THE #1 MAGAZINE FOR ATARI COMPUTER OWNERS

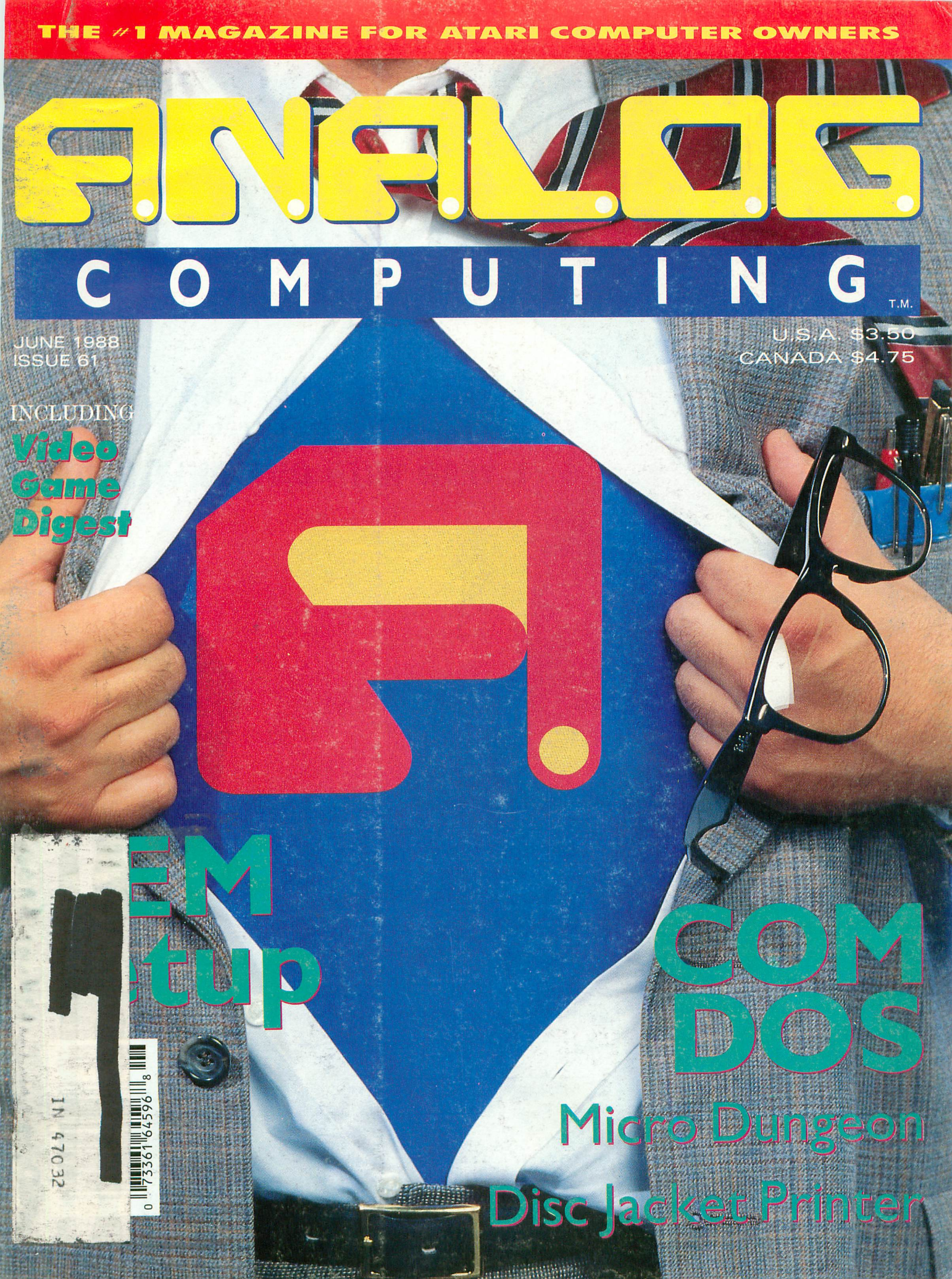
ANALOG COMPUTING

T.M.

JUNE 1988
ISSUE 61

U.S.A. $3.50
CANADA $4.75

INCLUDING
Video Game Digest

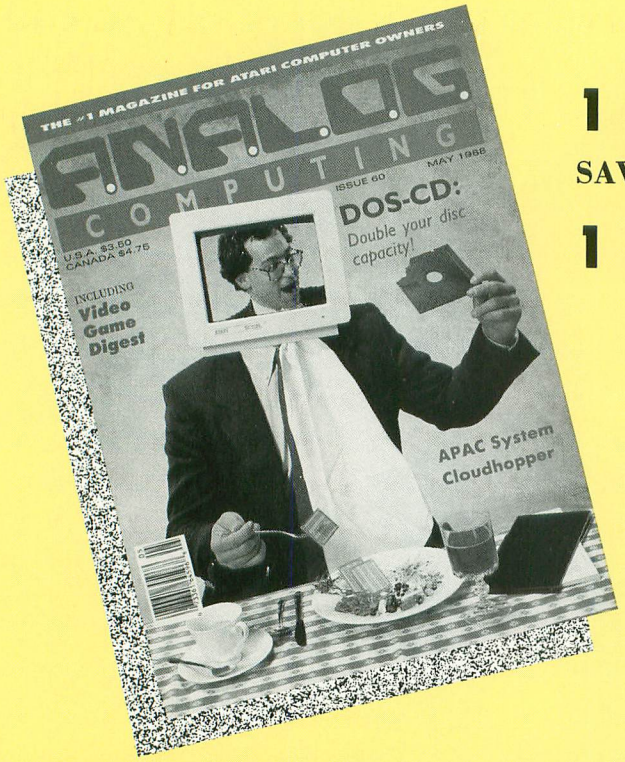


DOS

Micro Dungeon

Disc Jacket Printer

BOOT UP TO BIG SAVINGS!



1 YEAR FOR ONLY $28

SAVE $14 OFF THE COVER PRICE

1 YEAR *WITH DISK* ONLY $105

SAVE TIME AND MONEY SUBSCRIBE TO ANALOG

SAVE $14 OFF THE COVER PRICE WITH THE CONVENIENCE OF HAVING ANALOG DELIVERED DIRECTLY TO YOUR DOOR *BEFORE* IT EVEN HITS THE NEWSSTANDS. GET THE MOST OUT OF YOUR COMPUTER.

SUBSCRIBE TO ANALOG TODAY

☐ 1 YEAR @ $28 — SAVE $14! MCFYY
FOREIGN — ADD $7 PER YEAR
☐ 1 YEAR WITH DISK @ $105 DCFYY
FOREIGN — ADD $15 PER YEAR

☐ PAYMENT ENCLOSED ☐ BILL ME
CHARGE MY: ☐ VISA ☐ MC #_____

EXPIRATION DATE SIGNATURE

MONEY BACK ON ALL UNUSED PORTIONS OF SUBSCRIPTIONS IF NOT SATISFIED.

NAME

ADDRESS

CITY STATE ZIP

MAKE CHECK PAYABLE TO L.F.P., INC., P.O. Box 16927, N. Hollywood, CA 91615. Offer expires August 31, 1988. Your first issue will arrive in 6 TO 8 weeks.

WATCH FOR IT!

Contents

FEATURES

REVIEWS

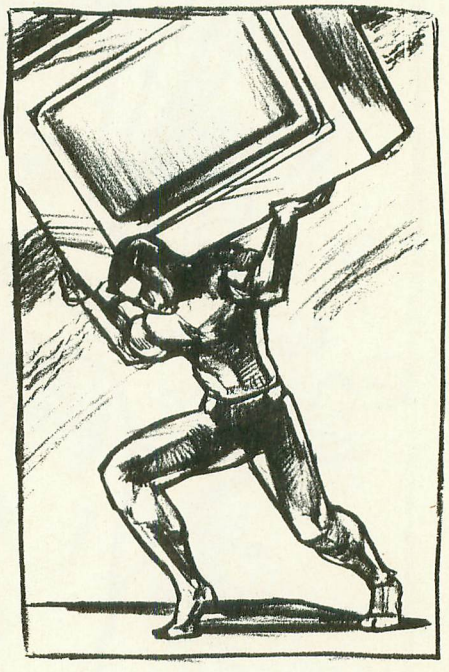
COLUMNS

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.

Editorial

by Clayton Walnum



Ever since the release of the **XE Game System,** the controversy has raged as Atari computer owners all over the country continue to question the company's wisdom in staying active in the electronic entertainment industry. Many believe that for Atari to be assured of continued success, it must obtain its niche in the business world, a place where a connection to **Space Invaders** and **Pole Position** is fatal. How, they say, can Atari gain the respect it deserves in the "real" world when they insist on being a toy manufacturer?

One might better ask: How can a company thrive if they choose to ignore existing opportunities in favor of pursuing long-shots?

The fact is that Atari's biggest market has always been in home electronics, not in business, an area that is almost wholly dominated by IBM fanatics. With this in mind, one has to wonder if Atari hasn't already—knowingly or unknowingly, with or without regrets—chosen the home electronics market as its permanent battleground. And if this is indeed the case, why fight it? The company image we might like to see Atari gain has very little to do with the realities of marketing—realities that are governed almost solely by profit.

Consider this: Atari has admitted that the sales of the 8-bit computer line have been poor at best, and except for a handful of releases from small developers, there's been very little new software. Atari itself released about a half dozen new titles—including **Silent Butler, Star Raiders II, Music Painter** and **Atari Planetarium**—a while back, but has since lapsed back into silence.

What's to be done?

According to Atari's Neil Harris, the XE Game System is part of the solution. The idea was to release a new machine that could provide some competition for Nintendo's entertainment system (it seems that the video game craze is not as dead as some believe, as evidenced by the sales of not only Atari's entries into the market, but by their two major opponents, Nintendo and Sega), while at the same time, boost interest in the existing 8-bit line of computers.

Remember, cartridges for the XE Game System are compatible with the 8-bit computers. It's Atari's hope that the new machine will rekindle the interest of third-party software developers (the fact that cartridge-based software is much harder to pirate will certainly help), who can take advantage of a double audience: both video game owners and computer owners.

Atari feels that they can sell enough of the new machines to make software developers sit up and take notice. They also believe that, because the XE Game System is expandable and comes with Atari BASIC, many people, in the midst of their gaming, will discover a latent interest in computing and move up to other machines. The more people that make that transition, the more demand there'll be for more "serious" software—and the more incentive there'll be for developers to meet that demand.

What's in it for you? Hopefully, this'll mean many new titles for your XL or XE computers—because the cartridges being released for the XE Game Machine are 100% compatible with the 8-bit computer line. And if the scenario discussed in the preceding paragraph—game system owners discovering an interest in computing—comes about, perhaps there'll be more forthcoming than just games.

Sound unlikely? Maybe. Evaluating a potential market is a tough job, one that is fraught with pitfalls. The number of variables involved, often variables the manufacturer is unaware of until it's too late, make marketing anything but a science.

All we can do right now is sit back and see what happens as the entertainment system giants jockey for position in the marketplace. If the XE Game System is the success that Atari predicts it'll be, it should bring good things for owners of 8-bit computer systems. If nothing else, it'll mean continued health for Atari in the future and a flow of revenue to finance other new and exciting projects.

Hey. We all have to make the money where we can. A

ANALOG COMPUTING STAFF

WHERE TO WRITE

Editor, **ANALOG Computing,** 9171 Wilshire Blvd., Suite 300, Beverly Hills, CA 90210

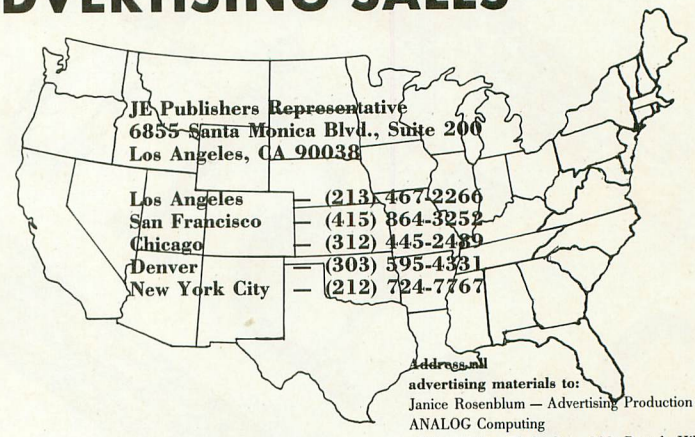
Correspondence regarding subscriptions, including problems and changes of address, should be sent to: **ANALOG Computing,** P.O. Box 16927, North Hollywood, CA 91615, or call 1-818-760-8983.

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.** For further information, write to **ANALOG Computing,** P.O. Box 1413-MO, Manchester, CT 06040-1413

8-Bit News

by Frank Cohen

ICD Buys OSS

In a move to expand their product line, ICD has begun marketing the Optimized Systems Software product line. OSS was one of the first software companies to support the Atari 800 in the early 1980's. Popular OSS products such as Action and Mac 65 are available through ICD and its dealer network. ICD began talking with OSS late in 1987 with a contract signed in January of 1988. OSS has been struggling for the past year. Bareware, OSS's line of inexpensive ST software, never really caught on due to OSS's financial woes. ICD is currently reviewing the feasibility of the Bareware line.

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

hard disk drive when upgrading to ST systems. A $20 cable is needed to connect the ICD hard disk to the ICD MIO Board.

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

ICD Announces SpartaDOS X

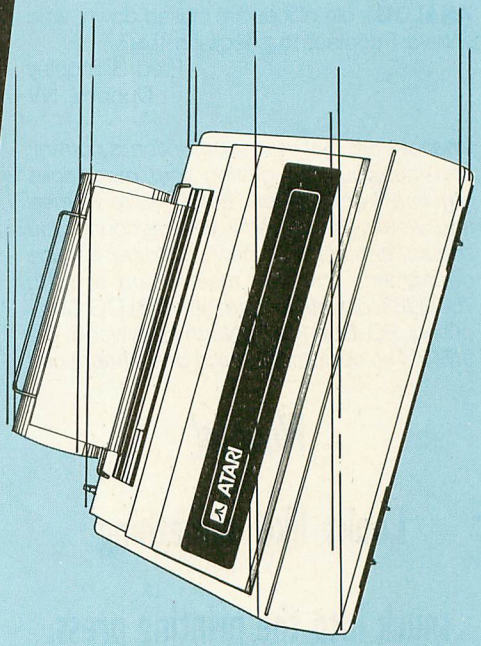
ICD will be shipping SpartaDOS X in mid-1988. The new disk operating system for 8 bit users will come in a plug-in cartridge and support 'fast' disk I/O routines. Loading of DOS will take half the time it takes to load the current system utilities. SpartaDOS X is compatible with the new Atari XF551 Disk Drive.

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

Dataque Offers PAL

The PAL system is a programmer's friend. PAL supports address vector lookup, hex calculator math, ATASCII to HEX conversion, 6502 instruction set conversions, trapping, disk utilities and more. PAL works with your stock XL/XE systems, requiring no changes or modifications to work. Contact Dataque for more information.

Dataque Software, 3308 Park Avenue West, P.O. Box 134, Ontario, Ohio 44862



MIO Board

ICD has temporarily discontinued production of the 1 Megabyte MIO board due to the high price of DRAM memory chips. The MIO board gives your Atari computer a hard disk controller, serial and parallel port, and an externally powered RAM Disk. ICD currently sells the 256K version for $199. DRAM chip prices have risen greatly, making the 1 Megabyte version of the MIO board very costly. Production will resume later this year when DRAM chip prices are expected to fall.

The ICD MIO board allows you to add an ST compatible ICD hard disk to your 8 bit system. ICD is currently shipping 20 to 100 megabyte disk drives that range in price from $699 to $1699. 8 bit owners can use the same

Atari Ships the XF551 Disk Drive

The XF551 floppy disk drive is now shipping. The XF551 is a replacement for the older Atari 1050 floppy disk drive. The XF551 is a double-sided double-density drive that comes with Atari DOS 2.5. The unit is capable of storing 130K bytes using the DOS 2.5, and will later be capable of storing 130K bytes using the new ADOS from Atari, when it becomes available later this year. The XF551 is also compatible with ICD's SpartaDOS 3.2. The XF551 carries a suggested list price of $219.95.

ATari Corp., 1196 Borregas Avenue, Sunnyvale, CA 94086

Atari Income Falls

Atari Corp. said net income for the fourth quarter '87 dropped to $18.70 million from $23 million. The decrease was mostly due to continued losses from Federated Group, an Atari subsidiary. The Federated Group was purchased in 1987 to provide an outlet for Atari computer products. Atari has estimated Federated's return to the black in the fourth quarter of 1988. Atari expects Federated to break-even for calendar '88. Atari sales overall for 1987 were posted at $493.17 million, indicating healthy sales in home computers (51% of sales), video games (23%) and retail business (26%).

Atari Corp., 1196 Borregas Avenue, Sunnyvale, CA 94086

Reader Comment

To say our mailbox has been full lately would be like saying the ocean contains a little water. Now that **ANALOG Computing** is back on the stands, it seems that everyone has the same questions regarding subscriptions, back issues and other related topics. This month's Reader Comment focuses on those questions.

I WANT SERVICE!

It's been a long time since I've received my magazines, and I'm worried that my subscription might not be current. Now that **ANALOG**'s old offices are closed down, who should I contact to check on this?

—Fred Billingsley
Dunphy, NV

Chances are that your subscription is current, and that you'll be getting your magazines regularly from now on. But if you're worried and would like to verify your account, you should contact our new customer-service department. You can reach tham at (818) 760-8983, or write to them at: ANALOG Computing, *P.O. Box 16927, North Hollywood, CA 91615. The customer-service department can handle questions on current subscriptions, ordering information, address changes and back issues.*

LABYRINTH DRAWS A BLANK

Help! I was busily typing in Labyrinth from the April issue, and when I got to Lines 2150 and 2160, I discovered that a couple of numbers were smudged. Please tell me what they're supposed to be so I can finish typing in the game. I'm trying to play it.

—Glen Richards
Windsor, CT

Whoops! Looks like something snuck into the printing press. For those of you trying to type in Labyrinth, the fourth number from the end of Line 2150 should be 164 and the third number in Line 2160 should be 133. And while we're on the subject, if you need program help or technical information on something related to the magazine, you may contact our Technical Line at (203) 645-6236 between the hours of 9:00 a.m. to 5:30 p.m. Monday, Wednesday and Thursday, and 9:00 a.m. to 4:00 p.m. Tuesday and Thursday, EST. Please note that this number is absolutely not *for subscription information or ordering.*

LOOKING BACK

My **ANALOG** collection is sadly missing several issues. I'd really like to fill as many of the holes as I can. Could you tell me whether back issues are still available, and if so, how I can go about getting them?

Elliot White
Portland, OR

We are in the process of moving our inventory of back issues to a new warehouse, as well as organizing them; so that we can process orders in the fastest possible way. Back issues may be ordered by calling (818) 760-8983 (this includes back disk issues as well). However, some issues are sold-out; we'll have information next month on which are still available.

I WANT THE DISK

I just picked up the April issue of **ANALOG Computing** on the newsstand, and now I would like to know how I can get the disk for that issue. The programs look great, but I'm afraid I'm not much of a typist.

—Al Wallach
Lancaster, SC

We are now setting up a new order/process center to handle disk ordering. Readers will be able to order a disk from a current issue and expect to have the disk shipped immediately. Information on disk ordering will be forthcoming. In addition, defective disk returns will be handled on a same-day basis. They should be returned for replacement to: ANALOG Computing, *P.O. Box 1413-M.O., Manchester, CT 06040-1413.*

SOMETHING EXTRA

I waited too long to buy my copy of the **ANALOG 8-bit Extra** (my allowance doesn't go as far as it used to), and now the bookstore where I saw it is sold-out. Can I still get a copy? I didn't see an advertisement for it in the April issue, and I'm afraid it may be sold-out completely.

—Dennis Anderson
San Diego, CA

Never fear. The ANALOG 8-bit Extra *can still be ordered from us. The same is true of the* ANALOG Pocket Reference Card. *Future issues of* ANALOG Computing *will carry advertisements and ordering information for these and other items.*

HERE'S TO THE FUTURE

Boy, it sure was great to go out to my mailbox the other day and find the April issue of **ANALOG** waiting for me. I had given up all hope of ever seeing my favorite Atari magazine again. The new issue is more of the fine stuff I've come to expect from you, and I was delighted to see all the familiar names in the Table of Contents. Welcome back, guys!

—Edward Parker
Portland, ME

Thanks, Ed. We're glad to have been able to give you that nice surprise. And you can expect a lot of other nice surprises in the future.

Whoops!

Looks like something snuck into the printing press.

***I had given up all hope* of ever seeing my favorite Atari magazine again.**

*With the new staff—a group of dedicated and professional people—*ANALOG Computing *will continue to grow and mature. We're striving as never before to attain our full potential—and you'll see the difference. We promise.*

When you want to talk Atari

XL/XE HARDWARE

INTERFACES

ICD
P:R Connection 59.99
Printer Connection 39.99
Supra
1150 . . . . . 38.99
1151 (1200 XL) 41.99
Xetec
Graphix Interface 38.99
Atari
850 Interface 109.00

COMPUTERS

Atari 130XE **$135**

Atari
65 XE 99.99

XL/XE ENHANCEMENTS

Axlon 32K Mem. Board (400/800) . 19.99
Atari 80 Column Card 79.99

MODEMS

Atari
SX212 300/1200 (ST) 89.99
835 . . . . . 19.99
XMM301 . . . . . 42.99
Anchor
VM520 300/1200 ST Dir. Con . . . 119.00
Avatex
1200 HC 99.99
2400 . . . . . 179.00
Supra
2400 Baud XL, XE 169.00
2400 Baud ST 169.00
2400 Baud (no software) 149.00

MONITORS

Magnavox
CM8502 13" Comp. & cables . . . 179.00

ST HARDWARE

520 ST FM RGB/COLOR **$789**

Includes: 520 ST FM with 3½" drive built-in, mouse, power supply and 1224 color monitor.
1040 RGB/Color System 939.00
1040 Monchrome System 779.00
1040 Computer (no monitor) . . . 639.00
520ST FM Monochrome System (Includes: 520 ST, internal drive, modulator, mouse, Basic and monochrome monitor) . . . . . Call
SM124 Monochrome Monitor . . 169.00
SM1224 Color Monitor 329.00

DRIVES

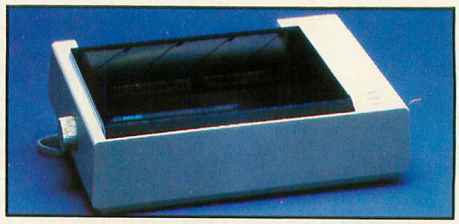
Atari
XF551 Drive (XL/XE) 189.00
AA314 DS/DD Disk (ST) 199.00
AA354 SS/DD Disk (ST) 129.00

Atari SHD204 20 Meg for ST **$559**

I.B.
5¼ ST Drive 209.00
Indus
GTS 100 3½" Drive (ST) 199.00
GT Drive (XL/XE) 179.00
Supra
20 Meg Hard Drive (XL/XE) 639.00
20 Meg Hard Drive (ST) 549.00
30 Meg Hard Drive (ST) 689.00

PRINTERS

Atari
1020 XL/XE Plotter 31.99



Atari XMM 801 **$189**

XM-M801 XL/XE Dot Matrix 189.00
XM-M804 ST Dot Matrix 179.00
XDM 121 Letter Qlty. XL/XE . . . 199.00

Brother
M-1109 100 cps Dot Matrix 199.00
M-1409 180 cps Dot Matrix 309.00
HR-20 22 cps Daisywheel 339.00

Citizen
120D 120 cps Dot Matrix 149.00
180D 180 cps Dot Matrix 169.00
Premier-35 35 cps Daisywheel . 479.00

Epson
LX-800 150 cps, 80 col. 179.00
Hi-80 4 pen plotter 249.00
FX-86E 240 cps, 80 col. . . . . Call
FX-286E 240 cps, 132 col. . . . . Call
LQ-500 180 cps, 24-wire . . . . . Call
LQ-850 330 cps, 80 col. . . . . Call
EX-800 300 cps, 80 col. . . . . Call

NEC
P2200 pinwriter 24-wire 379.00
P660 pinwriter 24-wire 459.00
P760 pinwriter 132 col. 669.00

Okidata
Okimate 20 color printer 129.00
ML-182 120 cps, 80 column 239.00
ML-192 + 200 cps, 80 column . . 359.00
ML-193 + 200 cps, 132 col. 469.00

Panasonic
KX-P1080i 144 cps, 80 col. 189.00
KX-P1091i 194 cps, 80 col. 199.00

Star Micronics
NX-1000 140 cps, 80 column . . . 169.00
NX-15 120 cps, 132 column 319.00

Toshiba
P321-SL 216 cps, 24-wire 499.00

WE SHIP 90% OF ALL ORDERS WITHIN 24 HOURS

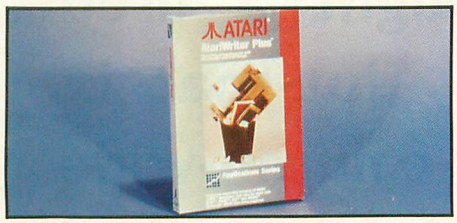
SELECT FROM OVER 3000 PRODUCTS

COMPUTER MAIL ORDER

.....you want to talk to us.

XL/XE SOFTWARE

Access
Leaderboard Golf .....(pk.) 15.99
Tournament Disk 13.99
Accolade
Hardball 19.99
Atari
Filemanager 11.99
Music Painter 11.99



Atariwriter Plus $35.99

Atari Program Exchange
Misc. Programs (cassettes) ....at 1.99
Broderbund
Printshop 26.99
Karateka 14.99
Cygnus
Starfleet I 32.99
Datasoft
Alternate Reality (city) 23.99
Alternate Reality (dungeon) 25.99
221 Baker St. 21.99
Electronic Arts
Touchdown Football 11.99
Pinball Construction 9.99
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 22.99
Printshop Interface 21.99

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 29.99
Multi I/O Board 256K 169.00
Multi I/O Board 1 Meg 299.00
US Doubler 28.99

ST SOFTWARE

BATTERIES INCLUDED
Degas Elite $38.99

Access
Leaderboard Golf 22.99
Activision
Hacker II Doomsday 29.99
Antic
Stereo CAD 3-D 54.99
Atari
Algebra I Vol II GRD 7-9 .....16.99
Avant Garde
PC Ditto 64.99
Comnet
ST Term 21.99

ST SOFTWARE

Electronic Arts
Gridiron Fooball/Auto Duel.(ea.) .30.99
Isgur Portfolio 129.00
Firebird
Jewels of Darkness 21.99
Infocom
Beyond Zork 34.99
Michtron
Major Motion 25.99
Microprose
Gunship 32.99
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/Deep Space .....(ea.) 26.99
Soft Logik Corp.
Publishing Partner 59.99
Strategic Simulations
Rings of Zilfin 23.99
Sublogic
Flight Simulator II 32.99
Timeworks
Swiftcalc/Wordwriter .....(ea.) 47.99
Partner ST 37.99

DAC
Easy Accounting $59.99

Unison World
Printmaster Plus 28.99
Word Perfect Corp
Word Perfect 4.1 .....199.00

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.

CIRCLE #103 ON READER SERVICE CARD.

B706

Gem Set-up

I recently bought a used GEMINI-15 dot-matrix printer with a 132-column-wide carriage. I took it home, hooked it up, loaded some paper, and began listing the BASIC programs I had been writing. My first clue that I had a problem was the sight of the print head marching off the right edge of the 80-column paper, inking its way across the rubber roller. The manual revealed that the printer's programmable features have default settings that load in automatically each time the power is turned on. The default right margin of a Gemini 15 turns out to be 132.

by Jackson Beebe

To change these settings, you must send Functional Commands, or function codes, to the printer, using LPRINT statements. Function codes consist of an ESC command in combination with other numbers and letters. For example, to select italics, you type LPRINT CHR$(27);CHR$(52). The CHR$ function returns the character represented by the ATASCII number code specified, e.g. CHR$(27) is ESCAPE, CHR$(52) is '4'. All printer options and examples of command statements needed are listed in the printer manual. Typing function codes for the options you want to use allows you to print or list text in any format. To avoid having to look up and type all those codes every time I wanted to use my printer I wrote Gem Set-up, a BASIC utility program that allows you to choose type styles and sizes, turn options on or off, alter the top, bottom, left and right margins, and change line spacing, all without any knowledge of printer function codes or programming. Instructions are available from the menu by choosing Help.

How It Works

Line 50 calls a subroutine at the end of the program (line 1190) that dimensions the string variables and initializes the default values for the menu. Lines 1230-1260 zero out the error register (memory location 195), and set error trapping to line 1250, which turns the screen red, and prints a message if the printer is off (error 138). Users may customize the default right margin, by resetting the value of R in line 1280.

Lines 70-290 turn the screen and border black, and print the menu with the current values of the options and functions, as well as their permissible ranges. Options that are on/off (such as Condensed print or Italics), use an asterisk on the menu to indicate that the option is on. Boxes are drawn around menu items (lines 70-90) using control characters found in your owner's manual. Line 290 opens an IOCB to the keyboard for input. This eliminates the question mark and the need to type RETURN at the COMMAND prompt.

Lines 310-590 are a stack of If-Then statements that respond to correct menu choices, in either upper or lower cases. These lines send the printer function codes, then reprint updated values on the menu and return to line 290 for input. They also send the program to input and error checking routines. Simple on/off choices are handled right in the If-Then statements. These make use of the asterisk's presence or absence, to determine whether to toggle a function on or off. If the asterisk is present the program assumes the function is on, and toggles it off, erases the asterisk from the menu, and loads a blank in the assigned string variable (C$). If the asterisk is absent, the program toggles the function on, assigns an asterisk to the variable (C$), and prints it. If the input 'falls through' all the If-Then statements without finding a match, line 600 considers it an error and returns to line 290 to await more input.

For options that require numerical input (such as Bottom Margin), the variable LINE is set to the first line number of the routine (see line 630). LINE holds the 'return address' of the function while GOSUB 1010 sends the program to an error checking subroutine. If the input is null, line 1010 POPs the stack (necessary when jumping out of a subroutine, to keep return addresses straight), and goes to the line number stored in LINE to reprint the input prompt. If input is not null, lines 1020-1040 check the input one character at a time to insure the ATASCII code is correct for numbers. Errors send the program to line 1050, which prints an error statement, waits for a count of 200, POPs the stack, and goes to the line number stored in LINE. If no errors are found the input is checked for correct numerical range. Input of numbers (line 620) is done using string variables (A$), to take advantage of Atari's string commands to check each character at subroutine 1010. On return, the string variable is converted to a numerical variable using the VAL function (line 640), then checked for correct numerical range. Errors send the program to line 1070, which prints an error statement, waits for a count of 200, and returns. Finally, control codes are sent to the printer, the previous menu entry is erased by printing blanks over it, the new value is printed at the proper place on the menu, and the program goes to line 290 to await more input. This process is similar for Page Length, Left Margin, Right Margin, Line Spacing, and Top Line.

Mini print selects superscript mode, which produces a very tiny print. When this option is in use, use a smaller value for line spacing, like 6/72. An excellent tiny print style can be obtained by choosing Condensed and Mini print with a Spacing of 5.

Proportional Spacing is slightly more involved. When it is selected from the menu, the program first checks the status of the Proportional Flag (PFL is initialized to zero at line 1210). If Proportional Spacing is off (PFL=0; line 790), an input prompt is printed, input is checked for numbers and value, and control codes are sent to the printer. The existing value of Font is erased, the Font status is stored in a temporary variable MEM$, and the Font variable is assigned a blank (F$=" "). Most of this is required in case the Help function is used while Proportional Spacing is turned on. When this happens, it is necessary to reprint all values correctly when leaving Help and returning to the menu. Finally, line 830 loads the correct value in P$, prints the current Proportional Spacing value on the menu, sets the Proportional Flag to on (PFL=1), and returns to line 290 for input.

If Proportional Spacing is already on, lines 770-780 send the printer function codes to turn Proportional Spacing off, reload the correct menu value of Font from MEM$, reprint the correct Font on the menu, erase the Proportional Spacing value on the menu, reset the flag (PFL=0), assign a blank to P$, and return to line 290 for input.

Line spacing may be set in 1/72 of an inch increments. Normal default spacing is 1/6 inch (12/72). You can double space by choosing 24 (24/72), triple space with 36 (36/72), etc. Line spacing should be set *after* type size is selected.

Selecting Help from the menu turns the screen green, and gives you instructions. Pushing RETURN reprints the menu.

Using Gem Set-up

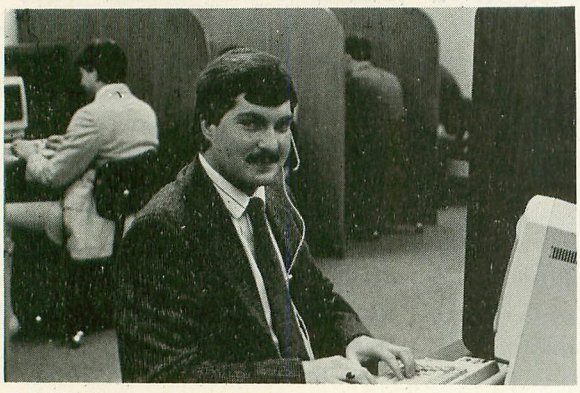
To use Gem Set-up, first align the

Lyco Computer

Marketing & Consultants

Since 1981

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. I 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.

Monitors

Thomson:

230 Amber TTL/12"	$79.95
4120 CGA	$225.95
4160 CGA	$259.95
4460 EGA	$319.95
4375 UltraScan	$389.95
GB 100 EGA Card	$129.95
GB 200 Super Card	$219.95

Magnavox:

BM7652	$84.95
BM7622	$84.95
7BM-613	$79.95
7BM-623	$79.95
CM8502	$189.95
CM8505	$209.95
CM8562	$239.95
CM8762	$249.95
8CM-515	$269.95

Blue Chip:

BCM 12" Green TTL	$64.95
BCM 12" Amber TTL	$69.95

NEC:

Multisync II	$599

Save $210 over NEC Multisync with Thomson 4375 UltraScan $389.95

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

OUR PRICE $89.95

Hayes Compatible

ATARI

Access:

Triple Pack	$11.95
Leader Board Pack	$14.95

Activision:

Music Studio	$19.95
Solid Gold Vol.#1	$10.95

Batteries Included:

Paperclip 80 Col	$31.95

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
'87 Team 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
Phantasie	$22.95
Wargame Construc.	$16.95
Wizards Crown	$22.95
Phantasie II	$22.95
Shiloh	$22.95
Eternal Dagger	$22.95

Sublogic:

Flight Simulator II	$31.49

ATARI ST

Microleague:

Microleague Baseball	$33.95
General Manager	$16.95
Wrestling	$25.95

Microprose:

Silent Service	$22.95
F-15 Strike Eagle	$24.95
Gunship	$28.95

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

Timeworks:

Wordwriter ST	$44.95
Partner ST	$39.95
Data Manager ST	$44.95

Unison World:

Art Gallery 1 or 2	$14.95
Print Master	$19.95
Fonts & Borders	$17.95
Music Studio	$27.95
Bureaucracy	$22.95

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	$11.95

Firebird:

Pawn	$25.95
Starglider	$25.95
Golden Path	$25.95
Guild of Thieves	$25.95
Tracker	$25.95

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

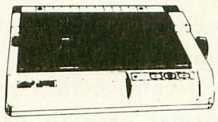
Joysticks

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

1-800-233-8760

star MICRONICS INC NX-1000

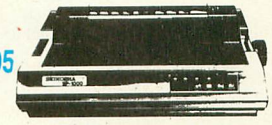
- 144 cps Draft
- 36 cps NLQ
- EZ Operation Front Panel Control



$179.95

SEIKOSHA Sp 180Ai

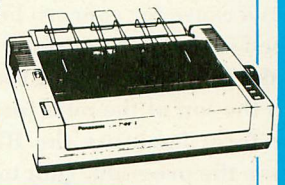
- 100 cps draft
- 20 cps NLQ



$129.95

Panasonic 1091 Model II

- 192 cps Draft
- 32 cps NLQ



$199.95

PRINTERS

star MICRONICS INC

NX-1000	$179.95
NX-1000C	$179.95
NX-1000 Color	$225.95
NX-1000C Color	$229.95
NX-15	$309.95
NR-10	$339.95
NR-15	$439.95
NB-15 24 Pin	$699.95
NB24-10 24 Pin	$425.95
NB24-15 24 Pin	$579.95

BROTHER

M1109	$195
M1409	$299
M1509	$335
M1709	$475
Twinwriter 6 Dot & Daisy	$899
M1724L	$599
HR20	$339
HR40	$569
HR60	$709.95

SEIKOSHA

SP 180Ai	$129.95
SP 180VC	$129.95
SP 1000VC	$139.95
SP 1000AP	$169.95
SP 1200VC	$155.95
SP 1200Ai	$165.95
SP 1200AS RS232	$165.95
SL 80Ai	$299.95
MP1300Ai	$269.95
MP5300Ai	$399.95
MP5420Ai	$879.95
SP Series Ribbon	$7.95
SK3000 Ai	$339.95
SK3005 Ai	$419.95
SPB 10	$CALL
SL 130Ai	$599.95

Toshiba

321SL	$489
341 SL	$659
P351 Model II	$899
351 SX 400 cps	$1019

EPSON

LX800	$179.95
FX86E	$279.95
FX286E	$424.95
EX800	$399.95
LQ500	$309.95
LQ1000 w/Tractor	$549.95
LQ2500	$819.95
GQ3500	$LOW
LQ850	$489.95
LQ1050	$659.95

CITIZEN

120 D	$169.95
180 D	$189.95
MSP-10	$259.95
MSP-40	$309.95
MSP-15	$349.95
MSP-50	$399.95
MSP-45	$459.95
MSP-55	$539.95
Premiere 35	$499.95
Tribute 224	$649.95

Panasonic

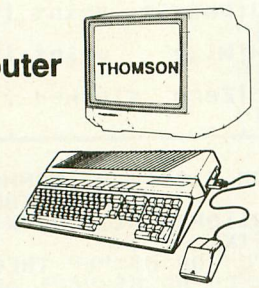
1080i Model II	$179.95
1091i Model II	$199.95
1092i	$319.95
1592	$409.95
1595	$459.95
3131	$299.95
3151	$479.95
KXP 4450 Laser	$CALL
1524 24 Pin	$559.95
Fax Partner	$589.95

OKIDATA an OKI AMERICA company

Okimate 20	$119
Okimate 20 w/cart	$179.95
120	$189.95
180	$219.95
182+	$225.95
183	$249.95
192+	$309.95
193+	$449.95
292 w/interface	$449.95
293 w/interface	$585.95
294 w/interface	$819.95
393	$955.95

ATARI 520 ST Computer

- Built-in Drive
- Thomson 4120 Monitor



$769.95

ATARI HARDWARE

520 ST FM Mono	$675.95
520 ST FM Color	$769.95
1040 ST Mono	$799.95
1040 ST Color	$955.95
130XE Computer	$135.95
SX551 Drive	$174.95
SF 314 Disk Drive	$219.95
Indus GT Atari Drive	$169.95
SHD 204 20 MEG Drive	$579.95
XM301 Modem	$42.95
SX212 Modem	$89.95
GTS 100 (3.5" DSDD ST)	$195.95
GTS 1000 5¼ DSDD ST	$CALL

ATARI 520 ST-FM Monochrome System

$675.95

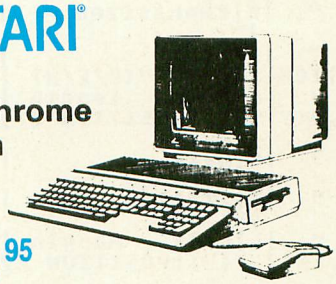
Internal drive included

ATARI 1040 ST Color System

$955.95

Attention Educational Institutions:
If you are not currently using our educational service program, please call our representatives for details.

ATARI 1040 Monochrome System



$789.95

CIRCLE #104 ON READER SERVICE CARD.

print head with the top of a form, then turn on the printer's power and run the program. Choose the type style, options, and margins that you need. Each time you select a new value for an option, the printer will 'creep' up one line. This is not a cause for concern because a form feed is sent to the printer when you Quit the program, aligning the print head with the top of the next form. To avoid the form feed, use the BREAK key to leave the program. Quit to BASIC, and send your output to the printer. To list a BASIC program, load it into memory, then type LIST 'P:' or LIST 'P:',10,50 to list the range of lines 10 through 50.

Perforations can be skipped when printing by altering the bottom margin and the top line. Setting the bottom margin to 2 and the top line to 3 gives you two blank lines before and after each perforation. Set margins last; they vary with the type size selected.

STAR Version

Gemini and Star printers have several small differences in their function codes. Listing 1 is the Gemini Version. To modify the program for printers, make the following changes to Listing 1.

line 10	change GEMSETUP to STRSETUP
line 80	change inverse GEMSETUP to inverse STRSETUP
line 200	change (1-6) to (1)
line 580	change CHR$(86) to CHR$(92)
line 590	change CHR$(86) to CHR$(92)
line 770	replace with: 770 IF PFL=1 THEN LPRINT CHR$(27); CHR$(112); CHR$(0):F$=MEM$:POSITION 22,7:? F$: POSITION 22,13:? " "
line 780	remains the same
line 790	replace with: 790 IF PFL=0 THEN P=1:LPRINT CHR$(27); CHR$(112); CHR$(1):POSITION 22,7:? " "; MEM$=F$:F$=" "
line 800	replace with: 800 IF PFL=0 THEN POSITION 22,13: ? P:PFL=1:GOTO 290
line 810-830	delete
line 1090	change inverse GEMSETUP to inverse STRSETUP
line 1100	change GEMINI to STAR

The Near Letter Quality option (Font=N) is only present in the Star 10/15 printers.

Type font may be toggled between Pica(P), Elite(E), and Near Letter Quality (N—Star 10/15.) Option N on Gemini produces Condensed print.

Unidirectional print may be used for accurate alignment of vertical and horizontal lines.

If you renumber the program, take special care to manually enter new values for LINE in the lines currently numbered 630, 680, 730, 800, 860, 910, 960 and 1160 (eight places). Without this, the program will not return to the correct line if errors are found in the error checking subroutine.

Jackson Beebe is a **health-care center director** in Champaign-Urbana, Illinois. He has owned his 800XL (upgraded to 256K) since the fall of 1984. He teaches **Introduction to BASIC** at Parkland College, and is **President of the Progressive Atari Computing User Group (*PAC*)** of Central Illinois.

Listing 1: BASIC

```
XT 10 REM ** GEMSETUP **  9/86
PU 20 REM Sets Gemini printer from Atari
NN 30 REM Jackson Beebe/807 W. Hill St/Ur
   bana/Illinois/61801
ZU 40 REM ** INITIALIZE **
QW 50 GOSUB 1190
RT 60 REM ** MAIN MENU **
EV 70 ? CHR$(125):POKE 710,0:POKE 712,0:P
   OSITION 4,0:? "┌──────────────────────
   ──────────────────────────────────────
   ──┐"
TH 80 ? "  │ Ver 3.1  GEMSETUP  J. Beebe
   ■│"
JP 90 ? "  ├──┬──────────────┬──┬──┬──┬──
   ──┤"
QV 100 ? "  |B|Bottom margin |";B;"  |";:
   POSITION 27,3:? "(0-16)  |"
TN 110 ? "  |C|Condensed     |";C$;"  |";
   "  |"
VK 120 ? "  |D|Double strike |";D$;"  |";
   "  |"
QK 130 ? "  |E|Emphasized    |";E$;"  |";
   "  |"
DL 140 ? "  |F|Font           |";F$;"  |";
   :POSITION 27,7:? "(P/E/N) |"
VX 150 ? "  |G|PaGe  length  |";G;:POSITI
   ON 25,8:? "|";:POSITION 27,8:? "(1-127
   ) |"
EQ 160 ? "  |H|Help          |   |
   |"
XO 170 ? "  |I|Italics       |";I$;"  |";
   "  |"
LS 180 ? "  |L|Left  margin |";L;:POSITI
   ON 25,11:? "|";:POSITION 27,11:? "(1-2
   55) |"
UI 190 ? "  |M|Mini    print |";M$;"  |";
   "  |"
EI 200 ? "  |P|Proportional  |";P$:POSITI
   ON 25,13:? "|";" (1-6)   |"
XV 210 ? "  |Q|Quit to BASIC |   |
   |"
KT 220 ? "  |R|Right  margin |";R;:POSITI
   ON 25,15:? "|":POSITION 27,15:? "(1-25
   5) |"
VD 230 ? "  |S|Spacing 72nds |";S$;" |";"
   (1-127)  |"
OP 240 ? "  |T|Top    line   |";T;:POSITI
   ON 25,17:? "|":POSITION 27,17:? "(1-16
   )  |"
HX 250 ? "  |U|Unidir  print |";U$;"  |";
   "  |"
GR 260 ? "  |W|Wide    print |";W$;"  |";
   "  |"
EI 270 ? "  |Z|Zero  slashed |";Z$;"  |";
   "  |"
FS 280 ? "  └──┴──────────────┴──┴──┴──┴──
   ──┘"
XU 290 POSITION 2,22:? "  COMMAND >
                    ";:POSITION 13,22:? "
   ";:CLOSE #1:OPEN #1,4,0,"K:":GET #1,K
PX 300 B$=CHR$(K)
DI 310 IF B$="B" OR B$="b" THEN 620
FX 320 IF (B$="C" OR B$="c") AND C$=" " T
   HEN LPRINT CHR$(15):C$="*":POSITION 22
   ,4:? C$:GOTO 290
BR 330 IF (B$="C" OR B$="c") AND C$="*" T
   HEN LPRINT CHR$(18):C$=" ":POSITION 22
   ,4:? C$:GOTO 290
AW 340 IF (B$="D" OR B$="d") AND D$=" " T
   HEN LPRINT CHR$(27);CHR$(71):D$="*":PO
   SITION 22,5:? D$:GOTO 290
PF 350 IF (B$="D" OR B$="d") AND D$="*" T
   HEN LPRINT CHR$(27);CHR$(72):D$=" ":PO
   SITION 22,5:? D$:GOTO 290
EP 360 IF (B$="E" OR B$="e") AND E$=" " T
   HEN LPRINT CHR$(27);CHR$(69):E$="*":PO
   SITION 22,6:? E$:GOTO 290
WK 370 IF (B$="E" OR B$="e") AND E$="*" T
   HEN LPRINT CHR$(27);CHR$(70):E$=" ":PO
   SITION 22,6:? E$:GOTO 290
XE 380 IF (B$="F" OR B$="f") AND F$="P" T
   HEN LPRINT CHR$(27);CHR$(66);CHR$(2):F
   $="E":POSITION 22,7:? F$:GOTO 290
PT 390 IF (B$="F" OR B$="f") AND F$="E" T
```

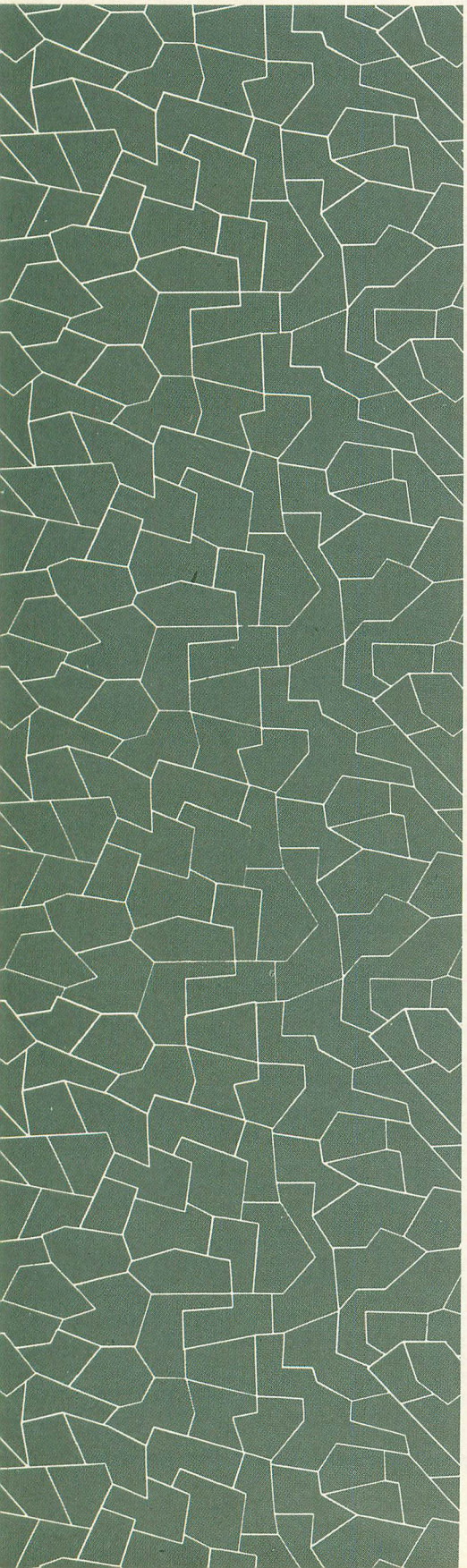
```
   HEN LPRINT CHR$(27);CHR$(66);CHR$(4):F
   $="N":POSITION 22,7:? F$:GOTO 290
YY 400 IF (B$="F" OR B$="f") AND F$="N" T
   HEN LPRINT CHR$(27);CHR$(66);CHR$(1):F
   $="P":POSITION 22,7:? F$:GOTO 290
PW 410 IF B$="G" OR B$="g" THEN 670
XD 420 IF B$="H" OR B$="h" THEN 1090
OM 430 IF (B$="I" OR B$="i") AND I$=" " T
   HEN LPRINT CHR$(27);CHR$(52):I$="*":PO
   SITION 22,10:? I$:GOTO 290
CV 440 IF (B$="I" OR B$="i") AND I$="*" T
   HEN LPRINT CHR$(27);CHR$(53):I$=" ":PO
   SITION 22,10:? I$:GOTO 290
RX 450 IF B$="L" OR B$="l" THEN 720
WS 460 IF (B$="M" OR B$="m") AND M$=" " T
   HEN LPRINT CHR$(27);CHR$(83);CHR$(0):M
   $="*":POSITION 22,12:? M$:GOTO 290
JZ 470 IF (B$="M" OR B$="m") AND M$="*" T
   HEN LPRINT CHR$(27);CHR$(84)
VG 480 IF (B$="M" OR B$="m") AND M$="*" T
   HEN LPRINT CHR$(27);CHR$(72):M$=" ":PO
   SITION 22,12:? M$:GOTO 290
DK 490 IF B$="P" OR B$="p" THEN 770
IY 500 IF B$="Q" OR B$="q" THEN POSITION
   1,22:? "                    ":POSITION 1,2
   1:LPRINT CHR$(12):GRAPHICS 0:END
EF 510 IF B$="R" OR B$="r" THEN 850
AU 520 IF B$="S" OR B$="s" THEN 900
ID 530 IF B$="T" OR B$="t" THEN 950
XI 540 IF (B$="U" OR B$="u") AND U$=" " T
   HEN LPRINT CHR$(27);CHR$(85);CHR$(1):U
   $="*":POSITION 22,18:? U$:GOTO 290
DH 550 IF (B$="U" OR B$="u") AND U$="*" T
   HEN LPRINT CHR$(27);CHR$(85);CHR$(0):U
   $=" ":POSITION 22,18:? U$:GOTO 290
YW 560 IF (B$="W" OR B$="w") AND W$=" " T
   HEN LPRINT CHR$(27);CHR$(87);CHR$(1):W
   $="*":POSITION 22,19:? W$:GOTO 290
EV 570 IF (B$="W" OR B$="w") AND W$="*" T
   HEN LPRINT CHR$(27);CHR$(87);CHR$(0):W
   $=" ":POSITION 22,19:? W$:GOTO 290
UZ 580 IF (B$="Z" OR B$="z") AND Z$=" " T
   HEN LPRINT CHR$(27);CHR$(86);CHR$(1):Z
   $="*":POSITION 22,20:? Z$:GOTO 290
AY 590 IF (B$="Z" OR B$="z") AND Z$="*" T
   HEN LPRINT CHR$(27);CHR$(86);CHR$(0):Z
   $=" ":POSITION 22,20:? Z$:GOTO 290
UI 600 GOTO 290:REM WRONG INPUT
JD 610 REM ** BOTTOM MARGIN **
IW 620 POSITION 4,22:? "NEW bottom margin
         ":POSITION 25,22:INPUT A$
AH 630 LINE=620:GOSUB 1010
OS 640 B=VAL(A$):IF B<0 OR B>16 THEN GOSU
   B 1070:GOTO 620
ID 650 LPRINT CHR$(27);CHR$(78);CHR$(B);:
   POSITION 22,3:? "  ";:POSITION 22,3:?
   B;:GOTO 290
HR 660 REM ** PAGE LENGTH **
ZW 670 POSITION 4,22:? "NEW page length
         ":POSITION 25,22:INPUT A$
CU 680 LINE=670:GOSUB 1010
US 690 G=VAL(A$):IF G<0 OR G>127 THEN GOS
   UB 1070:GOTO 670
KO 700 LPRINT CHR$(27);CHR$(67);CHR$(G):P
   OSITION 22,8:? "   ":POSITION 22,8:? G
   :GOTO 290
MP 710 REM ** LEFT MARGIN **
KE 720 POSITION 4,22:? "NEW left margin
         ":POSITION 25,22:INPUT A$
AS 730 LINE=720:GOSUB 1010
YF 740 L=VAL(A$):IF L<1 OR L>255 THEN GOS
   UB 1070:GOTO 720
FV 750 LPRINT CHR$(27);CHR$(77);CHR$(L):P
   OSITION 22,11:? "   ":POSITION 22,11:?
    L:GOTO 290
DN 760 REM ** PROPORTIONAL **
KM 770 IF PFL=1 THEN LPRINT CHR$(27);CHR$
   (90);CHR$(0):F$=MEM$:POSITION 22,7:? F
   $:POSITION 22,13:? " "
LM 780 IF PFL=1 THEN PFL=0:P$=" ":GOTO 29
   0
ZE 790 IF PFL=0 THEN POSITION 4,22:? "Let
   ter spacing          ":POSITION 25,22:
   INPUT A$
DM 800 LINE=790:GOSUB 1010
WS 810 P=VAL(A$):IF P<1 OR P>6 THEN GOSUB
    1070:GOTO 790
ID 820 LPRINT CHR$(27);CHR$(90):CHR$(P):P
   OSITION 22,7:? " ":MEM$=F$:F$=" "
PH 830 P$=STR$(P):POSITION 22,13:? P:PFL=
   1:GOTO 290
HU 840 REM ** RIGHT MARGIN **
SJ 850 POSITION 4,22:? "NEW right margin
         ":POSITION 25,22:INPUT A$
CQ 860 LINE=850:GOSUB 1010
RR 870 R=VAL(A$):IF R<1 OR R>255 THEN GOS
   UB 1070:GOTO 850
UE 880 LPRINT CHR$(27);CHR$(81);CHR$(R):P
   OSITION 22,15:? "   ":POSITION 22,15:?
    R:GOTO 290
YZ 890 REM ** LINE SPACING **
HL 900 POSITION 4,22:? "NEW line spacing
   X/72     ":POSITION 25,22:INPUT A$
AO 910 LINE=900:GOSUB 1010
PO 920 S=VAL(A$):S$=A$:IF S<1 OR S>127 TH
   EN GOSUB 1070:GOTO 900
FZ 930 LPRINT CHR$(27);CHR$(65);CHR$(S):P
   OSITION 22,16:? "   ":POSITION 22,16:?
    S$:GOTO 290
JZ 940 REM ** TOP LINE **
PG 950 POSITION 4,22:? "NEW top line
         ":POSITION 25,22:INPUT A$
DB 960 LINE=950:GOSUB 1010
IY 970 T=VAL(A$):IF T<1 OR T>16 THEN GOSU
   B 1070:GOTO 950
DC 980 LPRINT CHR$(27);CHR$(82);CHR$(T):P
   OSITION 22,17:? "  ":POSITION 22,17:?
   T:GOTO 290
YX 990 REM * * SUBROUTINES * *
GC 1000 REM ** CHECK FOR NUMBERS **
ID 1010 IF A$="" THEN POP :GOTO LINE
NW 1020 FOR X=1 TO LEN(A$)
KS 1030 IF ASC(A$(X,X))<48 OR ASC(A$(X,X)
   )>57 THEN 1050
MS 1040 NEXT X:RETURN
ZC 1050 POSITION 4,22:? "*  must be numbe
   rs  *      ";:FOR X=1 TO 200:NEXT X:POP
    :GOTO LINE
RN 1060 REM ** ERROR MESSAGE **
KU 1070 POSITION 4,22:? "*   WRONG number
   s  *      ";:FOR X=1 TO 200:NEXT X:RET
   URN
CM 1080 REM ** INSTRUCTIONS **
LS 1090 PRINT CHR$(125):POKE 710,180:POKE
    712,180:POSITION 2,0:? "          HELP
    for  GEMSETUP          "
WH 1100 ? :? "     Sets up GEMINI printer
   ":? "    for Atari 8-bit computer  "
TT 1110 ? :? "1.  Set paper and turn on p
   rinter.":? :? "2.  Using BASIC, run GE
   MSETUP."
TO 1120 ? :? "3.  Enter commands to set t
   he type,":? "    margins, and special
    functions."
VZ 1130 ? "    Re-selecting functions, to
   ggles":? "    them on and off."
NC 1140 ? :? "4.  Set margins last, as th
   ey vary":? "    with each type size."
BT 1150 ? :? "5.  Program performs a form
    feed on":? "    Quit, to realign pape
   r."
MD 1160 ? :? "6.  Default right margin ma
   y be":? "    custom set at line 1280."
GL 1170 POSITION 4,23:? "          push
    <RETURN>          ";:INPUT B$:POKE 710
   ,0:POKE 712,0:GOTO 70
HX 1180 REM ** INITIALIZE **
UO 1190 DIM B$(3)
IG 1200 DIM A$(3),C$(1),D$(1),E$(1),F$(1)
   ,I$(1),M$(1),S$(3),U$(1),MEM$(1),P$(1)
   ,W$(1),Z$(1)
UU 1210 B=0:G=66:L=1:PFL=0:S=12:T=1
LN 1220 C$=" ":D$=" ":E$=" ":F$="P":I$="
   ":M$=" ":S$="12":U$=" ":W$=" ":Z$=" "
KD 1230 POKE 195,0:TRAP 1250
JO 1240 LPRINT CHR$(27);CHR$(64)
SU 1250 IF PEEK(195)=138 THEN ? "K":POSIT
   ION 4,10:POKE 710,36:? "*  *   TURN  O
   N  PRINTER   *  *"
TJ 1260 IF PEEK(195)=138 THEN POSITION 4,
   20:? "   push  <RETURN>  to begin
    ";:INPUT B$:RUN
MV 1270 REM ** DEFAULT RIGHT MARGIN **
FR 1280 R=70:LPRINT CHR$(27);CHR$(81);CHR
   $(R):REM ( change R for new default )
BE 1290 RETURN
```

G a m e

48k disk or cassette

Micro

by Jerry Olejarz



Imagine a cold, dark, damp dungeon. Here are ten prisoners being held by the evil giant *Egsgard.* Your quest? *To rescue them, of course!*

Each captive is being held in a separate cell located in one of the rooms in the dungeon. The entire crypt consists of six levels, each level being a six-by-six square of rooms (that's 216 rooms!). While a cell will be in one room, the key you need to open it will be in another. Each lock has its own key—no other key will fit. Once you have a key, you must try it in every lock until you have a match. Being a puny human, you can only carry one giant key at a time; so it'll take some time before you rescue all the prisoners . . . if you do at all.

Your game is over when you die or when you have rescued all of the prisoners.

Screen Set-up

The upper half of the screen shows the room you presently occupy. Below and to the right of this display is a map of the level you're on (the room you're in is highlighted). Each position on the map represents one room. The symbol displayed shows the contents of that room (for example, if the map shows a monster to the right of the room you're in, then the room to your right contains a monster). An empty room is displayed as a dot on the map. The word *empty* is used loosely, though, for the entire dungeon is infested with the enchanted boulders of Endorra. These stones appear ordinary when observed, but are quick to move when not watched. They cause you no harm, but can get in your way while you travel around the crypt. Be careful in your travels because you never know what could be in a room—even one that was empty before. Monsters, stars and potions often show up in rooms that were empty when you last left them. It is wise to watch the map when passing through doors so you know an instant before you enter a room just what you're getting into.

To the left of the map are the star,

dungeon

key, level and saved indicators. You begin the game with two stars, but you will find more in the dungeon. Below this is the key indicator, which merely shows a key if you have one. The level reading tells which floor of the dungeon you're on, and the saved indicator tells you how many captives you've freed. At the bottom left are the score and bonus readings.

Throughout the dungeon are a number of monsters, left by the giant to guard the prisoners. They stay in their own rooms, but should you happen upon one, it will not hesitate to attack you.

Game Play

To move from room to room, just run through the open door in the desired direction (using joystick in PORT 0). To move from level to level, you must find the room on your level which contains an arrow pointing in the desired direction. This is the teleporter, and it appears pointing either up or down. Just touch it and you will instantly be sent to the next level. Note, however, that the transporters are not necessarily aligned. When you move to a new level, you will likely need to travel a bit to get to the opposite arrow in order to return.

Throughout the dungeon are a number of monsters, left by the giant to guard the prisoners. They stay in their own rooms, but should you happen upon one, it will not hesitate to attack you. If you touch him, you die. Being human, you have only one life; so it is wise to be careful when dealing with the beasts. They can be destroyed using the throwing stars and a little hand-eye coordination. The stars are thrown using the fire button while pointing the joystick in the desired direction. There are more monsters than stars; so don't be too generous with your ammo. Remember to watch the number of stars you have . . . it's not unusual to get killed thinking you have a star when you actually have none.

The crypt also contains a great many magic potions which bring about changes to your score or bonus level. Remember to watch your bonus, for when it runs out, you die.

To pick up any object, maneuver your player to touch it. If you can take it, a tune plays and it is yours; otherwise you hear a buzzing noise. You may carry only one key and seven stars at any one time. Magic potions, which can be drunk at any time, are taken the same way, but they are not carried.

Exercise great care in your adventures, and remember . . . *watch your map!*

Once you have a key and have found a cell, you must try the key in the lock to see if it fits. To do this, stand below the lock and push up on the joystick. If you have the right key, the cell and the key disappear and the prisoner teleports out of the dungeon. If you have the wrong key or no key at all, you hear the buzzer. Your game is over when you die or when you have rescued all of the prisoners.

MICRODUNGEON is a simple game which can be played by almost anyone, yet it takes skill and wisdom to play effectively. If you have any suggestions or any questions on game-play or programming, please write to me c/o *ANALOG Computing.*

Exercise great care in your adventures, and remember . . . *watch your map!*

Jerry Olejarz is 19 and has been programming since he was 14. He plans to study **Computer Sciences** at Waterloo University in Ontario (accompanied, of course, by his 800XL), and is aimed at a career in computer graphics (such as writing video games!!).

Micro dung

Listing 1: BASIC

```
MS 10 REM MICRODUNGEON
KZ 11 REM JERRY OLEJARZ
UI 12 REM FOR ANALOG MAGAZINE
FT 13 REM APR/86
JN 14 REM THANX TO OLSIR,PCS Fiesta
VW 50 DIM M(216),R(216),K(10),C(10),RD$(2
   20),Z$(20),O$(9),B$(80),G$(9),TS1$(24)
   ,TS2$(20)
NM 55 GOSUB 2100
LF 100 O$="→→→→→→→→→":Z$="]]]]]]]]]]]]]]
   ]]]]]":GM=1:GOSUB 2997:POKE 756,M
SR 110 PL=0:PY=0:PX=0:CK=-1:CS=2:SX=9:SY=
   3:SC=0:BL=1000:CO=0:AAB=41:COLOR 130
YD 120 PLOT 0,0:DRAWTO 19,0:PLOT 0,12:DRA
   WTO 19,12:DRAWTO 19,19:DRAWTO 0,19:DRA
   WTO 0,12:PLOT 12,12:DRAWTO 12,19
NE 130 POKE 752,1:? "↰":RESTORE 135:FOR X
   =1 TO 4:READ G$:POSITION 2,13+X:? #6;G
   $:NEXT X
WJ 135 DATA STARS    -,key,LEVEL     r,saved
   ♣"
ZL 140 GOSUB 2950:POSITION 5,4:? #6;"adve
   nturer":POSITION 7,6:? #6;"beware"
VY 155 FOR P=0 TO 216:R(P)=38:M(P)=93:NEX
   T P:M(0)=R(0):GOSUB 2950
LB 156 RESTORE 180:FOR X1=1 TO 7:READ V1,
   V2:V0=0:GW=198:IF V2=35 OR V2=36 THEN
   GW=195:IF V2=36 THEN V0=1
XE 157 FOR X2=V0 TO V1:GOSUB GW:R(X)=V2:I
   F V2=37 THEN K(X2)=X
DF 158 IF V2=33 THEN C(X2)=X
KU 159 NEXT X2:NEXT X1:GOSUB 2930
ID 180 DATA 9,39,19,41,9,37,9,33,5,36,4,3
   5,19,42
CP 190 ? "↰                       ┌────────
   ────────┐":? "  SCORE             |]]]M
   ICRODUNGEON]]]|"
WR 191 ? "  BONUS                |] 1988 ANAL
   OG MAG]|":? "                     └──────
   ──────────┘↓":GOSUB 2990:GOTO 200
GL 195 X=INT(RND(0)*36)+36*X2:IF R(X)<>38
    THEN 195
AH 196 RETURN
ZP 198 X=INT(RND(0)*215)+1:IF R(X)<>38 TH
   EN 198
AQ 199 RETURN
MA 200 RD$="::::::=:::::=:::::::::::=:
   +/::=::::::::::::::;<      ?::::::::::<
          ?:::::::::;<          ?>:::"
IT 205 RD$(101)=":::                :::::<
               :::<                 ?:<
                  ?"
KJ 207 RD$(181)=":::::::::    >:::::::::::
   :::::::::::::"
VP 210 IF PY=0 THEN RD$(30,31)="::"
UF 211 IF PX=0 THEN RD$(84,84)=":":RD$(10
   4,104)="<"
VW 212 IF PX=5 THEN RD$(97,97)=":":RD$(11
   7,117)="?"
OB 213 IF PY=5 THEN RD$(189,192)="::::":R
   D$(209,212)="::::"
NA 214 PR=36*PL+6*PY+PX:Z=INT(RND(0)*5):I
   F Z=0 THEN 216
LU 215 FOR P=1 TO Z:RC=173:GOSUB 310:NEXT
    P
KH 216 IF R(PR)<>38 THEN GOSUB 300
JY 217 IF M(PR)<>R(PR) THEN M(PR)=R(PR):S
   C=SC+2
EK 220 FOR P=0 TO 5:POSITION 13,13+P:FOR
   Q=0 TO 5:? #6;CHR$(M(36*PL+6*P+Q));:NE
   XT Q:NEXT P
QL 221 SOUND 1,200,10,10:LOCATE 13+PX,13+
   PY,Z:COLOR Z-32:PLOT 13+PX,13+PY:SOUND
    1,0,0,0
VN 222 AAA=AAA+1:IF AAA>2 THEN AAA=0:GOSU
   B 198:AAB=80-AAB:R(X)=AAB
DB 225 FOR P=0 TO 10:POSITION 0,P+1:? #6;
   Z$:NEXT P
VE 226 FOR P=10 TO 0 STEP -1:POSITION 0,P
   +1:? #6;RD$(P*20+1,P*20+20):NEXT P
UQ 227 COLOR PL+17:PLOT 10,16:COLOR (CK>-
   1)*5:PLOT 10,15:COLOR CS+16:PLOT 10,14
   :GOSUB 2990
WR 250 COLOR 8:PLOT SX,SY:IF PEEK(77)>0 T
   HEN POKE 77,0:BL=BL-1:GOSUB 2990:IF BL
   <1 THEN 500
QP 255 J=STICK(0):IF J=15 THEN 270
DR 260 PSX=SX+(J=6)+(J=7)+(J=5)-(J=10)-(J
   =11)-(J=9)
SL 261 PSY=SY+(J=9)+(J=13)+(J=5)-(J=10)-(
   J=14)-(J=6):IF STRIG(0)=0 THEN 370
MS 262 LOCATE PSX,PSY,PZ:IF PZ<>32 THEN 3
   50
FI 265 SOUND 1,250,10,10:COLOR 32:PLOT SX
   ,SY:SX=PSX:SY=PSY:COLOR 8:PLOT SX,SY:S
   OUND 1,0,0,0
TL 270 IF R(PR)=41 AND MM>4 THEN 274
SR 273 MM=MM+1:GOTO 250
YI 274 MM=0:POX=OX+(OX<SX)-(OX>SX):POY=OY
   +(OY<SY)-(OY>SY)
YZ 276 LOCATE POX,POY,W:IF W=8 THEN 480
MT 277 IF W<>32 THEN 250
BA 278 COLOR 32:PLOT OX,OY:OX=POX:OY=POY:
   COLOR 41:PLOT OX,OY:GOTO 250
MD 300 RC=R(PR):IF RC=33 THEN 320
ZU 310 MM=0:OX=7+INT(RND(0)*6):OY=4+INT(R
   ND(0)*5):PS=(OY-1)*20+OX+1:RD$(PS,PS)=
   CHR$(RC):RETURN
RH 320 RD$(106,109)="[=:;":RD$(125,129)="
   [<■[<":RD$(145,148)=",,!<":RETURN
BS 350 IF PZ>32 AND PZ<43 THEN GOTO ((PZ*
   10)+70)
UR 355 WZ=(SX=9 OR SX=10):IF WZ AND PSY=2
    AND PZ<>186 THEN SY=SY+7:PY=PY-1:GOTO
    200
YX 356 IF WZ AND PSY=11 THEN SY=SY-7:PY=P
   Y+1:GOTO 200
WB 357 IF (SX=16 AND SY=6) AND PSX=17 THE
   N SX=SX-13:PX=PX+1:GOTO 200
JT 358 IF (SX=3 AND SY=6) AND PSX=2 THEN
   SX=SX+13:PX=PX-1:GOTO 200
PD 360 GOTO 270
VD 370 IF CS=0 THEN 250
VI 372 CS=CS-1:COLOR CS+16:PLOT 10,14
CD 373 T=0:XA=PSX-SX:YA=PSY-SY:TX=SX:TY=S
   Y
NC 374 LOCATE TX+XA,TY+YA,W:T=T+1
XE 375 COLOR 32:IF  NOT (TX=SX AND TY=SY)
    THEN PLOT TX,TY
PS 376 IF W<>32 THEN 380
UF 377 SOUND 0,10*T,10,10:TX=TX+XA:TY=TY+
   YA:COLOR 39:PLOT TX,TY
LD 378 FOR DD=1 TO 10:NEXT DD:SOUND 0,0,0
   ,0:GOTO 374
BS 380 IF W<>41 THEN GOSUB 2980:GOTO 250
PH 385 PLOT OX,OY:R(PR)=38:M(PR)=38:COLOR
    6:PLOT 13+PX,13+PY:COLOR 0:GOSUB 2970
   :SC=SC+100:GOSUB 2990:GOTO 250
HS 400 P=-1:IF CK<0 THEN GOSUB 2980:GOTO
   250
JT 404 P=P+1:IF C(P)=PR THEN 408
PB 406 GOTO 404
SW 408 IF CK<>P THEN GOSUB 2980:GOTO 250
BK 410 RD$(106,109)="    ":RD$(125,129)="
    ■    ":RD$(145,148)="    ":POSITION 5,
   6:? #6;RD$(106,148)
QV 412 CK=-1:R(PR)=38:M(PR)=38:COLOR 6:PL
   OT 13+PX,13+PY:COLOR 0:PLOT 10,15:GOSU
   B 2970
IT 414 W=CHSET+112:FOR P=7 TO 0 STEP -1:P
   OKE W+P,0:SOUND 1,30*P,10,2*P:SOUND 0,
```

```
   15*P,10,P:FOR Q=1 TO 20:NEXT Q:NEXT P
BX 416 COLOR 32:PLOT 6,7:FOR P=0 TO 7:POK
   E W+P,PEEK(W-48+P):NEXT P:SC=SC+200:GO
   SUB 2990:CO=CO+1:IF CO=10 THEN 500
MY 417 COLOR CO+16:PLOT 10,17
OW 418 GOTO 250
XW 420 PL=PL+1
AQ 424 SX=9:SY=3:I=10*(PZ-35.5):FOR P=150
   *(I<0) TO 150*(I>0) STEP I:SOUND 0,P,1
   0,10:NEXT P:SOUND 0,0,0,0:GOTO 200
WB 430 PL=PL-1:GOTO 424
UX 440 P=-1:IF CK>-1 THEN GOSUB 2980:GOTO
    250
SR 444 P=P+1:IF K(P)=PR THEN 448
RB 446 GOTO 444
SG 448 CK=P:COLOR 5:PLOT 10,15
XW 450 M(PR)=38:R(PR)=38:SC=SC+50:GOSUB 2
   990:COLOR 32:PLOT OX,OY:COLOR 6:PLOT 1
   3+PX,13+PY:GOSUB 2970:GOTO 250
AC 460 IF CS>6 THEN GOSUB 2980:GOTO 250
PB 464 CS=CS+1:COLOR CS+16:PLOT 10,14:GOT
   O 450
EV 480 GOSUB 2980:GOTO 500
BN 490 Z=INT(RND(0)*10)
SO 491 IF Z<3 THEN SC=SC-INT(RND(0)*50):G
   OTO 450
OO 492 IF Z>3 AND Z<7 THEN Z=5*SGN(Z-5):S
   C=SC-50:BL=INT((Z-2)*BL/(Z-3)):GOTO 45
   0
NP 494 SC=SC-(50+INT(RND(0)*50)):IF SC<-5
   0 THEN SC=-50
PX 495 GOTO 450
HA 500 FOR P=0 TO 17:POSITION 0,P+1:? #6;
   Z$:NEXT P
KF 520 IF CO<10 THEN 530
GB 522 BL=BL-5:SC=SC+5:GOSUB 2990:SOUND 0
   ,BL/4,10,2:IF BL<5 AND BL>0 THEN SC=SC
   +BL-5:BL=5:GOTO 522
UG 523 IF BL>0 THEN 522
FI 525 POSITION 2,3:? #6;"congratulations
   ":POSITION 3,5:? #6;"you succeeded":GO
   TO 540
TR 530 POSITION 3,3:? #6;"your quest has"
   :POSITION 2,5:? #6;"ended in failure"
UP 540 POSITION 8,9:? #6;"game":POSITION
   8,10:? #6;"over"
UP 542 POSITION 2,14:? #6;"hit START key
   to":POSITION 4,16:? #6;"restart game"
YW 550 ? "  SCORE  ";SC:? "
   ";
BD 580 IF PEEK(53279)=7 AND STRIG(0)=1 TH
   EN GOSUB 2950:GOTO 580
MC 590 GOTO 100
ZJ 2100 M=PEEK(106)-8:POKE 106,M-1:CHSET=
   M*256:GM=0:GOSUB 2997:GOSUB 2200
IW 2102 RESTORE 2110
FO 2105 FOR P=1 TO 32:READ X:B$(P,P)=CHR$
   (X):NEXT P
ER 2110 DATA 104,104,133,213,104,133,212
NV 2112 DATA 104,133,215,104,133,214,162
JW 2114 DATA 4,160,0,177,212,145,214
PE 2116 DATA 200,208,249,230,213,230,215
SG 2118 DATA 202,208,240,96
QF 2120 Z=USR(ADR(B$),224*256,CHSET)
HH 2130 POKE M-1,0:RESTORE 2150
QX 2135 READ C:IF C>0 THEN FOR Q=0 TO 7:R
   EAD A:POKE (CHSET+C*8+Q),A:NEXT Q:GOTO
    2135
HE 2140 GOSUB 2250:RETURN
FB 2150 DATA 2,255,129,189,165,165,189,12
   9,255
EY 2151 DATA 5,28,20,28,8,8,24,8,24
HL 2152 DATA 9,62,127,73,91,127,127,127,8
   5
SW 2154 DATA 8,24,24,60,90,24,36,36,36
VK 2156 DATA 7,0,8,16,92,58,8,16,0
AZ 2158 DATA 26,255,255,255,255,255,255,2
   55,255
DX 2160 DATA 27,254,252,250,246,238,222,1
   90,126
QV 2162 DATA 28,254,252,248,240,224,192,1
   28,0
KY 2164 DATA 29,127,127,127,127,127,127,1
   27,127
ML 2166 DATA 30,127,63,95,111,119,123,125
   ,126
UN 2168 DATA 31,127,63,31,15,7,3,1,0
HT 2170 DATA 12,0,255,255,255,255,255,255
   ,255
LD 2172 DATA 11,255,128,176,176,160,160,1
   28,128
HF 2174 DATA 15,255,1,13,13,5,5,1,1
IV 2176 DATA 59,0,1,3,7,15,31,63,127
VM 2178 DATA 60,0,128,192,224,240,248,252
   ,254
GM 2180 DATA 61,170,85,170,85,170,85,170,
   85
DG 2182 DATA 6,0,0,0,24,24,0,0,0
GZ 2184 DATA 3,0,8,28,62,8,8,8,0
HQ 2186 DATA 4,0,8,8,8,62,28,8,0
XM 2188 DATA 1,28,54,34,127,119,119,62,0
HD 2190 DATA 14,24,24,60,90,24,36,36,36
KJ 2192 DATA 10,0,8,8,8,28,54,54,28
NF 2194 DATA 13,0,0,14,63,127,253,243,126
LJ 2199 DATA 0
OL 2200 Z=PEEK(560)+256*PEEK(561):POKE Z+
   3,66:POKE Z+11,7:POKE Z+12,6
GB 2210 POKE Z+23,6:POKE Z+24,6:POKE Z+25
   ,6:POKE Z+26,65:POKE Z+27,PEEK(560):PO
   KE Z+28,PEEK(561)
RX 2215 COLOR ASC("_"):PLOT 8,5:DRAWTO 31
   ,5:PLOT 8,8:DRAWTO 32,8
QP 2220 RESTORE 2225:FOR Z=1 TO 6:READ X,
   Y,TS1$:POSITION X,Y:? TS1$;:NEXT Z
RQ 2225 DATA 12,2,ANALOG COMPUTING,16,3,P
   resents,4,6,Microdungeon,12,7,By Jerry
    Olejarz
UH 2227 DATA 8,13,INITIALIZING EVERYTHING
   -,13,14,PLEASE WAIT...
AN 2230 RETURN
OT 2250 TS1$=" HIT fire BUTTON TO ":TS2$=
   "   START THE GAME   "
IT 2252 COLOR 32:PLOT 8,13:DRAWTO 31,13:P
   LOT 26,14:DRAWTO 13,14:POKE 756,M
PV 2255 FOR X=0 TO 19:POSITION 19-X,17:?
   TS1$(1,X+1):POSITION 0,18:? TS2$(20-X,
   20):IF STRIG(0)=0 THEN RETURN
QK 2257 SOUND 0,10*X,8,4:SOUND 1,10*(19-X
   ),8,4:NEXT X:KA=-2:K=8
YT 2260 IF STRIG(0)=0 THEN RETURN
WA 2265 K=K+KA:IF K<3 OR K>15 THEN KA=-KA
   :C=INT(RND(0)*16):GOSUB 2950:GOTO 2265
XR 2267 SETCOLOR 3,C,K:IF STRIG(0)=0 THEN
    RETURN
TA 2268 GOTO 2260
UK 2900 REM SMALL SUBS
HN 2930 FOR P=0 TO 3:SOUND P,0,0,0:NEXT P
   :RETURN
QY 2950 FOR P=0 TO 2:RESTORE 2955:Z=INT(R
   ND(0)*7):FOR Q=1 TO Z:READ G:NEXT Q:RE
   AD N
NN 2952 SOUND P,N,10,1:NEXT P:RETURN
SG 2955 DATA 243,193,162,121,96,81,60,47,
   40
EM 2960 GOSUB 198:AAB=80-AAB:R(X)=AAB:RET
   URN
FB 2970 RESTORE 2975:FOR P=0 TO 8:READ N:
   SOUND 0,N,10,8:FOR Q=1 TO 9:NEXT Q:NEX
   T P:RETURN
RX 2975 DATA 251,217,193,162,162,193,162,
   162,0
IT 2980 FOR V=15 TO 0 STEP -1:SOUND 0,255
   -10*V,10,V:SOUND 1,255-10*V,8,V/2:NEXT
    V:RETURN
LZ 2990 ? O$;SC;" ":? O$;BL;" ":? "↓":RET
   URN
RC 2997 GRAPHICS GM:IF GM<>1 THEN SETCOLO
   R 2,0,0:GOTO 2999
LL 2998 SETCOLOR 2,0,4
TP 2999 SETCOLOR 1,4,14:SETCOLOR 3,12,8:S
   ETCOLOR 0,8,6:POKE 82,0:POKE 752,1:? "
   R":RETURN
```

Character Transfer Utility

by Jim V. Bowles and Jenni M. Bowles

Yes, *Atari* fans, yet another accessory for the famous *ANALOG Create-A-Font* (Issue 22). This utility will allow you to move characters or blocks of characters between fonts, without resorting to the tedious task of recreating each character everywhere you wish to use it. And it'll work not only with Create-A-Font, but with any other font editor that saves its data in the conventional way.

Using the Program

Type in the program using Basic Editor II, then save it. Once you're up and running you'd better have a disk full of fonts ready to roll.

When you run the program, you will be asked for the filename of your Base Font. This is the font you wish to modify. If you want to start fresh, press RETURN and the standard Atari font will be used. If you need to see the disk's directory, press D.

Next, you can press T to load a Transfer Font (the font you'll be moving characters from). All transfers will be from the Transfer Font to the Base Font.

Briefly, the function menu is as follows. . .

Gr.1,2 — toggles between full and half font saves.

Base Font — allows you to view the base font at any time.

Save — asks for filename, then saves the Base Font to disk.

Dir — lists the directory of

drive one.

Transfer — starts the transfer sequence. First you choose between a block transfer or segment transfer. A block transfer moves complete rows of characters (full block) or half rows (half blocks). Full or half blocks can be transferred to any other full or half block positions. Segment transfer is for moving other than full or half blocks. To choose a segment, use the up and down arrows to select the main row to transfer from, and then, using the left and right arrows, position the left pointer to the leftmost end of the segment and the right pointer to the rightmost end. Note that the segment transferred is between the pointers only. Hit RETURN and choose the segment of the Base Font to move to. Press RETURN again, and the transfer will take place.

Load new — will load a new Transfer or Base Font. Files can be loaded from multiple disk drives, but if you don't specify a device name, D1: is assumed (the same with Save). You can escape a load or save (in case you change your mind or want to look at the directory) by hitting RETURN at the prompt.

Clear — will load the standard Atari font into either the Transfer or Base Fonts.

Quit — returns you to BASIC

Modifications

There's some room at the bottom of the function menu for any added features you might want to include for your own purposes. The program is laid out very simply, and modifications should be easy to add since the main section just uses GOSUBs to access a variety of small routines. Here are a few suggestions to help customize Character Transfer Utility to your own needs.

If you have BASIC XL/XE, the font loading and saving can be sped up tremendously by using

```
530 OPEN #2,4,0,FILE$:BGET
#2,CHB,1024: CLOSE #2:
RETURN
```

in place of lines 530 and 540 to load a font, and

```
560 OPEN
#2,8,0,FILE$:BPUT #2,NCB1,SV:
CLOSE #2:RETURN
```

in place of lines 560 and 570 to save a font. A POKE 54286,64 before will turn the screen a solid color (eliminate the flashing), and a POKE 54286,192 after will restore the display to normal.

In the Transfer routine, the segment pointers can be made to remain in their last positions between transfers (they are reset to the ends normally) by deleting the HP1=3 and HP2=36 in line 1870.

The Directory function, as is, lists all programs on the disk. All my fonts have the filename extender .FNT. If you keep track of your files using this method, the "D:*.*" at line 2220 can be changed to "D:*.FNT" to list only the font files.

Jim V. Bowles has an **Associate in Science Degree** and repairs video cameras and camcorders by day. In the evening he likes to adventure, destroy aliens, program his Atari, telecommunicate or read ANALOG. He has had his 800 XL for four years and his 130 XE for 2½ years, and still thinks it's the best deal he ever got.

Character Transfer Utility

Listing 1: BASIC

```
BG 10 REM CHRXFR.BAS -- ANALOG COMPUTING
GT 20 DIM FILE$(15),FN$(15),D$(2),B$(4),T
   $(8),NA$(8),DR$(17),ML$(20)
HE 30 DIM BL(4)
HG 40 D$="D:":B$="Base":T$="Transfer"
NK 50 VP=8:BL(0)=512:BL(1)=0:BL(2)=256:BL
   (3)=768:SV=1023
WY 60 SEG=0:HP1=3:HP2=36:ST1=0:ST2=0:BLK1
   =0:BLK2=0:STAD1=0:STAD2=0:ADV=0:LN=0
MV 70 L=0:R=0:CHB=0:HP3=2:VP2=17:CH=0:C=0
   :PIT=0:S=0
CX 80 RT=PEEK(106):NC1=RT-8:NC2=RT-12:NCB
   1=NC1*256:NCB2=NC2*256
RN 90 POKE 204,NC1:POKE 206,224
UQ 100 FOR Z=1 TO 20:READ A:ML$(Z,Z)=CHR$
   (A):NEXT Z
TW 110 DATA 104,162,4,160,0,177,205,145,2
   03,200,208,249,230,206,230,204,202,208
   ,242,96
TA 120 ? "K":POKE 559,0:POKE 752,1:POKE 7
   10,80
DE 130 POKE 1024,NC2:REM SET AT XFR FONT
EB 140 FOR Z=0 TO 200
SQ 150 READ A
JG 160 IF A=-1 THEN 190
BE 170 POKE 1536+Z,A
MV 180 NEXT Z
EJ 190 DLSTART=PEEK(560)+PEEK(561)*256
VJ 200 POKE DLSTART+8,130
HU 210 POKE DLSTART+11,130
MG 220 POKE DLSTART+17,130
IQ 230 POKE DLSTART+21,130
LQ 240 POKE DLSTART+25,130
EE 250 FOR Z=0 TO 200
ST 260 READ A
EE 270 IF A=-1 THEN 300
BA 280 POKE 1680+Z,A
MY 290 NEXT Z
BD 300 Q=USR(1680)
OH 310 POKE 54286,192
PD 320 DATA 72,169,6,141,10,212,141,24,20
   8,169
GC 330 DATA 0,141,23,208,169,26,141,0,2,1
   69
NR 340 DATA 6,141,1,2,104,64,72,173,0,4
TO 350 DATA 141,10,212,141,9,212,169,0,14
   1,24
NZ 360 DATA 208,169,10,141,23,208,169,58,
   141,0
SU 370 DATA 2,169,6,141,1,2,104,64,72,169
SZ 380 DATA 224,141,10,212,141,9,212,169,
   144,141
KL 390 DATA 24,208,169,10,141,23,208,169,
   89,141
SJ 400 DATA 0,2,169,6,141,1,2,104,64,72
KI 410 DATA 169,164,141,10,212,141,24,208
   ,169,0
FL 420 DATA 141,23,208,169,115,141,0,2,16
   9,6
YI 430 DATA 141,1,2,104,64,72,169,80,141,
   10
MQ 440 DATA 212,141,24,208,169,10,141,23,
   208,104
CH 450 DATA 64,-1
BW 460 DATA 104,160,176,162,6,169,7,32,92
EK 470 DATA 228,96,0,0,0,0,0,0,0,0
KS 480 DATA 0,0,0,0,0,0,0,0,0,0
LU 490 DATA 0,0,0,169,0,141,0,2,169,6
FN 500 DATA 141,1,2,76,98,228,-1
NE 510 GOTO 1080
WO 520 C=INT((40-(14+LEN(FILE$)))/2):GOSU
   B 630:POSITION C,18:? "** Getting ";FI
   LE$;" **"
ZR 530 POKE 54286,64:OPEN #2,4,0,FILE$:PO
   KE 54286,192:FOR Z=CHB TO CHB+1023
AT 540 GET #2,CH:POKE Z,CH:NEXT Z:CLOSE #
   2:RETURN
RJ 550 C=INT((40-(13+LEN(FILE$)))/2):POSI
   TION C,18:? "** Saving ";FILE$;" **"
QZ 560 POKE 54286,64:OPEN #2,8,0,FILE$:PO
   KE 54286,192:FOR Z=NCB1 TO NCB1+SV
DC 570 PUT #2,PEEK(Z):NEXT Z:POKE 54286,6
   4:CLOSE #2:POKE 54286,192:GOSUB 630:RE
   TURN
QT 580 ADV=0:STAD1=NCB1+BLK1+ST1:STAD2=NC
   B2+BLK2+ST2
PA 590 FOR Z=STAD2 TO STAD2+LN
ME 600 POKE STAD1+ADV,PEEK(Z):ADV=ADV+1
TC 610 NEXT Z:RETURN
ZA 620 CLOSE #1:OPEN #1,4,0,"K:":GET #1,K
   :CLOSE #1:RETURN
LP 630 FOR BL=17 TO 18:POSITION 4,BL:? "
                                      ":NEXT
    BL:RETURN
HN 640 FOR BL=19 TO 20:POSITION 4,BL:? "
                                      ":NEXT
    BL:RETURN
FZ 650 POSITION 1,VP:? "  ":RETURN
GZ 660 POSITION 1,VP:? VP-7;CHR$(127):RET
   URN
IB 670 POSITION HP1,7:? CHR$(255):POSITIO
   N HP1,12:? CHR$(255):RETURN
HC 680 POSITION HP2,7:? CHR$(254):POSITIO
   N HP2,12:? CHR$(254):RETURN
EP 690 POSITION HP1,7:? " ":POSITION HP1,
   12:? " ":RETURN
FZ 700 POSITION HP2,7:? " ":POSITION HP2,
   12:? " ":RETURN
HU 710 IF VP=8 AND K=45 THEN GOSUB 650:VP
   =11:GOSUB 660:RETURN
CJ 720 IF VP=11 AND K=61 THEN GOSUB 650:V
   P=8:GOSUB 660:RETURN
NH 730 IF K=61 THEN GOSUB 650:VP=VP+1:GOS
   UB 660:RETURN
QV 740 IF K=45 THEN GOSUB 650:VP=VP-1:GOS
   UB 660:RETURN
ZN 750 RETURN
FS 760 IF K=76 THEN L=1:R=0:RETURN
DO 770 IF K=82 THEN R=1:L=0:RETURN
FV 780 IF K=42 AND R=1 AND HP2=36 THEN RE
   TURN
KM 790 IF K=43 AND L=1 AND HP1=3 THEN RET
   URN
QQ 800 IF K=42 AND L=1 AND HP1=HP2-1 THEN
    RETURN
SL 810 IF K=43 AND R=1 AND HP2=HP1+1 THEN
    RETURN
DC 820 IF K=43 AND R=1 THEN GOSUB 700:HP2
   =HP2-1:GOSUB 680:RETURN
ZL 830 IF K=42 AND R=1 THEN GOSUB 700:HP2
   =HP2+1:GOSUB 680:RETURN
EK 840 IF K=43 AND L=1 THEN GOSUB 690:HP1
   =HP1-1:GOSUB 670:RETURN
AT 850 IF K=42 AND L=1 THEN GOSUB 690:HP1
```

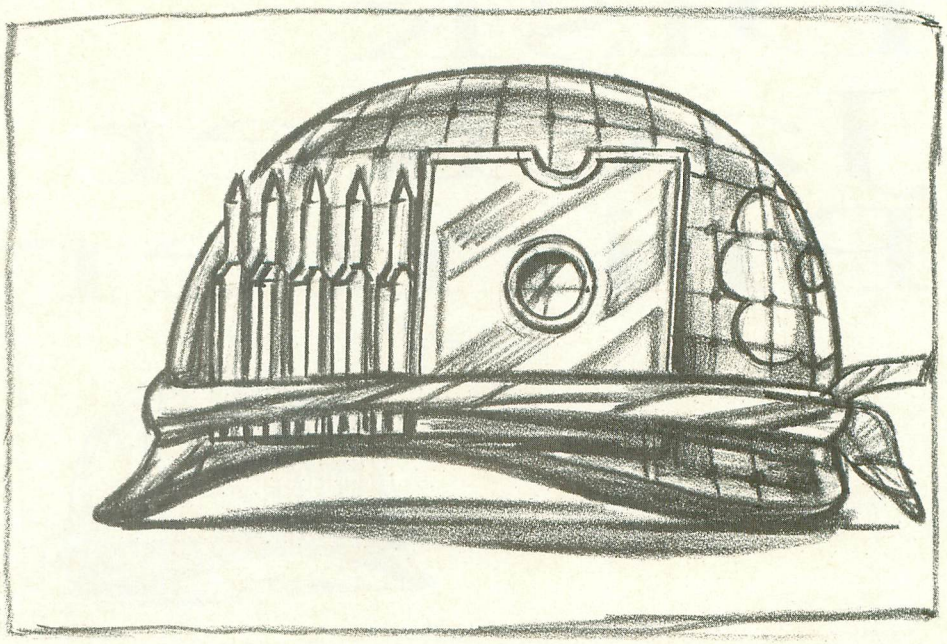
```
   =HP1+1:GOSUB 670:RETURN
ZQ 860 RETURN
UX 870 IF K=43 AND HP1=3 THEN RETURN
VO 880 IF K=42 AND HP2=36 THEN RETURN
YC 890 IF K=43 THEN GOSUB 690:GOSUB 700:H
   P1=HP1-1:HP2=HP2-1:GOSUB 670:GOSUB 680
   :RETURN
PI 900 IF K=42 THEN GOSUB 690:GOSUB 700:H
   P1=HP1+1:HP2=HP2+1:GOSUB 670:GOSUB 680
   :RETURN
ZH 910 RETURN
ZM 920 PIT=30:FOR S=4 TO 14 STEP 2
WG 930 SOUND 0,PIT,10,5:SOUND 1,PIT+10,10
   ,5
DM 940 PIT=PIT-5:NEXT S:SOUND 0,0,0,0:SOU
   ND 1,0,0,0:RETURN
BZ 950 POSITION 12,20:? "* Hit any Key *"
   :GOSUB 620:RETURN
JT 960 GOSUB 630:POSITION 10,18:? "Select
    Main Block..."
TU 970 POSITION 11,20:? "Use ";CHR$(28);C
   HR$(29);" Then RETURN":RETURN
DQ 980 POSITION 6,17:? "Enter ";NA$;" Fon
   t File Name":POSITION 6,18:? "Here..."
   ;:INPUT FN$:RETURN
BB 990 IF FN$(2,2)=":" OR FN$(3,3)=":" TH
   EN FILE$=FN$:RETURN
TA 1000 FILE$=D$:FILE$(3)=FN$:RETURN
XU 1010 IF SV=1023 THEN SV=511:POSITION 7
   ,22:? "Half":RETURN
IT 1020 IF SV=511 THEN SV=1023:POSITION 7
   ,22:? "Full":RETURN
XA 1030 CLOSE #2:POKE 54286,192:GOSUB 630
   :POSITION 9,18:? "...Font Not Found...
   ":GOSUB 950
KD 1040 POKE 766,1:GOSUB 630:GOSUB 640:RE
   TURN
FL 1050 GOSUB 1030:POKE 766,0:GOTO 1230
FQ 1060 GOSUB 1030:POKE 766,0:GOTO 1050
AB 1070 GOSUB 1030:GOTO 1290
DP 1080 POSITION 4,1:? "CHARACTER BLOCK T
   RANSFER UTILITY"
HV 1090 POSITION 13,3:? "Function Menu"
CO 1100 POSITION 3,4:? "GR.1,2     BASE F
   ONT    SAVE    DIR"
VO 1110 POSITION 3,5:? "TRANSFER   LOAD N
   EW    CLEAR   QUIT"
KE 1120 POSITION 15,16:? "Messages.."
YS 1130 POKE 766,1:H=4:V=8:FOR Z=0 TO 127
ZN 1140 IF H>35 THEN H=4:V=V+1
ME 1150 POSITION H,V:? CHR$(Z):H=H+1:NEXT
    Z:POKE 766,0
JH 1160 POSITION 6,17:? "Do you wish to s
   tart with a"
WG 1170 POSITION 2,22:? "Save=Full Font
    (C)1988 A.N.A.L.O.G"
QW 1180 POKE 204,NC2:POKE 206,224
KZ 1190 Q=USR(ADR(ML$))
YT 1200 POSITION 6,18:? "base font?":POKE
    559,34:GOSUB 620
YT 1210 GOSUB 630
AY 1220 IF K<>89 THEN 1260
PF 1230 TRAP 1050:NA$=B$:GOSUB 980
AI 1240 IF FN$="" THEN 1260
AB 1250 GOSUB 990:CHB=NCB1:GOSUB 520:GOSU
   B 920:GOTO 1280
QB 1260 POKE 204,NC1:POKE 206,224
KV 1270 Q=USR(ADR(ML$))
NX 1280 GOSUB 630:POKE 1024,NC2:POSITION
   13,13:? "transfer font":POKE 766,1
UN 1290 POSITION 6,18:? "  Please Select
   Function    ":GOSUB 620
ET 1300 IF K=66 THEN 1880
VG 1310 IF K=84 THEN 1400
AN 1320 IF K=76 THEN 1910
VX 1330 IF K=83 THEN 2040
BB 1340 IF K=67 THEN 2090
ES 1350 IF K=71 THEN GOSUB 1010:GOTO 1290
EV 1360 IF K=81 THEN GRAPHICS 0:END
XT 1370 IF K=68 THEN 2220
FB 1380 IF K=81 THEN GRAPHICS 0:END
TG 1390 GOTO 1290
SK 1400 GOSUB 660:POSITION 6,13:? "from..
   ."
YM 1410 GOSUB 630:POSITION 11,18:? "BLOCK
    or SEGMENT":GOSUB 620
ZE 1420 IF K=83 THEN SEG=1:GOSUB 670:GOSU
   B 680:GOTO 1450
FC 1430 IF K=66 THEN SEG=0:GOTO 1450
PR 1440 GOTO 1410
CG 1450 GOSUB 960
YZ 1460 GOSUB 620
MV 1470 GOSUB 710:BLK2=BL(VP-8)
EK 1480 IF K=155 AND SEG=1 THEN 1590
TU 1490 IF K=155 AND SEG=0 THEN 1510
RU 1500 GOTO 1460
EK 1510 GOSUB 640:POSITION 6,18:? "  Half
    Block or Full Block  ":GOSUB 620
YZ 1520 IF K=72 THEN 1550
IU 1530 IF K=70 THEN ST2=0:LN=255:GOTO 16
   60
QF 1540 GOTO 1510
MH 1550 POSITION 6,18:? "  First Half or
   Second Half ":GOSUB 620
HA 1560 IF K=70 THEN ST2=0:LN=127:GOTO 16
   60
PA 1570 IF K=83 THEN ST2=128:LN=ST2-1:GOT
   O 1660
SR 1580 GOTO 1550
XJ 1590 GOSUB 630:GOSUB 640:POSITION 6,17
   :? "Choose Left or Right pointer"
ER 1600 POSITION 9,19:? "Use ";CHR$(30);C
   HR$(31);" Then Hit RETURN":L=1
YO 1610 GOSUB 620
CZ 1620 IF K=155 AND (HP2-(HP1+1))=0 THEN
    GOSUB 630:GOSUB 640:GOTO 1290
JW 1630 IF K=155 THEN L=1:R=0:ST2=(HP1-3)
   *8:LN=((HP2-(HP1+1))*8)-1:GOTO 1660
BJ 1640 GOSUB 760
QW 1650 GOTO 1610
TT 1660 POKE 1024,NC1:POSITION 6,13:? "to
   .....  base font  "
ZN 1670 GOSUB 630:GOSUB 640:GOSUB 960
ZJ 1680 GOSUB 620
MM 1690 GOSUB 710:BLK1=BL(VP-8)
XV 1700 IF K=155 AND SEG=0 THEN 1730
EZ 1710 IF K=155 AND SEG=1 THEN 1780
UC 1720 GOTO 1680
JR 1730 IF LN=255 THEN ST1=0:GOTO 1840
MI 1740 POSITION 6,18:? "  First Half or
   Second Half ":GOSUB 620
AN 1750 IF K=70 THEN ST1=0:GOTO 1840
UC 1760 IF K=83 THEN ST1=128:GOTO 1840
TD 1770 GOTO 1740
TJ 1780 POSITION 10,18:? "Move Segment wi
   th ";CHR$(30);CHR$(31)
VB 1790 GOSUB 640:POSITION 12,20:? "Then
   Hit RETURN"
YP 1800 GOSUB 620
CW 1810 IF K=155 THEN L=0:R=0:ST1=(HP1-3)
   *8:GOTO 1840
CG 1820 GOSUB 870
RF 1830 GOTO 1800
BP 1840 GOSUB 580
IL 1850 GOSUB 630:GOSUB 640:POSITION 11,1
```

```
    8:? "TRANSFER COMPLETE..":GOSUB 950:GO
    SUB 640
LK  1860 GOSUB 690:GOSUB 700:POSITION 1,VP
    :? "  ":POSITION 6,13:? "          "
ZF  1870 HP1=3:HP2=36:POKE 1024,NC2:POSITI
    ON 13,13:? "transfer font":GOTO 1290
YA  1880 POKE 1024,NC1:POSITION 13,13:? "
     base font  ":POSITION 6,18:? "This is
     your new BASE FONT.."
CN  1890 GOSUB 950
DC  1900 GOSUB 640:POKE 1024,NC2:POSITION
    13,13:? "transfer font":GOTO 1290
WU  1910 POKE 766,0:POSITION 8,18:? "Base
    Font or Transfer Font"
YX  1920 GOSUB 620
VV  1930 IF K=66 THEN GOSUB 630:GOTO 1960
CT  1940 IF K=84 THEN GOSUB 630:GOTO 2000
SZ  1950 GOTO 1920
RC  1960 TRAP 1070:NA$=B$:GOSUB 980
VZ  1970 IF FN$="" THEN GOSUB 630:GOSUB 64
    0:POKE 766,1:GOTO 1290
XZ  1980 GOSUB 990:CHB=NCB1:GOSUB 520:GOSU
    B 920
JN  1990 POKE 766,1:GOSUB 630:GOTO 1290
DP  2000 TRAP 1070:NA$=T$:GOSUB 980
UQ  2010 IF FN$="" THEN GOSUB 630:GOSUB 64
    0:POKE 766,1:GOTO 1290
XN  2020 GOSUB 990:CHB=NCB2:GOSUB 520:GOSU
    B 920
IE  2030 POKE 766,1:GOSUB 630:GOTO 1290
OV  2040 POKE 766,0:GOSUB 630:POSITION 6,1
    7:? "Enter NEW FONT File Name":POSITIO
    N 6,18:? "Here..";:INPUT FN$
VC  2050 IF FN$="" THEN GOSUB 630:GOSUB 64
    0:POKE 766,1:GOTO 1290
FX  2060 GOSUB 990:GOSUB 630:GOSUB 550:GOS
    UB 920
EB  2070 POSITION 12,18:? "Save Complete..
    ":GOSUB 950
GX  2080 POKE 766,1:GOSUB 630:GOSUB 640:GO
    TO 1290
FK  2090 POSITION 8,18:? "Base Font or Tra
    nsfer Font"
YC  2100 GOSUB 620
IU  2110 IF K=66 THEN GOSUB 630:GOTO 2140
PT  2120 IF K=84 THEN GOSUB 630:GOTO 2190
NX  2130 GOTO 2100
GV  2140 POSITION 9,17:? "Your BASE FONT i
    s now":POSITION 12,18:? "ATARI STANDAR
    D"
PX  2150 POKE 204,NC1:POKE 206,224
KR  2160 Q=USR(ADR(ML$))
RU  2170 GOSUB 950
VX  2180 GOSUB 630:GOSUB 640:GOTO 1290
RA  2190 POKE 204,NC2:POKE 206,224
KB  2200 Q=USR(ADR(ML$))
SH  2210 GOTO 1290
VJ  2220 GOSUB 630:POKE 54286,64:OPEN #2,6
    ,0,"D:*.*"
DA  2230 INPUT #2,DR$
SW  2240 IF VP2>19 AND HP3=19 THEN HP3=2:V
    P2=17:POKE 54286,192:GOSUB 950:GOSUB 6
    30:GOSUB 640:POKE 54286,64
RC  2250 IF VP2>19 THEN HP3=19:VP2=17
PM  2260 IF DR$(5,8)="FREE" THEN 2290
QH  2270 POSITION HP3,VP2:? DR$:VP2=VP2+1
QN  2280 GOTO 2230
DY  2290 CLOSE #2:POKE 54286,192:POSITION
    HP3+2,VP2:? DR$:GOSUB 950:GOSUB 630:GO
    SUB 640:HP3=2:VP2=17:GOTO 1290
```

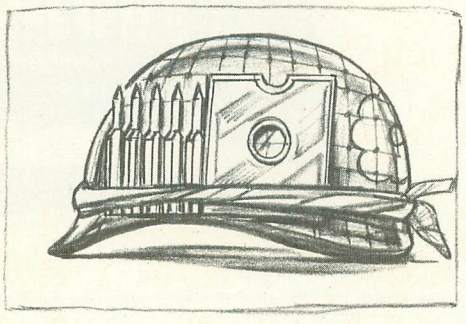
Disk Jacket Printer

by Robert Plotkin

Every disk drive owner has had the experience of being unable to find a file, no matter how many times the disk library is searched. Keeping a printed master disk directory works, but it is often very long and must be updated frequently to be useful. Disk labels can be used, but are often too small and hard to read.



Disk Jacket Printer



The answer is to print individual disk jackets which may be cut out and used for each disk. All disk information, including disk title, number of free sectors, and a sorted disk directory, is printed directly onto the jacket with any Epson compatible printer.

Using The Program

After typing in Listing 1, run the program. The screen will be cleared, and at the top of the screen "TITLE A" will appear to indicate that you are selecting a two line title for side A of the disk. Simply enter each line followed by RETURN, or RETURN alone for a blank line. Each line may contain up to 17 characters, and may include graphics characters and inverse characters. After both lines are entered, insert the indicated side of your disk, and press RETURN to load the directory into memory. If you do not want to print anything on the disk jacket for this side of the disk, press ESCAPE instead. There may be a slight delay while the directory is alphabetized in memory. Now follow the above directions for side B of your disk. Up to 48 filenames can be stored in memory for each side of the disk.

Next the screen will be cleared, and you will be given four choices. By typing "G" you can change the printer graphics density (1-Single density, 2-Double density). Printing with single density graphics will be faster, while double density print will be darker. You can type "N" to create a new disk jacket. This can either be used after a disk jacket is printed, or to correct any errors made while creating the jacket. Type "Q" to exit the program. The fourth option "P" will print the disk jacket. Disk titles will be centered when printed, and directories will be alphabetized. Be sure that your printer is connected and turned on. The screen will turn black during the printing to increase speed.

When the jacket is printed, cut it out, following the dotted lines in Figure 1. Now fold the paper along the space between both of the large rectangles. Both sides should now appear right side up, and the back side should be approximately 3/4" taller than the front. On either side of the front, there will be extra paper which should be folded around to the back, and taped or glued in place.

Robert Plotkin is a 16-year-old junior attending Edward R. Murrow High School in Brooklyn, New York. He has owned an Atari computer for over four years, and has been programming in **Action!** for two years. He holds a purple belt in jujitsu, enjoys science fiction, and collects comic books.

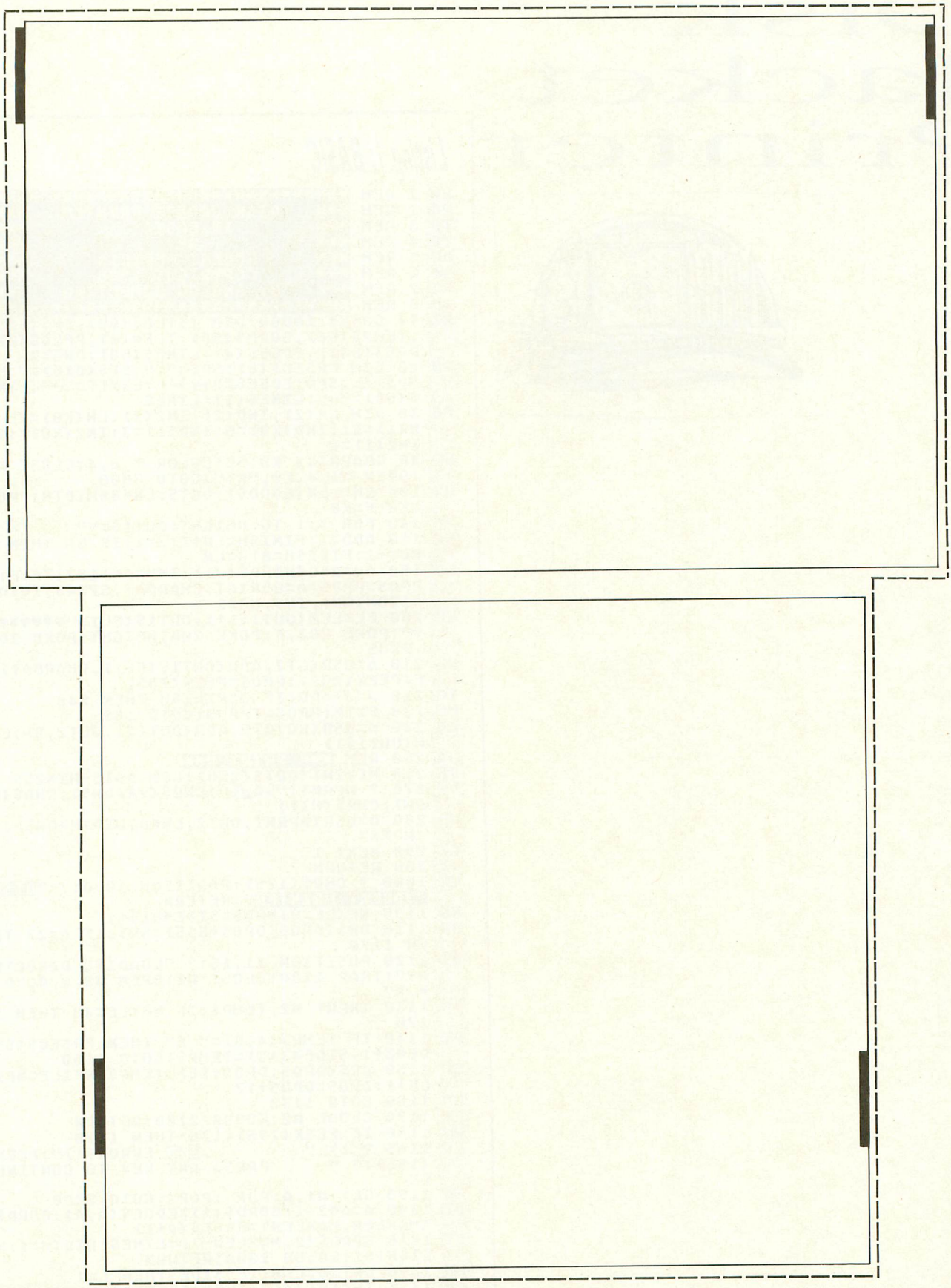
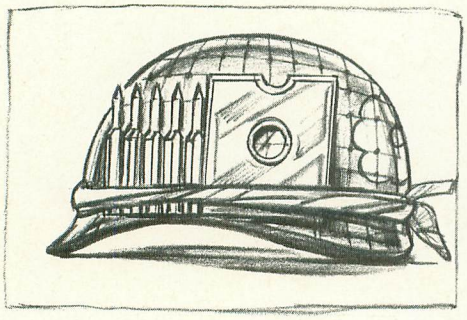


Figure 1

Disk Jacket Printer



Listing 1: BASIC

```
HW 1 REM ********************************
RB 2 REM ***   DISK JACKET PRINTER    ***
TP 3 REM ***       IN BASIC           ***
YQ 4 REM ***          BY              ***
AR 5 REM ***     ROBERT PLOTKIN       ***
GA 6 REM ***     COPYRIGHT 1988       ***
XG 7 REM ***  BY ANALOG COMPUTING     ***
ID 8 REM ********************************
XZ 10 GOSUB 10000:DIM TITLE$(68),SP$(816)
   ,TEMP$(50),BORD$(50),TLEN(4),PFLAG(2),
   DR$(1632),EDGE$(4),LINE$(50):DNS=2
VN 20 DIM FRSEC$(6):SP$=" ":SP$(816)=" ":
   SP$(2)=SP$:EDGE$="┌┐└┘":LINE$="─":LINE
   $(50)="─":LINE$(2)=LINE$
FC 30 DIM LN(2),IND(2),IN2(2):LN(K0)=25:L
   N(1)=21:IND(K0)=5:IND(1)=3:IN2(K0)=12:
   IN2(1)=8
BL 40 GRAPHICS K0:SETCOLOR 2,8,4:CLOSE #1
   :OPEN #1,4,K0,"K:":GOTO 5000
HY 100 LN=LEN(BORD$):DOTS=LN*8*WIDTH:PPOS
   =K0:F=K0
XS 140 FOR I=1 TO HEIGHT:OUT1$=""
UF 150 ADD=1:FINISH=LN+1:J=1:IF BW THEN A
   DD=-1:FINISH=0:J=LN
ED 160 AV=ASC(BORD$(J)):INV=AV>127:TF=F:T
   PPOS=PPOS:A=USR(GT,CHARDAT,57344,IV(AV
   -128*INV),INV,UPD)
GU 200 FL=LEN(OUT1$)+1:OUT1$(FL)="♥♥♥♥♥♥♥
   ♥":POKE 203,F:POKE 204,HEIGHT:POKE 205
   ,PPOS
VX 210 A=USR(GT2,ADR(OUT1$(FL)),CHARDAT):
   F=PEEK(203):PPOS=PEEK(205)
TC 220 J=J+ADD:IF J=FINISH THEN 240
ML 230 F=TF:PPOS=TPPOS:GOTO 160
EB 240 A=USR(ROTATE,ADR(OUT1$),OUT2,BW,LE
   N(OUT1$))
UG 250 REM *** PRINT ***
DE 260 HI=INT(DOTS/256):LOW=DOTS-HI*256
DW 270 ? #PRNT;"ESC◢E";CHR$(74+DNS);CHR$(L
   OW);CHR$(HI);
UO 280 A=USR(APRNT,OUT2,LN*8,WIDTH*DNS):?
    #PRNT
GK 290 NEXT I
YZ 300 RETURN
CR 1000 ? CHR$(125):POSITION 10,0:? "DISK
    JACKET PRINTER":RETURN
XG 1100 NFILE=0:DPOS=SIDE*816+1
UN 1110 DR$(DPOS,DPOS+815)=SP$:IF A=27 TH
   EN 1170
QS 1120 POSITION 11,16:? "LOADING DIRECTO
   RY":TRAP 1180:CLOSE #2:OPEN #2,6,0,"D:
   *.*"
SC 1130 INPUT #2,TEMP$:IF NFILE=48 THEN 1
   170
PT 1140 IF TEMP$(4,5)=" F" THEN FRSEC$(SI
   DE*3+1,SIDE*3+3)=TEMP$:GOTO 1160
EF 1150 DR$(DPOS,DPOS+16)=TEMP$:NFILE=NFI
   LE+1:DPOS=DPOS+17
PH 1160 GOTO 1130
FL 1170 CLOSE #2:GOSUB 2100:RETURN
WJ 1180 IF PEEK(195)=136 THEN 1170
FQ 1185 ? :? "          I/O ERROR!!)";PEEK
   (195):? "     PRESS ANY KEY TO CONTINU
   E"
WB 1190 GET #1,A:POP :POP :GOTO 5000
UI 1200 A=A*2+1:BORD$(1)=EDGE$(A,A):BORD$
   (MAXLEN,MAXLEN)=EDGE$(A+1)
DT 1210 BORD$(2,MAXLEN-1)=LINE$:WIDTH=1:H
   EIGHT=1:GOSUB 2000:RETURN
WH 1300 CNT=0:L=0:IF SIDE THEN L=1
AB 1310 IN=INT((17-TLEN(SIDE*2+L))/2):TEM
   P$=TITLE$(SIDE*34+L*17+1,SIDE*34+L*17+
   17):LN=LN(SIDE):IND=IND(SIDE)+IN
JM 1320 SLEN=17-IN:FILL=1:GOSUB 1400
OU 1330 IF SIDE THEN BORD$(1,1)=CHR$(2):B
   ORD$(LN,LN)=CHR$(2):GOTO 1350
```

```
GT 1340 BORD$(1)=CHR$(22):BORD$(LN)=CHR$(
   22)
XB 1350 WIDTH=2:HEIGHT=3:GOSUB 2000:L=1-L
   :CNT=CNT+1
MM 1360 IF CNT<>2 THEN 1310
BA 1370 RETURN
KQ 1400 BORD$="|":BORD$(LN)="|":BORD$(2,L
   N-1)=SP$
UV 1420 IF FILL AND PFLAG(SIDE) THEN BORD
   $(IND,IND+SLEN-1)=TEMP$
AQ 1430 RETURN
LE 1500 LN=MAXLEN:FILL=0:GOSUB 1400:WIDTH
   =1:HEIGHT=1:GOSUB 2000:RETURN
XD 1600 TEMP$="SIDE A         FREE SECTORS
   ":TEMP$(6,6)=CHR$(SIDE+65)
YS 1610 TEMP$(11,13)=FRSEC$(SIDE*3+1):LN=
   MAXLEN:SLEN=26:IND=IN2(SIDE):FILL=1:GO
   SUB 1400:WIDTH=1:HEIGHT=2
DP 1620 GOSUB 2000:RETURN
FK 1700 DPOS=1:IND=8:WIDTH=1:HEIGHT=1:SLE
   N=35:FILL=1:DA=34:IF SIDE THEN DPOS=15
   99:IND=4:DA=-34
BI 1710 FOR L=0 TO 23:TEMP$=DR$(DPOS,DPOS
   +16):TEMP$(18)=" ":TEMP$(19)=DR$(DPOS+
   17,DPOS+33)
HK 1730 LN=MAXLEN:GOSUB 1400:GOSUB 2000:D
   POS=DPOS+DA:NEXT L:RETURN
RN 2000 BW=0:UPD=0:IF SIDE THEN BW=1:UPD=
   1:BORD$(LEN(BORD$)+1)=SP$(1,(3-WIDTH)*
   2)
GF 2010 GOSUB 100:RETURN
DK 2100 CURR=SIDE*816+1:IF NFILE<2 THEN 2
   170
N5 2110 FOR OUT=1 TO NFILE-1:MATCH=CURR+1
   7
XG 2120 FOR IN=1 TO NFILE-OUT
GZ 2130 IF DR$(MATCH+2,MATCH+9)>=DR$(CURR
   +2,CURR+9) THEN 2150
OZ 2140 I=MATCH+16:J=CURR+16:TEMP$=DR$(MA
   TCH,I):DR$(MATCH,I)=DR$(CURR,J):DR$(CU
   RR,J)=TEMP$
FU 2150 MATCH=MATCH+17:NEXT IN
HX 2160 CURR=CURR+17:NEXT OUT
AX 2170 RETURN
XW 5000 REM *** START OF PROGRAM ***
LD 5010 FOR SIDE=0 TO 1:GOSUB 1000:POSITI
   ON 16,2:? "TITLE ";CHR$(SIDE+65)
TW 5020 TPOS=SIDE*34+1:POKE 752,0:TITLE$(
   TPOS,TPOS+33)=SP$
IR 5030 POSITION 8,4:? "ENTER A TWO LINE
   TITLE":? "   PRESS <RETURN> AFTER EACH
    LINE"
QV 5040 ? "   MAXIMUM 17 CHARACTERS PER L
   INE"
VH 5050 FOR I=0 TO 1:A=I*2:POSITION 12,9+
   A:? "----------------":POSITION 11,8+
   A:INPUT TEMP$
JO 5060 A=LEN(TEMP$):IF A>17 THEN A=17
VM 5070 TLEN(I+SIDE*2)=A:TITLE$(TPOS+I*17
   ,TPOS+I*17+16)=TEMP$:NEXT I
GY 5080 POKE 752,1:POSITION 9,12:? "INSER
   T DISK-->SIDE ";CHR$(SIDE+65):POSITION
    5,14
CA 5090 ? "OR PRESS <ESC> FOR NO DIRECTOR
   Y":GET #1,A:PFLAG(SIDE)=1:IF A=27 THEN
    PFLAG(SIDE)=0
EG 5100 GOSUB 1100:NEXT SIDE
IV 5110 GOSUB 1000:POKE 82,10:POSITION 10
   ,8:? "Graphics Density->":? "New Disk
   Jacket":? "Quit Program"
IE 5120 ? "     PRINT":POKE 82,2
KA 5130 POSITION 28,8:? CHR$(DN$+176):GET
    #1,A
PA 5140 IF A=71 THEN DN$=3-DN$:GOTO 5190
KA 5150 IF A=80 THEN GOSUB 5200:GOTO 5190
UL 5160 IF A=78 THEN 5010
DT 5170 IF A=81 THEN POSITION 15,13:? "QU
   IT(Y/N)?";:GET #1,A:POSITION 15,13:? "
                "
FH 5180 IF A=89 THEN GRAPHICS 0:END
RM 5190 GOTO 5130
```

```
NA 5200 REM *** PRINT JACKET ***
FF 5210 POKE 559,0:TRAP 5270:CLOSE #PRNT:
   OPEN #PRNT,8,K0,"P:":SIDE=0:MAXLEN=49
IQ 5220 A=0:GOSUB 1200:GOSUB 1300:GOSUB 1
   500:GOSUB 1600:GOSUB 1500:GOSUB 1700:A
   =1:GOSUB 1200
TB 5230 SIDE=1:MAXLEN=41:? #PRNT
DN 5240 A=1:GOSUB 1200:GOSUB 1700:GOSUB 1
   500:GOSUB 1600:GOSUB 1500:GOSUB 1300
EB 5250 FOR L=0 TO 6:GOSUB 1500:NEXT L:A=
   0:GOSUB 1200
UQ 5260 POKE 559,34:GOTO 5130
NQ 5270 POKE 559,34:? :? "     I/O ERROR!
   !)";PEEK(195):? "        PRESS ANY KEY
    TO CONTINUE"
TG 5280 GET #1,A:POP :POP :POSITION 14,13
   :? "              ":GOTO 5130
BG 10000 REM *** INITIALIZATION ***
JQ 10010 K0=0:PRNT=3
IY 10020 DIM IV(127),CHARDAT$(8),OUT1$(96
   0),OUT2$(960),ROTATE$(109),GT$(83),PRN
   T$(61),GT2$(45)
HR 10040 FOR I=K0 TO 31:IV(I)=(I+64)*8:NE
   XT I:FOR I=32 TO 95:IV(I)=(I-32)*8:NEX
   T I
OB 10050 FOR I=96 TO 127:IV(I)=I*8:NEXT I
HC 10070 OUT2$(1)="♥":OUT2$(960)="♥":OUT2
   $(2)=OUT2$:GT=ADR(GT$):GT2=ADR(GT2$):A
   PRNT=ADR(PRNT$)
FV 10080 CHARDAT$="♥♥♥♥♥♥♥♥":CHARDAT=ADR(
   CHARDAT$):OUT2=ADR(OUT2$):ROTATE=ADR(R
   OTATE$)
QA 10090 REM *** CREATE ML STRINGS ***
SM 10100 RESTORE 12000:FOR I=1 TO 109:REA
   D A:ROTATE$(I,I)=CHR$(A):NEXT I
UR 10110 FOR I=1 TO 83:READ A:GT$(I,I)=CH
   R$(A):NEXT I
VK 10120 FOR I=1 TO 45:READ A:GT2$(I,I)=C
   HR$(A):NEXT I
AD 10130 FOR I=1 TO 61:READ A:PRNT$(I,I)=
   CHR$(A):NEXT I
DQ 10140 RETURN
DB 12000 DATA 104,104,133,213,104,133,212
   ,104,133,215,104,133,214,104,104,133,2
   16,104,133,222,104,133,221,160,0
LW 12010 DATA 132,220,162,0,177,212,230,2
   12,208,2,230,213,149,228,232,224,8,208
   ,241,162,0,22,228,42,232
EB 12020 DATA 224,8,208,248,166,216,240,8
   ,133,219,152,73,7,168,165,219,145,214,
   230,220,165,220,164,220,201
SN 12030 DATA 8,208,222,162,8,165,221,56,
   233,1,133,221,176,2,198,222,230,214,20
   8,2,230,215,202,208,236
AN 12040 DATA 166,221,208,175,166,222,208
   ,171,96
DO 12050 DATA 104,162,10,104,149,212,202,
   208,250,166,215,240,4,198,215,198,215,
   165,219,133,223,165,220,133,224
OV 12060 DATA 165,217,24,101,223,144,2,23
   0,224,133,223,165,224,101,218,133,224,
   160,0,177,223,69,215,153,225
BY 12070 DATA 0,200,152,201,9,208,243,160
   ,0,132,217,165,213,240,4,152,73,7,168,
   185,225,0,164,217,145
RF 12080 DATA 221,200,152,201,8,208,233,9
   6
OP 12090 DATA 104,104,133,217,104,133,216
   ,104,133,219,104,133,218,162,0,134,220
   ,164,205,177,218,164,220,145,216
DR 12100 DATA 230,203,165,203,197,204,208
   ,6,169,0,133,203,230,205,232,224,8,208
   ,227,96
VK 12110 DATA 104,162,6,104,149,211,202,2
   08,250,162,11,142,114,3,162,0,142,120,
   3,142,121,3,160,0,132
JP 12120 DATA 220,177,216,162,48,32,86,22
   8,230,220,166,220,228,212,208,243,230,
   216,208,2,230,217,198,214,208
RB 12130 DATA 227,165,215,240,5,198,215,2
   4,144,218,96
```

Fast Print

by Bill Bodenstein

After using an Atari 800 for two years, I was immediately impressed with the faster screen output on XL and XE models. When you spend as much time LISTing as I do, speed is very important. But as fast as the newer print character routines in the operating system are, screen output can be made even faster. Two to three times faster.

Want proof? Just type in the data in Listing 1 using M/L Editor. M/L Editor will create an object file—name it AUTORUN.SYS, or FASTPRNT.OBJ to load from DOS—that when binary loaded, handles all output sent to the screen editor device ("E:"). You'll find the source code in Listing 2.

A Test Drive

Okay, you've got the Fast Print routine loaded and running. From BASIC, load any long BASIC program for a demonstration. Now type "LIST". Pretty speedy, huh? Notice that a "press any key" prompt appears at the bottom of the screen everytime the screen fills up, letting you read the text before it scrolls up and away. To disable this pause feature, just POKE a zero into location 847. POKE any non-zero value to restore the pause. Also, to disable Fast Print, type POKE 846,0. POKEing any non-zero value back into 846 will once again relieve the operating system of some of the work. Pressing RESET completely disables Fast Print, requiring reloading before you can use it again.

Changing the Object Code

If you understand assembly language, take a look at Listing 2. The program "pokes" most of the characters it receives directly into screen RAM, and lets the OS handle the complicated stuff, like control characters and scrolling.

The initialization routine sets both locations 846 and 847 ($34E and $34F) to one. If you'd like the pause feature off when loaded, change the LDA #1 in line 940 to LDA #0.

Because Fast Print is active constantly, I was forced to place it in Page 6. If put in LOMEM, going to DOS would write over it and cause a crash. If a more favored machine language program already resides in Page 6 of memory, you can always assemble the code elsewhere, higher up in RAM. If you choose unprotected memory, be very careful that you store nothing else there. Press RESET first if you should need to.

Bill Bodenstein is a **Computer Science Major** (for what seems to him like his eighth consecutive year) at the University of Cincinnati. He owns two Atari 800XLs (one sans a power supply) and one 800. He has taken courses at UC in PASCAL, FORTRAN, PL/I, BASIC, ASSEMBLY, and COBOL, but his favorite language is 6502 M/L. He has already taught himself 68000 assembly in preparation for the day he purchases an Atari ST—just as soon as the prices come down another $300 so he can afford one.

Fast Print

Listing 1: M/L Editor Data

```
1000 DATA 255,255,0,5,104,5,173,9,2,20
1,6,240,85,141,219,6,5052
1010 DATA 173,8,2,141,218,6,120,169,17
4,141,8,2,169,6,141,9,2715
1020 DATA 2,88,160,15,185,0,228,153,20
2,6,136,16,247,169,255,141,707
1030 DATA 208,6,169,5,141,209,6,169,20
2,141,33,3,169,6,141,34,3697
1040 DATA 3,162,0,169,12,157,66,3,32,8
6,228,169,3,157,66,3,2490
1050 DATA 169,12,157,74,3,169,103,157,
68,3,169,5,157,69,3,32,1141
1060 DATA 86,228,169,1,141,78,3,169,1,
141,79,3,96,69,58,0,67
1070 DATA 6,201,6,172,78,3,208,3,76,13
0,6,164,87,208,249,164,8492
1080 DATA 17,240,73,172,255,2,208,247,
172,79,3,240,82,201,125,208,1537
1090 DATA 7,160,0,140,201,6,240,224,20
1,155,240,6,166,85,228,83,657
1100 DATA 144,61,238,201,6,32,130,6,17
4,201,6,224,21,144,47,160,5948
1110 DATA 0,140,201,6,185,184,6,145,94
,200,192,17,208,246,169,255,3173
1120 DATA 141,252,2,205,252,2,208,9,16
6,17,208,247,160,128,132,17,8431
1130 DATA 96,141,252,2,160,0,152,145,9
4,200,192,17,208,249,96,166,744
1140 DATA 85,228,83,176,16,170,41,127,
201,125,176,8,201,32,176,16,6085
1150 DATA 201,27,144,12,138,168,173,7,
228,72,173,6,228,72,152,96,7385
1160 DATA 201,96,8,138,40,176,3,56,233
,32,160,0,145,94,230,85,6042
1170 DATA 230,99,230,94,208,2,230,95,1
77,94,133,93,73,128,145,94,7618
1180 DATA 96,72,169,0,141,201,6,104,10
8,218,6,0,0,156,240,242,7586
1190 DATA 229,243,243,128,225,238,249,
128,235,229,249,158,0,226,2,227,4332
1200 DATA 2,0,5,0,0,0,0,0,0,0,0,0,0,0,
0,0,1217
```

Listing 2: Assembly

```
10 ********************************
20 **     F A S T   P R I N T    **
30 **     By Bill Bodenstein     **
40 **        11/24/86            **
50 ********************************
60 ;
70 ;This M/L routine will put
80 ;characters directly into screen
90 ;memory for faster printing.
0100 ;Output sent to the screen
0110 ;editor is routed here.
0120 ;
0130 ** EQUATES **
0140 ;
0150 RGTMARGIN = $53
0160 CURSCOL = $55
0170 KEYPRESS = $02FC
0180 NOKEY = 255
0190 BRKKEY = $11
0200 BREAK = 128
0210 CLEAR = 125
0220 RETURN = 155
0230 ;
0240 ICCOM = $0342
0250 ICBADR = $0344
0260 ICBAUX = $034A
0270 FPFLG = $034E
0280 PAUSEFLG = $034F
0290 OPEN =  3
0300 CLOSE = 12
0310 INOUT = 12
0320 CIO =   $E456
0330 EHANDTAB = $E400
0340 ;
0350 VKEYBD = $0208
0360 ;
0370 ;
0380     *=  $0500
0390 ;
0400 INITIALIZE
0410 SETKEYINT
0420     LDA VKEYBD+1 ;Change keybd
0430     CMP # >NEWKEYINT ;interrupt
0440     BEQ SETFLGS ;vector to go
0450     STA OLDKEYINT+1 ;to here
0460     LDA VKEYBD   ;before
0470     STA OLDKEYINT ;handling
0480     SEI          ;keypress
0490     LDA # <NEWKEYINT
0500     STA VKEYBD
0510     LDA # >NEWKEYINT
0520     STA VKEYBD+1
0530     CLI
0540 ;
0550 COPYEHAND
0560     LDY #15      ;Move E handler
0570 COPYBYTE LDA EHANDTAB,Y ;table
0580     STA NEWEHAND,Y ;from o.s. to
0590     DEY          ;Fast Print
0600     BPL COPYBYTE
0610 ;
0620 SETNEWPUT
0630     LDA # <[EPUTCHAR-11 ;Set new
0640     STA NEWEHAND+6 ;put char
0650     LDA # >[EPUTCHAR-11 ;vector
0660     STA NEWEHAND+7 ;to F.P.
0670 ;
0680 SETNEWHAND
0690     LDA # <NEWEHAND
0700     STA $0321
0710     LDA # >NEWEHAND
0720     STA $0322
0730 ;
0740 CLOSESCR
0750     LDX #0       ;Close editor
0760     LDA #CLOSE
0770     STA ICCOM,X
0780     JSR CIO
0790 ;
0800 OPENSCR
0810     LDA #OPEN    ;Then open it
0820     STA ICCOM,X
0830     LDA #INOUT
0840     STA ICBAUX,X
0850     LDA # <EDEV
0860     STA ICBADR,X
0870     LDA # >EDEV
0880     STA ICBADR+1,X
0890     JSR CIO
0900 ;
0910 SETFLGS
0920     LDA #1       ;Set flags:
0930     STA FPFLG    ;Fast Print on
0940     LDA #1
0950     STA PAUSEFLG ;Pause on
0960     RTS
0970 ;
0980 ;
0990 EDEV .BYTE "E:"
1000 ;
1010 ;
1020     *=  $0600
1030 ;
1040 ;Receive a character being put
```

```
1050 ;to the screen editor.
1060 ;
1070 EPUTCHAR
1080 FPON?
1090      LDY FPFLG    ;Go to o.s. if
1100      BNE GR0?     ;F.P. off
1110 GO.OS JMP PRINTCHAR
1120 ;
1130 GR0?
1140      LDY $57      ;Go to o.s. if
1150      BNE GO.OS    ;not text mode
1160 ;
1170 BREAK?
1180      LDY BRKKEY   ;Abort if
1190      BEQ ABORTOUTPUT ;breaked
1200 ;
1210 CTRL1?
1220      LDY $02FF    ;Loop if CTRL-1
1230      BNE BREAK?   ;pressed
1240 ;
1250 PAUSEON?
1260      LDY PAUSEFLG
1270      BEQ FASTPRINTCHAR
1280 ;
1290 CLEAR?
1300      CMP #CLEAR   ;If clear char,
1310      BNE COUNTLINE ;reset row
1320      LDY #0       ;counter
1330      STY ROWCNTR
1340      BEQ GO.OS
1350 ;
1360 COUNTLINE
1370      CMP #RETURN ;Keep count of #
1380      BEQ INCLINE ;of lines
1390      LDX CURSCOL ;printed
1400      CPX RGTMARGIN
1410      BCC FASTPRINTCHAR
1420 INCLINE INC ROWCNTR
1430 ;
1440 ;Every 22nd line printed, pause
1450 ;output and wait for keypress.
1460 ;
1470 PAUSESCR?
1480      JSR PRINTCHAR
1490      LDX ROWCNTR
1500      CPX #21
1510      BCC LEAVE
1520 ;
1530 PUTMSG
1540      LDY #0       ;Print prompt
1550      STY ROWCNTR ;message
1560 PUTBYTE LDA MSG,Y
1570      STA ($5E),Y
1580      INY
1590      CPY # <[ROWCNTR-MSG]
1600      BNE PUTBYTE
1610 ;
1620      LDA #NOKEY   ;Clear last
1630      STA KEYPRESS ;key pressed
1640 ;
1650 WAITFORKEY
1660      CMP KEYPRESS ;Wait for user
1670      BNE CONTOUTPUT ;to hit a key
1680      LDX BRKKEY
1690      BNE WAITFORKEY
1700 ;
1710 ABORTOUTPUT
1720      LDY #BREAK   ;<BREAK> pressed
1730      STY BRKKEY
1740      RTS
1750 ;
1760 CONTOUTPUT
1770      STA KEYPRESS ;Clear keypress
1780 CLRMSG
1790      LDY #0       ;Clear prompt
1800      TYA          ;msg
1810 CLRBYTE STA ($5E),Y
1820      INY
1830      CPY # <[ROWCNTR-MSG]
1840      BNE CLRBYTE
1850 LEAVE RTS
1860 ;
1870 ;Put character directly in
1880 ;screen memory unless scrolling
1890 ;or control character.
1900 ;
1910 FASTPRINTCHAR
1920      LDX CURSCOL
1930      CPX RGTMARGIN
1940      BCS PRINTCHAR
1950      TAX
1960      AND #127     ;Is char a
1970      CMP #125     ;non-control
1980      BCS PRINTIT ;char?
1990      CMP #32
2000      BCS PUTIT
2010      CMP #27
2020      BCC PUTIT
2030 ;
2040 PRINTIT
2050      TXA
2060 PRINTCHAR
2070      TAY          ;Print this char
2080      LDA EHANDTAB+7 ;with o.s.
2090      PHA          ;routine
2100      LDA EHANDTAB+6
2110      PHA
2120      TYA
2130      RTS
2140 ;
2150 PUTIT
2160 CONVCHAR
2170      CMP #96      ;Convert to
2180      PHP          ;internal
2190      TXA          ;charset
2200      PLP
2210      BCS PUTCHAR
2220      SEC
2230      SBC #32
2240 ;
2250 PUTCHAR
2260      LDY #0       ;Put char in
2270      STA ($5E),Y ;curs pos in
2280      ;            screen mem
2290 MOVCURS
2300      INC CURSCOL ;Move cursor
2310      INC $63      ;pointers to
2320      INC $5E      ;next column
2330      BNE PUTCURS
2340      INC $5F
2350 ;
2360 PUTCURS
2370      LDA ($5E),Y ;Save char under
2380      STA $5D      ;cursor and
2390      EOR #128     ;inverse it
2400      STA ($5E),Y
2410 ;
2420 EXITPUTCHAR RTS
2430 ;
2440 ;
2450 ;Clear row counter for screen
2460 ;pause everytime a key is
2470 ;pressed.
2480 ;
2490 NEWKEYINT
2500      PHA
2510      LDA #0
2520      STA ROWCNTR
2530      PLA
2540      JMP (OLDKEYINT)
2550 ;
2560 ;
2570 MSG .SBYTE "  <press any key>"
2580 ROWCNTR .BYTE 0
2590 NEWEHAND *= *+16
2600 OLDKEYINT *= *+2
2610 ;
2620 ENDFP
2630 ;
2640 ;
2650      *=  $02E0
2660      .WORD INITIALIZE
```

COM-DOS

48k disk

by Robert Berry

The Atari Disk Utilities Package is a necessary tool in using any Atari computer with a disk drive. Well, it isn't *really* necessary, with all of the DOS-functions-from-BASIC utilities available. But these programs eat up memory otherwise usable by BASIC (or whatever language you happen to be programming in). So Atari has the right idea in having a disk-resident DOS instead of a memory-resident one. But the menu is sometimes awkward to use. For example, when you try to format a disk, you're first asked the drive number, then you're asked if you're sure, then you have to press RETURN. Wouldn't it be nice to *escape* from all those prompts?

One way would be to buy one of those DOS-from-BASIC utilities. Or you could invest in a completely new Disk Operating System. But if you don't want to spend any money, you can take a look at *COM-DOS*.

COM-DOS is a replacement Disk Utilities Package, compatible with Atari DOS 2.0 & 2.5. It is a command-driven DOS, so if you can type well, you'll probably like COM-DOS better than the menu of Atari's DOS. COM-DOS provides all functions on the DOS menu except duplicate disk (and, with DOS 2.5, the Format Single option). Plus, it adds several functions not on the menu. It functions just like the Atari DUP, MEM.SAV files work the same, and it is completely compatible with the 130XE ramdisk.

Typing It In

To use COM-DOS, first type in Listing 1, using M/L Editor found elsewhere in this issue. Name the resulting file DUP.SYS. Listing 2 is the assembly source listing, created using MAC/65.

Using COM-DOS

When the READY prompt comes up (assuming you booted with BASIC installed), type DOS. When it has loaded in, you'll see the title and a prompt to enter a command. To view the command list, type "HELP" and press RETURN. The screen will clear, and all of the commands supported by COM-DOS will be displayed. Before I discuss the commands themselves, I must warn you about the syntax these commands use. Unlike BASIC and most other languages, COM-DOS is very picky about syntax. BASIC will remove any extra spaces in a program line and insert any that were left out. COM-DOS will not. Each command must be entered with no spaces before it. It won't hurt anything if you do insert unnecessary spaces, but COM-DOS won't accept it as a valid command. Also, each command must use capital letters and no inverse characters. Again, failure to obey that rule won't hurt anything; the program will just refuse the command. Figure 1 lists the commands in their proper syntaxes, along with examples of their usage.

Here is a list of the COM-DOS commands and their functions:

HELP—As you have seen, HELP lists all of the commands supported by **COM-DOS.**
LOCK D:FILENAME.EXT—Locks the file specified.
UNLOCK D:FILENAME. EXT—Unlocks the file specified.
DELETE D:FILENAME. EXT—Deletes the file specified.
RENAME D:OLDNAME,NEWNAME—Renames file D:OLDNAME, NEWNAME.
DIRn—Displays the directory of drive #n. The number isn't necessary; the default is drive #1, but, if a different drive is specified, it becomes the default.

FORMATn—Formats the disk in drive #n. The default is drive #1, and it stays drive #1, even if a new drive is specified, so be careful with this command. Remember that you are not asked if you're sure; so be especially careful with this one!
WDOSn—Writes DOS.SYS to drive #n; the default is (and stays) drive #1. This command doesn't write DUP.SYS; so you must either COPY or DUPlicate DUP.SYS.
WMEMn—Writes MEM.SAV to the disk in drive #n. The default is always drive #1.

BSAVE D: FILENAME, START, END—Saves the memory from START to END on the file specified. START and END must be decimal numbers! When the file is saved, you will be prompted for a RUN/INIT address. If you want to append one to the file, enter it (again, in decimal), and press RETURN. The address will be appended to the file. If you don't want to specify a RUN/INIT address, just press RETURN. The file will be closed after either response.
BLOAD D:FILENAME.EXT—Loads a binary file, just as option 'L' from the DOS menu does.
COPY D:FILE1,D2:FILE2—Copies FILE1 to D2:FILE2. The first file can be appended to the second file by placing a slash (/) after the second filename. Any legal device can be substituted for either (or both) filenames.
DUP D:FILENAME—Duplicates the file specified. WARNING: This function will usually erase any programs in memory; so be careful with this command too. You'll be prompted to insert the source disk, then the destination disk.
BOOT filename.ext—Makes the file specified automatically run when the disk in drive #1 is booted. This command writes an AUTORUN.SYS file to the disk; so if one already exists, delete it. The file must be a SAVEd BASIC program. Also, note that the 'D:' designation is not specified in this command, just the filename itself.
GO ADDR—Executes a machine language routine at decimal address ADDR. The address must be a decimal number; if it isn't, there's no telling what might happen.
?hxnm—Converts the four-digit hexadecimal number specified into a decimal number. Notice that there is no space between the question mark and the hex number.
REBOOT—Reboots the computer. Remember, there are no questions asked, so anything in memory is erased (REBOOT is identical to turning the computer off and on again).
CLICK—On XL/XE computers, this command toggles the key-click on/off. It has no effect on 400/800 computers.
STATUS—Tells whether the write-verify function is on or off, the current active drives, and the maximum number of files that can be open at one time. If you want to change any of these, follow the prompts. The first question is whether you want to toggle the write-verify function. Press Y or N, then RETURN. Remember, turning off write-verify will make the computer save files faster, but the reliability is decreased. If you use high-quality disks, you can usually live without write-verify. Otherwise, I would suggest leaving write-verify on.

The next deals with active drives. If you don't want to change them, type N and press RETURN. If you do want to change them, answer Y to the

prompt and press RETURN. Type the drive numbers, one at a time, pressing RETURN after each one. Enter O when you're finished. For example, to activate only drive 1, you would type:

Y <RETURN>
1 <RETURN>
0 <RETURN>

Then the next question will come up. If you want to change the number of files that can be opened at one time, type Y, then RETURN. Then enter the maximum number of files and press RETURN.

The changes will be made after each entry. To save DOS with your defaults, use WDOS. If you want to escape from the questions and not change anything, you can press RETURN at any prompt to take you back to the 'Enter command' prompt.

BASIC—Returns control to BASIC (or whatever cartridge is plugged in). If a MEM.SAV file was present on the disk in drive #1 (or the ramdisk if you have a 130XE), and you didn't use the DUP command, anything you had in memory will still be there.

COM-DOS has some extra features I added for safety and comfort. First, the break key is disabled, so you can't accidentally press it and wipe out everything that was in memory. Second, the background color is changed to dark green and the cursor flashes. Third, any errors that are encountered during I/O will be reported to the user. Finally, if you have an XL/XE machine, fine scrolling is enabled, and the Click command allows you to toggle the keyclick on and off.

I wrote COM-DOS using MAC/65 and the macros provided in the manual. It started out as a command-driven DOS with nine commands. At the time I wrote the first version, it hadn't occurred to me to use the macros, and the program didn't work the way I wanted it to. I gave up that project and started the second version. In that one, I had it load into memory as an AUTORUN.SYS file, and it added the original nine commands to BASIC, eliminating the need for DUP.SYS. But it didn't work like I wanted it to either (it locked up for no apparent reason). After giving that up, I realized that the macros might help me. They did, but the resulting code is extremely long. The advantage is that the program works nearly flawlessly. Notice I said *nearly*. I would strongly suggest saving any important programs or data in memory before using the following commands: BLOAD (unless you know that it will load into safe memory); DUP (it usually erases anything in memory); REBOOT (it's obvious!); and GO (it might not always recover).

COM-DOS should work with any programs that Atari DOS works with, since it uses DOS.SYS and is almost exactly the same length as Atari's Disk Utilities Package. You can make copies of COM-DOS by using WDOS to write DOS.SYS, then copying or duplicating DUP.SYS to another disk.

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.

Command	Function
HELP	Lists commands.
LOCK D:FILENAME	Locks FILENAME.
UNLOCK D:FILENAME	Unlocks FILENAME.
DELETE D:FILENAME	Deletes FILENAME.
RENAME: D:FILE1,FILE2	Renames FILE1 in drive 1.
DIRn	Displays directory of drive #n.
DIR	Shows directory of last drive specified in DIR command.
FORMATn	Formats disk in drive #n. Defaults to drive 1.
WDOSn	Writes DOS.SYS to drive #n. Defaults to drive 1.
WMEMn	Writes MEM.SAV to drive #n. Defaults to drive 1.
BSAVE D:FILENAME,1536,1664	Saves memory between 1536 and 1664 (decimal) to FILENAME.
BLOAD D:FILENAME	Loads binary file FILENAME.
COPY D:FILE1,D4:FILE1	Copies FILE1 from drive 1 to drive 4.
COPY D:FILE1,D2:FILE2	Copies FILE1 from drive 1 to FILE2 on drive 2.
COPY D:FILE1,D2:FILE1/	Copies FILE1 from drive 1 and appends it to FILE1 already existing on drive 2.
COPY D:FILENAME,P:	Copies FILENAME to printer.
COPY E:,P:	Copies anything typed on the screen to the printer.
COPY C:,D:FILENAME	Copies a cassette file to D:FILENAME.
DUP D:FILENAME	Duplicates FILENAME.
BOOT FILENAME.BAS	Makes SAVEd BASIC program FILENAME.BAS run automatically when disk is booted.
GO 1536	Runs at decimal address 1536 (page six).
?E474	Converts 4-digit hex number E474 to a decimal number (58484).
?00CB	The hex number MUST be four digits! This example prints 203.
REBOOT	Reboots the computer.
CLICK	Toggles the keyclick on XL/XE computers.
STATUS	Shows current status described in article.
BASIC	Returns to cartridge.

COM-DOS

```
1000 DATA 255,255,26,29,21,30,49,32,22
4,2,225,2,0,0,255,255,5283
1010 DATA 255,255,128,6,188,6,169,2,13
3,84,162,0,169,9,157,66,3911
1020 DATA 3,169,170,157,68,3,169,6,157
,69,3,169,1,157,72,3,1591
1030 DATA 157,73,3,32,86,228,169,0,133
,84,169,12,141,252,2,96,5418
1040 DATA 82,85,78,34,68,58,0,0,0,0,0,
0,0,0,46,0,3040
1050 DATA 0,0,155,224,2,225,2,128,6,0,
1,2,3,4,5,6,5164
1060 DATA 7,8,9,0,0,0,0,0,0,0,0,10,11,12
,13,14,15,2140
1070 DATA 160,0,162,0,185,249,29,201,1
55,240,93,221,128,5,208,5,8350
1080 DATA 232,200,76,136,29,200,185,24
9,29,201,155,208,248,200,200,200,5812
1090 DATA 185,249,29,201,255,240,5,162
,0,76,136,29,76,203,29,253,7627
1100 DATA 67,111,109,109,97,110,100,32
,110,111,116,32,115,117,112,112,4618
1110 DATA 111,114,116,101,100,33,155,1
62,0,169,9,157,66,3,169,179,5252
1120 DATA 157,68,3,169,29,157,69,3,169
,24,157,72,3,169,0,157,2961
1130 DATA 73,3,32,86,228,76,85,33,185,
250,29,133,203,185,251,29,9642
1140 DATA 133,204,108,203,0,70,79,82,7
7,65,84,155,24,31,76,79,1723
1150 DATA 67,75,155,145,31,85,78,76,79
,67,75,155,185,31,68,69,3260
1160 DATA 76,69,22,30,17,31,84,69,155,
225,31,82,69,78,65,77,2137
1170 DATA 69,155,9,32,68,73,82,155,49,
32,66,65,83,73,67,155,2149
1180 DATA 76,34,82,69,66,79,79,84,155,
119,228,66,76,79,65,68,3917
1190 DATA 155,134,34,67,79,80,89,155,1
70,34,68,85,80,155,21,36,2460
1200 DATA 87,68,79,83,155,131,37,87,77
,69,77,155,242,37,66,83,4580
1210 DATA 65,86,69,155,203,38,66,79,79
,84,155,223,40,72,69,76,4322
1220 DATA 80,155,214,41,71,79,155,138,
43,83,84,65,84,85,83,155,4362
1230 DATA 159,43,67,76,73,67,75,155,82
,48,63,155,93,48,255,68,5077
1240 DATA 49,58,155,68,49,58,42,46,42,
155,68,105,114,101,99,116,3570
1250 DATA 111,114,121,32,45,32,68,114,
105,118,101,32,35,49,155,68,2059
1260 DATA 49,58,68,79,83,46,83,89,83,1
55,68,49,58,77,69,77,1661
1270 DATA 46,83,65,86,155,87,114,105,1
16,101,32,86,101,114,105,102,4510
1280 DATA 121,58,32,65,99,116,105,118,
101,32,68,114,105,118,101,115,4460
1290 DATA 58,32,77,97,120,105,109,117,
109,32,35,32,111,102,32,102,2013
1300 DATA 105,108,101,115,32,116,104,9
7,116,32,99,97,110,32,98,101,3325
1310 DATA 32,111,112,101,110,32,32,32,
115,105,109,117,108,116,97,110,4457
1320 DATA 101,111,18,31,13,32,117,115,
108,121,58,32,173,134,5,201,4437
1330 DATA 56,208,71,76,69,31,67,97,110
,39,116,32,102,111,114,109,3469
1340 DATA 97,116,32,100,114,105,118,10
1,32,56,32,102,114,111,109,32,2606
1350 DATA 67,79,77,45,68,79,83,33,155,
162,0,169,9,157,66,3,2041
1360 DATA 169,34,157,68,3,169,31,157,6
9,3,169,35,157,72,3,169,3570
1370 DATA 0,157,73,3,32,86,228,76,85,3
3,141,144,30,162,16,169,4771
1380 DATA 254,157,66,3,169,0,157,74,3,
169,0,157,75,3,169,143,4135
1390 DATA 157,68,3,169,30,157,69,3,32,
86,228,192,1,208,3,76,4113
1400 DATA 85,33,76,216,32,162,16,169,3
5,157,66,3,169,0,157,74,3622
1410 DATA 3,169,0,157,75,3,169,133,157
,68,3,169,5,157,69,3,2519
1420 DATA 32,86,228,192,1,208,3,76,85,
33,76,216,32,162,16,169,5109
1430 DATA 36,157,66,3,169,0,157,74,3,1
69,0,157,75,3,169,135,3839
1440 DATA 157,68,3,169,5,157,69,3,32,8
6,228,192,1,208,3,76,4038
1450 DATA 85,33,76,216,32,162,16,169,3
3,157,66,3,169,0,157,74,3654
1460 DATA 3,169,0,157,75,3,169,135,157
,68,3,169,5,157,69,3,2585
1470 DATA 32,86,228,192,1,208,3,76,85,
33,76,216,32,162,16,169,5159
1480 DATA 32,157,14,32,9,33,66,3,169,0
,157,74,3,169,0,157,1778
1490 DATA 75,3,169,135,157,68,3,169,5,
157,69,3,32,86,228,192,5706
1500 DATA 1,208,3,76,85,33,76,216,32,1
73,131,5,201,155,240,6,7111
1510 DATA 141,148,30,141,173,30,162,16
,169,3,157,66,3,169,6,157,3985
1520 DATA 74,3,169,0,157,75,3,169,147,
157,68,3,169,30,157,69,4468
1530 DATA 3,32,86,228,192,1,240,3,76,2
16,32,162,0,169,9,157,5590
1540 DATA 66,3,169,154,157,68,3,169,30
,157,69,3,169,255,157,72,7210
1550 DATA 3,169,0,157,73,3,32,86,228,1
62,16,169,5,157,66,3,2991
1560 DATA 169,128,157,68,3,169,5,157,6
9,3,169,255,157,72,3,169,6416
1570 DATA 0,157,73,3,32,86,228,192,136
,240,40,192,1,240,3,76,6925
1580 DATA 216,32,162,0,169,9,157,66,3,
169,128,157,68,3,169,5,3422
1590 DATA 157,69,3,169,255,157,72,3,16
9,0,157,73,3,32,86,228,4864
1600 DATA 76,129,32,162,16,169,12,157,
66,3,32,86,228,76,85,33,2951
1610 DATA 132,212,162,16,169,12,157,66
,3,32,86,228,162,32,169,12,4570
1620 DATA 157,66,3,32,86,228,169,0,133
,213,32,170,217,32,230,216,921
1630 DATA 160,255,200,185,128,5,16,250
,41,127,153,128,5,169,155,153,8484
1640 DATA 129,5,10,33,5,34,162,0,169,1
1,157,66,3,169,73,157,3466
1650 DATA 68,3,169,33,157,69,3,169,12,
157,72,3,169,0,157,73,3161
1660 DATA 3,32,86,228,162,0,169,9,157,
66,3,169,128,157,68,3,4026
1670 DATA 169,5,157,69,3,169,255,157,7
2,3,169,0,157,73,3,32,2823
1680 DATA 86,228,76,85,33,73,47,79,32,
69,114,114,111,114,32,45,2193
1690 DATA 32,169,202,141,197,2,169,240
,141,198,2,162,0,169,11,157,7588
1700 DATA 66,3,169,163,157,68,3,169,33
,157,69,3,169,16,157,72,4087
1710 DATA 3,169,0,157,73,3,32,86,228,1
69,255,141,252,2,162,0,7947
1720 DATA 169,5,157,66,3,169,128,157,6
8,3,169,5,157,69,3,169,4132
1730 DATA 255,157,72,3,169,0,157,73,3,
32,86,228,76,132,29,155,4835
1740 DATA 69,110,116,101,114,32,99,111
,109,109,97,110,100,58,155,169,6723
1750 DATA 255,141,110,2,162,96,169,3,1
57,66,3,169,8,157,74,3,2812
1760 DATA 169,0,157,75,3,76,207,33,69,
58,0,169,204,157,68,3,4031
1770 DATA 169,33,157,69,3,32,86,228,76
,5,34,125,67,79,77,45,1845
1780 DATA 68,79,83,32,32,32,32,32,32,8
6,101,114,46,32,51,46,9389
1790 DATA 49,32,32,32,32,32,32,32,48,5
0,47,48,54,47,56,54,8048
1800 DATA 155,162,6,34,1,35,0,169,9,15
7,66,3,169,223,157,68,5175
1810 DATA 3,169,33,157,69,3,169,38,157
,72,3,169,0,157,73,3,2263
1820 DATA 32,86,228,165,16,41,127,133,
16,141,14,210,169,62,141,40,5695
```

```
1830 DATA 2,169,34,141,41,2,169,10,141
,26,2,76,85,33,173,243,4829
1840 DATA 2,73,3,141,243,2,169,10,141,
26,2,96,165,8,240,3,3659
1850 DATA 76,116,228,76,101,34,253,78,
111,32,67,97,114,116,114,105,5966
1860 DATA 100,103,101,33,155,162,0,169
,9,157,66,3,169,86,157,68,4957
1870 DATA 3,169,34,157,69,3,169,15,157
,72,3,169,0,157,73,3,2142
1880 DATA 32,86,228,76,85,33,162,16,16
9,3,157,66,3,169,4,157,4004
1890 DATA 74,3,169,0,157,75,3,169,134,
157,68,3,169,5,157,69,4371
1900 DATA 3,32,86,228,32,200,21,76,85,
33,160,7,185,128,5,201,5679
1910 DATA 44,240,4,200,76,172,34,169,1
55,153,128,5,200,162,0,185,8469
1920 DATA 128,5,157,252,28,201,155,240
,5,200,232,76,191,34,162,16,9042
1930 DATA 169,3,157,66,3.169,4,157,74,
3,169,0,157,75,3,169,3548
1940 DATA 133,157,68,3,169,5,157,69,3,
32,86,228,192,1,240,3,5316
1950 DATA 76,216,32,160,255,200,185,25
2,28,201,47,240,41,201,155,208,3639
1960 DATA 244,162,2,35,253,35,32,169,3
,157,66,3,169,8,157,74,3932
1970 DATA 3,169,0,157,75,3,169,252,157
,68,3,169,28,157,69,3,4330
1980 DATA 32,86,228,192,1,240,45,76,21
6,32,169,155,153,252,28,162,516
1990 DATA 32,169,3,157,66,3,169,9,157,
74,3,169,0,157,75,3,2185
2000 DATA 169,252,157,68,3,169,28,157,
69,3,32,86,228,192,1,240,7439
2010 DATA 3,76,216,32,162,16,169,7,157
,66,3,169,210,157,68,3,5216
2020 DATA 169,48,157,69,3,169,128,157,
72,3,169,0,157,73,3,32,2370
2030 DATA 86,228,192,136,240,40,192,1,
240,3,76,216,32,162,32,169,7970
2040 DATA 11,157,66,3,169,210,157,68,3
,169,48,157,69,3,169,128,5974
2050 DATA 157,72,3,169,0,157,73,3,32,8
6,228,76,80,35,189,72,4598
2060 DATA 3,133,203,189,73,3,133,204,1
62,32,32,172,35,76,201,35,5928
2070 DATA 169,11,157,66,3,169,210,157,
68,3,169,48,157,69,3,165,5520
2080 DATA 203,157,72,3,165,204,157,73,
3,32,86,228,96,192,1,240,8377
2090 DATA 3,76,216,32,162,16,169,12,15
7,66,3,32,86,228,162,32,4948
2100 DATA 169,12,157,66,3,32,86,228,76
,244,35,70,105,108,101,32,4914
2110 DATA 99,111,112,105,101,100,46,15
5,162,0,169,9,157,66,3,169,4993
2120 DATA 231,157,254,35,249,36,68,3,1
69,35,157,69,3,169,13,157,5066
2130 DATA 72,3,169,0,157,73,3,32,86,22
8,76,85,33,76,58,36,2064
2140 DATA 73,110,115,101,114,116,32,11
5,111,117,114,99,101,32,100,105,5144
2150 DATA 115,107,44,32,112,114,101,11
5,115,32,82,69,84,85,82,78,3455
2160 DATA 46,155,162,0,169,9,157,66,3,
169,24,157,68,3,169,36,3430
2170 DATA 157,69,3,169,34,157,72,3,169
,0,157,73,3,32,86,228,4339
2180 DATA 169,255,141,252,2,173,252,2,
201,12,208,249,162,16,169,3,9236
2190 DATA 157,66,3,169,4,157,74,3,169,
0,157,75,3,169,132,157,5713
2200 DATA 68,3,169,5,157,69,3,32,86,22
8,192,1,240,3,76,216,7213
2210 DATA 32,162,16,169,7,157,66,3,169
,210,157,68,3,169,48,157,6554
2220 DATA 69,3,169,96,157,72,3,169,234
,157,73,3,32,86,228,192,8403
2230 DATA 136,240,3,76,216,32,189,72,3
,133,203,189,73,3,133,204,8438
2240 DATA 162,16,169,12,157,66,3,32,86
,228,76,235,36,73,110,115,6137
2250 DATA 101,114,116,32,100,101,115,1
16,105,110,97,116,105,111,110,32,5479
2260 DATA 100,105,115,107,44,32,112,11
4,101,115,115,32,82,69,84,85,3811
2270 DATA 82,78,155,162,0,169,9,157,66
,3,169,197,157,68,3,169,6543
2280 DATA 36,157,250,36,245,37,69,3,16
9,38,157,72,3,169,0,157,4887
2290 DATA 73,3,32,86,228,169,255,141,2
52,2,173,252,2,201,12,208,1439
2300 DATA 249,162,16,169,3,157,66,3,16
9,8,157,74,3,169,0,157,4173
2310 DATA 75,3,169,132,157,68,3,169,5,
157,69,3,32,86,228,162,6034
2320 DATA 16,32,172,35,192,1,240,3,76,
216,32,162,16,169,12,157,6132
2330 DATA 66,3,32,86,228,76,98,37,68,1
17,112,108,105,99,97,116,5792
2340 DATA 105,111,110,32,99,111,109,11
2,108,101,116,101,46,155,162,0,5613
2350 DATA 169,9,157,66,3,169,76,157,68
,3,169,37,157,69,3,169,4790
2360 DATA 22,157,72,3,169,0,157,73,3,3
2,86,228,76,85,33,173,4922
2370 DATA 132,5,141,176,30,162,16,169,
3,157,66,3,169,8,157,74,4432
2380 DATA 3,169,0,157,75,3,169,175,157
,68,3,169,30,157,69,3,4150
2390 DATA 32,86,228,192,1,208,68,162,1
6,169,12,157,66,3,32,86,3677
2400 DATA 228,76,206,37,68,79,83,46,83
,89,83,32,102,105,108,101,4275
2410 DATA 32,119,114,105,116,116,101,1
10,46,155,162,0,169,9,157,66,5785
2420 DATA 3,169,184,157,68,3,169,37,15
7,69,3,169,22,157,72,3,3554
2430 DATA 169,0,157,73,3,32,86,228,76,
85,33,76,216,32,173,132,6767
2440 DATA 5,201,246,37,241,38,56,208,6
8,76,28,38,67,97,110,39,3861
2450 DATA 116,32,119,114,105,116,101,3
2,77,69,77,46,83,65,86,32,2200
2460 DATA 116,111,32,100,114,105,118,1
01,32,56,33,155,162,0,169,9,3984
2470 DATA 157,66,3,169,252,157,68,3,16
9,37,157,69,3,169,32,157,5989
2480 DATA 72,3,169,0,157,73,3,32,86,22
8,76,85,33,141,187,30,5163
2490 DATA 162,16,169,3,157,66,3,169,8,
157,74,3,169,0,157,75,4001
2500 DATA 3,169,186,157,68,3,169,30,15
7,69,3,32,86,228,192,1,5534
2510 DATA 208,102,162,16,169,11,157,66
,3,169,210,157,68,3,169,48,6150
2520 DATA 157,69,3,169,249,157,72,3,16
9,21,157,73,3,32,86,228,5974
2530 DATA 192,1,208,68,162,16,169,12,1
57,66,3,32,86,228,76,167,6417
2540 DATA 38,77,69,77,46,83,65,86,32,1
02,105,108,101,32,119,114,4247
2550 DATA 105,116,116,101,110,46,155,1
62,0,169,9,157,66,3,169,145,6274
2560 DATA 157,68,3,169,38,157,69,3,169
,22,157,72,3,169,0,157,4426
2570 DATA 73,3,32,86,228,76,85,33,76,2
16,32,160,6,162,0,185,5966
2580 DATA 128,5,201,44,240,8,157,252,2
8,200,232,76,207,38,169,155,1814
2590 DATA 157,252,28,162,0,200,185,128
,5,201,44,240,8,157,11,29,5852
2600 DATA 200,232,242,38,237,39,76,230
,38,169,155,157,11,29,162,0,6533
2610 DATA 200,185,128,5,157,18,29,201,
155,240,5,200,232,76,253,38,1021
2620 DATA 162,16,169,3,157,66,3,169,8,
157,74,3,169,0,157,75,4131
2630 DATA 3,169,252,157,68,3,169,28,15
7,69,3,32,86,228,162,16,5636
2640 DATA 169,11,157,66,3,169,34,157,6
8,3,169,29,157,69,3,169,4694
2650 DATA 2,157,72,3,169,0,157,73,3,32
```

COM-DOS

```
,86,228,169,18,133,243,8083
2660 DATA 169,29,133,244,169,0,133,242
,32,0,216,32,210,217,165,212,2657
2670 DATA 133,203,165,213,133,204,32,1
70,217,32,182,221,169,11,133,243,3190
2680 DATA 169,29,133,244,169,0,133,242
,32,0,216,32,210,217,162,16,9496
2690 DATA 169,11,157,66,3,169,212,157,
68,3,169,0,157,69,3,169,5642
2700 DATA 2,157,72,3,169,0,157,73,3,32
,86,228,162,16,169,11,4842
2710 DATA 157,66,3,169,203,157,68,3,16
9,0,157,69,3,169,2,157,5164
2720 DATA 72,3,169,0,157,73,3,32,86,22
8,162,16,165,212,157,68,8389
2730 DATA 3,165,213,157,69,3,32,170,21
7,32,182,221,169,18,133,243,1536
2740 DATA 169,29,133,244,169,0,133,242
,32,0,216,32,96,218,32,210,9242
2750 DATA 217,162,16,230,212,165,212,2
40,2,208,2,230,213,157,72,3,688
2760 DATA 165,213,238,39,233,40,157,73
,3,169,11,157,66,3,32,86,3787
2770 DATA 228,76,40,40,69,110,116,101,
114,32,82,85,78,32,97,100,3840
2780 DATA 100,114,101,115,115,32,40,11
2,114,101,115,115,32,82,69,84,4438
2790 DATA 85,82,78,32,105,102,32,32,32
,32,110,111,110,101,41,155,4107
2800 DATA 162,0,169,9,157,66,3,169,252
,157,68,3,169,39,157,69,6883
2810 DATA 3,169,44,157,72,3,169,0,157,
73,3,32,86,228,162,0,4772
2820 DATA 169,5,157,66,3,169,128,157,6
8,3,169,5,157,69,3,169,5232
2830 DATA 255,157,72,3,169,0,157,73,3,
32,86,228,173,128,5,201,7516
2840 DATA 155,240,58,169,128,133,243,1
69,5,133,244,169,0,133,242,32,907
2850 DATA 0,216,32,210,217,165,212,141
,32,29,165,213,141,33,29,162,9176
2860 DATA 16,169,11,157,66,3,169,28,15
7,68,3,169,29,157,69,3,3442
2870 DATA 169,6,157,72,3,169,0,157,73,
3,32,86,228,162,16,169,6342
2880 DATA 12,157,66,3,32,86,228,76,190
,40,70,105,108,101,32,115,5574
2890 DATA 97,118,101,100,46,155,162,0,
169,9,157,66,3,169,178,157,7937
2900 DATA 68,3,169,40,157,69,3,169,12,
157,72,3,169,0,157,73,4439
2910 DATA 3,32,86,228,76,85,33,162,16,
169,3,157,66,3,169,4,3814
2920 DATA 157,74,234,40,229,41,3,169,0
,157,75,3,76,1,41,68,1987
2930 DATA 58,65,85,84,79,82,85,78,46,8
3,89,83,0,169,243,157,7557
2940 DATA 68,3,169,40,157,69,3,32,86,2
28,192,170,240,80,162,16,9289
2950 DATA 169,12,157,66,3,32,86,228,76
,65,41,253,65,85,84,79,5891
2960 DATA 82,85,78,46,83,89,83,32,102,
105,108,101,32,97,108,114,5002
2970 DATA 101,97,100,121,32,101,120,10
5,115,116,115,33,155,162,0,169,7338
2980 DATA 9,157,66,3,169,31,157,68,3,1
69,41,157,69,3,169,34,4257
2990 DATA 157,72,3,169,0,157,73,3,32,8
6,228,76,85,33,162,16,4274
3000 DATA 169,12,157,66,3,32,86,228,16
2,16,169,3,157,66,3,169,5788
3010 DATA 8,157,74,3,169,0,157,75,3,76
,142,41,68,58,65,85,2982
3020 DATA 84,79,82,85,78,46,83,89,83,0
,169,128,157,68,3,169,5691
3030 DATA 41,157,69,3,32,86,228,160,0,
185,133,5,153,90,29,201,7429
3040 DATA 155,240,4,200,76,157,41,162,
16,169,11,157,66,3,169,36,5242
3050 DATA 157,68,3,169,29,157,69,3,169
,73,157,72,3,169,0,157,5381
3060 DATA 73,3,32,86,228,162,16,169,12
,157,66,3,32,86,228,76,5851
3070 DATA 85,33,162,0,169,11,157,66,3,
169,247,157,68,3,169,41,6680
3080 DATA 157,69,230,41,225,42,3,169,1
47,157,72,3,169,1,157,73,6434
3090 DATA 3,32,86,228,76,85,33,125,32,
76,79,67,75,32,68,58,2540
3100 DATA 102,105,108,101,110,97,109,1
01,46,101,120,116,32,32,32,32,2835
3110 DATA 32,32,68,73,82,110,155,32,85
,78,76,79,67,75,32,68,2931
3120 DATA 58,102,105,108,101,110,97,10
9,101,46,101,120,116,32,32,32,3713
3130 DATA 32,70,79,82,77,65,84,110,155
,32,68,69,76,69,84,69,3719
3140 DATA 32,68,58,102,105,108,101,110
,97,109,101,46,101,120,116,32,5521
3150 DATA 32,32,32,87,68,79,83,110,155
,32,82,69,78,65,77,69,3593
3160 DATA 32,68,58,111,108,100,44,110,
101,119,32,32,32,32,32,32,965
3170 DATA 32,32,32,87,77,69,77,110,155
,32,66,83,65,86,69,32,2961
3180 DATA 68,58,102,105,108,101,44,98,
101,103,105,110,44,101,110,100,5978
3190 DATA 32,32,32,82,69,66,79,79,84,1
55,32,66,76,79,65,68,3243
3200 DATA 32,68,58,102,105,108,101,110
,97,109,101,46,101,120,116,32,5581
3210 DATA 32,32,32,32,71,79,32,97,100,
100,114,155,32,67,79,80,4192
3220 DATA 89,32,68,58,102,105,108,101,
49,44,68,50,58,102,105,108,4227
3230 DATA 101,50,91,47,93,32,66,65,83,
73,67,155,32,68,85,80,3528
3240 DATA 32,68,226,42,221,43,58,102,1
05,108,101,110,97,109,101,46,6333
3250 DATA 101,120,116,32,32,32,32,32,3
2,32,83,84,65,84,85,83,2052
3260 DATA 155,32,66,79,79,84,32,102,10
5,108,101,110,97,109,101,46,5426
3270 DATA 101,120,116,32,32,32,32,32,3
2,32,32,67,76,73,67,75,898
3280 DATA 155,32,63,104,120,110,109,32
,32,32,32,32,32,32,32,32,9583
3290 DATA 32,32,32,32,32,32,32,32,32,3
2,32,72,69,76,80,155,1907
3300 DATA 155,110,61,100,114,105,118,1
01,32,110,117,109,98,101,114,32,5985
3310 DATA 40,49,45,56,41,155,104,120,1
10,109,32,61,32,52,45,100,3213
3320 DATA 105,103,105,116,32,104,101,1
20,32,110,117,109,98,101,114,155,7722
3330 DATA 65,108,108,32,105,110,112,11
7,116,32,77,85,83,84,32,98,4502
3340 DATA 101,32,105,110,32,100,101,99
,105,109,97,108,46,155,169,131,8316
3350 DATA 133,243,169,5,133,244,169,0,
133,242,32,0,216,32,210,217,1655
3360 DATA 108,212,0,162,0,169,11,157,6
6,3,169,197,157,68,3,169,7476
3370 DATA 30,157,69,3,169,14,157,72,3,
169,0,157,73,3,32,86,2985
3380 DATA 228,173,121,7,201,80,240,39,
76,202,43,79,78,155,162,0,7561
3390 DATA 169,9,157,66,3,169,199,157,6
8,3,169,43,157,69,3,169,6763
3400 DATA 3,157,222,43,217,44,72,3,169
,0,157,73,3,32,86,228,5981
3410 DATA 76,16,44,76,242,43,79,70,70,
155,162,0,169,9,157,66,6231
3420 DATA 3,169,238,157,68,3,169,43,15
7,69,3,169,4,157,72,3,4530
3430 DATA 169,0,157,73,3,32,86,228,162
,0,169,11,157,66,3,169,6158
3440 DATA 211,157,68,3,169,30,157,69,3
,169,15,157,72,3,169,0,4136
3450 DATA 157,73,3,32,86,228,173,10,7,
```

```
41,1,240,5,169,49,32,4021
3460 DATA 236,44,173,10,7,41,2,240,5,1
69,50,32,236,44,173,10,5666
3470 DATA 7,41,4,240,5,169,51,32,236,4
4,173,10,7,41,8,240,5395
3480 DATA 5,169,52,32,236,44,173,10,7,
41,128,240,5,169,56,32,5386
3490 DATA 236,44,76,110,44,155,162,0,1
69,9,157,66,3,169,109,157,7448
3500 DATA 68,3,169,44,157,69,3,169,1,1
57,72,3,169,0,157,73,4956
3510 DATA 3,32,86,228,162,0,169,11,157
,66,3,169,226,157,68,3,7166
3520 DATA 169,30,157,69,3,169,54,157,7
2,3,169,0,157,73,3,32,3316
3530 DATA 86,228,173,9,7,133,212,169,0
,133,213,32,170,217,32,230,1761
3540 DATA 216,160,255,200,185,128,5,16
,250,41,127,153,128,5,169,155,139
3550 DATA 153,129,5,162,0,169,9,157,66
,3,169,128,157,68,3,169,6718
3560 DATA 5,157,218,44,213,45,69,3,169
,255,157,72,3,169,0,157,8130
3570 DATA 73,3,32,86,228,76,14,45,141,
26,29,162,0,169,11,157,4978
3580 DATA 66,3,169,26,157,68,3,169,29,
157,69,3,169,2,157,72,5187
3590 DATA 3,169,0,157,73,3,32,86,228,9
6,76,18,45,155,162,0,5103
3600 DATA 169,9,157,66,3,169,17,157,68
,3,169,45,157,69,3,169,5723
3610 DATA 1,157,72,3,169,0,157,73,3,32
,86,228,76,85,45,84,4907
3620 DATA 111,103,103,108,101,32,87,11
4,105,116,101,32,86,101,114,105,6418
3630 DATA 102,121,32,111,110,47,111,10
2,102,32,40,89,47,78,41,63,3011
3640 DATA 155,162,0,169,9,157,66,3,169
,51,157,68,3,169,45,157,6434
3650 DATA 69,3,169,34,157,72,3,169,0,1
57,73,3,32,86,228,162,6999
3660 DATA 0,169,5,157,66,3,169,252,157
,68,3,169,28,157,69,3,5987
3670 DATA 169,255,157,72,3,169,0,157,7
3,3,32,86,228,173,252,28,9078
3680 DATA 201,89,240,10,201,155,208,3,
76,85,33,76,170,45,173,121,8414
3690 DATA 7,73,7,141,121,7,76,201,45,6
7,104,97,110,103,101,32,5497
3700 DATA 97,99,116,105,118,101,32,100
,114,105,118,101,115,32,40,89,5536
3710 DATA 47,78,41,63,155,162,0,169,9,
157,66,3,169,173,157,68,7862
3720 DATA 3,169,214,45,209,46,45,157,6
9,3,169,28,157,72,3,169,6419
3730 DATA 0,157,73,3,32,86,228,162,0,1
69,5,157,66,3,169,252,8939
3740 DATA 157,68,3,169,28,157,69,3,169
,255,157,72,3,169,0,157,7886
3750 DATA 73,3,32,86,228,173,252,28,20
1,89,240,10,201,155,208,3,1845
3760 DATA 76,85,33,76,63,47,76,95,46,8
4,121,112,101,32,116,104,5392
3770 DATA 101,32,100,114,105,118,101,3
2,110,117,109,98,101,114,115,44,6760
3780 DATA 32,111,110,101,32,97,116,32,
97,32,116,105,109,101,44,112,5590
3790 DATA 114,101,115,115,105,110,103,
32,82,69,84,85,82,78,32,97,4635
3800 DATA 102,116,101,114,32,101,97,99
,104,32,111,110,101,46,155,162,7801
3810 DATA 0,169,9,157,66,3,169,25,157,
68,3,169,46,157,69,3,4567
3820 DATA 169,70,157,72,3,169,0,157,73
,3,32,86,228,76,158,46,6378
3830 DATA 69,110,116,101,114,32,48,32,
119,104,101,110,32,121,111,117,6414
3840 DATA 39,114,101,32,102,105,110,10
5,115,104,101,100,46,155,162,0,6872
3850 DATA 169,9,157,66,3,169,128,157,6
8,3,169,46,157,69,3,169,6762
3860 DATA 30,157,72,3,169,0,157,73,3,3
2,86,228,169,0,141,25,5701
3870 DATA 29,162,0,169,5,157,66,3,169,
252,157,68,3,169,28,157,8273
3880 DATA 69,3,210,46,205,47,169,255,1
57,72,3,169,0,157,73,3,6834
3890 DATA 32,86,228,173,252,28,201,48,
240,83,201,49,240,19,201,50,1679
3900 DATA 240,27,201,51,240,35,201,52,
240,43,201,56,240,51,76,57,9593
3910 DATA 47,169,1,24,109,25,29,141,25
,29,76,193,46,169,2,24,3465
3920 DATA 109,25,29,141,25,29,76,193,4
6,169,4,24,109,25,29,141,3999
3930 DATA 25,29,76,193,46,169,8,24,109
,25,29,141,25,29,76,193,4706
3940 DATA 46,169,128,24,109,25,29,141,
25,29,76,193,46,173,25,29,4356
3950 DATA 141,10,7,76,125,47,67,104,97
,110,103,101,32,110,117,109,6417
3960 DATA 98,101,114,32,111,102,32,102
,105,108,101,115,32,116,104,97,6605
3970 DATA 116,32,99,97,110,32,98,101,3
2,32,32,32,111,112,101,110,4701
3980 DATA 32,115,105,109,117,108,116,9
7,110,101,111,117,115,108,121,63,8269
3990 DATA 155,162,0,169,9,157,66,3,169
,66,157,68,3,169,47,157,6964
4000 DATA 69,3,169,59,157,72,3,169,0,1
57,73,3,32,86,228,162,7449
4010 DATA 0,169,5,157,66,3,169,252,157
,68,3,169,28,157,69,3,6337
4020 DATA 169,255,157,72,3,169,0,157,7
3,3,32,86,228,173,252,28,9428
4030 DATA 201,89,240,3,76,85,33,76,252
,47,72,111,119,32,109,97,6914
4040 DATA 110,121,206,47,201,48,32,102
,105,108,101,115,32,100,111,32,6040
4050 DATA 121,111,117,32,119,97,110,11
6,32,111,112,101,110,32,97,116,6778
4060 DATA 32,32,32,32,111,110,101,32,1
16,105,109,101,32,40,49,45,3494
4070 DATA 55,41,63,155,162,0,169,9,157
,66,3,169,198,157,68,3,7055
4080 DATA 169,47,157,69,3,169,54,157,7
2,3,169,0,157,73,3,32,3910
4090 DATA 86,228,162,0,169,5,157,66,3,
169,252,157,68,3,169,28,7902
4100 DATA 157,69,3,169,255,157,72,3,16
9,0,157,73,3,32,86,228,7374
4110 DATA 169,252,133,243,169,28,133,2
44,169,0,133,242,32,0,216,32,106
4120 DATA 210,217,165,212,141,9,7,76,8
5,33,173,219,2,73,255,141,278
4130 DATA 219,2,76,85,33,173,129,5,32,
192,48,133,206,173,130,5,8529
4140 DATA 32,192,48,133,205,32,200,48,
133,213,173,131,5,32,192,48,9196
4150 DATA 133,206,173,132,5,32,192,48,
133,205,32,200,48,133,212,32,9864
4160 DATA 170,217,32,230,216,160,255,2
00,185,128,5,16,250,41,127,153,2574
4170 DATA 128,5,169,155,153,129,5,162,
0,169,9,157,66,3,169,128,7461
4180 DATA 157,68,3,169,5,157,69,3,169,
255,157,72,3,169,0,157,8211
4190 DATA 73,3,32,86,228,76,85,33,56,2
33,48,170,189,109,29,96,8520
4200 DATA 165,206,202,48,209,48,10,10,
10,10,24,101,205,96,224,2,6125
4210 DATA 225,2,179,33,0,0,0,0,0,0,0,0
,0,0,0,0,5108
```

Listing 2: Assembly

```
10 ;COM-DOS   By Robert Berry
20 ;Replacement Disk Utilities
30 ;Package
40 ;
```

COM-DOS

```
50 ;Created using MAC/65 Macros
60 ; and another one, POKE.
70       .INCLUDE #D:SYSEQU.M65
80 LBUFF =    $0580
90 PTR   =    $CB
0100 AFP =    $D800
0110 FSUB =   $DA60
0120 FPI =    $D9D2
0130 FR0 =    $D4
0140 FR1 =    $E0
0150 INBUFF = $F3
0160 CIX =    $F2
0170 IFP =    $D9AA
0180 FASC =   $D8E6
0190 FMOVE = $DDB6
0200 TRAMSZ = $08
0210 RESETV = $E474
0220 REBOOT = $E477
0230 RAMTOP = $6A
0240 DOSLOAD = $15C8
0250 LOLEN = $CB
0260 HILEN = $CC
0270 DRVBYT = $070A
0280 WVERIFY = $0779
0290 SABYTE = $0709
0300 POKMSK = $10
0310 IRQEN = $D20E
0320 CONSOL = $D01F
0330 CDTMV2 = $021A
0340 CDTMA2 = $0228
0350 CHACT = $02F3
0360 NOCLIK = $02DB
0370 COLOR1 = $02C5
0380 COLOR2 = $02C6
0390 FINE =   $026E
0400 LOBYTE = $CD
0410 HIBYTE = $CE
0420      *=  $1CFC
0430 CBUFF
0440      *=  *+15
0450 N1BUFF
0460      *=  *+7
0470 N2BUFF
0480      *=  *+7
0490 DRNUM
0500      *=  *+1
0510 BUFF2
0520      .BYTE "1 "
0530 INIDAT ;         Run address data
0540      .BYTE 224,2,225,2
0550 RUNDAT
0560      .BYTE 0,0
0570 NNN
0580      .BYTE 255,255
0590 PROG
0600      .BYTE 255,255,128,6,188,6
0610      .BYTE 169,2,133,84,162,0
0620      .BYTE 169,9,157,66,3,169
0630      .BYTE 170,157,68,3,169,6
0640      .BYTE 157,69,3,169,1,157
0650      .BYTE 72,3,157,73,3,32
0660      .BYTE 86,228,169,0,133,84
0670      .BYTE 169,12,141,252,2,96
0680      .BYTE 82,85,78,34,68,58
0690 FILEDAT
0700      .BYTE 0,0,0,0,0,0
0710      .BYTE 0,0,46,0,0,0
0720      .BYTE 155,224,2,225,2,128,6
0730 HEX
0740      .BYTE 0,1,2,3,4,5,6,7,8,9,0
0741      .BYTE 0,0,0,0,0,0,10,11,12
0742      .BYTE 13,14,15
0750 CHECK
0760      LDY #0
0770      LDX #0
0780 CMPLOOP
0790      LDA TAB,Y
0800      CMP #EOL     ;End of command?
0810      BEQ GOTCOM   ;Yes!
0820      CMP LBUFF,X ;No, compare next
0830      BNE FINDEND ;branch FINDEND
0840      INX
0850      INY
0860      JMP CMPLOOP
0870 ;
0880 ;Commands are not the same so
0881 ; far, so we must move to next
0882 ; command.
0890 ;
0900 FINDEND
0910      INY
0920      LDA TAB,Y
0930      CMP #EOL     ;End of next comm
and?
0940      BNE FINDEND ;No, do it again
0950      INY          ;Skip over
0960      INY
0970      INY
0980      LDA TAB,Y
0990      CMP #255     ;End reached?
1000      BEQ ENDTAB
1010 ;
1020 ;Not end of table, so reset X-reg
1021 ;and jump back to CMPLOOP
1030 ;
1040      LDX #0
1050      JMP CMPLOOP
1060 ENDTAB
1070       PRINT  0,"Command not suppo
rted!"
1080      JMP MAINLOOP
1090 GOTCOM
1100 ;
1110 ;Goes here if a command is found.
1120 ;
1130      LDA TAB+1,Y
1140      STA PTR
1150      LDA TAB+2,Y
1160      STA PTR+1
1170      JMP (PTR)
1180 ;
1190 ;This is the command table. The
1200 ;format is:
1210 ;.BYTE "COMMAND",EOL
1220 ;.WORD COMMAND
1230 ;
1240 ;A 255 should be placed at the
1250 ;end of the table.
1260 ;
1270 TAB
1280      .BYTE "FORMAT",EOL
1290      .WORD FORMAT
1300      .BYTE "LOCK",EOL
1310      .WORD LOCK
1320      .BYTE "UNLOCK",EOL
1330      .WORD UNLOCK
1340      .BYTE "DELETE",EOL
1350      .WORD DELETE
1360      .BYTE "RENAME",EOL
1370      .WORD RENAME
1380      .BYTE "DIR",EOL
1390      .WORD DIR
1400      .BYTE "BASIC",EOL
1410      .WORD BASIC
1420      .BYTE "REBOOT",EOL
1430      .WORD REBOOT
1440      .BYTE "BLOAD",EOL
1450      .WORD BLOAD
1460      .BYTE "COPY",EOL
1470      .WORD COPY
1480      .BYTE "DUP",EOL
1490      .WORD DUPLICATE
1500      .BYTE "WDOS",EOL
1510      .WORD WDOS
1520      .BYTE "WMEM",EOL
1530      .WORD WMEM
1540      .BYTE "BSAVE",EOL
1550      .WORD BSAVE
1560      .BYTE "BOOT",EOL
1570      .WORD BOOT
1580      .BYTE "HELP",EOL
1590      .WORD HELP
```

```
1600      .BYTE "GO",EOL
1610      .WORD GO
1620      .BYTE "STATUS",EOL
1630      .WORD STATUS
1640      .BYTE "CLICK",EOL
1650      .WORD CLICK
1660      .BYTE "?",EOL
1670      .WORD HEXDEC
1680      .BYTE 255   ;End of table
1690 FMTFILE
1700      .BYTE "D1:",EOL
1710 DIRFILE
1720      .BYTE "D1:*.*",EOL
1730 DIRMSG
1740      .BYTE "Directory - Drive #1"
1741      .BYTE EOL
1750 DOSSYS
1760      .BYTE "D1:DOS.SYS",EOL
1770 MEMSAV
1780      .BYTE "D1:MEM.SAV",EOL
1790 WVERMSG
1800      .BYTE "Write Verify: "
1810 DRIVEMSG
1820      .BYTE "Active Drives: "
1830 MAXFILES
1840      .BYTE "Maximum # of files th"
1841      .BYTE "at can be open   simu"
1842      .BYTE "ltaneously: "
1850 FORMAT
1860      LDA LBUFF+6
1870      CMP #'8
1880      BNE OKFMT
1890       PRINT  0,"Can't format drive
 8 from COM-DOS!"
1900      JMP MAINLOOP
1910 OKFMT
1920      STA FMTFILE+1
1930       XIO  254,1,0,0,FMTFILE
1940      CPY #1
1950      BNE FMTERR
1960      JMP MAINLOOP
1970 FMTERR
1980      JMP ERROR
1990 LOCK
2000       XIO  35,1,0,0,LBUFF+5
2010      CPY #1
2020      BNE LCKERR
2030      JMP MAINLOOP
2040 LCKERR
2050      JMP ERROR
2060 UNLOCK
2070       XIO  36,1,0,0,LBUFF+7
2080      CPY #1
2090      BNE UNLERR
2100      JMP MAINLOOP
2110 UNLERR
2120      JMP ERROR
2130 DELETE
2140       XIO  33,1,0,0,LBUFF+7
2150      CPY #1
2160      BNE DELERR
2170      JMP MAINLOOP
2180 DELERR
2190      JMP ERROR
2200 RENAME
2210       XIO  32,1,0,0,LBUFF+7
2220      CPY #1
2230      BNE RENERR
2240      JMP MAINLOOP
2250 RENERR
2260      JMP ERROR
2270 DIR
2280      LDA LBUFF+3
2290      CMP #EOL
2300      BEQ GETDIR
2310      STA DIRFILE+1
2320      STA DIRMSG+19
2330 GETDIR
2340       OPEN  1,6,0,DIRFILE
2350      CPY #1
2360      BEQ SDIRLOOP
```

```
2370      JMP ERROR
2380 SDIRLOOP
2390       PRINT  0,DIRMSG
2400 DIRLOOP
2410       INPUT  1,LBUFF
2420      CPY #136
2430      BEQ DIRDONE
2440      CPY #1
2450      BEQ PRINTIT
2460      JMP ERROR
2470 PRINTIT
2480       PRINT  0,LBUFF
2490      JMP DIRLOOP
2500 DIRDONE
2510       CLOSE  1
2520      JMP MAINLOOP
2530 ERROR
2540      STY FR0      ;Store error
2550 ;Now close IOCB's 1 & 2
2560       CLOSE  1
2570       CLOSE  2
2580      LDA #0       ;Clear hi byte
2590      STA FR0+1
2600      JSR IFP
2610 ;
2620 ;Error code now in FP format,
2630 ;stored at FR0.
2640 ;
2650      JSR FASC
2660 ;
2670 ;Error code now in ATASCII format
2680 ;in LBUFF.
2690 ;
2700 ;Now find end of string, add
2710 ;carriage return.
2720 ;
2730      LDY #$FF
2740 ELOOP
2750      INY
2760      LDA LBUFF,Y
2770      BPL ELOOP
2780      AND #$7F     ;Un-invert char.
2790      STA LBUFF,Y
2800      LDA #EOL
2810      STA LBUFF+1,Y ;Store EOL
2820 ;
2830 ;Now in ATASCII format in LBUFF
2840 ;with EOL at end of it.
2850 ;
2860       BPUT  0,ERRMSG,12
2870       PRINT  0,LBUFF
2880      JMP MAINLOOP
2890 ;
2900 ;
2910 ;
2920 ERRMSG
2930      .BYTE "I/O Error - "
2940 ;
2950 ;
2960 ;
2970 MAINLOOP
2980      LDA #202
2990      STA COLOR1
3000      LDA #240
3010      STA COLOR2
3020       BPUT  0,EMSG,16
3030       POKE  764,255
3040       INPUT  0,LBUFF
3050      JMP CHECK
3060 EMSG
3070      .BYTE EOL,"Enter command:"
3071      .BYTE EOL
3080 ;
3090 ;
3100 ;
3110 BEGIN
3120      LDA #255
3130      STA FINE
3135 ; Do I/O to enable fine scrolling
3140       OPEN  6,8,0,"E:"
3150       PRINT  0,"KCOM-DOS      Ver.
```

COM-DOS

```
 3.1       02/06/86"
3160      LDA POKMSK
3170      AND #$7F
3180      STA POKMSK
3190      STA IRQEN
3200      LDA # <VBROUT
3210      STA CDTMA2
3220      LDA # >VBROUT
3230      STA CDTMA2+1
3240      LDA #10
3250      STA CDTMV2
3260      JMP MAINLOOP
3270 VBROUT
3280      LDA CHACT
3290      EOR #$03
3300      STA CHACT
3310      LDA #10
3320      STA CDTMV2
3330      RTS
3340 ;
3350 ;
3360 ;
3370 BASIC
3380      LDA TRAMSZ
3390      BEQ NOBAS
3400      JMP RESETV
3410 NOBAS
3420       PRINT  0,"No Cartridge!"
3430      JMP MAINLOOP
3440 BLOAD
3450       OPEN  1,4,0,LBUFF+6
3460      JSR DOSLOAD
3470      JMP MAINLOOP
3480 ;
3490 ;
3500 COPY
3510 ;Format: COPY D1:file,D2:FILE2
3520      LDY #7
3530 CLOOP
3540      LDA LBUFF,Y
3550      CMP #',
3560      BEQ FNDCOMMA
3570      INY
3580      JMP CLOOP
3590 FNDCOMMA
3600      LDA #EOL
3610      STA LBUFF,Y
3620      INY
3630      LDX #0
3640 CLOOP2
3650      LDA LBUFF,Y
3660      STA CBUFF,X
3670      CMP #EOL
3680      BEQ MOVEDONE
3690      INY
3700      INX
3710      JMP CLOOP2
3720 MOVEDONE
3730       OPEN  1,4,0,LBUFF+5
3740      CPY #1
3750      BEQ FRSTOK
3760      JMP ERROR
3770 FRSTOK
3780      LDY #$FF
3790 SLLOOP
3800      INY
3810      LDA CBUFF,Y
3820      CMP #'/      ;Append?
3830      BEQ APPENDIT
3840      CMP #EOL
3850      BNE SLLOOP
3860       OPEN  2,8,0,CBUFF
3870      CPY #1
3880      BEQ SCNDOK
3890      JMP ERROR
3900 APPENDIT
3910      LDA #EOL
3920      STA CBUFF,Y
3930       OPEN  2,9,0,CBUFF
3940      CPY #1
3950      BEQ SCNDOK
3960      JMP ERROR
3970 SCNDOK
3980       BGET  1,ENDCOMDOS,128
3990      CPY #136
4000      BEQ ITSOK
4010      CPY #1
4020      BEQ PUT128
4030      JMP ERROR
4040 PUT128
4050       BPUT  2,ENDCOMDOS,128
4060      JMP SCNDOK
4070 ITSOK
4080      LDA ICBLEN,X
4090      STA LOLEN
4100      LDA ICBLEN+1,X
4110      STA HILEN
4120      LDX #$20
4130      JSR PUTBYTES
4140      JMP CONTCOPY
4150 PUTBYTES
4160      LDA #CPBINR
4170      STA ICCOM,X
4180      LDA # <ENDCOMDOS
4190      STA ICBADR,X
4200      LDA # >ENDCOMDOS
4210      STA ICBADR+1,X
4220      LDA LOLEN
4230      STA ICBLEN,X
4240      LDA HILEN
4250      STA ICBLEN+1,X
4260      JSR CIO
4270      RTS
4280 CONTCOPY
4290      CPY #1
4300      BEQ ITSOK2
4310      JMP ERROR
4320 ITSOK2
4330       CLOSE  1
4340       CLOSE  2
4350       PRINT  0,"File copied."
4360      JMP MAINLOOP
4370 ;
4380 ;DUPLICATE - Will duplicate the
4390 ;file specified after command.
4400 ;
4410 DUPLICATE
4420       PRINT  0,"Insert source disk
, press RETURN."
4430       POKE  764,255
4440 DGLOOP
4450      LDA 764
4460      CMP #12
4470      BNE DGLOOP
4480       OPEN  1,4,0,LBUFF+4
4490      CPY #1
4500      BEQ DUPOK
4510      JMP ERROR
4520 DUPOK
4530       BGET  1,ENDCOMDOS,60000
4540      CPY #136
4550      BEQ DUPOK2
4560      JMP ERROR
4570 DUPOK2
4580      LDA ICBLEN,X
4590      STA LOLEN
4600      LDA ICBLEN+1,X
4610      STA HILEN
4620       CLOSE  1
4630       PRINT  0,"Insert destination
 disk, press RETURN"
4640       POKE  764,255
4650 GETLOOP
4660      LDA 764
4670      CMP #12
4680      BNE GETLOOP
```

(Continued on page 68)

Tutorial

48k disk or cassette

The Magic of Tesselations

by Allan Moose and Marian Lorenz

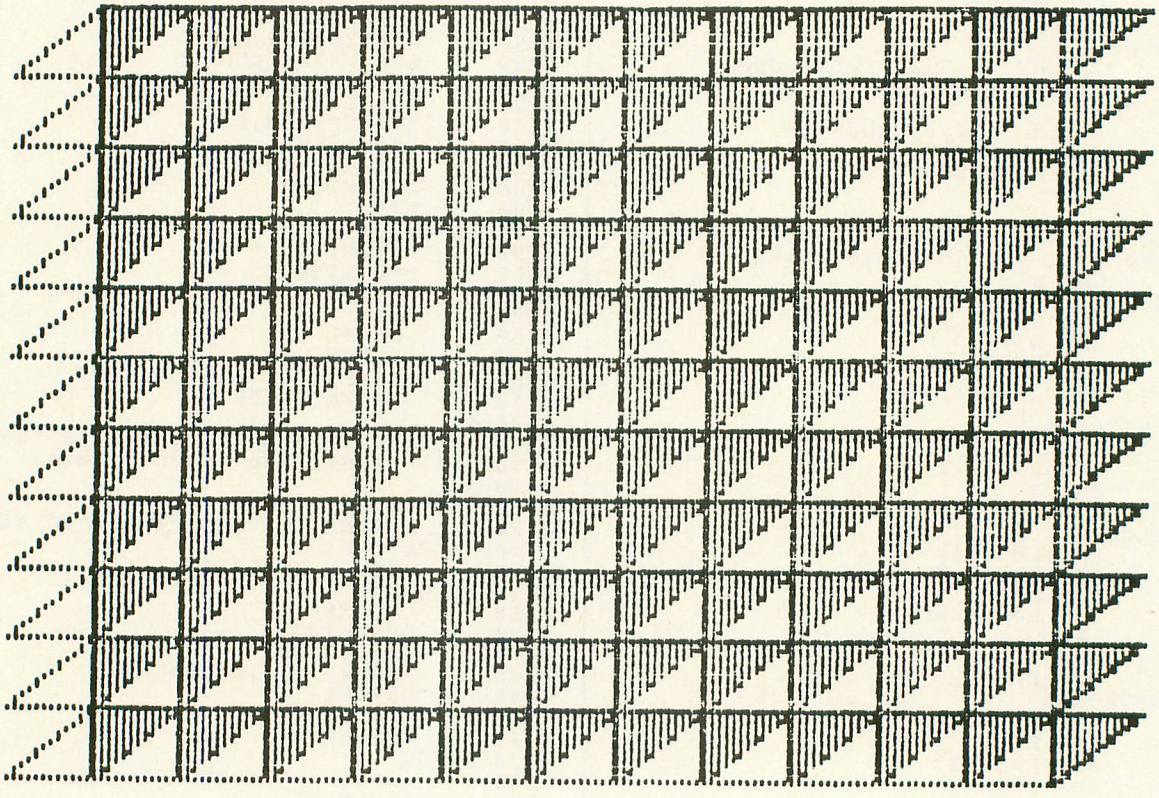


Figure 1

Tesselations may be an unfamiliar word, but you see them every day. They appear in the arrangements of bathroom tiles, linoleum patterns, parquet floors, or fabrics, just to name a few common places where they are found. A tesselation is the complete covering of a flat surface by one or more figures in a pattern where there are no overlapping of the figures and no open spaces. For many years tesselations were studied mainly by mathematicians. However, since the publication of Martin Gardner's "Mathematical Games" columns in *Scientific American* that were devoted to tiling, tesselations have become a pastime for people from all walks of life.

Most of us are familiar with portions of M. C. Escher's work from posters, calendars and jigsaw puzzles. He is one of the most famous graphic artists whose work includes some extremely intricate tesselations. In the commentary accompanying his book, *The Graphic Work of M. C. Escher*, Escher said that "the regular division of the plane . . . is the richest source of inspiration that I have ever struck; nor has it yet dried up." His original inspiration came from a study of the work of the Moors, particularly in the Alhambra in Spain, where the floors and walls were decorated with abstract geometrical tilings. Escher was first and foremost an artist, with no formal training in mathematics. Yet as he worked he found that he often had "more in common with mathematicians than with my fellow artists." This of course can be attributed to the fact that his works often exhibit symmetries of design that tend

The regular division of the plane...*is the richest source of inspiration that I have ever struck.*

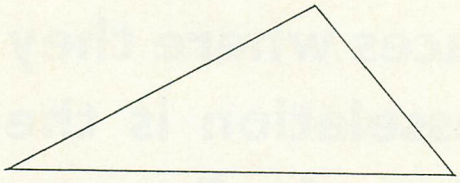
to be more often studied by scientists than by artists.

In this article we will discuss some basic concepts and present several programs that will allow you to experiment with plane tesselations on your computer. In other words, with these programs you can build a graphic composition around a geometrical theme. Since Escher was not bound by religious taboos as the Moors were, many of his tesselations often make use of two or more living figures carefully designed to interlock in such a way that they will tile a flat surface. Here our goal will be more modest – we'll work with polygons.

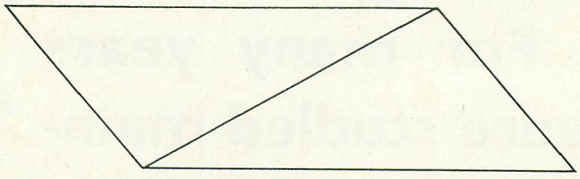
Of the regular polygons (geometric

The Magic of Tess

figures with all sides the same length)–only three, the equilateral triangle, square, and regular hexagon–can be used to make a tesselation. However, if we drop the requirement that the polygons must be regular, then the possibilities increase enormously. There is, in fact, an infinite number of irregular polygons that will tesselate. For example, take any triangle:

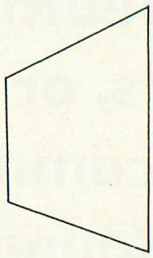


Flip it about one side and mate a pair



The resulting quadrilateral-like figure will tile a plane surface.

Figure 1 is the output from the program in Listing 2 which tiles a surface using triangles. Similarly, any quadrilateral, no matter whether it's convex (all angles less than 180):



or not convex:

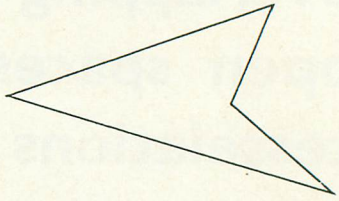
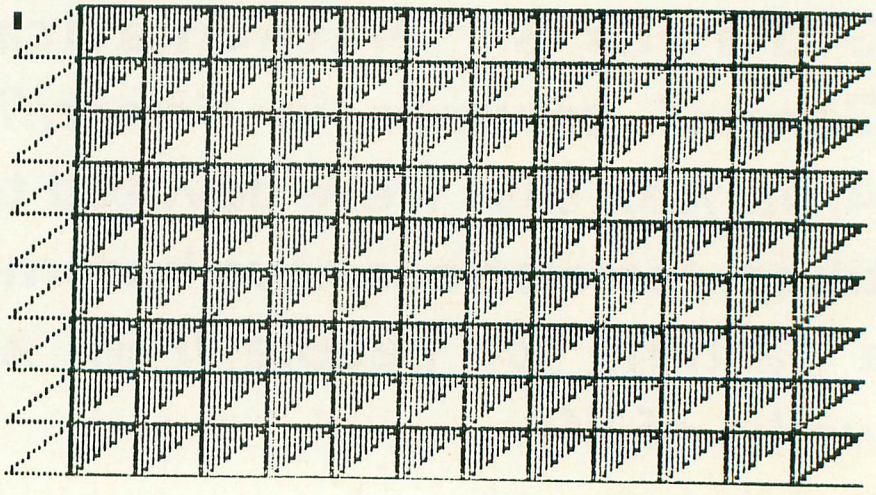
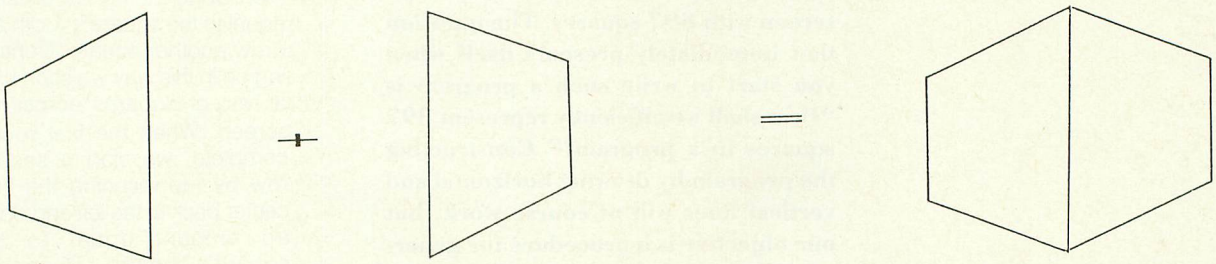


Figure 1



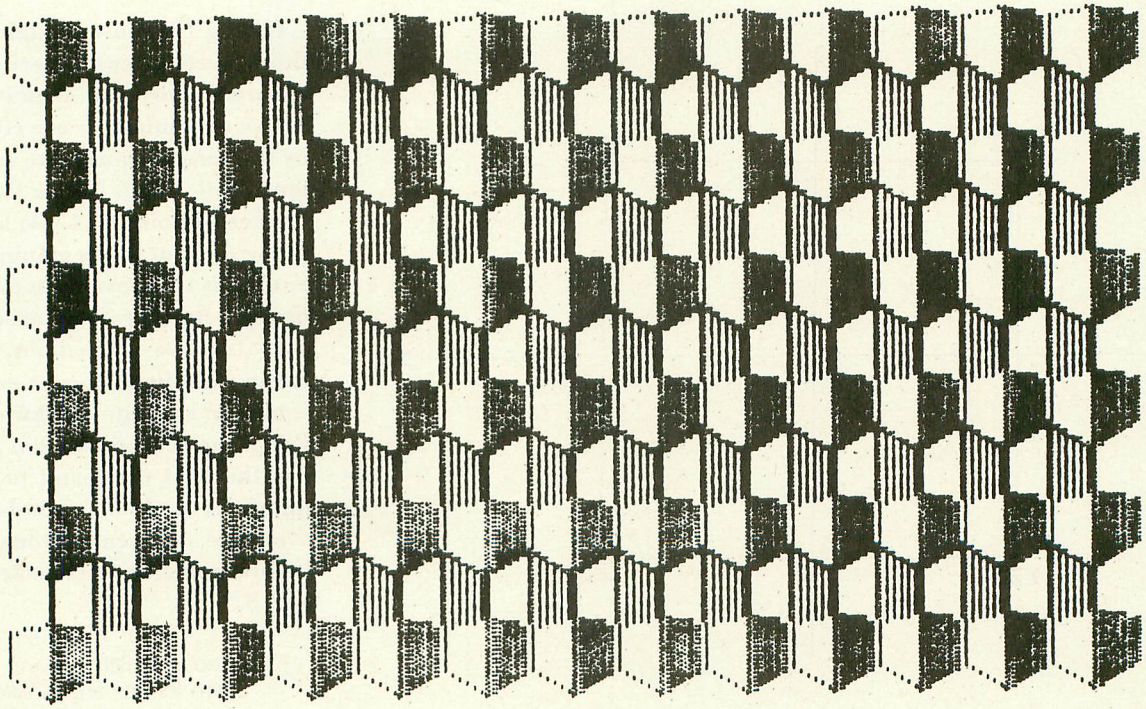
can tile the plane. This idea is similar to what we did with the triangle.



Flip and match

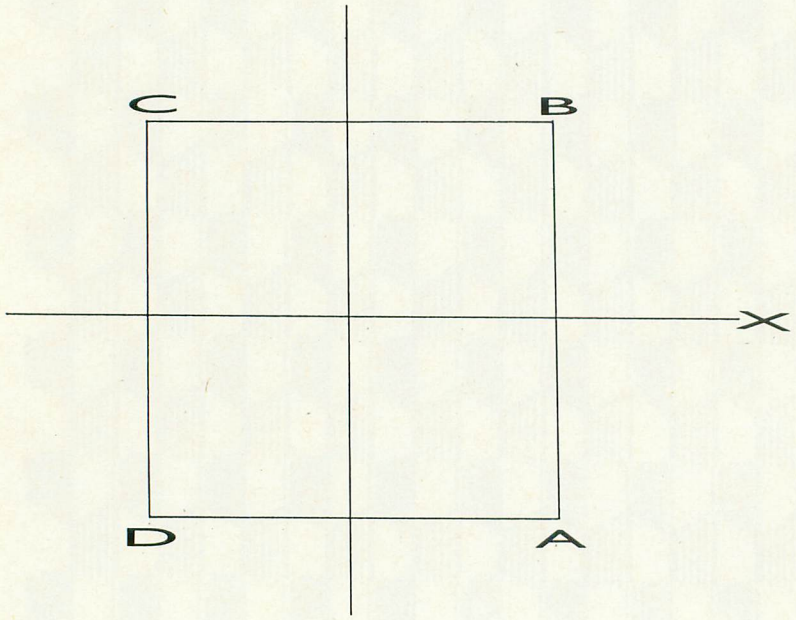
But now when each row of figures is drawn, we have to introduce a new flip. Figure 2, the output from the program in Listing 3, illustrates this.

Figure 2



The Magic of Tess

The purpose of the program in Listing 1 is to illustrate the essential concept of all subsequent tiling programs in as simple and direct a manner as possible. Listing 1 covers a full Graphics 8 screen with 897 squares. The question that immediately presents itself when you start to write such a program is "How shall we efficiently represent 897 squares in a program?" Constructing the program by drawing horizontal and vertical lines will of course work, but our objective is a procedure for generating tilings with arbitrary shapes. Do we have to include data numbers for the coordinates of each corner? (A little arithmetic shows that at the minimum we would need 897 points.) No; the idea is much simpler. Instead of drawing 897 different squares, we draw the "same" square 897 times. The way to do this is to imagine that our square is drawn on an X/Y coordinate system, which we'll call our "Local Coordinate System" (LCS):



The coordinates of the vertices (corners) A, B, C, D in the LCS are all the data we'll need.

Now, imagine that we position the LCS on the screen so that the square fits into the upper left-hand corner and we draw it in. Then we shift the center of the LCS to the right a distance equal to the square's width and draw another square. Continuing on in this way we soon have a row of squares across the screen. When the first row is complete, we start a second row by repositioning the LCS center back at the left and down an amount equal to the square's height. Repeating these steps soon fills in the whole computer screen.

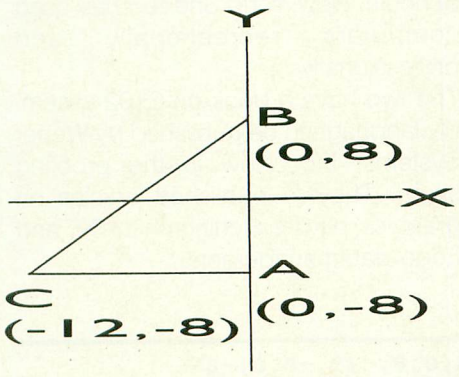
In order to implement these ideas as a program we need three parts:

- Two FOR-NEXT loops to move the local coordinate system's center.
- A routine to convert the position of the LCS's center and the square's coordinate data into CRT screen coordinates.
- A sequence of drawing commands.

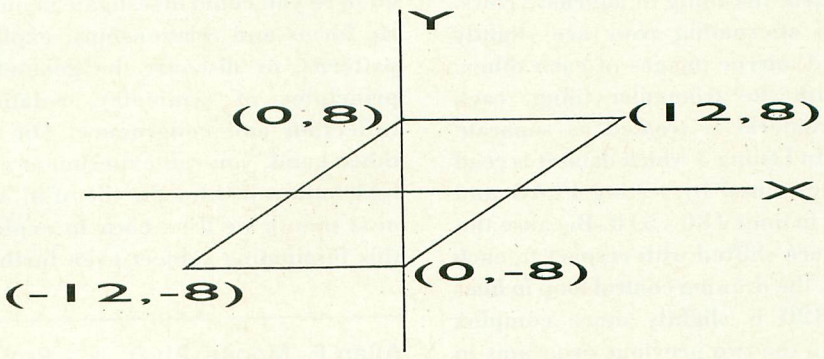
Listing 1 is short enough that each of these parts is easily recognized. The square's coordinate data is in line 50. These data numbers are read and used by the screen coordinate subroutine in lines 260 - 290. Notice that the coordinates of point A (4, -4) appear twice in the data: at the beginning and at the end. This is necessary in order to close up the leftmost column of squares. Also note that we started drawing at the lower right-hand corner and drew the square in a counterclockwise direction. Following this convention allows us to use the XIO command to fill in later tilings. It would be a good idea to take a minute and read through Listing 1 before going on to consider Listings 2 and 3.

Although Listing 2 and Listing 3 are longer and a little more complicated than Listing 1 the basic procedure is the same. The programs are longer because the fundamental title has

to be drawn in several different positions. For example, Listing 2, which tiles the plane in triangles, first draws a triangle like this

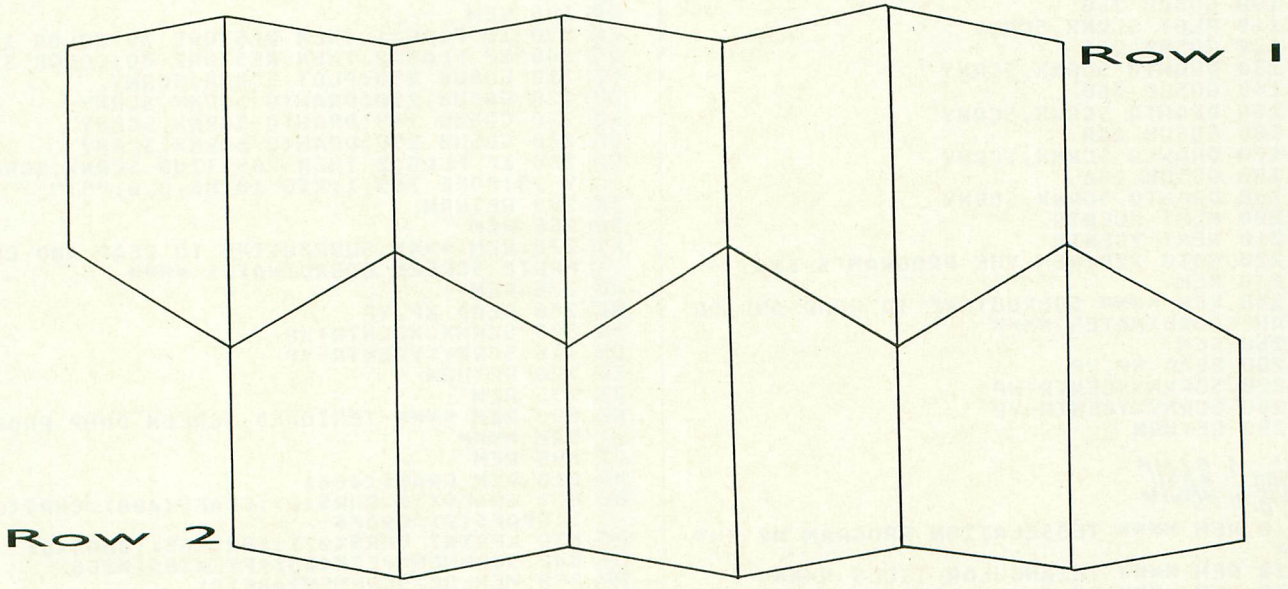


and then puts another of the same size and shape next to it like this



and fills it in with color using the XIO command in line 250. Each triangle is treated as a separate set of data. To control which triangle is drawn, a flag is set in line 110 or 130 and tested in lines 190 and 200. The graphics portion of the program is in lines 60 to 320.

Listing 3 is even more ambitious because it makes a tesselation using a quadrilateral as shown:



The Magic of Tess

Examining the quadrilaterals 1 through 4, we see that they're mated in pairs to form a hexagon. Since a hexagon formed from a pair of the quadrilaterals is irregular, we need to complete the tiling in alternate rows. These alternating rows are slightly shifted mirror images of each other. As with the triangular tiling, each quadrilateral is treated as separate data. In Listing 3 which dataset is read is determined by testing FLAG and ROW in lines 480 - 510. Because the rows are shifted with respect to each other, the drawing control loop in lines 160-320 is slightly more complex than in the two previous programs in that we can no longer use FOR-NEXT loops. However, the basic drawing procedure is the same.

By using the ideas we have presented here you could investigate geometric forms and relationships, explore patterns, or discover the geometric principles of symmetry, rotation, reflection and congruence. On the other hand, you can experiment with tesselations just for the tile of it! And next month we'll be back to explore this fascinating subject even further.

Allan E. Moose, Ph.D., is a **Professor of Mathematics and Physics** at Long Island University, Southampton campus, New York. Uses computers both recreationally and professionally.

Marian J. Lorenz, B.S., M.S., is **Preschool Leader** for the **Special Education Program** at Central Islip Public Schools, New York. She too has used computers recreationally and professionally.

The two have a book on 6502 assembly language to be published by Weber Systems, Inc., now in the proofing stage. They've published articles on graphics, bank-switching on the XE and Logo datamanagement.

Listing 1: BASIC

```
QZ 10 REM **** TESSELATION PROGRAM #1 ***
   *
IR 20 REM **** SQUARE TILES ****
OX 30 REM **** BY ALLAN MOOSE AND MARIAN
   LORENZ ****
RU 40 GRAPHICS 24:COLOR 1
QK 50 DATA 4,-4,4,4,-4,4,-4,4,-4
YQ 60 REM **** DRAWING CONTROL LOOP ****
RT 70 FOR YCENTR=4 TO 186 STEP 8
LR 80 FOR XCENTR=4 TO 314 STEP 8
KM 90 RESTORE 50
TR 100 GOSUB 260
WG 110 PLOT SCRNX,SCRNY
TV 120 GOSUB 260
PG 130 DRAWTO SCRNX,SCRNY
TZ 140 GOSUB 260
PK 150 DRAWTO SCRNX,SCRNY
UD 160 GOSUB 260
PO 170 DRAWTO SCRNX,SCRNY
UH 180 GOSUB 260
PS 190 DRAWTO SCRNX,SCRNY
VF 200 NEXT XCENTR
VR 210 NEXT YCENTR
VT 220 GOTO 220:REM THE PROGRAM'S END
QT 230 REM
JY 240 REM **** SUBROUTINE TO READ AND DR
   AW COORDINATES ****
QX 250 REM
NI 260 READ XP,YP
TD 270 SCRNX=XCENTR+XP
VZ 280 SCRNY=YCENTR-YP
ZQ 290 RETURN
```

Listing 2: BASIC

```
SH 10 REM **** TESSELATION PROGRAM #2 ***
   *
AY 20 REM **** TRIANGULAR TILES ****
OX 30 REM **** BY ALLAN MOOSE AND MARIAN
   LORENZ ****
ON 40 PRINT "{":PRINT "LOADING SCREEN DUM
   P..."
MH 50 GOSUB 1000
PL 60 GRAPHICS 31:POKE 708,46:POKE 709,19
   2:POKE 712,60
FF 65 REM **** COORDINATE DATA ****
GM 70 DATA 0,-8,0,8,-12,-8,0,-8
MU 80 DATA 0,-8,12,8,0,8,0,-8
ZC 85 REM **** DRAWING CONTROL LOOP ****
VV 90 FOR YCENTR=8 TO 176 STEP 16
QI 100 FOR XCENTR=12 TO 144 STEP 12
WL 110 FLAG=1
UU 120 GOSUB 190
WZ 130 FLAG=2
UY 140 GOSUB 190
VO 150 NEXT XCENTR
WA 160 NEXT YCENTR
FU 170 GOTO 810:REM GOTO THE SCREEN DUMP
RP 175 REM
ER 180 REM **** SUBROUTINE TO DRAW TRIANG
   LES ****
RR 185 REM
ZY 190 IF FLAG=1 THEN RESTORE 70:COLOR 1
DS 200 IF FLAG=2 THEN RESTORE 80:COLOR 3
WZ 210 GOSUB 290:PLOT SCRNX,SCRNY
UB 220 GOSUB 290:DRAWTO SCRNX,SCRNY
UD 230 GOSUB 290:DRAWTO SCRNX,SCRNY
UF 240 GOSUB 290:DRAWTO SCRNX,SCRNY
VD 250 IF FLAG=2 THEN POSITION SCRNX,SCRN
   Y-15:POKE 765,1:XIO 18,#6,0,0,"S:"
ZK 260 RETURN
RO 265 REM
KT 270 REM **** SUBROUTINE TO READ AND CO
   MPUTE SCREEN COORDINATES ****
RQ 275 REM
NO 290 READ XP,YP
SQ 300 SCRNX=XCENTR+XP
VM 310 SCRNY=YCENTR-YP
ZD 320 RETURN
RZ 795 REM
XX 800 REM **** TEXTURED SCREEN DUMP PROG
   RAM ****
RI 805 REM
NR 810 DIM GRAF$(400)
BU 820 GRAF$(1)=CHR$(0):GRAF$(400)=CHR$(0
   ):GRAF$(2)=GRAF$
MC 830 LPRINT CHR$(27);CHR$(65);CHR$(8)
GY 840 SCRNMEM=PEEK(88)+PEEK(89)*256
MB 850 MEMLOC=SCRNMEM+40*191
DY 860 POKE 203,0:POKE 204,0:POKE 205,0
AT 870 FOR SCRNCOL=MEMLOC TO MEMLOC+39
CU 880 HIBYTE=INT(SCRNCOL/256)
BV 890 LOBYTE=SCRNCOL-HIBYTE*256
LZ 900 POKE 208,LOBYTE:POKE 209,HIBYTE
ZS 910 GRHI=INT(ADR(GRAF$)/256)
MU 920 GRLO=ADR(GRAF$)-GRHI*256
OV 930 POKE 206,GRLO:POKE 207,GRHI
```

elations

```
DN 940 DUMP=USR(1536)
QU 950 LPRINT CHR$(27);CHR$(76);CHR$(144)
   ;CHR$(1);GRAF$
JX 960 NEXT SCRNCOL
OM 970 END
IO 1000 RESTORE 1030
YO 1010 FOR I=1536 TO 1697:READ ML:POKE I
   ,ML:NEXT I
AF 1020 RETURN
TQ 1030 DATA 104,169,192,133,203,160,0,17
   7,208,170,41,192,201,128,208,9,133,204
   ,169,64,133,205,76,29,6
FW 1040 DATA 133,204,133,205,138,41,48,20
   1,32,208,13,5,204,133,204,169,16,5,205
   ,133,205,76,57,6,5,204
DO 1050 DATA 133,204,5,205,133,205,138,41
   ,12,201,8,208,13,5,204,133,204,169,4,5
   ,205,133,205,76,85,6
QS 1060 DATA 5,204,133,204,5,205,133,205,
   138,41,3,201,2,208,13,5,204,133,204,16
   9,1,5,204,133,204
QC 1070 DATA 76,113,6,5,204,133,204,5,205
   ,133,205,230,206,208,2,230,207,165,204
   ,145,206,230,206,208,2,230,207
SF 1080 DATA 165,205,145,206,169,0,133,20
   4,133,205,198,203,240,18,216
TQ 1090 DATA 56,165,208,233,40,133,208,14
   4,3,76,5,6,198,209,76,5,6,96
```

Listing 3: BASIC

```
TP 10 REM **** TESSELATION PROGRAM #3 ***
   *
IK 20 REM **** QUADRILATERAL TILES ****
OX 30 REM **** BY ALLAN MOOSE AND MARIAN
   LORENZ ****
ON 40 PRINT "{CLEAR}":PRINT "LOADING SCREEN DUM
   P..."
MH 50 GOSUB 1000
ID 60 GRAPHICS 31:POKE 712,10:POKE 708,88
   :POKE 710,24
IU 70 REM **** DATA FOR THE FIRST ROW ***
   *
LQ 80 DATA -6,-4,0,-10,0,8,-6,6,-6,-4
QE 90 DATA 0,-10,6,-4,6,6,0,8,0,-10
IF 100 REM **** DATA FOR SECOND ROW ****
PI 110 DATA -6,4,0,10,0,-8,-6,-6,-6,4
WL 120 DATA 0,-8,6,-6,6,4,0,10,0,-8
TA 130 YCENTR=12:REM INITIALIZE Y-POSITIO
   N
IO 140 REM **** DRAWING CONTROL LOOP ****
KF 150 REM **** DRAW ROW #1 ****
DL 160 FOR XCENTR=6 TO 150 STEP 12
CZ 170 FLAG=1:ROW=1
TU 180 GOSUB 340
XL 190 FLAG=2
TF 200 GOSUB 340
VH 210 NEXT XCENTR
KY 220 REM **** DRAW ROW #2 ****
ZR 230 YCENTR=YCENTR+14:IF YCENTR>164 THE
   N GOTO 810
LC 240 REM **** DRAW ROW #2 ****
NY 250 FOR XCENTR=12 TO 150 STEP 12
DO 260 FLAG=1:ROW=2
TT 270 GOSUB 340
XK 280 FLAG=2
TX 290 GOSUB 340
VG 300 NEXT XCENTR
ZO 310 YCENTR=YCENTR+14:IF YCENTR>164 THE
   N GOTO 810
KV 320 GOTO 160:REM LOOP BACK TO DRAW ANO
   THER ROW
RH 325 REM
DK 330 REM **** MAIN DRAWING SUBROUTINE *
   ***
RJ 335 REM
PW 340 GOSUB 480:GOSUB 540
WQ 350 PLOT SCRNX,SCRNY
UO 360 GOSUB 540
PQ 370 DRAWTO SCRNX,SCRNY
US 380 GOSUB 540
PU 390 DRAWTO SCRNX,SCRNY
UD 400 GOSUB 540
PF 410 DRAWTO SCRNX,SCRNY
UH 420 GOSUB 540
PJ 430 DRAWTO SCRNX,SCRNY
VZ 440 IF FLAG=2 AND ROW=1 THEN POSITION
   SCRNX,SCRNY-17:POKE 765,2:XIO 18,#6,0,
   0,"S:"
UN 450 IF FLAG=2 AND ROW=2 THEN POSITION
   SCRNX,SCRNY-17:POKE 765,1:XIO 18,#6,0,
   0,"S:"
ZM 460 RETURN
ZC 470 REM **** CHOOSE CORRECT DATA NUMBE
   RS ****
RS 475 REM
VI 480 IF FLAG=1 AND ROW=1 THEN RESTORE 8
   0:COLOR 1
ZE 490 IF FLAG=2 AND ROW=1 THEN RESTORE 9
   0:COLOR 2
GN 500 IF FLAG=1 AND ROW=2 THEN RESTORE 1
   10:COLOR 3
GT 510 IF FLAG=2 AND ROW=2 THEN RESTORE 1
   20:COLOR 2
ZF 520 RETURN
RJ 525 REM
KO 530 REM **** SUBROUTINE TO READ AND CO
   MPUTE SCREEN COORDINATES ****
RL 535 REM
NH 540 READ XP,YP
TC 550 SCRNX=XCENTR+XP
VY 560 SCRNY=YCENTR-YP
ZP 570 RETURN
RZ 795 REM
WR 800 REM **** TEXTURED SCREEN DUMP ****
RI 805 REM
NR 810 DIM GRAF$(400)
BU 820 GRAF$(1)=CHR$(0):GRAF$(400)=CHR$(0
   ):GRAF$(2)=GRAF$
MC 830 LPRINT CHR$(27);CHR$(65);CHR$(8)
GY 840 SCRNMEM=PEEK(88)+PEEK(89)*256
MB 850 MEMLOC=SCRNMEM+40*191
DY 860 POKE 203,0:POKE 204,0:POKE 205,0
AT 870 FOR SCRNCOL=MEMLOC TO MEMLOC+39
CU 880 HIBYTE=INT(SCRNCOL/256)
BV 890 LOBYTE=SCRNCOL-HIBYTE*256
LZ 900 POKE 208,LOBYTE:POKE 209,HIBYTE
ZS 910 GRHI=INT(ADR(GRAF$)/256)
MU 920 GRLO=ADR(GRAF$)-GRHI*256
OV 930 POKE 206,GRLO:POKE 207,GRHI
DN 940 DUMP=USR(1536)
QU 950 LPRINT CHR$(27);CHR$(76);CHR$(144)
   ;CHR$(1);GRAF$
JX 960 NEXT SCRNCOL
OM 970 END
RK 980 REM
RX 990 REM **** MACHINE LANGUAGE DATA ***
   *
IO 1000 RESTORE 1030
YO 1010 FOR I=1536 TO 1697:READ ML:POKE I
   ,ML:NEXT I
AF 1020 RETURN
TQ 1030 DATA 104,169,192,133,203,160,0,17
   7,208,170,41,192,201,128,208,9,133,204
   ,169,64,133,205,76,29,6
FW 1040 DATA 133,204,133,205,138,41,48,20
   1,32,208,13,5,204,133,204,169,16,5,205
   ,133,205,76,57,6,5,204
DO 1050 DATA 133,204,5,205,133,205,138,41
   ,12,201,8,208,13,5,204,133,204,169,4,5
   ,205,133,205,76,85,6
QS 1060 DATA 5,204,133,204,5,205,133,205,
   138,41,3,201,2,208,13,5,204,133,204,16
   9,1,5,204,133,204
QC 1070 DATA 76,113,6,5,204,133,204,5,205
   ,133,205,230,206,208,2,230,207,165,204
   ,145,206,230,206,208,2,230,207
SF 1080 DATA 165,205,145,206,169,0,133,20
   4,133,205,198,203,240,18,216
TQ 1090 DATA 56,165,208,233,40,133,208,14
   4,3,76,5,6,198,209,76,5,6,96
```

DATABAS

by Michael A. Banks

One of the most pragmatically useful services on **DELPHI** is electronic mail, or **E-mail. ANALOG's ATARI SIG** provides direct access to **DELPHI**'s electronic mail system, which happens to be one of the most powerful available anywhere.

There are two ways to get to mail from the **ATARI SIG**. The first is obvious, because it's a selection on the **SIG** menu: simply type **MAIL** (the selection is "MAIL (Electronic)"). The other route involves an "invisible" menu item, **DELPHI** Mail. I'll cover both of these in this column.

E-Mail

When you type **MAIL**, you enter **DELPHI**'s electronic mail (**E-mail**) system. Basic E-mail operations include sending, reading, and forwarding messages, but **DELPHI E-mail** doesn't stop with the basics. There are a variety of enhancements you can use with these basic operations, including—among many others—sending the same message to several people at once, forwarding messages to other **DELPHI** members, and nonstop display of messages for fast download.

You can also send files from your personal Workspace (this is convenient when long messages are involved), and messages you read can be copied to Workspace files. Several customizing features are available, among them automatic carbon copies, a "personal name" that appears next to your membername in the message you send, and automatic message forwarding.

DELPHI E-mail sports some sophisticated message filing and handling features too. Need to keep messages on related topics together? Use E-mail's powerful filing system to create named folders and move or copy messages among them. Want to delete individual messages, or all the messages you've read? One simple command does it. (And, if you accidentally delete the wrong messages, you can get them back!)

Here's a quick-reference summary of the E-mail commands you'll use most:

SEND

Type **SEND**, and **DELPHI** prompts you to enter the membername of the person to receive a message (To:), and the subject of the message (Subj:). Enter these and press <**RETURN**>, and the system prompts you to enter the message. Type your message (or send a file as an **ASCII** upload—7-bit text only) and ^Z to send it. (Enter ^C at any point to cancel the message.)

You can send the same message to multiple users by

E DELPHI

entering all of their names, separated by commas, at the To: prompt.

To send a file from your Workspace, simply type **SEND** <filename>, and you'll be prompted for the addressee and subject. After you enter this information, **DELPHI** sends a copy of the specified file to the addressee.

DIRECTORY Displays the headers of the messages in the current folder. (**DIR** <folder name> lists the messages in the specific folder, and makes that the current folder.)

Use **DIR/FOLDERS** to see a list of available folders. The first time you try this, you'll find that one or two folders already exist: **MAIL**, which contains all messages that you've read and not deleted, and **NEWMAIL**, which contains unread mail. (If you have read some new mail and deleted it, you'll find a third folder, called **WASTEBASKET**. This is a temporary file that is automatically created to store deleted mail until you leave mail. It is purged and deleted when you leave E-mail.) It's worth noting here that you don't have to use the folder system if you don't want to. When you enter E-mail and have new messages, the **NEWMAIL** folder is always the current folder; when you enter E-mail and have no new messages, **MAIL** is always the current folder. The only time you need concern yourself with folders is when you wish to **MOVE** or **COPY** a message to a new folder, as explained below.

READ (or press <RETURN>) Displays the next message, one screen at a time, with "More" prompts. **READ** followed by a message number displays the designated message, as does entering the number alone.

FORWARD Sends a copy of the current message to a designated member or members. (You are prompted for an addressee and subject, as when you **SEND** a message.)

MOVE Moves the current message to a designated folder. You can simply type **MOVE**, in which case you will be prompted for the name of the folder, or **MOVE** <name of folder>. (The **MOVE**d message is deleted from the current folder.)

If the folder you specify does not exist, you'll be asked if you wish it to be created. Answer yes and **DELPHI** creates the named folder and moves the message to it.

COPY Copies the current message to a designated folder. Usage and options are the same as for **MOVE**, except the **MOVE**d message is not deleted from the current folder.

EXTRACT Copies the current message to a designated file in your personal Workspace. Usage and options are the same as for **MOVE**, but the message is not deleted.

Hint: If you wish to view or download a long message nonstop and without the "More" prompts, type **EXTRACT TT**. This in effect "extracts" the message to your screen (TT). If you want to display all of your messages nonstop, type **EXTRACT /ALL TT**.

SELECT Moves you to the designated folder, where all commands operate only on the messages in that folder.

DELETE Erases the current message. (Actually, it moves the message to the **WASTEBASKET** folder.) To delete all the messages in the current folder, type **DELETE/ALL**. (This is a command that you may use

often, since you are charged for Workspace storage beyond a certain minimum, and your Mail files are part of your Workspace.)

COMPRESS This command reduces the size of the Mail files in your Workspace—something you'll want to do if you keep messages online, in folders or not.

To use this command, simply type **COMPRESS** at the **MAIL>** prompt, then wait a few seconds while **DELPHI** creates a new, smaller Mail file. Then, exit Mail, go to your personal Workspace, and type **DELETE MAIL.OLD**.

HELP Type **HELP** to see a general overview of the E-mail system. Type **HELP** followed by a command (example: **HELP SEND**) to see specific information on using that command.

EXIT Exit mail and return to the menu from which you entered it. (^Z has the same effect.)

You can use several commands in sequence to perform important tasks. For instance, if you accidentally delete a message, use **SELECT WASTEBASKET** to move to the folder containing deleted messages. Once there, use **DIRECTORY** to find the message you wish to recover, **READ** it, then use **MOVE MAIL** to move it to your Mail file.

D-mail

Typing **DELPHI** takes you into the **DELPHI Mail** (or **D-mail**) system. D-mail offers a gateway to E-mail (type **MAIL**), and much more. For openers, there's Telex service, which gives you E-mail access to more than 1.7 million Telex terminals worldwide! (You can also receive Telex messages from any Telex user anywhere.) Type **HELP TELEX** for more information.

The D-mail menu also offers direct access to your personal Workspace, a language translation service, and some important Mail utilities, as shown on the menu below.

MAIL Menu:

BATCH Mailthru
CATALOG of Mail Files
EXIT
HELP
TII Translation Services
MAIL (Electronic)
SCAN for New Messages
TELEX/Easylink
Workspace
SetMail

DMAIL>(Mail, TII, Telex)

What's New on DELPHI

New Business Offerings

DELPHI's Business & Finance menu carries two recent additions: **CD Infoline** and **Mutual Funds Rates**. CD Infoline presents a daily update of the 20 U.S. banks with the highest "jumbo CD" interest rates. There is a surcharge of $1.75 per report. Mutual Funds Rates provides quotes for some 1,700 mutual funds. A surcharge of 7¢ per quote is levied. Type **BUSINESS** at the **DELPHI** Main Menu for access to these services.

Alternate News Source

An interesting new feature on **DELPHI's NEWS-WEATHER-SPORTS** menu is **Views On News**. Hosted by Ellen Kaufman, manager of the **Micro Artists SIG**, it features regular contributions by several well-known **DELPHI** personalities. Primarily a forum for discussing current events, Views On News also offers features like Bob Fried's "Articles of Lasting Strangeness" (and they are). Type **NEWS VIEWS** to sample Views On News.

Tuesday Night (and other) Realtime Conferences

A reminder: **ANALOG'S ATARI SIG** hosts a realtime conference each Tuesday at 10 p.m., EST. You'll find the conferences an excellent venue for sharing information about Atari computers, getting answers to questions, and participating in friendly discussions of all types.

If you've a yen to chat online, though, you don't have to wait for Tuesday to roll around. Type **WHO** whenever you enter the **SIG**; the system will tell you if anyone is in conference (and chances are very good there will be a conference going on). Even if there are no conferences going on when you enter the **SIG**, you can probably start one by going to the conference area and typing /**PAGE** followed by one of the membernames listed when you typed **WHO**.

In addition to having published science fiction novels and books on rocketry, Michael A. Banks is the author of **DELPHI: The Official Guide and The Modem Book**, both from Brady Books. To order **DELPHI:** The Official Guide, type **GO GUIDE** at any **SIG** prompt.

Look for his articles on telecommunications and using **DELPHI** in the **ATARI SIG** databases. You can contact Banks on **DELPHI** by sending E-mail to membername **KZIN**.

Rapid Swap

by Matthew J.W. Ratcliff

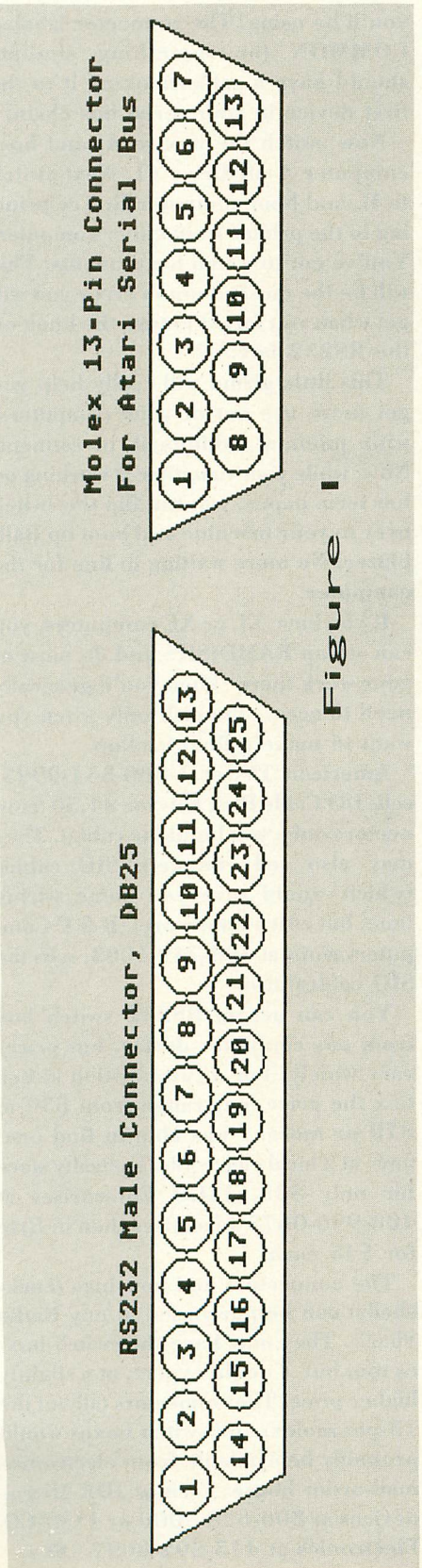


Figure 1

Many of us own several 8-bit Ataris for the simple reason that we want to keep up with the new technology. At the same time, it's pointless to sell the old equipment for a fraction of its original value. What to do? It's a shame to allow the extra computers around your house to go to waste, especially if you have children, a wife, husband, cat, or whomever, that vies with you for computer time on the one machine that is in use.

You could always hook up your extra computer to an old TV, and then juggle cables between the two Ataris. Not many people can afford, or justify, owning an extra set of peripherals (disk drive, printer, interface, etc.). But the hassle of juggling cables just isn't worth it, and is not good for the computers if you do so with the power on (a near necessity, if two people are to use the same disk drives frequently).

There *must* be a better way—and there is. I call it Rapid Swap. The best news is that it is inexpensive, and easy to build.

It would be handy to have a toggle switch that would automatically swap your entire serial bus full of Atari peripherals between systems. The problem is that it is tough to find a 13-pole toggle switch! Even those multipole rotary switches, like the ones found in RS232 switch boxes, are hard to come by.

Hey! That's it! Why not use an RS232 switch box? They're affordable, less than a $50 mail order. Oh, but the connectors aren't the 13-pin serial bus type that Ataris use.

Well, that's where the work begins. You'll need three spare serial bus cables (for two-computer swapping). If you want to get a 4 or 5 position switch box, for sharing as many computers, you'll have to make more cables.

The connectors at the rear of an RS232 data switch box are 25-pin D-type females. You will need to purchase three male connectors and appropriate assembly hardware. You'll also need the usual electronics-kit building tools, such as a good low-wattage soldering iron, wire stripper, solder, wire cutters, a small screwdriver, and a multimeter for measuring continuity. You'll need an RS232 switch box also. Take your Atari serial bus cables and cut one end off each. You can try to save yourself a few bucks by cutting one in the middle, but you're likely to come up short, requiring your computers to sit closer together than you'd like.

Each cable might have the same color wires, or they may be completely different. (Murphy's law dictates that it is the latter, since that means more work for you. Even if they do have the same color codes, double check each.) You'll need to check where each wire goes in the 13-pin connector at the opposite end. Set your meter on kilo-ohms or continuity checking. (Many have the latter, where you will hear a small beep if the test probes encounter a connection.)

See Figure 1 for appropriate pin numbering. Write down the consecutive pin numbers alongside the color wires they're associated with. There should be 13 unique colored wires (or solid with stripes). You may find that they're paired. The cables are usually lined with aluminum foil, and around that is twisted and uninsulated wire, called the shield. (This lies between the outer insulation of the cable and the wire bundles within.) There may be only six unique colors, each having a solid black wire twisted around it. If so, keep them twisted until ready to solder. If they get separated, you'll have to go back with the meter and check their routing again.

Note what type of backshells, or hoods your 25-pin connectors have. If they're two-piece, you can set them aside until everything else is done. If they're one piece, then you'll have to slide then onto the cable first. Next strip back the black outer insulation on the Atari cables about one to two inches. Treat the shield with care, this is soldered to pin 10 of the 13-pin molex connector (and will go to the same pin number on the 25-pin connector).

Strip back the insulation on each wire about 1/16th to 1/8th of an inch and warm up the soldering iron. Solder wires 1-13 from the 13-pin connector to the same numbered pins. If the exposed wire is short enough, you won't need to do any extra insulating when you've soldered it in place. You might want to put shrink tubing over the solder tabs and wires to be safe—if you aren't adept at soldering. Refer to Figure 1 and your colored-wire list frequently. Note that the pin numbering is shown as if you were looking at the business end of the connector (the part that plugs into the computer or switch box).

When you're done with each cable, use your meter to double check it. Make sure that pin 1 of the 13-pin molex connector goes to pin 1 of the RS232 connector, and so on. Double check them, triple check even. You don't want to cross any wires!

When all your cables are completed, it's time for testing. Plug the RS232 connectors into A and B (or 1 and 2, however your switchbox is labled) on the box. Route the other ends to the SIO bus connector on each computer you'll be using. The connector labeled COMMON (or something similar) should have a cable hooking it to the first device in your serial bus chain.

Now switch the box to A, and boot computer A off Drive #1. Next switch to B, and boot computer B. Try printing to the printer from either computer. You've got to watch for timeouts. This will be the most common error you will get when you forget to turn the knob on the RS232 box.

This little gizmo will really help you get more use out of your computers, with minimal additional investment. Now, while your daughter is working on her term paper, you can flip the switch over to your machine and boot up Ballblazer! No more waiting in line for the computer!

If you have XL or XE computers, you can set up RAMDISKS and do most of your work there. Then you'll generally need to access the disk only when you want to make a final backup.

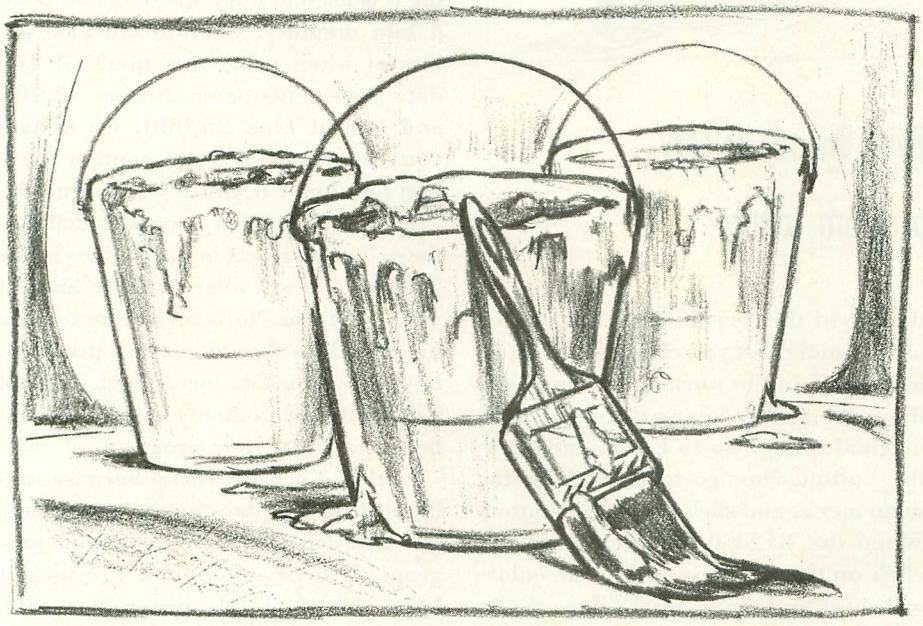
American TV, at 1-800-551-9995, sells I/O Cable Plug Kits for $4.50 (connectors only; you build the cable). They may also sell complete SIO cables (which would save you some wiring time, but cost a little more). B & C Computervisions at 408-749-1003, sells the SIO cables also.

You can get an RS232 switch box from any computer dealer, but prices vary widely. For a two-position switch box the price may range from $30 to $70 or more. I was able to find one, new, at a local electronics specialty store for only $35. JACO Enterprises at 408-996-0675 advertises then in *Byte* for $45 each.

The connectors and housings (backshells) can be purchased at any Radio Shack. They may have the switch boxes too, but, I would expect, at a slightly higher price. The connectors (all but the 13-pin molex) and switch boxes would probably be available from electronics mail-order houses such as JDR Microdevices at 800-638-5000 or JAMECO Electronics at 415-592-8097.

Paint Shop

by Jerry M. Beardsley



Welcome to *Paint Shop*, the program that lets you custom-mix your own colors, then save the data to disk for use in your own programs.

Before we get into the workings of Paint Shop, I would like to thank *James Luczak* for his very useful article "VDI Sampler", that appeared in the June issue of *ANALOG Computing*. I found myself turning to it so often, the cover fell off my copy. If you don't have a copy of that issue, it is well worth the time and effort spent to dig one up. And now, on to Paint Shop!

Typing In Paint Shop

Type in the program exactly as it appears in Listing 1. When you have finished, save a copy to disk, then use ST-Check (Issue 41) to make sure there are no typos. If your typing checks, you are ready to run Paint Shop.

Using Paint Shop

Paint Shop will run only in the low resolution mode. The first screen to come up will be the title screen. After a few seconds, a chime will sound, and a prompt will appear. When you press a key, a tone will sound, the screen will clear, and the Paint Shop work screen will come up. Notice that the screen is divided into three areas. The section at the top contains the 16 basic colors. The main menu is located in the area at the lower left. To the right of the menu is the work area.

Paint Shop is an easy program to use. The mouse is used for all input, except when loading or saving data. Move the

Paint Shop

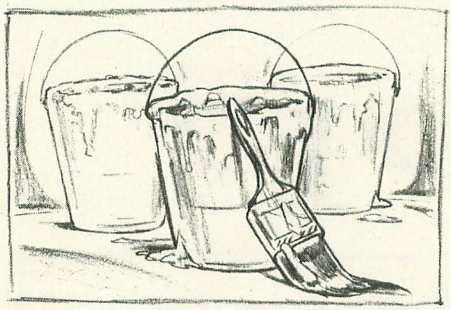
cursor to the MIX COLOR option on the main menu, and click the mouse button. MIX COLOR will lighten, the work area will clear, and the color mixing graphics will appear.

Move the cursor to NEW COLOR, and click the button. NEW COLOR will change to reverse video. Now go to the top of the screen, and click on one of the colors. The black box that appeared around the first color will move to this new color. This is the base color. You are now ready to begin your mixing.

The three sets of arrows in the work area are used for color mixing. They are labled R, G, and B for red, green and blue. The arrows at the top (with + at the tips) add to that color. For example, if you click on the top arrow of the R, you will add some red to the base color. The arrows at the bottom (with + at the tips) subtract from that color. Note that the colors wrap around. That is, if you are adding color and the value becomes greater than 1,000 (the maximum allowed), the value will change to zero. This is also true when subtracting color, but the value goes from less than zero to 1,000.

Let's save our new color data to disk. Go to EXIT, and click the mouse button. Go to SAVE DATA on the main menu, and click the button. The work area will clear, and the SAVE DATA routine will begin. Type in a filename of not more than eight characters in length. Do not use an extender! If you enter an invalid filename, the program will let you know, then prompt you for a new entry. Press RETURN. The drive will come on, and "WORKING" will appear on the screen. After a few moments, "SAVE COMPLETE" will appear, and the drive will turn off. You have just saved your new colors to disk in two files; one with a ".DAT" extender and one with a ".BAS" extender. The use of these two files will be explained later.

Now let's go back and reset all of the colors to their original values. Go to MIX COLOR, and click the mouse button. When the color mixing graphics come up, click on NEW COLOR. Go to the top of the screen, and click on one of the colors that you changed. Next, go to RESET in the work areas, and click the button. The color will return to its original shade. Go to EXIT, and click the button. Now go to RESET on the main menu, and click the mouse button. When the RESET graphics come up, click on the YES box. All of the colors will reset to their original values. Go to QUIT on the main menu, and click the button. Click on the YES box. All of the system values, including the colors, will be reset to their default settings. The screen will clear, and the command window will appear.



Paint Shop is an easy program to use.

Programming With Paint Shop

There are several ways to use the data generated by Paint Shop. The two methods given here are the easiest for BASIC programmers to use. Both use VDI attribute function 14. The difference lies in where the data is stored. This is where the ".DAT" and ".BAS" files mentioned earlier come into play. The ".DAT" file is made up of raw data that must be loaded and poked into memory. Listing 2 handles that job. The ".BAS" file is composed of BASIC program lines containing data statements that contain the data for your colors. This file may be added to your own programs by using the merge option on the File Menu at the top of the screen. Listing 3 is used to fetch the data and poke it into memory. Caution must be observed when using this method! The data statements begin at Line 10,000 and end at Line 25,000. Be certain your program does not contain these line numbers. If you try to merge the ".BAS" file with a program that has these line numbers in it, the lines in the ".BAS" file will overwrite the lines in your program. Not a nice thing to have happen! This should not be a problem, but forewarned is forearmed. If need be, the data statements may be renumbered to fit in your program.

Listing 3 is my favorite because, unlike listing 2, it does not have to access the disk drive when you run your program. Whichever method you pick, I hope you find this program useful. Good luck and happy programming!

Jerry M. Beardsley is a self-taught programmer with five years experience. He enjoys working in BASIC, Logo and C. He lives in Cuyahoga Falls, Ohio, with his wife Mary, his son Robbie and his mother. His other hobbies are reading Sci-Fi and Fantasy. This is his first published work.

Listing 1: ST BASIC

```
10 '****************** ST PAINT SHOP, VE
RSION 1.0 ******************
20 '************************ COPYRIGHT 1
986 ************************
30 '********************* BY JERRY M. BEA
RDSLEY **********************
40 '
50 '********** INITIALIZE VARIABLES, D
RAW TITLE SCREEN **********
60 FULLW 2:CLEARW 2:DIM DC(16,3):A=0:B
=1:C=2:D=3:E=4:F=5
70 G=6:H=7:I=8:J=9:K=10:L=22:M=15:BX1=
```

```
B:BY1=L:BX2=304:BY2=188
80 T$(A)="ANALOG COMPUTING":T$(B)="PRE
SENTS":F$(J)="COPYRIGHT"
90 T$(C)="PAINT SHOP":T$(D)=", VERSION
 1.0":T$(E)=" 1986 BY"
100 T$(F)="JERRY M. BEARDSLEY":T$(G)=C
HR$(189):T$(H)=CHR$(191)
110 T$(I)="One Moment Please":T$(J)="P
ress Any Key To Continue"
120 IFS=C:FSI=J:FLC=C:GOSUB SIFSTYLE:G
OSUB SFINDEX:GOSUB SFCOLOR
130 GOSUB BAR:IFS=B:FSI=B:FLC=B:BX1=BX
1+43:BY1=BY1+L
140 BX2=BX2-44:BY2=BY2-28:GOSUB SIFSTY
LE:GOSUB SFINDEX
150 GOSUB SFCOLOR:GOSUB BAR:COLOR B,I,
E,B,B:LW=D:GOSUB SPLWIDTH
160 LINEF BX1-D,BY1-L,BX2+B,BY1-L:LINE
F BX2-B,BY1-L,BX2-B,BY2-L
170 LINEF BX2+B,BY2-L,BX1-D,BY2-L:LINE
F BX1-B,BY2-L,BX1-B,BY1-L
180 COLOR M:WM=C:TE=41:P$=T$(A):SX=BX1
+E*K+C:SY=BY1+11
190 GOSUB SWMODE:GOSUB TXTEFFECTS:GOSU
B TXTPRINT:SX=BX1+H*K+E
200 SY=BY1+C*K+H:P$=T$(B):GOSUB TXTPRI
NT:COLOR A:TE=33
210 SX=BX1+K:SY=BY1+F*K:P$=T$(C):GOSUB
 TXTEFFECTS
220 GOSUB TXTPRINT:TE=A:COLOR D:SX=BX1
+J*K+C:P$=T$(H)
230 GOSUB TXTEFFECTS:GOSUB TXTPRINT:CO
LOR A:TE=33:SX=BX1+K*K
240 P$=T$(D):GOSUB TXTEFFECTS:GOSUB TX
TPRINT:SX=BX1+D*K
250 SY=BY1+H*K:COLOR C:P$=F$(J):GOSUB
TXTPRINT:COLOR D:TE=A
260 SX=BX1+K*K+E:SY=BY1+H*K-B:P$=T$(G)
:GOSUB TXTEFFECTS
270 GOSUB TXTPRINT:COLOR C:SX=BX1+11*K
+J:SY=SY+B:TE=33:P$=T$(E)
280 GOSUB TXTEFFECTS:GOSUB TXTPRINT:SX
=BX1+E*K-G:SY=BY1+J*K
290 P$=T$(F):GOSUB TXTPRINT:SX=BX1+E*K
:SY=BY1+11*K:COLOR M
300 P$=T$(I):TE=I:GOSUB TXTEFFECTS:GOS
UB TXTPRINT:CN$=","
310 M$(B)="MIX COLOR":M$(C)="LOAD DATA
":M$(D)="SAVE DATA"
320 M$(E)="RESET":M$(F)="QUIT":M$(G)="
Use Mouse To"
330 M$(H)="Make Selection":M$(I)="Ente
r Filename":M$(J)="Working"
340 F$(A)="Save":F$(B)="Load":F$(C)="
Complete"
350 F$(D)="(NO EXTENDER)":F$(E)="INCOR
RECT ENTRY"
360 F$(F)="Return To Exit":F$(G)=" YES
 ":F$(H)=" NO "
370 F$(I)="Are You Sure?":N=16:O=166:Q
=303:R=21:R1=1000
380 S=72:T=152:U=304:P8=112:P9=124:P10
=146:LN=10000
390 P1=20:P2=19:P3=14:P4=11:P5=13:P6=1
25:P7=158:DIM NC(N,D)
400 B$=" NEW COLOR ":C$="POINT AND CLI
CK":E$=" EXIT "
410 NT$="ANALOG COMPUTING"
420 M1$=SPACE$(B)+M$(E)+SPACE$(B):X1=J
:Y1=45:X2=30:Y2=61
430 FOR X=A TO M:POKE CONTRL,26:POKE C
ONTRL+C,A:POKE CONTRL+G,C
440 POKE INTIN,X:POKE INTIN+C,B:VDISYS
(A):DC(X,A)=PEEK(INTOUT+C)
450 DC(X,B)=PEEK(INTOUT+E):DC(X,C)=PEE
K(INTOUT+G)
460 NC(X,A)=DC(X,A):NC(X,B)=DC(X,B):NC
(X,C)=DC(X,C):NEXT
470 COLOR B:GOSUB TXTPRINT:COLOR M:SX=
BX1+I:NF$="FILE NOT FOUND"
480 P$=T$(J):GOSUB TXTPRINT:GOTOXY A,A
:PRINT CHR$(H):D$="DATA"
490 KEY=INP(C):SOUND B,M,E,G,B:SOUND B
,A,A,A,A:GOSUB HMCURSOR
500 '********************* MAIN PROGRA
M LOOP *********************
510 CLEARW C:GOSUB NEWTITLE:IFS=B:FSI=
B:FLC=H
520 RX1=C:RY1=L:RX2=Q:RY2=36:ID=J
530 GOSUB SIFSTYLE:GOSUB SFINDEX:GOSUB
 SFCOLOR:GOSUB RFRECT
540 IFS=A:FSI=A:FLC=B:RY1=21:RY2=37:GO
SUB SIFSTYLE:GOSUB SFINDEX
550 GOSUB SFCOLOR:GOSUB RFRECT:COLOR B
,A,B:SX=110:SY=35:TE=E:TH=K
560 P$=T$(C)+T$(H):GOSUB TXTEFFECTS:GO
SUB TXTHEIGHT
570 GOSUB TXTPRINT:LINEF A,N,U,N:LINEF
 Q,N,Q,O:LW=B
580 LINEF Q,O,A,O:LINEF A,O,A,N:BX1=X1
:BY1=Y1:BX2=X2:BY2=Y2:IFS=B
590 GOSUB SFINDEX:GOSUB SPLWIDTH:FSI=B
:GOSUB SIFSTYLE
600 FOR X=A TO M:IF X=I THEN BX1=J:BY1
=70:BX2=30:BY2=86
610 FLC=X:GOSUB SFCOLOR:GOSUB BAR:LINE
F BX1-C,BY1-L,BX2,BY1-L
620 LINEF BX2,BY1-L,BX2,BY2-R:LINEF BX
2,BY2-R,BX1-C,BY2-R
630 LINEF BX1-C,BY2-R,BX1-C,BY1-L:BX1=
BX1+38:BX2=BX1+20:NEXT
640 LW=D:GOSUB SPLWIDTH:LINEF A,S,U,S:
LINEF T,S,T,O:SX=40:SY=110
650 RX1=G:RY1=97:RX2=148:RY2=112:TE=A:
GOSUB TXTEFFECTS
660 FOR X=B TO F:IFS=B:FSI=B:FLC=H:GOS
UB SIFSTYLE:GOSUB SFINDEX
670 GOSUB SFCOLOR:GOSUB RFRECT:IFS=A:F
SI=A:FLC=B:GOSUB SIFSTYLE
680 GOSUB SFINDEX:GOSUB SFCOLOR:GOSUB
RFRECT:RY1=RY2+D:RY2=RY1+M
690 P$=M$(X):GOSUB TXTPRINT:SY=RY2-C:N
EXT:IFS=B:FSI=B:FLC=A
700 GOSUB SIFSTYLE:GOSUB SFINDEX:GOSUB
 SFCOLOR
710 COLOR B:TE=I:GOSUB TXTEFFECTS:GOSU
B PROMPT
720 PICK=A:GOSUB SMCURSOR:GOSUB SMBUTT
ON:IF BP=A THEN 720
730 IF MX<C OR MX>149 THEN 720
740 IF MY>=97 AND MY<=110 THEN PICK=B:
SY=110:GOTO 790
750 IF MY>=115 AND MY<=128 THEN PICK=C
:SY=128:GOTO 790
760 IF MY>=135 AND MY<=145 THEN PICK=D
:SY=146:GOTO 790
770 IF MY>=153 AND MY<=163 THEN PICK=E
:SY=164:GOTO 790
780 IF MY>=171 AND MY<=181 THEN PICK=F
:SY=182
790 IF PICK=A THEN 720 ELSE GOTOXY A,A
:PRINT CHR$(H)
800 COLOR A:TE=A:SX=40:P$=M$(PICK):GOS
UB TXTEFFECTS
810 GOSUB HMCURSOR:GOSUB TXTPRINT:BX1=
157:BY1=96:BX2=301:BY2=186
820 ON PICK GOTO MIXCOLOR,LOADDATA,SAV
EDATA,SETFINISH,SETFINISH
830 MIXCOLOR:'**********************
**************************
840 CI=A:GOSUB BAR:IFS=A:FSI=A:FLC=B:G
OSUB SIFSTYLE:GOSUB SFINDEX
850 GOSUB SFCOLOR:BX1=X1-C:BY1=Y1-B:BX
2=X2+C:BY2=Y2+C:GOSUB BAR
860 BX1=BX1-B:BY1=BY1-B:BX2=BX2+B:BY2=
BY2+B:GOSUB BAR:COLOR B
870 TH=C*K:GOSUB TXTHEIGHT:GOTOXY P2-B
,P4:PRINT CHR$(B)
880 GOTOXY P2-B,M:PRINT CHR$(C):GOTOXY
 P2+C,P4:PRINT CHR$(B)
890 GOTOXY P2+C,M:PRINT CHR$(C):GOTOXY
 P2+F,P4:PRINT CHR$(B)
```

Paint Shop

```
900 GOTOXY P2+F,M:PRINT CHR$(C):TH=J:G
OSUB TXTHEIGHT
910 SX=167:SY=138:P$="R":GOSUB TXTPRIN
T
920 SX=SX+R+G:P$="G":GOSUB TXTPRINT
930 SX=SX+27:P$="B":GOSUB TXTPRINT
940 SX=SX-54:SY=105:P$="+":GOSUB TXTPR
INT
950 SX=SX+R+G:GOSUB TXTPRINT:SX=SX+R+G
:GOSUB TXTPRINT
960 SX=SX-54:SY=170:P$="-":GOSUB TXTPR
INT
970 SX=SX+R+G:GOSUB TXTPRINT:SX=SX+R+G
:GOSUB TXTPRINT
980 SX=SX+K+I:SY=120:P$=M1$:GOSUB TXTP
RINT
990 BX1=SX-B:BY1=SY-H:BX2=295:BY2=SY+C
:GOSUB BAR
1000 BX1=SX-C:BY1=SY-I:BX2=296:BY2=SY+
D:GOSUB BAR
1010 SY=SY+37:P$=E$:GOSUB TXTPRINT
1020 BX1=SX-B:BY1=SY-H:BX2=287:BY2=SY+
C:GOSUB BAR
1030 BX1=SX-C:BY1=SY-I:BX2=288:BY2=SY+
D:GOSUB BAR
1040 SX=SX-53:SY=SY+23:P$=B$:GOSUB TXT
PRINT
1050 BX1=SX-B:BY1=SY-H:BX2=BX2-14:BY2=
SY+C:GOSUB BAR
1060 BX1=SX-C:BY1=SY-I:BX2=BX2+B:BY2=S
Y+D:GOSUB BAR
1070 BX1=X1-D:BY1=Y1-C:BX2=X2+D:BY2=Y2
+D
1080 PICK1=A:GOSUB SMCURSOR:GOSUB SMBU
TTON:IF BP=A THEN 1080
1090 IF MX<156 OR MX>299 OR MY<96 OR M
Y>185 THEN 1080
1100 IF MX>=165 AND MX<=175 AND MY>=P8
 AND MY<=P9 THEN PICK1=B
1110 IF MX>=165 AND MX<=175 AND MY>=P1
0 AND MY<=P7 THEN PICK1=B
1120 IF MX>=192 AND MX<=202 AND MY>=P8
 AND MY<=P9 THEN PICK1=C
1130 IF MX>=192 AND MX<=202 AND MY>=P1
0 AND MY<=P7 THEN PICK1=C
1140 IF MX>=219 AND MX<=230 AND MY>=P8
 AND MY<=P9 THEN PICK1=D
1150 IF MX>=219 AND MX<=230 AND MY>=P1
0 AND MY<=P7 THEN PICK1=D
1160 IF MX>=239 AND MX<=294 AND MY>=P8
 AND MY<=122 THEN PICK1=E
1170 IF MX>=239 AND MX<=285 AND MY>=14
9 AND MY<=P7 THEN PICK1=F
1180 IF MX>=188 AND MX<=274 AND MY>=17
2 AND MY<=181 THEN PICK1=G
1190 IF PICK1=A THEN 1080 ELSE GOTOXY
A,A:PRINT CHR$(H)
1200 ON PICK1 GOTO RED,GREEN,BLUE,PUTB
ACK,EXIT,NEWCOLOR
1210 RED:'**************************
**************************
1220 IF MY<=P9 THEN NC(CI,A)=NC(CI,A)+
P6
1230 IF MY>=P10 THEN NC(CI,A)=NC(CI,A)
-P6
1240 IT=A:GOSUB SETCOLOR:GOTO 1080
1250 GREEN:'**************************
**************************
1260 IF MY<=P9 THEN NC(CI,B)=NC(CI,B)+
P6
1270 IF MY>=P10 THEN NC(CI,B)=NC(CI,B)
-P6
1280 IT=B:GOSUB SETCOLOR:GOTO 1080
1290 BLUE:'**************************
**************************
1300 IF MY<=P9 THEN NC(CI,C)=NC(CI,C)+
P6
1310 IF MY>=P10 THEN NC(CI,C)=NC(CI,C)
-P6
1320 IT=C:GOSUB SETCOLOR:GOTO 1080
1330 PUTBACK:'**************************
**************************
1340 GOSUB HMCURSOR:P$=M1$:SX=239:SY=1
20:COLOR A:GOSUB TXTPRINT
1350 WM=E:GOSUB SWMODE:COLOR B:GOSUB T
XTPRINT
1360 POKE CONTRL,14:POKE CONTRL+C,A:PO
KE CONTRL+G,E
1370 POKE INTIN,CI:POKE INTIN+C,DC(CI,
A):POKE INTIN+E,DC(CI,B)
1380 POKE INTIN+G,DC(CI,C):VDISYS(A):N
C(CI,A)=DC(CI,A)
1390 NC(CI,B)=DC(CI,B):NC(CI,C)=DC(CI,
C):GOSUB PAUSE
1400 WM=B:GOSUB SWMODE:COLOR B:GOSUB T
XTPRINT
1410 GOSUB SMCURSOR:GOTO 1080
1420 EXIT:'**************************
**************************
1430 GOSUB HMCURSOR:P$=E$:SX=239:SY=15
7:COLOR A:GOSUB TXTPRINT
1440 WM=E:GOSUB SWMODE:COLOR B:GOSUB T
XTPRINT:FLC=A:GOSUB SFCOLOR
1450 WM=C:GOSUB SWMODE:GOSUB BAR:BX1=B
X1+B:BY1=BY1+B:BX2=BX2-B
1460 BY2=BY2-B:GOSUB BAR:GOSUB PAUSE:G
OSUB CLEANUP:GOTO 720
1470 NEWCOLOR:'**********************
**************************
1480 GOSUB HMCURSOR:P$=B$:SX=186:SY=18
0:COLOR A:GOSUB TXTPRINT
1490 WM=E:GOSUB SWMODE:COLOR B:GOSUB T
XTPRINT
1500 GOSUB SMCURSOR:WM=C:GOSUB SWMODE:
TX=-1
1510 GOSUB SMBUTTON:IF BP=A THEN 1510
1520 IF MX<J OR MX>294 OR MY<47 OR MY>
85 THEN 1510
1530 IF MY>=71 AND MY<=85 THEN AD=I EL
SE AD=A
1540 IF MX>=J AND MX<=28 THEN CI=AD:TX
=A:GOTO 1620
1550 IF MX>=47 AND MX<=66 THEN CI=AD+B
:TX=B:GOTO 1620
1560 IF MX>=85 AND MX<=104 THEN CI=AD+
C:TX=C:GOTO 1620
1570 IF MX>=123 AND MX<=142 THEN CI=AD
+D:TX=D:GOTO 1620
1580 IF MX>=161 AND MX<=180 THEN CI=AD
+E:TX=E:GOTO 1620
1590 IF MX>=199 AND MX<=218 THEN CI=AD
+F:TX=F:GOTO 1620
1600 IF MX>=237 AND MX<=257 THEN CI=AD
+G:TX=G:GOTO 1620
1610 IF MX>=275 AND MX<=294 THEN CI=AD
+H:TX=H:GOTO 1620
1620 IF TX<A THEN 1510 ELSE GOTOXY A,A
:PRINT CHR$(H)
1630 GOSUB HMCURSOR:FLC=A:GOSUB SFCOLO
R:GOSUB BAR
1640 BX1=BX1+B:BY1=BY1+B:BX2=BX2-B:BY2
=BY2-B:GOSUB BAR:FLC=B
1650 GOSUB SFCOLOR:IF CI>=I THEN BY1=7
0:BY2=86 ELSE BY1=Y1:BY2=Y2
1660 BX1=TX*38+H:BY1=BY1-B:BX2=BX1+24:
BY2=BY2+C:GOSUB BAR
1670 IF CI=A OR CI=I THEN BX1=X1-C:BX2
=X2+C:GOSUB BAR
1680 BX1=BX1-B:BY1=BY1-B:BX2=BX2+B:BY2
=BY2+B:GOSUB BAR
1690 WM=B:GOSUB SWMODE:GOSUB TXTPRINT:
GOSUB SMCURSOR:GOTO 1080
1700 LOADDATA:'**********************
**************************
1710 ON ERROR GOTO 1810
1720 ER=A:GOSUB BAR:TH=J:GOSUB TXTHEIG
HT:GOSUB GETNAME
1730 GOTOXY R,M:WRITE M$(J):OPEN "I",#
1,NAME$
1740 POKE CONTRL,14:POKE CONTRL+C,A:PO
KE CONTRL+G,E:CT=C
1750 FOR X=A TO M:FOR Y=A TO C:INPUT#1
```

```
,NC(X,Y)
1760 POKE INTIN,X:POKE INTIN+CT,NC(X,Y
):VDISYS(A)
1770 CT=CT+C:IF Y=C THEN CT=Y
1780 NEXT Y,X:CLOSE:COLOR B:P$=F$(B)+F
$(C)
1790 GOTOXY K+J,M:WRITE P$:GOSUB PAUSE
1800 GOSUB BAR:GOSUB CLEANUP:GOTO 720
1810 GOTOXY P2,M:PRINT NF$:GOSUB PAUSE
:RESUME 1700
1820 SAVEDATA:'**********************
**************************
1830 ER=A:GOSUB BAR:TH=J:GOSUB TXTHEIG
HT:GOSUB GETNAME
1840 GOTOXY R,M:WRITE M$(J):OPEN "O",#
1,NAME$
1850 FOR X=A TO M:FOR Y=A TO C:WRITE#1
,NC(X,Y):NEXT Y,X:CLOSE
1860 OPEN "O",#1,NAME1$:FOR X=A TO M
1870 PRINT #1,LN,D$;NC(X,A);CN$;NC(X,B
);CN$;NC(X,C)
1880 LN=LN+R1:NEXT:CLOSE
1890 COLOR A:GOTOXY R,M:WRITE M$(J):CO
LOR B:P$=F$(A)+F$(C)
1900 GOTOXY K+J,M:WRITE P$:GOSUB PAUSE
:GOSUB BAR
1910 GOSUB CLEANUP:GOTO 720
1920 SETFINISH:'**********************
**************************
1930 GOSUB BAR:IFS=A:FSI=A:FLC=B:GOSUB
 SIFSTYLE:GOSUB SFINDEX
1940 GOSUB SFCOLOR:GOSUB MAKESURE
1950 IF PICK=E AND CHOICE=G THEN GOSUB
 RESETIT
1960 IF PICK=F AND CHOICE=G THEN GOSUB
 ENDIT
1970 GOSUB CLEANUP:GOTO 720
1980 '********************** SUBROUTI
NES **********************
1990 MAKESURE:'**********************
**************************
2000 COLOR B:GOTOXY P2+B,K:PRINT F$(I)
:GOTOXY P2,P5:PRINT F$(G)
2010 BX1=172:BY1=133:BX2=212:BY2=150:G
OSUB BAR
2020 BX1=BX1-B:BY1=BY1-B:BX2=BX2+B:BY2
=BY2+B:GOSUB BAR
2030 GOTOXY 28,P5:PRINT F$(H):BX1=252:
BY1=133:BX2=285:BY2=150
2040 GOSUB BAR:BX1=251:BY1=132:BX2=286
:BY2=151:GOSUB BAR
2050 GOTOXY P2,N:PRINT C$:GOSUB SMCURS
OR
2060 GOSUB SMBUTTON:IF BP=A THEN 2060
2070 IF MY<134 OR MY>148 THEN 2060 ELS
E COLOR A
2080 IF MX>=174 AND MX<=208 THEN CHOIC
E=G:X=P2:Y=P5
2090 IF MX>=251 AND MX<=284 THEN CHOIC
E=H:X=28:Y=P5
2100 IF CHOICE=A THEN 2060 ELSE GOTOXY
 X,Y:PRINT F$(CHOICE)
2110 WM=E:GOSUB SWMODE:COLOR B:GOTOXY
X,Y:PRINT F$(CHOICE)
2120 PRINT CHR$(H):GOSUB PAUSE:GOSUB H
MCURSOR:RETURN
2130 GETNAME:'**********************
**************************
2140 COLOR B:GOTOXY P2,J:PRINT M$(I):G
OTOXY P2,P4:PRINT F$(D)
2150 GOTOXY P2,M:PRINT F$(F):GOTOXY P2
,P5:INPUT "",NAME$
2160 COLOR A:GOTOXY P2,M:PRINT F$(F):C
OLOR B
2170 FOR NM=B TO LEN(NAME$)
2180 IF MID$(NAME$,NM,B)="." THEN ER=B
:GOTO 2220
2190 NEXT:IF LEN(NAME$)=A THEN SOUND B
,M,E,G,B:SOUND B,A,A,A,A
2200 IF LEN(NAME$)=A THEN GOSUB BAR:GO
SUB CLEANUP:GOTO 720
2210 IF LEN(NAME$)>I OR VAL(NAME$)>A T
HEN ER=B
2220 IF ER=B THEN GOTOXY P2,M:PRINT F$
(E):GOSUB PAUSE
2230 IF ER=B AND PICK=C THEN LOADDATA
2240 IF ER=B AND PICK=D THEN SAVEDATA
2250 NAME1$=NAME$+".BAS":NAME$=NAME$+"
.DAT":RETURN
2260 CLEANUP:'**********************
**************************
2270 IF PICK=B THEN SX=40:SY=110
2280 BX1=157:BY1=96:BX2=301:BY2=186
2290 COLOR D:WM=C:IFS=B:FSI=B:FLC=A:TE
=A:TH=K:GOSUB SWMODE
2300 GOSUB SIFSTYLE:GOSUB SFINDEX:GOSU
B SFCOLOR:GOSUB BAR
2310 GOSUB TXTEFFECTS:GOSUB TXTHEIGHT:
P$=M$(PICK):GOSUB TXTPRINT
2320 COLOR B:GOSUB TXTEFFECTS:GOSUB TX
TPRINT
2330 TE=I:GOSUB TXTEFFECTS:GOSUB PROMP
T:RETURN
2340 PROMPT:'**********************
**************************
2350 GOTOXY P1,K:PRINT M$(G):GOTOXY P2
,P3:PRINT M$(H):RETURN
2360 PAUSE:'**********************
**************************
2370 FOR TIME=A TO 1000:NEXT:RETURN
2380 SMBUTTON:'**********************
**************************
2390 POKE CONTRL,124:POKE CONTRL+C,A:P
OKE CONTRL+G,A
2400 VDISYS(A):BP=PEEK(INTOUT):MX=PEEK
(PTSOUT)
2410 MY=PEEK(PTSOUT+C):RETURN
2420 SMCURSOR:'**********************
**************************
2430 POKE CONTRL,122:POKE CONTRL+C,A:P
OKE CONTRL+G,B
2440 POKE INTIN,A:VDISYS(A):RETURN
2450 HMCURSOR:'**********************
**************************
2460 POKE CONTRL,123:POKE CONTRL+C,A:P
OKE CONTRL+G,A
2470 VDISYS(A):RETURN
2480 BAR:'**********************
**************************
2490 POKE CONTRL,11:POKE CONTRL+C,C:PO
KE CONTRL+K,B
2500 POKE PTSIN,BX1:POKE PTSIN+C,BY1:P
OKE PTSIN+E,BX2
2510 POKE PTSIN+G,BY2:VDISYS(A):RETURN
2520 RFRECT:'**********************
**************************
2530 POKE CONTRL,11:POKE CONTRL+C,C:PO
KE CONTRL+G,A
2540 POKE CONTRL+K,ID:POKE PTSIN,RX1:P
OKE PTSIN+C,RY1
2550 POKE PTSIN+E,RX2:POKE PTSIN+G,RY2
2560 VDISYS(A):RETURN
2570 SIFSTYLE:'**********************
**************************
2580 POKE CONTRL,23:POKE CONTRL+C,A:PO
KE CONTRL+G,B
2590 POKE INTIN,IFS:VDISYS(A):RETURN
2600 SFINDEX:'**********************
**************************
2610 POKE CONTRL,24:POKE CONTRL+C,A:PO
KE CONTRL+G,B
2620 POKE INTIN,FSI:VDISYS(A):RETURN
2630 SFCOLOR:'**********************
**************************
2640 POKE CONTRL,25:POKE CONTRL+C,A:PO
KE CONTRL+G,B
2650 POKE INTIN,FLC:VDISYS(A):RETURN
2660 SWMODE:'**********************
**************************
2670 POKE CONTRL,32:POKE CONTRL+C,A:PO
KE CONTRL+G,B
2680 POKE INTIN,WM:VDISYS(A):RETURN
```

Paint Shop

```
2690 TXTEFFECTS:'**********************
**************************
2700 POKE CONTRL,106:POKE CONTRL+C,B:P
OKE CONTRL+G,A
2710 POKE INTIN,TE:VDISYS(A):RETURN
2720 TXTPRINT:'***********************
**************************
2730 POKE CONTRL,I:POKE CONTRL+C,B:POK
E CONTRL+G,LEN(P$)
2740 FOR CH=A TO PEEK(CONTRL+G)
2750 POKE INTIN+(CH-B)*C,ASC(MID$(P$,C
H,B)):NEXT
2760 POKE PTSIN,SX:POKE PTSIN+C,SY:VDI
SYS(A):RETURN
2770 TXTHEIGHT:'**********************
**************************
2780 POKE CONTRL,107:POKE CONTRL+C,A:P
OKE CONTRL+G,B
2790 POKE INTIN,TH:VDISYS(A):RETURN
2800 SPLWIDTH:'***********************
**************************
2810 POKE CONTRL,16:POKE CONTRL+C,B:PO
KE CONTRL+G,A
2820 POKE PTSIN,LW:VDISYS(A):RETURN
2830 SETCOLOR :'**********************
**************************
2840 IF NC(CI,IT)>1000 THEN NC(CI,IT)=
A
2850 IF NC(CI,IT)<A THEN NC(CI,IT)=100
0
2860 POKE CONTRL,14:POKE CONTRL+C,A:PO
KE CONTRL+G,E
2870 POKE INTIN,CI:POKE INTIN+C,NC(CI,
A):POKE INTIN+E,NC(CI,B)
2880 POKE INTIN+G,NC(CI,C):VDISYS(A):R
ETURN
2890 RESETIT:'************************
**************************
2900 POKE CONTRL,14:POKE CONTRL+C,B:PO
KE CONTRL+G,E:CT=C
2910 FOR X=A TO M:FOR Y=A TO C:POKE IN
TIN,X
2920 POKE INTIN+CT,DC(X,Y):VDISYS(A):N
C(X,Y)=DC(X,Y)
2930 CT=CT+C:IF Y=C THEN CT=Y
2940 NEXT Y,X:RETURN
2950 NEWTITLE:'***********************
**************************
2960 WW=GB:GINTIN=PEEK(WW+I):POKE GINT
IN+A,PEEK(SYSTAB+I)
2970 POKE GINTIN+C,C:WWW=GINTIN+E:NT$=
NT$+CHR$(A)
2980 POKE WWW,VARPTR(NT$):GEMSYS(105):
RETURN
2990 ENDIT:'***********************
**************************
3000 TE=A:TH=J:WM=B:GOSUB TXTEFFECTS:G
OSUB TXTHEIGHT
3010 GOSUB SWMODE:GOSUB RESETIT:CLEARW
 C:NT$="OUTPUT"
3020 GOSUB NEWTITLE:GOSUB SMCURSOR:END
```

Listing 1: Checksums

```
 10 data  837, 214, 484, 358, 697, 925
, 352, 809, 53, 292, 5021
 110 data  918, 12, 706, 997, 602, 340
, 343, 699, 494, 806, 5917
 210 data  278, 984, 349, 753, 702, 7,
 389, 320, 805, 591, 5178
 310 data  933, 798, 706, 136, 132, 71
7, 43, 850, 86, 62, 4463
 410 data  391, 636, 182, 443, 234, 85
2, 679, 122, 905, 460, 4904
 510 data  35, 205, 492, 672, 767, 976
, 833, 400, 729, 745, 5854
 610 data  894, 18, 215, 475, 6, 453,
833, 910, 221, 599, 4624
 710 data  374, 576, 727, 508, 727, 73
6, 742, 574, 454, 562, 5980
 810 data  491, 663, 125, 64, 646, 823
, 384, 704, 708, 887, 5495
 910 data  68, 799, 402, 410, 592, 417
, 598, 578, 167, 273, 4304
 1010 data  314, 268, 275, 470, 683, 7
40, 841, 980, 262, 444, 5277
 1110 data  389, 433, 376, 430, 371, 4
34, 357, 506, 698, 258, 4252
 1210 data  711, 511, 971, 416, 867, 5
17, 979, 421, 808, 516, 6717
 1310 data  980, 419, 985, 258, 166, 5
56, 907, 736, 973, 161, 6141
 1410 data  714, 813, 358, 81, 104, 90
4, 129, 362, 173, 784, 4422
 1510 data  615, 701, 322, 281, 434, 4
73, 634, 658, 668, 658, 5444
 1610 data  681, 371, 575, 625, 229, 2
36, 565, 848, 470, 0, 4600
 1710 data  722, 316, 698, 290, 717, 3
64, 637, 242, 178, 808, 4972
 1810 data  576, 48, 320, 720, 874, 7,
 797, 77, 10, 504, 3933
 1910 data  436, 183, 424, 458, 30, 78
7, 442, 393, 115, 52, 3320
 2010 data  481, 826, 472, 917, 296, 5
99, 698, 226, 225, 389, 5129
 2110 data  715, 176, 976, 545, 170, 5
86, 217, 512, 590, 358, 4845
 2210 data  735, 981, 457, 504, 148, 9
95, 864, 40, 779, 829, 6332
 2310 data  653, 251, 524, 13, 864, 89
2, 556, 131, 766, 303, 4953
 2410 data  902, 132, 759, 572, 124, 7
63, 754, 744, 569, 94, 5413
 2510 data  163, 970, 557, 235, 128, 7
56, 132, 562, 834, 970, 5307
 2610 data  560, 814, 996, 565, 798, 9
64, 563, 782, 245, 772, 7059
 2710 data  761, 183, 285, 774, 708, 1
3, 214, 784, 772, 99, 4593
 2810 data  571, 784, 70, 359, 374, 57
3, 954, 699, 43, 297, 4724
 2910 data  560, 652, 641, 759, 144, 3
35, 150, 857, 897, 181, 5176
 3010 data  191, 199, 390
```

Listing 2: ST BASIC

```
10 OPEN "I",#1,"FILENAME":X=0:CT=2
20 POKE CONTRL,14:POKE CONTRL+2,0:POKE
 CONTRL+6,4
30 WHILE X<>16:INPUT #1,CD:POKE INTIN,
X
40 POKE INTIN+CT,CD:VDISYS(0):CT=CT+2
50 IF CT=8 THEN CT=2:X=X+1
60 WEND:CLOSE
```

Listing 2: Checksums

```
10 data  999, 396, 75, 813, 197, 730,
3210
```

Listing 3: ST BASIC

```
10 POKE CONTRL,14:POKE CONTRL+2,0:POKE
 CONTRL+6,4:CT=2
20 FOR X=0 TO 15:FOR Y=0 TO 2:READ CD
30 POKE INTIN,X:POKE INTIN+CT,CD:VDISY
S(0)
40 CT=CT+2:IF Y=2 THEN CT=Y
50 NEXT Y,X
10000 DATA COLOR DATA,COLOR DATA,COLOR
 DATA
11000 DATA COLOR DATA,COLOR DATA,COLOR
 DATA
```

Listing 3: Checksums

```
10  data 36, 436, 681, 344, 438, 517,
520, 2972
```

Review

▽ ▽ ▽

GFL CHAMPIONSHIP FOOTBALL
by Mark Madlund, Scott Orr and Dennis Kirsch
Gamestar, Inc.
Distributed by Activision, Inc.
2350 Bayshore Parkway
Mountain View, CA 94039
(415) 960-0410
$44.95
Medium or High Resolution
520ST, 1040 ST (Joystick Required)

by Scott Wasser

The first time you boot up Gamestar's GFL Championship Football, it has the same kind of impact as a blind-side tackle. But after becoming more familiar with this football simulation, it sometimes seems as though the program's creators didn't quite connect on what could have been a very impressive touchdown pass. Had they done just a wee bit better, this program could have won the Super Bowl of computer sports simulations.

The concept behind GFL Championship Football is what separates it from the rest of the pack of football simulations. The program's creators deserve a lot of credit for going beyond the scope of other football games, which basically allow you to pick up teams and starting lineups and call the plays.

You do all that in this game too. But after you pick the plays—on offense, anyway—it's up to you to *run* them as well. If the offensive play is a pass, you become the intended receiver; if it's a running play, you carry the ball; and if it's a kick, you're the one who has to boot it.

Granted, a few other football simulations—even some created for cartridge-based home game systems—give the user some degree of control over on-screen activities. Typically, they present a view of most of the playing field as it would appear from above or from the sidelines and allow a player to manipulate an on-screen character or characters.

GFL Championship Football, however, is unique in that it presents a perspective that puts you right in the middle of the action. If your team has the ball, you'll be placed right in the shoes of the ball carrier, intended receiver, punter or kick returner. This perspective is effective at conveying the feel of playing real football. It's up to you to run precise pass routes, hit the right holes on running plays, sidestep tacklers on kick returns and unload a punt before it's blocked.

On defense, your role is much more limited. Whether playing against the computer or a human foe, you're limited to simply choosing alignments you hope will stop the opposing team's offensive progress. Once that's done, you don't even have to use the joystick until the next play. Although the field perspective remains the same, you just sit and watch while the computer manipulates the defensive players according to the strategy you selected.

By not allowing the joystick jockey to intercept passes and make tackles, GFL Championship Football loses some of its luster. Since one of the game's primary objectives is to put the computer gamer right in the middle of the action, and since this is what really makes GFL Championship Football stand out from the crowd, it seems strange that you should be forced to sit back and watch whenever your team goes on the defensive.

Another aspect of GFL Championship Football that I found awkward was its method of play selection. Whether you're on the offensive or defensive, you choose your plays with joystick and fire button after scrolling through an on-screen list of 34 possible offensive plays and 12 defensive alignments. In two-player games, either the defensive coach or offensive coach can make the first selection, but the offensive team will be penalized if a play is not called within 30 seconds.

The problem here is that whenever a play or alignment is selected, it is visible to the opposing coach. In real life, this would be like inviting the opposing coach into your huddle. Since the element of surprise—particularly on offense—is so important in the real game of football, this aspect of GFL Championship Football is very unrealistic. The only concession to the importance of tricky play-calling is the program's 'audibilizing' option, whereby either coach can call one other play after his opponent's first selection.

This can marginally help catch an opponent off-guard, but it still doesn't allow you to stun the defense with a bomb at a time when your foe should be expecting a short run over the middle. Perhaps the game's designers felt that because the program doesn't allow joystick control over the defensive players, it would put the defensive team at too big a disadvantage if the offense were allowed such leeway. In any event, the design effectively eliminates an important part of real football.

Some other elements of real football are also missing from GFL Championship Football. Their absence doesn't dramatically detract from game play, but could bother football afficionados who may be considering purchasing the program. Passes cannot be thrown to any player other than a wide receiver, and only the tailback can carry the ball on running plays. There are also no provisions for making laterals, fake punts, fake field goals or running other trick plays such as quarterback sneaks or ends-around.

My final criticism concerns the game's graphics. I feel somewhat guilty for registering this complaint, since the program's designers deserve plenty of credit for a graphic presentation that makes the player feel as though he or she is actually on the field running the plays. Nevertheless, the on-screen images are a little too cartoon-like for my taste. This was particularly noticeable on running plays, during which members of the offensive line seem to simply bob up and down, rather than block.

These criticisms notwithstanding, the copy-protected GFL Championship Football (the program is warranted for 90 days and thereafter will be replaced for $7.50) is an outstanding football simulation. It meets the criteria of a good sports simulation, which is to say it accurately captures and conveys the look, feel and strategy of the sport on which it is based.

▽ ▽ ▽

BATTLEZONE
Atari Corp.
1196 Borregas Avenue
Sunnyvale, CA 94086
Low or high resolution
$29.95

by Maurice Molyneaux

Back in the days before sword-swinging Ninjas and automotive games overran the arcades of America, there were games like *Asteroids* and *Galaxian*. They were simple games that made no attempt to leave the world of two dimensions. They lacked *depth*. In fact, most games of the time were like this. Then along came Atari with a 3-D tank simulator called *Battlezone*, and for the first time an arcade game really caught my eye. The game, with its dark screen and none-too-bright green vector graphics—viewed through a targeting scope that shut out the rest of the world—had a surreal mood about it. The animation was smooth, and the sense of depth was incredible.

The game was a 3-D variation of the old *Tank* game that appeared in the late '70s. In *Tank* you took control of the title vehicle and tried to hunt down and destroy an enemy tank (usually controlled by a second player). The view was strictly 2-D, from overhead, with *very* simple graphics. *Battlezone* was revolutionary because it gave the player the perspective from *inside* the tank (this clearly limited the game to one player at a time), moving over an obstacle-strewn landscape on a search-and-destroy mission. Your view was limited to an arc forwards, but a radar scope provided vital information on the position of out-of-sight enemies and their shots. The primary menace came in the form of enemy tanks and "supertanks," along with annoying cruise-type missiles that periodically charged your tank. Saucers, harmless but worth many points, provided a tempting distraction. The idea was simple: Blast the enemy without getting blasted!

The ST version is pretty true to the arcade original in many respects. In fact, a number of strategies used by players of the arcade machine do, in fact, work on the ST version, with slight modifications.

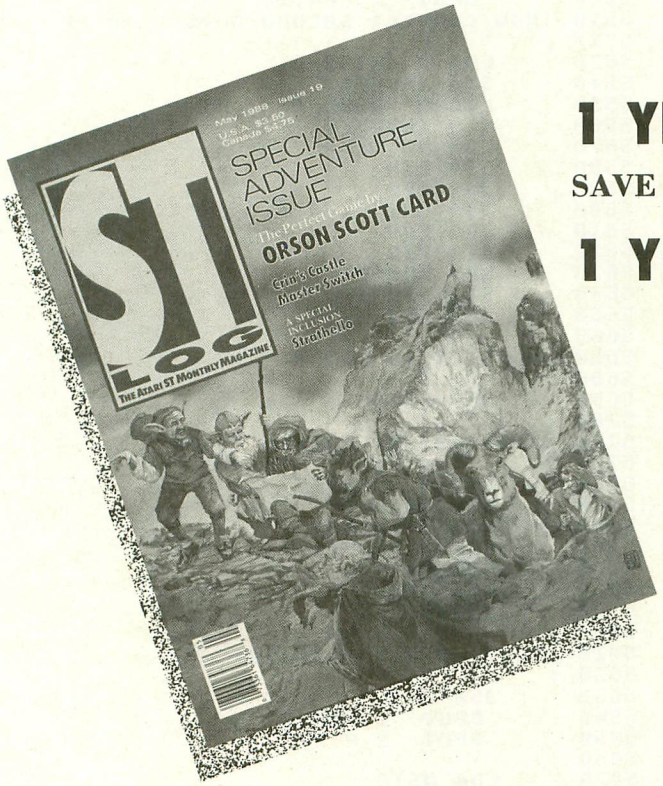
The graphics are not *exactly* like the arcade original's, but they're close enough. All the enemy vehicles are rendered in wireframe vector-type graphics, but the distant mountains and screen borders are "solid." The upper part of the screen, which contains the radar scope and scoring boxes, is rendered in relief and painted in camouflage! Further, on this *Battlezone* you can see the front of your tank's "treads" at the bottom of the screen, turning appropriately depending on how you are moving. The animation isn't as smooth as in a game like *Starglider*, but it's not bad. There are six skill levels to choose from, and options of one- or two-player games. When you quit to the Desktop, the high score is saved to disk.

The biggest problem with *Battlezone* for the ST is the controls. The arcade game featured twin levers, allowing the player to control each tank tread, putting it into forward or reverse, permitting complex maneuvers. The single joystick control on the ST is clumsy by comparison. You can play the game with the keyboard, using the cursor keys for movement and the space bar for "fire," but that control is no better, and in some ways worse, because the program can apparently ready only two keys at the same time. So if you're holding down the up and left arrow keys to go forwards and turn left, pressing "fire" will avail you nothing! You'll have to release one of the other keys first. Don't get me wrong, the game is perfectly playable with a joystick, but the control just isn't the same with one stick. The game allows two players, either using two sticks or sharing one. Unfortunately no mode was included where a single player could plug two sticks into his ST and emulate the arcade controls.

This game was developed for Atari by the Caesar Studio in Budapest, Hungary, and project management was by Andromeda Software (whose name appears on the game's title screen). The graphics for the game were designed with *Art Director* (also by Caesar Studio).

Battlezone is nowhere as complex as *Starglider* or *Arcticfox*, but that doesn't limit its appeal. Personally, I like blast-em games that involve some degree of strategy, but don't stray into the overly complex. *Battlezone* is simple, and challenging enough to merit attention, particularly at the low price at which Atari offers it. A final note: The game is not copy-protected, and the instructions urge you to make a backup and not use the master disk. This seems to be Atari's usual policy now, and I applaud it heartily. A

BOOT UP TO BIG SAVINGS!



1 YEAR FOR ONLY $28

SAVE $14 OFF THE COVER PRICE

1 YEAR *WITH DISK* ONLY $105

SAVE TIME AND MONEY SUBSCRIBE TO ST-LOG

SAVE $14 OFF THE COVER PRICE WITH THE CONVENIENCE OF HAVING ST-LOG DELIVERED DIRECTLY TO YOUR DOOR *BEFORE* IT EVEN HITS THE NEWSSTANDS! GET THE MOST OUT OF YOUR COMPUTER

SUBSCRIBE TO ST-LOG TODAY!

☐ 1 YEAR @ $28 — SAVE $14! MCFWW
FOREIGN — ADD $7 PER YEAR

☐ 1 YEAR WITH DISK @ $105 DCFWW
FOREIGN — ADD $15 PER YEAR

☐ PAYMENT ENCLOSED ☐ BILL ME

CHARGE MY: ☐ VISA ☐ MC #_____

EXPIRATION DATE SIGNATURE

MONEY BACK ON ALL UNUSED PORTIONS OF SUBSCRIPTIONS IF NOT SATISFIED.

NAME

ADDRESS

CITY STATE ZIP

MAKE CHECK PAYABLE TO L.F.P., INC., P.O. Box 16928, N. Hollywood, CA 91615. Offer expires August 31, 1988. Your first issue will arrive in 6 to 8 weeks.

WATCH FOR IT!

COM-DOS

(Continued from page 45)

```
4690       OPEN  1,8,0,LBUFF+4
4700      LDX #$10
4710      JSR PUTBYTES
4720      CPY #1
4730      BEQ DUPOK3
4740      JMP ERROR
4750 DUPOK3
4760       CLOSE  1
4770       PRINT  0,"Duplication comple
te."
4780      JMP MAINLOOP
4790 WDOS
4800      LDA LBUFF+4
4810      STA DOSSYS+1
4820       OPEN  1,8,0,DOSSYS
4830      CPY #1
4840      BNE WDOSERR
4850       CLOSE  1
4860       PRINT  0,"DOS.SYS file writt
en."
4870      JMP MAINLOOP
4880 WDOSERR
4890      JMP ERROR
4900 WMEM
4910      LDA LBUFF+4
4920      CMP #'8
4930      BNE WMEMSAV
4940       PRINT  0,"Can't write MEM.SA
V to drive 8!"
4950      JMP MAINLOOP
4960 WMEMSAV
4970      STA MEMSAV+1
4980       OPEN  1,8,0,MEMSAV
4990      CPY #1
5000      BNE WMEMERR
5010       BPUT  1,ENDCOMDOS,5625
5020      CPY #1
5030      BNE WMEMERR
5040       CLOSE  1
5050       PRINT  0,"MEM.SAV file writt
en."
5060      JMP MAINLOOP
5070 WMEMERR
5080      JMP ERROR
5090 BSAVE
5100      LDY #6
5110      LDX #0
5120 BSVLOOP
5130      LDA LBUFF,Y
5140      CMP #',
5150      BEQ BENDFL
5160      STA CBUFF,X
5170      INY
5180      INX
5190      JMP BSVLOOP
5200 BENDFL
5210      LDA #EOL
5220      STA CBUFF,X
5230      LDX #0       ;Copy 1st #
5240      INY
5250 CN1LOOP
5260      LDA LBUFF,Y
5270      CMP #',
5280      BEQ CN1DONE
5290      STA N1BUFF,X
5300      INY
5310      INX
5320      JMP CN1LOOP
5330 CN1DONE
5340      LDA #EOL
5350      STA N1BUFF,X
5360      LDX #0
5370      INY
5380 CN2LOOP
5390      LDA LBUFF,Y
5400      STA N2BUFF,X
5410      CMP #EOL
5420      BEQ CN2DONE
5430      INY
5440      INX
5450      JMP CN2LOOP
5455 ; Filename now in LBUFF, n1 in
5456 ; N1BUFF, n2 in N2BUFF.
5460 CN2DONE
5470 ;Open IOCB 1 for output
5480       OPEN  1,8,0,CBUFF
5490 ;
5500       BPUT  1,NNN,2
5510 ;Now convert second number to FP
5520 ;
5530      LDA # <N2BUFF
5540      STA INBUFF
5550      LDA # >N2BUFF
5560      STA INBUFF+1
5570      LDA #0
5580      STA CIX
5590      JSR AFP
5600      JSR FPI
5610      LDA FR0
5620      STA PTR
5630      LDA FR0+1
5640      STA PTR+1
5650      JSR IFP
5660 ;
5670 ;Move it to FR1
5680 ;
5690      JSR FMOVE
5700 ;
5710 ;Convert 1st number to FP
5720 ;
5730      LDA # <N1BUFF
5740      STA INBUFF
5750      LDA # >N1BUFF
5760      STA INBUFF+1
5770      LDA #0
5780      STA CIX
5790      JSR AFP
5800 ;
5810 ;Convert # to Integer
5820 ;
5830      JSR FPI
5840       BPUT  1,FR0,2
5850       BPUT  1,PTR,2
5860 ;
5870      LDX #$10
5880      LDA FR0
5890      STA ICBADR,X
5900      LDA FR0+1
5910      STA ICBADR+1,X
5920 ;
5930 ;Convert 1st # back to FP
5940 ;
5950      JSR IFP
5960      JSR FMOVE    ;Put it in FR1
5970 ;
5980 ;Convert 2nd # back to FP in FR0
5990 ;
6000      LDA # <N2BUFF
6010      STA INBUFF
6020      LDA # >N2BUFF
6030      STA INBUFF+1
6040      LDA #0
6050      STA CIX
6060      JSR AFP
6070 ;
6080 ;Do subtraction
6090 ;
6100      JSR FSUB
6110      JSR FPI      ;Convert to int
6120 ;
6130      LDX #$10
6140      INC FR0
6150      LDA FR0
6160      BEQ INCHI
6170      BNE STORIT
6180 INCHI
```

COM-DOS

```
6190      INC FR0+1
6200 STORIT
6210      STA ICBLEN,X
6220      LDA FR0+1
6230      STA ICBLEN+1,X
6240      LDA #CPBINR
6250      STA ICCOM,X
6260      JSR CIO      ;Put bytes
6270       PRINT  0,"Enter RUN address
(press RETURN if    none)"
6280       INPUT  0,LBUFF
6290      LDA LBUFF
6300      CMP #EOL
6310      BEQ GOMNLP
6320      LDA # <LBUFF
6330      STA INBUFF
6340      LDA # >LBUFF
6350      STA INBUFF+1
6360      LDA #0
6370      STA CIX
6380      JSR AFP
6390      JSR FPI
6400      LDA FR0
6410      STA RUNDAT
6420      LDA FR0+1
6430      STA RUNDAT+1
6440       BPUT  1,INIDAT,6
6450 GOMNLP
6460       CLOSE  1
6470       PRINT  0,"File saved."
6480      JMP MAINLOOP
6490 BOOT
6500       OPEN  1,4,0,"D:AUTORUN.SYS"
6510      CPY #170
6520      BEQ DOBOOT
6530       CLOSE  1
6540       PRINT  0,"□AUTORUN.SYS file
already exists!"
6550      JMP MAINLOOP
6560 DOBOOT
6570       CLOSE  1
6580       OPEN  1,8,0,"D:AUTORUN.SYS"
6590      LDY #0
6600 BOOTLOOP
6610      LDA LBUFF+5,Y
6620      STA FILEDAT,Y
6630      CMP #EOL
6640      BEQ DONEBOOT
6650      INY
6660      JMP BOOTLOOP
6670 DONEBOOT
6680       BPUT  1,PROG,73
6690       CLOSE  1
6700      JMP MAINLOOP
6710 HELP
6720       BPUT  0,HELPLST,ENDLST-HELPL
ST
6730      JMP MAINLOOP
6740 HELPLST
6750      .BYTE "K"
6760      .BYTE " LOCK D:filename.ext"
6761      .BYTE "      DIRn",EOL
6770      .BYTE " UNLOCK D:filename.ex"
6771      .BYTE "t    FORMATn",EOL
6780      .BYTE " DELETE D:filename.ex"
6781      .BYTE "t    WDOSn",EOL
6790      .BYTE " RENAME D:old,new"
6791      .BYTE "       WMEMn",EOL
6800      .BYTE " BSAVE D:file,begin,e"
6801      .BYTE "nd   REBOOT",EOL
6810      .BYTE " BLOAD D:filename.ext"
6811      .BYTE "     GO addr",EOL
6820      .BYTE " COPY D:file1,D2:file"
6821      .BYTE "2[/] BASIC",EOL
6830      .BYTE " DUP D:filename.ext"
6831      .BYTE "       STATUS",EOL
6840      .BYTE " BOOT filename.ext"
6841      .BYTE "       CLICK",EOL
6850      .BYTE " ?hxnm                "
6851      .BYTE "     HELP",EOL
6860      .BYTE EOL,"n=drive number (1"
6861      .BYTE EOL,"-8)",EOL
6870      .BYTE "hxnm = 4-digit hex nu"
6871      .BYTE "mber",EOL
6880      .BYTE "All input MUST be in "
6881      .BYTE "decimal.",EOL
6890 ENDLST
6900 GO
6910      LDA # <LBUFF+3
6920      STA INBUFF
6930      LDA # >LBUFF+3
6940      STA INBUFF+1
6950      LDA #0
6960      STA CIX
6970      JSR AFP
6980      JSR FPI
6990      JMP (FR0)
7000 STATUS
7010       BPUT  0,WVERMSG,14
7020      LDA WVERIFY
7030      CMP #80
7040      BEQ VEROFF
7050       PRINT  0,"ON"
7060      JMP DRIVES?
7070 VEROFF
7080       PRINT  0,"OFF"
7090 DRIVES?
7100       BPUT  0,DRIVEMSG,15
7110      LDA DRVBYT
7120      AND #$01
7130      BEQ NXTBIT
7140      LDA #'1
7150      JSR PRBIT
7160 NXTBIT
7170      LDA DRVBYT
7180      AND #$02
7190      BEQ NXTBIT2
7200      LDA #'2
7210      JSR PRBIT
7220 NXTBIT2
7230      LDA DRVBYT
7240      AND #$04
7250      BEQ NXTBIT3
7260      LDA #'3
7270      JSR PRBIT
7280 NXTBIT3
7290      LDA DRVBYT
7300      AND #$08
7310      BEQ NXTBIT4
7320      LDA #'4
7330      JSR PRBIT
7340 NXTBIT4
7350      LDA DRVBYT
7360      AND #$80
7370      BEQ ENDDRIVE
7380      LDA #'8
7390      JSR PRBIT
7400 ENDDRIVE
7410       PRINT  0
7420       BPUT  0,MAXFILES,54
7430      LDA SABYTE
7440      STA FR0
7450      LDA #0
7460      STA FR0+1
7470      JSR IFP
7480      JSR FASC
7490      LDY #$FF
7500 MAXLOOP
7510      INY
7520      LDA LBUFF,Y
7530      BPL MAXLOOP
7540      AND #$7F
7550      STA LBUFF,Y
7560      LDA #EOL
7570      STA LBUFF+1,Y
7580       PRINT  0,LBUFF
7590      JMP CHANGEM
7600 PRBIT
7610      STA BUFF2
7620       BPUT  0,BUFF2,2
7630      RTS
7640 CHANGEM
```

```
7650       PRINT  0
7660       PRINT  0,"Toggle Write Verif
y on/off (Y/N)?"
7670       INPUT  0,CBUFF
7680      LDA CBUFF
7690      CMP #'Y
7700      BEQ CHWV
7710      CMP #EOL
7720      BNE CHCONT1
7730      JMP MAINLOOP
7740 CHCONT1
7750      JMP CHDRV5?
7760 CHWV
7770      LDA WVERIFY
7780      EOR #$07
7790      STA WVERIFY
7800 CHDRV5?
7810       PRINT  0,"Change active driv
es (Y/N)?"
7820       INPUT  0,CBUFF
7830      LDA CBUFF
7840      CMP #'Y
7850      BEQ CHNGDR
7860      CMP #EOL
7870      BNE CHCONT2
7880      JMP MAINLOOP
7890 CHCONT2
7900      JMP CHFL5?
7910 CHNGDR
7920       PRINT  0,"Type the drive num
bers, one at a time,pressing RETURN af
ter each one."
7930       PRINT  0,"Enter 0 when you'r
e finished."
7940      LDA #0
7950      STA DRNUM
7960 DRLOOP
7970       INPUT  0,CBUFF
7980      LDA CBUFF
7990      CMP #'0
8000      BEQ DONEDR
8010      CMP #'1
8020      BEQ DR1
8030      CMP #'2
8040      BEQ DR2
8050      CMP #'3
8060      BEQ DR3
8070      CMP #'4
8080      BEQ DR4
8090      CMP #'8
8100      BEQ DR8
8110      JMP DONEDR
8120 DR1
8130      LDA #1
8140      CLC
8150      ADC DRNUM
8160      STA DRNUM
8170      JMP DRLOOP
8180 DR2
8190      LDA #2
8200      CLC
8210      ADC DRNUM
8220      STA DRNUM
8230      JMP DRLOOP
8240 DR3
8250      LDA #4
8260      CLC
8270      ADC DRNUM
8280      STA DRNUM
8290      JMP DRLOOP
8300 DR4
8310      LDA #8
8320      CLC
8330      ADC DRNUM
8340      STA DRNUM
8350      JMP DRLOOP
8360 DR8
8370      LDA #128
8380      CLC
8390      ADC DRNUM
8400      STA DRNUM
8410      JMP DRLOOP
8420 DONEDR
8430      LDA DRNUM
8440      STA DRVBYT
8450 CHFL5?
8460       PRINT  0,"Change number of f
iles that can be    open simultaneousl
y?"
8470       INPUT  0,CBUFF
8480      LDA CBUFF
8490      CMP #'Y
8500      BEQ CHFL5
8510      JMP MAINLOOP
8520 CHFL5
8530       PRINT  0,"How many files do
you want open at    one time (1-7)?"
8540       INPUT  0,CBUFF
8550      LDA # <CBUFF
8560      STA INBUFF
8570      LDA # >CBUFF
8580      STA INBUFF+1
8590      LDA #0
8600      STA CIX
8610      JSR AFP
8620      JSR FPI
8630      LDA FR0
8640      STA SABYTE
8650      JMP MAINLOOP
8660 CLICK
8670      LDA NOCLIK
8680      EOR #$FF
8690      STA NOCLIK
8700      JMP MAINLOOP
8710 HEXDEC
8720      LDA LBUFF+1
8730      JSR COMP
8740      STA HIBYTE
8750      LDA LBUFF+2
8760      JSR COMP
8770      STA LOBYTE
8780      JSR MULT
8790      STA FR0+1
8800      LDA LBUFF+3
8810      JSR COMP
8820      STA HIBYTE
8830      LDA LBUFF+4
8840      JSR COMP
8850      STA LOBYTE
8860      JSR MULT
8870      STA FR0
8880      JSR IFP
8890      JSR FASC
8900      LDY #$FF
8910 HDLOOP
8920      INY
8930      LDA LBUFF,Y
8940      BPL HDLOOP
8950      AND #$7F
8960      STA LBUFF,Y
8970      LDA #EOL
8980      STA LBUFF+1,Y
8990       PRINT  0,LBUFF
9000      JMP MAINLOOP
9010 COMP
9020      SEC
9030      SBC #48
9040      TAX
9050      LDA HEX,X
9060      RTS
9070 MULT
9080      LDA HIBYTE
9090      ASL A
9100      ASL A
9110      ASL A
9120      ASL A
9130      CLC
9140      ADC LOBYTE
9150      RTS
9160 ENDCOMDOS
9170      *=  $02E0    ;Run address
9180      .WORD BEGIN
```

ANALOG'S BEST!

Over 88 of ANALOG Computing's best and most popular programs available.

GAMES 1
Stuntman
Fill 'Er Up!
Adventure in the Fifth Dimension
Lumberjack
Space Assault
Darts
Harvey Wallbanger
Supereversion

GAMES 2
Retrofire
Roundup
Livewire
Bricklayer's Nightmare
Knights and Chalices
Air Attack
Avalanche

GAMES 3
Planetary Defense
Crash Dive!
Battle in the B-Ring
Bacterion!
Climber
Money Hungry
Buzz-zap!

GAMES 4
Race in Space
Bopotron!
Bopotron Construction Set
Fire Bug
Basic Burger
Cosmic Defender
Munch'In Clim'In
Shooting Stars

GAMES 5
Adventure at Vandenburg
Popcorn
Demon Birds (Action!)
R.O.T.O. (Action!)
Boulder Bombers
TwoGun
Dragonlord
Lunar Patrol

DISK UTILITIES
Disk Tool (1 and 2)
Burp!
Black Rabbit
Snail
Disk Cataloging Utility
Disk Directory Dump
BASIC Disk Utilities
AlterDOS
Disk Miser

GRAPHICS
Sketch Pad
Graphic Violence!
Multicolor Screen Generator
Moving Missiles in BASIC
Stars 3-D
Bar Chart Subroutine
Solid States
Scredit
Graph E's
P/M Creator/Animator

EDUCATION
Typing Trainer
Observational Astronomy
What Is It?
Spanish Study Guide
Math Attack
Word Scramble
Typing Evaluator
The Reading Program

UTILITIES 1
System Status
Buncrush
Unicheck
Creating an AUTORUN.SYS
NOREM
Hexpad
mUSE
Sound FX
Disassembler in BASIC
ConTEXT
Default

UTILITIES 2
Create-A-Font
Create-A-Font Datamaker
Fast Repeat Key
H:BUG
Auto Line Numbering
Binary File Menu Loader
Minicomp
Saturday Night Special
Touch-Tone™ Dialer
Stopwatch

To order, send $9.95 per disk
VISA (plus $2.50 per order shipping and handling) MasterCard
NO CODs ACCEPTED
to:

P.O. BOX 16927
N. HOLLYWOOD, CA 91615

THE #1 MAGAZINE FOR ATARI COMPUTER OWNERS
ANALOG COMPUTING

Review

▽ ▽ ▽

Colonial Conquest
SSI
1046 North Rengstorff Avenue
Mountain View, CA 94043
512K Disk $59.95

by Dan Cermak

While SSI has been a prolific producer of war simulations for the 8-bit, it has, for some time, neglected the ST. I don't mean to imply that it has produced nothing for this, the greatest gaming machine yet produced (as those of you still trapped in Phantasie can well attest). No, I mean that they have yet to produce an ST war simulation, putting this graphics and speed workhorse to task in the genre which has always been their forte. Fortunately, they have now begun to rectify this oversight.

Colonial Conquest is the first in what I hope will be a long line of war simulations for the ST. And while this game lacks the complexity, and hence the realism of some of SSI's greatest simulations, it more than makes up for it with great graphics and a simple user interface. Basically a complex version of the classic board game Risk, Colonial Conquest pits up to six players against one another and the computer in a bid to take control of the world.

The era is 35 years around the turn of the last century (1880-1914, to be precise). The world is dominated by six major powers: England, Germany, France, The United States, Japan, and Russia, each ready, willing and able to build and expand their own empire. The industrial revolution has created a need for raw materials to supply and people to consume the output churning from the thousands of new factories. Increasing one's empire in size and population is seen as an easy and simultaneous solution to each problem, as well as being a patriotic cause to rally the populace behind, helping them to forget, at least temporarily, their monotonous, factory-enslaved lives. And at this time in history, some parts of the world, especially Africa, were still unexplored and ripe for the picking. But while this historical backdrop is complex, the program itself is suprisingly simple.

Game play takes place in three phases; build, movement, and combat. This is a game of men and money, much like the games played by real world leaders. Individuals make little difference; it is the masses and the megabucks that rule. Each country has a given amount of each, and those with a lot of both have large naval fleets as well. Those with little are easy targets. In the building phase, you strengthen your units, lend money to allies, spy on enemies to learn their strengths, and try to subvert other countries' control over minor nations. Money ofttimes sways the allegiances of these tiny, impoverished nations. After deciding on a strategy (which, in multiple player games, could require treaties and trust between rival nations as well as the occasional backstab), you enter the movement phase. Here you move armies, 1,000 men at a time, to adjacent areas. You are allowed 20 such moves. Finally, the combat phase determines the outcomes of any battles arising from the movement of troops into enemy territories. At the end of each round, each player is scored, gaining points for battles won, losing them for battles lost. Points are also awarded for taking control of an area, the exact amount determined by reference to the value and strength of the conquered nation.

The screen display is superb. The monitor is filled with a world map containing the 125 major and minor countries. Each of the six major countries is represented by a color; the minor ones are either a neutral grey or flying the colors of one of the big six. While only a portion of the map is visible at one time, touching the arrow against an edge of the screen scrolls the display up and down, left and right. In fact, by scrolling left and right you can repeatedly circle the earth, scrolling so smoothly that some might become dizzy. The display resolution of the icons representing ports, supply centers, and fleets is so clear that one can't help but wonder what future simulations will be like, when control of thousands of units could literally fill the screen with these markers.

Like the screen images, documentation is likewise superb. The 26-page manual is organized simply and efficiently, with a page of short rules for the impatient, followed by a table of contents leading you to the various sections detailing the intricacies of the three phases, the implications of allowing a computer to control one or more major countries, and helpful hints and strategies. The last half of the booklet is devoted to an informative and entertaining essay on the politics and motivations of the era. Reading this section captures your imagination and fleshes out the blood battles waged on your screen. Tables and appendices contain the various values assigned to each country, and two maps (one plastic covered) show the world power structure in each of the historical scenarios. Even with all the instruction, I found it a little difficult to pick up on the game, probably due to the sparse ST-specific instructions, which failed to completely detail when to use the various windows and menus. Also, although you could play solo against the computer, the game yields the most enjoyment when three to six play, due to the fact that diplomacy plays a big part of the game.

But despite these blemishes, Colonial Conquest is an adequate war simulation. Although I'm dying for a complex simulation to make the crossover to this new medium, I am certain that with time more realistic, more strategic simulations will be offered. Until then, this one should satisfy all the war buffs. And if this is any indication of the future, then SSI is destined to conquer the ST.

Tutorial

Boot Camp

Quick, what values do you store at what addresses to enable player/missile graphics with single-line resolution? That's okay, I don't remember either. However, I learned long ago that the next best thing to knowing some useful tidbit of information is knowing where to find it when you need it.

by Karl Wiegers

Boot Camp

Bonzo (from "Attack of the Suicidal Road-Racing Aliens")

is fed up with being squashed.

Today he shoots back.

There are two ways to remember the useful tidbits required for player/missile graphics. The first is to keep a copy of *Mapping the Atari* by Ian Chadwick (COMPUTE! Books) handy. This book is absolutely indispensible for anyone programming an 8-bit Atari in assembly language (or any other language).

Even better, use your computer's memory instead of your own. Today I present a baker's dozen of macros that help you use player/missle graphics (PMG), display list interrupts (DLI), and vertical blank interrupts (VBI) in assembly programs. These macros simulate some of the commands Atari BASIC should have had but didn't. Many novice programmers are daunted by the minutiae associated with setting up PMG, but these macros are useful shortcuts to success. Along the way, we'll see how to manipulate missiles too. You see, Bonzo (from "Attack of the Suicidal Road-Racing Aliens") is fed up with being squashed. Today he shoots back.

Insecticide

We'd best begin with the "Whoops!" category. There's a small bug in the MOVE macro from two months ago. Please add this line to your MACRO.LIB file:

```
6115 LDY #0
```

Sorry about that.

Getting Started

Listing 1 contains the promised 13 new graphics macros. I decided to begin a new file of macros to be .INCLUDEd in future assembly programs, since the old MACRO.LIB file has become pretty long. Please enter Listing 1 into a file named GRAPHICS.LIB. If you write a program that doesn't use any of these macros, simply omit the .INCLUDE statement for this file.

If you're using the RAM disk file copier from last time, you should add D1:GRAPHICS.LIB to the list of files to be copied from the boot disk to the RAM disk. We can use the append fea-

ture of the DOS menu selection for copying files. Go to the DOS menu, choose item C to copy a file, and type:

```
E:,D:RAMDISK.FIL/A
```

This notation means that we want to copy from the screen editor (that is, the keyboard) to file D:RAMDISK.FIL, appending whatever we type on the keyboard to the present contents of D:RAMDISK.FIL. The cursor will then move to the beginning of the next line. Type:

```
D1:GRAPHICS.LIB
```

Press RETURN, and press control-3 to signify the end of the file. Your modified RAMDISK.FIL file should be written to the disk at this point. To verify that the change was made, copy from D:RAMDISK.FIL to E: and make sure all three lines appear:

```
D1:MACRO.LIB
D1:SUBS.LIB
D1:GRAPHICS.LIB
```

Graphics Shortcuts

Let's walk through the 13 macros in Listing 1. Most of the concepts will be familiar from our earlier graphics discussions, but I want to review a few points. These macros are all in MAC/65 format, but you should be able to adapt them to other macro assemblers with a little effort. The equates used by the macros are in Lines 170-280. You'll get a duplicate label error if any of these equates also appear elsewhere in your program.

The first entry is VBION in Lines 320-490. This routine simply turns on a vertical blank interrupt routine in your program. It requires one parameter, the address of the beginning of the VBI. I always label the beginning of my VBI routines as (guess what) "VBI"; so my calls to this macro are in the form :VBION VBI. It seems redundant, but it really isn't. All this routine does is insert your custom VBI routine into the deferred VBI vector so it gets executed every sixtieth of a second, as it should.

The obvious counterpart is the next macro, VBIOFF, which requires no parameters. It simply resets the deferred VBI vector to the system default, thereby disabling the user-written routine. For both VBION and VBIOFF, you can change the LDA #7 statement to LDA #6 if you wish to use an immediate, rather than deferred, VBI routine. See Boot Camp in issue 49 for a discussion of VBIs.

Similarly, the DLION macro (Lines 660-860) enables display list interrupts by setting bit 7 at address NMIEN ($D40E), Lines 780-800. DLION accepts one parameter, the address of your first DLI routine. I always call this (guess what) "DLI," so my use of this macro is in the form: DLION DLI. That address is stored in locations VDSLST, $200-$201 (Lines 810-840). Recall that if you're using multiple DLIs in the same screen, each DLI must itself store the address of the next DLI in VDSLST. Of course, it's still up to you to indicate the mode lines where you want the DLIs to occur, by setting bit 7 of each mode line instruction in the display list. See issue 46 for a DLI refresher.

As you might expect, the DLIOFF macro simply clears bit 7 in NMIEN if it's already set. Be careful, though. If you use DLIOFF before DLION, you can actually enable DLIs rather than disabling them. If you use these macros in the sensible order, all will be dandy.

Now to the player/missile graphics aids. SETPCOLOR (Lines 1020-1410) is virtually identical to our old SETCOLOR macro. However, SETPCOLOR sets one of the player color registers, whereas SETCOLOR processes a playfield color register. The four-player color registers are at addresses $2C0-$2C3, PCOLR0-PCOLR3. In case you ever need to change player colors using display list interrupts, these locations are the shadow registers for COLPM0-COLPM3 at $D012-$D015. Each color register controls the color of both a specific player and the missile associated with that player. Use SETPCOLOR just like you would SETCOLOR, with three parameters for the player number (0-3), hue (0-15) and luminance (0-15). Each parameter can be either a value or an address containing the values to be used.

The PWIDTH macro, Lines 1450-1660, lets you set each player independently to be normal (8 pixels), double (16 pixels), or quadruple (32 pixels) wide. Parameter 1 is the player number (0-3), and parameter 2 is the width to use (1, 2, or 4). The width of each player is determined by the bit pattern stored in bits 0 and 1 at addresses SIZEP0—SIZEP3 ($D008-$D00B). A bit pattern of 00 or 10 selects normal width; 01 doubles the player's width; and 11 produces quadruple width.

Several steps are required to actually enable player/missile graphics even after you've set up the player shapes, sizes and positions. Macro PMGON (Lines 1700-1880) does the dirty work. It takes one parameter, the address of the beginning of the block of RAM you reserved for PMG storage. Amazingly, I always give this address the label "PMG." Lines 1810-1820 tell the operating system where to find the PMG data. Line 1830 turns on players and missiles by setting bits 0 and 1 in GRACTL, $D01D. Lines 1840-1860 set bits 2 and 3 in SDMCTL, $22F, also required to activate PMG. Isn't a single statement like "PMGON PMG" a lot easier to remember than all this other junk? That's what macros are for.

Of course, the next macro is called PMGOFF, in Lines 1920-2020. It simply undoes most of what PMGON accomplished. No parameters are needed.

You probably recall that players can be displayed in either single-line or double-line resolution. The default is double-line, which means that each bit pattern in the player shape definition table shows up on two adjacent scan lines. The PMGRES macro in Lines 2060-2230 lets you choose the desired resolution. The parameter can either be 1 for single-line or 2 for double-line players. Recall also that PMG RAM al-

Boot Camp

All we do now is wait until Bonzo hits the car with a missile.

location and usage is different depending on the resolution you're using. Refer to issue 48 to refresh your memory.

You can also control the horizontal position of each player and missile, independently. I have two macros for these purposes, HPLAYER in Lines 2270-2430 and HMISSILE in Lines 2470-2630. These work in exactly the same way. Two parameters are needed, the player number (0-3) and the desired horizontal position, a value from 0-255. It wouldn't be difficult to modify these macros to accept as parameter 2 an address containing the desired horizontal position; give it a try. Remember that horizontal position values below about 48 and above 208 probably won't be visible on your TV or monitor screen.

Two sets of addresses are used in each of these macros. Locations HPOSP0-HPOSP3 ($D000-$D003) control horizontal positions for players, and HPOSM0-HPOSM3 ($D004-$D008) are used for missiles. However, these addresses are "write-only." You can't find out where a player is by peeking at the contents of one of these addresses. Hence, I set up parallel sets of data storage locations called XPOSP0-XPOSP3 and XPOSM0-XPOSM3. The HPLAYER and HMISSILE macros assume that you've done the same, and you'll get an undefined label error if you omit this step. Today's sample program will show what I mean.

Setting the widths of missiles is a bit more convoluted. A missile is just a 2-bit wide analog of the 8-bit wide player. Only one address, SIZEM ($D00C), is devoted to controlling missile widths. Bits 0 and 1 handle missile 0, bits 2-3 are for missile 1, bits 4-5 apply to missile 2, bits 6-7 take care of missile 3. The pattern in each pair of bits again controls the missile width: 00 and 10 are normal; 01 is double; and 11 is quadruple.

The MWIDTH macro first creates the desired bit pattern based on the value in parameter 2 (1, 2, or 4). The value of parameter 1 tells us which missile to set. The loop in Lines 2890-2960 shifts the desired bit pattern two bits to the left (more significant direction) until the bit pattern is in position corresponding to the correct missile . For example, for missile 0 we don't do any shifting, and for missle 2 we shift the pattern a total of four times (two passes through the loop), until our pattern is in bits 4-5. The resulting bit pattern is stored temporarily at address @TEMP within the macro definition (Line 2800). Finally, Lines 2980-3000 take the current contents of SIZEM, use the ORA instruction to set the desired two bits based on the contents of @TEMP (leaving the other six bits of SIZEM unchanged), and store the result back in SIZEM.

Confused? So was I. That's why I wrote the macro. Now I don't have to remember how it works every time I want to set the width of a missile. I simply let the computer do the thinking, while I try to handle the creativity end of business.

Our final macro sets the width of the playfield to normal (40 Graphics 0 characters), narrow (32 characters), or wide (48 characters) width. The playfield, of course, is the area of the monitor screen used for display of text, graphics and players. Our old friend SDMCTL ($22F) is the main actor here again. The PLFIELD macro in Lines 3060-3390 requires one parameter to specify the desired width. A parameter of 0 turns off the display screen entirely, 1 is for narrow, 2 for standard and 3 for the wide playfield. The bit pattern in bits 0 and 1 of SDMCTL controls the playfield setting. A value of 00 means off, 01 is narrow, 10 is standard and 11 is wide.

The logic in the PLFIELD macro gets a little harrowing. It turns out to be a little tricky to simply set and clear specific bits in a byte, without affecting other contents of the byte. The AND, ORA and EOR instructions are useful, but you have to think carefully about what they do and in what order to use them. In the case of the narrow playfield, for example, I want to clear bit 1 and set bit 0. I chose a rather odd method to do this, but it works. Lines

3230-3240 perform two LSR (Logical Shift Right) operations. This simply throws away the contents of bits 0 and 1, while shifting the remaining six bits two positions to the right. Then two ASL (Accumulator Shift Left) instructions put the six high-order bits back where they belong and clear both bits 0 and 1. After that I use the ORA instructions in Line 3270 to selectively set bit 0. Whew!

You may wonder why I gave this last macro the awkward name of PLFIELD. Why not just come right out and say PLAYFIELD? Well, I tried PLAYFIELD. Unfortunately, MAC/65 interpreted this as a PLA instruction followed by YFIELD as a piece of data. So, I tried PLYFIELD, thinking that at least PLY isn't a 6502 mnemonic. Right, except that MAC/65 supports some extra opcodes that apply only to an enhanced NCR 65C02 microprocessor, and PLY happens to be such an instruction. It means to pull the Y-register from the stack. Hence, the more contrived PLFIELD. The moral is to be careful when naming macros, so MAC/65 doesn't misinterpret your macro name as some bizarre kind of instruction.

So now your toolbox is crammed with even more goodies. Let's see some of these babies in action.

Revenge of Bonzo

Remember Bonzo? He's the little guy with the death wish from "Attack of the Suicidal Road-Racing Aliens." Bonzo's changed his tune, and he's out to get back at the cars that kept doing him in. Today's sample program lets Bonzo shoot back at the cars. We'll see how to manipulate missiles, and how easy it is to set up a graphics program using these new (and some old) macros. In fact,the program in Listing 2 uses about 20 macros. As a special treat, I'll show you how to create the famous Atari rainbow character effect.

Please type in Listing 2. You'll have to assemble this program to disk, rather than just to memory, which might slow things down a bit. If you're using the RAM disk, assemble to some file on drive D8: using a command like: ASM,,#D8:BC58.OBJ. Don't forget to save a copy of the source code on disk before you BLOAD the assembled object code. Otherwise, the object file might overwrite the tail end of your source code. If you aren't using a RAM disk, change the drive designations for the .INCLUDE statements in Lines 210, 220, and 2850.

Here's the plan. Bonzo will remain at the bottom of the screen, and you can move him left or right within specified boundaries using a joystick in Port 1. A car will move across the screen from left to right. Bonzo shoots a missile at the car whenever you press the fire button on the joystick. If Bonzo scores a hit, the car explodes and a message appears. You can then either press START to play again or press RESET to exit from the program.

We'll use a VBI to handle movement of the car, Bonzo, and the missile. I've also created a special shape for Bonzo to assume when he's actually firing the missile. The VBI will copy that form into the PMG RAM whenever you press the joystick fire button. Our main program sets up the PMG environment, waits for a collision, and handles the post-collision activities.

Of course, we need to .INCLUDE the two macro library files we've built, Lines 210-220. Some equates appear in Lines 260-330. You've seen most of these before. STRIG0 ($284) reads the joystick trigger (fire button). M0PL ($D008) checks for collisions between missile 0 (fired by Bonzo as player 0) and players.

I put the PMG dedicated RAM block (2K for single-line resolution) at address $3000 in Line 390. The .DS directives reserve chunks of RAM for each player and the missiles. The three pages from PMG to MIS aren't used in this program. My work variables which keep track of the horizontal and vertical positions of the players and missiles appear at the end of the PMG block, as do bytes to specify the limits of motion at the edges of the screen.

Vertical Blanking

The VBI routine, begins at $4000 (Line 590). There's quite a bit of unused space between the top of the PMG block and the beginning of the VBI, which might come in handy if you have a really large program. Much of the VBI code is adapted from the Boot Camp column in issue 49. Storing something in ATRACT (Line 620) prevents the computer from going into attract mode if no key is pressed for several minutes. Lines 630-650 move the car (Player 1) one pixel to the right. Lines 660-830 handle the left/right movement of Bonzo, making sure he doesn't go past the boundaries I set in the main program.

The MOVEMISSILE routine beginning at Line 840 checks to see if the missile has been fired already, indicated if the horizontal position (XPOSM0) is not zero. If so, the missile is moved upward using the method we covered in previous issues (see Lines 1010-1140) until it hits the top boundary. When it hits the top, Lines 900-990 reset the horizontal position to zero (off-screen) and zero out the missile section in the PMG RAM block to clear out any junk. Then we go to CHKTRIG to see if the fire button is being pressed.

If the fire button is pressed, location STRIG0 will contain a 0. Otherwise, it contains a 1. If the button isn't pressed, Line 1170 branches down to COPYBONZO at Line 1310. There the standard Bonzo shape is copied to the RAM block for player 0. I do this every time just in case the last shape displayed was the shooting form. We don't wan't the shooting shape to remain forever once it is first drawn, now, do we?

If you're pressing the fire button, the shooting shape stored at address SHOOTER (Lines 2680-2710) is copied into PMG RAM using the MOVE macro, Line 1190. If the missile is already fired we don't shoot another one. However, if it hasn't been fired yet, Lines 1220-1270 copy the missile form (defined in Lines 2780-2790) into the

Boot Camp

Confused? So was I. That's why I wrote the macro.

PMG RAM block and set the horizontal position to look like Bonzo really fired it. As with any VBI routine, the graceful way to exit is by jumping through the XITVBV ($E462) vector, Line 1350.

The Main Routine

As usual, the main program begins execution at address $5000, line 1410. Since I've termed this starting point START (creative labels, eh?), you could make this program autorun on loading, using the method we discussed last month.

The first orders of business are to set up a full screen of Graphics 2 and set the boundaries for player and missile movement (Lines 1450-1480). Lines 1490-1580 zero the required portions of the reserved PMG RAM block. The statements in Lines 1680-1820 set up the PMG environment. The player shapes are defined in Lines 2580-2610 (Bonzo) and 2630-2660 (the car).

Bonzo is yellow and the car is pink. Both players are single resolution, on a standard width playfield. Bonzo is normal width and the car is double width. The missile Bonzo fires will be normal width. After enabling player/missile graphics in Line 1770, Bonzo is moved to the middle of the screen. Both the car and Bonzo's missile begin offstage, at a horizontal position of 0. Finally, Line 1820 begins execution of the VBI routine, and the car starts to move across the screen. Now you can move Bonzo using the joystick and fire when ready, Gridley.

All we do now is wait until Bonzo hits the car with a missile. The loop in Lines 1910-1940 simply tests for this condition forever. Don't forget to reset the collision registers as in Line 1900 before checking for a new collision. We talked about collision detection in issue 50.

When Bonzo scores a hit, the game is over. First I turn off the VBI routine in Line 2020 so all player and missile movement ceases. The missile is moved offscreen in Lines 2030-2040. I replace the car shape with a wrecked car shape (WRECK, defined in Lines 2730-2760), Line 2050. The FOR/NEXT loop in Lines 2060-2110 simply changes the color of the wreck from bright to dark red and back rapidly ten times, pausing for three jiffies after each color change. This gives sort of a flashing explosion effect.

Lines 2220-2250 print some messages on the screen, which are defined in Lines 2520-2560. Notice that I've used characters in those lines to select different color registers for the different text lines.

Lines 2260-2340 are all it takes to generate the well-known Atari rainbow effect. It works by simply incrementing the value stored in a particular hardware color register. Line 2290 waits for horizontal synchronization before actually effecting the color change. The result is a new color on each scan line, moving down the screen at about 60 scan lines per second. By changing the offset in the Y-register (Line 2270) and/or the base address being affected (Line 2310), you can produce this effect in any of the playfield or player color registers.

The rainbow continues until you either press the START button to play again (Lines 2320-2330) or the RESET button to exit from the program entirely. We talked about reading the console buttons in issue 44. When START is pressed, Lines 2430-2460 close the screen IOCB, reset the collision registers, turn off player/missile graphics, and go back to let Bonzo get some more revenge.

Closing Argument

As you can see, ladies and gentlemen of the jury, macros make assembly programming much faster, easier and cleaner. It doesn't take an Atari expert to write effective graphics programs when the right macros are available. I ask you to find in favor of the macro assembler, and to purchase one if you plan to continue your pursuit of 6502 assembly language on the 8-bit Atari computers. I thank you.

Isn't a single statement like "PMGON PMG" a lot easier to remember than all this other junk? That's what macros are for.

Listing 1: Assembly

```
0100 ;Graphics macros for MAC/65
0110 ;by Karl E. Wiegers
0120 ;
0130 ;*********************************
0140 ;
0150 ;equates needed by macros
0160 ;
0170 VDSLST = $0200
0180 SDMCTL = $022F
0190 PCOLR0 = $02C0
0200 HPOSP0 = $D000
0210 HPOSM0 = $D004
0220 SIZEP0 = $D008
0230 SIZEM = $D00C
0240 GRACTL = $D01D
0250 PMBASE = $D407
0260 NMIEN = $D40E
0270 SETVBV = $E45C
0280 XITVBV = $E462
0290 ;
0300 ;*********************************
0310 ;
0320 ;VBION macro
0330 ;
0340 ;Usage:  VBION address
0350 ;
0360 ;'address' is the address or
0370 ;label for the beginning of your
0380 ;deferred VBI routine
0390 ;
0400      .MACRO VBION
0410        .IF %0<>1
0420        .ERROR "Error in VBION"
0430        .ELSE
0440        LDY # <%1
0450        LDX # >%1
0460        LDA #7
0470        JSR SETVBV
0480        .ENDIF
0490      .ENDM
0500 ;
0510 ;*********************************
0520 ;
0530 ;VBIOFF macro
0540 ;
0550 ;Usage:  VBIOFF
0560 ;
0570      .MACRO VBIOFF
0580      LDY # <XITVBV
0590      LDX # >XITVBV
0600      LDA #7
0610      JSR SETVBV
0620      .ENDM
0630 ;
0640 ;*********************************
0650 ;
0660 ;DLION macro
0670 ;
0680 ;Usage:  DLION address
0690 ;
0700 ;'address' is the starting
0710 ;address of the DLI routine to
0720 ;be executed
0730 ;
0740      .MACRO DLION
0750        .IF %0<>1
0760        .ERROR "Error in DLION"
0770        .ELSE
0780        LDA NMIEN
0790        ORA #$80
0800        STA NMIEN
0810        LDA # <%1
0820        STA VDSLST
0830        LDA # >%1
0840        STA VDSLST+1
0850        .ENDIF
0860      .ENDM
0870 ;
0880 ;*********************************
0890 ;
0900 ;DLIOFF macro
0910 ;
0920 ;Usage:  DLIOFF
0930 ;
0940      .MACRO DLIOFF
0950      LDA NMIEN
0960      EOR #$80
0970      STA NMIEN
0980      .ENDM
0990 ;
1000 ;*********************************
1010 ;
1020 ;SETPCOLOR macro
1030 ;
1040 ;Usage:  SETPCOLOR p#,hue,lum
1050 ;
1060 ;p# is player number (0-3)
1070 ;hue is color number (0-15)
1080 ;lum is luminance value (0-15)
1090 ;all can be values or addresses
1100 ;
1110      .MACRO SETPCOLOR
1120        .IF %0<>3
1130        .ERROR "Error in SETPCOLOR"
1140        .ELSE
1150          .IF %1>3
1160          LDX %1
1170          .ELSE
1180          LDX #%1
1190          .ENDIF
1200          .IF %2>15
1210          LDA %2
1220          ASL A
1230          ASL A
1240          ASL A
1250          ASL A
1260          .ELSE
1270          LDA #%2*16
1280          .ENDIF
1290          .IF %3>15
1300          LDY %3
1310          .ELSE
1320          LDY #%3
```

```
1330          .ENDIF
1340        STA PCOLR0,X
1350        TYA
1360        AND #$0F
1370        CLC
1380        ADC PCOLR0,X
1390        STA PCOLR0,X
1400        .ENDIF
1410      .ENDM
1420 ;
1430 ;*********************************
1440 ;
1450 ;PWIDTH macro
1460 ;
1470 ;Usage:  PWIDTH p#,width
1480 ;
1490 ;p# is player number (0-3)
1500 ;width is width factor (1,2,4)
1510 ;
1520      .MACRO PWIDTH
1530        .IF %0<>2
1540        .ERROR "Error in PWIDTH"
1550        .ELSE
1560        LDX #%1
1570        LDA #0
1580          .IF %2=2
1590          LDA #1
1600          .ENDIF
1610          .IF %2=4
1620          LDA #3
1630          .ENDIF
1640        STA SIZEP0,X
1650        .ENDIF
1660      .ENDM
1670 ;
1680 ;*********************************
1690 ;
1700 ;PMGON macro
1710 ;
1720 ;Usage:  PMGON address
1730 ;
1740 ;'address' is the address of the
1750 ;reserved PMG RAM block
1760 ;
1770      .MACRO PMGON
1780        .IF %0<>1
1790        .ERROR "Error in PMGON"
1800        .ELSE
1810        LDA # >%1
1820        STA PMBASE
1830         POKE  GRACTL,3
1840        LDA SDMCTL
1850        ORA #$0C
1860        STA SDMCTL
1870        .ENDIF
1880      .ENDM
1890 ;
1900 ;*********************************
1910 ;
1920 ;PMGOFF macro
1930 ;
1940 ;Usage:  PMGOFF
1950 ;
1960      .MACRO PMGOFF
1970      LDA #0
1980      STA GRACTL
1990      LDA SDMCTL
2000      EOR #$0C
2010      STA SDMCTL
2020      .ENDM
2030 ;
2040 ;*********************************
2050 ;
2060 ;PMGRES macro
2070 ;
2080 ;Usage:  PMGRES res
2090 ;
2100 ;res is 1 for single-line, 2
2110 ;for double-line resolution
2120 ;
2130      .MACRO PMGRES
2140        .IF %0<>1
2150        .ERROR "Error in PMGRES"
2160        .ELSE
2170          .IF %1=1
2180          LDA SDMCTL
2190          ORA #$10
2200          STA SDMCTL
2210          .ENDIF
2220        .ENDIF
2230      .ENDM
2240 ;
2250 ;*********************************
2260 ;
2270 ;HPLAYER macro
2280 ;
2290 ;Usage:  HPLAYER p#,X
2300 ;
2310 ;p# is player number (0-3)
2320 ;X is horizontal position
2330 ;
2340      .MACRO HPLAYER
2350        .IF %0<>2
2360        .ERROR "Error in HPLAYER"
2370        .ELSE
2380        LDX #%1
2390        LDA #%2
2400        STA HPOSP0,X
2410        STA XPOSP0,X
2420        .ENDIF
2430      .ENDM
2440 ;
2450 ;*********************************
2460 ;
2470 ;HMISSILE macro
2480 ;
2490 ;Usage:  HMISSILE m#,X
2500 ;
2510 ;m# is missile number (0-3)
2520 ;X is horizontal position
2530 ;
2540      .MACRO HMISSILE
2550        .IF %0<>2
2560        .ERROR "Error in HMISSILE"
2570        .ELSE
2580        LDX #%1
2590        LDA #%2
2600        STA HPOSM0,X
2610        STA XPOSM0,X
2620        .ENDIF
2630      .ENDM
2640 ;
2650 ;*********************************
2660 ;
2670 ;MWIDTH macro
2680 ;
2690 ;Usage:  MWIDTH m#,width
2700 ;
2710 ;m# is missile number (0-3)
2720 ;width is 1, 2, or 4
2730 ;
2740      .MACRO MWIDTH
2750        .IF %0<>2
2760        .ERROR "Error in MWIDTH"
2770        .ELSE
2780        CLC
2790        BCC @SKIPMWIDTH
2800 @TEMP .BYTE 0
2810 @SKIPMWIDTH
2820        LDA #0
2830          .IF %2=2
2840          LDA #1
2850          .ENDIF
2860          .IF %2=4
2870          LDA #3
2880          .ENDIF
2890        LDY #%1
2900        BEQ @SHDONE
2910 @SHLOOP
2920        ASL A
2930        ASL A
2940        DEY
```

```
2950        BNE @SHLOOP
2960 @SHDONE
2970        STA @TEMP
2980        LDA SIZEM
2990        ORA @TEMP
3000        STA SIZEM
3010        .ENDIF
3020     .ENDM
3030 ;
3040 ;*******************************
3050 ;
3060 ;PLFIELD macro
3070 ;
3080 ;Usage:  PLFIELD width
3090 ;
3100 ;'width' is 0 to turn screen off,
3110 ;1 for narrow playfield, 2 for
3120 ;standard, 3 for wide
3130 ;
3140     .MACRO PLFIELD
3150       .IF %0<>1
3160       .ERROR "Error in PLFIELD"
3170       .ELSE
3180       LDA SDMCTL
3190         .IF %1=0
3200         LDA #0
3210         .ENDIF
3220         .IF %1=1
3230         LSR A
3240         LSR A
3250         ASL A
3260         ASL A
3270         ORA #1
3280         .ENDIF
3290         .IF %1=2
3300         LSR A
3310         ORA #1
3320         ASL A
3330         .ENDIF
3340         .IF %1=3
3350         ORA #3
3360         .ENDIF
3370       STA SDMCTL
3380       .ENDIF
3390     .ENDM
```

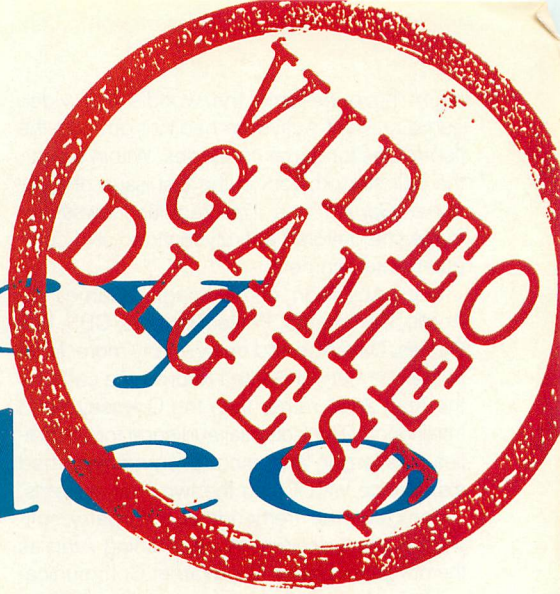
Listing 2: Assembly

```
0100 ;Demonstration of player/missile
0110 ;graphics macros
0120 ;
0130 ;by Karl E. Wiegers
0140 ;
0150     .OPT OBJ,NO LIST
0160 ;
0170 ;*******************************
0180 ;  PULL IN MACRO LIBRARIES
0190 ;*******************************
0200 ;
0210     .INCLUDE #D8:MACRO.LIB
0220     .INCLUDE #D8:GRAPHICS.LIB
0230 ;
0240 ;equates we need today
0250 ;
0260 ATRACT = $4D
0270 STICK0 = $0278
0280 STRIG0 = $0284
0290 M0PL =  $D008
0300 COLPF0 = $D016
0310 HITCLR = $D01E
0320 CONSOL = $D01F
0330 WSYNC = $D40A
0340 ;
0350 ;*******************************
0360 ;  SET UP PMG STORAGE
0370 ;*******************************
0380 ;
0390     *=  $3000
0400 ;
0410 PMG .DS $0300
0420 MIS .DS $0100
0430 PL0 .DS $0100
0440 PL1 .DS $0100
0450 PL2 .DS $0100
0460 PL3 .DS $0100
0470 XPOSP0 .DS 4
0480 YPOSP0 .DS 4
0490 XPOSM0 .DS 4
0500 YPOSM0 .DS 4
0510 LEFT .DS 1
0520 RIGHT .DS 1
0530 TOP .DS 1
0540 ;
0550 ;*******************************
0560 ;  VBI ROUTINE STARTS HERE
0570 ;*******************************
0580 ;
0590     *=  $4000
0600 ;
0610 VBI
0620      POKE  ATRACT,0
0630     INC XPOSP0+1 ;move car 1
0640     LDA XPOSP0+1 ;pixel to right
0650     STA HPOSP0+1
0660     LDA STICK0  ;get stick 1
0670     AND #4      ;left?
0680     BNE CHKRIGHT ;no,check right
0690     LDA XPOSP0  ;yes - at left
0700     CMP LEFT    ;edge?
0710     BEQ MOVEMISSILE ;yes, go on
0720     DEC XPOSP0  ;no, move Bonzo
0730      POKE  HPOSP0,XPOSP0 ;to left
0740     BNE MOVEMISSILE ;go on
0750 CHKRIGHT
0760     LDA STICK0  ;get stick 1
0770     AND #8      ;right?
0780     BNE MOVEMISSILE ;no, go on
0790     LDA XPOSP0  ;yes - at right
0800     CMP RIGHT   ;edge?
0810     BEQ MOVEMISSILE ;yes, go on
0820     INC XPOSP0  ;no, move him to
0830      POKE  HPOSP0,XPOSP0 ;right
0840 MOVEMISSILE
0850     LDA XPOSM0  ;missile fired?
0860     BEQ CHKTRIG ;no, check trig
0870     LDA YPOSM0  ;yes - at the
0880     CMP TOP     ;top?
0890     BNE MOVEM   ;no, move it
0900      POKE  XPOSM0,0 ;yes - move
0910      POKE  HPOSM0,0 ;missile
0920      POKE  YPOSM0,$B8 ;offscreen
0930     LDX #0      ;zero out
0940     TXA         ;missile 1 area
0950 ZMISSILE
0960     STA MIS,X
0970     INX
0980     CPX #$B8
0990     BNE ZMISSILE
1000     BEQ CHKTRIG ;check trigger
1010 MOVEM
1020     LDA # >MIS  ;move missile
1030     STA MOVEFROM+1 ;up 1 scan
1040      POKE  MOVEFROM,YPOSM0 ;line
1050     LDY #1
1060 LOOPUP
1070     LDA (MOVEFROM),Y
1080     DEY
1090     STA (MOVEFROM),Y
1100     INY
1110     INY
1120     CPY #10     ;missile is 10
1130     BNE LOOPUP  ;bytes tall
1140     DEC YPOSM0
1150 CHKTRIG
1160     LDA STRIG0  ;trigger pressed?
1170     BNE COPYBONZO ;no, go on
1180 ;yes-copy shooting form of Bonzo
1190      MOVE  SHOOTER,PL0+$C0,17
1200     LDA XPOSM0  ;missile fired?
1210     BNE VBIEXIT ;yes, exit
1220     LDX XPOSP0  ;no, copy missile
1230     INX         ;form into PMG
```

Boot Camp

```
1240      INX           ;and move to
1250      STX HPOSM0    ;Bonzo's location
1260      STX XPOSM0
1270       MOVE  MISSILE,MIS+$B8,10
1280      CLC
1290      BCC VBIEXIT
1300 ;copy normal Bonzo form
1310 COPYBONZO
1320       MOVE  BONZO,PL0+$C0,17
1330 ;leave VBI routine
1340 VBIEXIT
1350      JMP XITVBV
1360 ;
1370 ;*********************************
1380 ;  MAIN PROGRAM STARTS HERE
1390 ;*********************************
1400 ;
1410      *=  $5000
1420 ;
1430 START
1440      CLD           ;binary mode
1450       GRAPHICS  2+16 ;open screen
1460       POKE  LEFT,56 ;set limits
1470       POKE  RIGHT,191
1480       POKE  TOP,30
1490      LDX #0        ;zero PMG area
1500      TXA
1510 INIT
1520      STA MIS,X
1530      STA PL0,X
1540      STA PL1,X
1550      STA PL2,X
1560      STA PL3,X
1570      INX
1580      BNE INIT
1590 ;
1600 ;-------------------------------
1610 ;now point to PMG area, move
1620 ;car and Bonzo shapes into PMG
1630 ;RAM, set colors, widths, and
1640 ;positions, and resolution, and
1650 ;turn on PMG and VBI
1660 ;-------------------------------
1670 ;
1680       MOVE  BONZO,PL0+$C0,17
1690       MOVE  CAR,PL1+$80,16
1700       SETPCOLOR  0,1,12
1710       SETPCOLOR  1,5,6
1720       PLFIELD  2
1730       PMGRES  1
1740       PWIDTH  0,1
1750       PWIDTH  1,2
1760       MWIDTH  0,1
1770       PMGON  PMG
1780       HPLAYER  0,120
1790       HPLAYER  1,0
1800       HMISSILE  0,0
1810       POKE  YPOSM0,$B8
1820       VBION  VBI
1830 ;
1840 ;-------------------------------
1850 ;clear collision registers;
1860 ;loop until you get a collision
1870 ;between the missile and the car
1880 ;-------------------------------
1890 ;
1900       POKE  HITCLR,0
1910 CHKCOL
1920      LDA M0PL
1930      AND #2
1940      BEQ CHKCOL
1950 ;
1960 ;-------------------------------
1970 ;when collide, turn off VBI,
1980 ;move missile offstage, copy
1990 ;wreck shape on car; flash colors
2000 ;-------------------------------
2010 ;
2020       VBIOFF
2030       POKE  HPOSM0,0
2040       POKE  XPOSM0,0
2050       MOVE  WRECK,PL1+$80,18
2060       FOR  I,1,10
2070       SETPCOLOR  1,4,12
2080       PAUSE  3
2090       SETPCOLOR  1,4,2
2100       PAUSE  3
2110       NEXT  I
2120 ;
2130 ;-------------------------------
2140 ;change color registers, print
2150 ;messages, turn on rainbow for
2160 ;color register 0, wait for press
2170 ;of START or RESET keys
2180 ;-------------------------------
2190 ;
2200       SETCOLOR  1,5,8
2210       SETCOLOR  2,12,8
2220       POSITION  4,0
2230       PRINT  6,WINNER
2240       POSITION  1,2
2250       PRINT  6,WHATNEXT
2260 RAINBOW
2270      LDY #0
2280      INX
2290      STX WSYNC
2300      TXA
2310      STA COLPF0,Y
2320      LDA CONSOL
2330      CMP #6
2340      BNE RAINBOW
2350 ;
2360 ;-------------------------------
2370 ;close screen, reset collision
2380 ;registers, turn off PMG, go back
2390 ;and start it all over if START
2400 ;was pressed
2410 ;-------------------------------
2420 ;
2430       CLOSE  6
2440       POKE  HITCLR,0
2450       PMGOFF
2460      JMP START
2470 ;
2480 ;-------------------------------
2490 ;lines to print, player shapes
2500 ;-------------------------------
2510 ;
2520 WINNER
2530      .BYTE "BONZO WINS!",EOL
2540 WHATNEXT
2550      .BYTE "start to go again "
2560      .BYTE "    RESET TO EXIT",EOL
2570 ;
2580 BONZO
2590      .BYTE 0,60,24,126,189
2600      .BYTE 189,189,189,60,60
2610      .BYTE 36,36,36,102,0,0,0
2620 ;
2630 CAR
2640      .BYTE 0,0,126,195,219,219
2650      .BYTE 91,219,219,219,219
2660      .BYTE 91,219,219,195,126
2670 ;
2680 SHOOTER
2690      .BYTE 24,24,36,66,129,189
2700      .BYTE 153,126,60,60,60,60
2710      .BYTE 60,36,66,36,102
2720 ;
2730 WRECK
2740      .BYTE 20,89,98,86,156,41
2750      .BYTE 86,146,89,108,184,86
2760      .BYTE 40,84,86,8,16,32
2770 ;
2780 MISSILE
2790      .BYTE 1,1,1,1,1,1,1,1,1,0
2800 ;
2810 ;*********************************
2820 ;   DON'T FORGET THE SUBS!
2830 ;*********************************
2840 ;
2850      .INCLUDE #D8:SUBS.LIB
```

The History of Video Games, Part III



The Golden Age

by Arnie Katz, Joyce Worley & Bill Kunkel

In chapter two of our story, we left a somewhat bemused George Plimpton trying to sound enthusiastic about video games. Meanwhile, somewhere in Silicon Valley, a feisty marketing guy from the record industry, Jim Levy, is enjoying lunch with four Interesting Strangers.

Upon closer inspection, the four interesting strangers turned out to be Alan Miller, David Crane, Bob Whitehead and Larry Kaplan. There is no record of what they and Levy ordered, but the result of their meetings burst upon the video game market like a bombshell.

At the core of Levy's concept was his analogy between the music and video game industries. Just as some companies make stereos and others distribute records, he reasoned, a company could make video game software without marketing a hardware system of its own.

Whitehead, Crane, Miller and Kaplan joined Levy in a new venture called Activision, which opened its doors for business in 1980. It may surprise younger video gamers, but Atari was not pleased by the infant publisher's decision to produce cartridges for the VCS (2600) system. The industry leader viewed the upstart as competition, rather than support. Determined to wipe out the enemy on the beaches, Atari mounted a campaign to convince consumers and the industry that only the outfit which created the VCS ought to manufacture cartridges for it. Activision overcame all obstacles, including a ridiculous rumor that ACTV carts would damage the VCS console, and its first group of four titles reached retail stores before the end of 1980.

No one was more enthusiastic about the debut of Activision than Arnie Katz and Bill Kunkel, who had started a monthly video game review column in *VIDEO* magazine. Atari's output was barely sufficient to fill the alloted editorial space. The world's first video game critics

might have become the world's last video game critics if Activision had not opened the floodgates for game cartridges. Within a year, the pair started work in the first issue of *Electronic Games*, the first regular newsstand magazine devoted entirely to video and computer entertainment.

After Activision came Imagic, Games by Apollo, CommaVid, Parker Brothers, CBS, Fox Games, Data Age and many, many more. Most of the companies elected to produce carts for the Atari 2600 rather than the Odyssey2 and Intellivision. The potential audience for 2600 titles was larger, and more programmers had experience with it than the two rival systems.

Ironically, the tremendous third-party software proved decisive in establishing Atari as the dominant brand. The Warner Communications division was forced to reverse gears and start touting the great number of games made for the 2600 by other companies.

Down at the Arcade

Business boomed in the coin-op field. Pinball arcades transformed into plush family amusement centers overnight as the quarter-snatchers lured adults as well as children with play-for-pay video games.

The only cloud on the horizon was that arcaders were almost all male. Women didn't seem to enjoy the complex shooting games as much as men, and they tended to play video games in the home, if at all.

The antidote to female arcade apathy arrived in America in 1981 after conquering the hearts of Japanese joystickers. The wocka-wocka sound effects, simple play-mechanics, and low violence quotient of Pac-Man overcame all resistance. The maze-chase format offers more strategic scope than invasion contests like Space Invaders, but it actually requires less physical skill to play. And when people discovered that they could learn the movement patterns for those pesky ghosts, Pac-mania swept America.

The interest in Pac-strategy led to an upswing of interest in this phase of video gaming. Soon, a couple of dozen books were telling score-hungry arcaders how to gobble up Pac-Man and smash Asteroids.

There Once was a Leather Goods Company . . .

The Connecticut Leather Company got into the toy business before World War II through a contract to produce Tom Mix holsters. The company subsequently expanded its line of recreational products to include above-ground swimming pools and electronics games. Along the way, it shortened the corporate name to Coleco.

On June 1, 1982, Coleco introduced its "third generation" video game system, the ColecoVision. Almost overnight, it seemed ColecoVision challenged the previously unassailable 2600 and, in the process, pushed the Odyssey2, Intellivision, and the Astrocade (a re-launch of the Bally Arcade home system) to the sidelines.

Under the leadership of the energetic Michael Katz, who had helped launch stand-alone electronic games at Mattel a few years earlier, Coleco displayed an uncanny ability to find overlooked coin-op gems and translate them beautifully into home cartridges. **Cosmic Avengers, Venture,** and **Lady Bug** were among the titles which flowered in the home video gaming environment. And Coleco also packed **Donkey Kong** with the system, which attracted many new customers.

The ColecoVision games utilized greatly increased memory, as much as 32K, to produce games which looked, sounded, and played better than anything previously available on the market. The first million consoles sold in record time.

Things looked mighty rosy in mid-1983. Manufacturers had sold more than $1 billion worth of video game consoles the previous year, and prospects for a further increase looked good. Software sales, which reached $1.2 billion in 1982, were still climbing. Everyone loved video games, and all was right with the world.

Or maybe not.

Will Atari designers all buy solid-gold Cadillacs? Can Coleco make the Nutmeg State the center of the video game universe? And what are those big bulky boxes? Learn the answers to these and other thought-provoking questions next month in "The History of Video Games, Part IV: The Great Fall."

Hotline—Video Game News Update

Bandai America doesn't want video gamers to turn into couch potatoes. The company plans to keep us lean and mean by playing on the NES machine with the Family Fun Fitness accessory. This rug-controller responds to players jumping, hopping and skipping to move the on-screen cursor. The unusual exercise device/video game controller comes with a special game, the **Athletic World Game Pak**.

Capcom went west, with its latest release for the Nintendo Entertainment System. **Gunsmoke** blew 'em away in the coin-ops, and the home version promises the same kind of high-action cowboy thrills. It's a blast-athon to save a mining town from ruthless bandits.

Broderbund has two new titles to add to the list of Nintendo entertainments. **Deadly Towers** changes the NESser to a prince, then makes him defend the kingdom against Rubas, king of devils. This bell-ringing scourge brings forth armies of demons, dragons and other dreadfuls. There's an element of role-playing to add some depth to the arcade action, plus a welcome boon to joystick jockies: A password system lets the game in progress continue, instead of having to start over every time.

Mixing water with video games is always risky; the list of designers who tried and drowned is about as long as the list of games that attempted to dunk arcade-style action in the briney deep. Broderbund's second new title, **Sqoon**, may break this aquatic tradition. This watery title features aliens who've melted the polar caps to flood Earth. The NESser has to take to the seas in the SQOON submarine to destroy the invaders and save the world.

Activision's designers are dusting off their joysticks, with a dozen titles that earned their place in the video gamer's Hall of Fame. **Pitfall, Grand Prix, Kaboom!, Chopper Command, Keystone Kapers, Ice Hockey, Ghostbusters, River Raid, Enduro, Space Shuttle, Freeway** and **Boxing** are Golden Oldies worth hunting for on your dealer's shelves, and can also be ordered direct from Activision, for the Atari 2600 and 7800 video game machines.

Here's a freebee to call for: Dial Nintendo at 800-422-2602 (206-822-2040 in Washington State) for a copy of their brochure, "The Facts on Video Games From The Man Who Plays Games For a Living." Written by Howard Phillips, Nintendo's product-analysis manager, it gives ten tips on how to purchase video games and other high-tech toys. It also contains a brief history of video games, and some quotes from educators and scientists on the educational and therapeutic use of games—useful ammunition when convincing your parents to pop for a new video game system!

The Nintendo version of Epyx's classic **Winter Games** was written by PONY, a Japanese company, and will be distributed and marketed by Acclaim Entertainment. This version contains four cold-weather sports: Speed Skating, Hot Dog Aerials, Figure Skating, and Bobsled. It's a one-Megabyte ROM cartridge that utilizes split-screen graphics, and optional two-player competition.

Scott Carpenter, one of the original seven Mercury astronauts (second man to orbit the earth, on May 24, 1962), says, "If video games had existed when I was child, I could have exercised those skills (hand/eye coordination and quick reactions) at a much earlier age. It's a fact: The more you play these games, the more your non-verbal skills improve."

Scott Carpenter is currently touring with the Sega Challenge, talking to kids and parents about ways to foster pride and achievement. Carpenter believes video games function as a training device, and that they "can literally provide the secret of self-esteem."

The Sega Challenge is the creation of a group of computer software experts, working in consultation with Professor Philip Merrifield of New York University. They've developed a series of five activities to stimulate kids to see how they can improve on their non-verbal skills. Dubbed The Sega Challenge, it incorporates two video games, plus additional dexterity games which test the players' concentration and ability to learn new skills.

Players get three one-minute tries at **Outrun**, a round of **Shooting Gallery**, then have to try to write as many letters as possible backwards in a contest called "The Write Stuff." Space Balls challenges players to catch only the right colored balls, as they arrive via an airstream, sort, and place them in matching tubes. Finally, the Ring Maze test requires players to maneuver a ring through a metal maze of letters (which spell SEGA, what else?) without touching the maze, which makes a buzzer sound.

The SEGA Challenge is being played in malls and youth centers across the country.

The Atari XE Game System (about $150) comes complete with Atari XE console, keyboard, light gun, one joystick (though there are two ports, so a second controller can be added), plus adapter, cables and switch box. Also included are three games: **Missile Command** (resident in the system's memory banks), **Flight Simulator II**, and **Bug Hunt**, a high-action shoot-'em up. **Atari BASIC** is resident in ROM, and additional peripherals can be attached, including programmer recorder, disk drive, printer, modem, etc.

The 64K game system uses the same controllers that worked with the Atari computer—joystick, Track-Ball, light gun, mouse, or keyboard.

There were literally hundreds of cartridges manufactured for the Atari 400/800 XE/XL computers in the first golden age of computer gaming. There were well over a hundred just from third-party developers, not counting the library of titles developed by Atari. And, of course, with the addition of a disk drive, the game system can draw on a huge library of available software.

Konami has a toll-free hotline for gamers. If you get stuck on one of their games, call the company for a hint. The number is packed with each title. There's also a hint book available that might help in some tight spots.

The newest trilogy of games from Konami put the gamer in the hot seat. **Top Gun** casts the arcader in a navy jet, battling bogeys at Mach2. In **Goonies II**, you have to fight off Ma Fratelli and her two sons, and in **Stinger**, you battle aliens with your jet fighter. This one features simultaneous play for two gamers, or play against the computer. All three titles are for the Nintendo Entertainment System.

Reader Replay Letters from Digest Readers

The Bottom Line

Concerning the video game supplement in your most recent edition, I think it's great. I would probably buy a copy of a revised electronic games magazine *if* I saw one on the store shelves, but feel that having it combined with *ANALOG* is your best bet. By taking *ANALOG* and calling it something like "Electronic Games/Analog Computing" you will be achieving two important points.

First, the game portion is sure to haul in much more advertising than the *ANALOG* section, while readership will increase. Second, and most importantly, the people who would buy "EG/A" for the game portion of the magazine will be introduced (probably for the first time) to a line of real computers in the *ANALOG* section. I'm sure this will revive interest in the Atari computers, especially for those who own only a video game or are planning to buy one in the near future.

Louis J. Ferro, New Jersey

Actually, demographic studies indicate that many video game purchasers are quite familiar with computers already. In fact, many already own and use computers. But while the computer sits in the den in its solitary work station, the video game system is hooked up to the big TV in the family room.

Users interested in family entertainment are likely to be more satisfied with a video game system than an 8-bit computer.

Go VGD!

I'm definitely in favor of Videogame Digest. I think the first outing was very, very good and would love to see it become bigger and better. With Atari, Nintendo, INTV and Sega/Tonka—plus over a dozen software publishers—now in the video game business, I believe VGD could even survive as a magazine itself if it carried a low pricetag and covered the gamut of computer and video game releases. Whatever its form, please continue the Digest, especially to inform us on new and upcoming games and equipment.

Dennis Sellers, Nashville TN

Today Video games; Tomorrow . . . ?

VIDEO GAME DIGEST

VGD is a great idea! I love it! Look, I'll be honest, I can't say anything else about how I feel about this idea, so I'll just give some suggestions that would make this the best magazine around.

I'm glad to hear that you guys are from *Electronic Games (Computer Entertainment)*. I mourned the passing of that magazine. I was taken aback, though, when I read "the world of cartridge games." Why just cartridge? Please cover *all* games! Computer games, arcade video *and* cartridge. And please, if a game isn't good, *say so*. All the reviews in the first VGD sounded suspiciously cheery to me!

Henning Hoffman Waterloo, Ontario CANADA

You needn't worry about any reluctance on our part to produce critical reviews—as several entries in almost any issue will show. Since space is at a premium, however, we concentrated on the superior games in our first installment.

As far as non-cartridge games go: sorry, Henning, this mini-magazine is devoted to home video game systems; at present, we just don't have the room to cover the entire universe of electronic gaming as the subject deserves. But who knows what may happen in the future?

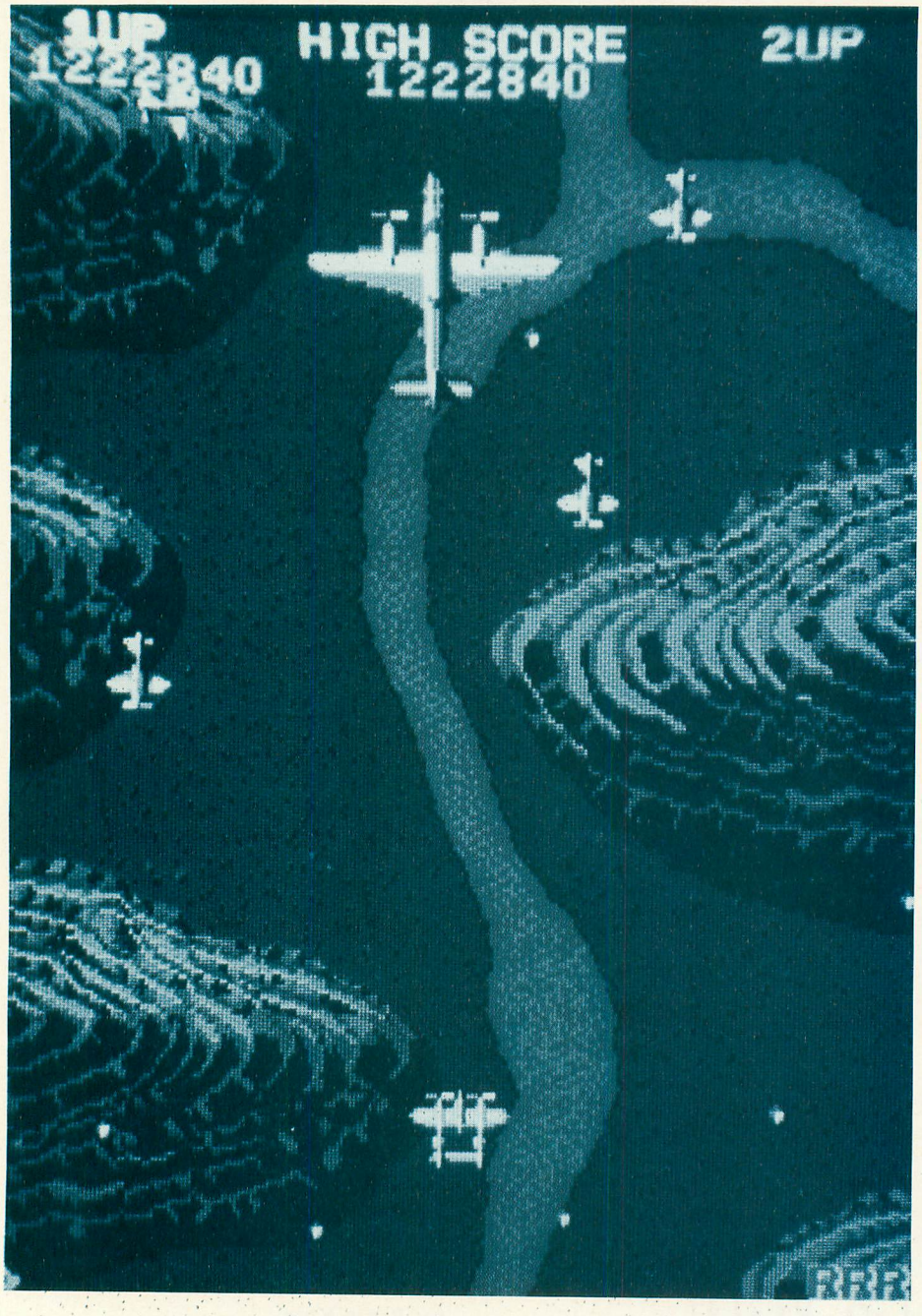
1942
Capcom
1283C Old Mountain View, Alviso Rd.
Sunnyvale, CA 94089
(408) 745-7081
Nintendo Entertainment System; $29.95

by Arnie Katz

There's always room for a good all-out destruction festival, and **1942** is guaranteed to win the heart of every joystick jockey. In this latest entry in Capcom's "Captain Commando" action video game line, the player becomes the pilot of the Super Ace.

At the beginning of play, the Super Ace takes to the skies from the deck of an aircraft carrier to battle the red formations. The player earns bonuses, including improvements for the Super Ace, by destroying the enemy in 32 different scenes. Combat takes place over land and sea as the Super Ace blasts through wave upon wave of deadly foes to reach Tokyo and destroy the remaining Japanese aircraft.

The video gamer employs the control pad to move the Super Ace in the corresponding



direction on the playfield, which scrolls down from the top at a steady, slow rate. Button "B" fires the front-mounted dual cannon at the machine-controlled enemies, which can rush at the player's craft from any edge of the screen.

Button "A" is the Super Ace's main evasion weapon, apart from dodging. Pressing this button permits the Super Ace to "loop the loop," to evade oncoming fire and midair collisions. The Super Ace can only execute a limited number of these special maneuvers, so they must be saved for really tight spots, when simple control-pad movement won't save one of the player's three precious lives.

The key to 1942 is wiping out red formations. When a whole squadron bites the dust, the word "POW" appears on the playfield. Flying over the word gives the player some kind of advantage. The nature of the bonus depends on which formation the armchair pilot destroyed. Some possibilities include extra machine guns, wingmen on each side of Super Ace, and a big batch of points.

The documentation is much too sketchy. Though there are illustrations of Captain Commando and the control unit, it's a safe bet that most gamers would prefer a diagram of the playfield which explained all the on-screen notations.

The graphics, based on an overhead view of the terrain, are decent, but playaction is definitely the focus of 1942. Those who enjoy relentlessly battling against a wide range of opponents will get many hours of explosive fun from this Capcom title. As the song lyric says, 1942 was "a very good year."

BurgerTime
Data East USA
470 Needles Drive
San Jose, CA 95112
(408) 286-7074
Nintendo Entertainment System; $29.95

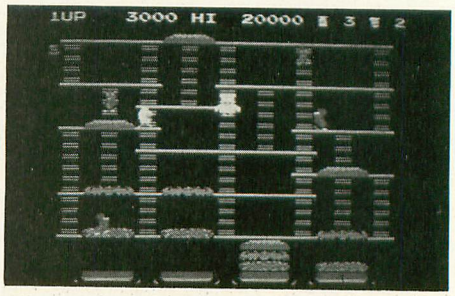
by Arnie Katz

Peter Pepper, the cartoonish hero of **BurgerTime**, is a chef in a peck of trouble. He must prepare a batch of hamburgers while avoiding interference from the "Food Foes," Mr. Hot Dog, Mr. Pickle, and Mr. Egg. That's the premise in this climbing and jumping game, first introduced in coin-op arcades by Data East in 1982.

The playfields of BurgerTime are constructed from arrangements of small horizontal platforms connected by ladders of various heights. Pressing the control pad moves the chef in the corresponding direction. Novice players may find it tough to position Peter Pepper when leaving a ladder for an intersecting platform, but a little practice soon remedies any problem.

Components of hamburgers are stacked on many of the platforms. If the Peter Pepper passes over a platform with an ingredient, it drops a couple of levels. When the two halves of the bun, lettuce, and burger fall to the bottom of the playfield, the gamer gets credit for a complete hamburger. When the chef assembles all the hamburgers on the screen, the game advances to the next playfield.

The Food Foes take one of the player's five lives each time they touch the little chef. Running is the best way to avoid this lethal contact, but Peter Pepper packs a couple of other potent powers. Throwing pepper at a Food Foe momentarily paralyzes it so that the chef can run over it for points. "Momentarily" is the operative term, since the effects of peppering wear off in a few seconds. It is a good idea not to linger over a fallen food foe.



The chef's other tactic is to drop parts of the hamburger onto a Food Foe. If one of the nasties gets buried under a burger, lettuce, or bun, the player collects bonus points.

The player also gains extra points for picking up special objects like ice cream cones and French fries. They appear briefly, but they're worth a little extra effort to snag.

The rulebook is much too vague. Key elements of the game, such as the method for dropping burger components (by walking completely across the platform on which they sit), are not even mentioned! The rules for BurgerTime are not hard to guess, but explaining game procedures in more detail would remove all doubt.

The main characteristic of this one- or two-player contest is nonstop motion. This makes BurgerTime very exciting, even though it lacks blazing lasers and roaring explosions. Excellent animation and charming characters dovetail well with the light-hearted theme.

Considering that it is almost six years old, BurgerTime has held up very well. This NES cartridge is a faithful translation of the coin-op hit, and it should especially appeal to younger video games.

Lode Runner
Broderbund Software
17 Paul Dr.
San Rafael, CA 94903
(415) 492-3200
Nintendo Entertainment System; $29.95

by Arnie Katz

The action-packed advertures of the nimble **LodeRunner** have entertained computerists for several years in a series of three programs published by Broderbund. Now owners of the Nintendo video game system can take command of the intrepid agent as he invades the multi-level fortress of the Bungling Empire in this all-time classic.

The player controls the Lode Runner, who must move back and forth on the horizontal platforms, climb up and down ladders, and shinny along poles to collect all the gold on the level while avoiding the lethal touch of the system-controlled guards. When the Lode Runner has gathered all the loot on a playfield, a previously invisible ladder appears. The well-animated on-screen character scampers up to the next playfield, and the game continues.

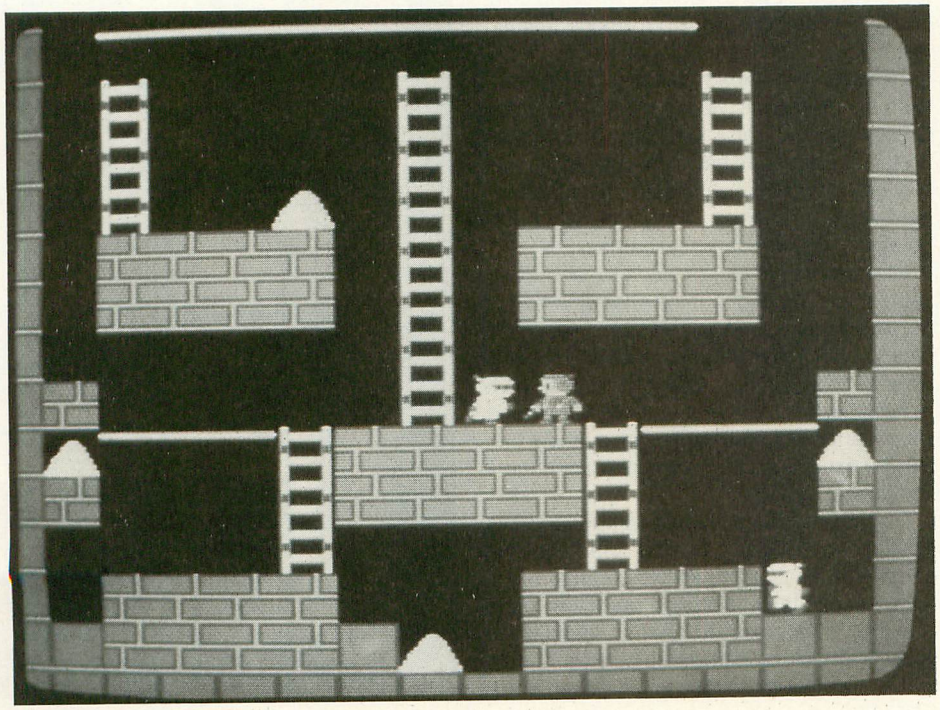
Although the Lode Runner is always outnumbered by the Bungling guards, he has a few useful tricks. The most important is the ability to dig holes in the brick platforms. Although a guard eventually hops out of a pit, unless it closes up and crushes him first, it gives the Lode Runner time to reverse direction and try a different strategy. Button "B" digs a hole to the left of the on-screen hero, while button "A" does the same to the right. Of course, the Lode Runner must move carefully, because falling into a pit costs a life.

Unlike most other climbing and jumping games, the height of a fall is immaterial. The Lode Runner gently floats down to a safe landing regardless of how far it is to the ground. Unfortunately, the guards have the same ability, so the action moves up and down the screen a lot.

Lode Runner offers 50 different playfields, each a worthy challenge to mind as well as muscle. And when the player has solved all of them, there's a construction module to create an infinite array of customized ones. Few video games equal Lode Runner for replayability. This cartridge is as fresh two months after purchase as it was the first time it popped into the slot.

The Nintendo edition of Lode Runner is visually superior to the original computer game. The drawings of the hero and the guards are more detailed than in the computer version, and the major features of the horizontally scrolling playfield are larger and easier to see.

Even though there's no shooting in Lode Runner, the game does not lack excitement. Narrow escapes and tight squeezes abound, but quick thinking is just as crucial to success as fast reflexes. Few cartridges offer a better blend of action and strategy than Lode Runner. It's a prize worth capturing for any video gamer's library of games.



Dig Dug
INTV Corp.
3541 "B" Lomita Blvd.
Torrance, CA 90505
(213) 539-0100
Intellivision; $19.95

by Arnie Katz

If maze-chase games like **Pac-Man** have one drawback, it is predictability. Though some arcaders enjoy memorizing dozens of playfield patterns, many others grow bored with the same old maze. **Dig Dug** disrupts patterns by permitting the character to excavate underground tunnels wherever desired. This makes each round of play quite different from the ones which preceed and follow it.

Dig Dug, the player's character, burrows down into the ground from the surface, located at the top of the playfield, to start the game. After that, the home arcade directs the merry miner with the control pad as he creates tunnels in the multi-colored earth.

Although the gamer earns points for digging, the real prizes are the monsters Pookah and Fygar. The latter are more dangerous, since they can breathe deadly fire to the left or right. A monster can kill Dig Dug with a single touch, but the hero has an air gun, energized by hitting one of the action buttons, which can stun or even explode a monster.

Rocks buried in the hard-packed earth offer another way to dispatch a pesky monster. Dig Dug can clear away the ground beneath a rock and cause it to drop on any unlucky creature which happens to be passing underneath at the wrong time. It takes a few tries to get the timing right on this maneuver, but it offers a low-risk way of eliminating a foe once properly mastered.

The video gamer starts with three "lives." The player earns an extra one when the score reaches 10,000 points and another when it hits 40,000 points. Thereafter, the supply of Dig Dugs increases by one every 40,000 points.

A little helmet in the lower left corner of the playfield symbolizes each tunneler currently held in reserve.

Dig Dug is best described as relaxing rather than exciting. Though there is some time-pressure, players are generally able to take a moment to plot strategy before Pookah and Fygar come within range. The play-mechanic, which requires neither blinding speed or pinpoint accuracy, makes Dig Dug a good choice for family gaming sessions. Intellivision owners should try this one; they'll really dig it.

Galaga
Atari Corp.
1196 Borregas Ave.
Sunnyvale, CA 94086
(408) 745-2000
Atari 7800; $19.95

by Joyce Worley

Midway Manufacturing raked in so many coins on **Galaxian**, the classic shoot-'em-up earned a sequel, **Galaga**. Like the game that preceded it, Galaga became a mega-hit in the play-for-pay palaces. Now this super blastathon has come home, and it's a great addition to every joystick jockey's collection. Galaga for the Atari 7800 game system has all the action thrills that made this one of the best-selling sequel games in history.

The gamer controls a ship which moves horizontally across the screen, facing the ranks of oncoming invaders. Aliens appear from above and from the sides of the screen, then flit around the sky like pesky mosquitos. After an airshow of fancy flying, they fall into formation, a sort of flying wedge, with less valued alien ships in the forefront, and the flagships bringing up the rear.

Moving the control ship back and forth across the screen lines up the gamer's shots at the oncoming aliens. But these creatures from beyond learned from their earlier earthly encounters; they don't just sit like pidgeons waiting to be plunked off. Instead, Galaga's ships are in almost constant motion, darting around the screen like gnats. They peel off from the formation to make head-on attacks on the player's position, then reappear at the top of the screen. Unfortunately, the aliens twist and turn like living things, making them devilishly hard to hit.

Periodically, a flagship swoops down to send out a tractor beam. If the player's ship gets caught in its focus, it's captured. The ship actually flies with the alien, as if they were yoked together. If he has another command ship in his arsenal, the player can regain his ship by shooting the alien when it is attacking. Shooting it at any other time destroys the hostage vessel. If this maneuver succeeds, there's a great reward: now the player controls two ships, which move in tandem, doubling the firepower available to defeat the invaders.

The charm -- and the difficulty -- in Galaga comes from the twisting, turning, writhing patterns that the alien ships form in their attacks. The best strategy for racking up high scores is to anticipate the movement patterns these colorful little space cruisers take. Ships loop across the screen, sometimes doubling back on their own path, and the gamer must master the technique of aiming at where they're going, instead of where they are the moment the missile is launched. Flagships have to be hit twice to destroy them, but there's a payoff. The entire fleet stops firing for a few seconds to mourn their fallen leader; this gives the gamer a chance to blast away before the retaliation begins anew.

Scoring is tied to the difficulty of the shot. Hence, shooting an alien while it is flying in formation earns only half the amount of points. Three game modes, novice, advanced or expert, tailor the game to suit.

The excellent documentation that accompanies the game adds to the fun. In only two and half pages, it presents the background story, complete instructions and a full explanation of scoring, plus some strategy hints that actually work.

Galaga for the home lives up to its coin-op antecedents. It's a high-action, high-skill contest that will keep video gamers hitting the replay switch over and over again.

Pro Wrestling
SEGA c/o TONKA CORP.
6000 Clearwater Drive
Minnetonka, MN 55343
(800) 328-3628
Sega Master System' $30

by Bill Kunkel

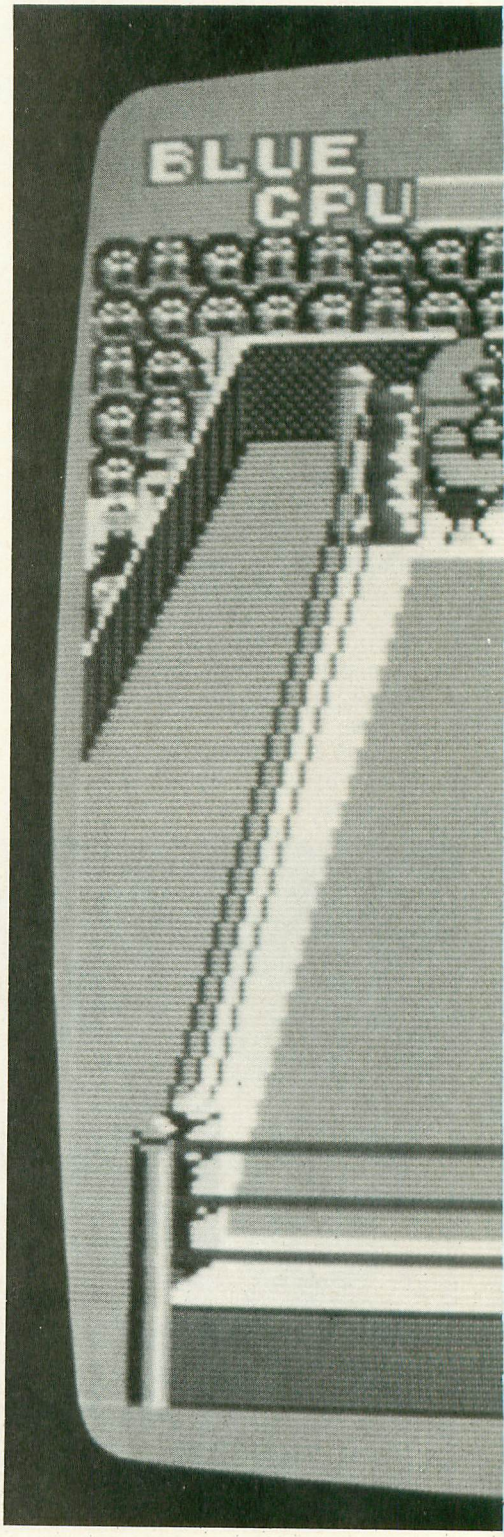
Sega's version of **Pro Wrestling** is a dreary rehash of Data East's mediocre wrestling coin-ops. The player chooses from among four tag teams (or, in non-team competition, eight single wrestlers): the Road Warrior-like Mad Soldiers; the Samoan-style Orient Express; the Great Maskmen; and the baby-faced Crush Brothers. Each team has eight custom maneuvers (each wrestler has four), including a German suplex, dropkick, hangman-style neckbreaker, bodyslam and clothesline.

All wrestlers can punch and kick and move freely about the ring, but the specialty moves, or "techniques" as Sega calls them, are largely situational. Some maneuvers, for example, only work outside the ring (one of the heel team techniques lets them grab a folding chair and smack a foe over the skull), while others can only be executed after hurling one's opponent into the ropes.

The control system is a little complicated, but it's the terrible graphics that sabotage this game. The on-screen wrestlers appear to be no more than three feet tall, though the ring is designed in proportion to "normal"-sized wrestlers. This lack of visually articulated bodies makes it almost impossible to gauge two grapplers' relative positions, much less determine what they are actually *doing* to one another.

The animation is so choppy and inadequate, the designers deemed it necessary for each hold's name to appear prominently on-screen as it is being executed.

There are some pointless frills, including a brief ring entrance with theme music, for each team or wrestler. It would have been far more fruitful for that time and memory to have been devoted to improving the game itself.



The documentation is occasionally confusing in its use of European/Japanese technology, which will be totally "Greek" to most users (what, for example, is a "senton from corner post"?).

Players must choose among three competing "Leagues" for singles or tag-team competition. These leagues differ in the number of "rounds" required in order to win. The concept of "rounds" is all-but-alien to American mat fans, however, and players will wonder why a pin isn't enough to win.

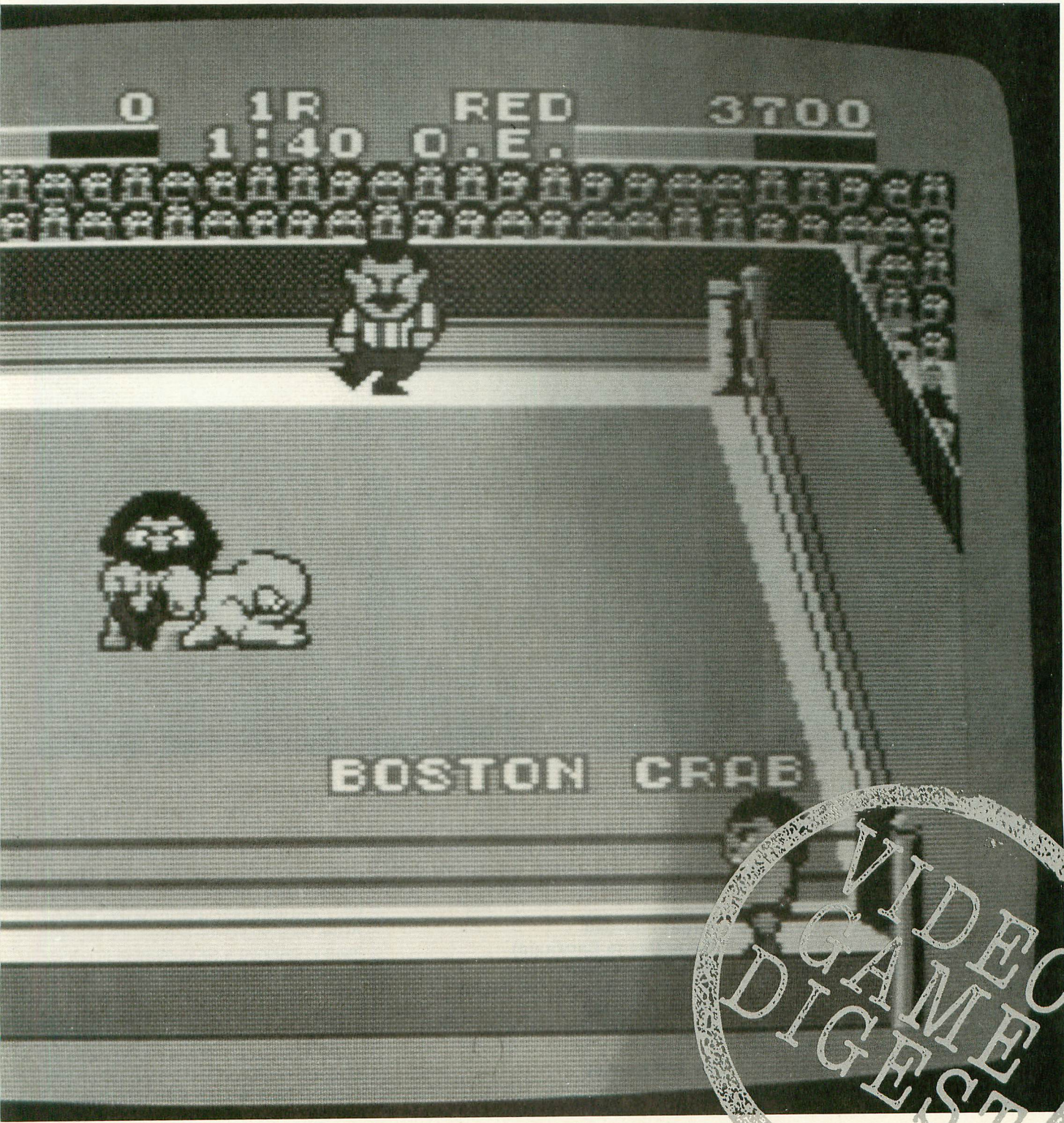
In singles competition, different opponents come with different referees. Since the referee should have no impact on the match, however, this looks like yet another example of energy expended in the wrong direction.

All in all, Sega's Pro Wrestling is a loser on a count-out—before it even gets to the ring!

Kung-Fu Master
Activision
P.O. Box 7287
Mountain View, CA 94039
Atari 2600; $13.95

by Bill Kunkel

Stop me if you've heard this one: A beautiful princess has been kidnapped and taken to the castle of the Evil Wizard. The fortress is a vast deathtrap, filled with all manner of menace, and the only warrior with even a prayer of res-



cuing her is you: the "Kung-Fu Master."

The Wizard's stronghold is no cracker box; it consists of five levels, each one crawling with the malevolent mage's misanthropic minions. You've got your basic, Dacoit-like henchmen, easily dispatched with a single punch or kick; knife-throwers are deadlier and more durable, as are the acrobatic martial-arts midgets; dragons, snakes, killer moths, and, at the end of each level, one of the nearly-indestructable Five Biggies.

The player's surrogate fighter and his various opponents appear on-screen against a horizontally-scrolling background which occupies approximately one-half of the screen. This combat area is rendered in simple lines which attempt to suggest an Oriental atmosphere.

Combat consists of left/right movement jumps, squats, high and low kicks, high and low punches and breaking a hold (achieved by jiggling the joystick from side to side).

There are also a pair of energy bars at the base of the screen. The first bar monitors the player's energy level while the second gauges the enemies' strength reserves. When the bar runs out, the corresponding character dies.

Kung-Fu Master is played against a time limit. An on-screen timer gives a warning signal when it reaches 200; at zero the player loses a life.

Kung-Fu Master is limited by the lack of possible combat movements. Users familiar with arcade and computer martial-arts games may be disappointed by the lack of sweep kicks, flips, spin kicks, blocks, etc., but the game should provide first-rate action.

Great Football
Sega (Mega Cartridge)
573 Forbes Blvd.
S. San Francisco, CA 94080
(415) 742-9300
Sega Master System; $30

by Bill Kunkel

Sega's **Great Football** for the Sega Master System is a beautiful-looking action gridiron simulation with the emphasis on running, passing and kicking.

In the one-player version, the game is *exclusively* offense. The player begins by selecting a "division" and team. Though they are dubbed "AFC" and "NFC," these are *not* NFL divisions; the teams have names like "Spartans," "Dukes" and "Boomers." Moreover, the documentation makes no distinction between "AFC" and "NFC," or even among teams, so these selections appear to be totally arbitrary.

The game begins with the computer-controlled team kicking off to the user's squad. At the beginning of the contest, the opposing team is assigned a set number of points (example: 35), which the player's team *must surpass*. The user's team remains eternally on offense, with only the clock and some rather indifferent defenders between him and victory.

On rushing plays, unless the defenders get lucky and bury your RB in the backfield, a minimum five yards is a lock. Of course, there isn't enough time to rush all game to score enough

points to pass the opposition. Passing plays aren't much harder; interceptions are rare and pass rushing even rarer. The computer plays a kind of ultimate "Pre-vent" defense, happily surrendering five, ten, 15 or even 20 yards in exchange for a couple of ticks off the game clock.

Great Football is a more satisfactory, realistic contest when played in two-player format. Humans invariably present a greater challenge on defense, and the game is much less distorted as a result.

The visuals get mixed reviews. The field is an eye-popper; lush green with a mammoth full-color eagle painted on the middle of the 50 yard line. The players, however, flicker outrageously—even when they're not moving!—and periodically split into top and bottom halves. This, combined with the all-offense orientation, creates the impression of football in the Twilight Zone.

Great Football is not exactly great, but offense junkies and players who can find opponents should get off on it.

Q & A by The Game Doctor

I tell you, it's absolutely amazing. Yours truly, the Original Game Doctor, has only to dust off the old shingle and patients are all over me like plastic on a joystick.

Of course, it's been a while since I was involved in active practice. Aside from the occasional cocktail party guest with a cranky RF modulator, the Dr. Gillespie of electronic gaming has spent most of his time messing with golf simulations and perusing snapshots of former Game Nurses.

Fortunately, game diagnosis is like riding a bicycle to a man of my vast experience, so let's jump right in with our first question:

Q: Which company, if any, now owns the right to translate Exidy, Williams, and Bally/Midway arcade games to videogame? In other words, can we hope to see such classics as **Pac Man, Joust, Tapper**, etc. available for the Nintendo system?

Dennis Sellers — Nashville, TN

A: These days, no one publisher buys exclusive rights to home versions of any arcade game. Once upon a time, companies like Atari would buy "exclusive home rights" to smash hits like Pac Man. The upshot of this was that only Atari system owners could ever hope to play these games.

Thankfully, those days are over. Today, licensing deals are cut on a system-by-system basis with a variety of publishers often buying individual system rights to the same game. We have seen computer games, like **Choplifter** (Broderbund), appearing under separate license on both the Sega and Atari 7800 videogame systems.

When it comes to home versions of arcade games, however, it's a different story. The videogame system manufacturers themselves have strong roots in the arcade business, as well as access to veritable libraries of past and present coin-ops. So if Nintendo wants to publish a home version of an arcade game, it's likely to be a Nintento arcade game and not an Atari or even a Williams game.

Sega, for example, is unlikely to publish a translation of the Atari coin-op, **Pole Position**. When Sega wanted a driving game for its videogame system, it sensibly turned to its own arcade hit, **Out Run**.

One of the big differences between the current videogame boom and the peak sales period of the early 80s is that the system manufacturers exert much more control over the output of third-party publishers. So not only is Sega unlikely to publish Pole Position, it is equally doubtful that it would encourage third-party publishers in such a venture.

The rights to the arcade classics you ask about are still available, I'm sure, but I know of no current plans to publish them for the NES. If Nintendo perceives a real demand for these games, and has nothing similar in its own coin-op inventory, however, you might someday see them.

Q: I want to know why software companies can easily take 2-meg arcade games and squeeze them into a 16K cartridge, but MicroProse can't squeeze a 64K program [**Gunship**, which MicroProse is not translating for Atari 8-bit systems because so many of them have insufficient memory] into 48K?

Louis J. Ferro — New Jersey

A: That's a good question, Louis.

For one thing, those "2-meg" arcade games you refer to are almost exclusively action games. They have a minimum of game logic, and almost all that lavish memory is devoted to sound and graphics. It is very easy to scale down sound and graphics.

Look, for example, at the many fairly acceptable Atari 2600 versions of arcade hits like Pole Position, **Ms. Pac Man** and **Joust**: They don't look exactly like the originals, but they play okay and the graphics are recognizable. Then take a game like **Zork** (Infocom), which has no sound or graphics and can be played on virtually anything south of a touch-tone phone, and just try to put it on even the top-line videogame systems!

Having said that, I'm sure it's within MicroProse's power to produce an acceptable 48K version of Gunship for the 8-bit Atari system; it just doesn't make any economic sense.

When a publisher produces a "rollover" or new translation of an existing game for a system with secondary sales potential (such as the Atari 8-bit systems), it has got to be a fairly straightforward process. That program must be transportable to the new system, with a minimum of reworking, and extensive code-crunching does not qualify under that criterion.

MicroProse obviously feels that potential sales do not justify the time and effort which would be required in order to crunch that code down to 48K.

Q: What's wrong with Sega's distribution? I live in Willoughby, Ohio, just outside Cleveland, and the Toys R Us and video store I go to gets me stuff from Comtron, Sega's distributor. Forty miles east of here is a Hills store, and they've had **Shooting Gallery** since May, but Toys R Us still doesn't have it. TRU has **Great Soccer**, however, and Hills doesn't. On September 3, the video store got me 3-D glasses and **3-D Missile Defense, Quartet, Great Vollyball, Great Football** and **Gangster Town**, which no one else has.

From what I've seen, Nintendo games come out everywhere at the same time; doesn't Sega/Tonka know you can't sell product if it isn't available?

Michael Gunn — Willoughby, OH

A: Oh, they know it, Michael, they know it. But remember, Sega is the new kid on the block, while Nintendo has had several years to establish a smooth-running distribution chain. NES has been able to build up its software line slowly, one and two titles at a time.

Sega only really went national in the last year. Game distribution is still uneven, but it is bound to improve quickly. The recent agreement with Tonka should bring the company's products to parity very quickly, as Sega equipment and supplies begin to be distributed through the Tonka marketing channels. Meanwhile, distribution is a little scattershot, and loyal Segaphiles like yourself will have to range a little farther and a little wider in order to complete your collection.

That's all for this issue, gang! Send your questions to "The Game Doctor" c/o ANALOG Computing.

End Jet

by Arthur Leyenberger

I have just returned from a week-long business trip in Columbus, Ohio. On the trip, I did something that I had not done before—I carried along a Zenith MS-DOS portable laptop computer. This was not the first time I had taken a computer along with me when I traveled, but it was my first trip with the Zenith Z-181 portable. Computing on the go is an excellent way to maximize your productivity. And surprisingly, almost any laptop computer can be interfaced with your Atari machine—either ST or 8-bit—when you return home. I'll give you all of the exciting interfacing details later in the column.

The Good, the Bad and the Ugly

The Zenith Z-181 is basically a good computer, as portables, and especially laptops, go. It has one of the nicest, most readable screens of the genre. Using a Supertwist LCD screen with backlighting, the 25-line by 80-column screen is readable in any lighting condition. There are two controls for the screen—contrast and brightness—and together they provide you with all the screen controls you need. The white on blue or blue on white screen is almost eerie to see and requires little time to become used to.

The machine has two 3½-inch floppy disk drives, each with the ability to store 720K bytes of programs and data. These disks are exactly the same as those used by the Atari ST. That's twice the amount that can be stored on a standard 5¼-inch MS-DOS diskette. Other

GEAR UP YOUR DISK DRIVE FOR BIG SAVINGS!

TO SUBSCRIBE TO ANALOG
TURN TO PAGE 2
TO SUBSCRIBE TO ST-LOG
TURN TO PAGE 99

SAVE MONEY AND TIME BY HAVING YOUR FAVORITE COMPUTER MAGAZINE MAILED DIRECTLY TO YOUR DOOR AT A FRACTION OF THE COVER PRICE!

MegaByte

Computers and Electronics

Call TOLL FREE 1-800-255-5786 ORDERS
For any inquiries 1(713)338-2231 ONLY

109 W. Bay Area
Webster, Texas —
77598

Please Call For Low Prices on Many Other Items!!

Printers - Modems - Computers - Software

GAMES	Applications
DARK CASTLE 27.97	Cad-3D 2.02 59.87
DUNGEON MASTER (IN STOCK!) 27.97	Cyber Control 43.97
F-15 STRIKE EAGLE 24.87	Cyber Paint 47.97
FLIGHT SIM II 33.87	Data Manager
GAUNTLET 23.97	Swift Calc
GONE FISHIN 27.97	Word Writer 52.97
MOEBIUS 38.97	PC-Ditto 3.01 74.95
OIDS 23.97	Publishing Partner 62.97
Police Quest 29.97	SuperCharged EasyDraw 94.97
President Elect 28.97	
RoadWar Europa 25.87	*Kuma Transputers*
Ultima IV 38.97	*Now Only $995 and up!*
Wargame Const. Set 33.87	

GRADEPLUS
(A complete teachers gradebook system)
$39.95

TURBO ST
ENHANCEMENT
SUPERCHARGE
Your ST to
"14" MHZ!!!
$199.95

CUSTOMIZED
Hard Disks
$499 and up...

EXTERNAL Floppies
$199.95 !

ORDERING INFORMATION:

DEALER INQUIRIES WELCOME!

Prices are subject to change without notice. All goods include FULL factory warranty. Texas residents please add 6.38% sales tax. Returns only if product is defective. Prices reflect a cash discount, please add 3.00 for Visa or MasterCard. Blue label shipping is available upon request.

CIRCLE #106 ON READER SERVICE CARD.

ATARI®

COMPUTER SOFTWARE SERVICES

P.O. BOX 17660, ROCHESTER, N.Y. 14617
PHONE (716) 467-9326

DEALER/DISTRIBUTOR/USER GROUP Discounts available call for info.
Phone Orders - MASTER CARD, VISA
Mail - Money Orders, Check

ATARI®

THE $69.95 "SUPER ARCHIVER"!®

NOW "BEST BUY" RATED!

(for ATARI 1050 drives)

The new **SUPER ARCHIVER,** obsoletes **all copying devices** currently available for the **ATARI 1050!** It **eliminates** the need for Patches, **PDB files,** Computer Hardware, etc. Copies are **exact duplicates** of originals and **will run on any drive;** without exaggeration, the **SUPER ARCHIVER** is the **most powerful PROGRAMMING/COPYING** device available for the 1050! Installation consists of a plug-in chip and 6 **simple** solder connections. Softwares included. Features are:

- **TRUE DOUBLE DENSITY**
- **ULTRA-SPEED** read/write
- **FULLY AUTOMATIC COPYING**
- **SUPPORTS EXTRA MEMORY**
- **SCREEN DUMP** to printer
- **TOGGLE HEX/DEC DISPLAY**
- **SECTOR** or **TRACK TRACING**
- **AUTOMATIC DIAGNOSTICS**
- **DISPLAYS HIDDEN PROTECTION**
- **ADJUSTABLE/CUSTOM SKEWING**
- **AUTOMATIC SPEED COMPENSATION**
- **AUTOMATIC/PROGRAMMABLE PHANTOM SECTOR MAKER**
- **ARCHIVER/HAPPY ARCHIVER COMPATIBLE**
- **BUILT-IN EDITOR**-reads, writes, displays upto 35 sectors/track (short)
- **BUILT-IN CUSTOM FORMATTER** - upto 40 sectors/track
- **BUILT-IN DISASSEMBLER**
- **BUILT-IN MAPPER** - upto 42 sectors/track
- **DISPLAYS/COPIES** Double Density **HEADERS**
- **AUTOMATIC FORMAT LENGTH CORRECTION**
- **SIMPLE INSTALLATION**

The **SUPER ARCHIVER** is so **POWERFUL** that the only programs we know of that can't be copied are the newer **ELECTRONIC ARTS** and **SYNFILE/SYNCALC** (34 **FULL** sectors/track). If you want it **ALL** . . . . buy the **"BIT—WRITER"!** also. . . . then you'll **be able to copy** even these programs!

THE SUPER ARCHIVER $79.95 "BIT-WRITER"! $79.95

The Super Archiver **"BIT-WRITER"!** is capable of **duplicating even** the "uncopyable" **EA** and **SYN** series which employ 34 **FULL** sectors/track. **"BIT-WRITER"!** is capable of reproducing these and **FUTURE** protection schemes of non physically damaged disks. **PLUG-IN** circuit board and 4 **simple** solder connections. The **SUPER ARCHIVER** with **"BIT-WRITER"!** is the ultimate **PROGRAMMING/COPYING** device for Atari 1050's **EXACT DUPLICATES** of originals are made! Copies run on **ANY** drive.

$69.95 "ULTRA SPEED PLUS!" $69.95

Imagine a universal XL/XE Operating System so easy to use that anyone can operate it instantly, yet so versatile and powerful that every Hacker, Programer and Ramdisk owner will wonder how they ever got along without it! Ultra Speed Plus puts unbelievable speed and convenience at your fingertips. Use **ANY DOS** to place an **ULTRA SPEED** format on your disks, boot any drive (1-9) upon power-up, format your **RAMDISK** in Double Density, activate a built-in **400/800 OS** for software compatibility, plus dozens of other features to numerous to mention! Below are just a **FEW** features you'll find in the amazing **OS:**

- **ULTRA** Speed S10 for most modified drives
- **ULTRA** Speed is toggleable
- Boot directly from **RAMDISK**
- Special timer circuits not required for 1 or 2 Meg upgrades
- Background colors adjustable
- Reverse use of **OPTION** key
- Cold-start without memory loss
- Built in floppy disk configuration editor (1-9)
- Built in **RAMDISK** configuration editor (1-9)
- **RAMDISK** exactly duplicates floppy drive so sector copying and sector editing are now possible
- Built in **MINI** Sector Copier
- Toggle **SCREEN OFF** for up to 40% increase of processing speed
- Toggle internal **BASIC**
- Rom resident disk loader program **(MACH 10** menu)
- **DOUBLE DENSITY RAMDISK** capable
- Entire **MEMORY** test that pinpoints defective **RAM** chip
- Boot any drive (1-9) upon power-up or cold-start
- Supports memory upgrades up to **TWO MEGABYTES**
- **THREE** Operating Systems in one **(XL/XE, 400/800, ULTRA SPEED PLUS)**

$29.95 RAMDISK "WRITE-PROTECTOR!" $29.95

Hackers, Programers, or BBS users. . . if you own a **RAMDISK** (memory upgrades for your XL or XE computer), think about this: Every disk drive ever manufactured has **WRITE-PROTECT** capabilities. . . except your **RAMDISK.** Without it, your valuable stored data/program lie naked, awaiting that one mistake that will wipe out hours, maybe weeks of precious programming efforts. End the fears of accidental formatting or overwriting by installing our universal **RAMDISK** "Write-Protector!" Works on all memory upgrades up to 2 megabytes. Simple installation. Only $29.95.

$29.95 "XF551 ENHANCER!" $29.95

The XF551 Atari drive is a fine product with one major flaw. . . it writes to side **TWO** of your floppy disks **BACKWARDS.** This causes read/write incompatibility problems with all other single sided drives made for Atari such as Indus, Trak, Rana, Percom, Astra, Atari 1050, Atari 810, etc. Add the **XF551 ENHANCER** to the new XF551 drive and your problems are over! This device will restore 100% compatibility between all drives while retaining all of the original design qualities of Ataris' super new drive. The **XF551 ENHANCER** is a **MUST** for all XF551 owners. Installation is simple. Only $29.95.

CIRCLE #107 ON READER SERVICE CARD.

laptop computers also use the 3½-inch disk, and as ST users already know, the disks are more rugged, compact and easier to handle than the 5¼- inch disks that the 8-bit Atari and most other computers use.

The Zenith Z-181 uses a CMOS version of the 8088 microprocessor running at 4.77 megahertz, making the computer's processing speed typical of MS-DOS PCs and clones. However, by IBM AT and AT clone (machines that use the 80286 processor) standards, this machine seems to crawl. All else considered though, the Zenith machine ranks with the best. The disappointment comes, however, with its usability and (whew!) weight. Usability is primarily hampered by the pivoting screen.

The screen is a good 12 inches tall, enough to cover the entire machine when it's closed. When open, it's almost impossible to use the computer on an airplane when traveling in coach or business class. The tray table has barely enough room to hold the computer itself, and there is no way the screen can be pivoted up to the normal viewing position. If the person in the seat in front of you decides to recline his seat, you can forget about computing altogether.

I had the opportunity to travel first class on the return trip (a $15 Continental upgrade, mind you), and the machine was quite usable. Since the first-class seats have their tray tables attached to the occupants' seats, there is plenty of room for the computer, and the screen can be pivoted up to the proper position.

If you can afford to travel first class all of the time, this computer still may not be for you. Why? Because it is heavy. How heavy is heavy? The advertisements for the Zenith Z-181 declare it as an 11-pound laptop computer. That weight must have been determined on the moon, with its ⅙ of the Earth's gravity level. Here on Terra, the machine really weighs 16 pounds according to the Continental baggage scale at Newark International Airport. That weight is the actual carrying weight with the case, battery pack, charger and one box of ten floppy disks.

It doesn't take long for 16 pounds to get really heavy. Even when carrying the whole kit and caboodle by the shoulder strap, it's still a substantial weight. If computing on an airplane with a Z-81 is important to you, then you had better be well-heeled and also regularly work out with weights. If you don't care to compute in the air and don't expect to be toting the machine around very much, it's a nice computer with a street price of approximately $1,800. But there's a better alternative for computing on the go and then linking up with your Atari at home.

The Ultimate Atari Peripheral

I use a laptop computer mostly when I travel and mostly on an airplane. You may recall reading several Consumer Electronics Show reports over the last couple years, written on the return plane ride from Las Vegas or Chicago. High-altitude computing makes boring and somewhat wasted air-travel time productive.

For the last several years, I have been using a Radio Shack Model 102 laptop computer and portable disk drive. This machine is truly one of the best kept secrets in all of computerdom. Transferring files to the Atari is painless too.

The Radio Shack Model 102 computer is an improved version of the original Model 100 computer. Both machines share almost identical features, except that the Model 102 weighs a mere three pounds rather than four, and stands one inch shorter than its predecessor, roughly 1½ inches tall. The width and depth of both machines is about the size of an 8½ by 11-inch sheet of paper held sideways, hence the nickname "notebook" computer.

The Model 102 has a flat LCD screen that is flush with its keyboard. The screen contains eight lines of 40 characters with large letters that are easy to read, assuming you have good ambient light. The contrast knob does help, though. The computer doesn't contain a built-in disk drive and can be outfit-

ted with a maximum of 32K bytes of Random Access Memory. Fortunately, there are several programs contained in Read Only Memory along with a useful, somewhat limited version of BASIC.

The internal applications consist of a text editor, telecommunications, address and scheduler programs. The text editor is simply that—a simple but useful program that allows text to be entered, minimally edited and saved in the battery-backed-up, nonvolatile memory as a file. The telecommunications program works with the built-in 300 baud modem. The other two applications are not worth discussing.

In addition to the built-in modem, the computer contains bar-code reader, serial, parallel, external disk drive and telephone (RJ-11) jacks for connecting to other devices. Four AA batteries are used to power the computer, usually lasting about 15-20 hours a set. The street price of a Radio Shack Model 102 with the maximum 32K memory is about $400.

The Radio Shack portable disk drive uses 3½-inch floppy disks and stores 180K bytes per disk. It also uses four AA batteries and weighs less than two pounds. The list price of the disk drive is $200. A complete Radio Shack Model 102 computer outfit consisting of computer, disk drive and power transformer is under five pounds.

Preparing to Beam Up

As far as I am concerned, both the Zenith and the Radio Shack computers have limited usefulness by themselves. They become truly useful when you interface them with your Atari computer back home. The mechanics of uploading files from the portable computer to the Atari (either 8-bit or ST) are essentially the same. You need a serial cable to connect the two machines together via their RS-232 ports and a modem program running on each machine. You also need one additional, very important item: a null modem adapter which is a connector that attaches to one end of the communications link to allow the two machines to talk to each other.

You can buy a null modem adapter for about $20 from Radio Shack, or if you want to construct your own, reverse the wires on pins 2 & 3, 5 & 8 and 6 & 20 on *one* end of your cable only. In addition, jumper pins 4 to 5 together on both ends of the cable. If you are not familiar with soldering techniques or are not sure how to take an RS-232 connector apart, cut and strip the wires, etc., then it would be best to have it done by someone who knows what they are doing, or simply buy the adapter which can be used on any serial cable.

Once the cable is correctly attached to both computers, the procedure is simple. The telecommunications program is run on both machines, at the same speed and with the same configuration, and the laptop becomes the sending computer, and the Atari becomes the receiving computer.

Beam Me Up

Connect the Model 102 computer to your Atari 850 interface (or other RS-232 connection) with a serial cable and null modem adapter or a modified RS-232 cable. The RS-232 jack on the Model 102 needs a male plug and the RS-232 jack on the Atari interface needs a male 9-pin connector. Next, turn on the Model 102 computer. Then boot up the Atari computer and run a telecommunications program. I like the communications program from *Homepak* called *Hometerm*. Although Batteries Included is no longer in business, Homepak should still be available in many stores. If not, any terminal program will work.

Set the terminal parameters on the Atari to 2400 baud, 8-bit word length, 1 stop bit, no parity, full Duplex, no emulation and XON/XOFF enabled. I have not been able to consistently transfer files at higher speeds without errors; so I always use 2400 baud. Use the "capture" command on the Atari to receive the transmission from the Radio Shack and then store it as a file.

You could use the internal modem on the Model 102 to transfer a file to the Atari, but at a maximum speed of 300 baud, it might take a while. Instead, you should not only bypass the internal modem but also the telecommunications program. Then, from within the Radio Shack's text editor, save a file to the "COM" port (the Model 102 serial port). To do this, press the F3 function key and type "COM:68N1E." This means save a file to the COM port, transfer it at 2400 baud, use an 8-bit word length, no parity transmission with 1 stop bit and XON/XOFF status enabled.

Once the command is given, the file will be transferred to the Atari and saved as ASCII text. Once saved as a file, just about any Atari 8-bit word processor can then be used to access the file and edit it as you wish.

The procedure is exactly the same for the Model 102 and Atari ST combination. The only difference is the particular communications program that is used on the ST to interact with the Radio Shack computer.

Wrapping Up

Comparing the prices of the Zenith portable to the Radio Shack portable, you get similar numbers if you calculate it by the pound, $106 vs. $120. However, the Radio Shack computer is cheaper, more value for the money and weighs less than a ⅓ of the Zenith. Enough advantages to, ah, "outweigh" the competition of the Zenith and other portable laptops currently available. Choosing the Model 102 computer to compute on the go is both productive and rewarding. After one or two tries, you'll master the art of transferring files from one machine to another. My only advice is to be sure to have plenty of extra AA batteries on hand for the Model 102 and portable disk drive. They seem to have an unending appetite.

Arthur Leyenberger is a human factors psychologist and freelance writer living in New Jersey. He has written over 100 articles about computers in the last four years and continues to be an Atari enthusiast. When not computing, he enjoys playing with robotic toys.

ST-Notes

by Frank Cohen

Clout: A word describing the power that most major computer hardware and software manufacturers rely on when they attempt to move or change their markets. Atari has sought for it, but on most occasions failed. Apple recently began to show it, but is still having problems. IBM always had it, but is finding it needs more. Word Perfect Corp. has it and uses it effectively. Just look at their recent announcement that they are considering pulling out of the Atari ST software market.

Mr. Acerson, Director of Corporate Communications, Word Perfect Corp. (WPC), has announced that they are dismayed at the availability of their word processor, Word Perfect, on Pirate Bulletin Board Systems (BBS). WPC has found complete versions of Word Perfect on three BBSs just months after the release of its first ST product. WPC won't be hurt by the loss of a couple of sales of its powerful word processor, however, WPC has openly said that it is in the business of selling software and not of hunting after software pirates.

WPC's announcement has reignited the controversy over software piracy. The Atari XL computer was plagued with numerous problems, largest of which was the issue of piracy. Software publishers found that larger and larger numbers of illegal copies of their products were making their way up the gravevine. Piracy has long been a problem with most computers, however, Atari has held the reputation.

WPC has said it has not found similar piracy problems with the Amiga or IBM user community. Odd. Of all the people I know with IBM PCS just about every one of them has a pirate copy of Lotus 123, dBase III Plus, Word Perfect and Microsoft Word. The honest ones later bought legitimate copies. But most don't.

Software piracy exists on all machines. IBM, Macintosh, Amiga and Atari all have problems with software piracy. So why has WPC chosen not to "go with the flow?" The answer to this question goes back a couple of years to Atari Corp.'s desperate search for major software vendors (Lotus, Ashton Tate, Microsoft, etc.) to support their new ST machine. WPC was the first major company to move their highly acclaimed word processor onto the ST. At a time when the highest price for a word processor was $79.95, Word Perfect was wheeled into the ST market on its six-disk set at a whopping $399.95. Word Perfect has since basked in the splendor of its own elegance and strength. Atari Corp. and all of its supporters now had some clout.

WPC is a very large company that is used to marketing software to a huge industry. Software markets for the IBM PC play games by determining who has the most clout. So, it takes little reasoning to see how an announcement like WPC "pulling out of the ST market" might shake things up with the pirate BBSs. Hopefully, it will reduce the amount of piracy that normally resides in the ST software market. If it does then the fate of the ST lies more in the hands of the ST users community than it does in the hands of pirates.

FTL Dungeon Master

Gaming on the ST has become really fun. The new line of game releases has included some technically superb graphics, game play and sound effects. FTL Games is providing a number of the better games. Their credits include Sundog, Oids and Dungeon Master.

In a recent BBS conference, Wayne Holder, president of FTL Games, said that sales of Dungeon Master (DM) were brisk. DM ($39.95 list) was first shown in 1986. The demonstrations showed a three-dimensional high-resolution

graphic dungeon that you had to move through. And move you did—the motion was animated, giving you a real sense of depth and complexity. DM's development crew originally worked with Pascal. Development eventually bogged down and FTL switched horses to the C language. According to Doug Bell, DM programmer, the C learning curve can be brutal. Eventually, they developed a games compiler which was used to complete the project.

Recently, FTL Games released Dungeon Master 1.1, which corrects some bugs and adds new features to this graphic adventure game. If you're wondering if you have the latest version, look to the upper right corner of the save game screen.

DM is typical of how complex it is to develop games for a machine as advanced as the ST. Mike Newton developed the dungeon layout, while Andy Jaros, created the graphics.

FTL has completed German and French versions of DM. The European marketing will be handled by Mirrorsoft, Ltd. You might recognize the name from Mirrorsoft's desktop publishing program Fleet Street Publisher.

Holder said that Tracy Hickman is writing a hint book for DM. Tracy is the creator of the Dragon Lance series for TSR. Holder expects the book to be out within the next few months. In the meantime, they do accept support phone calls at (619) 453-5711.

Holder is trying to release four to six games per year. However, translations of their games to other computers slow this product release schedule. So far this year they have released DM and OIDS.

OIDS ($34.95 list) is an arcade-quality game which has flavors of Asteroids, Lunar Lander, Choplifter and Gravitar, all popular arcade games. The graphics and game play are excellent. OIDS' neatest feature is the ability to construct your own playfields using a construction set. The game is filled with interesting characters and animation, so you should find hours of exciting game play. Dan Hewit, OIDS programmer, has incorporated many functions to customize the game to your level of play.

Soft Logik Publishing Partner 2

The 1986 release of Publishing Partner (PP) was heralded as the answer to the missing Atari desktop publishing system. At $149.95, PP was a bargain when comparing it to similar software packages on the Mac and IBM PC. The program let you import text and graphics and visually determine the layout of the printed page. PP was originally supposed to support Digital Research's GDOS operating system, but later they went to their own font/device driver system. Later, PP 1.1 was released to fix some bugs that had been found. At the same time a number of fonts and device drivers became available, which further established PP as 'the' desktop publishing system of the ST.

Soft Logik has now released Publishing Partner Professional. At a high retail price of $199.95, the new system sports auto text flow around graphic images, auto hyphenation, kerning, an UNDO command, special text effects like slant, twist and rotate, and more included fonts.

If you have used PP 1.1 to do serious layout work, you will probably remember how the program redraws the entire screen every time an object is moved, selected or changed in any way. Redrawing complex screens holding more than a few objects would slow down your creativity to a crawl. The new system fixes this limitation. Now only the affected objects on the screen are redrawn, making the program vastly quicker and easier to use. Objects may now also be grouped, making it easier to cut, copy and paste more than one object at a time.

Word processor files may be imported directly from Word Perfect, First Word, Regent Word II, and Word Writer files. Soft Logic seems to be confident enough to include Timework's Word Write compatibility, even though Timeworks is releasing their own desktop publishing system.

Upgrades for PP 1.1 owners are available directly from Soft Logik at a cost of $99. PP 1.1 will continue to be marketed at a lower price, so you will have the option to try the lower priced package first, then upgrade to PP Profession later.

Broderbund Changes Its Mind

The long-awaited U.S. release of Art Director and Film Director will have to wait a little longer. Art Director is a comprehensive drawing package that is filled with powerful graphics functions. Film Director takes Art Director graphics and creates animated presentations. Both were originally developed in Europe two years ago. Broderbund showed the programs at the Atarifests in 1987, but has decided now not to release the products.

Karateka, a very slick Kung Fu combat game, is set for a mid-1988 release. This one is a winner.

Companies Mentioned:

Word Perfect Corp., 1555 N. Technology Way, Orem, UT, 84057, (801)227-4288, Broderbund Software, 17 Paul Drive, San Rafael, CA, 94903, (415)479-1170, FTL Games, 6160 Losk Blvd., Suite C206, San Diego, CA, 92121, (619)453-5711, Soft Logik Corp., P.O. Box 290071, St. Louis, MO 63129, (314)894-8608

About the author: Frank Cohen has been developing Atari programs since his first commercial product, Clowns & Balloons. When Atari Corp. began marketing the 16 Bit St computer, he founded Regent Software. Frank developed Regent Base, an SOL 4GL database, and is currently involved with several other St related productivity and small business software packages. you may contact Frank directly on Delphi (REGENTWARE), Genie (FCOHEN) or Compuserve (72457, 3171).

BOOT UP TO BIG SAVINGS!

1 YEAR FOR ONLY $28

SAVE $14 OFF THE COVER PRICE

1 YEAR *WITH DISK* ONLY $105

SAVE TIME AND MONEY SUBSCRIBE TO ANALOG

SAVE $14 OFF THE COVER PRICE WITH THE CONVENIENCE OF HAVING ANALOG DELIVERED DIRECTLY TO YOUR DOOR *BEFORE* IT EVEN HITS THE NEWSSTANDS. GET THE MOST OUT OF YOUR COMPUTER.

SUBSCRIBE TO ANALOG TODAY

☐ 1 YEAR @ $28 — SAVE $14! MCFYY
FOREIGN — ADD $7 PER YEAR
☐ 1 YEAR WITH DISK @ $105 DCFYY
FOREIGN — ADD $15 PER YEAR

☐ PAYMENT ENCLOSED ☐ BILL ME
CHARGE MY: ☐ VISA ☐ MC #_____

EXPIRATION DATE SIGNATURE

MONEY BACK ON ALL UNUSED PORTIONS OF SUBSCRIPTIONS IF NOT SATISFIED.

NAME

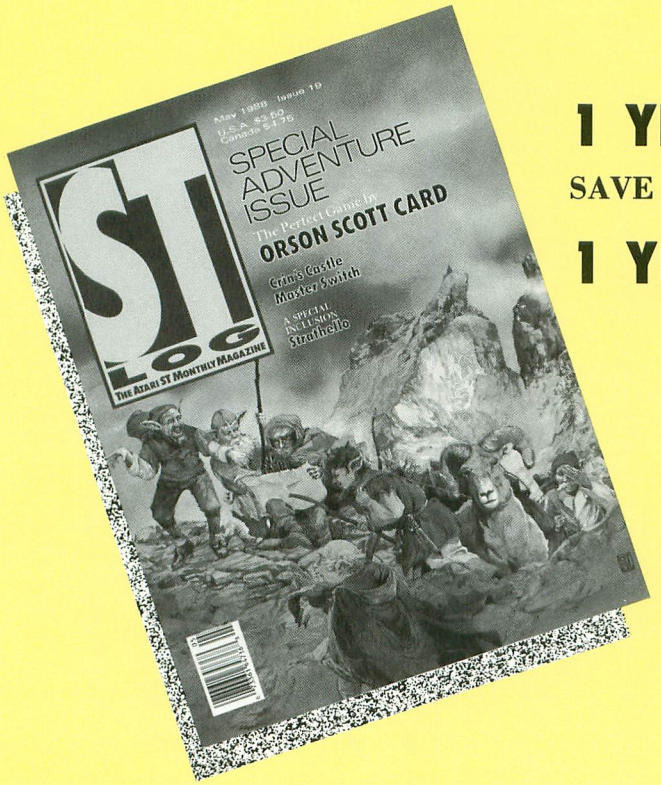
ADDRESS

CITY STATE ZIP

MAKE CHECK PAYABLE TO L.F.P., INC., P.O. Box 16927, N. Hollywood, CA 91615. Offer expires August 31, 1988. Your first issue will arrive in 6 TO 8 weeks.

WATCH FOR IT!

BOOT UP TO BIG SAVINGS!



1 YEAR FOR ONLY $28

SAVE $14 OFF THE COVER PRICE

1 YEAR *WITH DISK* ONLY $105

SAVE TIME AND MONEY SUBSCRIBE TO ST-LOG

SAVE $14 OFF THE COVER PRICE WITH THE CONVENIENCE OF HAVING ST-LOG DELIVERED DIRECTLY TO YOUR DOOR *BEFORE* IT EVEN HITS THE NEWSSTANDS! GET THE MOST OUT OF YOUR COMPUTER

SUBSCRIBE TO ST-LOG TODAY!

☐ 1 YEAR @ $28 — SAVE $14! MCFWW
FOREIGN — ADD $7 PER YEAR

☐ 1 YEAR WITH DISK @ $105 DCFWW
FOREIGN — ADD $15 PER YEAR

☐ PAYMENT ENCLOSED ☐ BILL ME

CHARGE MY: ☐ VISA ☐ MC #_____

_____

EXPIRATION DATE SIGNATURE

MONEY BACK ON ALL UNUSED PORTIONS OF SUBSCRIPTIONS IF NOT SATISFIED.

_____

NAME

_____

ADDRESS

_____

CITY STATE ZIP

MAKE CHECK PAYABLE TO L.F.P., INC., P.O. Box 16928, N. Hollywood, CA 91615. Offer expires August 31, 1988. Your first issue will arrive in 6 to 8 weeks.

WATCH FOR IT!

Refresh Your Memory

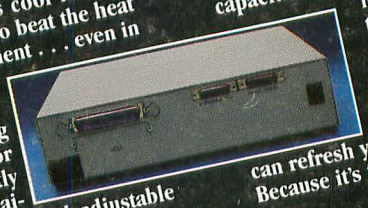
And Keep Your Cool.

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