  '
WL 10090 DATA '     E    <=     34    3
    34         E  '
OX 10100 DATA '
     2         E  '
EP 10110 DATA '                   210/.-,=
     4         E  '
ZL 10120 DATA ' E 210/.-,=     345678C
               E  '
UP 10130 DATA '<D;8987654
               E  '
KV 10140 DATA ' E           A+++++++++,-.
   /012            '
UL 10150 DATA ' E          A**********C B*
   ****+,-./012 2'
DT 10160 DATA ' E          *****F***C   3
   B***********DC'
AP 10170 DATA '     <=     **C  E BC
    4 B********E  '
FG 10180 DATA '            B*   E
       BC 5678CE  '
VF 10190 DATA ' <D=      0  8   E     <=
               E  '
US 10200 DATA '  E   <=  31    //
       2   0   E  '
IZ 10210 DATA '  E       4    43
       4   3   E  '
TG 10220 DATA '++D+HGI+HI+HI+++HGGGGI++HI
   +HGIHGGGGGI+++'
DG 10230 DATA '74E4567898765434567898765 4
   34567898765434'
OT 10240 DATA '  E
                  '
PB 10250 DATA '++++++++=    <,-./012
                  '
FY 10260 DATA '89876543      3456789;,-./
   012            '
SR 10270 DATA '                      3456
   789;+++++++D++'
XV 10280 DATA ' AD>
      345678CE34'
DX 10290 DATA '+CE*   <HGGGGGI=
               E  '
QG 10300 DATA '*>E*    4567654
               E  '
ZU 10310 DATA '**F*>          00  <=
       2       E  '
MW 10320 DATA '*CEB*   <++=   43       <
   ++=    4     E  '
KB 10330 DATA '* E *>   34
   34              '
CW 10340 DATA '*>E **HGGGGGGGGGGGGGGGGGGG
   GGGGGGGGGGGGGG'
DK 10350 DATA '**EA**********************
   ***************'
TM 10360 DATA '*CEB****F******F****F****F
   *********F****F'
ZY 10370 DATA '3 E 3454E345654E3453E 343E
   34678765E3453E'
UE 10380 DATA '  E     E     E         E
              E    E'
FL 10390 DATA '  E     E     E         E
                E'
AH 10400 DATA '  E           E         E
                E'
DA 10410 DATA '             E
                E'
PI 10420 DATA '
                E'
PM 10430 DATA '
                E'
VX 10440 DATA '++++++HGGI+++++++++HGI++++
   +++++++++++++D'
GR 10450 DATA '754345678987654345678987654
   345678987654E'
PY 10460 DATA '
                E'
GX 10470 DATA '<D++++++++++++++++++++++++
   +HGGI+++++++++'
CA 10480 DATA ' EB789876543456789876543456
```

```
   67898765434567'
OI 10490 DATA ' E
                 '
BM 10500 DATA '++=  <=   <+=   <+++=    <
   +++=   <=   <D'
GY 10510 DATA '43          3     *C3
   43            E'
LU 10520 DATA '                 A*
               E'
QP 10530 DATA '   <++++++++++++*C
               E'
EL 10540 DATA '    3456789876543210/.-,=
   <++++++++++++'
NL 10550 DATA '          210/.-,;8876B**
   **C 3456788'
JE 10560 DATA ' 210/.-,;9876543       B*HG
   GGI*C          '
LG 10570 DATA 'D9765433                B*9
   898C           '
NX 10580 DATA 'E                        3
                 '
WN 10590 DATA '++++++++++++++++++++++++++
   ++++++++++++D'
ZW 10600 DATA '5434567898765434567898765 4
   3456789876543E'
IU 10610 DATA '                2
               E'
SM 10620 DATA '+D++++=    A+>  4    <+++++
   +++++++++++++'
SU 10630 DATA 'CEB876     BFC          3456
   789876543456 77'
KR 10640 DATA ' E          E    <++=
                 '
PC 10650 DATA ' E  2222    E     34
                 '
DK 10660 DATA 'GGGGK**C    E
   <+++++++D=   '
XO 10670 DATA 'GGGGI*C     E
      345678CE   '
OV 10680 DATA 'GGI**C    <+++++++++++++++
   +HI+=       E   '
EW 10690 DATA '***C3      4457654345678987
   6543        E   '
TU 10700 DATA '653
               E   '
NN 10710 DATA '   <D=    <+++++++HGI+++++++
   +++++++=    E   '
GA 10720 DATA '    E      54345678987654345
   6789863       A'
MH 10730 DATA '    E             2    2
                A*'
PF 10740 DATA '++++++++++++++++*HGI*++++++
   +++++++++++**'
```

Listing 3.
Action! listing.

```
;UPWARD EDITOR by Greg Knauss

;       CHECKSUM DATA
;[B4 F9 4B 46 88 45 83 FF
; 5A 0C 27 3D EC 4A 1F 34
; 24 27 7E 19 65 C4 11 E8
; 69 63 6C C8 DE AC 77 1F
; 4F 3A 10 9D 15 33 5B 6D
; 65 AD 02 5F 74 60 ]

BYTE
  P,I,X=[1],Y=[1],S,STG=[0],SLO,SHI,
  STC=[32],SN=[1],CLR=[32],ERR=[0],
  L=[1]

INT
  X1,Y1

CARD
  J,SCN,DL,CH

BYTE ARRAY
  ST="D:UPSET.A"


PROC DLI()
  [ 72 169  56 141  10 212 141  22
   208 169  54 141  23 208 169  10
   141  24 208 169  68 141  25 208
   169   0 141  10 212 169   0 141
    26 208 169   0 141   9 212 104
    64]
RETURN


PROC NEWLIST()
  [112 112 112  71   0   0   6 130
     0   5  32 101   0   0  37  37
    37  37  37  37  37  37  37   5
    65   0   0]
RETURN


PROC CHSTORE()
  [170 170 170 170 170 170 170 170
    85 170 170 170 170 170 170 170
     0  85 170 170 170 170 170 170
     0   0  85 170 170 170 170 170
     0   0   0  85 170 170 170 170
     0   0   0   0  85 170 170 170
     0   0   0   0   0  85 170 170
     0   0   0   0   0   0  85 170
     0   0   0   0   0   0   0  85
   168  32   0   0   0   0   0   0
   170  42  32   0   0   0   0   0
   170  42  34   0   0   0   0   0
   170 170 170 168  32   0   0   0
   170 170 170 170 168  32   0   0
   170 170 170 170 170 168  32   0
   170 170 170 170 170 170  40  32
     0   0   0   0   0   0   0   0
    85 170 170 170 170 170 170  32
    85 170  42  42  42  10  10   0
    85 170 170 168 168 160 128 128
     0   0  64 128 144 164 168 169
     0   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0
     1   2   2   6  10  26  42 106
   170  42  42  10  10   2   2   2
   170 168 160 128 128 128   0   0
   125 190 190 190 174 190 190 190
    48  60  60  60  12  60  60  60
   186 190 190 190 174 190 190 190
   255 255 255 255 255 255 255 255
   255 191 191 191 175 175 171 170
   255 254 254 250 250 234 170 170
   170 171 175 175 191 191 255 255
   170 234 234 250 250 250 254 254

The following data is not used in

"Upward" and is unique to this
program.

   255 195 195 195 195 195 195 255
     0  63  48  63   3   3  63   0
     0  60  51  51  51  51  60   0
    15  15   3  63 207 207  12  60
   192 204  12 240 192 192 192 240]
RETURN


PROC TOPSCRN()
  Poke(83,39)
  Position(0,1)
  Print(" *+,-./012 3456789; <=>ABC")
```

```
  Print(" DEF GHIJK   ")
  Position(0,2)
  Print(" *+,-./012 3456789; <=>ABC")
  Print(" DEF GHIJK MN ")
  Poke(83,0)
RETURN

PROC SETUP()
  Graphics(0)
  Poke(752,1)
  Poke(82,1)
  PrintE(" ")
  DL=PeekC(560)
  SHI=Peek(106)-32
  SHI==+8
  SLO=0
  MoveBlock(DL,NEWLIST,50)
  SCN=PeekC(88)
  J=PeekC(88)
  PokeC(DL+4,J)
  Poke(DL+12,0)
  Poke(DL+13,SHI)
  J=PeekC(560)
  PokeC(DL+25,J)
  CH=(Peek(106)-40)*256
  MoveBlock(CH+80,CHSTORE,512)
  Zero(CH,8)
  Poke(DLI+35,CH/256)
  PokeC(512,DLI)
  Poke(54286,192)
  Poke(708,0)
  Poke(709,6)
  Poke(710,15)
  Poke(712,148)
  Zero(SHI*256+SLO,256*12+40)
  Print("K  UPWARD  EDITOR")
  PrintE("          LEVEL: 1")
  TOPSCRN()
RETURN

PROC OOPS()
  Poke(559,34)
  Position(0,1)
  Print("   DISK ERROR! ")
  Print(" Any key to continue.K")
  I=GetD(1)
  ERR=1
  Close(2)
RETURN

PROC SETNO()
  FOR J=1 TO 15000 DO OD
  WHILE STrig(0)=1 DO
    IF Stick(0)=7 AND SN<25 THEN
      SN==+1
    FI
    IF Stick(0)=11 AND SN>1 THEN
      SN==-1
    FI
    IF Stick(0)<>15 THEN
      Position(24,1)
      Print("  ++")
      PrintB(SN)
    FI
    FOR J=1 TO 7000 DO OD
  OD
RETURN

PROC CONFIRM()
  S=0
  Position(31,1)
  Print("Y/N")
  FOR J=1 TO 15000 DO OD
  WHILE STrig(0)=1 DO
    IF Stick(0)=7 THEN
      Position(31,1)
      Print("Y/N")
      S=0
    FI
    IF Stick(0)=11 THEN
      Position(31,1)
      Print("Y/N")
      S=1
    FI
  OD
RETURN

PROC SAVE()
  Position(0,1)
  Print("                Save set:")
  Print("                ")
  Position(24,1)
  PrintB(SN)
  SETNO()
  Position(6,1)
  Print("Confirm: Save as set ")
  PrintB(SN)
  Print("?")
  CONFIRM()
  IF S=0 THEN
    RETURN
  FI
  Poke(559,0)
  SHI=Peek(106)-32
  SLO=0
  L=1
  Position(33,0)
  Print("  ++")
  PrintB(L)
  Poke(DL+12,SLO)
  Poke(DL+13,SHI)
  Close(2)
  Error=OOPS
  ERR=0
  Poke(ST+9,SN+64)
  Open(2,ST,8,0)
  IF ERR=0 THEN
    FOR J=0 TO 2999 DO
      S=Peek(SHI*256+J)
      PutD(2,S)
    OD
    Close(2)
  FI
  Poke(559,34)
RETURN

PROC LOAD()
  Position(0,1)
  Print("                Load set:")
  Print("                ")
  Position(24,1)
  PrintB(SN)
  SETNO()
  Position(6,1)
  Print("Confirm: Load set ")
  PrintB(SN)
  Print("?")
  CONFIRM()
  IF S=0 THEN
    RETURN
  FI
  Poke(559,0)
  SHI=Peek(106)-32
  SLO=0
  L=1
  Position(33,0)
  Print("  ++")
  PrintB(L)
  Poke(DL+12,SLO)
  Poke(DL+13,SHI)
```

```
  Error=OOPS
  ERR=0
  Poke(ST+9,SN+64)
  Open(2,ST,4,0)
  IF ERR=0 THEN
    FOR J=0 TO 2999 DO
      S=GetD(2)
      Poke(SHI*256+J,S)
    OD
    Close(2)
  FI
  Poke(559,34)
RETURN

PROC CLEAR()
  Position(6,1)
  Print("Confirm: Clear set?")
  Print("              ")
  CONFIRM()
  IF S=1 THEN
    SHI=Peek(106)-32
    SLO=0
    L=1
    Position(33,0)
    Print("  ↔↔")
    PrintB(L)
    Poke(DL+12,SLO)
    Poke(DL+13,SHI)
    Zero(SHI*256+SLO,3000)
    PokeC(88,SCN)
  FI
RETURN

PROC DISK()
  Poke(83,39)
  Position(0,1)
  Print("      EXIT     SAVE     LOAD")
  Print("    CLEAR      ")
  S=1
  WHILE STrig(0)=1 DO
    IF S>1 AND Stick(0)=11 THEN
      S==-1
    FI
    IF S<4 AND Stick(0)=7 THEN
      S==+1
    FI
    IF Stick(0)=7 OR Stick(0)=11 THEN
      Position(0,1)
      Print("      EXIT     SAVE")
      Print("    LOAD     CLEAR      ")
    FI
    IF S=1 THEN
      Position(5,1)
      Print(" EXIT ")
    FI
    IF S=2 THEN
      Position(13,1)
      Print(" SAVE ")
    FI
    IF S=3 THEN
      Position(21,1)
      Print(" LOAD ")
    FI
    IF S=4 THEN
      Position(29,1)
      Print(" CLEAR ")
    FI
    FOR J=1 TO 5000 DO OD
  OD
  FOR J=1 TO 10000 DO OD
  IF S=2 THEN
    SAVE()
  FI
  IF S=3 THEN
    LOAD()
  FI
  IF S=4 THEN
    CLEAR()
  FI
  X=38
  STC=32
  TOPSCRN()
  Position(38,2)
  Print("L")
RETURN

PROC LEV()
  PokeC(88,SCN)
  Poke(83,39)
  Position(33,0)
  Print("  ↔↔")
  PrintB(L)
  Poke(83,0)
  Poke(88,SLO)
  Poke(89,SHI)
RETURN

PROC SCROLLDN()
  Color=CLR
  Plot(X,Y)
  FOR S=1 TO 6 DO
    FOR I=0 TO 15 DO
      Poke(54277,I)
      FOR J=1 TO 150 DO OD
    OD
    Poke(54277,0)
    SLO==+40
    IF SLO<40 THEN
      SHI==+1
    FI
    Poke(DL+12,SLO)
    Poke(DL+13,SHI)
    Poke(88,SLO)
    Poke(89,SHI)
  OD
  L==+1
  LEV()
  CLR=Locate(X,Y)
RETURN

PROC SCROLLUP()
  Color=CLR
  Plot(X,Y)
  FOR S=1 TO 6 DO
    Poke(54277,15)
    SLO=SLO-40
    IF SLO>215 THEN
      SHI==-1
    FI
    Poke(DL+12,SLO)
    Poke(DL+13,SHI)
    Poke(88,SLO)
    Poke(89,SHI)
    I=15
    FOR P=1 TO 16 DO
      Poke(54277,I)
      I==-1
      FOR J=1 TO 150 DO OD
    OD
  OD
  L==-1
  LEV()
  CLR=Locate(X,Y)
RETURN

PROC GETNPRNT()
  STC=GetD(1)
  IF STC<33 OR STC>'K THEN
    STC=32
  FI
```

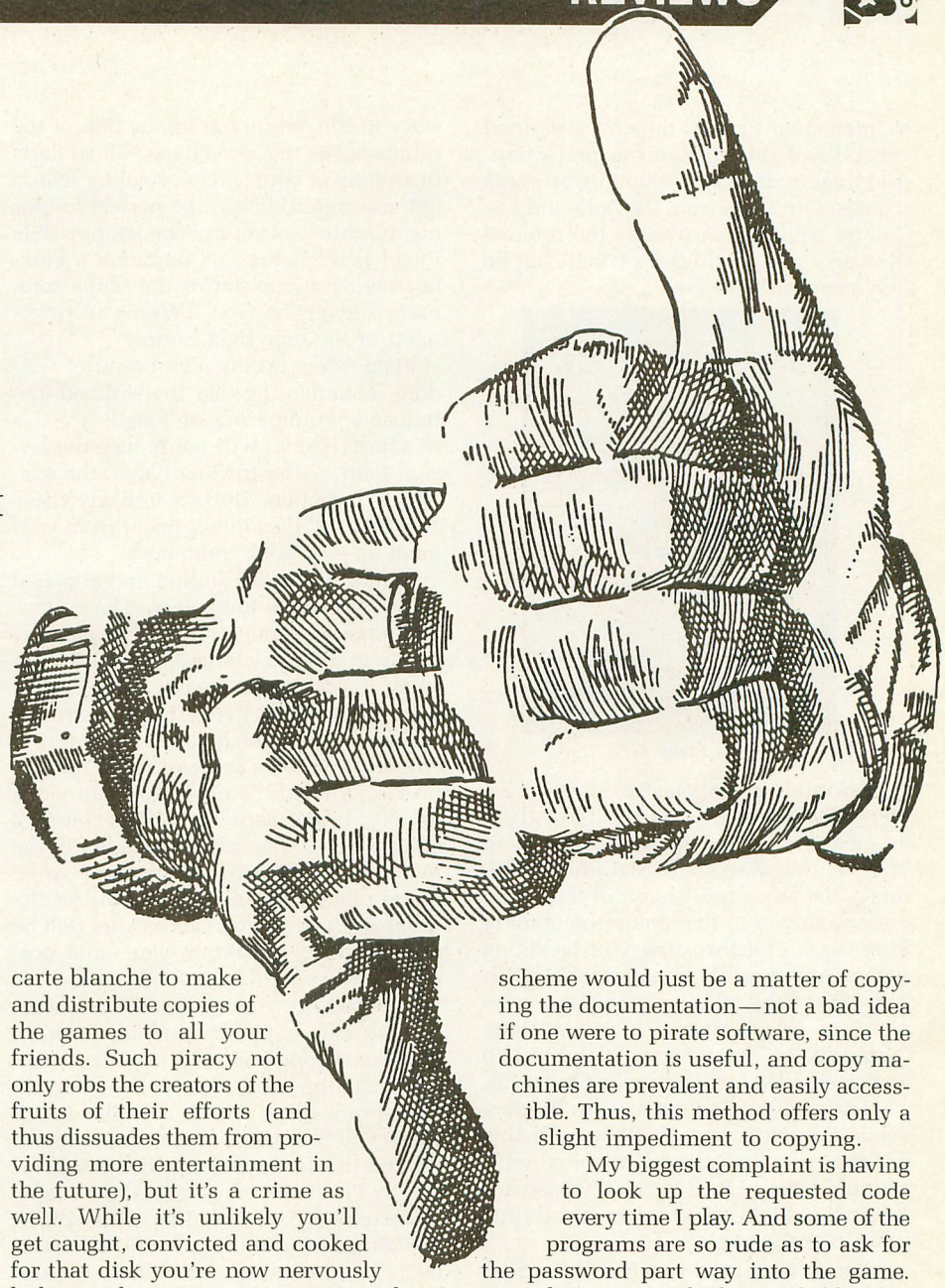
```
  IF STC>'F THEN
    STC==+128
  FI
  Color=STC
  Plot(X,Y)
  CLR=STC
  STG=0
  IF X<39 THEN
    X==+1
    CLR=Locate(X,Y)
  FI
  Color='L
  Plot(X,Y)
  Poke(764,255)
RETURN

PROC CHOOSE()
  Color=CLR
  Plot(X,Y)
  PokeC(88,SCN)
  TOPSCRN()
  Y=2
  CLR=Locate(X,Y)
  Color='L
  Plot(X,Y)
  DO
    S=Stick(0)
    X1=0
    IF S=11 AND X>0 THEN
      X1=-1
    FI
    IF S=7 AND X<39 THEN
      X1=1
    FI
    IF S<>15 AND (X1<>0 OR Y1<>0) THEN
      S=Locate(X+X1,Y+Y1)
      Color=CLR
      Plot(X,Y)
      X==+X1
      Color='L
      Plot(X,Y)
      CLR=S
    FI
    FOR J=1 TO 2000 DO OD
    IF STrig(0)=0 THEN
      STC=CLR
      FOR J=1 TO 255 DO
        Sound(0,J,8,15-J/17)
      OD
      STG=0
    FI
    IF STC='N THEN
       DISK()
    FI
    UNTIL Stick(0)=13
  OD
  TOPSCRN()
  Poke(88,SLO)
  Poke(89,SHI)
  Y=0
  CLR=Locate(X,Y)
  Color='L
  Plot(X,Y)
  IF STC='M THEN
    STG=1
    STC='3
  FI
RETURN

PROC MAIN()
  SETUP()
  Poke(88,SLO)
  Poke(89,SHI)
  Color=32
  For J=0 TO 8 DO
    P=Locate(0,J)
    Plot(0,J)
  OD
  FOR J=3000 TO 3079 DO
    Poke(SHI*256+J,10)
  OD
  Poke(SHI*256+2954,47)
  Poke(SHI*256+2955,48)
  Open(1,"K:",4,0)
  CLR=32
  DO
    S=Stick(0)
    X1=0
    Y1=0
    IF (S=10 OR S=14 OR S=6) AND
       Y>0 THEN
      Y1=-1
    FI
    IF (S=9 OR S=13 OR S=5) AND
       Y<8 THEN
      Y1=1
    FI
    IF S<12 AND S>8 AND X>0 THEN
      X1=-1
    FI
    IF S<8 AND S>4 AND X<39 THEN
      X1=1
    FI
    IF Stick(0)=14 AND Y=0 THEN
      CHOOSE()
    FI
    IF (X1<>0 OR Y1<>0) AND
       S<>15 THEN
      S=Locate(X+X1,Y+Y1)
      Color=CLR
      Plot(X,Y)
      X==+X1
      Y==+Y1
      CLR=S
    FI
    Color='L
    Plot(X,Y)
    IF Peek(53279)=6 AND SLO<>80 THEN
      SCROLLDN()
    FI
    IF Peek(53279)=5 AND SLO<>0 THEN
      SCROLLUP()
    FI
    IF Peek(764)<>255 THEN
      GETNPRNT()
    FI
    IF STrig(0)=0 AND STC<>'M THEN
      CLR=STC
    FI
    IF STrig(0)=0 AND STG<>0 THEN
      STC==+STG
      IF STC='9 THEN
        STG=255
      FI
      IF STC='3 THEN
        STG=1
      FI
    FI
    FOR J=1 TO 2000 DO OD
  OD
RETURN
```

Panak strikes!

Reviews of the latest software.



by Steve Panak

Let's talk about copy protection. Or, rather, the lack of it.

I've noticed that quite a few companies have exorcised the curse of copy protection from their entertainment software. I'm sure you know the curse. It often manifests itself in the 49th hour of a 50-hour adventure game when, during the final play session—and often with spectators gathered for the climactic slaying of the dragon—the disk fails to load properly. Despite numerous retries (and perhaps flexing and/or blowing on the disk), rather than felling an awesome beast, the gallant warrior is himself struck down—by a few bytes of corrupt data. This horror is slowly being rectified.

The companies have been somewhat sympathetic, while trying to balance customer relations (as well as the cost and burden of disk replacement) against the very real danger of piracy. And they have come up with a number of innovative, but oft-times clumsy solutions. Some trusting souls just allow unlimited copies to be made. That is, until the realities of the marketplace sink them deep in a sea of red ink, because only a fraction of their programs are actually being paid for.

The survivors have come up with a number of alternatives, all tied around the basic theme of requiring some sort of unique user input. These alternatives allow the disk to be copied, but don't give carte blanche to make and distribute copies of the games to all your friends. Such piracy not only robs the creators of the fruits of their efforts (and thus dissuades them from providing more entertainment in the future), but it's a crime as well. While it's unlikely you'll get caught, convicted and cooked for that disk you're now nervously hiding under your mattress, it *is* stealing, which is something I have trouble sleeping with. Unfortunately, the practicality and effectiveness of these methods most likely give their designers and users nightmares, equally.

Probably the best method employed requires that a small device be placed in one of the joystick ports. Assuming the game doesn't require all such ports for play, this is the least annoying, and does the least possible evil. While I'm sure someone can duplicate whatever's in that little port plug (have I coined a term?), it looks like a pretty tough job.

Another solution requires some random user input or password obtained from the documentation. Of course, defeating this scheme would just be a matter of copying the documentation—not a bad idea if one were to pirate software, since the documentation is useful, and copy machines are prevalent and easily accessible. Thus, this method offers only a slight impediment to copying.

My biggest complaint is having to look up the requested code every time I play. And some of the programs are so rude as to ask for the password part way into the game. Most distracting. Which is probably what this rambling of mine is getting to be. So let's get on with the first game.

Just remember: if you see something you like here, support the people responsible for bringing it to you. The cost of good software is really quite reasonable.

Stationfall

by Steve Meretzky
INFOCOM
125 CambridgePark Drive
Cambridge, MA 02140
48K Disk $34.95

Infocom has a long history of innovation. Okay, so the company has only been around for about eight years. But in those years, as the first (and foremost) producers

of interactive fiction, they have defined and refined this medium to perfection. And they've done so without resorting to sequels. Oh, there were the Zork and Enchanter trilogies, as well as the interactive version of Hitchhiker's Guide, but no true sequels. Until now.

Stationfall.

Stationfall is the latest work by Steve Meretzky, whose twisted mind has given us such visions as Leather Goddesses of Phobos and Planetfall. **Stationfall** continues the adventure begun in the latter, promoting you to the rank of Lieutenant First Class and thrusting you headlong into an exciting life of paperpushing. In fact, your first mission requires you to travel to Space Station Gamma Delta Gamma 777-G 59/59 Sector Alpha-Mu-79 to pick up a load of Request for Stellar Patrol Issue Regulation Black Form Binders Request Form Forms. To lighten up this mission, you find yourself reunited with your old friend Floyd, a robot whose antics have made it one of Infocom's all-time favorite characters.

After solving (and surviving) the first in a series of entertaining and logical puzzles, you leave in a spacecraft with your loyal pal. Upon arriving at your destination—a space station with a seedy colony attached—you find the entire place deserted, except for a strange alien spacecraft and the mummified remains of one of its crew members. Upon further exploration, you discover another robot named Plato, and he and Floyd become what will probably be known as the greatest comedy team ever in interactive fiction. After hours of careful exploration and clever problem solving, you will (if you're lucky) discover the secret of the strange saucer.

The prose in this game is up to Infocom's high standards. Detailed descriptions of the various locations bring the story to life, while the interaction of the robots keeps the game lively. Floyd darts in and out of your screen, stealing scenes left and right. Plato is the perfect foil for his mischievous antics. The irrepressible Floyd is scrawling his name on a bulkhead one minute, staring out of the computer screen the next, defying infringement of the copyright notice.

Plato offers up his own brand of wisdom, commenting that the stubbed-toe-induced whimpering of Floyd is ". . . a reaction [which] will not reduce the level of pain." After tickling Floyd, the mishap is forgotten. But it's unlikely these characters will soon disappear from your memory—or your computer's.

I found the space station and attached colony great fun to explore. The colony, a strange amalgamation of "Gunsmoke" and *Star Wars*, contains a large variety of small rooms, with many gadgets and devices. Some of the bad points are the unavailability of the *X* abbreviation for examine (I've grown accustomed to that abbreviation) and the repetitive nature of a lot of Floyd's activities. These are, of course, due to the limited memory of the machine. While it may seem lame when Floyd plays with his paddle ball for the umpteenth time, you can be sure that he will ultimately entertain you—and possibly save your life.

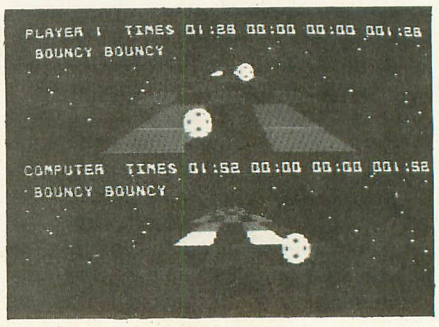
The packaging continues the use of Infocom's new book-box. Included within the nested container are a technical manual (containing information on program operation), three forms, a Stellar Patrol Patch and blueprints of the space station. I found this latter packet to be most helpful, as I'm lazy and hate to draw maps. Unfortunately for me, the prints didn't contain a layout of the parasitic colony attached to the station. You'll have to carefully chart out its many dirty and winding alleys and corridors if you want any chance of escaping from them. Finally, a mail-in coupon allows you to experience Planetfall for a special price, if you've not already had the pleasure.

Generally, I don't like sequels. This is because, by their very nature, they cash in on (and ultimately degrade) the good name of their predecessors—usually without delivering anything new. But **Stationfall** delivers a lot of "new," in the form of a great plot, setting and characters. The main similarity between the two games is Floyd, your robot helper, and he ends up providing more than his share of entertainment. Fall into **Stationfall** and you'll have trouble climbing back out.

TrailBlazer

MINDSCAPE
3444 Dundee Road
Northbrook, IL 60062
48K Disk $29.95

One thing missing in most arcade software is originality. Uniqueness. Although many games display a certain amount of this elusive quality, they usually end up being enhancements; different slants on the same, tired old themes. Rarely does a game break completely new ground,



TrailBlazer.

changing the way we play with our computers, like Pong or Pac-Man did. Such a piece of software in the arcade genre is even rarer.

TrailBlazer may not be the next Pac-Man, in terms of sales, but it is a unique, new arcade game which immediately grabbed my attention—then held it for an extended period. The double-teaming of an addictive quality and a simple concept is a one-two punch that will glue you to your monitor for days.

Envision yourself as a small ball with the ability to speed up, slow down and jump at will. You're on a vast, endless, checkerboard-like racetrack, suspended (somehow) in space. Now, put holes in the racetrack, add special effect squares, and you'll find yourself starting to get the idea behind **TrailBlazer**.

After booting up, you choose a game to play from a number of one- and two-player options. In the arcade game, you're limited to seven jumps per course, each course having a time limit for completion. Your goal is to complete as many of the courses as you can, racking up the maximum points (points are earned for crossing squares, with additional points awarded for extra time remaining).

A bonus round, like the Simon electronic game, in which you try to remember, then repeat a flashing color sequence, is good for even more points. But it's so hard to reach this bonus round that, after a month, we still hadn't played it.

A trial game allows one person to practice on any of the tracks, while match play pits two players against each other—and the clock—in a no-holds-barred run for the checkered flag. While the latter option was my personal favorite, somewhere in the main menu should be a play option that suits everyone.

Using the joysticks, one or two players compete on a screen split horizontally in half. Control is very simple and easily learned. Forward on the joystick speeds you up; pulling back acts as a brake; left and right perform in a predictable manner. The button makes the ball jump. As you race down the track, a number of squares effect the ball in various ways: black holes swallow you up; red slows you down; green speeds you up; yellow bounces you; and purple reverses your controls. In the arcade version, a flashing square kicks you into warp speed. Yet even with all these features, the true star of this show is graphics.

The displays are crystal clear, the action fast moving, and the control swift and sure. While each player's ball races primarily on one of twin tracks, at times both balls are visible on one display. For instance, as you catch up to your opponent, you can see him in front of you. When he falls into a hole, he'll see you shoot by him on his display. Also notable is the way in which two balls can collide with each other, fighting for position.

Mindscape has adopted the record album style packaging many companies are using today. These are the most space efficient containers I can imagine. Inside you'll find a flippy with the Commodore version on the B side, and a simple and concise manual that completely explains the program's operation and includes control and scoring reference sheets.

I only had a few complaints about the game. While there are twenty-one different courses of varying complexity, which can be combined in groups of three, I would have liked more flexibility in selecting the courses I could play in each of the options. Also, a randomly generated course option, one that would always keep you guessing, would be nice.

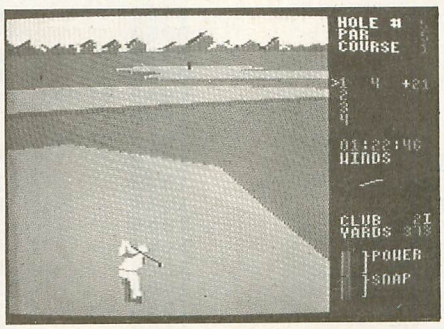
I found that the program won't let your ball fall from the side of the track; there's an invisible barrier acting like a guardrail, to protect you from the chasm—a little too safety-minded for me. Often, you have to go through the selection process after each race, making you wait too long to get back on the track. And, finally, high scores weren't saved to disk.

But these are only slight potholes marring the great track record of **TrailBlazer**. Overall, the game is the most original arcade action wristbuster to come down the pike in a long time, and one of the best two-player competition games I've seen. My main complaint is the severity of the blisters on my hands.

Leaderboard

by Bruce and Roger Carver
ACCESS SOFTWARE, INC.
#A 2561 South 1560 West
Woods Cross, UT 84087
48K Disk $39.95

I've been waiting for an 8-bit golf simulation. Seeing and playing golf games on various computer systems (especially the ST) had, until now, left me jealous of the bigger systems, dying to get my 800 out



Leaderboard.

on the course, but afraid I'd have to send it out to pasture. And I yearned to play golf, without regard to the realities of fickle weather. Now I can.

Leaderboard is the first 8-bit golf simulation I've had the pleasure to examine. And, since it satisfies the basic requirements for playability, I consider it the best one available. This status, however, is not due entirely to its monopoly; it's because this game lets you do just about everything it should, without any unnecessary input or awkward control.

Golf diehards can play up to seventy-two holes on four different courses. (Unfortunately, most holes are pretty straight.) These courses are challenging. It will take considerable practice to score any pars (much less birdies).

Up to four players can compete in three difficulty levels. An amateur's ball is unaffected by the wind, although it may hook or slice. A pro has to deal with hooks, slices and wind, while a novice has none of these. This handicapping assures a close competition. After initial options are chosen, players move to the practice range.

Access has placed all control on the joystick—where it belongs. Moving the stick in various directions makes menu choices (club selection, aim and desired viewpoint). Pressing the button starts your backswing; releasing it, your power stroke. A final press snaps your wrists. This system simulates the control of a golf swing nicely, recognizing the components. A power gauge on the screen provides a timing stimulus.

Like the real game, a little too late on the wrist snap and you slice; a little early and you hook. Putting works in a similar way, with a magnified power gauge calibrated in feet and another indicator showing the slope of the green. But there's more to read here than just the greens.

There's a well written manual which completely explains control of this complex simulation. Numerous charts help illuminate screen displays, as well as the average ranges associated with various clubs. Also documented are layouts of the four courses. I was disappointed that, while the manual contained a section on the different methods of scoring, the program did not support items such as match play. Pen and paper must be resorted to when using these scoring methods.

Although the program is not copy protected, a special key must be inserted in joystick port 1 to get the game going. The hard part is determining which port is number 1 on the XE (it's the back one). While the key is just another thing (an important one) to lose, it's better than having your game on a single, uncopyable disk of limited lifespan.

The graphics are great, so sharp and distinct you see the splash as your ball hits the many water hazards. Realistic sound effects also enhance play. On the down side, I would have liked an editor for custom courses, and a save game feature was sorely missed.

Despite these divots, **Leaderboard** is a nice addition to everyone's software library—and a must for golf enthusiasts. Its only downfall is that it might simulate the game a little too well. As on the outdoor course, you still have to watch out for lightning.

I don't think anyone could go wrong with any of the above games. In fact, they're all so good I refuse to name a single best. Each is a class act in its own field. As for next month, we're not so lucky. But I'll wait till then to give you the bad news.

 continued from page 21

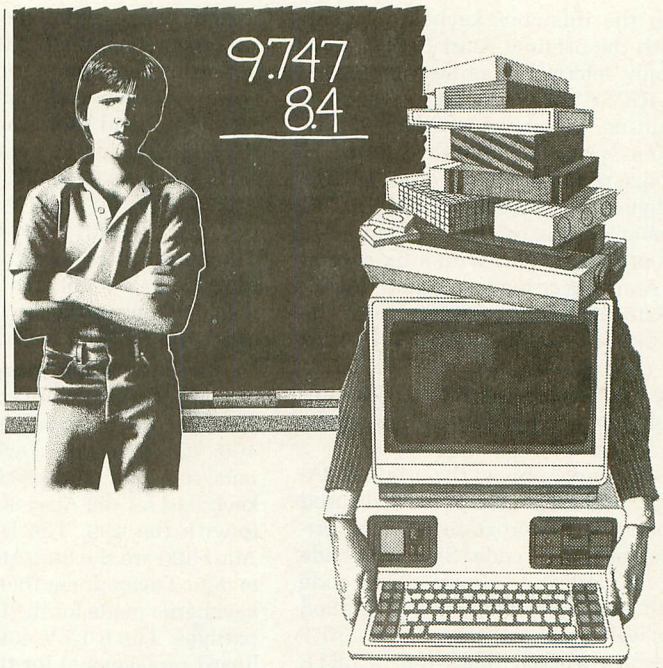
```
   ,203,200,169,255,145,203,200,192,74,20
   8,249,169,0,145,203,200,192
KX 905 DATA 128,208,249,152,24,101,203,13
   3,203,165,204,105,0,133,204,162,4,169,
   0,168,145,203,200,208,251
LL 906 DATA 230,204,202,208,245,165,206,1
   41,1,211,173,10,7,9,128,141,10,7,32,22
   4,7,165,20,201,64
NX 907 DATA 144,250,162,0,142,68,3,142,69
   ,3,142,72,3,142,73,3,169,11,141,66,3,1
   69,125,76,86
TA 908 DATA 228,2,243,1,243,1,160,210,193
   ,205,201,206,201,212,226,2,227,2,0,6
●
```

Listing 2.
Assembly listing.

```
0100 ; RAMDISK.COM WITHOUT
0110 ; MEM.SAV, DUP.SYS
0120 ;
0130 ROWCRS   = $54
0140 COLCRS   = $55
0150 ICCOM    = $342
0160 ICPTB    = $0B
0170 ICBAL    = $344
0180 ICBAH    = $345
0190 ICBLH    = $349
0200 ICBLL    = $348
0210 CIOV     = $E456
0220          *=$0600
0230 RAMINI   = *
0240          LDA #0          ;INIT,
0250          STA $13         ;CLOCK,
0260          STA $14         ;BYTES.
0270          LDA #14         ;COLUMN ,
0280          STA COLCRS      ;FOR MSG.
0290          LDA #8          ;ROW FOR,
0300          STA ROWCRS      ;MSG ALSO,
0310          STA ICBLL       ;LENGTH.
0320          LDA #0          ;LEN-HI,
0330          STA ICBLH       ;ALSO,
0340          TAX             ;IOCB NUM.
0350          LDA #MSG&255    ;SET MSG,
0360          STA ICBAL       ;ADDRESS,
0370          LDA #MSG/256    ;IN EDITR,
0380          STA ICBAH       ;IOCB0.
0390          LDA #ICPTB      ;PUT ,
0400          STA ICCOM       ;MESSAGE,
0410          JSR CIOV        ;WITH CIO.
0420          LDA $D301       ;STORAGE,
0430          STA $CE         ;CNTL.
0440          LDA #$EB        ;BANK 2
0450          STA $D301       ;MEM CNTL
0460          LDA #29696&255  ;VTOC LO
0470          STA $CB         ;POINTER
0480          LDA #29696/256  ;VTOC HI
0490          STA $CC         ;POINTER
0500          LDY #0
0510 VLOP1    =   *           ;FIRST,
0520          LDA MASK,Y
0530          STA ($CB),Y     ;SECTION,
0540          INY
0550          CPY #5          ;OF
0560          BNE VLOP1
0570          LDA #0          ;VTOC,
0580 VLOP2    =   *
0590          STA ($CB),Y     ;IN,
0600          INY
0610          CPY #10         ;DATA,
0620          BNE VLOP2       ;MASK.
0630          LDA #$0F
0640          STA ($CB),Y
0650          INY
0660          LDA #$FF
0670 VLOP3    =   *           ;FREE,
0680          STA ($CB),Y     ;SECTORS.
0690          INY
0700          CPY #55
0710          BNE VLOP3
0720          LDA #0          ;DIR.,
0730          STA ($CB),Y     ;AND,
0740          INY             ;VTOC,
0750          LDA #$7F        ;IN,
0760          STA ($CB),Y     ;USE.
0770          INY
0780          LDA #$FF        ;MORE,
0790 VLOP4    =   *           ;MORE,
0800          STA ($CB),Y     ;EMPTIES.
0810          INY
0820          CPY #74
0830          BNE VLOP4
0840          LDA #0
0850 VLOP5    =   *           ;FINAL,
0860          STA ($CB),Y     ;SEGMENT,
0870          INY             ;IN USE.
0880          CPY #128        ;NON-EXIST
0890          BNE VLOP5
0900          TYA             ;128 IN A
0910          CLC
0920          ADC $CB         ;POINTER,
0930          STA $CB         ;AT,
0940          LDA $CC         ;DIRECT.
0950          ADC #00
0960          STA $CC
0970          LDX #4          ;X256=1024
0980          LDA #0
0990 DLOP1    =   *
1000          TAY             ;LOOP 256
1010 DLOP2    =   *
1020          STA ($CB),Y     ;ZERO,
1030          INY             ;WHOLE,
1040          BNE DLOP2       ;DIRECT.
1050          INC $CC         ;NEXT,
1060          DEX             ;PAGE,
1070          BNE DLOP1       ;LOOP.
1080          LDA $CE         ;NORMAL,
1090          STA $D301       ;BANK.
1100          LDA $70A        ;DRVBYT
1110          ORA #$80        ;DRIVE 8
1120          STA $70A
1130          JSR $7E0        ;INIT D8:
1140 TIMER    LDA $14         ;SHORT,
1150          CMP #$40        ;DELAY FOR
1160          BCC TIMER       ;MESSAGE.
1170          LDX #0          ;IOCB0,
1180          STX ICBAL       ;SET UP,
1190          STX ICBAL+1     ;FOR PUT,
1200          STX ICBLL       ;ONE BYTE,
1210          STX ICBLL+1     ;ROUTINE.
1220          LDA #ICPTB      ;PUT BUFF,
1230          STA ICCOM       ;COMMAND.
1240          LDA #$7D        ;CLEAR,
1250          JMP CIOV        ;SCREEN.
1260 ;
1270 MASK     .BYTE 02,$F3,01,$F3,01
1280 MSG      .BYTE "■RAMINIT"
1290          .END
●
```

Then and now

The 8-bit Atari computer.



by Matthew J.W. Ratcliff

Back in 1984, Atari's 800XL was something of a wonder. Unlike the 1200XL—which was unanimously rejected by all developers—the 800XL had quite a bit to offer over the "Old Faithful" Atari 800 model.

First, the 800XL was far more compact. The entire computer was built on a single circuit board, as opposed to the ten boards of the Atari 800. Fewer parts generally indicate a lower price and higher reliablity for any product. The original 16K Atari 800 sold for about $800, while the newer 800XLs were going for under $200—with a full 64K of memory.

The extra RAM of the 800XL was difficult to get at, and couldn't be used as part of BASIC memory. However, it did set a "standard," since the extra RAM and the "bank switching" technology were produced by Atari. There were extra memory boards for the 800, but none were ever produced by Atari, thus, there were no standards. The boards were expensive and had little or no software support.

It wasn't long before OSS came out with a version of DOS XL that would remain resident in memory at all times, and still give you about 4K more RAM to play with while programming in BASIC. This neat trick was done by hiding a large portion of DOS XL "under" the operating system ROMs in an extra bank of RAM. SpartaDOS, which went unnoticed for some time on the older 800, started using similar techinques. It even gave users an 8K (62-sector) RAMdisk that was called RDBASIC.COM.

The DOS 3 fiasco.

During this same period, Atari released their 1050 disk drives with the "new and improved DOS 3." I've never understood why Atari placed a true double-density "mechanism" (drive chassis) in the 1050, but only had enough electronics to support "enhanced density." The result is a wasted 50K of storage on every disk!

Fortunately, ICD wasn't long in developing the US (Ultra Speed) Doubler. Not only did it give your drive the extra memory to handle full double-density sectors, but it improved the speed of the drive significantly when using SpartaDOS in the US sector skew mode. More recently, Duplicating Technologies has come out with a high-speed density doubler of their own, though it looks as if it's simply a clone of the one produced by ICD.

While ICD and OSS were being innovative, Atari gave new users a DOS that was completely incompatible with *anything* ever produced for the 8-bit Atari. Instead of 128-byte sectors, DOS 3 gave us 1000-byte blocks. This resulted in a total of 128 blocks on an enhanced-density DOS 3 disk, or about 128K of storage.

My speculation when it was released, in 1984, was that this DOS used a single byte sector pointer. Any sector reference less than 128 would refer to side one of the disk, and those 128 and greater to side two. "But Atari never released a double-sided version of the 1050 drive," you say. True, but they had planned a complete console system, the 1450XLD. It was even listed in their new XL product catalog. It was to have one or two double-sided drives, built-in modem and speech synthesizer, plus many other goodies. Unfortunately, this unit never made it to market.

Bill Wilkinson of OSS published "fixes" for Atari DOS 2.0S to make it run in enhanced-density mode (giving you 128K per disk), while maintaining compatibility with the older, more universal format. Not long after that, Atari came out with DOS 2.5, which gave you single density, or optional enhanced-density support, a RAMdisk for the new 128K Atari XE machine, and a utility to convert from DOS 3 to DOS 2.5 format. DOS 3 was dropped by Atari completely.

What ever happened to Atari DOS 3? Recently, I saw a bargain on floppy disks in a B & C Computer Visions Flier. The ad read, "Diskettes as low as 20 cents, 1000 for $200." It sounded too good to be true. (The lowest price I've ever gotten is 27 cents per disk). The fine print below revealed that these disks were "unnotched with DOS 3"!

The ABCs of Atari BASIC bugs.

One of the nicest features of the 800XL over its predecessors was that it had BASIC built in, though it could be disabled by pressing the OPTION key at boot time, making 8K more RAM available to your assembly language programs.

I'm sure most of you "old timers" re-

member the infamous keyboard lockup bug with the original Atari BASIC, now commonly referred to as Revision A (or simply REV A). The bug was associated with editing. Whenever you deleted program lines, or parts of lines, resulting in a code size reduction that was a multiple of 256 bytes, the computer locked up. Bill Wilkinson diagnosed the problem early on and provided Atari with a fix several times. Atari never made another "mask" for the BASIC ROMs, however, until the 800XL was developed and REV B Atari BASIC came about.

Apparently, several other fixes were implemented, which just caused more problems.

Whoever programmed the original BASIC fix, must have looked at the code used for expanding the program when inserting lines of BASIC code. Since the code looked the same as the buggy delete code that had just been fixed, it was applied there as well—a big mistake. The INSERT code always worked fine—the DELETE code had the problem. So the keyboard lockup didn't go away; it just moved. Since you delete lines of BASIC code more often than you insert them, this particular bug doesn't bite as often in REV B.

The result of these extra fixes was a major headache for over 100,000 800XL buyers. The most common problem was that every time you saved a BASIC program from this REV B BASIC, your file expanded by 16 bytes. This "growth" was cumulative. If you loaded and saved the same file enough times, you'd run out of memory.

There is a simple way around this problem: just LIST, then ENTER your files. But sometimes, after entering a large BASIC file, you get an *ERROR 9* (string not dimensioned), on the very line where the DIM statement occurs! Usually you can recover by "editing" the program (anything to change its length), or by doing a SAVE and LOAD.

These bugs were documented *early* in the production cycle of the 800XL computers; Atari was well aware of them. However, the machine was apparently manufactured for a year or more with the defective REV B BASIC, even though the REV C was completed shortly after the REV B bugs were revealed. Apparently, Atari had a large stock of the REV B ROMs and didn't feel the bugs were significant enough to discard them in favor of REV C. It was about this time that Atari was in dire financial straits—the year it lost nearly one-half *billion* dollars.

Atari began selling the REV C cartridge to consumers for only $15. It was helpful, but didn't seem fair. I did some PEEKing around one day and found that Atari BASIC REV B and REV C are virtually identical, except for 12 bytes!

Do you have an 800XL? Do you have revision B BASIC? To find out, try this:

```
PRINT PEEK(43234)
```

The current REV C returns a *234*, while REV B has a *96* at this location. If your machine returns the *96*, then by all means order the REV C cartridge from Atari.

Keyboards and power supplies.

The only computer I could afford when I started in this business was the Atari 400. Its membrane keyboard drove me nuts, so I soon purchased a "replacement" keyboard for the Atari 800 and interfaced it with the 400. The keyboards for the Atari 800 are the *best* Atari has produced to date. I even prefer them over the "best" keyboards made for the IBM PC and compatibles. The B-KEY 400 membrane keyboard replacement for the Atari 400 was popular for a year or two, but had its own problems. It didn't take long for the keytops to develop a lot of friction from accumulated dirt, so they required regular cleaning.

When the 800XL came out, there were apparently two different commonly used keyboards. One had a nice, smooth feel; the other had very wobbly keytops and a great deal of friction in the key mechanisms. It could slow down a good typist by 10 to 20 words per minute.

During the height of Atari's game production years, I started getting calls from friends about 800 and 800XL keyboard problems, especially with the space bar. The first question I always asked was, "Have you been playing Defender?" Most of the time I'd get a surprised reply: "Yes! How did you know?" In Defender, you launch your "smart bombs" with the space bar. Normally, this is a last resort strategy. In the frenzy of a hot shootout, it's easy to bang the poor space bar to death.

Atari will be rereleasing Defender later this year for the new XE Game System. The new version will accept either the space bar or one of the console keys to launch a smart bomb. If you get this version, use the console key—it's difficult to pound it as hard as the space bar.

With the release of the XE computer line came a new keyboard of dubious quality. The tactile feedback was less than desirable, with a very mushy feel. The bases of the keys were made of a conductive coated rubber, and seemed to have a pretty high failure rate with heavy use. Keyboards are not repaired at service centers; they're replaced with new ones—when they're available. Not all XE keyboards are constructed exactly like this, however. It appears that several versions (all with about the same tactile feel) for the XE series are in use.

Early on, I noticed the 800XL power supply ran awfully hot. It was in a solid black case with no ventilation, and was quite hefty. Mine burned up after just two months of use. I promptly built a new one, and tore the old one apart. That was no small task, since the entire transformer and circuitry were encased in a solid plastic resin, a good electrical insulator and an excellent thermal insulator.

The efficiency with which a power supply can shed heat generated from regulating the output voltage largely determines the longevity of the supply. My local dealer always keeps spare 800XL power supplies in stock. Atari has gone through two or three versions of XL/XE power supplies to date. The latest is in a larger case, well ventilated, capable of providing 5 volts at 1.5 amps, and 50 percent more power capacity than the original. This was done to support the extra RAM of the 130XE, and is adequate for all the RAM upgrades available for the XL/XE machines.

New and improved operating system.

Shortly after Atari came out with the 800XL, they released the translator disk. Several other translators have been making the rounds over the past two years as well. We should never have needed them, however. Here's why:

In the early days of the Atari, some hacker types discovered two "illegal" entry points in the system. Called EOUTCH and EGETCH in *Mapping the Atari*, these illegal entry points would print a single character to the screen and get a single character from the keyboard. These locations never changed throughout the life of the 800/400 systems, or during two revisions of the operating system. Because the illegal entry points were documented in the magazine, many people assumed this was a "safe place" to do a quick and dirty screen write and keyboard read.

Unfortunately, when the 800XL came out, these locations moved. Everyone pointed the finger at Atari, and blamed them for coming out with a new, "incompatible" computer. But, as anyone who's read Atari's Technical Reference guide for the 8-bit knows, these illegal entry points

were never documented by Atari. Bill Wilkinson's "Insight Atari," column in *Compute!* showed users how to perform these functions "legally" with calls to the CIO (Central Input/Output) utility in the operating system. With the CIO and the proper setup code, you can legally perform I/O with any device on the Atari, including screen, keyboard, cassette, printer and disk.

It wasn't Atari's fault that a lot of public domain and some good copyright covered software (written by irresponsible programmers) wouldn't run on the new XL and XE machines. Atari helped bridge the gap with the translator disk (at a $10 charge), until the old code was replaced with newer and better programs.

Setting up a CIO call to do the exact equivalent of the EOUTCH and EGETCH routines only takes six lines of assembly code each. It's unfortunate that, around the time the XLs came out, *Compute!* brought out "Assembly Language for Beginners" using the *illegal calls!*

The Turbo XL.

One of the "improvements" in the 800XL was a little card edge connector called the parallel bus. Actually, there already was a parallel bus connector in all the 400/800 computers produced. It was used to interface the mother boards to a mainframe computer system for "burn in" testing during the manufacturing process. Once the 400/800 computers were assembled, this connector was hidden underneath the ¼-inch aluminum RFI shield.

The connector on the 800XL was special, however. It provided some additional control lines for hooking parallel devices to the computer. They could be DMA (Direct Memory Access) for transferring up to 100K of information per second. That's mind boggling, when you consider that the Atari 800XL has only 64K to begin with.

I had hoped Atari would come out with a hard drive interface or 80-column adapter card to make use of this connector. Many thought it would never happen, and discarded their new XL OS ROMs in favor of some "upgrades" that made the 800XL compatible with the old 800 system. They ignored the fact that eliminating the XL OS was throwing out all support for the Parallel Bus Interface (PBI).

Well, a couple of guys at ICD were making a little money off the ever-growing popularity of SpartaDOS. But they wanted to do something different, exciting—something that could be called the "ultimate" for the 8-bit. After a lot of talk and design work, they created the MIO (Multiple Input/Output) board. For starters, this box uses the Atari parallel bus to connect your 800XL or 130XE to 1 meg of memory. This 1 meg of RAM can be configured as printer buffer and/or RAMdisks—extremely fast RAMdisks. The box also gives you an RS232 interface that's completely compatible with all software written for the R1: port of the Atari 850 interface.

While they were at it, they stuck a printer interface in there, too. Then, as if that wasn't spectacular enough, they added a hard drive interface connector with enough room to add an 80-column board later.

What innovations has Atari brought us? Well, the XEP80, an 80-column display board, is now beginning to make its way to reviewers' hands. It uses the Atari joystick port, a "bit-banging" serial interface. This configuration is slow, and probably not as reliable as the parallel bus.

Why didn't Atari use the parallel bus on its 800XL and 130XE computers for this product? Was it to give us continued support of the old 400/800 computers, which are now outnumbered by XL/XEs by approximately 8 to 1? Not at all. I think Atari really wanted to use the parallel bus on the XL/XEs, and begin to phase out support for the 400/800 machines, but they couldn't. You see, Atari "forgot" to put the parallel bus connector on the 65XE. Since this is a "current production" model, Atari was obligated to support it.

Enter the XE.

The Atari 130XE is a serious competitor for the ever-popular Commodore 128. In this machine, Atari has expanded memory to 128K. Since it's an "official" configuration (e.g., it comes from the Atari factory with 128K), it's supported by PaperClip 2, with an integral spelling checker. The Syn series has all been updated to give you more workspace for databases and spreadsheets, and—for general purpose use—all the popular DOSs for the Atari have been updated with additional RAMdisk support.

There could be even *more*, however. There are several popular RAM upgrades for the 800XL which take it up to a full 256K of memory. This upgrade is so popular, in fact, that PaperClip and a few other commercial applications support it—even though it's not an official version of Atari's 8-bit line. These upgrades are generally compatible with the 130XE bank switching technology, but provide more 16K banks of RAM to swap in and out.

Now that the Atari 130XE is so popular, why aren't games like Trinity (which uses Infocom's "interactive fiction plus" parsing technology) ported over to the 130XE? They run on the Commodore 128, so you'd expect to see them for the 130XE as well.

The problem is the "standard" disk technology. True double-density disk drives have been produced by Commodore for some time, and the newer game software requires all the disk storage it can get. Until Atari releases its own true double-density disk drive, additional support for the extra RAM of the 130XE will be limited by the disk drive itself.

Smaller is better?

This brings me to the latest "new technology" for the 8-bits: the 3½-inch drives we've been hearing about. Atari is one of the largest purchasers of 3½-inch drive mechanisms in the world, in support of the ST product line. In an admirable effort to bring you more "power without the price," Atari announced the development of a new 3½-inch disk drive for the 8-bit line. We 8-bit owners get more storage on a smaller disk (340K), and ST buyers get a better price because Atari is buying more 3½-inch disks.

The latest word from some of the Atari Fests is that this product has been dropped, before it ever got beyond the "vaporware" stage. It's never been unveiled at the computer shows, or photographed for the magazines, so where is it? It does seem to have been dropped.

Why would Atari drop it? Maybe it's because developers would have to ship *both* 5¼- and 3½-inch format disks to reach the greatest market share. I've heard that developers are not pleased with the prospect of the extra cost of producing software on "double disks," considering the low profit margin and high piracy factor they face at the outset. Many developers would completely *drop* their support of the Atari 8-bit before going that route.

Future hardware alternatives.

The 5¼-inch disk standard is universal. The 5¼-inch disks are much cheaper than the 3½-inch floppies, which is important to those with the less expensive 8-bit machines.

The extra RAM power of the 130XE is limited by its standard peripherals. It's time for a new disk drive, another 5¼-inch drive. It should be double-sided, double-density and a full 360K. These drives have been used in the IBM PCs for years now. Since the AT (which uses 1.2-

meg floppies) has gained in popularity—and gotten cheaper on the clone market—the PC XT clones and associated hardware are glutting the market. This includes those 360K drives, which are commonly advertised in trade journals for $60 or less. Better yet, how about putting an AT drive mechanism on the 8-bit Atari, with a full 1.2 meg of storage per floppy disk? These mechanisms are now selling for about $100.

I'm happy to report the latest word is that Atari has in fact dropped the 3½-inch 340K drive in favor of the XF551, a 360K 5¼-inch drive mechanism, which was shown at the June CES. This should be inexpensive for Atari to implement, in comparison to the 3½-inch format, since Atari already has 5¼-inch cases, and has developed double-sided 5¼-inch drive control technology in the past (for the 1450-XLD). So we should be seeing a full 360K drive for the XE line soon.

The new and improved, powerful A-DOS under development for the 3½-inch drives will run just as well on the 5¼-inch units. The drive will run on an older 400/800 system, but I hear the DOS may be XL/XE compatible only (requiring additional computer memory banks to run).

If you're hanging on to your old faithful Atari 800, you may need to go to SpartaDOS 1.1 to get full use out of one of the new 360K 5¼-inch drives from Atari. However, I can't be sure until the new drive and ADOS are released.

The 800XL continues to stand above the 65XE, particularly among loyal Atarians. The 256K RAM upgrade for the 800XL makes it a more powerful machine than the 130XE. The 130XE has been upgraded to 512K by hackers, but isn't nearly as simple to modify as the 800XL. Atarians are ready for more RAM, and the prices are right. The 130XE should become the "low end machine," with a new 256K or 512K XE to top off the line. The 65XE should be dropped, so that the entire current 8-bit line of machines have the parallel bus connector. Then maybe we'd see some really innovative hardware from Atari to place on this parallel bus.

It looks as if Atari wants to keep producing the current line, however, and leave the upgrades to ICD and talented hackers. If any more technology is added to the XEs, then a complete bundled system with drive and monitor will approach the base price of the 520STs.

Considering all the custom chip technology they've produced over the past few years, Atari appears to be serious about hardware. I've heard that they recently purchased a small chip manufacturing company in the Silicon Valley, but haven't been able to confirm it. Apparently, Atari wants to produce their own ST blitter chips, since schedules and yields have been less than desirable with outside firms. If it goes well, Atari may be more responsive in updating ROMs in the 8-bit and ST computers. Maybe we'll even see some new game programs on cartridge once again. After seeing the new XE Game System at CES, it looks as if we'll be seeing a lot of new software in cartridge format, with the new super bank switch technology.

Modems are getting cheaper all the time. A new 2400-baud modem chip set will bring 2400-baud modems into the realm of affordability (under $200) within a year or two—just as Atari is finally preparing to release their SX212, making this project another that may not make it to market.

One of the most interesting ideas never brought to fruition by Atari was the integral speech synthesizer for the 1450XLD. If speech were part of the hardware of the machine, it would be a standard part of the user interface in application, educational and game software for the 8-bits, for years to come.

The Alien Voice box was quite popular, considering its relatively high price. S.A.M. and other software-driven synthesizers were nice, but too memory hungry—and less than ideal when it came to sound reproduction quality, because of the 60Hz vertical blank interrupt. Turning off the VBI interrupt improves sound quality, but also shuts off the screen display. While little beeps, bops and boops are the norm for sound effects in IBM and Apple game programs, Atari has been way out in front for years with its four powerful sound registers.

The hype and hoopla has centered around the ST for about two years now. We've heard enough about blitters, math coprocessors and desktop publishing. The 8-bits are *serious* machines. The Apple II series is alive and well because of the innovations and continued compatibility Apple has provided for the II GS. If Atari rests on its laurels now, we're going to see many more 8-bits at garage sales—while a few of us 8-bit diehards continue to support technology (software and hardware) companies like COVOX, ICD, Epyx and MicroProse. With the introduction of the XE Game System, we should see more cartridge-based games and less piracy on the 8-bit Ataris. Let's hope that this generates a resurgence of interest in the Atari 8-bit product line, so the XE computer systems will continue to be enhanced by Atari and third-party developers.

MICHTRON

BBS 2.0	49.95
Business Tool	32.95
Cards	25.95
Calendar	19.95
Cornerman	32.95
DOS Shell	25.95
Echo	25.95
Eight Ball	19.95
Financial Future	25.95
GFA Basic	49.95
GFA Book	27.95
GFA Companion	32.95
GFA Compiler	49.95
GFA Draft	64.95
GFA Object	67.95
GFA Vector	32.95
Goldrunner	25.95
Hard Disk Backup	25.95
Karate Kid II	25.95
Major Motion	25.95
Make It Move	46.95
Match-point	25.95
M-Disk +	25.95
M-Cache	25.95
Mi-Term	32.95
Mighty Mail	32.95
Michtron Utilities	39.95
Personal Money Mgr	32.95
Pinball Factory	25.95
Realtizer.	150.95
Score Writer	25.95
Shuttle 2	25.95
Tanglewood	25.95
The Animator	25.95
Time Bandits	25.95
Trimbase	64.95
Tune Up	32.95

MUSIC

Bach Song Bach	19.95
Copyist 1.5	149.95
CZ Droid	64.95
CZ Patch	69.95
Digi-Drum	24.95
Dr. Drum	19.95
Dr. Keys	19.95
Dr. Patches	34.95
EZ Track	39.95
Fingers	34.95
Keyboard Contr Sequencer	149.95
Middiplay	32.95
Midi Maze	25.95
Music Construction Set	33.95
Music Studio	32.95
Pro Sound Designs	89.95
ST Replay	114.95

ST BUSINESS

A-Calc Prime	39.95
DAC Easy Accounting 2.0	64.95
Dollars and Sense	64.95
Financial Cookbook	14.95
Isgur Portfolio.	124.95
Inventory Master	67.95
Logistix Jr.	57.95
Logistix Sr.	89.95
Payroll Master	49.95
Micro Lawyer	39.95
Swiftcalc ST	48.95
Sylvia Porter	48.95
Tax Advantage	48.95
VIP Professional	CALL

ST ADVENTURES

Alternate Reality	26.95
Apshai Trilogy	14.95
Autoduel	32.95
B-24	25.95
Balance of Power	32.95
Bard's Tale	33.95
Beyond Zork	32.95
Black Cauldron	25.95
Breech	25.95
Colonial Conquest	25.95
Empire	36.95
Dark Castle	25.95
Defender of Crown	32.95
Deja Vu	32.95
Dungeonmaster	24.95
Hacker	17.95
Hacker II	25.95
Golden Path	29.95
Guild of Thieves	29.95
Jewels of Darkness	19.95
Leisure Suit Larry	32.95
Lurking Horror	25.95
Nord & Bert	25.95
Moebius ST	38.95
Kings Quest 1,2 or 3	32.95
Knight Orc	29.95
Mercenary	26.95
Ogre	25.95
President Elect '88	17.95
Phantasie 1,2 or 3	25 95
Plundered Hearts	25.95
Police Quest	32.95
Portal	32.95
Rings of Zilfin	25.95
Roadwar 2000	25.95
Roadwar Europa	29.95
S.D.I.	32.95
Shadowgate	32.95
Sinbad	32.95
Silicon Dream	19.95
Space Quest	32.95
StationFall	25.95
Starglider	29.95
Sundog	24.95
The Pawn	29.95
Tass Times	25.95
Thexedar	22.95
Tracker	29.95
Universe II	44.95
Uninvited	32.95
Ultima III or IV	38.95
Wizard's Crown	25.95
221 Baker Street	26.95

CASIO KEYBOARDS ..... CALL

ABACUS

Abacus Books	CALL
Assempro	39.95
Chart Pak St	32.95
Datatrieve	32.95
Electra - spell	25.95
Forth MT.	32.95
Paintpro	32.95
Powerplan	49.95
Textpro	32.95

DISKETTES

Maxell 3.5 SS/DD (10 PK)	12.95
Maxell 3.5 DS/DD (10 PK)	17.95

Note: Buy Diskettes at these low prices when added to any other order.

ST EDUCATIONAL

AB - Zoo	19.95
Animal Kingdom	24.95
All About America	36.95
Algebra 1 or 2	34.95
Arakis (each)	16.95
Arithmetic	34.95
Aesop Fables	31.95
Buzzword	27.95
Decimal Dungeon	24.95
First Shapes	33.95
Fraction Action	24.95
Invasion	19.95
Kid Talk	33.95
Kinderama	24.95
Math Wizard	24.95
Math Talk	33.95
Math Talk Fractions	33.95
Mathematicians Tool Kit	32.95
Magical Myths	31.95
Mavis Beacon Typing	26.95
Mother Goose	19.95
Read & Rhyme	24.95
Read-A-Rama	31.95
Space Math	25.95
Speller Bee	33.95
Trigonometry	34.95
Winnie the Pooh	16.95
1st Letters and Words	33.95

ADVANCED ORDERS

Adv. of Sinbad	31.95
Arabian Nights	31.95
Calculus	34.95
Discreet Math	34.95
Ghostly Grammers	31.95
Lands of the Unicorn	21.95
Logic Master	31.95
Planetarium	44.95
Probability	34.95
True Stat	34.95

ST DATABASES

Data Manager ST	48.95
DB Man	96.95
Regent Base 1.1	79.95
Superbase Gem	94.95
The Informer	67.95

DESKTOP PUBLISHING

Fleet ST Desktop Pub	74.95
Publish It	84.95
Publishing Partner	64.95
Partner Fonts	22.95
Parnter Forms	19.95

ST ACCESSORIES

Anti-Glare Screen	19.95
Dustcovers	CALL
Flip'n' File II- Micro	19.95
3.5 Drive Clean Kit	16.95
6 Way Surge Protector	19.95
6 FT SF 354/314 Cable	19.95
Mouse Pad	8.95
Mouse House	6.95

ST TELECOMM

QMI BBS ST	31.95
BBS Express	56.95
Deluxe Minicom	39.95
I.S Talk	33.95
Minicom	25.95
ST Talk Ver 2.0	CALL

ANTIC

A-Calc Prime	39.95
A-Chart	25.95
Architecural Design Disk	19.95
Base Two	39.95
Crystal	17.95
Cyber Control	39.95
Cyber Paint	44.95
Datamaps	17.95
Flash 1.5	19.95
Future Design Disk	19.95
G.I.S.T.	22.95
Genesis	49.95
Human Design Disk	19.95
LCS Wanderer	25.95
Maps and Legends	22.95
PHA$ar	64.95
Quicktran	22.95
Spectrum 512	49.95
Stereotek 3D Glasses	149.95
Stereo CAD 3-D	67.95
Shoot the Moon	25.95
The Cyber Studio	64.95
The Navigator	32.95
3D Developer's Disk	19.95
3D Font Package	17.95
3D Plotter & Printer Driver	17.95

ST GRAPHICS

Advanced Art Studio	29.95
Aegis Animator	48.95
Athena II	67.95
Degas Elite	52.95
Desktop Publishing Lib/ Graphic Artist	49.95
Graphic Artist	124.95
Easy Draw	64.95
Font Paki for Easy Draw	25.95
Font Pak for Graphic Artist	32.95
Font Editor for Graphic Editor	49.95
Neochrome	34.95
Paintworks	25.95
Personal Draw Art I	19.95
Pro Sprite Designer	39.95
ST Sprite Factory	25.95
ST Art Director	48.95
Technical Draw Art I	19.95
1st Cadd	31.95
3-D Graphics	34.95

ST WORD PROCESSORS

Microsoft Write	CALL
Regent Word II	48.95
Thunder	26.95
ST Becker Text ST	67.95
Word Perfect	229.95
Wordwriter ST	48.95
1st Word-Plus	57.95

INFOCOM ST

Infocom Invisiclues	CALL
Bureaucracy	25.95
Enchanter	19.95
Hitchiker	19.95
Hollywood Hijinxs	25.95
Leather Goddess	25.95
Lurking Horror	25.95
Moonmist	25.95
Stationfall	25.95
Trinity	25.95
Wishbringer	11.95
Zork I	25.95
Zork II or III	28.95

ST ARCADE GAMES

Airball	25.95
Aitball Construction Kit	17.95
Alien Fires	25.95
Arena	14.95
Atari Plane Tarium	29.95
Arctic Fox	26.95
Brattacus	32.95
Bridge 5.0	22.95
Barbarian	25.95
Boulder Dash Cons Kit	17.95
Breech Scenario Disk	17.95
Champ Baseball '86	25.95
Champ. Wrestling	14.95
Chessmaster 2000	29.95
Crystal Castles	20.95
Deep Space	19.95
F-15 Strike Eagle	27.95
Flight Simulator II	33.95
Female Data Strip Poker	16.95
Famous Course Disk 1 or 2	14.95
Gen Mgr/for MLB	19.95
Gone Fishing	26.95
Gauntlet	32.95
Gato	24.95
GFL Football	25.95
Gridiron Football	33.95
Guardians of Infinity	22.95
Hardball	24.95
Harrier Srike	24.95
Harrier Combat Simulator	32.95
Hunt for Red October	26.95
Indoor Sports	32.95
Into the Eagle's Nest	25.95
Joust	20.95
Karateka	22.95
Leader Board	25.95

ST UTILITIES

Back Pak	64.95
Desk Cart	72.95
Electro Calendar	35.95
Fast	31.95
Flashback	79.95
Labelmaster Elite	27.95
K-Switch	25.95
Micro Cookbook	32.95
Partner ST	44.95
PC Ditto	69.95
Smooth Talker	33.95
St Doctor	24.95
Tempus	32.95
Time Link	33.95
Video Wizard	39.95
Write 90	18.95

ST LANGUAGES

Alice Pascal	49.95
Cambridge Lisp.	139.95
Fast Basic (Philon)	48.95
ISO Pascal	69.95
Lattice C	99.95
LDW Basic 2.0	54.95
Mark Williams C.	114.95
Macroassembler	49.95
Modula II	67.95
Metacommco Make	49.95
Micro C shell	34.95
Modula II Developer	Call
MT C Shell	64.95
Personal Pascal 2.0	64.95
True Basic	49.95
True Basic Dev. Kit	34.95
True Basic Run Time	69.95

Little Computer People	16.95
Marble Madnss	24.95
Master Ninja	22.95
Mean 18	27.95
Metropolis 2000 st	22.95
Micro League Baseball	39.95
Night on the Town	22.95
Paper Boy	32.95
PinBallWizard	21.95
Psion Chess	38.95
Plutos	19.95
RPV	24.95
Roadrunner	32.95
Rogue	24.95
Q-Ball	19.95
Sentry	29.95
Scabble	26.95
Scruples	26.95
Shanghai	25.95
Silent Service	25.95
Skyfox	14.95
Super Cycle	14.95
Super Huey	25.95
ST Pool	22.95
Space Stat/Protector	25.95
Sub Battle Simulator	25.95
Super Bike Challenge	19.95
Super Star Hockey	32.95
Speed Buggy	29.95
Spy vs Spy 3	17.95
Star Raiders	20.95
Star Fleet 1	38.95
Star Fleet II	38.95
Strip Poker	25.95
ST Comes Alive	19.95
ST Karate	22.95
ST Wars	24.95
Tenth Frame	25.95
Temple of Doom	32.95
Test Drive	24.95
The Wanderer	25.95
Terapods	24.95
Tracker	29.95
Trailblazer	32.95
Two on Two Basketball	25.95
Uridium	25.95
Vegas Gambler	21.95
Vegas Craps	21.95
Video Vegas	24.95
Wargames Constuction Kit	22.95
Winter Games	14.95
Wiz ball	14.95
World Games	24.95
WWF Micro Wrestling	39.95
Xevious	19.95
3-D Helicopter	32.95
'86 Team K For MLBB	16.95

ST PRINT UTILITIES

Art Gallery 1 or 2	18.95
Award Maker	24.95
Certificate Maker	25.95
Fonts & Borders/P.M.	21.95
Deluxe Print 2	34.95
Library 1/Certificate Maker	21.95
Megafont ST	24.95
PM Interface	18.95
Print Shop	31.95
Printmaster Plus	24.95
Rubber Stamp	18.95
Typesetter Elite	31.95
220 ST	32.95

Easter Super Printer Package Sale

Atari ST

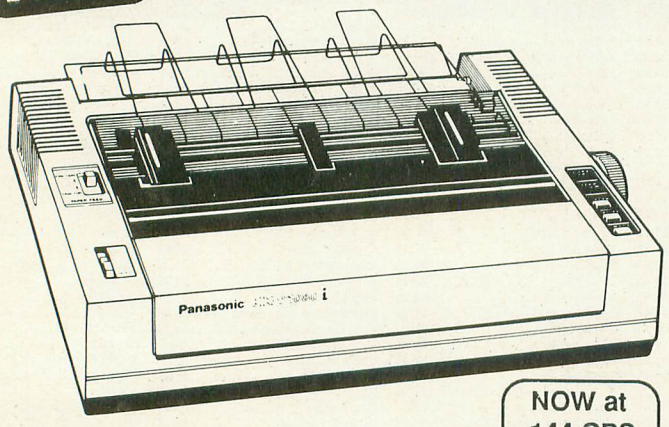
Panasonic Office Automation

8 Bit Atari

&

ComputAbility

NEW



NOW at 144 CPS

Panasonic Office Automation **KX-P1080i-II**

Super Printer Package
*for 8 Bit Atari computers

with Xetec Graphic AT **$215**

with Supra 1150 interface **$225**

*Package price includes Delivery in continental U.S.A.

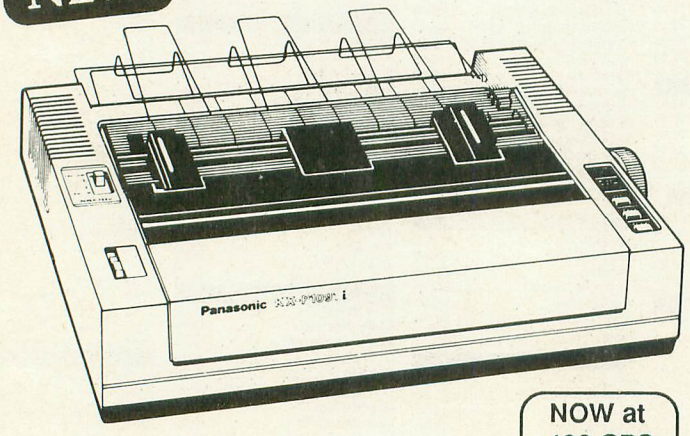
Panasonic Office Automation **KX-P1080i-II**

Super Printer Package
*for All Atari ST computers

with ST Printer cable **$199**

*Package price includes Delivery in continental U.S.A.

NEW



NOW at 192 CPS

Panasonic Office Automation **KX-P1091i-II**

Super Printer Package
*for 8 Bit Atari computers

with Supra 1150 interface **$245**

with Xetec Graphic AT **$235**

*Package price includes Delivery in continental U.S.A.

Panasonic Office Automation **KX-P1091i-II**

Super Printer Package
*for All Atari ST computers

with ST Printer cable **$219**

*Package price includes Delivery in continental U.S.A.

* Price for APO & Non Continental U.S.A. orders. See Special order information in our 2 page spread

Mon-Fri 9am-9pm CST
Sat 11am-5pm

order Call Toll Free
800-558-0003

SINCE 1982

ComputAbility Consumer Electronics

No surcharge for Mastercard or Visa

Inquiries, or for Wisc. Order
414-357-8181

CIRCLE #123 ON READER SERVICE CARD

INDEX TO ADVERTISERS

READER SERVICE #	ADVERTISER	PAGE #
114	Advan Language Designs	34
126	Alpha Systems	70
102	American Techna-Vision	2
131	B&C Computer Visions	66
122	Beckemeyer Development Tools	62
127	CMO	96, IBC
123	ComputAbility	63, 64 65
106	Computalk TCS	20
111	Computer Direct	30, 31, 32, 33
125	Computer Elite	70
110	Computer Games Plus	22
124	Computer Repeats, Inc.	34
134	Computer Serviceland	75
104	Electronic One	6
132	Future Systems Software, Inc.	14
101	ICD	IFC
112	I.M.P. Software	14
—	Jesse Jones	54
118	Lyco Computer	88, 89
117	Magna Systems	39
119	Megamax, Inc.	54
130	Micromiser Software	34
103	Microtyme	2
128	Mindscape	OBC
121	Newell Industries	62
109	S & S Wholesalers, Inc.	8, 9
107	Side-Line Computer	20
105	Sierra Services	6
115	Southern Software	34
120	subLOGIC Corporation	10
113	Terrific Peripherals	34
133	Thompson Electronics	75
108	Twenty-Fifth Century	20
116	Wizztronics	39

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.

The End User

THIS MONTH:

The great debate and constructive criticism.

Arthur Leyenberger is a human factors psychologist and free-lance writer living in New Jersey. He has been an Atari enthusiast for over five years. When not computing, he enjoys playing with robotic toys.

CompuServe — 71266,46
Delphi — NJANALOG

by Arthur Leyenberger

Just when you thought it was safe to sit down at the keyboard again, the nasty debate concerning STs vs. 8-bit Ataris rears its ugly head. We've been through this before. On the mild side, we have the 8-bit users steadfastly refusing to switch to the ST—*not* because the 8-bit machine is already fulfilling their needs, but because: (1) it's a new machine; (2) it's made by Jack Tramiel and company, which is really the old Commodore, etc.; or (3) the ST won't run 8-bit software.

On the more insidious side, we have readers of **ANALOG Computing** accusing us of promoting the doom of 8-bit computers, being in bed with Jack and company (I assume it must be a pretty *big* bed), and writing programs that destroy the 8-bit computers they're run on (hey, it was an April Fool's joke, read my lips: *J-O-K-E*).

Of course, the ST owners aren't free of blame, either. Those willing to get involved claim that the ST is the greatest thing since the invention of the wheel, and that *they* are in the vanguard and are trendsetters by purchasing such a wonderful product. Further, these (often) self-righteous ST users won't tolerate any criticism of the hardware, software, or, in some cases, even of Atari.

I see this unuseful debate occurring on CompuServe, on Delphi and at local user group meetings. There's no question that the debate is nonproductive. To quote Spock, both sides "proceed from false assumptions." The debate is further fueled by the discussion of the 8-bit emulator for the ST. Regarding the ST vs. 8-bit questions, the following thoughts seem appropriate.

First, why does buying an ST computer necessarily require that one dump one's 8-bit associated hardware and software? If nothing else, an 8-bit computer will *always* be a better means of running 8-bit software than will *any* 8-bit emulator.

Second, if one were to get an ST, why should one automatically replace (or think one has to replace) all one's 8-bit software? Sure, it would be smart to use the ST's power, speed and ease of use for such major applications as word processing. However, there are certain programs that run just fine with the 8-bit Atari's 6502 processor. So why replace them? And there are many programs that will *never* be ported to the ST, because the producer no longer exists, or there's no market for the programs, or for a number of other reasons. This is especially true of many games.

Third, neither I nor the **ANALOG** staff are recommending that 8-bit Atari users rush out and buy an ST just for the sake of new technology. If you have an 8-bit and some good software—*and* it fulfills your needs—by all means, keep it. I don't advocate upgrading to the ST unless you think it can do better or faster whatever it is you do with a computer.

There are certain features of the ST that, by definition, make it superior for specific applications. The best example is

in word processing. With an 80-column screen, an excellent black-and-white monitor (although medium resolution color is okay, too), the capability to handle large text files, and word processing software that can easily display multiple files in separate windows for editing, the ST outperforms 8-bit computers. Period. Now, if word processing isn't that important an application for you, then this argument has no merit in your case. See? If you need the features, fine; if you don't need them, fine. I'm easy.

Finally, for the amount of money you might get if you sold your 8-bit hardware and software, it may make more sense to keep the stuff, even if you *do* get an ST. In 1982, I paid $2000 for an Atari 800, full memory, one 810 disk drive and an Epson printer. I'd be lucky to get $250 for the same equipment if I were to sell it as used hardware.

The bottom line is that we all bought Atari computers, whatever model, to fulfill a need. As long as that need's still being satisfied, there's no reason to change. However, if something comes along that can better satisfy that need—and you can afford it—go for it.

Just doin' my job, man.

There's trouble in River City. Well, maybe not *that* much trouble, but in reading the mail that comes into **ANALOG Computing** and talking with users, Atari representatives and other vendors, I get a sense that misunderstandings occasionally occur. Specifically, some people tend to confuse critical questioning, *á la* Socratic method, with a negative attitude.

In my line of work (which is evaluating hardware and software for usability), I must constantly raise critical questions. Does the product fulfill its intended purpose? Does it meet the user's needs? Can the intended user figure out how to perform a particular task with the product? Does the perceived need for the product justify its existence? Etc., etc. The purpose is *not* to be negative with these questions, but to be constructively critical.

Sure, some of the questions might be provocative, but the point of raising them is to stimulate debate and thereby get at the core issues of whatever we're discussing. This process of raising questions is even *more* critical when the readers of **ANALOG** are going to rely on the information we provide to make their buying decisions. It's more important that we play devil's advocate, if we must, than not say anything and risk printing incomplete information.

Readers are not the only ones who seem to miss the point. Certain people at Atari seem to hear only what they want to. A typical example of this occurred during the Atari press conference at the January 1987 Consumer Electronics Show. During the question and answer session following the announcement of the Atari PC, I asked Sam Tramiel what plans Atari had for overcoming the "game image" many people have of Atari. (Long-time Atari users know we've been fighting this issue since long before Atari became the "new" Atari.) I was concerned that the popular perception might make it difficult to sell into corporate America. Sam's answer was that Atari was relying on people in the know to buy their machines within the companies.

I followed up by mentioning that it's the "guys with green eyeshades" (accountant types or data processing managers) who buy the computers in many companies, and they may not want to buy (1) from a toy store, (2) from a company with no corporate service contract, or (3) from a "game" company. Sam Tramiel replied by saying that he expected the "techies" to buy the machines.

From that press conference on, Neil Harris—and others within Atari—have thought that I'm "negative" and "anti-Atari." No! I am *pro*-Atari! That's why I'm concerned about the image and wanted to know what Atari would be doing to combat the problem.

Asking tough questions, out of loyalty, is *not* the same thing as being negative. Got it?

```
MN 3240 ? "ERROR ";PEEK(195);" AT LINE ";
   PEEK(186)+PEEK(187)*256:END
LF 3250 IF PEEK(195)=136 THEN GOTO 1150
RG 3260 GOTO 3240
AA 3270 GRAPHICS N0:POKE 710,N0:POKE 752,
   N1:SOUND N0,75,N12,N8:FOR X=N1 TO 50:N
   EXT X:SOUND N0,N0,N0,N0
OG 3280 POSITION N6,N8:? "TROUBLE WITH IT
   EMS!":POSITION N6,N10:? "NO ";Q$;C$;Q$
   ;" FOUND!"
YC 3290 POSITION N6,N12:? "PRESS ANY KEY"
AK 3300 CLOSE #N1:OPEN #N1,N4,N0,"K:":GET
    #N1,A:CLOSE #N1:END
TK 3310 IF PEEK(195)=136 THEN 1330
QW 3320 GOTO 3240
TO 3330 IF PEEK(195)=136 THEN 2320
RC 3340 GOTO 3240
UY 3350 IF PEEK(195)=136 THEN 1520
RI 3360 GOTO 3240
WG 3370 IF PEEK(195)=136 THEN 1710
RO 3380 GOTO 3240
TE 3390 IF PEEK(195)=136 THEN 2130
QS 3400 GOTO 3240
```

Listing 3.
BASIC listing.

```
JJ 50 A$="Jo!b!dbcjo":W=N6:RETURN
IQ 60 A$="Jo!uif!gspou!zbse":N=N7:W=N8:RE
   TURN
JF 70 A$="Jo!b!gjfme":S=N6:RETURN
NT 80 A$="Jo!b!gpsftu":N=N9:S=N8:E=N6:W=N
   9:RETURN
VI 90 A$="Jo!b!gpsftu":N=N8:S=N8:E=N9:W=N
   10:RETURN
UO 100 A$="Cftjef!b!ijmm":E=N9:RETURN
KJ 110 A$="Jo!b!dbwf":E=N10:RETURN
GJ 120 A$="Jotjef!uif!tife":W=N7:RETURN
AV 6302 IF IN$<>"EJBM!45.65.38" OR I(N3)
   OR R<>N5 THEN 6304
TP 6303 A$="Uif!dpncjobujpo!xpslt/":I(N3)
   =-5:UL=N1:GOTO N950
ZR 6304 IF IN$<>"EJH!HSPVOE" OR I(N15) OR
    R<>N11 OR I(N6)<>-N1 THEN 6306
YV 6305 A$="Zpv!ejh!tpnfuijoh!vq=":I(N15)
   =-11:UL=N1:GOTO N950
WK 6306 IF IN$="EJH!HSPVOE" AND R=N11 AND
    NOT I(N15) THEN A$="Zpv!offe!tpnfuij
   oh/":GOTO N950
CJ 6502 IF IN$<>"GJMM!MBOUFSO" OR I(N2)<>
   -N1 OR I(N10)<>-N1 OR I(N20)<>N0 THEN
   6504
MQ 6503 I(20)=-N1:I(N2)=N0:UL=N1:A$="Plbz
   -!ju(t!gvmm/":GOTO N950
XH 6504 IF IN$="GJMM!MBOUFSO" AND (I(N2)<
   >-N1 OR I(N10)<>-N1) THEN A$="Opu!zfu/
   ":GOTO N950
RC 6602 IF IN$="HP!TIFE" AND R=N7 AND ABS
   (I(N18))=R THEN A$="Plbz":R=N12:UL=N1:
   GOTO N950
JH 6604 IF IN$="HP!TIFE" AND R=7 THEN A$=
   "Ju(t!dmptfe!vq!ujhiu/":GOTO N950
IC 6606 IF IN$="HP!DBWF" AND R=N10 AND AB
   S(I(N14))=R THEN A$="Plbz":R=N11:UL=N1
   :GOTO N950
ZK 6608 IF IN$="HP!DBWF" AND R=10 THEN A$
   ="Zpv!dbo(u!ep!uibu!zfu/":GOTO N950
HL 7102 IF IN$="MPPL!QBSSPU" AND R=N5 THE
   N A$="Ju!mpplt!mjlf!b!qbsspu/":GOTO N9
   50
OR 7104 IF IN$="MPPL!DBCJOFU" AND R=N5 TH
   EN A$="Ju!ibt!b!mpdl/":GOTO N950
QI 7106 IF IN$="MPPL!MPDL" AND R=N5 THEN
   A$="Ju(t!b!dpncjobujpo!mpdl/":GOTO N95
   0
CB 7108 IF IN$="MPPL!TIFE" AND R=N7 AND A
   BS(I(N19))=R THEN A$="Tpnfpof!cpbsefe!
   ju!vq/":GOTO N950
AQ 7110 IF IN$="MPPL!MBOUFSO" AND I(N2)=-
   N1 THEN A$="Ju!offet!gjmmjoh/":GOTO N9
   50
NF 7112 IF IN$="MPPL!MBOUFSO" THEN A$="Zp
   v!epo(u!ibwf!ju/":GOTO N950
DO 7114 IF IN$="MPPL!LFSPTFOF" AND I(N10)
   =-N1 THEN A$="Mbcfm;!GPS!MBOUFSOT":GOT
   O N950
LI 7116 IF IN$="MPPL!LFSPTFOF" THEN A$="Z
   pv!ibwf!up!ibwf!ju!gjstu":GOTO N950
BL 7118 IF IN$="MPPL!EPPS" AND R=N7 AND A
   BS(I(N19))=R THEN A$="Uifsf(t!b!cpbse!
   pwfs!ju/":GOTO N950
OZ 7120 IF IN$="MPPL!SVCCMF" AND R=N10 TH
   EN A$="Uifsf(t!tpnf!cpvmefst/":GOTO N9
   50
SN 7122 IF IN$="MPPL!DBWF" AND R=N11 THEN
    A$="Uif!hspvoe(t!ejtuvsdfe/":GOTO N95
   0
KL 7124 IF IN$<>"MJHIU!MBOUFSO" OR I(N20)
   <>-N1 OR I(N4)<>-N1 OR I(N21)<>N0 THEN
    7126
GL 7125 I(21)=-N1:I(N20)=N0:UL=N1:A$="Zpv
   !mjhiu!uif!mboufso/":GOTO N950
HW 7126 IF IN$="MJHIU!MBOUFSO" AND (I(N20
   )<>-N1 OR I(N4)<>-N1) THEN A$="Zpv!off
   e!tpnfuijoh!npsf/":GOTO N950
VK 7128 IF IN$="MPPL!DIFTU" AND R=N11 AND
    I(N21)=-N1 THEN A$="Zpv!gpvoe!uif!nbh
   b$joft=":GOTO 13000
RW 7130 IF IN$="MPPL!DIFTU" AND R=11 THEN
    A$="Zpv!dbo(u!rvjuf!tff///":GOTO N950
HN 7402 IF IN$<>"PQFO!DBCJOFU" OR I(N2) O
   R R<>N5 OR ABS(I(N3))<>R THEN 7404
GQ 7403 A$="Uifsf(t!tpnfuijoh!jotjef=":I(
   N2)=5:UL=N1:GOTO N950
YH 7404 IF IN$="PQFO!DBCJOFU" AND R=N5 AN
   D  NOT I(N2) THEN A$="Ju(t!mpdlfe!vq/"
   :GOTO N950
GF 7702 IF IN$<>"SBLF!MFBWFT" OR I(N6) OR
    R<>N6 OR I(N9)<>-N1 THEN 7704
TO 7703 A$="Uifsf(t!tpnfuijoh!ifsf/":I(N6
   )=6:UL=N1:GOTO N950
SS 7704 IF IN$="SBLF!MFBWFT" AND R=N6 AND
     NOT I(N6) THEN A$="Zpv!dbo(u///zfu/"
   :GOTO N950
LX 7706 IF IN$<>"SFNPWF!CPBSE" OR I(N11)<
   >-N1 OR I(N18)<>N0 OR R<>N7 THEN 7708
XF 7707 I(18)=-R:I(N19)=N0:UL=N1:A$="///x
   ju!uif!ujsf!jspo/":GOTO N950
EX 7708 IF IN$="SFNPWF!CPBSE" AND I(N11)<
   >-N1 THEN A$="Opu!opx/":GOTO N950
OB 7802 IF IN$<>"TNBTI!CPVMEFS" OR I(N14)
    OR R<>N10 OR I(N8)<>-N1 THEN 7804
XE 7803 A$="Zpv!tnbti!ju!up!qfccmft=":I(N
   14)=10:UL=N1:GOTO N950
NK 7804 IF IN$="TNBTI!CPVMEFS" AND R=N10
   AND  NOT I(N14) THEN A$="Zpv!dbo(u///b
   u!uif!npnfou":GOTO N950
IV 7902 IF IN$="UBML!QBSSPU" AND R=N5 THE
   N A$="If!tbzt-(Ejbm!45.65.38(":GOTO N9
   50
MN 10060 NN=25:NV=14:SI=0:L=286
TV 10130 INTRO$(1)="Somewhere out in the
   wilds, a chest   of old ANALOG Magazin
   es has been      hi"
PK 10131 INTRO$(79)="dden.  These collect
   or's items are  just what you need to
   round out your  coll"
TY 10132 INTRO$(157)="ection, something y
   ou've been      trying to do for years.
     If you can    find t"
AK 10133 INTRO$(235)="hem, you'll be the
   envy of the  entire neighborhood!
                          "
ER 10200 IT$="QBSMBOPQFNBUMFBTIPTIFTMFSBL
   LFSUJSCPVTNBDBWDIFDBCMPDPMECPBGVMMJU45
   .SVCEPPHSP"
TL 10210 VB$(N1)="HFUESPMPPUBMPQFEJBSBL!!
```

The Wizard *continued*

```
   !TNBEJHSFNHP!MJHGJM"
TH 12001 DATA QBSSPU,-5
CV 12002 DATA MBOUFSO,0
OV 12003 DATA PQFO!MPDL,0
XT 12004 DATA NBUDIFT,5
HI 12005 DATA MFBWFT,-6
BW 12006 DATA TIPWFM,0
ZX 12007 DATA TIFE,-7
VT 12008 DATA TMFEHFIBNNFS,12
ZH 12009 DATA SBLF,12
XI 12010 DATA LFSPTFOF,12
EQ 12011 DATA UJSF!JSPO,5
BI 12012 DATA CPVMEFS,-41
YO 12013 DATA TNBMM!SPDLT,0
OF 12014 DATA DBWF,0
UR 12015 DATA DIFTU,0
CM 12016 DATA DBCJOFU,-5
TE 12017 DATA MPDL,0
FQ 12018 DATA PME!CPBSE,0
XG 12019 DATA CPBSEFE!EPPS,-7
UB 12020 DATA GVMM!MBOUFSO,0
DJ 12021 DATA MJU!MBOUFSO,0
RH 12022 DATA 45.65.38,0
LC 12023 DATA SVCCMF,-10
ZO 12024 DATA EPPS,0
BE 12025 DATA HSPVOE,0
FH 12200 DATA CLAYTON WALNUM,magazine hun
   t
```

The history of video gaming

Part 1: In the beginning . . .

by Arnie Katz and Joyce Worley

As with many other great inventions, from baseball to television, the origination of video games is shrouded in mystery and controversy. Several people made enormous contributions to the development of interactive electronic entertainment, and each could be viewed—with some justice—as the "Father of Video Gaming."

Some trace the idea back to Steve Russell. While a graduate student at M.I.T. in 1962, he created the first computer game, Spacewar. A circle of talented computer scientists seized upon this simple space combat program as a vehicle for increasing their programming knowledge and expertise. Their successive revisions ultimately produced the ancestor of all those flying and shooting epics in which a single defending craft challenges the rest of the universe.

One of the minor members of the M.I.T. group was Nolan Bushnell. More a visionary than a researcher, he perceived the potential market for such games and tried to develop a coin-op electronic game machine for arcades.

After failing with a commercial version of Spacewar called Computer Space, Bushnell tried to interest Bally in a host of different designs, but the Chicago-based company rejected all of them as too complex. Bally execs felt that computerized games should be as immediately understandable as the pinball machines which had always dominated the amusement centers.

At this point in the story, the third and final claimant to the title made his presence felt. The Magnavox division of North American Philips had hired Sanders Associates to find another way to utilize the 62 million television sets then in American homes.

Ralph Baer found the answer: games. Working with Bill Harrison and Bill Busch, Baer developed the prototype of a console which presented video games in color with FM sound on a TV set, as early as 1967.

In 1971, the Magnavox Odyssey, the first home video game unit, burst forth to a surprisingly lukewarm public reception. Three problems turned the Odyssey into a dud:

(1) The manufacturer never adequately explained the concept to consumers. Many passed up the Odyssey under the mistaken belief that it only worked with the same company's brand of TV sets.

(2) The Odyssey received very little promotional and ad support.

(3) Magnavox was late switching to solid-state technology, which made the hardware look clumsy and work in a not-always-reliable way.

The Odyssey was unbelievably crude by today's standards. It used interchangeable game programs, an idea many years ahead of its time, but each contest required the player to slap a playfield overlay on the screen. The Odyssey's memory was so small it couldn't draw both the playfield boundaries and the moving objects. Each of the controllers was the size of a clock radio.

Nolan Bushnell put aside his personal idea of what a video game should be and began to seek a concept simple enough to appeal to coin-op fanciers. He had reportedly seen the laboratory video game experiments conducted by the Sanders Associates, and decided to pin his hopes on a contest in which two players control "paddles" and bat a "ball" back and forth, as in tennis.

With much advice from sympathetic bystanders, Bushnell assembled a prototype of Pong and got a friendly tavern-keeper in Sunnyvale, California to put it in his establishment. Skeptics had warned that delicate electronics would not stand up to the punishment a coin-op generally suffers, and Bushnell was understandably dismayed when the bar owner called the very next day to report that the Pong machine had malfunctioned.

Bushnell rushed to the bar and found that there was, indeed, a problem. It was one he could live with. It turned out that play was so heavy that the coin receptacle had overflowed and had temporarily jammed the machine.

What did Nolan Bushnell do after he counted the money?

Did Magnavox turn the Odyssey into landfill?

Learn the answers to these and other questions in "The history of video games, Part 2," which will describe the dizzying rollercoaster that was the First Golden Age of Video Gaming.

Joust

ATARI CORP.
1196 Borregas Ave.
Sunnyvale, CA 94086
(408) 745-2000
Atari 7800
$20.00

by Bill Kunkel

This is the best home version of **Joust** ever produced. The William's coin-op classic comes to the home screen intact, complete with its great graphics, playability and high excitement level. Most players will want to relive the role of the high-flying hero again and again in this outstanding action game.

Joust's unique premise casts the player as a "buzzard rider." Members of this warring race seemingly spend all their time fighting with one another from the backs of large birds which resemble winged ostriches more than buzzards.

Once astride a faithful buzzard, the gamer faces a series of jousts with rival riders. To win these confrontations, the warrior must fly directly at each foe and strike him from a slightly higher position on the screen. A successful joust eliminates the opposing rider and causes an egg to fall from the buzzard. The player must grab this egg before it hatches in order to end the confrontation. Otherwise, the buzzard picks up the newborn rider and renews the battle.

The gamer views the world of **Joust** from a side perspective. The playscreen shows a series of rocky ledges which are perches for the avian steeds. As the rounds advance, some of these perches are eliminated, forcing the player to remain airborne for extended periods, while the number of enemy riders grows.

The black sky, the backdrop for all the action, highlights the bright colors of the buzzards. This helps the player follow the multitude of on-screen characters that wing in from dizzying angles and directions.

The game includes periodic bonus racks, or "egg rounds," in which the perches are stocked with unprotected buzzard rider eggs. The player must collect these eggs before they mature, for, once the hatchlings pop out of their shells, buzzards gather them up and transform them into enemy jousters.

The joystick action button makes the player's buzzard flap its wings and rise into the air. The harder the wings flap, the higher the big bird flies. Moving the joystick causes the buzzard to fly in the corresponding direction.

Eventually, additional menaces show up, such as the saw-toothed Pterrys. Any contact with a Pterry is fatal, but they're tough to avoid when they snake their way down the screen.

This isn't the only threat. At the game's beginning, the bottom perch extends across the screen. This disappears after a few rounds, to be replaced by flaming pits filled with monsters whose massive claws reach up to snatch unwary riders.

The one- or two-player game pits contestants against the machine, and it plays pretty well. There are four skill levels, so players can adjust the game to suit their ability.

Joust is a superior game, with lots of action and movement. Its animated characters, which travel in different patterns at several speeds, provide an ultimate challenge. This makes **Joust** a must for every 7800 game library.

Asteroids

ATARI CORP.
1196 Borregas Ave.
Sunnyvale, CA 94086
(408) 745-2000
Atari 7800
$20.00

by Bill Kunkel

The 7800 version of **Asteroids** is a revelation. **Asteroids** was the first mega-hit produced by Atari's own coin-op division, and it later proved very successful on many home video game and computer systems (2600, 5200, 400/800 computer). However, none of the previous incarnations even approached the impressive visual standards of this outstanding cartridge.

In **Asteroids**, the player moves an armed spaceship through a deadly field of meteoric debris. The action starts with the player's ship in the center of the screen. Space rocks hurtle across the screen in all directions. The player can either shoot or dodge these deadly missiles, but the rocket jockey must clear the whole screen before the game advances to the next rack.

A hit on an asteroid splits it into two smaller rocks. When the player shoots one of these fragments, it generates a pair of more diminutive shards, which then disappear when hit.

Each successive asteroid wave is deadlier than its predecessor. Additional challenging opponents are periodically introduced into play. These extra menaces include a pair of enemy spacecraft. The "dumb" ship is pretty much a sitting duck, but its "smart" cousin changes direction and relentlessly stalks the player's craft.

Asteroids introduced the "thrust and fire" play mechanic, which has subsequently become an arcade standard. In these contests, moving the joystick left or right rotates the player's ship. Moving the stick up engages the ship's thrusters and propels it in the direction toward which it's currently pointing. The action button fires the cannon. When trapped in an indefensible position, the player can send the ship into "hyperspace" by moving the joystick down. This risky strategy randomly transports it to a new position on the playfield.

Like most other Atari cartridges for the 7800, **Asteroids** can be enjoyed by one or two players, each competing against the system. Four difficulty levels promote replayability.

The eye-popping visuals really make this version of **Asteroids** something special. The original arcade machine used "vector graphics" technology, which showed on-screen objects in vivid, monochrome lines, but couldn't "fill" large areas with color. Vector graphics gave

Video gamers timeline: The early years

1962 — Steve Russell designed Spacewar, the first computer video game.

1966 — Ralph Baer begins development of the first unit for playing games through a TV set.

1968 — Two commercial versions of Spacewar, Computer Space and Galaxy Game fail to make a dent in the coin-op arcade market.

1972 — Magnavox introduces Ralph Baer's first video game, the Odyssey.

1972 — Nolan Bushnell tests the first coin-op, Pong.

everything a crystalline look, as if the ships and asteroids were made of glass.

This 7800 incarnation, however, employs state-of-the-art rasterscan technology to produce brightly colored, almost glittering graphics. The designers have used tones of the same color to shade the space rocks. This gives them a solidity the coin-up meteorites lacked, and it allows the player to actually see them spin as they zoom through space. The fabulous explosions glow against the stark, black background.

If you're one of the millions who always enjoyed **Asteroids**, or part of the new generation of fans who know it only as a legend, this 7800 program will knock you out.

Kung Fu

NINTENDO
P.O. Box 957
Redmond, WA 98052
(206) 882-2040
Nintendo Entertainment System
$24.95

by Bill Kunkel

Kung Fu, on the NES, is a first-rate adaptation of the Irem coin-op hit (published by Data East in the U.S.) The player controls martial arts hero Thomas, as he searches a five-story building for his captured girlfriend Sylvia. ("Thomas" and "Sylvia"? What kung fu devotee came up with those names?) The way, of course, is littered with stick fighters, boomerang throwers, giants, black magicians, knife throwers, snakes, dragons, poison moths and other unsavory types intent on standing between our Romeo and Juliet.

Thomas, for his part, can deliver several different martial arts moves. Hitting the A button on the controller throws a punch, while the B button instigates a kick. The user manipulates the directional rosette to let the hero move horizontally, crouch and jump. Since the NES controllers are not designed for diagonal movement, the player first moves Thomas to the left or right, *and then* presses the top of the control pad to leap in the desired direction.

Kung Fu uses a side view display and devotes the top of the screen to score-related data, including the time remaining and a series of markers to indicate the player's current location. It can be played in one- or two-player versions, at two different difficulty settings.

The visuals are simple and highlight the characters at the expense of elaborate backgrounds. The enemies are rendered with individuality and style. Each floor features a specific kind of fighter, and all are instantly recognizable. The stick fighters, for example, are mustachioed and clad in white *gis*; the knife throwers wear headbands; and the supremely powerful Mr. X, the gang leader on the fifth floor, is garbed in a sleeveless dungaree jacket and wristband.

Dragons arrive on the scene in balls which drop from the ceiling. Snakes show up in little baskets.

Successfully striking any of the bad guys earns points, but the hero's mode of attack often determines how *many*. Taking out one of the drone-like "Grippers" with a kick is worth 100 points, while a victory via punch or jump kick is good for twice that.

Once the player explores all five floors and rescues Sylvia, the game is still not done. Escape requires the gamer to reverse the original route, but at a heightened difficulty level.

Though not quite a classic, this NES version is a faultless reproduction of the arcade game. The arcade machine enjoyed tremendous popularity in the wake of the martial arts mania which swept the coin-op scene in 1984, and the NES version shoud win new converts today.

Hogan's Alley

NINTENDO
P.O. Box 957
Redmond, WA 98052
(206) 882-2040
Nintendo Entertainment System
(Zapper Light Gun required)
$29.95

by Bill Kunkel

Hogan's Alley brings Nintendo's own coin-op favorite to the NES. It also gives owners of the Zapper Light Gun something exciting to do with that weapon.

The game is based on the classic police test, in which the candidate moves down "Hogan's Alley," a simulated crime-infested block where potential disaster lurks in every window and doorway. Possible friends and foes pop up unexpectedly, testing the candidate's reaction time and discretion—to the max.

Hogan's Alley features three settings: Alley A, Alley B and Trick Shot. In the first, the player concentrates on three constantly rotating panels. These panels are "open" for target shooting for a very brief time. During that period, the player must determine whether the face which appears in each of these panels is a bad guy, a fellow cop, or an innocent bystander. If the face belongs to a crook, the player has a split second to fire before proceeding to the next panel to repeat the process.

Of course, everything happens in a heartbeat, with penalties incurred whenever the player either misses a bad guy or shoots a good guy. The panels move faster as the game progresses, so quick thinking is almost as important as straight shooting.

Any player who masters Alley A can confront Alley B, with its five dual-level panels. Master marksmen can then advance to the Trick Shooting mode. The machine "throws" cans onto the screen from the right edge. The player must shoot at these cans, keeping them wobbling through the air, and guide them over to the platforms on the left side of the screen.

The Zapper is a beautiful piece of weaponry. It looks lethal, fits even small hands comfortably, and has excellent accuracy. Adjustments of the TV's brightness and contrast are properly set. The ideal target range is six feet, but the actual distance to sit from the screen varies according to the size of the set's picture tube.

The graphics feature highly detailed drawings of the gangsters, cops and bystanders. Each face is unique. An especially clever effect signals each hit: hits make the target "card" spin around.

Hogan's Alley is a frenetic target game which should delight all Zapper-partisans. And the poor Duck Hunt targets will appreciate the rest.

Fantasy Zone

SEGA
2149 Paragon Drive
San Jose, CA 95131
(408) 435-0201
Sega Master System Mega Cartridge
$30.00

by Arnie Katz

When capitalistic creatures from Menon Planet start stealing the currency of other worlds to build a gigantic fortress, it's up to Opa-Opa to safeguard the civilized universe in **Fantasy Zone**. This scrolling shoot-out sends the player on a mission to destroy enemies on eight varied worlds.

The game gives the player three Opa-Opas with which to accomplish the task.

The direction disk steers Opa-Opa around the beautifully designed playfields, and each button fires a different weapon. Destroying creatures earns points, but the only way to move from one round to the next is to destroy all the bases on each planet. The gamer then battles a huge creature and, if successful, flies to the next planet—and new peril.

When Opa-Opa shoots a target, coins drop toward the ground. There are three sizes, with the largest worth the most. If the hero catches a coin before it lands, the amount is added to an account. If Opa-Opa touches a special symbol which periodically appears on the screen, it opens access to the Parts Shop.

This is clearly the most innovative aspect of **Fantasy Zone**. In the Parts Shop, money from accumulated coins can buy improved weapons and devices which increase movement speed. Opa-Opa needs these special aids to survive some of the intense attacks. Much of the strategy in the later rounds comes from deciding which parts will help the hero most on each planet.

The invaders are quite colorful and attractive, though they could have been a little less abstract and geometrical in appearance. On the plus side, it's easy to tell them apart at a glance, and each has a distinctive method of attack.

Despite the emphasis on strategy, **Fantasy Zone** remains essentially a tough test of hand-eye coordination. The enemies appear often and in great numbers. The action is not blindingly fast, but there are so many dangers that gamers will welcome the short rest periods between rounds.

Fantasy Zone is an exciting blend of strategy and blazing combat. Cap its hot action with lush animated art and lilting theme music, and the result is a treat for video gamers.

Hover Force

INTV CORP.
3541-B Lomita Blvd.
Torrance, CA 90505
(213) 539-0100
Intellivision Master Game Component
$20.00

by Arnie Katz

When terrorists overrun the peaceful island city of Seeburg, it's up to the **Hover Force** to prevent them from demolishing it. A dispatch from Col. N.K. Newcombe, leader of this elite corps, sends the player into the skies over Seeburg in a top secret JAF-3000 helicopter. This fast and maneuverable craft can fire a laser cannon at enemy choppers and pour water on fires which the terrorists have set in an effort to destroy major buildings.

A special briefing screen, which summarizes the control system for the helicopter, supplements the detailed manual which accompanies this air combat epic. The double dose of guidance makes it surprisingly easy to pick up the rudiments of what is a fairly complex game.

The most notable aspect of **Hover Force** is how well the control system communicates the feel of piloting a helicopter. Players accustomed to instant reversals of direction are in for a shock. If the JAF-3000 is heading east, for instance, it takes more than pressing the 9:00 position on the direction disk to send it flying west. Timely use of the air brake kills momentum and allows relatively sharp turns. The greatest challenge for the gamer is learning how to steer accurately enough to put the sighting cursor on top of a moving target long enough to blast it.

The playfield background, an aerial view of Seeburg, serves as an attractive backdrop to the fighting in the sky overhead. The helicopters are not drawn in comparable detail, but the spinning blades look quite good during intense air battles.

Some strategic factors affect the flying and shooting. The radar screen, reached by hitting *0* on the controller keypad, shows the percentage of damage sustained by the 'copter, the level of fuel remaining, the number of enemy aircraft destroyed, the dollar cost of terrorist destruction, and the amount of ammo on board for the laser and water cannons. A visual display indicates the condition of the engine and navigation systems. A radar screen on the left side of this display presents a broad overview of the entire operation.

Landing on either of the helipads at home base enables the player to repair, rearm, and refuel the JAF-3000. The player gets three choppers during the course of the game, one at a time. Unfortunately, the enemy realizes the value of the helipads, and there's often a swarm of hostile helicopters ready to pounce on the **Hover Force** unit as it returns to its base.

Hover Force is a one-player contest which can be enjoyed at any of three skill levels. More intelligent and aggressive opposition makes the highest setting, "Ace," a true test of video gaming prowess. Fans of air combat should not delay their visit to Intellivision's unfriendly skies.

Midnight Magic

Broderbund/Atari
ATARI CORP.
1196 Borregas Ave.
Sunnyvale, CA 94086
(408) 745-2000
Atari 2600/7800
$10.00

by Bill Kunkel

Once upon a time, in the days before video games filled arcades, the pinball machine reigned supreme. Even Space Invaders and Pac-Man couldn't *totally* displace the flipper machines. The combination of random, almost haphazard action, and the degree of skill required to master manipulation of the flippers has fascinated gamesters for decades.

Pinball meets video game, and the marriage improves both species in **Midnight Magic**, a dynamite flipper simulation based on the Broderbund computer software bestseller by David Snyder. **Midnight Magic** offers 2600 owners a simple but colorful table that includes two sets of flippers, five drop targets, two rollovers, three bumpers, a spinner, several targets and six bottom lanes.

At any given time, one of the five drop targets is designated with an arrow. Hitting this target (in the A difficulty setting) activates a pair of "kickers" which appear in the "gutter" lanes, saving what would ordinarily be lost balls. (In the B, or easy setting, the kickers and center post are always on.)

Midnight Magic is playable by one or two players at either of these two difficulty settings.

Flippers are activated, via joystick, either individually or collectively. The ball

We Want Letters

The fate of *Video Game Digest*, **ANALOG Computing**'s magazine-within-a-magazine, is in your hands. If you want a monthly video game magazine, with reviews, news and insights into the fast-changing world of cartridge games, write and tell us so.

We'd like to hear from readers on every aspect of video gaming, and want to know what you'd like to see in *Video Game Digest*.

is launched by pulling back on the joystick until the trigger is cocked at the desired impact level. The trigger is released by pressing the action button.

The audio and visual components are of prime importance in any video pinball simulation. The sights and sounds of pinball are second nature to most game players, who can instantly detect it when a ball, bumper or flipper behaves in a manner inconsistent with the real thing. On this score, **Midnight Magic** rates high. The balls shoot, roll and rebound with unfailing verisimilitude.

The colors are beautiful, effective without being gaudy. The rendering of the table surface is flawlessly realistic. The only problem is the lack of a "backboard" and scoreboard. Instead, the score appears on the table itself, just above the exit lanes. This causes quite a bit of confusion in the heat of play, as the score occasionally obscures the action.

The sounds of pinball are also well portrayed in **Midnight Magic**. Every rebound and rollover produces an appropriate audio effect, greatly enhancing the overall experience.

Midnight Magic is a superior video pinball simulation that is all the more impressive for its having been produced for the 2600.

VIDEO GAME NEWS UPDATE

A couple of popular computer games are scheduled for video game translations from Atari. Robert Neve's **Crack'ed**, a struggle to protect eggs from hungry creatures, will soon hatch for the 2600 and 7800. Spectrum Holobyte's classic **GATO** puts Atari XE system owners in control of a WWII sub. This complex computer game monitors speed, depth, heading and radar tracking, as the gamer tries to overcome the enemies' strategies. Cosmi's **Super Huey** turns the 7800 into a helicopter flight simulator, then charges the gamer with the rescue of stranded personnel, defense of military installations, and surveillance of unexplored territory.

Sega's racing simulation, **Out Run**, made tracks as a top play-for-pay game. Now it's roaring onto the video game screen, via a 2-meg cartridge. This 2096K game is packed with an assortment of courses and road scenes, lane changes, hills, and curves that bring **Out Run** home with almost no changes in the appearance and action that made it a superhit.

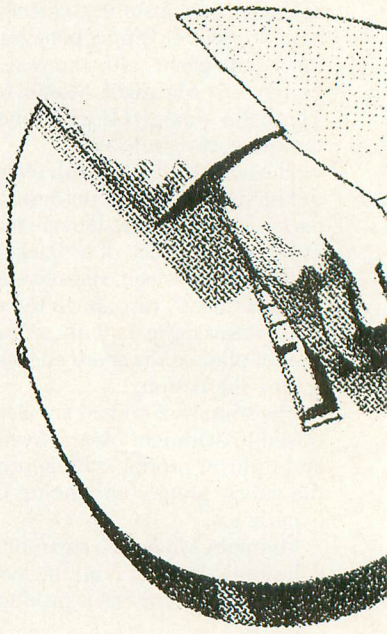
Here's a haunting echo of the first age of video games. Activision, the world's first second-party software house, is going to produce video games for the Nintendo Entertainment System and the Atari 2600. The new games will reach the stores this autumn. There are also quite a few titles in inventory which were first introduced in the early part of the 80s.

Several top designers of award-winning software joined forces to found Absolute Entertainment, to publish video games for the Atari 2600. Garry "GameMaker" Kitchen, John "Computer Fireworks Celebration Kit" Van Ryzin, Dan "Greeting Card Maker" Kitchen and Alex "Great American Cross Country Road Race" De Meo combined talents on a pair of cartridges to be distributed by Activision. **Title Match Pro Wrestling** gives gamers a choice of four wrestlers, whom they can match against a computer opponent or battle in head-to-head mode. **Skateboardin'** is a wild and crazy ride through city streets and sidewalks, as the racer tries to get to school on time.

Data East USA has seven games for the Nintendo Entertainment System this year. The company promises the same thrills 'n chills from these translationsthat made them hits in the arcades. Among the titles announced are a trio of combat games: **Ring King, Karate Champ** and **Tag Team Wrestling**.

CAPCOM U.S.A. has introduced five game packs for the Nintendo Entertainment System. The Sunnyvale, California company, best known for its play-for-pay designs, has started marketing games for home computer and video gamers. The titles scheduled for play on the NES are: **1942, Commando, Trojan, Ghosts 'N Goblins** and **Section Z**.

Guitar Wizard



BAUDVILLE
1001 Medical Park Drive, S.E.
Grand Rapids, MI 49506
(616) 957-3036
48K Disk $24.95

by Clayton Walnum

Baudville's latest "flippy" (the Atari version is found on the flip side of a Commodore disk) is a product aimed specifically at those of us who like to abuse our fingers by dragging them across six steel cables stretched tightly over a wooden box. In other words, if you aren't interested in playing the guitar, read no further. For those of you who have lost all feeling in your fingertips, due to thick callouses developed while coaxing brilliant patterns of sound from those aforementioned steel cables, **Guitar Wizard** may be a product of some interest.

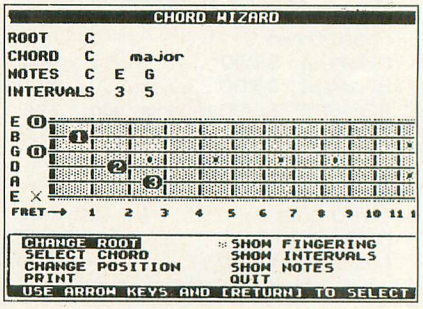
Guitar Wizard is actually four programs in one: Chord Wizard, Scale Wizard, Fretboard Wizard and Improvisation Wizard.

Chord Wizard allows you to graphically find chords in various positions on the neck. The screen displays the first 12 frets and indicates the chord by placing little ovals on the strings where your fingers should go (where they should go on the guitar, dummy, not on the screen). The graphics are well done. Many of you will have seen similar "charts" in guitar books before and will feel right at home with the technique used.

The fingering markers can be set up to show several types of information: the suggested fingering (using numbers from 1 to 4, the index finger through the pinky, respectively); the notes of the chord; or the intervals within the chord. You may, at any time, change the chord's root, the chord type (i.e., minor, major, augmented, etc.) or the chord's position, after which the display will be updated. If you move a chord's position past the 12th fret, the guitar graphic will move up to accomodate the new position; you're not stuck with viewing only the first 12 frets.

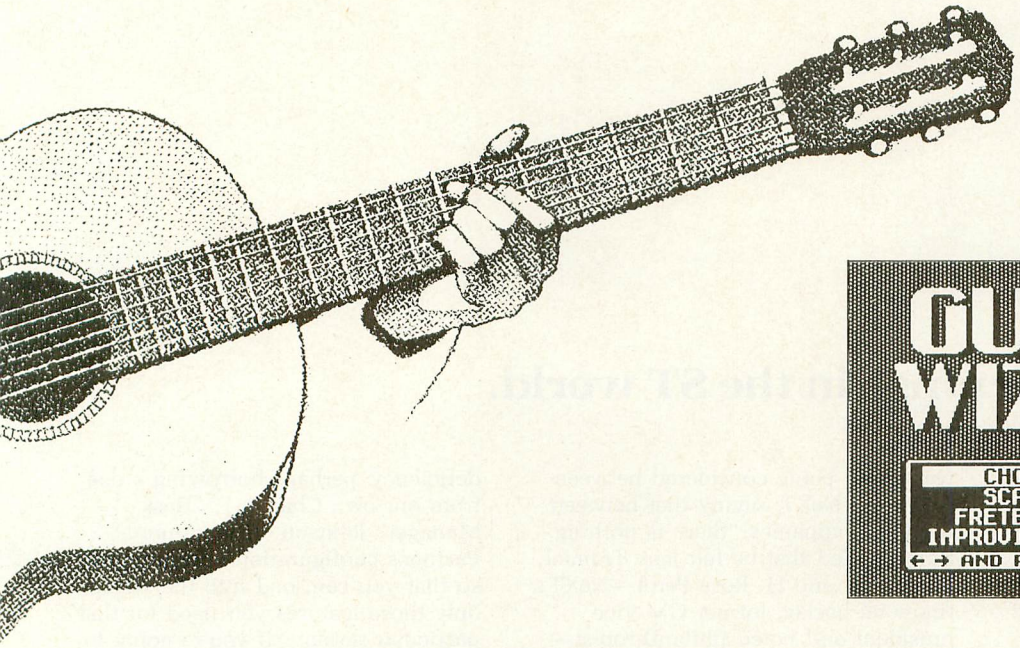
All selections are made by moving a cursor over a menu, with either the arrow keys or the joystick. Pressing RETURN or the fire button finalizes your decision. The menu setup works so smoothly and intuitively, you'll hardly need to read the manual.

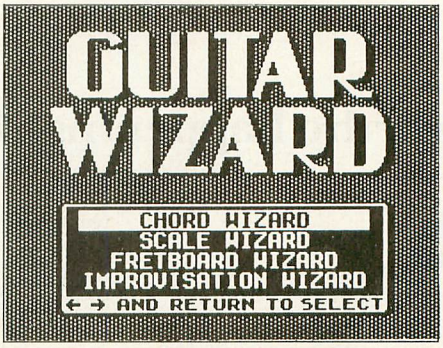
The second program section, Scale Wizard, works much like Chord Wizard, except, obviously, it displays scales—every scale you could imagine. In fact, this program includes not only the mundane standbys such as major and minor, but some exotic entries you'll have a hard time pronouncing, let alone playing: phrygian, mixolydian, aeolin, lydian and locrian. If this program is nothing else, it's complete; over a dozen and a half scales can be studied.



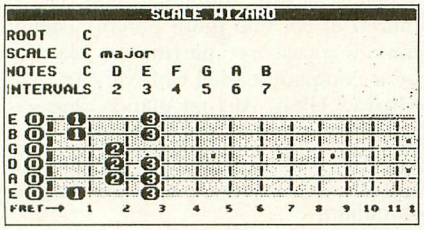
Chord Wizard

Of course, you may change the root and scale type if the whim strikes. In addition, many of the menu selections offered in the chord program are available here. In fact, with the exception of the "Select Scale" entry, this section's menu is a duplicate of that used with Chord Wizard.





Fretboard Wizard, the third program available from the main menu, displays scales and chords over the entire neck (well, at least the first 12 frets; they start repeating up there anyway), rather than one position at a time. With this selection, you can also change the instrument's tuning to your liking, and save the tunings to disk for later retrieval.



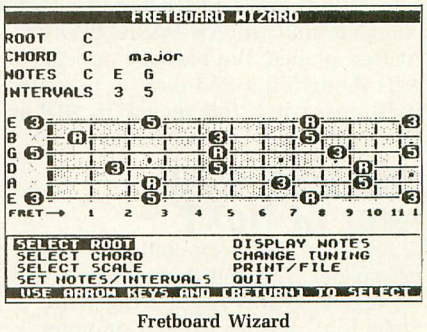
Scale Wizard

The fretboard is marked with those familiar little ovals used in Chord Wizard and Scale Wizard, but since fingering doesn't apply here, you can have only the notes or intervals displayed within the markers. Not wanting to be unlike its cousin programs, Fretboard Wizard lets you choose any root, plus chord and scale type. You can also manually add notes or intervals to the chord or scale displayed.

Finally, the fourth program, Improvisation Wizard, allows you to set a chord and chord type, after which a list of suggested scales for improvisation will be displayed. You have your choice of eight scales. If you're not familiar with some of them, Scale Wizard will be delighted to fill the gaps in your guitar education.

Guitar Wizard is a nice piece of work, and when used with the optional book (*Guitar Wizard Study Guide*, $12.95) and the built-in provision to dump displays to your printer (Epson, Star, Gemini and Okidata only), will provide the guitar student with many valuable lessons. The manual is concise and complete and includes a primer on fretboard harmony for those who are just starting out. The user interface is nicely thought out, the menus quick-acting and easy to use.

My only reservation with this program (and others of its type) is that, since it presents information, most of which is readily—and possibly even more conveniently—found in books, it may be more effort to use than it's worth. (Booting a disk, though not a backbreaking chore, can become a major inconvenience when repeated with any frequency.) There are some definite advantages to choosing **Guitar Wizard**, however, not the least of which is the ability to experiment with different tunings and immediately see their results on the scales and chords.



Fretboard Wizard

One thing is for sure: With a reasonable price like $24.95, the program won't cost you more than an equivalent set of books. If you don't mind booting a disk to study your chords and scales, you'll get your money's worth.

Newsworthy happenings in the ST world.

GFA This, GFA That...

Without a doubt, MichTron's GFA BASIC is becoming for ST programmers what Turbo Pascal is for the PC world. (In Germany, the *f* in GFA is in lowercase, standing for *fur*—*for*, in English.) Every few months, it seems, Mark Brutell of MichTron Customer Service introduces another GFA-supporting product. Now it has introduced its first "toolkit" manufactured outside of Germany: GFA BASIC Companion, by John B. Holder of Marathon Computer Press.

The package includes: a simple-form resource construction set, for the creation of *.RSC files which define custom-design input boxes, and help screens; a GEM routine library so that BASIC programs may utilize AES/VDI subroutines and GEMDOS calls; and a complete multiscreen BASIC tutorial. At first glance, the package appears well thought out and easy to execute. We'll give you a full review in **ST-Log** soon. //

The laST we'll hear of NeXT

A report in a weekly computer newspaper rumored that Atari management was negotiating with Steve Jobs, former Apple Computer Chairman and current CEO of NeXT, Inc., to reach an agreement making Atari a licensed manufacturer of NeXT's new 68030-based **RISC workstations**.

Shiraz Shivji, speaking on behalf of Atari, categorically denied any deals were even being considered between Atari and NeXT, saying that between the two companies "there is nothing." Shivji added that he felt Jack Tramiel, Steve Jobs and H. Ross Perot—NeXT's financial backer, former GM vice president and noted philanthropist—are "three men of different lines." //

Yet another do-everything accessory bonanza

Timeworks has released **Partner ST**, which includes the familiar calculator, alarm clock, address book, memo pad and TOS utility features. To save a lot of space, let's focus on Partner's original features.

There's a 60,000-word thesaurus, independent of any spelling checker; an expense account manager for keeping track of your ever-increasing credit card debt; a table of vital statistics like metric conversion equivalents, mileage between cities, and commonly used toll-free numbers; an expanded printer control panel for setting specific type styles; and a Breakout-type game window. The calculator is specially equipped for financial calculations, like interest and loan amortization, and can operate in standard or reverse polish notation, like an HP-15.

The primary deficiency in most do-everything accessory packages is the amount of memory they consume. After you've made your ST capable of doing everything, it seems you have no space left to do anything else. Timeworks may have conquered this deficiency, perhaps borrowing a cue from our own Charles F. "Desk Manager" Johnson. At boot-up, Partner's configuration screen pops up, so that you can load into the system only those features you need for that particular session. If you're going to be churning out C source code, for instance, you certainly don't need a thesaurus taking up space. As for the Breakout game, you'll proabably need that continually. The RAM normally occupied by those functions you don't want, is thus freed.

The list price is a sensible $69.95. //

Life in a memory residence.

The memory-resident space in my computer is becoming rather crowded these days, and is in dire need of a landlord.

Over the hill, though, I hear the march of yet one more potential tenant for my accessory apartments: MaxPak, by a company called Softwerx in Murray, Utah. At first glance, one will notice it contains much of the standard fare of utilities: RAMdisk, printer spooler, clock with alarm, calculator.

The differences are these: it lets you configure macro keys to perform mathematical calculations on the fly. You can print out any text file any time, in the manner it was formatted—boldface, italic, proportional—with varying page sizes so perforations will be skipped. The calculator can send a result as input to a program. A 1st-Word-like character

ST notes

by D.F. Scott

table is available, and a "screen saver" timed monitor disconnection switch is also included.

The *major* difference in this package is that these accessories are *non-GEM*, so they may be accessible via macro keys while running non-GEM programs such as Micro EMACS, Zoomracks and Neo-Chrome. Whether you'd need macro keys or a calculator from within Neo-Chrome is entirely your affair. Suggested retail price is $49.95. ///

If only it were produced in "Sensurround..."

Our game pick of the month is **Terrorpods**, which was plucked out of a 1950s sci-fi scenario, but designed by one of the 1980's leading interactive computer art groups, Psygnosis. Some of you will remember that I considered Deep Space, a previous Psygnosis product, to be slow. I'm very pleased to announce that Terrorpods is *not* slow.

The game is set on a lunar mining colony—on some other moon—which is under attack by a swarm of terrorpods. These are machines with heads like Spy vs. Spy characters and bodies that are a cross between Beetle Bailey and a burnt Terminator. Your job is to save the various installations from impending oblivion, and to rebuild those which have already lost an argument to a terrorpod.

These installations, however, are not noted for their gratitude; and although you need *their* fuel and minerals so you can rescue them later, you still have to barter, beg and plead with them. The feeling somewhat resembles running a corner jewelry stand during the Bombing of London. Your main vehicle—from which you have a first-person perspective of the impending carnage—burns fuel more like a freightliner than a Ford Escort.

The sound is well digitized; your torpedoes give a rousing shriek rather than a boring beep. Psygnosis's animative skills are improving with each new game. I have a feeling if Deep Space were produced today, it would be a *much* better game. ///

Thirty-second notes

In the program Music Studio, thirty-second notes are the shortest notes there are. So, for that matter, are these:

Electronic Arts has finally released **Marble Madness**. And Mindscape—not Atari as once expected—has released **Gauntlet**, and will be releasing **Paperboy**, for the ST. These three games are ports from the coin-op editions of the same name produced by Atari Games Co.—now the American subsidiary of Namco, the creator of Pac-Man, Galaga, and Pole Position.

Regent Software's graphic database **The Informer** is now available nationwide, listing for $99.95. In this base, data is tabulated in spreadsheet form and reported in a user-defined graphic format... Bantam Books, publisher of the slick, well typeset series of reference guides for Amigas and Apple II computers, is now releasing similarly stylish guides for the ST. Its first entry is ***Atari ST Application Programming*** by Lawrence J. Pollock and Eric J. T. Weber of Diatech Publications. The price is $24.95.

By the time you read this, Origin Systems should have released **Ultima V**, the latest in Lord British's series of adventures through the Dark Ages in search of inspiration, truth and cash. Ultima's new publisher—barring any legal complications—will be Broderbund... Hybrid Arts has publicly released **MIDI Maze**, the favorite game of Atari conventions and Neil Harris. Included in the final version for public consumption are a player-vs.-computer mode (no MIDI cables required here), and mazes which are editable with an ASCII text editor.

Those are the **notes** for this month, a virtual concerto of facts and ideas, interlaced with interludes of entertainment. I'll see you on Delphi. ///

M/L Editor

For use in machine language entry.

by Clayton Walnum

M/L Editor provides an easy method to enter our machine language listings. It won't allow you to skip lines or enter bad data. For convenience, you may enter listings in multiple sittings. When you're through typing a listing with **M/L Editor**, you'll have a complete, runnable object file on your disk.

There is one hitch: it's for disk users only. My apologies to those with cassette systems.

Listing 1 is **M/L Editor**'s BASIC listing. Type it in and, when it's free of typos, save a copy to disk, then run it.

On a first run, you'll be asked if you're starting a new listing or continuing from a previously saved point. Press *S* to start, or *C* to continue.

You'll then be asked for a filename. If you're starting a new listing, type in the filename you want to save the program under, then press RETURN. If there's already a file by that name on the disk, you'll be asked if you wish to delete it. Press *Y* to delete the file, or *N* to enter a new filename.

If you're continuing a file, type in the name you gave the file when you started it. If the program can't find the file, you'll get an error message and be prompted for another filename. Otherwise, **M/L Editor** will calculate where you left off, then go on to the data entry screen.

Each machine language program in **ANALOG Computing** is represented by a list of BASIC data statements. Every line contains 16 bytes, plus a checksum. Only the numbers following the word *DATA* need be considered.

M/L Editor will display, at the top of the screen, the number of the line you're currently working on. As you go through the line, you'll be prompted for each entry. Simply type the number and press RETURN. If you press RETURN without a number, the default is the last value entered.

This feature provides a quick way to type in lines with repetitions of the same number. As an added convenience, the editor will not respond to the letter keys (except *Q*, for "quit"). You must either enter a number or press RETURN.

When you finish a line, **M/L Editor** will compare the entries' checksum with the magazine's checksum. If they match, the screen will clear, and you may go on to the next line.

If the checksums *don't* match, you'll hear a buzzing sound. The screen will turn red, and the cursor will be placed back at the first byte of data. Compare the magazine listing byte by byte with your entries. If a number's correct, press RETURN.

If you find an error, make the correction. When all data's valid, the screen will return to grey, and you'll be allowed to begin the next line.

Make sure you leave your disk in the drive while typing. The data is saved continuously.

You may stop at any time (except when you have a red screen) by entering the letter *Q* for byte #1. The file will be closed, and the program will return you to BASIC. When you've completed a file, exit **M/L Editor** in the same way.

When you've finished typing a program, the file you've created will be ready to run. In most cases, it should be loaded from DOS via the L option. Some programs may have special loading instructions; be sure to check the program's article.

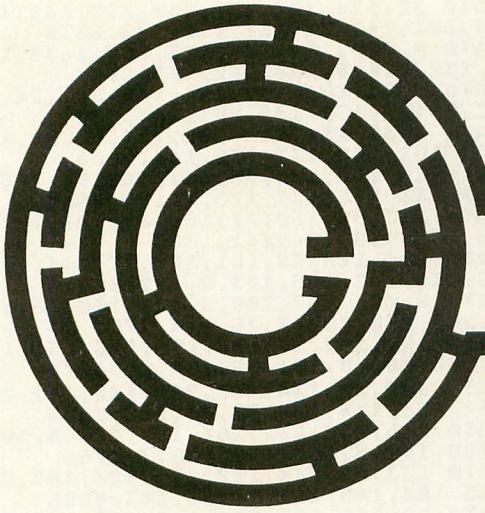
If you want the program to run automatically when you boot the disk, simply name the file AUTORUN.SYS (make sure you have DOS on the disk).

That's **M/L Editor**. Use it in good health.

The two-letter checksum code preceding the line numbers here is *not* a part of the BASIC program. For further information, see the "BASIC Editor II," in issue 47.

Listing 1.
BASIC listing.

```
AZ 10 DIM BF(16),M$(4),A$(1),B$(1),F$(15)
   ,F1$(15)
LF 11 DIM MOD$(4)
BN 20 LINE=1000:RETRN=155:BACK5P=126:CHK5
   UM=0:EDIT=0
GO 30 GO5UB 450:POSITION 10,6:? "Start or
    Continue? ";:GO5UB 500:? CHR$(A)
ZG 40 POSITION 10,8:? "FILENAME";:INPUT F
   $:POKE 752,1:? " "
FE 50 IF LEN(F$)<3 THEN POSITION 20,10:?
   "  ";:GOTO 40
NF 60 IF F$(1,2)<>"D:" THEN F1$="D:":F1$(
   3)=F$:GOTO 80
KL 70 F1$=F$
TN 80 IF CHR$(A)="5" THEN 120
FD 90 TRAP 430:OPEN #2,4,0,F1$:TRAP 110
HQ 100 FOR X=1 TO 16:GET #2,A:NEXT X:LINE
   =LINE+10:GOTO 100
HM 110 CLOSE #2:OPEN #2,9,0,F1$:GOTO 170
VT 120 TRAP 160:OPEN #2,4,0,F1$:GO5UB 440
   :POSITION 10,10:? "FILE ALREADY EXIST5
   !!":POKE 752,0
ZU 130 POSITION 10,12:? "ERA5E IT? ";:GO5
   UB 500:POKE 752,1:? CHR$(A)
VH 140 IF CHR$(A)="N" OR CHR$(A)="n" THEN
    CL05E #2:GOTO 30
QG 150 IF CHR$(A)<>"Y" AND CHR$(A)<>"y" T
   HEN 130
BH 160 CL05E #2:OPEN #2,8,0,F1$
IE 170 GO5UB 450:POSITION 10,1:? "NOW ON
   LINE: ";LINE:CHK5UM=0
GH 180 L1=3:FOR X=1 TO 16:POSITION 13*(X(
   10)+12*(X)9),X+2:POKE 752,0:? "BYTE #"
   ;X;": ";:GO5UB 310
KH 190 IF EDIT AND L=0 THEN BYTE=BF(X):GO
   TO 210
FY 200 BYTE=VAL(N$)
OZ 201 MOD$=N$
BU 210 POSITION 22,X+2:? BYTE;"  "
YZ 220 BF(X)=BYTE:CHK5UM=CHK5UM+BYTE*X:IF
    CHK5UM>9999 THEN CHK5UM=CHK5UM-10000
M5 230 NEXT X:CHK5UM=CHK5UM+LINE:IF CHK5U
   M>9999 THEN CHK5UM=CHK5UM-10000
IG 240 POSITION 12,X+2:POKE 752,0:? "CHEC
   K5UM: ";:L1=4:GO5UB 310
EW 250 IF EDIT AND L=0 THEN 270
QM 260 C=VAL(N$)
SY 270 POSITION 22,X+2:? C;"   "
IL 280 IF C=CHK5UM THEN 300
DI 290 GO5UB 440:EDIT=1:CHK5UM=0:GOTO 180
LW 300 FOR X=1 TO 16:PUT #2,BF(X):NEXT X:
   LINE=LINE+10:EDIT=0:GOTO 170
FV 310 L=0
KZ 320 GO5UB 500:IF (A=A5C("Q") OR A=A5C(
   "q")) AND X=1 AND  NOT EDIT THEN 420
PO 330 IF A<>RETRN AND A<>BACK5P AND (A<4
   8 OR A>57) THEN 320
DX 331 IF A=RETRN AND N$="" THEN N$=MOD$
TD 335 IF A=RETRN AND L=0 AND X>1 THEN 35
   0
JR 340 IF ((A=RETRN AND  NOT EDIT) OR A=B
   ACK5P) AND L=0 THEN 320
DW 350 IF A=RETRN THEN POKE 752,1:? " ":R
   ETURN
GG 360 IF A<>BACK5P THEN 400
5A 370 IF L>1 THEN N$=N$(1,L-1):GOTO 390
A5 380 N$=""
RE 390 ? CHR$(BACK5P);:L=L-1:GOTO 320
BB 400 L=L+1:IF L>L1 THEN A=RETRN:GOTO 35
   0
WK 410 N$(L)=CHR$(A):? CHR$(A);:GOTO 320
KN 420 GRAPHIC5 0:END
YT 430 GO5UB 440:POSITION 10,10:? "NO 5UC
   H FILE!";:FOR X=1 TO 1000:NEXT X:CLOSE
   #2:GOTO 30
FD 440 POKE 710,48:50UND 0,100,12,8:FOR X
   =1 TO 50:NEXT X:50UND 0,0,0,0:RETURN
MY 450 GRAPHIC5 23:POKE 16,112:POKE 53774
   ,112:POKE 559,0:POKE 710,4
XR 460 DL=PEEK(560)+256*PEEK(561)+4:POKE
   DL-1,70:POKE DL+2,6
HW 470 FOR X=3 TO 39 5TEP 2:POKE DL+X,2:N
   EXT X:FOR X=4 TO 40 STEP 2:POKE DL+X,0
   :NEXT X
ZW 480 POKE DL+41,65:POKE DL+42,PEEK(560)
   :POKE DL+43,PEEK(561):POKE 87,0
AC 490 POSITION 2,0:? "analog M1 editor":
   POKE 559,34:RETURN
WZ 500 OPEN #1,4,0,"K:":GET #1,A:CL05E #1
   :RETURN
```



Labyrinths

An aMAZE-ing 3-D adventure game

by Steven Lashower

Finally, some time to yourself—the first time in many days. You sit down at your computer for a nice intellectual game of Star Raiders. Just as you're ready to zap some Zylons, the doorbell rings.

"Somebody better have a very good reason for this," you mumble as you go to the door and open it.

On the other side of the door is an alien. "Hello, earthling," he says. "My name is Beef Strovanoff and I'm from the planet of Argonia."

The alien continues: "You have been selected to represent your local star system in the Labyrinths of Argonia. If you accomplish this impossible task, you will be rewarded with treasures beyond your wildest dreams. But if you fail, you will be disposed of. . .We love incentive."

The alien then raises what looks like a "Kill-O-Zap" brand atomic teleport gun. He fires it at you.

Playing *Labyrinths*

Labyrinths is a one-player maze game written in 100 percent machine language and will run on all 8-bit Atari computers. It features colorful first perspective graphics that really place you in the maze. Type in Listing 1 using M/L Editor (see page 80) to create your copy of the game. Load the game using option L from the Atari DOS menu.

When the game boots up, you're presented with the title page. Here, you can select between several options.

Pressing OPTION will change the number of "Argonian Patrollers" that are in the maze with you. You can choose from 0 to 6 Patrollers. The more Patrollers in the maze, the more difficult the game is. Pressing SELECT will determine whether the game is played in a "daytime" or "nighttime" atmosphere and pressing START will, of course, begin the game. At any point during the game, pressing ESC will return you to the title page.

You—the flashing blue dot—appear in the upper-left corner of the maze. The object, simple enough, is to escape from the maze. The exit is in the lower-right corner of the maze. In the center of the screen is a graphic window of what appears in the maze. To the right of this is a black area that will fill up with the maze as you explore new areas of it. If you're playing with Argonian Patrollers, they will also appear in this area as white dots.

At the bottom of the screen is a timer that displays how long you've been in the maze. There's also a compass to show which direction you're facing.

You move through the maze by using a joystick plugged into port 1. Pushing the stick forward will move you in the direction the compass is facing. Pulling back on the stick will move you backward (away from the direction the compass is facing). Pushing the stick to the left or right will turn you in the appropriate direction.

If you happen to come across a Patroller in the maze, it will appear as a pulsing square in the graphic window. The size of this square depends on how close the Patroller is to you. Coming in contact with a Patroller means immediate destruction for you (these things really are mean).

So, what are you waiting for? Get lost in **Labyrinths**.

*Steven Lashower, a high school senior, has owned an Atari for seven years. He's been programming in assembly language for almost two years. **Labyrinths** is his first major machine language project. He also has a part-time job at Disneyland.*

Listing 1
"M/L Editor" Data

```
1000 DATA 255,255,0,120,251,120,169,0,
141,47,2,133,138,169,255,133,8873
1010 DATA 154,32,42,128,169,92,133,132
,169,135,133,133,32,147,132,169,8338
1020 DATA 197,141,47,142,169,65,141,15
9,126,169,129,141,160,126,169,0,8016
1030 DATA 162,0,149,148,232,224,5,208,
249,133,147,169,60,141,28,2,6856
1040 DATA 160,186,162,120,169,7,32,92,
228,169,1,133,140,169,2,133,6078
1050 DATA 141,169,3,133,144,169,50,133
,142,32,206,125,76,160,122,72,6792
1060 DATA 138,72,166,147,224,0,208,25,
232,134,147,162,144,142,23,208,9726
1070 DATA 232,232,141,10,212,224,160,2
08,244,169,10,141,23,208,76,182,368
1080 DATA 120,169,10,141,24,208,169,4,
141,23,208,166,147,165,155,201,256
1090 DATA 0,208,6,189,78,124,76,150,12
0,189,96,124,141,20,208,141,8149
1100 DATA 21,208,232,134,147,224,17,20
8,19,169,0,133,147,169,248,141,341
1110 DATA 22,208,169,10,141,23,208,169
,0,141,24,208,104,170,104,64,6232
1120 DATA 238,192,2,238,193,2,173,193,
2,201,159,208,5,169,144,141,9552
1130 DATA 193,2,165,187,201,0,240,2,19
8,187,165,138,201,255,208,3,1745
1140 DATA 76,98,228,173,28,2,201,0,208
,27,169,60,141,28,2,230,5003
1150 DATA 148,162,0,181,148,221,36,134
,208,6,169,0,149,148,246,149,9610
1160 DATA 232,224,252,120,247,121,5,20
8,238,162,0,181,148,24,105,16,6761
1170 DATA 188,31,134,153,147,129,232,2
24,5,208,240,165,154,201,255,240,6585
1180 DATA 8,198,154,24,101,158,141,1,2
10,162,0,181,223,201,69,240,860
1190 DATA 17,201,160,208,8,169,0,142,0
,208,76,98,228,232,224,5,8855
1200 DATA 208,233,189,41,134,141,0,208
,189,47,134,141,8,208,189,62,7965
1210 DATA 134,133,157,160,0,169,0,153,
0,116,200,208,250,189,57,134,9710
1220 DATA 168,189,52,134,153,0,116,200
,196,157,208,245,76,98,228,169,2681
1230 DATA 92,133,132,169,135,133,133,1
36,192,0,240,8,165,142,32,124,7213
1240 DATA 122,76,109,121,138,32,124,12
2,162,0,165,144,201,1,208,46,6535
1250 DATA 165,142,32,124,122,160,0,177
,132,133,146,165,142,24,105,1,5154
1260 DATA 32,134,122,160,0,177,132,149
,203,200,177,132,149,193,200,177,3573
1270 DATA 132,149,213,232,165,142,32,1
34,122,224,8,208,230,96,165,144,1275
1280 DATA 201,3,208,46,160,0,165,142,3
2,134,122,177,132,133,146,165,8888
1290 DATA 142,56,233,1,32,124,122,160,
0,177,132,149,213,200,177,132,631
1300 DATA 149,193,200,177,132,149,203,
232,165,142,32,124,122,224,8,208,889
1310 DATA 230,96,165,144,201,4,208,64,
160,0,177,132,153,193,0,200,8662
1320 DATA 192,8,248,121,243,122,208,24
6,160,0,169,1,32,134,122,177,8392
1330 DATA 132,133,146,169,1,32,124,122
,165,142,32,134,122,177,132,153,8240
1340 DATA 203,0,200,192,8,208,246,165,
142,24,101,142,32,124,122,160,8116
1350 DATA 0,177,132,153,213,0,200,192,
8,208,246,96,160,0,162,0,7233
1360 DATA 177,132,149,193,169,1,32,134
,122,232,224,8,208,242,169,9,9916
1370 DATA 32,124,122,177,132,133,146,1
69,1,24,101,142,32,134,122,162,6334
1380 DATA 0,177,132,149,213,169,1,32,1
34,122,232,224,8,208,242,169,2084
1390 DATA 8,24,101,142,101,142,32,124,
122,162,0,177,132,149,203,169,9283
1400 DATA 1,32,134,122,232,224,8,208,2
42,96,24,101,132,133,132,144,9055
1410 DATA 16,230,133,96,133,143,56,165
,132,229,143,133,132,176,2,198,9929
1420 DATA 133,96,160,32,160,32,197,197
,0,0,10,10,0,10,164,141,1386
1430 DATA 166,140,32,101,121,32,244,13
3,160,0,132,138,185,193,0,162,8192
1440 DATA 0,201,197,208,5,133,138,76,1
14,124,201,160,208,3,76,114,7769
1450 DATA 124,152,24,133,159,125,154,1
22,168,185,203,0,201,69,208,2,8407
1460 DATA 169,32,134,153,221,148,122,2
08,7,164,159,134,138,32,43,123,7133
1470 DATA 166,153,232,164,159,224,6,20
8,216,200,192,5,208,190,169,7,1266
1480 DATA 133,85,244,122,239,123,169,2
,133,145,164,145,169,0,133,143,8806
1490 DATA 32,219,126,230,145,165,145,2
01,12,208,239,160,5,185,193,0,9883
1500 DATA 200,201,160,208,5,169,5,76,3
5,123,201,197,208,5,169,0,6525
1510 DATA 76,35,123,169,7,133,143,32,2
19,126,76,195,124,132,134,185,9628
1520 DATA 162,127,170,202,165,138,201,
3,240,8,201,5,240,4,201,2,7072
1530 DATA 208,12,169,12,56,229,134,133
,135,160,0,76,85,123,169,2,5094
1540 DATA 24,101,134,133,135,160,0,232
,200,152,24,105,4,133,84,165,6849
1550 DATA 135,133,85,165,138,201,0,240
,29,201,2,240,19,201,3,240,8801
1560 DATA 9,201,1,240,5,169,0,76,135,1
23,189,13,124,76,135,123,5930
1570 DATA 189,214,123,76,135,123,189,1
59,123,134,136,132,137,133,143,32,8695
1580 DATA 219,126,166,136,164,137,192,
11,208,189,164,134,169,0,133,138,9741
1590 DATA 96,6,1,1,1,1,1,1,1,1,1,3,0,6
,1,1,1912
1600 DATA 1,1,1,1,1,3,0,0,0,6,1,1,1,1,
1,3,1806
1610 DATA 0,0,0,0,0,6,1,1,1,3,0,0,0,0,
0,0,1700
1620 DATA 0,6,1,3,0,0,0,0,4,1,1,1,1,1,
1,1,1774
1630 DATA 1,1,2,0,4,1,1,1,1,1,1,1,2,0,
0,0,1748
1640 DATA 4,1,240,123,235,124,1,1,1,1,
2,0,0,0,0,0,4833
1650 DATA 4,1,1,1,2,0,0,0,0,0,0,0,4,1,
2,0,1769
1660 DATA 0,0,0,0,5,5,5,5,5,5,5,5,5,0,
0,0,2065
1670 DATA 5,5,5,5,5,5,5,0,0,0,0,0,5,5,
5,5,2100
1680 DATA 5,0,0,0,0,0,0,0,5,5,5,0,0,0,
0,0,1835
1690 DATA 0,0,0,0,5,0,0,0,0,0,9,7,5,3,
1,3,2068
1700 DATA 4,5,6,7,0,159,157,155,153,15
1,149,147,145,240,242,244,4122
1710 DATA 246,248,250,252,0,0,0,0,0,0,
0,0,0,0,0,0,4210
1720 DATA 0,2,4,6,8,10,0,0,136,132,134
,185,68,124,133,139,4937
1730 DATA 185,73,124,133,85,185,162,12
7,170,202,133,134,166,134,160,0,9705
1740 DATA 152,133,84,230,84,230,84,200
,232,132,137,134,136,165,138,201,3205
1750 DATA 197,208,5,169,0,76,165,124,1
89,13,124,133,143,164,84,32,6375
```

```
1760 DATA 219,126,164,137,166,136,192,
10,208,214,198,139,165,139,240,9,2034
1770 DATA 230,85,166,134,160,0,76,138,
124,32,221,127,165,194,133,163,495
1780 DATA 165,146,133,164,165,155,201,
1,240,33,32,70,128,162,0,169,6780
1790 DATA 0,157,0,117,232,208,248,165,
140,24,105,145,141,1,208,24,7782
1800 DATA 165,141,236,124,231,125,24,1
05,74,168,169,1,153,0,117,162,6917
1810 DATA 1,189,137,132,56,233,128,188
,142,132,153,147,129,232,224,5,1616
1820 DATA 208,239,164,144,185,137,132,
190,142,132,157,147,129,169,62,141,108
3
1830 DATA 47,2,32,170,126,166,140,164,
141,32,101,121,32,244,133,173,9322
1840 DATA 120,2,201,15,240,236,173,120
,2,201,14,208,40,165,163,201,583
1850 DATA 32,240,14,201,69,240,10,201,
197,208,3,76,67,134,76,112,7148
1860 DATA 125,166,144,202,189,12,134,1
68,24,185,140,0,125,4,134,153,6601
1870 DATA 140,0,76,185,125,201,13,208,
43,165,164,201,32,240,17,201,64
1880 DATA 69,240,13,201,197,208,3,76,6
7,134,32,178,127,76,24,125,5640
1890 DATA 166,144,202,189,12,134,168,2
4,185,140,0,125,8,134,153,140,7018
1900 DATA 0,76,185,125,201,11,208,18,3
2,207,127,230,144,165,144,201,1851
1910 DATA 5,208,32,169,1,133,144,76,18
5,125,201,7,240,3,76,24,5418
1920 DATA 125,32,207,127,198,144,165,1
44,201,0,208,7,169,4,133,144,8132
1930 DATA 76,185,125,32,191,125,76,160
,122,169,0,133,20,32,170,126,6054
1940 DATA 165,20,56,233,6,144,246,96,1
69,13,141,199,2,169,0,141,6867
1950 DATA 198,2,133,189,169,4,141,196,
2,169,144,141,193,2,169,2,6819
1960 DATA 141,111,232,125,227,126,2,16
9,52,141,48,2,169,126,141,49,6066
1970 DATA 2,169,120,141,1,2,169,89,141
,0,2,169,192,141,14,212,6537
1980 DATA 169,3,141,10,208,141,11,208,
169,104,141,2,208,169,116,141,9447
1990 DATA 3,208,162,0,169,0,157,0,116,
157,0,117,169,255,157,0,6979
2000 DATA 118,157,0,119,232,208,237,16
9,112,141,7,212,169,2,141,29,8170
2010 DATA 208,96,112,112,240,70,188,13
4,12,32,2,0,2,2,48,128,474
2020 DATA 78,0,144,14,14,14,14,14,14,1
4,14,14,142,14,14,14,6014
2030 DATA 14,14,14,14,14,142,14,14,14,
14,14,14,14,142,14,14,6494
2040 DATA 14,14,14,142,14,14,14,142,14
,14,142,14,142,14,142,142,2520
2050 DATA 14,14,142,14,142,14,14,14,14
2,14,14,14,14,14,142,14,8050
2060 DATA 14,14,14,14,14,14,142,14,14,
14,14,14,14,14,14,142,6908
2070 DATA 14,14,14,14,14,14,14,14,14,1
4,142,112,66,0,0,2,6636
2080 DATA 2,2,2,2,2,65,52,126,32,0,103
,162,255,165,191,240,9567
2090 DATA 17,232,181,165,197,140,208,6
,181,172,197,141,240,5,228,191,3977
2100 DATA 208,239,96,169,113,141,159,1
26,169,131,141,160,126,169,69,133,751
2110 DATA 223,169,0,141,0,210,76,79,13
4,185,86,127,133,128,169,0,7241
2120 DATA 133,129,228,126,223,127,6,12
8,38,129,6,128,38,129,165,128,6699
2130 DATA 24,105,0,133,130,165,129,105
,80,133,131,6,128,38,129,6,4069
2140 DATA 128,38,129,165,128,24,101,13
0,133,130,165,129,101,131,133,131,9020
2150 DATA 165,130,24,101,85,105,12,133
,130,144,2,230,131,[illegible]4,143,185,9750
2160 DATA 169,127,[illegible],128,169,0,133,12
9,133,162,24,105,127,133,129,165,8731
2170 DATA 128,24,105,98,133,128,144,2,
230,129,164,162,177,128,160,0,9111
2180 DATA 145,130,230,162,165,162,201,
8,240,13,165,130,24,105,20,133,7066
2190 DATA 130,144,231,230,131,208,227,
96,1,5,13,21,29,37,45,53,1353
2200 DATA 61,69,77,85,0,0,0,0,0,0,0,0,
255,191,255,247,6736
2210 DATA 255,255,223,255,255,127,63,3
1,15,7,3,1,255,254,254,248,2289
2220 DATA 240,224,192,128,1,3,7,15,31,
63,127,255,87,255,213,213,858
2230 DATA 213,255,87,87,128,192,224,24
0,248,252,254,255,0,129,227,255,8739
2240 DATA 255,195,129,0,0,11,22,33,44,
55,66,0,8,16,24,32,6628
2250 DATA 40,48,56,64,169,70,141,0,210
,169,15,133,154,169,0,133,6899
2260 DATA 158,173,120,2,133,188,32,170
,126,165,188,205,120,2,240,246,2945
2270 DATA 96,169,0,141,0,210,169,8,133
,154,169,32,133,158,96,169,8840
2280 DATA 0,133,224,127,219,128,128,16
9,80,133,129,169,0,133,130,169,9850
2290 DATA 144,133,131,165,128,24,105,8
0,133,128,144,2,230,129,160,14,7417
2300 DATA 169,89,133,143,162,32,177,12
8,145,130,200,192,25,208,247,160,3494
2310 DATA 14,165,128,24,105,20,133,128
,144,2,230,129,165,130,24,105,7133
2320 DATA 40,133,130,144,2,230,131,198
,143,165,143,201,0,208,215,96,2078
2330 DATA 169,137,133,129,169,0,133,12
8,160,0,169,0,145,128,200,192,9536
2340 DATA 0,208,249,230,129,165,129,20
1,160,208,237,96,165,140,133,160,4508
2350 DATA 165,141,133,161,169,2,133,15
6,162,0,169,16,133,132,169,144,8801
2360 DATA 133,133,164,144,134,136,165,
144,201,1,240,9,201,4,240,5,8536
2370 DATA 138,24,105,10,170,181,203,20
1,160,208,11,185,26,134,170,214,1925
2380 DATA 160,32,194,128,246,160,166,1
36,165,144,201,2,240,9,201,3,9607
2390 DATA 240,5,138,24,105,10,170,181,
203,201,160,208,11,185,26,134,9733
2400 DATA 170,246,160,32,194,128,214,1
60,166,136,181,193,201,160,240,20,4120
2410 DATA 24,185,21,134,170,181,160,12
1,16,134,149,160,166,136,232,224,3596
2420 DATA 6,208,161,96,32,194,128,96,7
2,138,72,152,72,165,161,10,7162
2430 DATA 133,128,169,0,42,133,129,6,1
28,38,129,6,128,165,128,133,6330
2440 DATA 130,38,220,128,215,129,129,1
65,129,133,131,6,128,38,129,6,6121
2450 DATA 128,38,129,165,128,24,101,13
0,133,128,165,129,101,131,133,129,9278
2460 DATA 165,132,24,105,10,24,101,128
,133,128,165,133,101,129,133,129,8372
2470 DATA 165,160,41,3,170,165,160,74,
74,24,101,128,133,128,165,129,8255
2480 DATA 105,0,133,129,164,156,189,61
,129,57,53,129,133,159,189,57,8631
2490 DATA 129,160,0,49,128,5,159,145,1
28,104,168,104,170,104,96,0,6472
2500 DATA 85,170,255,63,207,243,252,19
2,48,12,3,0,0,81,82,82,3996
2510 DATA 82,82,82,82,82,82,82,82,82,8
2,69,0,0,0,0,0,7779
2520 DATA 0,0,0,0,0,0,0,35,111,109,112
,97,115,115,0,0,390
2530 DATA 0,0,0,0,0,124,52,105,109,101
,0,37,108,97,112,115,3195
2540 DATA 101,100,124,0,0,0,0,0,0,0,0,
```

```
0,0,0,0,0,3213
2550 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,12
4,0,16,4542
2560 DATA 26,16,16,26,16,16,14,16,16,0
,124,0,0,0,0,0,4680
2570 DATA 0,0,0,0,0,0,0,0,0,46,111,114,1
16,104,0,0,0,8092
2580 DATA 0,0,0,0,0,0,90,82,82,82,82,82,
82,82,82,82,82,2550
2590 DATA 82,82,67,0,0,0,0,0,0,0,0,0,0
,0,0,0,3037
2600 DATA 0,0,216,129,211,130,92,0,0,0
,0,0,0,0,0,0,6243
2610 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2610
2620 DATA 0,0,0,0,0,0,0,0,0,55,101,115,1
16,0,94,0,95,9618
2630 DATA 0,37,97,115,116,0,0,0,0,0,0,
0,0,0,0,0,4035
2640 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2640
2650 DATA 0,0,0,0,0,0,0,93,0,0,0,0,0,0,0
,0,0,3301
2660 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2660
2670 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,51,1
11,117,116,8498
2680 DATA 104,0,0,0,0,0,0,0,0,0,0,0,0,
35,111,110,6699
2690 DATA 103,114,97,116,117,108,97,11
6,105,111,110,115,0,48,97,116,5244
2700 DATA 104,102,105,110,100,101,114,
1,1,0,0,0,0,0,0,0,5684
2710 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2710
2720 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2720
2730 DATA 0,0,0,0,0,0,0,0,0,0,0,0,30,57,
111,117,0,7140
2740 DATA 104,97,118,101,0,101,115,99,
97,112,101,100,0,116,104,101,5103
2750 DATA 0,44,97,98,121,114,105,110,1
16,104,28,0,0,0,0,0,8817
2760 DATA 0,0,212,130,207,131,0,0,0,0,
0,0,0,0,0,0,5737
2770 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2770
2780 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2780
2790 DATA 0,0,0,57,111,117,114,0,116,1
05,109,101,0,119,97,115,4539
2800 DATA 0,16,26,16,16,26,16,16,14,16
,16,0,0,0,0,0,3912
2810 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2810
2820 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2820
2830 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2830
2840 DATA 0,0,48,114,101,115,115,0,214
,180,242,233,231,231,229,242,8168
2850 DATA 194,0,116,111,0,50,101,115,1
16,97,114,116,0,0,0,0,423
2860 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,57,1
11,117,0,6910
2870 DATA 104,97,118,101,0,98,101,101,
110,0,99,97,112,116,117,114,5931
2880 DATA 101,100,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,3181
2890 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2890
2900 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2900
2910 DATA 0,0,0,0,0,0,0,0,0,57,111,117,0
,119,105,108,108,2185
2920 DATA 0,98,208,131,203,132,101,0,1
00,105,115,112,111,115,101,100,7505
2930 DATA 0,111,102,0,97,116,0,116,104
,101,0,0,0,0,0,0,7513
2940 DATA 0,0,0,0,0,0,110,101,120,116,0,
116,105,109,101,0,99,3262
2950 DATA 111,110,118,105,101,110,101,
110,116,0,102,111,114,0,121,111,5378
2960 DATA 117,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,57,3989
2970 DATA 111,117,114,0,116,105,109,10
1,0,119,97,115,0,16,26,16,945
2980 DATA 16,26,16,16,14,16,16,0,0,0,0
,0,0,0,0,0,3438
2990 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,2990
3000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,3000
3010 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
48,114,5554
3020 DATA 101,115,115,0,214,180,242,23
3,231,231,229,242,194,0,116,111,5254
3030 DATA 0,50,101,115,116,97,114,116,
0,0,0,0,0,0,0,6781
3040 DATA 220,222,221,223,0,69,108,149
,110,165,132,133,128,165,133,133,1406
3050 DATA 129,162,0,160,0,169,160,145,
128,200,192,49,208,249,169,0,2014
3060 DATA 145,128,24,165,128,105,50,13
3,128,144,2,230,129,232,224,35,1096
3070 DATA 208,225,169,0,133,130,165,13
2,133,128,165,133,133,129,24,165,314
3080 DATA 128,105,204,132,199,133,53,1
33,128,144,2,230,129,160,0,169,9781
3090 DATA 5,145,128,173,10,210,41,7,13
3,153,165,153,56,233,4,176,9358
3100 DATA 242,165,153,133,143,165,128,
133,130,165,129,133,131,165,153,201,36
87
3110 DATA 0,208,14,24,165,130,105,2,13
3,130,144,62,230,131,76,64,7833
3120 DATA 133,165,153,201,1,208,14,56,
165,130,233,100,133,130,176,42,54
3130 DATA 198,131,76,64,133,165,153,20
1,2,208,14,56,165,130,233,2,8824
3140 DATA 133,130,176,22,198,131,76,64
,133,165,153,201,3,208,11,24,7411
3150 DATA 165,130,105,100,133,130,144,
2,230,131,160,0,177,130,201,160,1595
3160 DATA 208,35,165,153,24,105,1,160,
0,201,0,208,2,169,32,145,6280
3170 DATA 130,32,166,133,160,0,169,32,
145,128,165,130,133,128,165,131,685
3180 DATA 133,129,76,217,132,165,153,2
01,3,208,7,169,0,133,153,76,8581
3190 DATA 122,133,230,153,165,153,197,
143,240,3,76,235,132,160,0,177,1780
3200 DATA 128,133,153,169,32,145,128,1
65,153,56,233,5,176,95,198,153,1571
3210 DATA 166,153,189,0,134,133,153,32
,166,133,32,166,133,76,217,132,372
3220 DATA 165,153,201,0,208,14,24,165,
128,105,1,133,128,144,62,230,9005
3230 DATA 129,76,243,133,165,153,201,1
,208,14,56,165,128,233,50,133,342
3240 DATA 128,176,200,133,195,134,42,1
98,129,76,243,133,165,153,201,2,2033
3250 DATA 208,14,56,165,128,233,1,133,
128,176,22,198,129,76,243,133,1467
3260 DATA 165,153,201,3,208,11,24,165,
128,105,50,133,128,144,2,230,8678
3270 DATA 129,96,162,0,181,193,149,223
,232,224,7,208,247,96,2,3,501
3280 DATA 0,1,255,255,1,1,1,1,255,255,
1,0,1,0,0,255,4042
3290 DATA 255,1,1,0,1,0,1,0,0,0,1,0,1,
8,7,5,3883
3300 DATA 4,2,10,6,10,6,10,110,110,110
,110,122,0,3,3,3,9297
3310 DATA 3,0,255,126,60,24,24,84,92,1
00,108,116,154,146,138,130,8470
3320 DATA 122,169,89,141,159,126,169,1
```

```
30,141,160,126,133,131,173,160,126,277
7
3330 DATA 133,131,173,159,126,24,105,1
80,133,130,144,2,230,131,162,0,9188
3340 DATA 165,223,201,69,240,42,169,25
5,133,138,169,3,141,8,208,141,1298
3350 DATA 9,208,32,22,126,162,166,169,
255,232,157,0,116,157,0,117,9995
3360 DATA 224,229,208,245,169,32,141,0
,208,169,192,141,1,208,162,0,391
3370 DATA 181,148,24,105,16,188,31,134
,145,130,232,224,5,208,241,165,3913
3380 DATA 223,201,69,240,11,173,10,210
,141,18,208,141,19,208,230,159,2597
3390 DATA 173,132,2,201,1,240,232,76,0
,105,0,0,0,0,0,108,1092
3400 DATA 97,98,196,134,91,135,121,114
,105,110,116,104,115,0,0,0,3905
3410 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,3410
3420 DATA 0,0,0,0,0,0,0,0,0,0,0,0,52,104,1
01,0,53,108,9076
3430 DATA 116,105,109,97,116,101,0,19,
13,36,0,45,97,122,101,0,1310
3440 DATA 0,33,100,118,101,110,116,117
,114,101,0,0,0,0,0,0,9227
3450 DATA 0,0,0,8,35,9,0,17,25,24,23,0
,0,33,114,103,8385
3460 DATA 111,51,111,102,116,0,48,114,
111,100,117,99,116,105,111,110,7119
3470 DATA 115,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,3585
3480 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,3480
3490 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,103,5138
3500 DATA 251,103,173,252,2,201,28,208
,3,76,0,105,165,187,201,0,8385
3510 DATA 240,3,76,106,103,169,16,133,
187,166,189,169,0,133,156,32,9277
3520 DATA 220,103,165,144,133,186,166,
189,181,179,133,144,181,172,168,181,62
59
3530 DATA 165,170,169,32,133,190,32,43
,104,166,189,181,172,168,181,165,3833
3540 DATA 170,32,101,121,166,189,165,1
86,133,144,165,194,201,32,208,58,3057
3550 DATA 32,246,103,166,189,169,3,133
,156,230,189,164,189,136,196,191,6199
3560 DATA 240,6,32,220,103,76,19,103,1
69,0,133,189,169,69,133,190,166
3570 DATA 166,189,228,191,240,15,181,1
72,168,181,165,170,32,43,104,230,2930
3580 DATA 189,197,191,208,231,169,0,13
3,189,96,173,10,210,24,105,128,174
3590 DATA 144,18,165,203,201,160,240,3
,76,188,103,165,213,201,160,240,6246
3600 DATA 21,76,204,103,165,213,201,16
0,240,3,76,204,103,165,203,201,4971
3610 DATA 160,240,3,76,188,103,173,10,
210,24,105,128,144,16,181,179,9908
3620 DATA 168,185,199,103,149,179,76,8
1,103,0,2,3,4,1,181,179,4796
3630 DATA 168,185,215,103,149,179,76,8
1,103,0,4,1,2,3,181,165,4630
3640 DATA 133,160,181,172,133,161,169,
16,133,132,169,144,133,133,165,155,291
6
3650 DATA 201,0,208,3,32,194,128,96,16
6,189,181,179,170,202,252,103,5464
3660 DATA 74,104,189,18,104,24,101,189
,168,24,185,165,0,125,14,104,6855
3670 DATA 153,165,0,96,255,255,1,1,7,0
,7,0,20,48,48,2,9181
3680 DATA 48,9,11,20,34,2,20,20,2,2,4,
4,4,1,4,3,4645
3690 DATA 3,169,92,133,132,169,135,133
,133,136,192,0,240,8,169,50,9758
3700 DATA 32,124,122,76,51,104,138,32,
124,122,160,0,165,190,145,132,9939
3710 DATA 96,0,105,251,105,169,64,141,
14,212,169,0,141,0,212,141,9614
3720 DATA 1,210,162,0,142,8,208,142,9,
208,149,223,232,224,6,208,4023
3730 DATA 249,142,0,208,142,1,208,162,
0,189,22,104,149,165,232,224,3254
3740 DATA 20,208,246,169,255,133,138,1
69,0,133,145,133,192,76,149,105,1977
3750 DATA 72,138,72,152,72,165,192,201
,0,208,16,162,144,142,22,208,942
3760 DATA 232,141,10,212,224,159,208,2
45,76,131,105,201,1,240,4,201,2852
3770 DATA 2,208,17,162,0,142,22,208,23
2,232,141,10,212,224,16,208,3096
3780 DATA 244,76,131,105,164,145,185,7
6,108,141,24,208,230,145,165,145,3539
3790 DATA 201,6,208,4,169,0,133,145,23
0,192,165,192,201,9,208,4,1611
3800 DATA 169,0,133,192,104,168,104,17
0,104,64,169,91,141,48,2,169,8518
3810 DATA 106,141,49,2,169,59,141,0,2,
169,105,141,1,2,169,192,6742
3820 DATA 141,14,212,169,0,141,198,2,1
69,34,141,47,2,165,155,76,7402
3830 DATA 48,106,164,191,162,0,185,144
,108,168,185,82,108,201,91,240,3697
3840 DATA 8,157,242,107,200,232,76,197
,105,165,191,24,105,16,141,167,1176
3850 DATA 107,173,31,208,201,5,208,7,1
69,1,133,159,76,0,106,201,8471
3860 DATA 3,240,18,201,6,208,226,169,0
,141,252,2,169,0,133,138,19
3870 DATA 133,252,105,247,106,159,76,0
,120,169,48,141,0,210,169,15,8531
3880 DATA 141,28,2,173,28,2,24,105,160
,141,1,210,201,160,208,243,3177
3890 DATA 141,1,210,165,159,201,0,240,
41,169,0,133,159,230,155,165,3151
3900 DATA 155,201,2,208,4,169,0,133,15
5,168,162,0,185,169,108,168,1329
3910 DATA 185,151,108,201,91,208,3,76,
189,105,157,194,107,200,232,76,3550
3920 DATA 55,106,230,191,165,191,201,7
,240,3,76,189,105,169,0,133,228
3930 DATA 191,76,189,105,112,112,112,6
6,128,106,2,240,7,0,240,6,6705
3940 DATA 128,6,112,2,112,2,112,2,112,
240,0,130,0,130,112,112,6056
3950 DATA 130,0,130,0,130,0,65,91,106,
0,0,0,0,0,33,114,9576
3960 DATA 103,111,51,111,102,116,0,51,
111,102,116,119,97,114,101,0,5591
3970 DATA 48,114,111,100,117,99,116,10
5,111,110,115,0,0,0,0,0,1174
3980 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,48
,114,111,8138
3990 DATA 117,100,108,121,0,48,114,101
,115,101,110,116,115,0,0,0,3151
4000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
44,33,5188
4010 DATA 34,57,50,41,46,52,40,51,0,0,
0,0,0,0,0,52,6534
4020 DATA 40,37,0,53,44,52,41,45,33,52
,37,0,19,13,36,0,7718
4030 DATA 0,248,106,243,107,0,0,0,45,3
3,58,37,0,33,36,54,34
4040 DATA 37,46,52,53,50,37,0,0,0,0,0,
0,0,0,0,0,5009
4050 DATA 0,0,33,14,46,14,33,14,44,14,
47,14,39,14,0,35,7346
4060 DATA 111,109,112,117,116,105,110,
103,0,0,0,0,0,0,0,0,7997
4070 DATA 0,0,0,0,0,0,48,114,111,103,1
14,97,109,109,101,100,5823
4080 DATA 0,0,34,121,0,51,116,101,118,
101,110,0,44,97,115,104,5193
```

```
4090 DATA 111,119,101,114,0,0,0,0,0,0,
8,35,9,0,17,25,6478
4100 DATA 24,23,0,33,114,103,111,51,11
1,102,116,0,51,111,102,116,5573
4110 DATA 119,97,114,101,0,48,114,111,
100,117,99,116,105,111,110,115,8103
4120 DATA 0,0,0,0,0,46,117,109,98,101,
114,0,111,102,0,33,2632
4130 DATA 114,103,111,110,105,97,110,0
,48,97,116,114,111,108,108,101,7337
4140 DATA 114,115,13,30,16,0,0,0,0,0,0
,0,57,111,117,0,8773
4150 DATA 55,105,108,108,0,33,116,116,
101,109,112,116,0,33,0,0,2194
4160 DATA 0,0,0,0,0,0,0,0,0,37,115,99,97
,112,101,0,0,766
4170 DATA 0,0,0,214,175,240,244,233,23
9,238,194,0,36,105,102,102,2678
4180 DATA 105,99,117,108,116,121,0,44,
101,118,101,108,0,13,0,0,1602
4190 DATA 0,244,107,170,108,0,0,0,0,0,
0,0,0,0,0,214,9643
4200 DATA 179,229,236,229,227,244,194,
0,36,97,121,116,105,109,101,0,8841
4210 DATA 111,114,0,46,105,103,104,116
,105,109,101,0,37,115,99,97,5806
4220 DATA 112,101,0,0,0,0,0,214,179,24
4,225,242,244,128,194,0,3550
4230 DATA 37,110,116,101,114,0,116,104
,101,0,44,97,98,121,114,105,6368
4240 DATA 110,116,104,0,0,0,0,0,0,0,0,
0,0,194,178,162,2872
4250 DATA 146,130,0,34,101,103,105,110
,110,101,114,91,37,97,115,121,7376
4260 DATA 0,0,0,0,91,45,111,100,101,11
4,97,116,101,91,40,97,5809
4270 DATA 114,100,0,0,0,0,91,52,114,10
5,118,105,97,108,91,51,5225
4280 DATA 97,100,105,115,116,105,99,91
,41,110,115,97,110,101,0,0,4725
4290 DATA 91,0,9,18,27,36,44,53,36,97,
121,108,105,103,104,116,5707
4300 DATA 91,46,105,103,104,116,105,10
9,101,91,0,9,226,2,227,2,6363
4310 DATA 0,105,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,4520
```

●

Listing 2
Assembly listing

```
        .OPT NO LIST
        .OPT OBJ
;-------------------------
;Labyrinths: Revision 09/15/87
;Programmed By Steven Lashower
;(C) 1987 Argosoft Productions
;-------------------------
        *=  $80
;--------------------
;Page Zero Variables
;--------------------
LO      .DS 2       ;Srce Copy adrs
DESTLO  .DS 2       ;Dest Copy adrs
MTEMP   .DS 2       ;Maze location
YTEMP   .DS 1       ;Y Coord Temp
COLTMP  .DS 1       ;X Coord Temp
XTMP    .DS 1       ;X Coord Temp
YTMP    .DS 1       ;Y Coord Temp
FLAG    .DS 1       ;Variable Flag
LEN     .DS 1       ;Maze Draw Lngth
H       .DS 1       ;Player X Coord
V       .DS 1       ;Player Y Coord
W       .DS 1       ;Maze width
TEMP    .DS 1       ;Temporary Data
DIRECTION .DS 1     ;Player Dirction
DUMMY   .DS 1       ;Temporary Data
BACK    .DS 1       ;Behnd plyr data
COUNTR  .DS 1       ;Main DLI pointr
SEC1    .DS 5       ;Time "h:MM:ss"
J       .DS 1       ;Maze gen varble
VOLUM1  .DS 1       ;Sound volume
TIME    .DS 1       ;Day/Night Flag
COLOR   .DS 1       ;Draw/Plot Color
OFF     .DS 1       ;Patroler Bottom
TONE    .DS 1       ;Tone for sound
HOLD    .DS 1       ;Temp storage
PLOTX   .DS 1       ;X Coord Value
PLOTY   .DS 1       ;Y Coord Value
COPCNT  .DS 1       ;Byte copy count
FRONT2  .DS 1       ;Center cpy byte
BACK2   .DS 1       ;Copy of BACK
H2      .DS 7       ;Patrol X Coords
V2      .DS 7       ;Patrol Y Coords
DIR2    .DS 7       ;Patrol Drection
DIRTMP  .DS 1       ;Temp. DIRECTION
CDTMV4  .DS 1       ;Patrol Move Chk
OLDSTK  .DS 1       ;Stick 0 Copy
MNUM    .DS 1       ;Patroller #
PAT     .DS 1       ;Patrol Plot Var
DIFLEV  .DS 1       ;Diffculty level
POINTER .DS 1       ;Title DLI pntr
CENTER  .DS $0A     ;Center data
LEFT    .DS $0A     ;Left side data
RIGHT   .DS $0A     ;Right side data
CENTER2 .DS $0A     ;Copy of CENTER
RTCLOK  =   $14
ROWCRS  =   $54
COLCRS  =   $55
VDSLST  =   $0200
CDTMV3  =   $021C
SDMCTL  =   $022F
SDLSTL  =   $0230
GPRIOR  =   $026F
STICK0  =   $0278
STRIG0  =   $0284
PCOLR0  =   $02C0
COLOR0  =   $02C4
COLOR2  =   $02C6
COLOR3  =   $02C7
CH      =   $02FC
HPOSP0  =   $D000
HPOSP1  =   $D001
SIZEP0  =   $D008
SIZEP1  =   $D009
SIZEP2  =   $D00A
SIZEP3  =   $D00B
COLPM0  =   $D012
COLPM1  =   $D013
COLPM2  =   $D014
COLPM3  =   $D015
COLPF0  =   $D016
COLPF1  =   $D017
COLPF2  =   $D018
GRACTL  =   $D01D
CONSOL  =   $D01F
AUDF1   =   $D200
AUDC1   =   $D201
RANDOM  =   $D20A
DMACTL  =   $D400
PMBASE  =   $D407
WSYNC   =   $D40A
NMIEN   =   $D40E
SETVBV  =   $E45C
XITVBV  =   $E462
DISP    =   $9000
SCREEN  =   $5000
TIMSC   =   BOTTOM+$52
PRINT1  =   TEXT+$0172
PRINT2  =   TEXT+$0127
PRINT3  =   TEXT+$0142
PMG     =   $7000
PL1     =   $7500
        *=  $7800
;--------------------
;Get things ready...
;--------------------
LABYRINTHS LDA #$00  ;Turn screen
        STA SDMCTL  ;off.
        STA FLAG
        LDA #$FF
        STA VOLUM1
        JSR CLSCN   ;Clear scrren.
        LDA # <MAZE
        STA MTEMP   ;Copy maze
        LDA # >MAZE ;Addresss.
        STA MTEMP+1
        JSR GENMZE  ;Generate Maze!
        LDA #$C5    ;Exit in lower
        STA MAZE+$06D3 ;right corner
        LDA # <BOTTOM
        STA TEXTB   ;Print timer &
        LDA # >BOTTOM ;compass at
        STA TEXTB+1 ;screen bottom.
        LDA #$00
        LDX #$00
CLTIME  STA SEC1,X  ;Set elapsed
        INX         ;time to 0:00:00
        CPX #$05
        BNE CLTIME
        STA COUNTR
        LDA #$3C    ;Set timer for 1
        STA CDTMV3  ;Second.
        LDY # <VBI
        LDX # >VBI
        LDA #$07    ;Initialize VBI.
        JSR SETVBV
        LDA #$01    ;Player starts
        STA H       ;game at (1,2).
        LDA #$02
        STA V
        LDA #$03    ;Player's facing
        STA DIRECTION ;south,
        LDA #$32    ;Maze Width = 50
        STA W
        JSR SETUP   ;Set up screen.
        JMP MAINPROG ;Main Program.
;----------------------
;Display List Interrupt
;----------------------
DLI     PHA         ;Save Accum and
        TXA         ;X Register.
        PHA
        LDX COUNTR  ;Which DLI?
        CPX #$00
        BNE DLI1
        INX         ;Point to next
        STX COUNTR  ;DLI.
        LDX #$90
DLI0    STX COLPF1  ;Shade game
        INX         ;title in 8
        INX         ;shades of blue.
        STA WSYNC   ;Wait for SYNC.
        CPX #$A0    ;Shaded yet?
        BNE DLI0    ;Nope...
        LDA #$0A    ;Set Text Luma.
        STA COLPF1
        JMP ENDDLI  ;Return from DLI
DLI1    LDA #$0A    ;Set maze wall
        STA COLPF2  ;colors to
        LDA #$04    ;shades of grey.
        STA COLPF1
        LDX COUNTR
        LDA TIME    ITS
        CMP #$00    ;Daytime?
        BNE DLI2    ;Nope...
        LDA COLOR5,X ;Yep!  Dayglow
        JMP DLI3    ;colors.
DLI2    LDA COLOR52,X ;Night colors
DLI3    STA COLPM2  ;Store colors
        STA COLPM3  ;in players 2&3.
DLI3B   INX         ;Point to next
        STX COUNTR  ;DLI...
```

```
         CPX #$11    ;All colors yet?
         BNE ENDDLI  ;Nope! Exit DLI.
         LDA #$00    ;Reset pointer
         STA COUNTR  ;to screen top.
         LDA #$F8
         STA COLPF0  ;Set all colors
         LDA #$0A    ;for bottom
         STA COLPF1  ;of screen.
         LDA #$00
         STA COLPF2
ENDDLI   PLA         ;Restore X reg.
         TAX
         PLA         ;Restore Accum.
         RTI
;-----------------------
;Vertical Blank Interrupt
;-----------------------
VBI      INC PCOLR0  ;Increase
         INC PCOLR0+1 ;Patroller and
         LDA PCOLR0+1 ;player in maze
         CMP #$9F
         BNE XX1     ;Make sure
         LDA #$90    ;Player flashes
         STA PCOLR0+1 ;blue.
XX1      LDA CDTMV4  ;Decrease timer
         CMP #$00    ;if and only if
         BEQ XX2     ;not zero.
         DEC CDTMV4
XX2      LDA FLAG    ;Flag set?
         CMP #$FF
         BNE XX3     ;Yep. Exit VBI!
         JMP XITVBV
XX3      LDA CDTMV3  ;Second elapsed?
         CMP #$00
         BNE X1      ;Nope...
         LDA #$3C    ;Ya! Reset
         STA CDTMV3  ;timer!
         INC SEC1    ;Increase second
         LDX #$00    ;digit.
X01      LDA SEC1,X  ;Check digits
         CMP COMP,X  ;for carry.
         BNE X0      ;No carry.
         LDA #$00    ;Carry! Zero
         STA SEC1,X  ;digit. Inc next
         INC SEC1+1,X ;digit.
X0       INX
         CPX #$05    ;Done checking?
         BNE X01     ;Nope...
X1       LDX #$00    ;Ya! print time.
PRTIME   LDA SEC1,X  ;Load digit to
         CLC         ;print and add
         ADC #$10    ;ATASCII offset.
         LDY YVAL,X  ;Add offset and
         STA TIM5C,Y ;print it!
         INX
         CPX #$05    ;All printed?
         BNE PRTIME  ;Nope...
         LDA VOLUM1  ;Play a sound?
         CMP #$FF
         BEQ X2      ;Nope...
         DEC VOLUM1  ;Yes! Decrease
         CLC         ;volume of sound
         ADC TONE    ;and add
         STA AUDC1   ;distortion.
X2       LDX #$00
DI1      LDA CENTER2,X ;Can player
         CMP #$45    ;view patroller?
         BEQ DI2     ;Yes...
         CMP #$A0    ;Is a wall in
         BNE DL1A    ;the way?
         LDA #$00    ;If one is, then
         STX HPOSP0  ;don't display
         JMP XITVBV  ;it so exit VBI.
DL1A     INX
         CPX #$05    ;All view check?
         BNE DI1     ;Nope...
DI2      LDA HARGO,X ;Set horizontal
         STA HPOSP0  ;position & size
         LDA SARGO,X ;depending its
         STA SIZEP0  ;distance "X".
         LDA OFARG,X ;Where's its
         STA OFF     ;bottom?
         LDY #$00
         LDA #$00    ;Erase player's
VB1      STA PMG+$0400,Y ;bytes.
         INY
         BNE VB1
         LDA ONARG,X ;Find patrollers
         TAY         ;top and put
VB2      LDA BARGO,X ;bytes into
         STA PMG+$0400,Y ;PMAREA.
         INY
         CPY OFF     ;Bottom reached?
         BNE VB2     ;No, next byte.
         JMP XITVBV  ;Ya, exit VBI
;-------------------------
;Read Surrounding Maze Data
;-------------------------
START    LDA # <MAZE ;Get maze
         STA MTEMP   ;address and
         LDA # >MAZE ;copy it.
         STA MTEMP+1
S0       DEY         ;Decrease "Y"
         CPY #$00    ;Zero yet?
         BEQ S1      ;Yep.
         LDA W       ;Add maze width.
         JSR ADD
         JMP S0      ;Do it again.
S1       TXA         ;Add horizontal.
         JSR ADD
         LDX #$00
         LDA DIRECTION ;Facing north?
         CMP #$01
         BNE T1      ;Nope...
         LDA W       ;By adding maze
         JSR ADD     ;width, we can
         LDY #$00    ;get the space
         LDA (MTEMP),Y ;in back of
         STA BACK    ;player.
         LDA W
         CLC         ;Subtract width.
         ADC #$01
         JSR SUBTRACT
         LDY #$00    ;Get 8 bytes
S45      LDA (MTEMP),Y ;for
         STA LEFT,X  ;left
         INY
         LDA (MTEMP),Y
         STA CENTER,X ;center
         INY
         LDA (MTEMP),Y
         STA RIGHT,X ;& right walls.
         INX
         LDA W       ;Get ready
         JSR SUBTRACT ;for next.
         CPX #$08    ;Add 8 bytes?
         BNE S45     ;Nope...
         RTS         ;
T1       LDA DIRECTION ;Facing south
         CMP #$03
         BNE T2
         LDY #$00
         LDA W
         JSR SUBTRACT
         LDA (MTEMP),Y
         STA BACK
         LDA W
         SEC
         SBC #$01
         JSR ADD
         LDY #$00
T45      LDA (MTEMP),Y
         STA RIGHT,X
         INY
         LDA (MTEMP),Y
         STA CENTER,X
         INY
         LDA (MTEMP),Y
         STA LEFT,X
         INX
         LDA W
         JSR ADD
         CPX #$08
         BNE T45
         RTS
T2       LDA DIRECTION ;Facing east
         CMP #$04
         BNE T3
         LDY #$00
U1       LDA (MTEMP),Y
         STA CENTER,Y
         INY
         CPY #$08
         BNE U1
         LDY #$00
         LDA #$01
         JSR SUBTRACT
         LDA (MTEMP),Y
         STA BACK
         LDA #$01
         JSR ADD
         LDA W
         JSR SUBTRACT
U4       LDA (MTEMP),Y
         STA LEFT,Y
         INY
         CPY #$08
         BNE U4
         LDA W
         CLC
         ADC W
         JSR ADD
         LDY #$00
U6       LDA (MTEMP),Y
         STA RIGHT,Y
         INY
         CPY #$08
         BNE U6
         RTS
T3       LDY #$00
         LDX #$00
V1       LDA (MTEMP),Y ;Facing West
         STA CENTER,X
         LDA #$01
         JSR SUBTRACT
         INX
         CPX #$08
         BNE V1
         LDA #$09
         JSR ADD
         LDA (MTEMP),Y
         STA BACK
         LDA #$01
         CLC
         ADC W
         JSR SUBTRACT
         LDX #$00
V4       LDA (MTEMP),Y
         STA RIGHT,X
         LDA #$01
         JSR SUBTRACT
         INX
         CPX #$08
         BNE V4
         LDA #$08
         CLC
         ADC W
         ADC W
         JSR ADD
         LDX #$00
V6       LDA (MTEMP),Y
         STA LEFT,X
         LDA #$01
         JSR SUBTRACT
         INX
         CPX #$08
         BNE V6
         RTS
ADD      CLC         ;Add Accumulator
         ADC MTEMP   ;to MTEMP with
         STA MTEMP   ;carry.
         BCC SUB2
         INC MTEMP+1
         RTS
SUBTRACT STA TEMP    ;Subtracts
         SEC         ;TEMP from
         LDA MTEMP   ;MTEMP with
         SBC TEMP    ;carry.
         STA MTEMP
         BCS SUB2
         DEC MTEMP+1
SUB2     RTS
WCOMP    .BYTE $A0,$20,$A0,$20
         .BYTE $C5,$C5
WALO     .BYTE $00,$00,$0A,$0A
         .BYTE $00,$0A
;---------------------------
;WALL DATA/VIEWING PROCESSOR
;---------------------------
MAINPROG LDY V       ;Get player's
         LDX H       ;coords and pass
         JSR START   ;to routine.
         JSR COPCEN  ;Copy CENTER.
         LDY #$00
         STY FLAG
LOOP     LDA CENTER,Y ;Get byte in
         LDX #$00    ;front of player
         CMP #$C5    ;Is it an exit?
         BNE L00     ;No...
         STA FLAG    ;Yes, set flag
         JMP FRONTWALL ;and draw it!
L00      CMP #$A0    ;Is it a wall?
         BNE N1A2    ;No.
         JMP FRONTWALL ;Yes! Draw it!
N1A2     TYA
         CLC
         STA HOLD
         ADC WALO,X  ;Offset for LEFT
         TAY         ;and RIGHT side.
         LDA LEFT,Y  ;Get side wall.
         CMP #$45    ;Patrollr there?
         BNE N1A25   ;No.
         LDA #$20    ;Yes...Print
N1A25    STX J       ;passage there.
         CMP WCOMP,X
         BNE N1A3
         LDY HOLD
         STX FLAG
         JSR LEFTWALL
N1A3     LDX J
         INX
         LDY HOLD
         CPX #$06
         BNE N1A2
         INY
         CPY #$05
         BNE LOOP
         LDA #$07
         STA COLCR5
         LDA #$02
         STA DUMMY
N22      LDY DUMMY
         LDA #$00
         STA TEMP
         JSR PRINT
         INC DUMMY
         LDA DUMMY
         CMP #$0C
         BNE N22
         LDY #$05
         LDA CENTER,Y
         INY
         CMP #$A0
         BNE N3
         LDA #$05
         JMP N4
N3       CMP #$C5
         BNE N3A
         LDA #$00
         JMP N4
```

(continued on page 90)

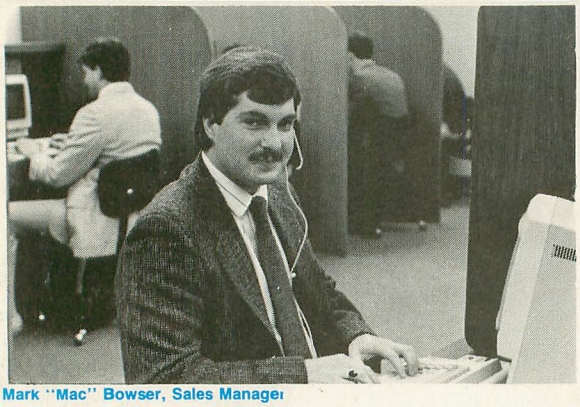
Lyco Computer

Marketing & Consultants

Since 1981

Order processed within 24 hours.

Lyco Means Total Service.



Mark "Mac" Bowser, Sales Manager

I would personally like to thank all of our past customers for helping to make Lyco Computer one of the largest mail order companies and a leader in the industry. Also, I would like to extend my personal invitation to all computer enthusiasts who have not experienced the services that we provide. Please call our trained sales staff at our toll-free number to inquire about our diverse product line and weekly specials.

First and foremost our philosophy is to keep abreast of the changing market so that we can provide you with not only factory-fresh merchandise but also the newest models offered by the manufacturers at the absolute best possible prices. We offer the widest selection of computer hardware, software and accessories.

Feel free to call Lyco if you want to know more about a particular item. can't stress enough that our toll-free number is not just for orders. Many companies have a toll-free number for ordering, but if you just want to ask a question about a product, you have to make a toll call. Not at Lyco. Our trained sales staff is knowledgeable about all the products we stock and is happy to answer any questions you may have. We will do our best to make sure that the product you select will fit your application. We also have Saturday hours — one more reason to call us for all your computer needs.

Once you've placed your order with Lyco, we don't forget about you. Our friendly, professional customer service representatives will find answers to your questions about the status of an order, warranties, product availability, or prices.

Lyco Computer stocks a multimillion dollar inventory of factory-fresh merchandise. Chances are we have exactly what you want right in our warehouse. And that means you'll get it fast. In fact, orders are normally shipped within 24 hours. Free shipping on prepaid orders over $50, and there is no deposit required on C.O.D. orders. Air freight or UPS Blue/Red Label shipping is available, too. And all products carry the full manufacturers' warranties.

I can't see why anyone would shop anywhere else. Selection from our huge in-stock inventory, best price, service that can't be beat—we've got it all here at Lyco Computer.

TO ORDER, CALL TOLL-FREE: 1-800-233-8760
New PA Wats: 1-800-233-8760
Outside Continental US Call: **1-717-494-1030**

Hours: 9AM to 8PM, Mon. - Thurs.
9AM to 6PM, Friday — 10AM to 6PM, Saturday

For Customer Service, call 1-717-494-1670,
9AM to 5PM, Mon. - Fri.
Or write: Lyco Computer, Inc.
P.O. Box 5088, Jersey Shore, PA 17740

VISA MasterCard

Risk-Free Policy: • full manufacturers' warranties • no sales tax outside PA • prices show 4% cash discount; add 4% for credit cards • APO, FPO, international: add $5 plus 3% for priority • 4-week clearance on personal checks • we check for credit card theft • compatability not guaranteed • return authorization required • price/availability subject to change • Prepaid orders under $50 in con., U.S. add $3.00.

1-800-233-8760

SOFTWARE

ATARI ST

Access:
Leader Board $22.95
Tournament #1 ..... $11.95
10th Frame $22.95

Activision:
Champion. Baseball $22.95
Champion. Basketball $22.95
Championship Golf ..... $New
GFL Football $22.95
Leather Goddesses $22.95
Music Studio $27.95
Paint Works $22.95
Shanghai $22.95
Bureaucracy $22.95
Top Fuel $19.95

Electronic Arts:
Arctic Fox $25.95
Empire $32.95
Starfleet I $32.95
Chess Master 2000 ..... $25.95
Gridiron $32.95

Epyx:
Sub Battle Simulator ... $22.95
World Games $22.95
Wrestling $22.95
Winter Games $22.95

Firebird:
Pawn $25.95
Starglider $25.95
Golden Path $25.95
Guild of Thieves $25.95
Tracker $25.95

Microleague:
Microleague Baseball .. $33.95
General Manager $16.95
Wrestling $25.95

Microprose:
Silent Service $22.95
F-15 Strike Eagle $24.95

Strategic Simulations:
Phantasie $22.95
Phantasie II $22.95
Road War 2000 ..... $22.95
Colonial Conquest $22.95

Sublogic:
Flight Simulator II $31.49
Scenery Disk $14.95

ATARI

Access:
Triple Pack $11.95
Leader Board Pack $14.95

Activision:
Hitch Hikers $13.95
Music Studio $19.95

Broderbund:
Print Shop $25.49
Print Shop Compan. $22.95
Graphic Lib. I, II, III $13.49
Bank St. Writer $27.95

Electronic Arts:
Pinball Con Set $8.95
Lords of Conquest $8.95
Starfleet I $32.95
Chess Master 2000 ..... $25.95
Music Con Set $8.95
Super Boulderdash $8.95
One on One $8.95

Firebird:
The Pawn $22.95

Microleague:
Microleag. Baseball $22.95
General Manager $16.95
Stat Disk $13.95

Microprose:
Conflict in Vietnam $22.95
F-15 Strike Eagle $19.95
Kennedy Approach $13.95
Silent Service $19.95
Top Gunner $13.95

Strategic Simulations:
Battle of Antetiem $28.95
Battlecruiser $33.95
Nam $22.95

Joysticks

Tac 3 ..... $9.95
Tac 2 ..... $10.95
Tac 5 ..... $12.95
Tac 1 + IBM/AP $26.95
Economy $5.95
Slik Stick $6.95
Black Max $10.95
Boss $11.99
3-Way $19.99
Bathandle $16.75
Winner 909 ..... $24.95
Wico IBM/AP $29.95
Lipstick Plus $14.95

Monitors

THOMSON 4120 Monitor

- 14" RGBI/video composite/analog
- Compatible with IBM and Commodore
- RGB data cable included

$199.95

Modems

Avatex:
1200e $69.95
1200i PC Card $69.95
1200hc Modem $89.95
2400 ..... $179.95
2400i PC Card $169.95

Hayes:
Smartmodem 300 ..... $149.95
Smartmodem 1200 ..... $285.95
Smartmodem 2400 ..... $425.95

Smarteam
1200 Baud Modem
$86.95
(Hayes Compatible)

Disc Storage

QVS-10 5¼ $3.99
QVS-75 5¼ $16.95
QVS-40 3½ $11.95

Generic Ribbons

Save up to 50% off brand prices! Call for your make and model!

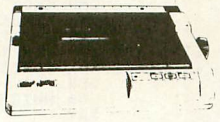
PRICE GUARANTEE

Since 1981, we have led the industry by continuing to offer the lowest national prices while providing quality service. Many companies have come and gone trying to imitate our quality and service. If by some oversight we do not have the lowest prices advertised on the products you desire, then we would appreciate the opportunity to rectify this oversight.

star NX-1000

- 144 CPS Draft
- 36 NLQ

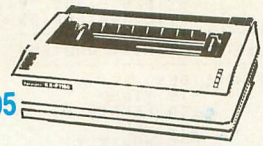
$165.95



Panasonic 1091i *Model II*

- 192 CPS Draft
- NLQ Mode

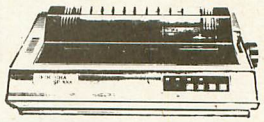
$185.95



SEIKOSHA SP 180Ai

- 100 cps draft
- 20 cps NLQ

$129.95



PRINTERS

star MICRONICS INC

NL-10 w/o Cart	$169.95
NX-1000	$165.95
NX-1000 Color	$225.95
NX-1000C Color	$229.95
NX-15	$295.95
ND-10	$265.95
ND-15	$379.95
NR-10	$309
NR-15	$425.95
NB-15 24 Pin	$699.95
NB24-10 24 Pin	$425.95
NB24-15 24 Pin	$559.95

Panasonic

1080i Model II	$165.95
1091i Model II	$185.95
1092i	$295.95
1592	$399.95
1595	$449.95
3131	$269.95
3151	$419.95
KXP 4450 Laser	$CALL
1524 24 Pin	$559.95
Fax Partner	$589.95

EPSON

LX800	$175.95
FX86E	$289.95
FX286E	$429.95
EX800	$374.95
EX1000	$499.95
LQ500	$CALL
LQ800	$369.95
LQ1000	$549.95
LQ2500	$849.95
GQ3500	$LOW
LQ850	$499.95
LQ1050	$679.95

SEIKOSHA

SP 180Ai	$129.95
SP 1200Ai	$165.95
SP 1200AS RS232	$165.95
SL 80Ai	$299.95
MP1300Ai	$269.95
MP5300Ai	$375.95
MP5420Ai	$879.95
SP Series Ribbon	$7.95
SK3000 Ai	$349.95
SK3005 Ai	$429.95
800 CPS	$CALL!

BROTHER

M1109	$195
M1409	$299
M1509	$365
M1709	$475
Twinwriter 6 Dot & Daisy	$899
M1724L	$599
HR20	$339
HR40	$569
HR60	$709.95

OKIDATA

Okimate 20	$119
Okimate 20 w/cart	$179.95
120	$189.95
180	$219.95
182 +	$225.95
192+	$309.95
193+	$449.95
292 w/interface	$449.95
293 w/interface	$585.95
294 w/interface	$819.95
393	$955.95

CITIZEN

120 D	$149.95
180 D	$169.95
MSP-10	$259.95
MSP-40	$289.95
MSP-15	$324.95
MSP-50	$389.95
MSP-45	$425.95
MSP-55	$489.95
Premiere 35	$464.95
Tribute 224	$624.95

Toshiba

321SL	$489
341 SL	$659
P351 Model II	$899

DIABLO

D25	$499.95
635	$779.95

Call About ATARI PC Availability!

ATARI 520 ST-FM Color System

$799.95

Internal drive included

ATARI 520 ST-FM Monochrome System

$499.95

Internal drive included

ATARI HARDWARE

520 ST FM Mono	$499.95
520 ST FM Color	$799.95
1040 ST Mono	$LOW
1040 ST Color	$LOW
130XE Computer	$135.95
SX551 Drive	$CALL
SF 314 Disk Drive	$199.95
Indus GT Atari Drive	$175.95
SHD 204 20 MEG Drive	$559.95
XM301 Modem	$42.95
SX212 Modem	$89.95
GTS 100 (3.5" DSDD ST)	$195.95

NOTE:

Please Order Early - Further Price Increases By Atari Possible

ATARI 1040 ST Color System

$975.95

ATARI 1040 Monochrome System

$825.95

Join the thousands who shop Lyco and Save!

```
N3A     LDA #$07
N4      STA TEMP
        JSR PRINT
        JMP LINE139
LEFTWALL STY YTEMP
        LDA MULTI,Y
        TAX
        DEX
        LDA FLAG
        CMP #$03
        BEQ L3
        CMP #$05
        BEQ L3
        CMP #$02
        BNE L1
L3      LDA #$0C
        SEC
        SBC YTEMP
        STA COLTMP
        LDY #$00
        JMP DOWALL
L1      LDA #$02
        CLC
        ADC YTEMP
        STA COLTMP
        LDY #$00
DOWALL  INX
        INY
        TYA
        CLC
        ADC #$04
        STA ROWCRS
        LDA COLTMP
        STA COLCRS
        LDA FLAG
        CMP #$00
        BEQ D3
        CMP #$02
        BEQ D1
        CMP #$03
        BEQ D4
        CMP #$01
        BEQ D4
        LDA #$00
        JMP D2
D4      LDA PASS,X
        JMP D2
D1      LDA RWAL,X
        JMP D2
D3      LDA LWAL,X
D2      STX XTMP
        STY YTMP
        STA TEMP
        JSR PRINT
        LDX XTMP
        LDY YTMP
        CPY #$0B
        BNE DOWALL
        LDY YTEMP
        LDA #$00
        STA FLAG
        RTS
;-----------------------------
;Character bytes for left wall
;-----------------------------
LWAL    .BYTE $06,$01,$01,$01,$01
        .BYTE $01,$01,$01,$01,$01,$03
        .BYTE $00,$06,$01,$01,$01,$01
        .BYTE $01,$01,$01,$03,$00
        .BYTE $00,$00,$06,$01,$01,$01
        .BYTE $01,$01,$03,$00,$00
        .BYTE $00,$00,$00,$06,$01,$01
        .BYTE $01,$03,$00,$00,$00
        .BYTE $00,$00,$00,$00,$06,$01
        .BYTE $03,$00,$00,$00,$00
;-----------------------------
;Character bytes for right wall
;-----------------------------
RWAL    .BYTE $04,$01,$01,$01,$01
        .BYTE $01,$01,$01,$01,$01,$02
        .BYTE $00,$04,$01,$01,$01,$01
        .BYTE $01,$01,$01,$02,$00
        .BYTE $00,$00,$04,$01,$01,$01
        .BYTE $01,$01,$02,$00,$00
        .BYTE $00,$00,$00,$04,$01,$01
        .BYTE $01,$02,$00,$00,$00
        .BYTE $00,$00,$00,$00,$04,$01
        .BYTE $02,$00,$00,$00,$00
;-----------------------------
;Character bytes for passageways
;-----------------------------
PASS    .BYTE $00,$05,$05,$05,$05
        .BYTE $05,$05,$05,$05,$05,$00
        .BYTE $00,$00,$05,$05,$05,$05
        .BYTE $05,$05,$05,$00,$00
        .BYTE $00,$00,$00,$05,$05,$05
        .BYTE $05,$05,$00,$00,$00
        .BYTE $00,$00,$00,$00,$05,$05
        .BYTE $05,$00,$00,$00,$00
        .BYTE $00,$00,$00,$00,$00,$05
        .BYTE $00,$00,$00,$00,$00
LENGTH  .BYTE $09,$07,$05,$03,$01
HORIZ   .BYTE $03,$04,$05,$06,$07
;-----------------
;Day/Night Colors
;-----------------
COLOR5  .BYTE $00,$9F,$9D,$9B,$99
        .BYTE $97,$95,$93,$91,$F0,$F2
        .BYTE $F4,$F6,$F8,$FA,$FC
        .BYTE $00,$00
COLOR52 .BYTE $00,$00,$00,$00,$00
        .BYTE $00,$00,$00,$00,$00
        .BYTE $00,$02,$04,$06,$08
        .BYTE $0A,$00,$00
;-----------------------
;Print Front Wall Routine
;-----------------------
FRONTWALL DEY
        STY YTEMP
        LDA LENGTH,Y
        STA LEN
        LDA HORIZ,Y
        STA COLCRS
        LDA MULTI,Y
        TAX
        DEX
        STA YTEMP
        LDX YTEMP
        LDY #$00
F1      TYA
        STA ROWCRS
        INC ROWCRS
        INC ROWCRS
        INY
        INX
        STY YTMP
        STX XTMP
        LDA FLAG
        CMP #$C5
        BNE F1A
        LDA #$00
        JMP F1B
F1A     LDA PASS,X
F1B     STA TEMP
        LDY ROWCRS
        JSR PRINT
        LDY YTMP
        LDX XTMP
        CPY #$0A
        BNE F1
        DEC LEN
        LDA LEN
        BEQ LINE139
        INC COLCRS
        LDX YTEMP
        LDY #$00
        JMP F1
;--------------------
;Main Display Routine
;--------------------
LINE139 JSR MOVE    ;Print Screen!
        LDA CENTER+1 ;Copy center.
        STA FRONT2
        LDA BACK    ;Copy back.
        STA BACK2
        LDA TIME
        CMP #$01    ;Is it night?
        BEQ CE      ;Yes...
        JSR POSITION ;No..Draw maze.
        LDX #$00
PL1CL1  LDA #0      ;Clear PMAREA
        STA PL1,X   ;[Player Maze
        INX         ;Position]
        BNE PL1CL1
        LDA H       ;Add offset for
        CLC         ;Gr. $E to
        ADC #$91    ;player horiz
        STA HPOSP1  ;position!
        CLC
        LDA V       ;Add offset to
        CLC         ;vert position
        ADC #$4A    ;to plot on a
        TAY         ;Gr. $E screen.
        LDA #$01
        STA PL1,Y   ;And store!
CE      LDX #$01
CE1     LDA DIRCHAR,X ;Let's clear
        SEC         ;the compass
        SBC #$80    ;arrows.
        LDY OFFSETT,X
        STA TIMSC,Y
        INX
        CPX #$05
        BNE CE1
CE2     LDY DIRECTION ;Inverse the
        LDA DIRCHAR,Y ;compass arrow
        LDX OFFSETT,Y ;in the dirctn
        STA TIMSC,X ;player's facing
        LDA #$3E
        STA SDMCTL  ;Turn on screen.
;-------------------------
;Joystick/Movement Routine
;-------------------------
LINE140 JSR MVCHK   ;Patrol Move.
        LDX H       ;Get Player's
        LDY V       ;coordinates.
        JSR START
        JSR COPCEN
        LDA STICK0
        CMP #$0F
        BEQ LINE140
        LDA STICK0
        CMP #$0E
        BNE LINE190
        LDA FRONT2
        CMP #$20
        BEQ LINE1401
        CMP #$45
        BEQ LINE1401
        CMP #$C5
        BNE LINE140A
        JMP ESCAPE
LINE140A JMP LINE1901
LINE1401 LDX DIRECTION
        DEX
        LDA WHICH,X
        TAY
        CLC
        LDA H,Y
        ADC TABLE,X
        STA H,Y
        JMP LINE275
LINE190 CMP #$0D
        BNE LINE230
        LDA BACK2
        CMP #$20
        BEQ LINE190A
        CMP #$45
        BEQ LINE190A
        CMP #$C5
        BNE LINE1901
        JMP ESCAPE
LINE1901 JSR THUMP
        JMP LINE140
LINE190A LDX DIRECTION
        DEX
        LDA WHICH,X
        TAY
        CLC
        LDA H,Y
        ADC TABLE2,X
        STA H,Y
        JMP LINE275
LINE230 CMP #$0B
        BNE LINE250
        JSR RAZZ
        INC DIRECTION
        LDA DIRECTION
        CMP #$05
        BNE LINE275
        LDA #$01
        STA DIRECTION
        JMP LINE275
LINE250 CMP #$07
        BEQ LINE260
        JMP LINE140
LINE260 JSR RAZZ
LINE265A DEC DIRECTION
        LDA DIRECTION
        CMP #$00
        BNE LINE275
        LDA #$04
        STA DIRECTION
        JMP LINE275
LINE275 JSR PAUSE
        JMP MAINPROG
;--------------
;Delay Routine
;--------------
PAUSE   LDA #$00    ;Zero System
        STA RTCLOK  ;Timer.
P1      JSR MVCHK   ;Check for
        LDA RTCLOK  ;Patrol move and
        SEC         ;wait for timer
        SBC #$06    ;goes past 6.
        BCC P1
        RTS         ;Pause finished!
;--------------
;Set up display
;--------------
SETUP   LDA #$0D
        STA COLOR3
        LDA #$00    ;Background's
        STA COLOR2  ;black.
        STA MNUM
        LDA #$04
        STA COLOR0
        LDA #$90
        STA PCOLR0+1
        LDA #$02    ;Set P/M Prior.
        STA GPRIOR
        LDA # <DLIST ;Tell ANTIC
        STA SDLSTL  ;where the D/L
        LDA # >DLIST ;is.
        STA SDLSTL+1
        LDA # >DLI  ;Point system
        STA VDSLST+1 ;to my DLI.
        LDA # <DLI
        STA VDSLST
        LDA #$C0    ;Allow for DLIs.
        STA NMIEN
        LDA #$03    ;Set players
        STA SIZEP2  ;2 & 3 to quad
        STA SIZEP3  ;size.
        LDA #$68    ;Set Player's
        STA HPOSP0+2 ;2&3 horiz
        LDA #$74    ;position.
        STA HPOSP0+3
```

```
        LDX #0
PL1CL   LDA #$00    ;Clear and init
        STA PMG+$0400,X ;PMAREA.
        STA PMG+$0500,X
        LDA #$FF
        STA PMG+$0600,X
        STA PMG+$0700,X
        INX         ;Copy 255 Bytes.
        BNE PL1CL
        LDA #PMG/256 ;Store P/M
        STA PMBASE  ;Address.
        LDA #$02    ;Enable P/M
        STA GRACTL  ;Graphics
        RTS
;----------------------
;Main Game Display List
;----------------------
DLIST   .BYTE $70,$70,$F0,$46
        .WORD TOP
        .BYTE $0C,$20,$02,$00,$02
        .BYTE $02,$30
        .BYTE $80,$4E
        .WORD DISP
        .BYTE $0E
        .BYTE $0E,$0E,$0E,$0E,$0E,$0E
        .BYTE $0E,$0E,$8E
        .BYTE $0E,$0E,$0E,$0E,$0E,$0E
        .BYTE $0E,$0E,$8E
        .BYTE $0E,$0E,$0E,$0E,$0E,$0E
        .BYTE $0E,$8E,$0E
        .BYTE $0E,$0E,$0E,$0E,$8E,$0E
        .BYTE $0E,$0E,$8E
        .BYTE $0E,$0E,$8E,$0E,$8E,$0E
        .BYTE $8E,$8E,$0E
        .BYTE $0E,$8E,$0E,$8E,$0E,$0E
        .BYTE $0E,$8E,$0E
        .BYTE $0E,$0E,$0E,$0E,$8E,$0E
        .BYTE $0E,$0E,$0E
        .BYTE $0E,$0E,$0E,$8E,$0E,$0E
        .BYTE $0E,$0E,$0E
        .BYTE $0E,$0E,$0E,$8E,$0E,$0E
        .BYTE $0E,$0E,$0E
        .BYTE $0E,$0E,$0E,$0E,$0E,$8E
        .BYTE $70,$42
TEXTB   .WORD $00
        .BYTE $02,$02,$02,$02,$02
        .BYTE $02,$41
        .WORD DLIST
;-------------------------
;Check for patroller move
;-------------------------
MVCHK   JSR $6700   ;Check for move.
        LDX #$FF
        LDA DIFLEV  ;Difficulty=0?
        BEQ MV25    ;Yes...
MV1     INX
        LDA H2,X    ;Check if there
        CMP H       ;are any patrols
        BNE MV2     ;occuping the
        LDA V2,X    ;same place
        CMP V       ;where the
        BEQ MV3     ;player is.
MV2     CPX DIFLEV  ;More patrols?
        BNE MV1     ;YA...
MV25    RTS
MV3     LDA # <CAPTXT ;Print Player
        STA TEXTB   ;capture text on
        LDA # >CAPTXT ;screen bottom
        STA TEXTB+1
        LDA #$45    ;Put patroler in
        STA CENTER2 ;front of player
        LDA #$00    ;Zero all audio.
        STA AUDF1
        JMP EX1     ;End routine
;-------------------------------
;Routine to Print to Graphics $E
;Adapted from Retrofire!
;-------------------------------
PRINT   LDA YPOS,Y
        STA LO      ;Save Y Position
        LDA #$00
        STA LO+1
        ASL LO      ;*16
        ROL LO+1
        ASL LO      ;*32
        ROL LO+1
        LDA LO
        CLC
        ADC # <SCREEN ;Add Display
        STA DESTLO  ;Start
        LDA LO+1
        ADC # >SCREEN
        STA DESTLO+1
        ASL LO      ;*16
        ROL LO+1
        ASL LO      ;*32
        ROL LO+1
        LDA LO
        CLC
        ADC DESTLO  ;Get character's
        STA DESTLO  ;screen address
        LDA LO+1
        ADC DESTLO+1
        STA DESTLO+1
        LDA DESTLO  ;Add X offset
        CLC         ;for final
        ADC COLCRS  ;address.
        ADC #$0C
        STA DESTLO
        BCC NODHIN
        INC DESTLO+1
NODHIN  LDY TEMP    ;Get offset into
        LDA MULT8,Y ;character set.
        STA LO
        LDA #$00
        STA LO+1
        STA COPCNT  ;Zero copy count
        CLC
        ADC # >CHARSET ;get info
        STA LO+1    ;from char set.
        LDA LO
        CLC
        ADC # <CHARSET
        STA LO
        BCC COPNUM
        INC LO+1
COPNUM  LDY COPCNT
        LDA (LO),Y  ;Get character
        LDY #$00    ;image byte.
        STA (DESTLO),Y ;To screen...
        INC COPCNT  ;Next char byte.
        LDA COPCNT  ;Are all 8 bytes
        CMP #$08    ;in char moved?
        BEQ FINISH  ;Ya!
        LDA DESTLO  ;Add 20 bytes to
        CLC         ;point to next
        ADC #20     ;line on screen.
        STA DESTLO
        BCC COPNUM
        INC DESTLO+1
        BNE COPNUM
FINISH  RTS         ;All done!
YPOS    .BYTE $01,$05,$0D,$15,$1D
        .BYTE $25,$2D,$35,$3D,$45
        .BYTE $4D,$55
;----------------------
;Character Set for Maze
;----------------------
CHARSET
        .BYTE $00,$00,$00,$00,$00,$00
        .BYTE $00,$00
        .BYTE $FF,$BF,$FF,$F7,$FF,$FF
        .BYTE $DF,$FF
        .BYTE $FF,$7F,$3F,$1F,$0F,$07
        .BYTE $03,$01
        .BYTE $FF,$FE,$FE,$F8,$F0,$E0
        .BYTE $C0,$80
        .BYTE $01,$03,$07,$0F,$1F,$3F
        .BYTE $7F,$FF
        .BYTE $57,$FF,$D5,$D5,$D5,$FF
        .BYTE $57,$57
        .BYTE $80,$C0,$E0,$F0,$F8,$FC
        .BYTE $FE,$FF
        .BYTE $00,129,227,$FF,$FF,$C3
        .BYTE $81,$00
MULTI   .BYTE $00,$0B,$16,$21,$2C
        .BYTE $37,$42
MULT8   .BYTE $00,$08,$10,$18,$20
        .BYTE $28,$30,$38,$40
;--------------
;Sound Routines
;--------------
THUMP   LDA #$46    ;Pitch for THUMP
        STA AUDF1
        LDA #$0F    ;THUMP volume.
        STA VOLUM1
        LDA #$00    ;THMP distortion
        STA TONE
        LDA STICK0  ;Load joystick
        STA OLDSTK  ;and save.
TH1     JSR MVCHK   ;Patrol move...
        LDA OLDSTK  ;Stick changed
        CMP STICK0  ;direction yet?
        BEQ TH1     ;No, check again
        RTS
RAZZ    LDA #$00    ;Pitch for TURN
        STA AUDF1
        LDA #$08    ;Volume for TURN
        STA VOLUM1
        LDA #$20    ;TURN distortion
        STA TONE
        RTS
;-------------------------------
;Move newly drawn maze to screen
;-------------------------------
MOVE    LDA # <SCREEN ;MOVE source
        STA LO
        LDA # >SCREEN
        STA LO+1
        LDA # <DISP ;MOVE dstination
        STA DESTLO
        LDA # >DISP
        STA DESTLO+1
        LDA LO
        CLC
        ADC #$50
        STA LO
        BCC MB
        INC LO+1
MB      LDY #$0E
        LDA #$59
        STA TEMP
```

```
        LDX #$20
M0      LDA (LO),Y  ;Get FROM byte
        STA (DESTLO),Y ;and MOVE it!
        INY
        CPY #$19    ;Line finished?
        BNE M0      ;No!
        LDY #$0E
        LDA LO      ;Point SOURCE to
        CLC         ;next line.
        ADC #$14
        STA LO
        BCC M1
        INC LO+1
M1      LDA DESTLO  ;Point DEST. to
        CLC         ;next line.
        ADC #$28
        STA DESTLO
        BCC M2
        INC DESTLO+1
M2      DEC TEMP
        LDA TEMP    ;Any lines left?
        CMP #$00    ;
        BNE M0      ;Yes,do it again
        RTS         ;Nope, done!
;-------------------------
;Clear Screen/Memory Routine
;-------------------------
CLSCN   LDA #$89    ;Start clearing
        STA LO+1    ;at $8900
        LDA #$00
        STA LO
LX2     LDY #$00
        LDA #$00    ;Erase BYTE
LX1     STA (LO),Y
        INY
        CPY #$00
        BNE LX1
        INC LO+1
        LDA LO+1
        CMP #$A0    ;At $A000 yet?
        BNE LX2     ;No,more to copy
        RTS         ;Done!
POSITION LDA H      ;Get X & Y cords
        STA PLOTX   ;and store them.
        LDA V
        STA PLOTY
        LDA #$02    ;Dark Grey color
        STA COLOR
        LDX #$00
        LDA #$10    ;Store screen
        STA MTEMP   ;address of maze
        LDA #$90    ;in MTEMP.
        STA MTEMP+1
        LDY DIRECTION
P01     STX XTMP
        LDA DIRECTION
        CMP #$01    ;The rest of
        BEQ PLEFT   ;this routine
        CMP #$04    ;draws the maze
        BEQ PLEFT   ;on the right
        TXA         ;of the screen.
        CLC
        ADC #$0A
        TAX
PLEFT   LDA LEFT,X
        CMP #$A0
        BNE P01A
        LDA DRAW2X,Y
        TAX
        DEC PLOTX,X
        JSR PLOTCL
        INC PLOTX,X
P01A    LDX XTMP
        LDA DIRECTION
        CMP #$02
        BEQ PRIGHT
        CMP #$03
        BEQ PRIGHT
        TXA
        CLC
        ADC #$0A
        TAX
PRIGHT  LDA LEFT,X
        CMP #$A0
        BNE P01B
        LDA DRAW2X,Y
        TAX
        INC PLOTX,X
        JSR PLOTCL
        DEC PLOTX,X
P01B    LDX XTMP
        LDA CENTER,X
        CMP #$A0
        BEQ P02
        CLC
        LDA DRAWX,Y
        TAX
        LDA PLOTX,X
        ADC DRAWY,Y
        STA PLOTX,X
        LDX XTMP
        INX
        CPX #$06
        BNE P01
        RTS
P02     JSR PLOTCL
        RTS
;-----------------------
;Plot Calculator
;Adapted from Retrofire!
;-----------------------
PLOTCL  PHA         ;Save accum.
        TXA         ;Save X register
        PHA
        TYA         ;Save Y Register
        PHA
        LDA PLOTY
        ASL A
        STA LO
        LDA #$00
        ROL A
        STA LO+1    ;*2
        ASL LO
        ROL LO+1    ;*4
        ASL LO
        LDA LO
        STA DESTLO
        ROL LO+1
        LDA LO+1
        STA DESTLO+1 ;*8
        ASL LO
        ROL LO+1    ;*16
        ASL LO
        ROL LO+1    ;*32
        LDA LO
        CLC
        ADC DESTLO
        STA LO
        LDA LO+1
        ADC DESTLO+1
        STA LO+1
        LDA MTEMP   ;Add the display
        CLC         ;address to get
        ADC #$0A    ;the actual
        CLC         ;address of the
        ADC LO      ;byte that will
        STA LO      ;be altered for
        LDA MTEMP+1 ;the plot.
        ADC LO+1
        STA LO+1
        LDA PLOTX   ;Mask PLOTX for
        AND #$03    ;plot index.
        TAX
        LDA PLOTX   ;Get PLOTX and
        LSR A       ;divide
        LSR A       ;by 4.
        CLC         ;Add to plot
        ADC LO      ;address for
        STA LO      ;final plot
        LDA LO+1    ;address.
        ADC #$00
        STA LO+1
        LDY COLOR   ;Get color
        LDA BMASK2,X ;and mask off
        AND COLRS,Y ;pixel position.
        STA HOLD    ;Save it.
        LDA BMASK1,X ;Mask off pixel
        LDY #$00    ;of the address
        AND (LO),Y  ;to be altered.
        ORA HOLD    ;Set the plot
        STA (LO),Y  ;bits and store!
        PLA
        TAY         ;Restore Y Reg
        PLA
        TAX         ;Restore X Reg
        PLA         ;Restore accum
        RTS         ;Finished!
COLRS   .BYTE $00,$55,$AA,$FF
BMASK1  .BYTE $3F,$CF,$F3,$FC
BMASK2  .BYTE $C0,$30,$0C,$03
;---------------
;Screen Displays
;---------------
BOTTOM
        .SBYTE "  ┌──────────┐    "
        .SBYTE "      Compass     "
        .SBYTE "  |Time Elapsed|  "
        .SBYTE "  └──────────┘    "
        .SBYTE "  | 0:00:00.00 |  "
        .SBYTE "       North      "
        .SBYTE "  ┌──────────┐    "
        .SBYTE "            ↑     "
        .SBYTE "                  "
        .SBYTE "     West ← → East"
        .SBYTE "                  "
        .SBYTE "            ↓     "
        .SBYTE "                  "
        .SBYTE "        South     "
ESCTXT
        .SBYTE "      Congratulation"
        .SBYTE "s Pathfinder!!      "
        .SBYTE "                    "
        .SBYTE "                    "
        .SBYTE "    )You have escape"
        .SBYTE "d the Labyrinth(    "
        .SBYTE "                    "
        .SBYTE "                    "
        .SBYTE "         Your time wa"
        .SBYTE "s 0:00:00.00        "
        .SBYTE "                    "
        .SBYTE "                    "
        .SBYTE "       Press Trigge"
        .SBYTE "r to Restart         "
CAPTXT
        .SBYTE "          You have be"
        .SBYTE "en captured           "
        .SBYTE "                      "
        .SBYTE "                      "
        .SBYTE "     You will be dis"
        .SBYTE "posed of at the      "
        .SBYTE "       next time conv"
        .SBYTE "ienent for you        "
        .SBYTE "         Your time wa"
        .SBYTE "s 0:00:00.00          "
        .SBYTE "                      "
        .SBYTE "                      "
        .SBYTE "         Press Trigge"
        .SBYTE "r to Restart         "
DIRCHAR .BYTE $00,$DC,$DE,$DD,$DF
OFFSETT .BYTE $00,$45,$6C,$95,$6E
;---------------------
;Random Maze Generator
;---------------------
GENMZE  LDA MTEMP   ;This routine,
        STA LO      ;taken from a
        LDA MTEMP+1 ;BASIC routine,
        STA LO+1    ;generates a
        LDX #$00    ;maze with only
L0      LDY #$00    ;one correct
        LDA #$A0    ;path to the
MLOOP   STA (LO),Y  ;exit.
        INY
        CPY #$31
        BNE MLOOP
        LDA #$00
        STA (LO),Y
        CLC
        LDA LO
        ADC #$32
        STA LO
        BCC L2
        INC LO+1
L2      INX
        CPX #$23
        BNE L0
LINE20  LDA #$00
        STA DESTLO
LINE30  LDA MTEMP
        STA LO
        LDA MTEMP+1
        STA LO+1
        CLC
        LDA LO
        ADC #$35    ; Start gen here
        STA LO
        BCC LINE50
        INC LO+1
LINE50  LDY #$00
        LDA #$05
        STA (LO),Y
LINE60  LDA RANDOM
        AND #$07
        STA J
        LDA J
        SEC
        SBC #$04
        BCS LINE60
LINE70  LDA J
        STA TEMP
LINE80  LDA LO
        STA DESTLO
        LDA LO+1
        STA DESTLO+1
        LDA J
        CMP #$00
        BNE LINE801
        CLC
        LDA DESTLO
        ADC #$02
        STA DESTLO
        BCC LINE90
        INC DESTLO+1
        JMP LINE90
LINE801 LDA J
        CMP #$01
        BNE LINE802
        SEC
        LDA DESTLO
        SBC #$64
        STA DESTLO
        BCS LINE90
        DEC DESTLO+1
        JMP LINE90
LINE802 LDA J
        CMP #$02
        BNE LINE803
        SEC
        LDA DESTLO
        SBC #$02
        STA DESTLO
        BCS LINE90
        DEC DESTLO+1
        JMP LINE90
LINE803 LDA J
        CMP #$03
        BNE LINE90
        CLC
        LDA DESTLO
```

```
        ADC #$64
        STA DESTLO
        BCC LINE90
        INC DESTLO+1
LINE90  LDY #$00
        LDA (DESTLO),Y
        CMP #$A0
        BNE LINE100
        LDA J
        CLC
        ADC #$01
        LDY #$00
        CMP #$00
        BNE LINE90A
        LDA #$20
LINE90A STA (DESTLO),Y
        JSR LINE200
        LDY #$00
        LDA #$20
        STA (LO),Y
        LDA DESTLO
        STA LO
        LDA DESTLO+1
        STA LO+1
        JMP LINE60
LINE100 LDA J
        CMP #$03
        BNE LINE101A
        LDA #$00
        STA J
        JMP LINE110
LINE101A INC J
LINE110 LDA J
        CMP TEMP
        BEQ LINE110A
        JMP LINE80
LINE110A LDY #$00
LINE120 LDA (LO),Y
        STA J
LINE130 LDA #$20
        STA (LO),Y
        LDA J
        SEC
        SBC #$05
        BCS LINE210
        DEC J
        LDX J
        LDA NEWJ,X
        STA J
        JSR LINE200
        JSR LINE200
        JMP LINE60
LINE200 LDA J
        CMP #$00
        BNE LINE2001
        CLC
        LDA LO
        ADC #$01
        STA LO
        BCC LINE210
        INC LO+1
        JMP LINE210
LINE2001 LDA J
        CMP #$01
        BNE LINE2002
        SEC
        LDA LO
        SBC #$32
        STA LO
        BCS LINE210
        DEC LO+1
        JMP LINE210
LINE2002 LDA J
        CMP #$02
        BNE LINE2003
        SEC
        LDA LO
        SBC #$01
        STA LO
        BCS LINE210
        DEC LO+1
        JMP LINE210
LINE2003 LDA J
        CMP #$03
        BNE LINE210
        CLC
        LDA LO
        ADC #$32
        STA LO
        BCC LINE210
        INC LO+1
LINE210 RTS
COPCEN  LDX #$00
COP2    LDA CENTER,X
        STA CENTER2,X
        INX
        CPX #$07
        BNE COP2
        RTS
;-------------------------
;Miscellaneous Data Tables
;-------------------------
NEWJ    .BYTE $02,$03,$00,$01
TABLE   .BYTE $FF,$FF,$01,$01
TABLE2  .BYTE $01,$01,$FF,$FF
WHICH   .BYTE $01,$00,$01,$00
DRAWY   .BYTE $00,$FF,$FF,$01,$01
DRAWX   .BYTE $00,$01,$00,$01,$00
DRAW2X  .BYTE $00,$00,$01,$00,$01
YVAL    .BYTE $08,$07,$05,$04,$02
COMP    .BYTE $0A,$06,$0A,$06,$0A
HARGO   .BYTE $6E,$6E,$6E,$6E
        .BYTE $7A,$00
SARGO   .BYTE $03,$03,$03,$03,$00
BARGO   .BYTE $FF,$7E,$3C,$18,$18
ONARG   .BYTE $54,$5C,$64,$6C,$74
OFARG   .BYTE $9A,$92,$8A,$82,$7A
;-------------------------
;Escape from Maze/Death Routine
;-------------------------
ESCAPE  LDA # <ESCTXT ;Print
        STA TEXTB     ;"Escape from
        LDA # >ESCTXT ;Maze" text at
        STA TEXTB+1 ;screen bottom.
        STA DESTLO+1
EX1     LDA TEXTB+1
        STA DESTLO+1
        LDA TEXTB
        CLC           ;Point DESTLO to
        ADC #$B4      ;proper place on
        STA DESTLO    ;screen to print
        BCC EX2       ;final time.
        INC DESTLO+1
EX2     LDX #$00
        LDA CENTER2 ;Check if player
        CMP #$45      ;is dead.
        BEQ EX3       ;yep...
EX21    LDA #$FF      ;Turn of timer.
        STA FLAG
        LDA #$03      ;Quadruple Player
        STA SIZEP0    ;Width for
        STA SIZEP1    ;Players 0 & 1.
        JSR PL1CL
        LDX #$A6
        LDA #$FF      ;Display Players
EX25    INX           ;on bottom 1/3rd
        STA PMG+$0400,X ;of screen.
        STA PMG+$0500,X
        CPX #$E5
        BNE EX25
        LDA #$20      ;Display players
        STA HPOSP0    ;in left and
        LDA #$C0      ;right margin of
        STA HPOSP0+1 ;the screen.
        LDX #$00
EX3     LDA SEC1,X  ;Print elapsed
        CLC           ;time, one
        ADC #$10      ;character at a
        LDY YVAL,X  ;time in proper
        STA (DESTLO),Y ;space at the
        INX           ;bottom of the
        CPX #$05      ;screen.
        BNE EX3
ES1     LDA CENTER2 ;Are Patroller &
        CMP #$45      ;player together?
        BEQ ES2       ;YES...Captured!
        LDA RANDOM  ;Not Captured...
        STA COLPM0  ;Change PM Colors
        STA COLPM1  ;for victory
        INC HOLD      ;display.
ES2     LDA STRIG0  ;Trigger Pressed?
        CMP #$01
        BEQ ES1       ;Nope...
        JMP $6900     ;Yes...Restart!
TOP
        .SBYTE "     labyrinths     "
        .SBYTE "                    "
        .SBYTE "    The Ultimate 3-D"
        .SBYTE " Maze  Adventure    "
        .SBYTE "     (C) 1987  Argos"
        .SBYTE "oft Productions     "
        .SBYTE "                    "
        .SBYTE "                    "
MAZE    .DS $0400
        *=  $6700
;-------------------------
;Patroller Movement Routine
;-------------------------
        .LOCAL
?BEGIN  LDA CH        ;[ESC] pressed?
        CMP #$1C
        BNE BEG1      ;Nope...
        JMP $6900     ;Yes...Restart!
BEG1    LDA CDTMV4  ;Timer counted
        CMP #$00      ;down to 0 yet?
        BEQ B15       ;Yes...
        JMP RET2      ;Nope...
B15     LDA #$10      ;Reset Timer.
        STA CDTMV4
        LDX MNUM
        LDA #$00      ;Erase patroller
        STA COLOR     ;position on maze
        JSR EDOT      ;drawing.
B2      LDA DIRECTION ;Save player
        STA DIRTMP    ;direction.
        LDX MNUM      ;Which patroller?
        LDA DIR2,X  ;Get its dirction
        STA DIRECTION ;and store it!
        LDA V2,X      ;Get "Y" Coord.
        TAY           ;and put it in Y.
        LDA H2,X      ;Get "X" Coord.
        TAX           ;and put it in X.
        LDA #$20      ;Put "Space" in
        STA PAT       ;patroller pos.
        JSR PUTPT     ;in memory maze.
        LDX MNUM      ;Which patroller?
        LDA V2,X      ;Transfer the
        TAY           ;"X" and "Y"
        LDA H2,X      ;coords again.
        TAX
        JSR START     ;Get local walls.
        LDX MNUM      ;Which patroller?
        LDA DIRTMP    ;Transfer temp.
        STA DIRECTION ;back to orig.
        LDA CENTER+1 ;Space in front
        CMP #$20      ;of patroller?
        BNE TURN      ;Nope. Turn
        JSR ?LINE1401 ;Ya. Staight.
RET     LDX MNUM      ;Get patroller #.
        LDA #$03      ;White color for
        STA COLOR     ;plot.
        INC MNUM      ;Point next one.
        LDY MNUM      ;Check to see if
        DEY           ;all patrollers
        CPY DIFLEV    ;have moved.
        BEQ B15X      ;Yep...
        JSR EDOT      ;Nope. Replot.
        JMP B15       ;Next move...
B15X    LDA #$00      ;Reset patroller
        STA MNUM      ;number.
RET2    LDA #$45      ;Replot all
        STA PAT       ;patrollers
        LDX MNUM      ;in memory maze.
        CPX DIFLEV    ;All ploted?
        BEQ RET3      ;Yep...
        LDA V2,X      ;Get "X" coord
        TAY           ;and
        LDA H2,X      ;"Y" coordinate
        TAX           ;and
        JSR PUTPT     ;print in maze/
        INC MNUM
        CMP DIFLEV    ;All ploted?
        BNE RET2      ;Nope...
RET3    LDA #$00      ;Reset Patroller#
        STA MNUM
        RTS           ;Back to program!
TURN    LDA RANDOM  ;Get a random #.
        CLC
        ADC #$80      ;If number>128,
        BCC PTR       ;turn right.
PTL     LDA LEFT    ;Is there a wall
        CMP #$A0      ;to the left?
        BEQ PTL1      ;Yep. Right turn.
        JMP TLEFT     ;Nope. Left turn.
PTL1    LDA RIGHT   ;Is there a wall
        CMP #$A0      ;to the right?
        BEQ RNDTURN ;Ya. Random turn.
        JMP TRIGHT  ;No. Right turn.
PTR     LDA RIGHT   ;Is there a wall
        CMP #$A0      ;to the right?
        BEQ PTR1      ;Yep. Left turn.
        JMP TRIGHT  ;No. Right turn.
PTR1    LDA LEFT    ;Is there a wall
        CMP #$A0      ;to the left?
        BEQ RNDTURN ;Ya. Random turn.
        JMP TLEFT     ;No. Turn left.
RNDTURN LDA RANDOM  ;Get Random #.
        CLC
        ADC #$80      ;If <128 then
        BCC TRIGHT    ;turn right.
TLEFT   LDA DIR2,X  ;Load current
        TAY           ;direction of
        LDA LTDIR,Y ;patroller and
        STA DIR2,X  ;change with
        JMP RET       ;new value.
LTDIR   .BYTE $00,$02,$03,$04,$01
TRIGHT  LDA DIR2,X  ;Load current
        TAY           ;direction of
        LDA RTDIR,Y ;patroller and
        STA DIR2,X  ;change with
        JMP RET       ;new value.
RTDIR   .BYTE $00,$04,$01,$02,$03
EDOT    LDA H2,X      ;Get "X" and "Y"
        STA PLOTX     ;coords from
        LDA V2,X      ;patroller.
        STA PLOTY
        LDA #$10      ;Set MTEMP to
        STA MTEMP     ;point to maze
        LDA #$90      ;on screen
        STA MTEMP+1
        LDA TIME    ;What time is it?
        CMP #$00      ;Daytime?
        BNE ED01      ;Nope...
        JSR PLOTCL  ;Yes, plot point.
ED01    RTS
?LINE1401 LDX MNUM    ;Patroller #.
        LDA DIR2,X  ;Find direction
        TAX           ;its moving.
        DEX
        LDA ?WHICH,X ;Add to Horiz
        CLC           ;or vert.
        ADC MNUM      ;Offset to
        TAY           ;proper
        CLC           ;patroller #.
        LDA H2,Y      ;Add or Subtrct
        ADC ?TABLE,X ;from horiz or
        STA H2,Y      ;vert.
        RTS
```

```
?TABLE    .BYTE $FF,$FF,$01,$01
?WHICH    .BYTE $07,$00,$07,$00
SPOS      .BYTE $14,$30,$30,$02,$30
          .BYTE "▗▄▖"
          .BYTE $14,$22,$02,$14,$14
          .BYTE " ▌▐"
          .BYTE $04,$04,$04,$01,$04
          .BYTE "▟▄▙"
PUTPT     LDA # <MAZE ;Copy maze
          STA MTEMP   ;address to
          LDA # >MAZE ;MTEMP
          STA MTEMP+1
?S0       DEY         ;Decrease "Y"
          CPY #$00    ;until zero to
          BEQ ?S1     ;find vert pos.
          LDA #$32    ;Add maze width.
          JSR ADD
          JMP ?S0
?S1       TXA         ;Add horiz val.
          JSR ADD
          LDY #$00
          LDA PAT     ;Store
          STA (MTEMP),Y ;plot/erase
          RTS         ;here.
;-----------------------------
;Title Page and Option Selection
;-----------------------------
          *=  $6900
          LDA #$40    ;Disable DLIs.
          STA NMIEN
          LDA #$00    ;Turn off Screen
          STA DMACTL
          STA AUDC1   ;and sounds.
          LDX #$00
          STX SIZEP0  ;Zero P/M sizes.
          STX SIZEP1
SUB3      STA CENTER2,X ;Erase
          INX         ;CENTER2.
          CPX #$06
          BNE SUB3
          STX HPOSP0  ;Position PM0 &
          STX HPOSP1  ;PM1 off screen.
          LDX #$00
SUB4      LDA SPOS,X  ;Reset
          STA H2,X    ;Patrollers'
          INX         ;starting
          CPX #$14    ;positions.
          BNE SUB4
          LDA #$FF    ;Turn off timer.
          STA FLAG
          LDA #$00    ;Zero Dummy and
          STA DUMMY   ;pointer
          STA POINTER
          JMP ?DLINIT ;Init.
;--------------
;Title Page DLI
;--------------
?DLI      PHA         ;Save Accumulator
          TXA
          PHA         ;Save X register
          TYA
          PHA         ;Save Y register
          LDA POINTER ;Which DLI?
          CMP #$00
          BNE ?DLOOP2
          LDX #$90
?DLOOP1   STX COLPF0  ;Shade line
          INX         ;in 16 shades
          STA WSYNC   ;of blue
          CPX #$9F
          BNE ?DLOOP1
          JMP ?DRTS
?DLOOP2   CMP #$01    ;2nd DLI?
          BEQ ?DLOOP25
          CMP #$02    ;3rd DLI?
          BNE ?DLOOP5
?DLOOP25  LDX #$00    ;2nd DLI...
?DLOOP3   STX COLPF0  ;Shade
          INX         ;in 8 colors
          INX         ;of grey.
          STA WSYNC
          CPX #$10
          BNE ?DLOOP3
          JMP ?DRTS
?DLOOP5   LDY DUMMY   ;Change colors
          LDA ?COLORS,Y ;on bottom
          STA COLPF2  ;of titles.
          INC DUMMY
          LDA DUMMY   ;All colors
          CMP #$06    ;displayed?
          BNE ?DRTS   ;nope..
          LDA #$00    ;Set color pntr
          STA DUMMY   ;to start color.
?DRTS     INC POINTER
          LDA POINTER ;All DLIs
          CMP #$09    ;done?
          BNE ?DR1    ;Nope...
          LDA #$00    ;Set pointer
          STA POINTER ;to 1st DLI.
?DR1      PLA         ;Restore Y Reg
          TAY
          PLA         ;Restore X Reg
          TAX
          PLA         ;Restore A
          RTI
;-------------------------
;Title Page Initialization
;-------------------------
?DLINIT   LDA #?DLIST&255 ;Set D/L
          STA SDLSTL  ;address to
          LDA #?DLIST/256 ;pointer.
          STA SDLSTL+1
          LDA #?DLI&255 ;Set DLI
          STA VDSLST  ;address
          LDA #?DLI/256 ;to pointer.
          STA VDSLST+1
          LDA #$C0    ;Enable DLIs.
          STA NMIEN
          LDA #$00
          STA COLOR2  ;Black Background
          LDA #$22    ;Turn on Screen
          STA SDMCTL  ;Display.
          LDA TIME    ;Load time and
          JMP E4      ;print it.
PLV       LDY DIFLEV  ;Load dificulty
          LDX #$00    ;level.
          LDA POINT,Y ;Get offset
          TAY         ;pointer.
PLV1      LDA LEVELS,Y ;Get Character
          CMP #$5B    ;My EOL Char?
          BEQ END     ;Ya...
          STA PRINT1,X ;Nope print
          INY         ;character.
          INX         ;Inc POINTERS.
          JMP PLV1    ;Print next.
END       LDA DIFLEV
          CLC         ;Print level #
          ADC #$10    ;in proper spot
          STA PRINT2  ;of screen.
          LDA CONSOL
          CMP #$05    ;SELECT pressd?
          BNE END1
          LDA #$01    ;Yes...Set flag
          STA HOLD
          JMP E3      ;Beep!
END1      CMP #$03    ;OPTION pressd?
          BEQ E3      ;Beep!
          CMP #$06    ;START Pressed?
          BNE END     ;nope...
          LDA #$00    ;Zero last key
          STA CH      ;pressed and
          LDA #$00    ;flags.
          STA FLAG
          STA HOLD
          JMP $7800   ;Main program!
E3        LDA #$30
          STA AUDF1   ;Frequency
          LDA #$0F    ;Store volume
          STA CDTMV3  ;in timer.
E1        LDA CDTMV3  ;Add an offset
          CLC         ;to timer for
          ADC #$A0    ;note volume.
          STA AUDC1
          CMP #$A0
          BNE E1
          STA AUDC1
          LDA HOLD    ;SELECT pressed?
          CMP #$00
          BEQ E5      ;Nope...
          LDA #$00
          STA HOLD
          INC TIME    ;Toggle between
          LDA TIME    ;Day and Night
          CMP #$02    ;TIME=0 ; Daytime
          BNE E4      ;TIME=1 ; Night
          LDA #$00
          STA TIME
E4        TAY
          LDX #$00
          LDA TPOINT,Y ;Get offset for
          TAY         ;Time Print
PLV3      LDA TPRINT,Y ;Print until
          CMP #$5B    ;My EOL
          BNE PLV4
          JMP PLV
PLV4      STA PRINT3,X
          INY
          INX
          JMP PLV3
E5        INC DIFLEV  ;Increase
          LDA DIFLEV  ;difficulty
          CMP #$07    ;level (0-6)
          BEQ PLV5
          JMP PLV     ;and print it.
PLV5      LDA #$00
          STA DIFLEV
          JMP PLV
;-------------------------
;Title Page Display List
;-------------------------
?DLIST    .BYTE $70,$70,$70
          .BYTE $42
          .WORD TEXT
          .BYTE $02,$F0,$07,$00,$F0
          .BYTE $06,$80,$06
          .BYTE $70,$02,$70,$02,$70,$02
          .BYTE $70,$F0,$00,$82,$00,$82
          .BYTE $70,$70,$82,$00,$82,$00
          .BYTE $82,$00,$41
          .WORD ?DLIST
;-----------------------------
;Title Page and Selection Data
;-----------------------------
TEXT
          .SBYTE "     ArgoSoft Softwa"
          .SBYTE "re Productions      "
          .SBYTE "              Proudly "
          .SBYTE "Presents            "
          .SBYTE "     LABYRIN"
          .SBYTE "TH5        THE ULTIMA"
          .SBYTE "TE 3-D      MAZE ADVE"
          .SBYTE "NTURE              A.N"
          .SBYTE ".A.L.O.G. Computing"
          .SBYTE "                Progra"
          .SBYTE "mmed  By Steven Lash"
          .SBYTE "ower"
          .SBYTE "       (C) 1987 Argo5"
          .SBYTE "oft Software Product"
          .SBYTE "ions"
          .SBYTE "     Number of Argon"
          .SBYTE "ian Patrollers->0"
          .SBYTE "        You Will Atte"
          .SBYTE "mpt A              "
          .SBYTE "Escape"
          .SBYTE "      Option Diffic"
          .SBYTE "ulty Level -        "
          .SBYTE "      Select Daytim"
          .SBYTE "e or Nightime "
          .SBYTE "Escape "
          .SBYTE "      Start Enter t"
          .SBYTE "he Labyrinth        "
          .SBYTE "  "
?COLORS   .BYTE $C2,$B2,$A2,$92
          .BYTE $82,$00
LEVELS    .SBYTE "Beginner&Easy     "
          .SBYTE "&Moderate&Hard    &T"
          .SBYTE "rivial&Sadistic&Insa"
          .SBYTE "ne  &"
POINT     .BYTE $00,$09,$12,$1B
          .BYTE $24,$2C,$35
TPRINT    .SBYTE "Daylight&"
          .SBYTE "Nightime&"
TPOINT    .BYTE $00,$09
          *=  $02E0
          .WORD $6900
```

Shuttle II

MICHTRON
576 S. Telegraph
Pontiac, MI 48053
(313) 334-5700
Low resolution $39.95

by Andy Eddy

MichTron is one of the ST's biggest software development supporters, and,with the assistance of their British sister company, MicroDeal, they have provided ST users with a wealth of quality titles for work and play. But you have to figure that, every once and a while, their quest for excellence slips, letting a below-par effort through. With that in mind, we'll introduce **Shuttle II**, which was produced by MicroDeal. It's not a *terrible* game *per se;* it just has problems in some of its player interaction.

Your objective is to control a Space Shuttle mission through its paces, from choosing a launch site until the touchdown. Along the way, you face challenges such as takeoff, booster and external tank separation, tracking and retrieving a satellite, atmospheric reentry, and, finally, landing. Every main flying stage has an accompanying chart showing the flight path you're supposed to take, which you must keep to as much as possible, to get the highest score.

As explained in the manual, it's not an arcade like simulation, but rather a "highly enjoyable pointsscoring game." Unfortunately, gameplay is inconsistent. I frequently found myself watching the screen with little to do; at other times, I was at a loss to keep up with the chores required of me. At further stages, my keyboard inputs didn't do what they were supposed to. This supplanted most of the intended "enjoyable pointsscoring."

Goals like tank separations (tasks that will earn you an added bonus if you complete them successfully and at the correct time) are accomplished by hitting a particular key on the keyboard. These game sections either didn't work for me, or the time allowed to strike the key was so short I found it difficult to hit at the right moment. Strangely enough—though I am thankful, under the circumstances—the mission won't be squashed by those failings. The computer will take care of missed jobs, but you lose the bonus you would have received.

When you've reached outer space, your next task is to find a wayward satellite. Maneuvering with the aid of your onboard guidance system is easy enough, but using the mouse to control thrusters (nose up/down, wing roll and forward thrust) lacks the necessary quick response. Similarly, trying to keep your eyes on your guidance system reading *and* the mouse pointer location is a strain.

Retrieving the satellite is a bother, also. The documentation claims that all you have to do is space-walk your astronaut up to the floating cargo and "touch" its wings to dock with it. Many times, I cruised him up, down and crossways in juxtaposition to the satellite—to no avail; when I did link with it, it seemed to be at arbitrary moments.

During the last two phases, you're reentering the atmosphere and landing the shuttle. While reentering, the craft is subject to severe heat, so you must monitor the shell temperature, as well as control the craft through an S-turn on approach to the runway. I've had a hard time getting out of this with my ship intact. But, regardless of how you complete this phase, you're moved on to the landing phase—a difficult phase to complete with a hunk of unpilotable scrap metal.

The landing screen is equally difficult to clear, and, no matter what the outcome —usually unsuccessful—you reach the high score table with congratulations on a fine mission. Hmmm, not the kind of reception you'd expect for a poor navigating job.

I don't want to say that **Shuttle II** is a total waste. It's just that, with so many choices in the ST marketplace, you could do better. On the other hand, the programming is decent and the graphics well conceived, showing that the creators have the ability to put together a satisfactory contest, *provided the concept is good.* Too bad it wasn't in this case.

Andy Eddy works as a cable TV technician in Connecticut, but has been interested in computers since high school. While his family's Atari 800 is four years old, he's been avidly playing arcade games since Space Invaders and is a former record holder on Battlezone.

When you want to talk Atari

XL/XE HARDWARE

INTERFACES

ICD
P:R Connection $58.99
Printer Connection 39.99
Supra
1150 ..... 38.99
1151 (1200 XL) 39.99
Xetec
Graphix Interface 38.99
Atari
850 Interface 109.00

COMPUTERS

Atari 130XE **$139**

Atari
65 XE 99.99

XL/XE ENHANCEMENTS

Axion 32K Memory Board (400/800) 19.99
Atari 80 Column Card 84.99

MODEMS

Atari
SX212 300/1200 (ST) 88.99
835 ..... 19.99
XMM301 ..... 42.99
Anchor
VM520 300/1200 ST Dir. Con ... 119.00
Avatex
1200 HC 99.99
2400 ..... 209.00
Supra
2400 Baud XL, XE 169.00
2400 Baud ST 169.00

MONITORS

Magnavox
CM8502 13" Composite & cables 169.00

ST HARDWARE

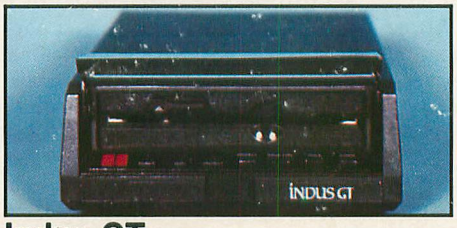
520 ST FM RGB/COLOR **$759**

Includes: 520 ST FM with 3½" drive built-in, mouse, power supply and 1224 color monitor.

1040 RGB/Color System 929.00
1040 Monchrome System 769.00
1040 Computer (no monitor) 639.00
520ST FM Monochrome System (Includes: 520ST, internal drive, modulator, mouse 1, Basic and monochrome monitor) . 649.00
SM124 Monochrome Monitor 159.00
SM1224 Color Monitor 329.00

DRIVES

Atari
XF551 Drive (XL/XE) 179.00
SHD204 20 Meg Hard Drive (ST) 549.00
AA314 DS/DD Disk (ST) 199.00
AA354 SS/DD Disk (ST) 125.00

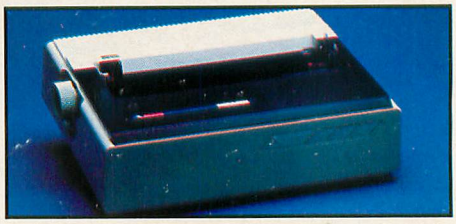


Indus GT Disk Drive (XL/XE) **$179**

I.B.
5.25 ST Drive 229.00
Indus
GTS100 3½" Drive (ST) 199.00
Supra
20 Meg Hard Drive (XL/XE) 639.00
20 Meg Hard Drive (ST) 539.00
30 Meg Hard Drive (ST) 689.00

PRINTERS

Atari
1020 XL/XE $31.99
XXM804 ST 179.00
XMM801 XL/XE 189.00



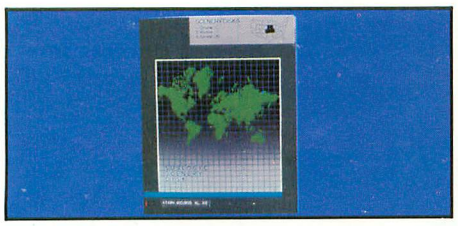
Atari XDM121 Letter Quality XL/XE **$149**

Brother
M-1109 100 cps, 80 col. 199.00
M-1409 180/45 NLQ w/tractor ... 299.00
HR20 22 cps Daisywheel 339.00
Citizen
MSP-10 160 cps, 80 col. 269.00
Premier 35 cps Daisywheel 469.00
MSP-25 200 cps, 132 col. 299.00
Espon
LX-800 150 cps, 80 col. 179.00
FX-86E 240 cps, 80 col. ..... Call
FX-286E 240 cps, 132 col. ..... Call
EX-800 300 cps, 80 col. ..... Call
LQ-850 180 cps, 24 wire ..... Call
HI-80 4 pen plotter 249.00
NEC
P2200 24 wire pinwriter 369.00
P660 (180 cps draft/65 LQ) 459.00
P760 132 col (180 cps draft/65 LQ) 609.00
Okidata
Okimate 20 ..... 129.00
ML-192, ML-193 ..... Call
ML-292, ML-293 ..... Call
Panasonic
KX-1080i 144 cps 80 col. 189.00
KX-1091ii 194 cps 132 col. 199.00
KX-1092 240 cps 80 col. 319.00
1595 200 cps 132 col. 449.00
Star Micronics
NX-1000 144 cps 80 col. 169.00
NX-15 120 cps 132 col. 319.00
ND-15 180/45 NLQ 419.00
Toshiba
P321SL 216 cps 24 Pin 519.00

COMPUTER MAIL ORDER

.....you want to talk to us.

XL/XE SOFTWARE



Sublogic Scenery $15.99

Access
Leaderboard Golf 24.99
Tournament Disk 13.99
Accolade
Hardball 19.99
Atari
Atari Writer Plus 35.99
Music Painter 11.99
Batteries Included
Home Pack 18.99
Broderbund
Printshop 25.99
Karateka 12.99
Cygnus
Starfleet I 32.99
Datasoft
Alternate Reality (City) 23.99
Electronic Arts
Touchdown Football 11.99
Pinball Construction 7.99
Infocom
Zork III 24.99
Microprose
Top Gunner 15.99
F-15 Eagle Strike 21.99
Silent Service 21.99
Origin Systems
Ultima 4 ..... 36.99
Roklyn SPECIAL
Gorf/Wizard of Wor/AT Deluxe
Anti-Sub/Journey to Planet (ea.) ... 3.99
Strategic Simulations
Gemstone Warrior 11.99
Sublogic
Flight Simulator II 31.99
Scenery FL, NC, SC 14.99
X-Lent
Typesetter 21.99
Printshop Interface 22.99

ACCESSORIES

Maxell
MD1-M SS/DD 5¼" $7.99
MD2-DM DS/DD 5¼" 8.99
MF-1DDM SS/DD 3½" 12.49
MF2-DDM DS/DD 3½" 18.49
Sony
MD1D SS/DD 5¼" 7.99
MD2D DS/DD 5¼" 9.49
MFD-1DD SS/DD 3½" 12.49
MFD-2DD DS/DD 3½" 19.49
Allsop Disk Holders
Disk File 60-5¼" 9.99
Disk File 30-3½" 9.99
Curtis
Emerald 39.99
Safe Strip 19.99
Universal Printer Stand 14.99
Tool Kit 22.99
ICD (XL/XE)
Sparta DOS Construction Set 28.99
US Doubler/Sparta DOS 47.99
Real Time Clock 48.99
Rambo XL 28.00
Multi I/O Board 256K 169.00
Multi I/O Board 1 Meg 299.00

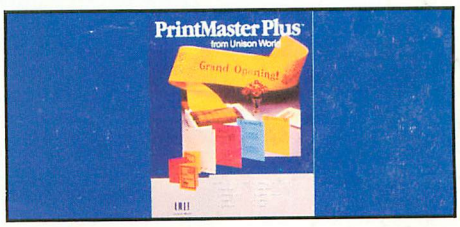
ST SOFTWARE

Activision Music Studio $29.99

Access
Leaderboard Golf 24.99
Antic
Stereo CAD 3-D 54.99
Atari
Algebra I Vol II GRD 7-9 ..... 16.99
Avant Garde
PC Ditto 74.99
Batteries Includes
Degas Elite 39.99
Cygnus
Starfleet I 33.99

ST SOFTWARE

DAC
Payroll 44.99
Easy Accounting 64.99
Electronic Arts
Gridiron Football/Auto Duel .. (ea.) 30.99
Firebird
Guild of Thieves 25.99
Infocom
Beyond Zork 39.99
Enchanter 22.99
Michtron
Major Motion 25.99
Microprose
Silent Service 25.99
F-15 Strike Eagle 24.99
Miles Software
ST Wars 28.99
Mark Williams
C 119.00
Paradox
Wanderer (3D) 25.99
War Zone/Fireblaster 26.99
Psygnosis
Barbarian 25.99
Deep Space 26.99
Wood Perfect 4.1 ..... 239.00
Strategic Simulations
Rings of Zilfin 23.99
Sublogic
Flight Simulator II 32.99
Timeworks
Swiftcalc/Wordwriter ..... (ea.) 47.99
Partner ST 41.99



Unison World Printmaster Plus $23.99

VIP
Professional (GEM) 144.99

In the U.S.A. and in Canada

Call toll-free: 1-800-233-8950

Outside the U.S.A. call 717-327-9575, Fax 717-327-1217

Educational, Governmental and Corporate Organizations call toll-free 1-800-221-4283

CMO. 477 East Third Street, Dept. B7, Williamsport, PA 17701

ALL MAJOR CREDIT CARDS ACCEPTED.

POLICY: Add 3% (minimum $7.00) shipping and handling. Larger shipments may require additional charges. Personal and company checks require 3 weeks to clear. For faster delivery use your credit card or send cashier's check or bank money order. Credit Cards are NOT charged until we ship. Pennsylvania residents add 6% sales tax. All prices are U.S.A. prices and are subject to change and all items are subject to availability. Defective software will be replaced with the same item only. Hardware will be replaced or repaired at our discretion within the terms and limits of the manufacturer's warranty. We cannot guarantee compatibility. All sales are final and returned shipments are subject to a restocking fee.

NOW PLAYING AT A SOFTWARE DEALER NEAR YOU

Available for Amiga, Commodore 64, IBM PC, Apple IIgs, and Atari ST, which are trademarks respectively of Commodore-Amiga, Commodore Electronics, Ltd., International Business Machines, Apple Computer Inc., and Atari Inc.

Cinemaware Corporation, 4165 Thousand Oaks Blvd., Westlake Village, CA 91362

CIRCLE #128 ON READER SERVICE CARD