U.S.A. $3.50
CANADA $4.75

ST-LOG

THE ATARI ST MONTHLY MAGAZINE

JULY 1987 / AUGUST 1987

ISSUE 16

FOUR-STAR SOFTWARE ISSUE:

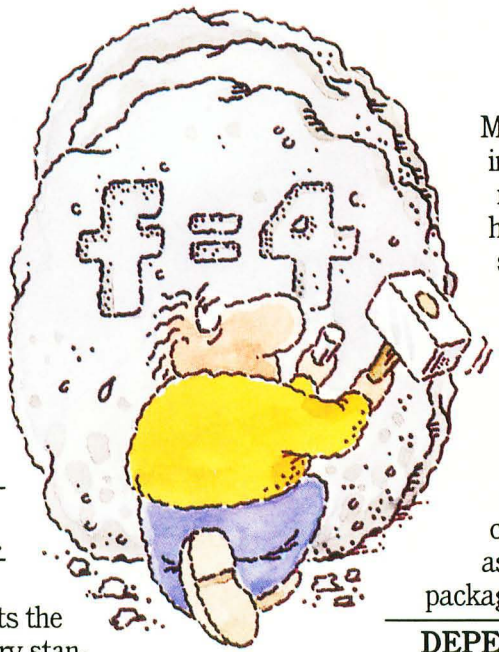
A comparison of five ST spreadsheets

Four-Star Software

Inside Electronic Arts

Desk Manager

MARK WILLIAMS C. AN ENLIGHTENING DEVELOPMENT FOR ATARI ST USERS.



If you've tried your hand at developing applications on the Atari ST, you know the problem. Programming tools aren't only hard to come by, they're hard to use. One might even say primitive. But now for some enlightening news: you can have all the power, portability and versatility of the C language from a leader in professional C programming tools, Mark Williams.

BRING YOUR PROGRAMMING UP TO SPEED.

The Mark Williams C compiler produces fast, dense code and supports the complete Kernighan & Ritchie industry standard C. You'll have access to GEM's AES and VDI libraries for programs using graphics, icons and the Atari mouse. And Mark Williams C lets you take advantage of the full 16 megabytes in Atari's 68000 microprocessor.

STREAMLINE DEVELOPMENT WITH POWER UTILITIES.

Mark Williams C is loaded with everything you'll need for professional development. Bring the power of the UNIX environment to your Atari ST with our Microshell Command Processor including pipes, I/O redirection and more. Edit your program with the highly acclaimed MicroEMACS full screen editor. Accelerate and simplify compiling with *make* which finds and recompiles only those modules affected by your changes. Then, when you're ready for debugging, call on our db Symbolic Debugger with single step, breakpoint and stack traceback functions. Over 40 commands, including a linker and assembler, provide a total development package for your Atari ST.

NEW VERSION 2.0 FEATURES

- ★ 50% faster compile time
- ★ Improved code: 20% faster dhrystone
- ★ Full access to AES/VDI libraries now with 200 new pages of documentation and dozens of examples
- ★ New integrated edit-compile cycle: editor automatically points to errors
- ★ Source and object code for RAM disk cuts compile time in half
- • Complete Kernighan & Ritchie C plus extensions
- • Microshell Command Processor, a powerful UNIX style shell
- • MicroEMACS Full Screen Editor with commented source code
- • Make Program Building Discipline
- • Complete symbolic debugger
- • Assembler, linker and archiver
- • Powerful Utilities Package: one-step compiling/linking with cc command, egrep, sort, diff and more
- • Over 600 pages of documentation including 120 sample C programs
- • Not copy protected

Sieve benchmark*:
Compile and link time in seconds.

Mark Williams C	46.0
Megamax	78.5

MARK WILLIAMS C FOR THE ATARI ST: $179.95
60 DAY MONEY BACK GUARANTEE

*Sieve benchmark from Byte, 1983, done on one double-sided floppy disk and included RAM disk.

DEPEND ON A NAME WITH A HISTORY OF PERFORMANCE.

Mark Williams C for the Atari ST is part of our growing line of C compilers. A line that includes the C compiler chosen by DEC, Intel, Wang and thousands of professional programmers. Now our Atari C compiler is earning its own reputation:

"Finally a great C compiler that exploits the power of the ST"–Sigmund Hartmann, President, Atari Software Group

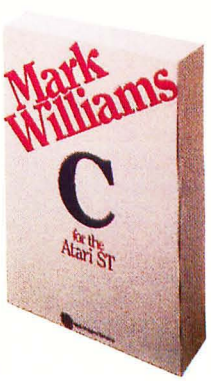
"The all-around best choice for serious software development on the ST."–Douglas Weir of *ANALOG COMPUTING*

GET WHAT YOUR ATARI ST HAS BEEN WAITING FOR.

Mark Williams C is just what your Atari ST was made for: powerful, professional programming. So now that you can have Mark Williams C for just $179.95, what are you waiting for?

Ask your Atari dealer about Mark Williams C or order today by calling **1-800-MWC-1700.***

*In Illinois call: 312-472-6659



Mark Williams Company

1430 West Wrightwood, Chicago, Illinois 60614

CIRCLE #101 ON READER SERVICE CARD

THE ATARI ST MONTHLY MAGAZINE

ST-LOG

ISSUE 16
JULY/AUGUST 1987

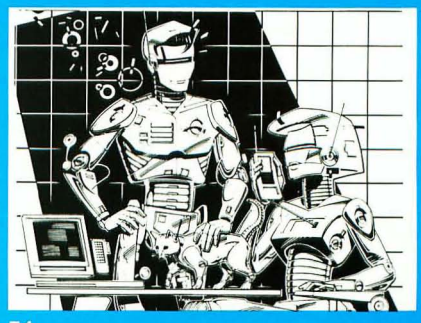
FEATURES

REVIEWS

COLUMNS

13

28



51

ST-Log (ISSN 0890-9601) is published monthly for $28.00 ($36.00 in Canada, $39.00 foreign) per year by ANALOG 400/800 Corp., 565 Main Street, Cherry Valley, MA 01611. Second class postage paid at Worcester, MA and additional mailing offices. POSTMASTER: Send address changes to **ST-Log**, P.O. Box 625, Holmes, PA 19043.

Make the connection!

ANALOG Computing on Delphi puts you on-line with the world.

Delphi, an on-line, full-service information network, offers news and sports from the Associated Press, weather reports, movie reviews, shopping services, travel information, and more.

ANALOG Computing, the #1 magazine for Atari owners, brings you the Atari Users' Group on Delphi. We offer a message forum and an extensive database for up- or downloading—all from as little as 10 cents per minute from most U.S. cities, with no additional telephone charges and no extra charge for 1200 or 2400 bps. We'll use the group's conference feature for electronic meetings with well known Atarians and, of course, **ANALOG** staff. Bring on your toughest questions!

Special rates

Subscribers to **ANALOG Computing** or **ST-Log** may join without charge, and will receive a free lifetime Delphi membership, a Delphi Command Card and $5.00 of line-time credit applicable to their account. If you purchase the *Delphi Handbook*—the highly detailed manual on using the whole Delphi system—for $29.95, you will get an additional $20.00 of line-time credit. And you can subscribe to either **ANALOG Computing** or **ST-Log** directly, while on-line, to be eligible for these bonuses.

How to connect

First, select a data communications network: Telenet or Tymnet (in the U.S.), or DataPac (in Canada). In the Boston area, dial Delphi direct (617-576-0862). To determine your local Telenet number, dial 800-TELENET or 703-689-5700 (in Alaska, 907-264-7391). To obtain a Tymnet number, call 800-336-0149. If you have difficulty, call Delphi at 800-544-4005 (in Massachusetts, 617-491-3393). Current subscribers to **ANALOG Computing** or **ST-Log** should type *JOINATARI* when asked for user name. When asked for a password, type *ANALOG*. Those who wish to subscribe to either magazine on-line should, instead, type *SUBSCRIBE* at the password prompt. Once on Delphi, you'll find our group on the "Groups & Clubs" menu. To get there, just type *GR ATARI* from the main-menu prompt.

THE #1 MAGAZINE FOR ATARI COMPUTER OWNERS

ANALOG COMPUTING

P.O. BOX 23, WORCESTER, MA 01603

MOVING?

DON'T MISS A SINGLE ISSUE

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

DO YOU HAVE A QUESTION ABOUT YOUR SUBSCRIPTION?

Check the appropriate boxes below:

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

MAIL TO: **ST-LOG,** P.O. BOX 625, HOLMES, PA 19043

Name _____

Street Address _____

City _____ State ___ Zip ___

ATTACH LABEL HERE

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

ST-Log Staff

Editors/Publishers: Michael J. Des Chenes, Lee H. Pappas. *Managing Editor:* Diane L. Gaw. *East Coast Editor:* Arthur Leyenberger. *Midwest Editor:* Matthew J.W. Ratcliff. *West Coast Editor:* Charles F. Johnson. *Contributing Editors:* Ian Chadwick, Arnie Katz, Bill Kunkel, Maurice Molyneaux, Steve Panak, D.F. Scott, Douglas Weir, Joyce Worley. *Technical Division:* Charles Bachand, Clayton Walnum. *Cover photo:* Jim Langone. *Contributing Artist:* Gary Lippincott. *Production Editor:* Kathy Wiesner. *Graphics:* Connie Moore. *Typography:* Edythe Stoddard, Lisa Openshaw. *Advertising Manager:* Michael J. Des Chenes. *Accounting/ Circulation:* Robin Levitsky. *Contributors:* Peter Beery, Deborah Elder, Dale Hughes, Paul Lay, Peter Naleskiewicz, David Plotkin, Matthew Stern.

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

Where to write

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

Correspondence regarding subscriptions, including problems and changes of address, should be sent to: **ST-Log**, 100 Pine Street, Holmes, PA 19043, or call 1-800-345-8112.

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.

Advertising Sales

East Coast	Gerald F. Sweeney & Associates — (212) 242-3540 P.O. Box 662, New York, NY 10113
West Coast	J.E. Publishers Representatives — Los Angeles: (213) 467-2266. San Francisco: (415) 864-3252. Chicago: (312) 445-2489. Denver: (303) 595-4331. 6855 Santa Monica Blvd., Suite 200, Los Angeles, CA 90038
ST-Log offices	Michael Des Chenes, National Advertising — (617) 892-9230

Address all advertising materials to: Michael Des Chenes — Advertising Production, **ST-Log**, 565 Main Street, Cherry Valley, MA 01611.

Permissions

Subscriptions

ST-Log, P.O. Box 625, Holmes, PA 19043; or call toll-free: 1-800-345-8112. Payable in U.S. funds only. U.S.: $28.00-1 year; $52.00-2 years; $76.00-3 years. Canada: $36.00-1 year; $68.00-2 years; $99.00-3 years. Foreign: $39.00-1 year; $72.00-2 years; $104.00-3 years. 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.

Editorial

This month, as displayed on our cover, **Four-Star Software** is our prominent feature. We asked several enthusiasts, who have been long-time Atari users, to think carefully and provide us with a list of what they consider to be "classic" software for their ST computer. Each of these selections is based on products they personally enjoy and use frequently.

Before you survey our choices, you may want to check out this issue's **Ian's Quest** on page 67. Ian Chadwick pinpoints what makes software good or bad, and his column also makes an excellent introduction to the four-star list which starts on page 56. It's easy to see by our choices that what makes a piece of software a good, viable product has much to do with personal opinions. For instance, the list of entertainment software is fairly long, giving evidence that what appeals to one user may not overly thrill a dozen others.

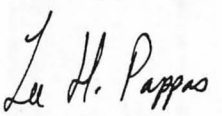
What our software picks also demonstrate is that there's a wide variety of products on the market, that the ST line is fairly well supported and that there's still a need for more quality software—to fill certain voids in the areas of productivity and specific applications. Software developers take note.

Back to the future. . .Our May issue, specializing in music and MIDI on the ST, initiated such an overwhelming response that we plan to have continued coverage of the subject on a frequent basis. As of this writing, we will be attending the NAMM show in Chicago (late June), which is oriented toward the music industry, including production people and recording artists.

Atari will also be there, as the first computer company ever to exhibit at one of these shows, proving that the computer/music combination is here to stay—and in a big way, with Atari at the forefront. In addition, many of the various companies supporting ST software and hardware aspects will also be on hand.

Our leading reporter on the music front is Charles "Icarus" Johnson, our West Coast Editor, based in the LA area. Charles recently completed a tour of Indonesia and Australia with Al Jarreau, has previously played with the group Chicago, and is now playing with his own band, Catzeye. He also happens to be a crackerjack Atari programmer on both the STs and the 8-bits. Charles can be found on Delphi if you need musical help in a hurry.

Finally, we're working hard with the Boston Computer Society Atari Users Group, to make the upcoming Northeast Atari Computer Fair a mega-event for the Atari community. We'll be announcing continued plans for the show as they develop. This is going to be the Atari show of the year, so start making your plans to visit New England this October, on Columbus Day weekend!



Lee H. Pappas
Publisher
ST-Log

NEW

ATARI 520ST SYSTEM FM PACKAGE

* RGB Monitor or Monchrome Monitor
* Built in 3 1/2" SS Double Density Drive
* Basic
* TOS on ROM
* RF Modulator
* Full Manufacter's Warranty

CALL FOR LOWEST PRICE

ATARI 1040ST SYSTEM PACKAGE

* RGB Monitor or Monochrome Monitor
* Built in 3 1/2" DS Double Density Drive
* Basic
* TOS on ROM
* Full Manufacter's Warranty

CALL FOR LOWEST PRICE

ATARI SF 314 DISK DRIVE

DOUBLE SIDED/1 MEGABYTE STORAGE

CALL

OKIMATE 20 COLOR PRINTER & ST PLUG N' PRINT

$205

SUPRA 20 MEG HARD DRIVE ONLY

$539

PC DITTO

$84.95

ST MODEM PACKAGE

* AVATEX 1200HC MODEM
* ST MODEM CABLE
* ST TALK TELECOMMUNICATIONS PACKAGES

$145

PANASONIC

PANASONIC 1080i	159
PANASONIC 1091i	189
PANASONIC 1092i	299
PANASONIC 3131	269
PANASONIC 3151	409
PANASONIC 1592	419
PANASONIC 1595	449

EPSON

LX 86	189
FX 86E	339
EX 800	409
EX 1000	549
FX 286E	479
LQ 800.	469
LQ 1000	669
LQ 2500	929

STAR MICRONICS

NX-10	179
NL-10	CALL
NP-10	149
NX-15	329
ND-10	349
NR-15	479
NB-24/10	439
NB-24/15 (24 wire)	639

OKIDATA

OKIDATA 120	205
OKIDATA 182	249
OKIDATA 192+	369
OKIDATA 193+	559
OKIDATA 292E	449
OKIDATA 293E.	599

CANON

CANON A-50	279
CANON A-55	389
CANON A-60	349

SEIKOSHA

SP-180	169
SP-1000A	199
SP-1200	209

BROTHER

BROTHER 1009	139
BROTHER 1509	379
BROTHER 1709	489
BROTHER HR 35	749
BROTHER 2024 LD	819
WINWRITER 5	839

MICHTRON

BBS 2.0	48.95
Business Tool	31.95
Calendar	18.95
Cornerman	31.95
DOS Shell	24.95
Echo	24.95
M-Disk +	24.95
Major Motion	24.95
Make It Move	44.95
Mi-Term	31.95
Michtron Utilities	37.95
Cards	24.95
The Animator	24.95
Time Bandits	24.95
Mighty Mail	31.95
Personal Money Mgr	31.95
Pinball Factory	24.95
Financial Future	24.95
Eight Ball	18.95
Hard Disk Backup	24.95
Match-point	24.95
Karate Kid II	24.95
GFA Basic	48.95
Trimbase	62.95
Shuttle 2	24.95
M-Cache	24.95
GFA Draft	62.95
GFA Vector	31.95
Goldrunner	24.95
GFA Compiler	48.95
Realtizer	149.95

ST ACCESSORIES

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

ADVANCED ORDERS

Magical Myths	31.95
All About America	36.95
Adv. of Sinbad	31.95
Arabian Nights	31.95
1st Letters and Words	33.95

ST ADVENTURES

Hacker	17.95
Hacker II	25.95
Sundog	24.95
Black Cauldron	25.95
Apshai Trilogy	14.95
Universe II	44.95
The Pawn	29.95
Starglider	29.95
Dungeonmaster	24.95
Alternate Reality	26.95
Tass Times	25.95
Mercenary	26.95
Autoduel	32.95
Ogre	25.95
Defender of Crown	32.95
S.D.I.	32.95
Balance of Power	32.95
Ultima III or IV	38.95
Portal	32.95
Kings Quest 1,2 or 3	32.95
Space Quest	32.95
Tracker	29.95
Colonial Conquest	25.95
Roadwar 2000	25.95
Wizard's Crown	25.95
Rings of Zilfin	25.95
Sinbad	32.95
Golden Path	29.95
Phantasie 1,2 or 3	25.95
Leisure Suit Larry	32.95
Guild of Thieves	29.95
221 Baker Street	26.95

ABACUS

Textpro	32.95
Datatrieve	32.95
Forth MT	32.95
Paintpro	32.95
Assempro	39.95
Powerplan	49.95
Abacus Books	Call

ST DATABASES

DB Man	96.95
Regent Base	57.95
Data Manager ST	48.95
Zoomracks II	77.95
Superbase Gem	94.95

ST EDUCATIONAL

Aesop Fables	31.95
Decimal Dungeon	24.95
Fraction Action	24.95
Kinderama	24.95
Read & Rhyme	24.95
Math Wizard	24.95
Animal Kingdom	24.95
Speller Bee	33.95
Kid Talk	33.95
Math Talk	33.95
First Shapes	33.95
Read-A-Rama	31.95
Math Talk Fractions	33.95
Winnie the Pooh	16.95
Buzzword	27.95

ST UTILITIES

Music Studio	32.95
Digi-Drum	24.95
Time Link	33.95
Middiplay	32.95
Micro Cookbook	32.95
CZ Droid	64.95
EZ Track	39.95
ST Replay	99.95
Write 90	18.95
Fast	31.95
Labelmaster Elite	27.95
St Doctor	24.95
Back Pak	64.95
Video Wizard	39.95
K-Switch	25.95
Partner ST	44.95
Desk Cart	72.95

ST PRINT UTILITIES

Rubber Stamp	18.95
Printmaster Plus	24.95
Megafont ST	24.95
Art Gallery 1 or 2	18.95
Typesetter Elite	31.95
PM Interface	18.95
Certificate Maker	31.95
Library 1/Certificate Mkr	21.95
Print Shop	Call

ST LANGUAGES

Personal Pascal	49.95
Alice Pascal	49.95
Mark Williams C	114.95
Metacomco Pascal	69.95
Macroassembler	49.95
Lattice C	99.95
Cambridge Lisp	139.95
Modula II	67.95
Metacommco Make	49.95
Micro C shell	34.95
Modula II Developer	Call
Fast Basic	48.95
True Basic	49.95
Fortran 77	124.95
LDW Basic	44.95
MT C Shell	84.95

ST BUSINESS

VIP Professional	Call
Swiftcalc ST	48.95
Isgur Portfolio	124.95
Financial Cookbook	14.95
DAC Payroll	32.95
DAC Easy Accounting	44.95
Dollars and Sense	64.95
Sylvia Porter	48.95
A-calc Prime	39.95
Logistix Jr.	57.95
Logistix Sr.	89.95

DESKTOP PUBLISHING

Drawrite	99.95
Publishing Partner	99.95
Partner Fonts	19.95
Fleet ST Desktop Pub	74.95

ST GRAPHICS

Degas Elite	52.95
Easy Draw	64.95
Cad 3-D	32.95
Graphic Artist	124.95
Paintworks	25.95
1st Cadd	31.95
Aegis Animator	48.95
Pro Sprite Designer	36.95
Neochrome	34.95
Stereo Cad 3-D	67.95

ST ARCADE GAMES

Winter Games	24.95
Rogue	24.95
Super Huey	25.95
Mean 18	27.95
Famous Course Disk	14.95
Leader Board	25.95
Brattacus	32.95
Silent Service	25.95
Flight Simulator II	33.95
Champ. Wrestling	24.95
World Games	24.95
Video Vegas	24.95
Strip Poker	25.95
Little Computer People	16.95
Gato	24.95
Joust	20.95
Star Raiders	20.95
Chessmaster 2000	29.95
Tenth Frame	25.95
Shanghai	25.95
3-D Helicopter	32.95
Micro League Baseball	39.95
Gen Mgr/for MLB	19.95
'86 Team K For MLBB	16.95
WWF Micro Wrestling	39.95
Skyfox	14.95
Super Cycle	14.95
ST Pool	22.95

DISKETTES

Sony 3.5 SS/DD (10 PK)	13.95
Sony 3.5 DS/DD (10 PK)	18.95

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

ST WORD PROCESSORS

Regent Word II	48.95
Wordwriter ST	48.95
Thunder	26.95
Microsoft Write	Call
Word Perfect	Call

CASIO KEYBOARDS

CZ 101	Call
CZ 230S	Call

ST ARCADE GAMES

Indoor Sports	32.95
F-15 Strike Eagle	27.95
High Roller Simulator	32.95
Two on Two Basketball	25.95
Deep Space	19.95
Arena	14.95
ST Karate	22.95
The Wanderer	25.95
Champ Baseball '86	25.95
GFL Football	25.95
Harrier Srike	24.95
Space Stat/Protector	25.95
Sub Battle Simulator	25.95
Arctic Fox	26.95
RPV	24.95
Crystal Castles	20.95
Gridiron Football	33.95
ST Wars	24.95
Hardball	24.95
Psion Chess	38.95
Plutos	19.95
Q-Ball	19.95
Vegas Gambler	21.95
Vegas Craps	21.95
Airball	24.95
Bridge 5.0	22.95
Karateka	22.95

INFOCOM ST

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

ST TELECOMM

ST Talk Ver 2.0	CALL
BB/ST	31.95
Flash	27.95
I.S Talk	33.95

No surcharge for Master Card

To Order Call Toll Free
800-558 0003

P.O. BOX 17882, MILWAUKEE, WI 53217
ORDER LINES OPEN
Mon.-Fri. 9AM-9PM CST Sat. 11AM-5PM. CST

ComputAbility
Consumer Electronics

TELEX NUMBER 9102406440
(ANSERBACK = COMPUT MLW UQ)

No surcharge for Visa

For Technical Info, Order Inquiries, or for Wisc. Orders
414-357-8181

ORDERING INFORMATION: Please specify system. For fast delivery send cashier's check or money order. Personal and company checks allow 14 business days to clear. School P.O.'s welcome. C.O.D. charges are $3.00. In Continental U.S.A. include $3.00 for software orders 5% shipping for hardware, minimum $5.00. MasterCard and Visa orders please include card #, expiration date and signature. WI residents please include 5% sales tax. HI, AK, FPO, APO, Puerto Rico and Canadian orders, please add 6% shipping, minimum $6.00. All other foreign orders add 15% shipping, minimum $15.00. All orders shipped outside the Continental U.S.A. are shipped first class insured U.S. mail. If foreign shipping charges exceed the minimum amount, you will be charged the additional amount to get your package to you quickly and safely. All goods are new and include factory warranty. We do not guarantee compatability. Due to our low prices all sales are final. **All Defective returns must have a return authorization number.** Please call (414) 357-8181 to obtain an R.A. # or your return will not be accepted. Prices and availability subject to change without notice.

CIRCLE #102 ON READER SERVICE CARD

REGULAR FEATURE

Reader comment

Shape up, Atari.

Congratulations on a great pair of issues for the month of May. I follow both **ANALOG Computing** and **ST-Log**, and enjoy them very much.

I'm finding myself more and more frustrated with the behavior of Atari Corp., with regard to getting promised hardware out on time. Atari is giving new meaning to the words *airware* and *vaporware*.

Perhaps part of it is the fault of some Atari users for being so quick to spread rumors (I've heard in the neighborhood of eight different stories about the graphics enhancement Atari's supposed to be making, none of them substantiated *in the least* by Atari officials), but it cannot be denied that Atari is making serious mistakes in not keeping their promises.

When other companies announce new computers or hardware products, users can expect to see them in anywhere from a few weeks to two or three months. When Atari announces a new product, users can expect to see it in a year—if they're lucky. Often, though, the product is dropped from production without any explanation, leaving people to wonder for months when the product will hit dealers' shelves. They finally give up and stop waiting.

The bottom line is: Atari doesn't seem to care if they come out with what they announce or not.

Take the blitter chip, for instance. Rumors on it started to surface about a year ago, a while after I got my ST. I said, "Oh, how nice; now GEM will actually be up to a workable speed." I waited. . .and waited. The last I heard was that they'll be putting blitters in the Mega STs first, get people to buy Megas, and (maybe) make 1040 and 520 versions later.

What about the AMY chip? Rumors I've heard indicate that it will have a lot of voices and better sound than an Apple IIGS. People claim to have heard this spectacular sound, yet Atari has seemingly made no progress in many months of work.

Other products we're waiting for are the IBM Emulator, the GST (whatever that is—some have said a graphics enhancement with an unbelievable number of colors), the TT (I hear it will have *no* enhanced graphics, but will be merely a 68020 number cruncher), and a few million other things. In fact, the *only* hardware products they've come through on are the SH204 hard disk and the PC-something-or-other, which is an RGB monitor with a drive built in, just like a Macintosh. The release dates on the Mega ST and the Atari PC have been delayed again and again.

ST users are starting to get tired of waiting for Atari to make what they say they're going to make. At least two ST users near me have gone to Amiga for that reason. It's a shame, and it's hurting Atari's image very much.

I wonder how long it will take Atari to get on track and start fulfilling their commitments.

Peter Payne
San Diego, CA

This has, indeed, been a common complaint in the Atari world. Atari Corp. is aware of their credibility problem—it was a major topic of discussion when some of our writers met with Sam Tramiel, Neil Harris and Jerry Brown, Atari's new Vice President of U.S. Operations, at the June CES. Arthur Leyenberger reports: "Both Sam Tramiel and Jerry Brown agreed that premature announcements of new products have hurt Atari's credibility. The good news: Atari will be striving not to announce or discuss new products until they're reasonably sure they can deliver the items to consumers in a timely manner. I look forward to this new era in Atari credibility."

We all do. —*Ed.*

DX-position.

Congratulations on your May 1987 **ST-Log** (issue 14). As a long-time Atari owner (8-bit since 1981, 1040ST for a year), I've felt a bit strange buying a bunch of music magazines for their ST-related articles and reviews each month. The MIDI port on my ST has opened a whole new world for me. It's good to see that the ST's musical capabilities are starting to receive the recognition in the computer media that they've had for some time in the music press.

As an enthusiastic Yamaha DX100 owner, I was also pleased to see your brief review of my synthesizer. I feel that I must take issue with a couple of points in the review, though.

First, in many respects, the DX100 is

Attention Programmers!

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

All submissions for publication, both program listings and text, should be provided in printed and magnetic form. Typed or printed copy of text is mandatory and should be in upper and lower case with double spacing. By submitting articles to **ST-Log**, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of **ST-Log**. If not accepted for publication, the articles and/or programs will remain the property of the author. If submissions are to be returned, please supply a self-addressed, stamped envelope. All submissions of any kind must be accompanied by the author's full address and telephone number.

Send programs to:
Editor, **ST-Log**
P.O. Box 23, Worcester, MA 01603.

not easy to use. Yamaha's FM synthesis technique is generally acknowledged to be one of the least intuitive approaches available. Although it's simple to hook up and play the DX100, getting to half of its 192 voices requires a fair amount of research —*nobody* could figure out how to do it from the front panel controls/labels. Programming the machine using those controls can be a lengthy, frustrating exercise in trial and error. Fortunately, a good patch editor on the ST (I use and recommend Beam Team's **Transform** software) can make the job almost easy.

Second, the manual is not very good. It's about as clear as early Digital Research CP/M documentation—a legend in the annals of obfuscation. You'll need to read it repeatedly to understand how to do relatively simple things. You'll have to read it often, because the nonintuitive nature of those "simple things" makes the keystroke commands unmemorable.

Third, and finally, there isn't nearly as much software available to support the DX100 as there is to support Casio's CZ101. Much "DX" software on the market actually supports Yamaha's flagship DX7 synths and is incompatible with the DX100. The DX100 software situation is improving daily, but it may never equal the support available to the CZ101.

The intent of the above is not to dissuade ST owners from running out and buying a DX100. Indeed, I consider that the cost/performance ratio of the DX100 makes it the "ST" of the synthesizer world. I just don't think that prospective buyers should walk into their local musical instrument stores with unrealistic expectations. The DX100 is a powerful, complex music-making system, and it's going to take some time for a new user to explore its many features. The work is well worth the effort.

Please continue your editorial coverage of the music-making aspects of the ST, along with your outstanding coverage of the other features of our wonderful computing machine.

Dan Deckert
Los Angeles, CA

A call to Arts (Electronic, that is).

The news in issue 14 that Batteries Included is being taken over by Electronic Arts certainly came as a shocker. What was even more disheartening was reading about the piracy that plagued BI prior to the takeover.

When are people going to learn that ripping off software only hurts the user by resulting in no new product?

That point aside, we all know how pro-Commodore Electronic Arts is. Could this takeover mean the end of Atari versions of BI software? Only time will tell...

In that same issue, EA President Trip Hawkins comments that he thinks **Starglider** is the only worthwhile game for the ST. Come on! What about **Brataccas**? **Time Bandit**? **Sundog: The Frozen Legacy**? **The Pawn** and **Guild of Thieves**, or Epyx's **Rogue**?

Once more, Mr. Hawkins has put his Adidas in his mouth where the ST is concerned (after all, he is the fellow who said the ST had no future). Instead of making

ill-informed statements about the ST, Mr. Hawkins should be getting the EA staff to make Atari 8-bit ports of such long overdue games as **Skyfox** and **Marble Madness**.

Zephery Hughes

*Mr. Hawkins has some good points, made in an interview conducted by D.F. Scott: **Inside Electronic Arts**, on page 19. We can probably swing his favor toward Atari STs by* buying, *not stealing, his software.* —*Ed.*

Back to BASICs.

I have been a subscriber to **ANALOG Computing** for four years or better, and have recently subscribed to **ST-Log**. I have let my subscription to **ANALOG** expire, although I will continue to buy it from newsstands when programs and articles I find useful appear, in respect to the Atari ST computer.

Frankly, I subscribe to or buy all ST-related computer magazines from newsstands, and, to be quite honest . . . **ANALOG/ST-Log** and *ST Applications* do the best jobs in giving us the most interesting articles and programs.

You really outdid yourselves in **ANALOG Computing**'s issue 54, May 1987, with a very fine program in BASIC titled **Étude in C: Minor**. Graphically, the program was excellent, and the article was well written by David Lindsley.

Lo and behold, in the April 1987 **ST-Log**, you gave us **Escher Cubes** by always-popular James Luczak, and also **ARC Shell II** by Charles F. Johnson.

Thanks for the excellent BASIC programs. I'm sure many readers will feel as I do.

A lot of us old Atari 800 and 800XL carryovers to the ST happen to enjoy programming in BASIC and truly appreciate well executed programs in that language.

I have written many letters to Atari-related computer magazines; I've probably made a pest of myself to some. The point is, as a subscriber, I feel I should make myself heard, and would like to receive at least some programs that I enjoy, or subscribing to computer magazines would be useless.

I thank you for the BASIC programs, useful ones at that. I have stated in the past that several ST magazines only promote games. I think most of us enjoy games, but also would like to see useful and entertaining programs in graphics and sound.

Being a recent purchaser of **GFA BASIC Interpreter** and **GFA BASIC Compiler**, I've found—in the short time I've played around with GFA BASIC—it an outstanding version of BASIC. In my humble opinion, it far surpasses ST BASIC, as far as speed and simplicity of the many commands is concerned. MichTron has compiled a new BASIC manual, and is in the process of publishing a new book on GFA BASIC. This new BASIC should help the Atari ST sales, as well doing away with the multiple programming window nightmare of ST BASIC.

I would like to see some routines and programs (in both of your magazines) in GFA BASIC. I believe they would add enthusiasm for the ST.

I'm sure BASIC is not for everybody, but we now have a powerful BASIC available to us. In many cases, it will do what Modula-2 and Pascal have done. I certainly intend to make GFA BASIC my language of choice. I need not look any further. I happen to feel comfortable in BASIC, and am perfectly content, now that we have an excellent version available to us.

Please give us a little help, and include programs in this new BASIC. It will certainly be appreciated.

Very respectfully,
Mario Sala
Torrance, CA

You heard it, programmers. Let's see those GFA BASIC masterpieces you've been working on. —*Ed.*

Send your letters to:

Reader comment

ST-LOG
P.O. Box 23
Worcester, MA 01603

ON NEXT MONTH'S DISK VERSION:

ST-Xformer
AN 8-BIT EMULATOR

MasterPlan™

FINANCIAL SPREADSHEET

MasterPlan ™ is an inexpensive, powerful, easy to use spreadsheet written totally within Digital Research's "GEM" operating system. MasterPlan utilizes extensively the graphics and mouse capabilities of the ATARI ST personal computer. HELP is built right into the program. You will be able to create worksheets and graphs within minutes.

Atari ST™ Special Features.

Mouse or keystroke operation.
GEM interface with pull-down menus, icons, dialog boxes, ranging, scroll bars, column grabber, etc.
Sparse matrix for conserving memory.
Supports hard drives & printers supported by GEM.
Supports color or monochrome monitors.
Ultra-fast floating point math. Multiple windows for viewing graphs and worksheet at the same time.

Spreadsheet Features

Reads/Writes .WKS data files, so MasterPlan will work with both Lotus 1-2-3 or VIP Professional data files. Giant 8192 x 256 spreadsheet grid. Date, Financial, Logical, Mathematical, Special and Statistical Functions. Cell and range operations. Cell and range naming. Cell Pointer expansion. Automatic, manual & natural recalculation. Optional display of formulas in cells. Print formatting options include printing borders, margin control, headers & footers, automatic pagination. Separately formattable windows. Frozen titles horizontally, vertically or both. Copy or move cell or range contents. Delete or insert rows or columns. Undo editing command. Protection of all or portions of the worksheet. Change column-width globally or with mouse. Import data from other business programs.

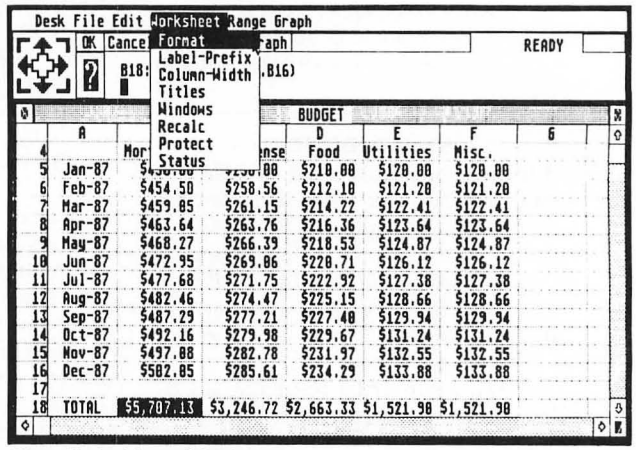
Graph Features

Line, bar, stacked-bar, pie, and scatter graphs. "Exploded" pie graph. "UNDO" toggle graph. Displayable in color or monochrome. Up to six separate data ranges. Four different graph titles. Automatic or manual scaling. Symbols, line or data-labels for line graphs. Legends for labelling data ranges. Horizontal and/or vertical grids. Multiple graphs can be created and stored for each worksheet. Saves files compatible with DEGAS & Publishing Partner.

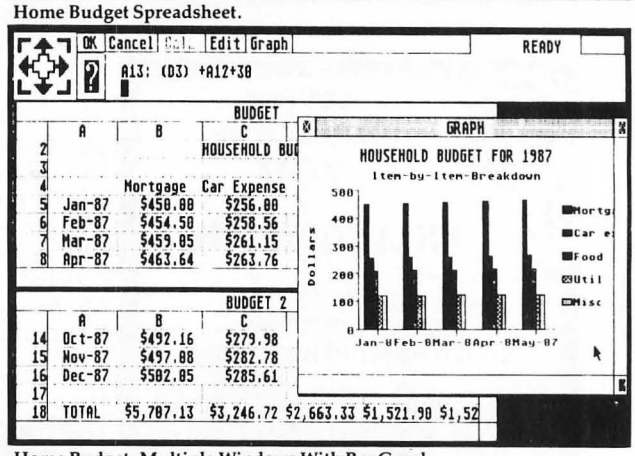
SYSTEM REQUIREMENT: Atari ST minimum 512K. 0.5 meg drive & color or monochrome monitor.

Receive all the power of MasterPlan with color presentation graphs for only **US$129.99*** suggested retail.

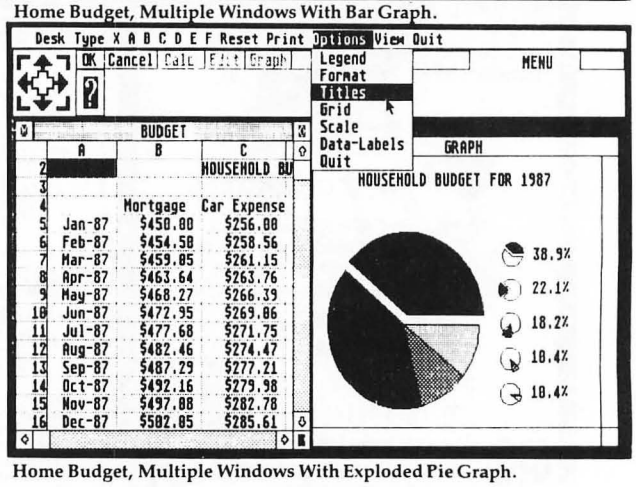
*Dealers may sell for less.



Home Budget Spreadsheet.



Home Budget, Multiple Windows With Bar Graph.



Home Budget, Multiple Windows With Exploded Pie Graph.

CONTACT YOUR LOCAL ATARI DEALER AND ASK FOR THE POWER OF

MasterPlan™

CIRCLE #113 ON READER SERVICE CARD

ISD MARKETING, INC., 2651 John St., Unit 3, Markham, ONT, Canada L3R 2W5. (416) 479-1880. Fax # 1-416-479-1882

Atari sets off fireworks!

Using *Zoomracks* to organize the light show.

by Matthew Stern

The Atari 800XL at Astro Pyrotechnics produces the most spectacular graphics you've ever seen. Its screen is the entire night sky. Sprites burst into ribbons of color. It can punctuate images with thunderous bangs and cannon bursts. The Astro Pyrotechnics Atari doesn't paint with pixels. It paints with fire.

Astro Pyrotechnics is using Atari computers to run its fireworks business. Their computers not only produce information; they can produce an entire fireworks show.

Instrumental in adapting Ataris to the fireworks industry is Robert Veline, who works in research and development. He brought his fascination with computers and electrical engineering to his profession. With Atari computers, Robert says, "We are using technology to produce better art."

Preparing for blast-off.

Producing spectacular fireworks shows requires careful planning and organization. Large shows, like the ones Astro Pyrotechnics puts on for the Hollywood Bowl, Magic Mountain in Valencia, California and Knott's Berry Farm in Buena Park, California, can use 700 to 1,000 rounds of fireworks. To produce a successful show, all the fireworks have to be in the right place at the right time.

For this important task, Robert uses an Atari 520ST and **Zoomracks** from Quickview Systems.

Robert enters into **Zoomracks** a schedule, called a *queue sheet*, which lists all the events in the show (called *queues*) and the fireworks needed for them. With this information, **Zoomracks** provides a number of reports: a pull sheet listing the fireworks needed for the show, an inventory list indicating which items are available in stock, and packing lists designating which fireworks go to which location.

Before using **Zoomracks**, Astro Pyrotechnics had to put together inventory and packing lists by hand. Workers had to carefully check and recheck their queue sheets to make sure they had all the fireworks they needed. Even then, it was easy to make mistakes.

Robert said putting queue sheets on computer has greatly reduced the chance of error. "We can double and triple check to see if we have all the items there. It's not as likely you're going to miss one queue. It eliminates those times when the truck is in the driveway and someone shouts, 'Hey, we forgot the yellows!' "

Zoomracks also makes it easy to locate fireworks for certain special effects. For example, if a queue calls for red, white and blue aerial bursts, **Zoomracks** can search for fireworks that can perform the effect, and tell how many of them are in inventory.

The ST also simplifies the complicated job of calculating expenses. Figuring the manufacturing cost of each firework requires a sophisticated formula based on labor costs and the amount of chemicals used. Astro Pyrotechnics uses about 120 different formulas.

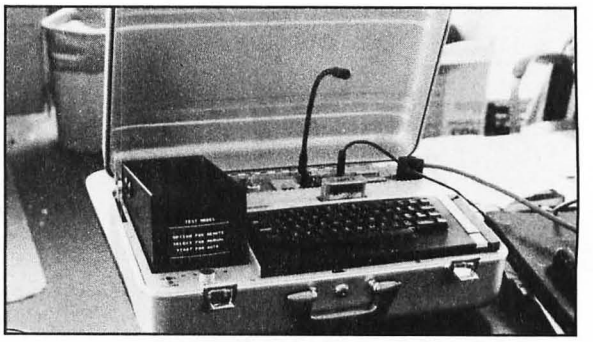
Plant Manager Stewart Carlton admitted that, be-

fore using computers, they seldom calculated these costs because it was too time consuming. Since Robert put this information in a **VIP Professional** spreadsheet, Astro Pyrotechnics can generate cost information in seconds. When material prices change, the totals can be updated instantly.

Stewart is a fireworks industry veteran who still likes to do things traditionally. But he likes using the Atari ST, because "it is simple enough for someone who is not a computer nut to use."

Robert remarked that the computer still requires time to key in the information. However, computers "present information in more comfortable formats to work with. We can get information in packages we didn't have before."

Robert Veline with the ST, which Astro uses to produce the shows.



Computerized firing box uses the Atari 800XL.

Getting a bang out of an Atari.

Robert and his step-brother Rick Rolle designed and built a computerized firing box with an Atari 800XL. A firing box is a device that sends an electrical charge to the fireworks, to set them off.

The Atari firing box is in a metal briefcase. Inside, there's a normal 800XL and cover. A small black-and-white TV serves as a monitor. The software has been transferred to a cartridge, so it's loaded upon starting up. The box contains a built-in rechargeable battery that can run for an hour and a half. (Most fireworks shows last just fifteen minutes.)

Robert explained, "It is still a full-functioning Atari. You can run BASIC programs on it just as on any other Atari."

But this firing box *isn't* like any other Atari. It contains additional features for testing and launching the fireworks. Instead of a normal on/off button, there's a locking switch. This prevents just anyone from turning on the machine and launching the fireworks. Transistorized circuits send electrical charges to the fireworks when signaled by the computer.

Four serial ports link the computer to up to four terminal boxes. Each box has five rows of screws: one screw for each queue. From the screws, workers attach long wires to electric matches (called *squibs*) which launch the fireworks. The firing box can launch up to 400 queues, each having 10 to 20 rounds of fireworks. Most shows only use 100 to 150 queues.

The Atari firing box can check all the circuits before firing, and display the ones that aren't working. This eliminates the chance of missing a queue because the circuits weren't ready.

Fireworks can be launched manually or automatically. The firing box has a jack for plugging in a manual firing button. In live shows, like musical concerts at the Hollywood Bowl, fireworks are launched manually. Robert told me that, at a performance of the *1812 Overture*, a firing button was given to the drummer so he could fire the cannon at the precise beat.

For shows with prerecorded soundtracks, fireworks can be launched automatically. The soundtrack sends a tone to the computer when a display is to be launched. The Atari firing box launches the next queue of fireworks. Robert keeps the manual firing button handy, in case the computer doesn't receive the tone or launch the fireworks.

After the show, the firing box indicates any fireworks that weren't launched. The crews can thus find out which fireworks may still be dangerous, to make cleaning up much safer.

Before putting the Atari firing box together, Astro Pyrotechnics used a TTL terminal. It could launch fireworks by pushbutton or tone, but it had limited testing capabilities.

High-flying plans for the future.

Astro Pyrotechnics plans to use their Ataris to further automate their shows.

Robert said he's just started scratching the surface of what the Atari firing box can do. "It offers maximum flexibility," he explained, "because it's a computer."

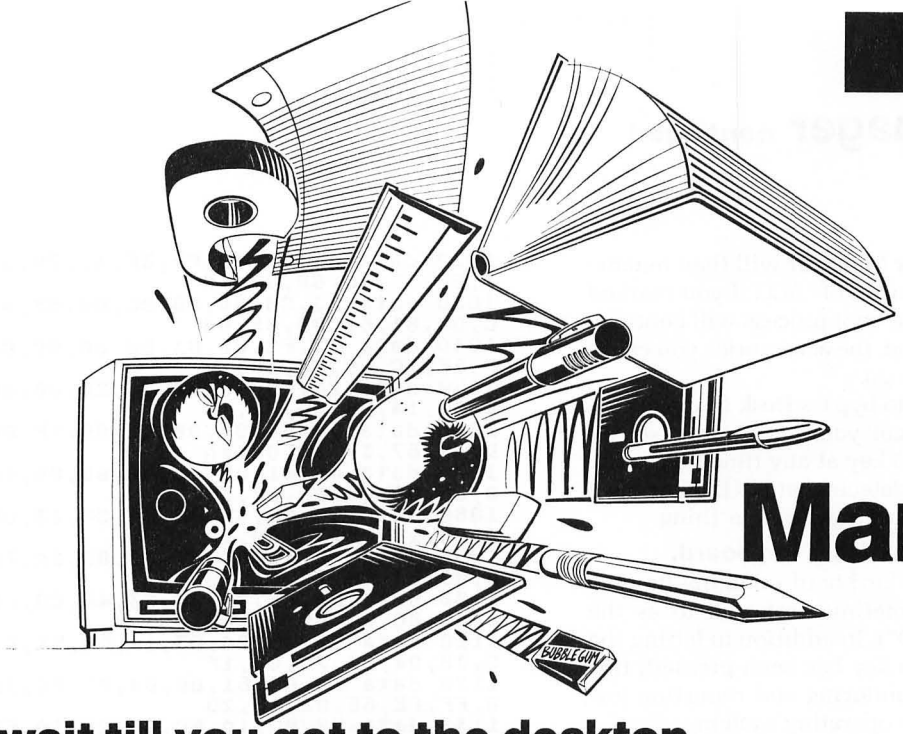
One feature Robert is working on is the ability to disable fireworks prohibited under adverse weather conditions. "If it's too windy," he explained, "the Fire Marshall won't allow us to launch certain fireworks. I'll be able to flag those queues and skip them under those conditions." The Atari firing box can also change to alternate queues with acceptable fireworks.

Eventually, Robert wants to take a queue sheet generated on the ST, transfer it to the cartridge, and have the firing box launch the fireworks automatically.

Robert also hopes to design fireworks displays right on his computer screen. One step toward that goal is a program he wrote on the ST in GFA BASIC. It flashes multicolored aerial fireworks against a black screen. The colored lights fall to earth according to their fireworks type and gravity. "Some day," Robert mused, "I'd like to see a fireworks CAD program."

The field of fireworks is still one for craftsmen. Most fireworks are still made by hand. Displays are set up and wired by skilled workers. With Atari computers, Astro Pyrotechnics is bringing high technology to an ancient art. //

Matthew Stern has written about a number of computer systems, and currently works as a technical writer at AST Research. You can reach him at the ATARI16 Forum on CompuServe, ID number 73547, 2420.



Desk Manager

Don't wait till you get to the desktop—choose your accessories in advance

by Charles F. Johnson

I like desk accessories—they're fun. And today's ST owner has a variety of excellent commercial and public domain accessories to choose from. Desk accessories (we'll call 'em DAs, from here on) provide a sort of limited multitasking for the ST, since they can usually be accessed from any GEM program that displays a menu bar. The ANALOG Atari SIG on Delphi has a wide assortment of accessories available to download—from calculators to clocks, to simple word processors.

A maximum of six DAs can be loaded into the ST when you boot up. If your computer doesn't have TOS in ROM (and why not, may I ask?) the DAs must be named DESK 1.ACC, DESK2.ACC, etc. With ROM TOS, accessories can have any name, as long as the extension is .ACC. The problem with either scheme is that it's rather inconvenient to change the DAs you want to load. Essentially, you have two choices: you can copy them to another disk, then delete them—or you can rename them to something like .ACX. This renaming process can be tedious, particularly if you only want a DA active at certain times, or want to test a new one.

Which brings us to the purpose of **Desk Manager**, my latest ST assembly language "masterpiece" (tongue planted firmly in cheek). **Desk Manager** lets you choose the accessories you want to use at boot time, before you ever reach the GEM desktop. It reads every DA's name from the disk you specify (drive A: or C:, for hard drive owners), puts them on the screen and lets you mark the ones you wish to load. This makes it very easy to use different DAs or to try new ones; this program can save you a lot of time.

How to use it.

If you don't already have one, create a folder on your boot disk called AUTO, and copy DESKMGR.PRG into it. Make sure to use that name exactly, or some program features may not work correctly. That's all there is to it.

The first time you boot with this disk, **Desk Manager** will ask you which drive holds your accessories. Normally, accessories will load from drive A, but the Atari hard disk driver forces TOS to look for DAs on the hard disk, as drive C:. If you have a hard drive, type *C*; otherwise, type *A*. **Desk Manager** will then modify itself, storing the new accessory drive specifier at the proper location within the DESKMGR.PRG file. Make sure you don't change the disk in drive A; if you do so during this procedure, the program will abort.

Whenever **Desk Manager** runs from now on, it will use the default drive you entered, saving you from having to type the drive letter every time you boot your ST. If you ever wish to change this drive data (say, if you upgrade to a hard disk), hit the HELP key before **Desk Manager** loads, and you'll see the *Which drive* prompt again.

When all this is finished, **Desk Manager** searches the main directory of the chosen drive for all filenames with an extension of .AC? (the question mark means "any character"), sorts them alphabetically, and displays them on the top of the screen with an arrow pointer resting on the first name. The program allows a maximum of 128 accessory names to be displayed.

If a DA already has an extension of .ACC, it will be shown in inverse (marked), so you can have a default selection of DAs without having to mark them every time you boot. Use the arrow keys or the mouse to move this pointer around in any direction, and press RETURN or the left mouse button to select and deselect accessories. If you want to deselect all the accessories at once, press CLR HOME. Since GEM allows a maximum of six accessories, **Desk Manager** lets you mark only six names; you'll hear a pleasant bell sound if you try to mark more.

When you're done choosing DAs, press the ESCAPE key

or the right mouse button. **Desk Manager** will then rename your accessories with an extension of .ACC if you marked them, or .ACX if you didn't. The boot process will continue and, when you reach the desktop, the accessories you chose will be installed and ready to go.

By the way, if you ever wish to bypass **Desk Manager** entirely (e.g., if you've already got your accessories set up properly), just press the UNDO key at any time before the program loads. If the program detects that UNDO has been pressed, it will abort without doing a single thing.

The mouse that thinks it's a keyboard.

As you may know, the ST's keyboard contains its own microprocessor, the 6301 (sometimes referred to as the IKBD, or "Intelligent KeyBoarD"). In addition to letting the operating system know when a key has been pressed, this chip is also responsible for monitoring and reporting joystick and mouse events to the operating system.

The IKBD makes life a lot easier for the 68000 chip in your ST; without it, the 68000 would be responsible for keeping track of all this stuff by itself. What makes the IKBD especially interesting is that many of its functions can be altered, by using an XBIOS call to send a string of command bytes directly to the chip. **Desk Manager** sets the IKBD so that mouse movements are reported as if the user had pressed one of the cursor control keys on the keyboard.

Any GEMDOS or BIOS keyboard input function will read the mouse in this way—which means that, if you've already written code to handle cursor keys (like **Desk Manager**), not a thing needs to be changed to let the program use either mouse or keyboard for input. And it works quite independently of GEM, allowing you to use this technique even with programs designed to run from the AUTO folder before GEM is initialized.

If you'd like to learn more about how to program the IKBD, I suggest the Abacus book *ST Internals*, pages 67 through 84. //

Charles F. Johnson is a musician by trade, drawn into the Atari computers by ***Star Raiders*** *about five years ago. When he got the BASIC cartridges, he was seriously hooked. He currently lives in Los Angeles with his wife and the most intelligent cat in the world.*

Listing 1.
ST BASIC listing.

```
100 filename$="a:\DESKMGR.PRG"
110 fullw 2:clearw 2:gotoxy 0,0:print
"creating file..."
120 option base 0
125 dim a%(16000):def seg=1:v$=""
130 p=varptr(a%(0)):bptr=p+1
140 for i%=1 to 3243
150 read v$:code%=val("&H"+v$)
160 poke p, code%:print ".";
170 p=p+1
180 next
190 bsave filename$,bptr,3243
200 print "file written":end
1000 data 60,1A,00,00,09,74,00,00,02,7
8,00,00,0A,9E,00,00
1010 data 00,00,00,00,00,00,00,00,00,0
0,00,00,60,02,58,00
1020 data 3F,3C,00,FF,3F,3C,00,06,4E,4
1,58,8F,4A,80,67,18
1030 data 23,C0,00,00,0C,00,48,40,B0,7
C,00,61,67,00,05,04
1040 data 33,FC,00,01,00,00,0C,04,2A,4
F,2E,7C,00,00,16,72
1050 data 2A,6D,00,04,20,2D,00,0C,D0,A
D,00,14,D0,AD,00,1C
1060 data D0,BC,00,00,01,00,2F,00,2F,0
D,42,67,3F,3C,00,4A
1070 data 4E,41,DF,FC,00,00,00,0C,4A,8
0,66,00,04,C6,3F,3C
1080 data 00,04,4E,4E,54,8F,33,C0,00,0
0,0C,10,3F,3C,00,2A
1090 data 4E,41,54,8F,38,00,2A,7C,00,0
0,09,D4,C0,7C,01,E0
1100 data EA,48,34,00,53,40,C0,FC,00,0
4,D0,BC,00,00,0B,B7
1110 data 20,40,7A,03,1A,D8,51,CD,FF,F
C,30,04,C0,7C,00,1F
1120 data 34,00,61,00,04,8E,0C,2D,00,3
0,FF,FE,66,04,1B,25
1130 data 52,8D,1A,FC,00,2C,1A,FC,00,2
0,1A,FC,00,31,1A,FC
1140 data 00,39,30,04,C0,7C,FE,00,E0,4
8,E2,48,D0,7C,00,50
1150 data 34,00,61,00,04,5E,1A,FC,00,2
0,1A,FC,00,7C,1A,FC
1160 data 00,20,30,04,C0,7C,00,1F,B0,7
C,00,09,6E,04,1A,FC
1170 data 00,20,3F,3C,00,2C,4E,41,54,8
F,38,00,C0,7C,F8,00
1180 data E0,48,E6,48,34,00,4A,40,66,0
6,70,0C,42,46,60,16
1190 data B0,7C,00,0B,6E,04,42,46,60,0
C,B0,7C,00,0C,67,04
1200 data 90,7C,00,0C,7C,01,61,00,04,0
A,0C,2D,00,30,FF,FE
1210 data 66,06,1B,7C,00,20,FF,FE,1A,F
C,00,3A,30,04,C0,7C
1220 data 07,E0,EA,48,34,00,61,00,03,E
A,1A,FC,00,3A,30,04
1230 data C0,7C,00,1F,E3,48,34,00,61,0
0,03,D8,1A,FC,00,20
1240 data 4A,46,66,08,28,7C,00,00,09,F
6,60,06,28,7C,00,00
1250 data 09,F8,1A,DC,1A,DC,1A,FC,00,2
0,1A,FC,00,7C,7A,03
1260 data 1A,FC,00,20,51,CD,FF,FA,1A,F
C,00,1B,1A,FC,00,71
1270 data 42,1D,4A,79,00,00,0C,10,66,2
A,3F,3C,00,01,2F,3C
1280 data FF,FF,FF,FF,2F,3C,FF,FF,FF,F
F,3F,3C,00,05,4E,4E
1290 data DF,FC,00,00,00,0C,42,67,3F,3
C,00,03,3F,3C,00,07
1300 data 4E,4E,5C,8F,20,7C,00,00,09,7
4,61,00,03,5A,0C,39
1310 data 00,58,00,00,00,02,67,0A,0C,3
9,00,62,00,00,0C,01
1320 data 66,7A,20,7C,00,00,0A,89,61,0
0,03,3C,3F,3C,00,07
1330 data 4E,41,54,8F,C0,7C,00,DF,B0,3
C,00,41,67,06,B0,3C
1340 data 00,43,66,DE,13,C0,00,00,00,0
2,7A,02,2A,7C,00,00
1350 data 0B,4D,61,00,07,20,6B,00,03,0
A,33,C0,00,00,0C,12
1360 data 42,67,3F,39,00,00,0C,12,2F,3
C,00,00,00,1E,3F,3C
1370 data 00,42,4E,41,DF,FC,00,00,00,0
A,48,79,00,00,00,02
1380 data 2F,3C,00,00,00,01,3F,39,00,0
0,0C,12,3F,3C,00,40
1390 data 4E,41,DF,FC,00,00,00,0C,61,0
0,07,04,10,39,00,00
1400 data 00,02,13,C0,00,00,0B,86,13,C
0,00,00,0B,8F,13,C0
1410 data 00,00,0B,9F,13,C0,00,00,0B,8
```

```
1,20,7C,00,00,0A,C7
1420 data 61,00,02,A4,42,45,1A,39,00,0
0,00,02,9A,3C,00,41
1430 data 3F,05,3F,3C,00,0E,4E,41,58,8
F,42,79,00,00,0C,06
1440 data 42,79,00,00,0C,0A,2A,7C,00,0
0,0C,42,3A,3C,04,7F
1450 data 1A,FC,00,20,51,CD,FF,FA,26,7
C,00,00,11,42,28,7C
1460 data 00,00,10,C2,2A,7C,00,00,0C,4
A,7A,7F,42,1B,42,1C
1470 data 42,15,DB,FC,00,00,00,09,51,C
D,FF,F2,42,79,00,00
1480 data 0C,0E,42,79,00,00,0C,08,23,F
C,00,00,0C,42,00,00
1490 data 0B,F0,23,FC,00,00,0B,86,00,0
0,0B,EC,61,00,02,CA
1500 data 4A,80,66,1E,61,00,02,F2,32,3
9,00,00,0C,0A,B2,7C
1510 data 00,80,67,0E,61,00,02,C6,4A,8
0,66,06,61,00,02,DA
1520 data 60,E6,4A,79,00,00,0C,0A,66,0
E,20,7C,00,00,0B,5F
1530 data 61,00,01,F4,60,00,01,EC,61,0
0,05,68,61,00,03,1C
1540 data 61,00,02,38,20,7C,00,00,09,F
A,61,00,01,DA,42,39
1550 data 00,00,0C,14,42,39,00,00,0C,1
5,42,79,00,00,0C,0C
1560 data 13,FC,00,20,00,00,0A,F2,13,F
C,00,20,00,00,0A,F3
1570 data 20,7C,00,00,0A,F0,61,00,01,A
E,2A,7C,00,00,0A,F5
1580 data 61,00,04,E0,4A,79,00,00,0C,0
4,67,0E,20,39,00,00
1590 data 0C,00,42,79,00,00,0C,04,60,0
8,3F,3C,00,07,4E,41
1600 data 54,8F,B0,3C,00,1B,67,00,01,1
8,B0,3C,00,0D,66,0E
1610 data 61,00,03,D4,33,FC,00,01,00,0
0,0C,08,60,DC,48,40
1620 data B0,3C,00,74,67,EA,B0,3C,00,7
5,67,00,00,F4,B0,3C
1630 data 00,4B,66,32,2A,7C,00,00,0B,0
6,61,00,04,86,42,84
1640 data 38,39,00,00,0C,0C,88,FC,00,0
8,48,44,4A,44,67,06
1650 data 53,79,00,00,0C,0C,61,00,04,3
4,2A,7C,00,00,0A,F5
1660 data 61,00,04,60,60,94,B0,3C,00,4
D,66,36,2A,7C,00,00
1670 data 0B,06,61,00,04,4E,42,84,38,3
9,00,00,0C,0C,52,44
1680 data 88,FC,00,08,48,44,4A,44,67,C
C,32,39,00,00,0C,0A
1690 data 53,41,B2,79,00,00,0C,0C,67,B
C,52,79,00,00,0C,0C
1700 data 60,B4,B0,3C,00,48,66,1C,2A,7
C,00,00,0B,06,61,00
1710 data 04,12,0C,79,00,08,00,00,0C,0
C,6D,9A,51,79,00,00
1720 data 0C,0C,60,92,B0,3C,00,50,66,2
A,2A,7C,00,00,0B,06
1730 data 61,00,03,F0,50,79,00,00,0C,0
C,32,39,00,00,0C,0A
1740 data B2,79,00,00,0C,0C,6E,00,FF,6
E,51,79,00,00,0C,0C
1750 data 60,00,FF,64,B0,3C,00,47,66,0
0,FF,00,61,00,00,BA
1760 data 2A,7C,00,00,0B,06,61,00,03,B
A,61,00,01,AE,42,79
1770 data 00,00,0C,0E,33,FC,00,01,00,0
0,0C,08,60,00,FE,90
1780 data 61,00,00,CC,4A,79,00,00,0C,0
8,67,4E,20,7C,00,00
1790 data 0A,CE,61,52,42,79,00,00,0C,0
C,23,FC,00,00,10,C2
1800 data 00,00,0B,F4,20,79,00,00,0B,F
4,4A,10,67,08,28,7C
1810 data 00,00,0B,B3,60,06,28,7C,00,0
0,0B,AF,61,00,01,EC
1820 data 52,B9,00,00,0B,F4,52,79,00,0
0,0C,0C,30,39,00,00
1830 data 0C,0A,B0,79,00,00,0C,0C,66,C
A,20,7C,00,00,0B,24
1840 data 61,04,42,67,4E,41,2F,08,3F,3
C,00,09,4E,41,5C,8F
1850 data 4E,75,20,4D,72,01,10,FC,00,3
0,51,C9,FF,FA,72,01
1860 data 48,C0,81,FC,00,0A,48,40,11,0
0,06,10,00,30,48,40
1870 data 51,C9,FF,EE,54,8D,4E,75,2A,7
C,00,00,10,C2,28,7C
1880 data 00,00,11,42,7A,7F,4A,1D,67,0
8,0A,14,00,01,42,2D
1890 data FF,FF,52,8C,51,CD,FF,F0,4E,7
5,2F,3C,00,00,0B,E7
1900 data 3F,3C,00,02,3F,3C,00,19,4E,4
E,50,8F,4E,75,2F,3C
1910 data 00,00,0B,EA,42,67,3F,3C,00,1
9,4E,4E,50,8F,2F,3C
1920 data FF,FF,FF,FF,2F,3C,FF,FF,FF,F
F,2F,3C,FF,FF,FF,FF
1930 data 3F,3C,00,10,4E,4E,DF,FC,00,0
0,00,0E,2A,40,42,2D
1940 data 00,1D,42,6D,00,1E,4E,75,61,1
E,42,67,2F,39,00,00
1950 data 0B,EC,3F,3C,00,4E,4E,41,50,8
F,4E,75,61,0A,3F,3C
1960 data 00,4F,4E,41,54,8F,4E,75,48,7
9,00,00,0C,16,3F,3C
1970 data 00,1A,4E,41,5C,8F,4E,75,2A,7
C,00,00,0C,34,28,79
1980 data 00,00,0B,F0,7A,07,10,1D,B0,3
C,00,2E,67,08,18,C0
1990 data 51,CD,FF,F4,52,8D,20,7C,00,0
0,0B,B4,7A,02,BB,08
2000 data 66,26,51,CD,FF,FA,0C,79,00,0
6,00,00,0C,0E,67,18
2010 data 52,79,00,00,0C,0E,20,7C,00,0
0,10,C2,30,39,00,00
2020 data 0C,0A,11,BC,00,01,00,00,06,B
9,00,00,00,09,00,00
2030 data 0B,F0,52,79,00,00,0C,0A,4E,7
5,2A,7C,00,00,10,C2
2040 data 23,FC,00,00,0C,42,00,00,0B,F
0,13,FC,00,20,00,00
2050 data 0A,F2,13,FC,00,21,00,00,0A,F
3,3A,39,00,00,0C,0A
2060 data 53,45,20,7C,00,00,0A,F0,61,0
0,FE,9C,4A,1D,67,0C
2070 data 20,7C,00,00,0B,1E,61,00,FE,8
E,60,0A,20,7C,00,00
2080 data 0B,21,61,00,FE,82,20,79,00,0
0,0B,F0,61,00,FE,78
2090 data 06,39,00,0A,00,00,0A,F3,10,3
9,00,00,0A,F3,B0,3C
2100 data 00,71,66,0E,13,FC,00,21,00,0
0,0A,F3,52,39,00,00
2110 data 0A,F2,06,B9,00,00,00,09,00,0
0,0B,F0,51,CD,FF,A4
2120 data 20,7C,00,00,0B,21,60,00,FE,3
E,42,80,30,39,00,00
2130 data 0C,0C,20,7C,00,00,11,42,4A,3
0,00,00,67,00,00,86
2140 data C0,FC,00,09,D0,BC,00,00,0C,4
2,20,40,22,7C,00,00
2150 data 0B,92,24,7C,00,00,0B,A2,70,0
7,12,D0,14,D8,51,C8
2160 data FF,FA,42,80,20,7C,00,00,0B,8
A,22,7C,00,00,0B,92
2170 data 7A,07,10,11,B0,3C,00,20,67,0
6,52,89,51,CD,FF,F4
2180 data 7A,03,12,D8,51,CD,FF,FC,42,1
1,2A,7C,00,00,0B,A2
2190 data 7A,07,10,15,B0,3C,00,20,67,0
6,52,8D,51,CD,FF,F4
2200 data 7A,03,1A,DC,51,CD,FF,FC,42,1
```

```
5,48,79,00,00,0B,9F
2210 data 48,79,00,00,0B,8F,42,67,3F,3
C,00,56,4E,41,DF,FC
2220 data 00,00,00,0C,4E,75,42,81,32,3
9,00,00,0C,0C,20,01
2230 data C2,FC,00,09,D2,BC,00,00,0C,4
2,2A,41,28,7C,00,00
2240 data 11,42,08,74,00,00,00,00,26,7
C,00,00,10,C2,08,73
2250 data 00,00,00,00,66,32,0C,79,00,0
6,00,00,0C,0E,66,1A
2260 data 08,74,00,00,00,00,08,73,00,0
0,00,00,3F,3C,00,07
2270 data 3F,3C,00,02,4E,41,58,8F,4E,7
5,52,79,00,00,0C,0E
2280 data 20,7C,00,00,0B,1E,60,0C,53,7
9,00,00,0C,0E,20,7C
2290 data 00,00,0B,21,61,00,FD,30,13,F
9,00,00,0A,F2,00,00
2300 data 0B,1B,13,F9,00,00,0A,F3,00,0
0,0B,1C,52,39,00,00
2310 data 0B,1C,20,7C,00,00,0B,19,61,0
0,FD,0C,20,4D,61,00
2320 data FD,06,20,7C,00,00,0B,21,60,0
0,FC,FC,42,85,3A,39
2330 data 00,00,0C,0C,8A,FC,00,08,13,C
5,00,00,0A,F2,06,39
2340 data 00,20,00,00,0A,F2,48,45,CA,B
C,00,00,FF,FF,CA,FC
2350 data 00,0A,13,C5,00,00,0A,F3,06,3
9,00,20,00,00,0A,F3
2360 data 4E,75,20,7C,00,00,0A,F0,61,0
0,FC,BC,BB,FC,00,00
2370 data 0B,06,66,06,20,4D,60,00,FC,A
E,3F,3C,00,03,3F,3C
2380 data 00,05,3F,3C,00,03,4E,4D,5C,8
F,20,4D,61,00,FC,98
2390 data 3F,3C,00,04,3F,3C,00,05,3F,3
C,00,03,4E,4D,5C,8F
2400 data 4E,75,42,80,30,39,00,00,0C,0
A,C0,FC,00,09,23,C0
2410 data 00,00,0B,F8,90,BC,00,00,00,0
9,23,C0,00,00,0B,FC
2420 data 20,7C,00,00,0C,42,70,00,72,0
9,26,7C,00,00,10,C2
2430 data 42,83,78,01,43,F0,00,00,45,F
0,10,00,B5,09,6D,30
2440 data 67,FA,52,84,D2,BC,00,00,00,0
9,B2,B9,00,00,0B,F8
2450 data 66,E2,52,83,28,03,52,84,D0,B
C,00,00,00,09,22,00
2460 data D2,BC,00,00,00,09,B0,B9,00,0
0,0B,FC,66,C6,4E,75
2470 data 43,F0,00,00,45,F0,10,00,49,F
3,30,00,4B,F3,40,00
2480 data 74,08,1A,11,12,D2,14,C5,51,C
A,FF,F8,1A,14,18,95
2490 data 1A,85,60,AE,3F,05,2F,0D,3F,3
C,00,3D,4E,41,50,8F
2500 data 4A,40,4E,75,2F,0D,2F,05,3F,3
9,00,00,0C,12,3F,3C
2510 data 00,3F,4E,41,DF,FC,00,00,00,0
C,4A,80,4E,75,3F,39
2520 data 00,00,0C,12,3F,3C,00,3E,4E,4
1,58,8F,4A,40,4E,75
2530 data 1B,45,1B,66,1B,59,32,20,1B,7
0,20,20,20,44,45,53
2540 data 4B,20,4D,41,4E,41,47,45,52,2
0,20,BD,20,31,39,38
2550 data 37,20,43,68,61,72,6C,65,73,2
0,46,2E,20,4A,6F,68
2560 data 6E,73,6F,6E,20,20,20,20,20,2
0,20,20,20,20,20,20
2570 data 20,20,20,20,20,20,20,20,20,2
0,20,20,20,20,20,20
2580 data 20,20,20,20,20,20,20,20,20,2
0,1B,59,32,4F,7C,20
2590 data 20,20,20,20,20,20,20,20,20,2
0,20,20,20,20,20,20
2600 data 20,20,20,20,00,00,00,00,00,0
0,00,00,00,00,00,00
2610 data 00,00,61,6D,70,6D,1B,59,34,2
1,1B,4A,1B,59,34,35
2620 data 41,72,72,6F,77,20,6B,65,79,7
3,2F,4D,6F,75,73,65
2630 data 3A,20,4D,6F,76,65,20,70,6F,6
9,6E,74,65,72,1B,59
2640 data 35,33,52,65,74,75,72,6E,2F,4
C,65,66,74,20,42,75
2650 data 74,74,6F,6E,3A,20,53,65,6C,6
5,63,74,20,61,63,63
2660 data 65,73,73,6F,72,69,65,73,1B,5
9,36,3D,43,6C,72,20
2670 data 48,6F,6D,65,3A,20,43,6C,65,6
1,72,20,61,6C,6C,20
2680 data 73,65,6C,65,63,74,69,6F,6E,7
3,1B,59,37,35,45,73
2690 data 63,2F,52,69,67,68,74,20,42,7
5,74,74,6F,6E,3A,20
2700 data 45,78,69,74,00,1B,59,34,20,1
B,4A,1B,59,35,31,57
2710 data 68,69,63,68,20,64,72,69,76,6
5,20,68,6F,6C,64,73
2720 data 20,74,68,65,20,61,63,63,65,7
3,73,6F,72,69,65,73
2730 data 20,28,1B,70,41,1B,71,20,6F,7
2,20,1B,70,43,1B,71
2740 data 29,3F,00,1B,59,34,21,1B,4A,0
0,1B,59,34,20,1B,4A
2750 data 1B,59,35,3B,52,65,6E,61,6D,6
9,6E,67,20,61,63,63
2760 data 65,73,73,6F,72,69,65,73,2E,2
E,2E,00,1B,59,00,00
2770 data 00,1B,43,1B,43,1B,43,1B,43,1
B,43,1B,43,1B,43,1B
2780 data 43,00,20,1B,43,1B,43,1B,43,1
B,43,1B,43,1B,43,1B
2790 data 43,1B,43,20,00,1B,59,00,00,0
0,1B,70,00,1B,71,00
2800 data 1B,59,34,20,1B,4A,1B,59,35,3
9,43,6F,6E,74,69,6E
2810 data 75,69,6E,67,20,62,6F,6F,74,2
0,70,72,6F,63,65,73
2820 data 73,2E,2E,2E,0A,0D,0A,0D,00,5
C,41,55,54,4F,5C,44
2830 data 45,53,4B,4D,47,52,2E,50,52,4
7,00,1B,59,34,20,1B
2840 data 4A,1B,59,35,3A,4E,6F,20,61,6
3,63,65,73,73,6F,72
2850 data 69,65,73,20,6F,6E,20,64,72,6
9,76,65,20,58,2E,2E
2860 data 2E,00,58,3A,5C,2A,2E,41,43,3
F,00,58,3A,5C,20,20
2870 data 20,20,20,20,20,20,20,20,20,2
0,00,58,3A,5C,20,20
2880 data 20,20,20,20,20,20,20,20,20,2
0,00,2E,41,43,58,2E
2890 data 41,43,43,4A,61,6E,20,46,65,6
2,20,4D,61,72,20,41
2900 data 70,72,20,4D,61,79,20,4A,75,6
E,20,4A,75,6C,20,41
2910 data 75,67,20,53,65,70,20,4F,63,7
4,20,4E,6F,76,20,44
2920 data 65,63,20,0A,16,12,08,00,00,0
0,00,16,12,08,3C,10
2930 data 14,DA,08,26,32,0C,0A,08,22,0
8,0E,08,18,0C,16,06
2940 data 06,06,06,06,0C,14,06,06,12,0
6,06,18,06,06,04,06
2950 data 04,12,1A,08,1A,0A,06,06,08,0
8,06,0A,0A,08,06,24
2960 data 1E,0C,10,0A,12,0C,12,08,08,0
E,0C,08,0E,0A,06,06
2970 data 0A,16,0E,08,0E,08,08,06,04,0
6,0A,08,0A,06,06,06
2980 data 08,3E,06,1C,14,3E,1C,10,06,1
```

```
8,12,08,06,06,10,06
2990 data 08,06,04,08,08,06,08,0E,0C,0
A,0C,06,0E,06,0A,0A
3000 data 0C,06,12,08,06,12,06,20,20,0
6,18,0C,08,0C,10,22
3010 data 06,08,06,0A,04,06,04,06,06,1
0,0C,0A,08,12,08,08
3020 data 0A,38,0A,0C,06,0A,20,1C,42,1
6,00
3030 data *
```

●

ST CHECKSUM DATA.

(see page 32)

```
 100 data  924, 948, 117, 614, 503, 21
3, 410, 427, 14, 109, 4279
 190 data  645, 357, 621, 497, 871, 72
0, 734, 865, 796, 895, 7001
 1080 data  769, 887, 957, 5, 828, 981
, 958, 894, 854, 852, 7985
 1180 data  819, 831, 855, 973, 890, 8
84, 755, 42, 93, 747, 6889
 1280 data  357, 821, 779, 607, 738, 9
63, 754, 698, 719, 765, 7201
 1380 data  671, 754, 768, 766, 699, 7
68, 746, 969, 796, 959, 7896
 1480 data  711, 895, 819, 804, 798, 7
56, 777, 640, 839, 847, 7886
 1580 data  714, 701, 823, 812, 970, 7
21, 785, 749, 783, 706, 7764
 1680 data  851, 818, 804, 707, 792, 7
13, 852, 910, 796, 778, 8021
 1780 data  767, 852, 808, 824, 759, 8
10, 819, 977, 773, 39, 7428
 1880 data  734, 172, 854, 872, 650, 8
74, 770, 938, 827, 831, 7522
 1980 data  853, 43, 886, 678, 649, 80
9, 767, 809, 873, 782, 7149
 2080 data  875, 780, 744, 942, 731, 6
56, 845, 880, 873, 937, 8263
 2180 data  998, 954, 6, 908, 654, 875
, 662, 640, 598, 790, 7085
 2280 data  728, 775, 745, 762, 904, 8
16, 78, 756, 992, 859, 7415
 2380 data  861, 807, 844, 838, 715, 8
23, 979, 817, 983, 851, 8518
 2480 data  990, 894, 783, 895, 800, 7
29, 791, 816, 690, 573, 7961
 2580 data  674, 575, 497, 777, 894, 8
79, 833, 853, 857, 855, 7694
 2680 data  880, 858, 779, 860, 833, 7
72, 755, 861, 816, 756, 8170
 2780 data  742, 672, 855, 886, 830, 7
78, 871, 854, 799, 641, 7928
 2880 data  652, 784, 830, 816, 626, 7
34, 598, 653, 694, 657, 7044
 2980 data  669, 682, 598, 589, 203, 1
94, 2935
```

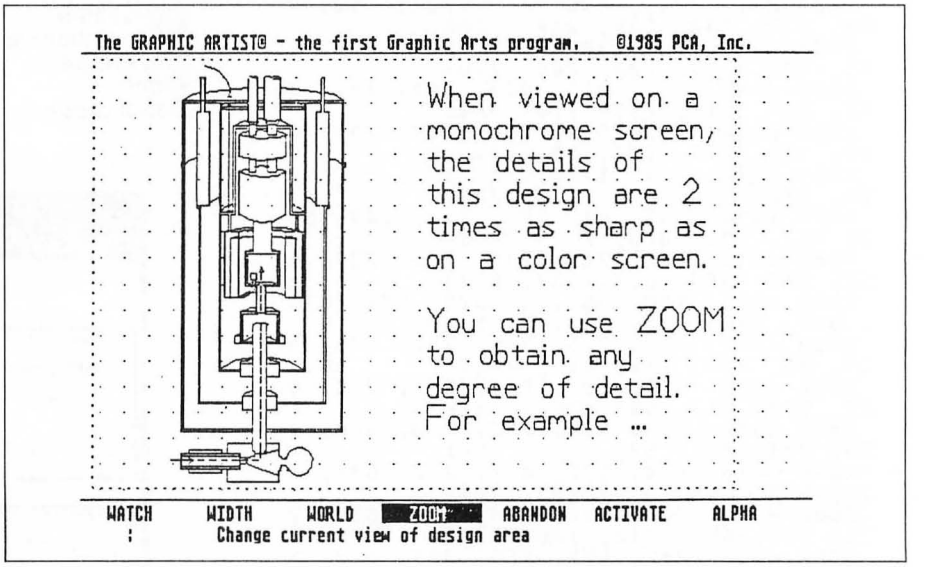
●

TUTORIAL

The *Graphic Artist* Tutorial
PART 1

Introduction—what does it do?

by Deborah Elder and
Peter Naleszkiewicz



You may have seen a sophisticated grey binder sitting on your Atari ST dealer's shelves. **The Graphic Artist**, you read. Sounds like some kind of graphics program. I *would* like to do some graphics, especially if I could add text to my designs. Could **The Graphic Artist** be what I need? Is it a type of paint program?

Computer Aided Design.

No, **The Graphic Artist** (**TGA**) is very different from paint programs. Paint programs are dot or pixel oriented. The purpose of paint programs is to light up dots on the screen (pixels), to create colorful video pictures. Paint programs output on dot-matrix printers using only the screen's resolution (often called a "screen dump"), so the printout looks blocky.

The Graphic Artist is a Computer Aided Design (CAD) and DeskTop Publishing (DTP) program in the same environment. The object of **TGA** is to create printed output. Additionally, the design and printouts are to scale, and corrected for aspect ratio (aspect ratio is a measure of squashedness; correcting for this means printed squares are really square, circles are never eggs). The screen image is not the product, but only a tool to help you create the printed output. Printouts from **TGA** are typically ten to twenty times *sharper* than the image on the screen!

TGA is "object-oriented." This means you draw items which have locations, sizes, rotations, etc. Examples of objects include a line with endpoint positions, or a circle with a center point position and radius size. **TGA** supports a very powerful type of object called a "symbol." A symbol is a drawing all by itself, which is used over and over again in other designs, in any size, and at any degree of rotation. You can overlay symbols in a design, and each symbol will remain intact. For example, you can draw a tree and then draw a bush at the base of the tree (which covers the base of the tree). If you remove the bush, the base of the tree is still intact. If you save the tree as a symbol, you could put the tree in another drawing, for example, on a hill at one-third the original size and at a rotation of 29 degrees. If you develop a library of symbols, complex design is simplified to the arrangement of items.

Plus desktop publishing.

The CAD capabilities are further enhanced by desktop publishing capabilities. You can freely combine text in multiple font styles, in any size or rotation, along with your CAD drawings. A mini word processor is included for entering text. You can have paragraphs of text that automatically word wrap, add attributes such as variable amounts of horizontal

(continued on page 50)

Inside Electronic Arts

An interview with Trip Hawkins—suddenly a leading ST producer.

by D.F. Scott

It's easy for two computer enthusiasts to find themselves talking about the state of the computer industry until their voices tire. When Trip Hawkins, President of Electronic Arts, was interviewed for ***ST-Log****'s exclusive on EA's acquisition of Batteries Included, we took the opportunity to spend an hour with him talking about other things. . .*

TH: We're developing products for the ST *and* Amiga; we're really not focused solely on any one machine, although I continue to feel disappointed in what I see as the market potential for ST software. Now that we're selling ST products, some of my worst fears have been confirmed, in terms of the size of the market. Ultimately, we're not trying to run a charitable organization. If we can't sell enough products to recoup our expenses, then that's going to diminish the degree of support we're going to have.

I'll give you an example: **Skyfox** was developed by the same company for both the Amiga and the ST—and is virtually the same code on both machines—buyers should like or dislike it to the same extent on either machine. Yet the Amiga version sold over 20,000 units, and the ST version sold about 6,000 units.

DF: *Couldn't the reason be that, when you released the Amiga version of* ***Skyfox****, there wasn't a heavy degree of competition; whereas, when you released* ***Skyfox*** *for the ST, there was* ***Harrier Strike Mission, High Roller, Flight Simulator****—all already selling well [enough] that you had a sufficient degree of* competition, *where you did not have a corner on the market?*

TH: I can give you lots of other examples, if you like, but we don't have a single incident so far in which an ST product has sold as well as the Amiga version.

DF: *To broaden the scope of my question: could that be because there's a* competitive *market in the ST right now; whereas Electronic Arts has proven itself pretty much a leader in the Amiga market?*

TH: I think there are probably a lot of different explanations, but let me put it this way: if you were running my company, would you want to develop more ST products, or more Commodore 64, Amiga, IBM PC products? Aside from the emotional content of the issue, just from a business standpoint, what do you think you could justify? Ultimately, you've got to justify it based on the numbers.

DF: *It would be based on the numbers, and it would be based on what currently* isn't *out there to a sufficient degree. I wouldn't want to print a copy of somebody else's program, almost to the letter.*

TH: I agree with that. Some of the products we're going to be bringing out for the ST this year [cover] market applications we don't think have been well covered in the ST market. But you have to be selective. We've received a lot of complaints in the past, from ST customers who don't understand why we don't produce our entire catalog—from our entire history—for the ST, and think that we're wimps for not doing that. It's just that we don't have the people to manage that kind of activity level; and software development's a very risky business. . .The biggest problem is that not that many Atari STs have been sold to consumers who want to spend a lot of money on entertainment.

I think that some people may have misunderstood, taking it as a kind of *personal* thing, that we said, "Look, we like this Amiga machine, and the reason we like it is because of the new architectural features that they put in the machine, which we thought augured well for the future of home computing." Of course, subsequent to that, Atari spent a lot of time talking about introducing a blitter chip. So obviously, some of the things that they did *did not* fall on deaf ears.

Since then, we've also had the announcement of the CD-I technology. I'm very excited about CD-I. See, I think we're going to continue to have difficulty making good software as long as there are many, many different models of computers, each of which is fairly expensive and only manufactured by one company, each of which is different in operating characteristics—so that we have to reengineer the software each time we make it. Rather than having half-a-dozen different 68000-based computers on the mar-

ket, if you look at every other mature electronic media business, eventually the industry settles down to a media standard, so that the software producers can efficiently make the software.

When you go to buy a hi-fi or a video player, you know that there are a lot of manufacturers making that product; and that, when you turn it on, there's going to be a tremendous variety of software you can use for it. We've been hamstrung, as software companies; it's really impractical to develop a lot of software when machines are so different from one another that the installed base of each machine is so small.

I think, unfortunately, there's been a tendency among the American companies to prefer to compete, to have the attitude, "Well, you know, I'd rather have a big part of a small market, and beat the other guys' brains out, than have a smaller part of a really *gigantic* market." They may be able to get away with that in the short term, but probably won't in the long term—eventually, they have to go find a ditch market they can survive in. Frankly, that's what happened with both the Amiga and the ST.

I think the Amiga's primary success has been as a creativity tool for creative professionals and hobbyists. I think if you look at where the ST is *really* selling on a worldwide basis, its dominant market is as a monochrome small-business and desktop-publishing machine in Germany. That's where at least half of the STs in the world are located.

DF: *So your impression is, if you really wanted to do business with 50 percent of the ST market, you'd need to go overseas or open up an overseas division?*

TH: Right. We've just started a subsidiary in England, and we will be manufacturing and selling some ST products over there. We feel that some of the products we market over there for the ST will come to the U.S. as well.

Our primary interest, in the long term, is in the *consumer* market, and consumer markets need standards. There are always going to be people who'll say, "Gee, I don't like standards, because you have to slow down the pace of technology." But . . . it's very frustrating for me to see a computer owner buy a brand new computer, then discover there's not a lot of software for it and blame the software companies; because, as far as I'm concerned, the hardware companies are making their bed, and they have to lie in it.

Initially, when the Atari ST came out, my attitude about it was, "Look, does the world really *need* another *slightly* different computer architecture?" With the Amiga, we at least felt there was enough of a step forward there in architecture, that this was a good foundation to learn from as preparation for the future.

> *"Does the world really* need *another* slightly *different computer architecture?"*

In the meantime, I think it's going to be very difficult for the Amiga, and the Atari ST, and anybody else—even for Macintosh—as a home machine. None of those machines really has a chance to go in and become a dominant consumer machine, because they're trying to do it all by themselves. If one of those companies would say, "Look, why don't we get together? Bring in a standard, license it, and let's get lots of manufacturers to support it," then they'd have much better success at getting software companies behind them.

DF: *Let me play devil's advocate with you on the issue of standardization. You have a dominance of the PCs and PC clones, in which you have so much market* saturation, *so many companies producing basically the same machine, that unless you're a small company, you hardly have a chance to make a substantial profit. In the case of IBM itself, which developed the technology—or at least bought the technology from other people—the originator company puts itself in a position, through standardization, where it is not able to profit from its own machinery.*

TH: I don't think that's a good analogy for the consumer market. What's going on with PC clones in the home right now is kind of an aberration. Their announcement [of the PS/2] confirms the fact that IBM doesn't want to make it easy for people to make clones of their own machines. If you look at the world ten years from now, I'd be willing to bet that the IBM Personal Computers are primarily in the business market, and you won't find that either of the IBM brands, or clones, are at home, because it won't be possible to make clones.

I think what's going on right now is that consumers are very frustrated by the lack of a standard. Because of the strength of IBM in the business market, a lot of consumers have made a mistake, really, in thinking that what they should get for the home is the PC. I'm trying to put out entertainment and creativity software [for] everybody who buys a machine like that. In some sense, I'm disappointed, because they're not getting the kind of machine that has the audiovisual features that I'm a real believer in.

DF: *I have a little saying, which I repeat frequently: "Where there is standardization, there is also mediocrity." Usually this is the case if you're somebody from ANSI trying to make a standard programming language; it's the same if you're trying to make a standard computer format where you have absolutely no fluctuation, and in a sense no* creativity, *which is your market.*

TH: But on the other hand, look at the audio market. Way back in the 30s and 40s, there was 78 rpm. So CBS, in 1948, announced the LP format. That was a pretty exciting breakthrough, when you look back on it.

RCA didn't want CBS to be too successful with that format, so they introduced the 45 within six months. The effect was that the record industry went into a four-year slump, because everyone was confused about what to do.

Twenty years later, Philips introduced the LaserVision technology, which was a complete flop. You might remember RCA came out with another flavor of movie disk—that also flopped. Then Sony came out with Betamax video; they picked Betamax over VHS technology, and the guys who developed VHS went to work over at one of the Matsushita companies. Matsushita decided to license [VHS] to a lot of other companies. So, as a consequence of all that licensing, there was much more competition, and the price of VHS video dropped dramatically.

Meanwhile, if you look at what happened with the compact disk: Philips and Sony got together in 1982, and they could look back on a lot of history at that point and see: number one, what had happened in the record business when the LP had been introduced; number two, they had their own personal experiences failing to establish new markets, even with pretty nifty technology. Having learned those lessons, they said, "Maybe what we need to do is to get together and work together as an industry to establish [CD technology], rather than having trivial variations in it."

Frankly, technologically speaking, is there really much of a difference between a Macintosh and an ST? There isn't; but, instead, you have this huge software incompatibility in the marketplace. You could have several different models of compact disk out there, but they would all be based on the same kind of optical disk technology, the same digital processing rate—and yet they'd all be different, and the whole industry would suffer for it.

DF: *Well, in the analogy you gave, with record players and video recorders, you're talking basically about* output *machines, and the selection of a specific, universally acclaimed format for each of these output machines. The computer is an* input *and output machine.*

TH: Yes, it's much more complicated to determine a standard for that.

DF: *In my opinion, the consumer does not want to be* forced *into data or creative input in a standardized format.*

> *"...technologically speaking, is there really much of a difference between a Macintosh and an ST?"*

TH: But that's not necessary. If the hardware *player* is standardized, there's a lot of variety that can be applied.

Let's put this into the context of the Atari ST. If you like the ST, then you would like it *even better* if there were ten companies making it, and the hardware sold for less than a hundred dollars, and there was ten or twenty times more software available, at much lower prices.

DF: *I can understand the twenty times more software available; but why would I want ten companies making it?*

TH: Because now they're going to compete with one another around a standard; which means they're going to try to offer better service, better support...

DF: *And, in the end, there are going to be—after a couple of years— thirty-three hundred companies making it, because the technology has been practically made public domain, and none of the companies will be able to profit from their own material.*

TH: That's not necessarily true. I think if you look to the compact disk, there's a renewed excitement in the audio and record companies because of it.

Literally, the whole issue boils down to the fact that if Atari can get out there and sell fifty million Atari STs, then it would become a *de facto* standard, and lots of people would make software for it. But there's no company in the consumer market that has that kind of power.

DF: *There's no* one *company because we have so many of them producing the* de facto *standard PC.*

TH: But IBM's not really a consumer company. In the *consumer* market, nobody really has the power to waltz in and develop the market to the point where they become a *de facto* standard.

DF: *So you think, five years from now, compatibility won't be an issue?*

TH: Well, *IBM*-compatibility. You have to separate in your mind the computer *hobbyists*, who particularly bought a lot of the Atari STs initially, who like to do their own programming—and the Atari ST's a great product for them, because they can get a 68000-based machine for very little money. But that doesn't necessarily mean it's a great machine to develop software for; because, if there's a limited population of the machine and the company manufacturing it doesn't have the market power to get it to become a *huge* seller, then it's never going to become a great vehicle for software applications, unless you happen to make one of the handful of applications which is going to achieve a very high penetration.

So, in the case of Macintosh, it's turned out that if you make desktop publishing software, you can sell a great deal of it; but if you make anything else, you're going to go broke. There isn't any software other than desktop publishing software that has done really well for the Macintosh. Our Macintosh software sales have been very disappointing—that's in spite of the fact that more than a *million* Macintoshes have been sold. So it's not because there aren't that many machines, it's because they're all being used for *one* particular kind of application.

We depend on a customer who owns a computer to think of it as a *media machine*. Unlike television or video, they're not going to settle for doing just one or two things; they want it to do a whole variety of things.

DF: *You think that the general consumer is going to use a computer—especially a*

> *"[Commodore] didn't pay us* any *money to develop any of our Amiga titles."*

home-based computer—as an output *machine, as a VCR?*

TH: I'm not thinking of it as output-only; in fact, that's why I'm in the computer business. I like the fact that it's the only interactive medium. With television, you have a *passive* experience, you've got to *watch* what's going on. Maybe you're watching the Super Bowl, maybe you're watching Christopher Columbus in a drama; whereas, with a computer, you can play a game like **Seven Cities of Gold**, or you can play **Touchdown Football**—you can actually *be* there and be part of the action. The subject matter may be very similar to other media, but it's the *interaction* that makes it unique.

DF: *Computers are given attributes which are purposely unique, so that they cannot be copied, so that the* type *of output, the type of interactive experience achievable with an Atari ST, is going to be somewhat different than the type of experience achieved on an Amiga—even though the "engine" of both machines is the same.*

TH: That's right, and that's unfortunate, I think. If it happens at a fairly arbitrary level, then it's unfortunate. It's very shortsighted on the part of the hardware manufacturers to think, "We're going to introduce these somewhat artificial, arbitrary changes just to frustrate that other hardware company, just to frustrate the software companies that are supporting that other hardware company. We're going to try to get everybody to support us in a unique fashion, by going way out of our way." I think that's somewhat naive, don't you?

DF: *Couldn't it be more than just to frustrate the other company; couldn't it be so "we, and we alone, can profit from our own ideas?" Let's look at the software market; would you want somebody creating a* ***Skyfox*** *emulator?*

TH: If Atari was happy to just make hardware that would be sold as a programming tool and they wouldn't have to provide any applications software—hey, that's fine. Then they can profit from their own cre-

ation. If I were an Atari ST customer, I would want there to be a lot of software available for the machine.

DF: *But how can any one company, how can any group of companies, decide to produce one type of machine, and have that machine be as programmable—?*

TH: The same way that one company decides how to produce one kind of machine—by working cooperatively rather than against each other.

DF: *We'd have a whole bunch of companies producing basically the same thing.*

TH: Same as what we have with television and video.

DF: *Not really. With output machines, if you want a graphic equalizer, you can buy one, but you don't necessarily need one. Sound components sound different to different ears.*

TH: With PC compatibles, if you want a mouse you can buy one, an EGA display you can buy one.

DF: *Because of the open architecture—but, then again, those are the many different ways to get around the problem of standardized machines because people want their own custom machines.*

TH: It's still possible to do that, even if the computer architecture is standardized, if people like the thing to *look* a certain way, they want it *priced* a certain way. They want to have certain bells and whistles, or they don't want to have certain bells and whistles.

DF: *So you feel it's possible for all the manufacturers to get together and say, "At the basic heart of the unit, let's decide on some degree of equal ingenuity—we'll make this same product—but then allow proprietary expansion systems?"*

TH: That's exactly what happened with the compact disk, by the way. They call it the "base case"—they agree on a minimal standard. The way the rules work, if you're a licensee and you want to put the CD logo on hardware or software [you have to build the base case first]. Software companies know what the base case is, and they can make a product to, at a minimum, support the base case from all the different companies.

DF: *So you feel that, maybe five to ten years from now, we can have a kind of consortium of computer corporations?*

TH: CD-I is being put together right now, so it may be sooner than five years; but then, it may fail. It all depends on how good the machine turns out to be, how quickly it gets it to market, and what price it'll go for, and who supports it.

For companies like Atari and Commodore, personally, I think it'd be terrific if they support it; because Atari can say, "Let's build a machine that's an Atari ST, but one of the features of the Atari ST is that it's CD-I-compatible." Now, if somebody wants a really great computer, he gets the best of both worlds—he can use all the CD-I software and also his Atari ST software. He can do things with it as a computer that maybe the standard base case CD-I player *can't* do; but then he's got this whole world of other software that's opening up to him.

Here's the interesting problem: companies like Atari and Commodore are so accustomed to competing with each other in a much more hostile environment, that they tend to be too suspicious of one another to work together to make the industry grow. Personally, I think that's very unfortunate.

I'll tell you one of the most absurd rumors that floats around: that Commodore paid us a lot of money to develop Amiga software. They didn't pay us *any* money to develop any of our Amiga titles. We developed one demo for them, and they paid us something like $35,000—totally a break-even proposition for us for developing it. All the actual products we developed for the Amiga were totally on our own nickel. A rumor circulated that Commodore paid us a million dollars or two, and that's the only reason we did it. That's ridiculous.

What a lot of people forget about the Commodore 64 is the reason it succeeded: at that point in time, a lot of families and parents believed they should buy a computer for their kids, for educational purposes. The reason the Commodore 64 beat out the Atari 800, which was technologically superior, was on *price*.

DF: *That's one point I can* definitely *agree with you on.*

TH: So, suddenly, all these Commodore 64s are out there, and the reason a software business finally developed was that, suddenly, software companies could stay alive—because there were enough people who had the Commodore 64 that they could afford to develop software. It wasn't that people switched from developing the Atari 800 software to the Commodore because they didn't like the Atari 800—

DF: *Not because of dislike for the company; it was due to the larger user base.*

TH: Yes, basically, we couldn't make enough money on the Atari 800 software. I think, in terms of Jack Tramiel's attitude, he assumed that the same thing would happen all over again with the ST. Unfortunately, I don't think that the market conditions since then have made it possible for a new piece of hardware—from a company other than maybe IBM—to come out and have everybody immediately just say, "Okay, we're going to *assume* there are going to be several million of these out there in the future, so we're all going to develop a huge product line for it."

We are going to continue to develop ST products; but I just hope that people can have a realistic viewpoint about the problems that software developers have. //

D.F. Scott is an artist, writer, educator and programmer living in Oklahoma City. He is currently engaged in the study of quantum physics, computing and other ways in which elementary particles interact with each other. Otherwise, he fills infinite pieces of paper.

Step 1
LOOSE ENDS II

More help for the first-time computer user.

by Maurice Molyneaux

In our last episode, the author smugly jotted comments and explanations on a number of miscellaneous ST topics not included in previous articles. By the end, he realized he had bitten off more than he could chew—and still had 16K of material he couldn't use in the June issue! Tonight, watch to see if he can pull it off. Can he take the leftover text from last month and find enough other material to make a regular-sized article out of it? Stay tuned for the thrilling conclusion!

Okay, I've had my fun this month. A lot of disjointed and only vaguely related topics this time, so stand by for some rather sudden "plot twists" in the conclusion of this roller-coaster ride we call **Loose Ends**.

Accessory to a crime.

Desk accessories confuse not only beginners on the ST. I've briefly discussed them in the past (notably in the April 1987 **ST-Log**), but will go into more detail now. The facts: desk accessories *are* programs, but they *aren't* in ROM, nor are they part of the ST's operating system, despite the fact that they appear under the desk menu on the GEM desktop.

Accessories are loaded *only* from the disk in drive A, unless you have a program to make the OS read them from other drives (like a hard disk). They must be in the main (root) directory, or the system won't find them—so clustering them neatly in a folder won't help at all. Furthermore, while they are programs, accessories aren't quite the same animal as "runnable" applications, like those you run from the GEM desktop (.APP, .PRG, .TOS and .TTP programs). Double-clicking on (or opening) an accessory will not run it.

Desk accessories are run by GEM from its desk menu. A properly written one can be utilized while another program is running, then "put away" without disturbing or damaging the program—just as you might use a pocket calculator to do some math while writing a paper, then put it aside when you're ready to continue writing. This is a simple form of "multitasking," which means that your computer is handling more than a single task at a time (in this case, running a program and an accessory simultaneously).

When you start up or reset (boot) your computer, one of the first things the ST does (after checking for the presence of cartridges and an AUTO folder on the disk in drive A) is search the root directory of the disk in drive A for any desk accessories. Accessories are programs identified by the filename extension .ACC (as in CONTROL.ACC), and your ST will load into memory (RAM) all the files with this extension, up to a maximum of six (which is all GEM currently allows.)

Accessories which use up two "slots" under the desk menu (such as the control panel) use up both of those slots, allowing you only four other accessories. Once loaded, they'll remain in memory until the ST is switched off or you reset it (changing resolutions from the desktop causes a reset, though not as thorough a one as pushing the RESET button does). They stay quietly out of the way, until you click on one of their names under the desk menu, and it's called into action.

The accessory will usually display a window or dialog box. Accessories which use GEM windows can often be shuffled just like other windows on the desktop or in a program, meaning that you can leave the accessory in view and call it into action merely by clicking on it, then go back to your program (or desktop) just by clicking on one of the program's windows!

Those which appear as dialog/alert boxes are nonrelocatable, and will not allow you to access other windows or accessories until you close them. When you're finished working with the accessory, clicking on its EXIT button or "close window" box (located, as usual, in the upper left-

hand corner) will make it disappear. However, it isn't gone or erased, just put back into your ST's "pocket" for the moment. You can call it back through the desk menu again.

To sum up: accessories are programs loaded at start-up/reset time, remaining in and consuming memory (usually small amounts), which can be called into action almost anytime the desk menu is available, whether you're on the desktop or deep in a word processor. Programs which do not display the menu bar (like **NEO-Chrome**, or any .TOS or .TTP application) will obviously not allow you to use accessories while they're running, but most programs which allow access to the menu bar (like **DEGAS Elite**) will.

Accessories are specially designed applications that operate differently from other programs. You cannot transform an accessory into a .PRG application just by renaming its extension from .ACC to .PRG. This applies in reverse, too: you can't rename a .PRG, .TOS, etc., file to .ACC and expect it to work as an accessory. Doing either of these things will probably cause your ST to crash.

If clicking on a given accessory's label under the desk menu does nothing, don't panic. You may have loaded one of the accessories that does a special job independent of the user, and that announce their presence on the desk menu just to remind you they're present. Examples of such user-inaccessible accessories (did that make sense?) would include some print spoolers, icon loaders, or even RAMdisk-creating accessories.

Finally, a small warning. **ST-Log** West Coast Editor Charles Johnson has informed me that one of the biggest mistakes a programmer can make when writing an accessory is to have it read an outside resource file from a disk. This is because, every time you change resolution, the system reloads that accessory and the resource file, but GEM doesn't clear the previously loaded resource out of memory. So, every time you do this, more and more of your precious RAM will be gobbled up by multiplying copies of resource files, until *bombs away!* The only way to prevent this with such an accessory is to reset or reboot your machine, and use different disks to boot in low or medium resolutions. If you have only a monochrome monitor, you don't have to worry about multiple resolutions. And, if an accessory requires an RSC file, you must have that file available to the accessory if you plan to use it. You have no choice.

Missing: 8 to 16K of RAM.

Last issue, when I said "a 512K 520ST," I'll bet some of you were a bit confused. "Isn't the 520ST supposed to have 520K of RAM?" you ask. Well, no. And a 1040ST has 1024K of RAM, not 1040K. Next I'll be telling you there's no Tooth Fairy.

Were you robbed? No, Atari didn't lie to you about your ST's memory. It's just that people assume the numbers on the machines refer to total kilobytes (K) of RAM in them. Nope. The numbers are derived from the number of free *bytes* of RAM in the ST. A stock 520ST has 524,288 and a 1040ST has 1,048,576 bytes of RAM available. If 1 kilobyte were indeed 1000 bytes, as the *kilo* would seem to indicate, a 520ST would have approximately 524.3K of RAM and a 1040ST 1,048.6K.

But a kilobyte is *not* 1000 bytes; it's 1024 bytes, so 524,288 divided by 1024 equals 512. Atari apparently chose to base the numbers for their machines on the number of bytes, not kilobytes, dividing the total by 1000 and rounding the figure off to the next lowest 10s value. By one reckoning, Atari has given you *more* RAM (byte-wise) than the numbers would indicate. To be more accurate in the byte count, the machines should be called the 524ST and 1048ST, but these don't have the marketing zing of "five-twenty ST" and "ten-forty ST."

Resurrection!

We've all known the frustration of accidentally deleting a file. However, when you delete a file, all is not lost; it really isn't gone! You can unerase the file, if you have the proper tools.

What form of witchcraft is this? None, really. When you delete a file, your computer doesn't actually go to the disk and destroy the data in the file. What it does do is go to the disk's directory, where all the filenames are stored, and mark the file you want deleted as "ok for overwriting." This usually means the system changed the first character in the filename to a special symbol which indicates the file is no longer valid for use and access. The file's data is still there, just as it was, but the name has been pulled from the directory.

If you delete a file, you can usually recover it . . . *if* you attempt to do so *before* saving or resaving *anything* on that disk. Because the system has marked the deleted file as free space, new files can be written over the old file's data. So, if you save anything to the disk after deleting a file and before trying to recover it, you may find that parts—or all—of the deleted file are completely destroyed.

To restore a deleted file, you'll need a disk editing program or a special unerasing utility. Most good disk editors feature file recovery. Unerasing programs (like UNDELETE.PRG) are usually dedicated to that single task. If you don't have one, *get one*. These programs will usually work fine on floppies, but using them on hard disks is a risky business, indeed. I don't recommend it.

Unerasing files from a RAMdisk is usually impossible, because most RAMdisks just dump the deleted file from memory.

Shake, rattle and roll.

If you have an SM124 (monochrome) monitor, does the screen jitter, display black bars, or waver sickeningly when disk drives are accessed? If so, this is usually a hardware problem.

Inside your ST is a video controller chip called the "shifter." According to the owner of my local ST dealership, shifter chips are the most common hardware-related problem he sees with the ST. If you have this problem, take your machine back to your dealer and tell him what's wrong. He should, in most cases, only have to replace the shifter (this takes, maybe, ten minutes). If you purchased your ST via mail-order or a mass-market store, you'll have to try to return the computer (you need not return monitors or disk drives with it) for a replacement.

For these reasons, I recommend purchasing a machine from a local Atari dealership, because they can help you with your problems. A clerk in a Toys 'R' Us store, or a voice

on the phone at a mail-order company, is of little help when you have this kind of problem. You may pay more for your system from a local dealer, but those extra dollars buy you local support—and help when you find yourself in a jam. Consequently, if you bought your system or software somewhere other than a local ST dealer, you should be ready to pay for their help. After all, they aren't responsible for what you bought somewhere else.

Control, control, you *must* learn control.

The mouse supplied with your ST is fine for most programs that need an input device. Still, some games require the use of a joystick—if you play a lot of games, you'll no doubt need a joystick sooner or later. Fortunately, any standard joystick designed to work on the Atari 800/XL/XE series or the Atari 2600 and 7800 game systems will work just fine on an ST. But don't try plugging in any controllers for the Atari 5200 game system, even though a few may appear to have the proper plug. They're not wired the same way as the ST's ports, and could damage your computer if you try to use them. Joysticks for Apples and IBMs are also taboo. Joysticks used on a Commodore 64 are okay, but only because it uses the same joysticks as Atari's computers.

It would be nice to be able to use the paddle controllers, trackballs, light pens and touch tablets available for the 800/XL/XE line, but their wiring is different from what the ST requires; you can't use them as is. Still, if you'd like to use a trackball in place of a mouse, you *can* do it if you rewire the device. Instructions for such operations can be found on many telecommunications networks, including Delphi. Be forewarned, though: most operations of this nature require some knowledge of electronics, and often require soldering and modification best handled by the experienced.

BASIC gray.

Scads of people complain about ST BASIC. The most common gripe I hear is that, when using the edit window, the moment you start typing on a line, the text style changes to lightened and skewed, and some special symbols are replaced by others. Impossible to edit with this, you say? Easier to just completely retype the line from the command window? Nope. Here's how to fix things. When you run ST BASIC, as soon as you can type in the command window, enter the following:

```
POKE SYSTAB+2,0
```

The edit window will now behave itself and leave your lines readable as you edit. Of course, you'll have to enter this command every time you run ST BASIC, but it's an easy fix that can eliminate at least one of the program's many frustrations.

Another fine point about ST BASIC: you can't really leave the edit window just by clicking on another. You must either click on "Exit Edit" under the edit menu or press the F10 key, which will add the edited lines to your program. This will put you back in the command window. The reverse is also true. You must type *edit* and RETURN from the command window or click on "Start Edit" under the edit menu.

Don't ask me about **Logo**, I'm about as fluent in it as I am in Swahili!

Squash and stretch.

The above subtitle has more to do with the principles of character animation (as in a Disney or Warner Bros. cartoon) than anything else, but, my poetic license unrevoked, it'll work here too. Let's talk a bit about how to compress and uncompress data for your computer. . .

On second thought, my explaining the principles involved in such "bit-crunching" would probably be somewhat overwhelming, so I'll skip the techie aspect and just give you the gist of it. Computers store data in a variety of arrangements, and, while most methods of data clustering are effective for a given program's purpose, there are ways to further compress or "squeeze" files and programs, making them anywhere from 20 to 60 percent smaller than before.

In theory, this would make it possible for you to put lots more data on your disks, but, once again, I must engage in "the slaying of an original, beautiful hypothesis by an ugly fact" (Thomas Huxley). The trouble is that your computer and its programs can't usually use a squeezed file. "Great," you moan. "What good are compressed files if you can't use them?"

Oh, you can't usually use them squeezed, but you *can* unsqueeze them for use. Still can't see the value? Okay, suppose you need to send a lot of data to someone on one double-sided floppy. The disk can handle a maximum of about 800K of data, and your data adds up to 900K. You don't have any other blank disks, nor do you want to send out more than a single floppy. By squeezing the files, you can get them all on one disk, and the recipient can copy the squeezed files to other disks and unsqueeze them.

Still not good enough? Okay, you're using your modem on Delphi and find a neat game. In normal "usable" form, the file is 50K, but squeezed it's only 33K. You're paying to be on Delphi, plus possible long-distance charges. To download the squeezed file takes a lot less time (and saves you money) than if the same file were its normal size.

See the benefits? There are a number of compression utilities available for use on the ST, but I'll talk only about the most common ones. The first are two TTP programs called SQUEEZE.TTP and UNSQUEEZ.TTP (or a variation on these names), which have been commonly in use since 1985. SQUEEZE reads a file and writes a compressed copy of it under a new filename identical to the original—except that the file's extension is modified to contain a Q as the second letter, denoting its "squeezed" state. Thus, DIANE GAW.NEO would be compressed as DIANEGAW.NQO.

UNSQUEEZE reads squeezed files, then writes unsqueezed versions of them. Since these are TTP programs, you'll have to specify the target filename in the PARAMETERS dialog box which appears when you run them (see **Cracking GEM** in the April 1987 **ST-Log** (issue 13) for more on TTP programs).

Another form of file compression, and one gaining prominence in the ST community, is "archiving." A program called ARC.TTP is in the public domain and is used for archiving and unarchiving disk files.

Archiving data is similar to using one of the squeeze pro-

grams, but with one major—and important—difference. SQUEEZE can only compress single files, while ARC can compress multiple files into a single new file, to which you can add files—or from which you can extract one, a few, or all the files! This is great for BBS systems and such, where all related files for a given application can be ARCed together so users can get everything they need by downloading a single file. I obtained **ST Writer** version 1.70 from the Atari Base BBS, downloading a single ARCed file which contained the program, printer configuration setup program, printer data file, reference guide and some template files. When downloading an earlier version of the program many months earlier, I had to download each of the related files separately, because no one had ARCed them.

Now, while ARC.TTP is a wondrous thing in many ways, as a TTP application it's also tough to use. The commands are cryptic and complex, and confuse many beginners. If you obtain a copy of ARC.TTP and really want to get some use out of it, then I suggest you use Charles Johnson's **ARC Shell II** utility, which appeared in **ST-Log**'s issue 13.

This shell (see below) gives you a simple GEM interface to the program, freeing you from having to deal with mysterious commands in a TTP parameters box.

For those uninterested in details, who only want to unarchive files, a program called ARCX.TTP does just that. You can extract several or all files, but the easiest way to use it is to make it an installed application (see **Step 1** in **ST-Log** 14) with ARC as the document type, then just double-click on an archived file from the desktop, to de-archive it.

Shells.

You can't put these to your ear and hear the ocean, but they're useful, nonetheless. Simply put, a shell is a program which surrounds or covers another program, providing a different interface than the program it covers.

For example, GEM could be called a shell over TOS. To use TOS would require learning many difficult type-in commands. GEM interposes itself between you and TOS, taking, for example, a cryptic error code from TOS, but presenting you with a clear English-message alert box. It will interpret your mouse operations and tell TOS what that means in its own language. Shells generally make a program easier to use, or customize it for certain tasks. Charles Johnson's **ARC Shell II** (mentioned above) is an example, in this case interposing itself between the user and the ARC.TTP application, to make it simpler to use.

Exit, screen left.

Well, that's that. Next issue, it's back to more of a single topic of discussion. Personally, I'll be glad, because all this bouncing around was giving me motion sickness! But, before I go, here are some more. . .

Addenda.

More clarifications and corrections on past **Step 1**s.

—In **Soft Wares**, I stated that, when you're using a color monitor, any programs run from the AUTO folder would be run in low resolution. One user expressed disappointment at not being able to run **ST Writer** from the AUTO folder, so I offer this solution: in the public domain there's an application called MEDRES.PRG, which, when run, puts your ST in medium resolution (but not, of course, if you're using monochrome). To use this in an auto-boot situation, create a new AUTO folder on a disk, put MEDRES.PRG in that folder, and *then* copy into the folder the utility you wish to run. You must put MEDRES.PRG in first, because TOS runs the programs in AUTO in the order they appear on the disk. Whichever was copied first will run first. Don't take the order of items in a window as an indicator, as these are sorted by GEM and are *not* displayed in the order they appear on disk. MEDRES.PRG can be found on most BBS systems with an ST download section, or obtained through a users' group.

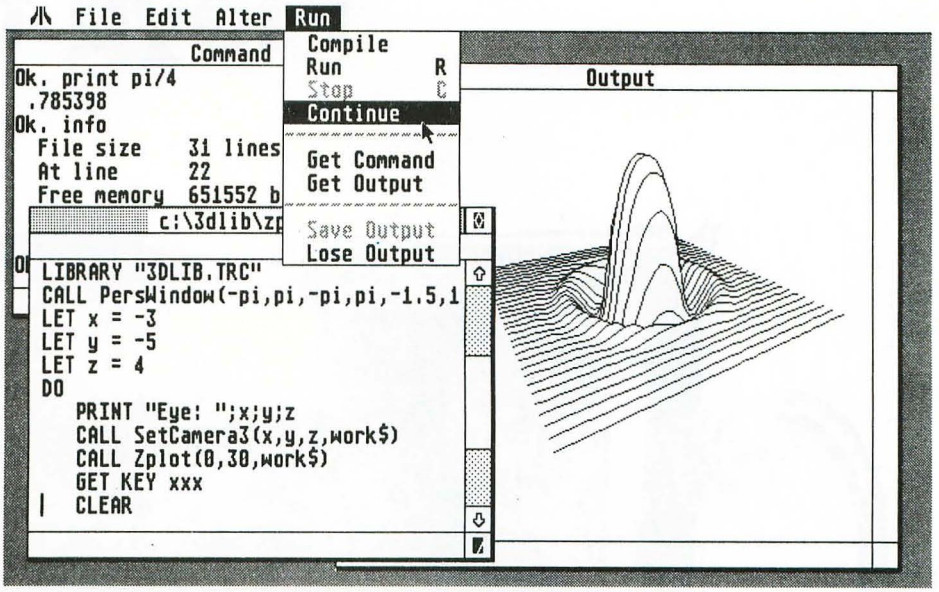
—In **Loose Ends 1**, I stated that you can't normally use a disk copy function to copy the contents of a "real" disk to a RAMdisk. I mentioned copying all the files, but forgot to state that you could drag the icon for the source disk and drop it into the *window* of the RAMdisk, rather than on the RAMdisk icon. This will result in the system file-copying the contents of the floppy to the RAMdisk, whereas, if you dragged the floppy icon to the RAMdisk icon, you'd be attempting a disk copy—and the system would probably inform you it can't do that.

—I was a little late getting the CUSTOMGEM.DOC file (mentioned in **Customizing the GEM desktop**) into the Atari Users' Group SIG on Delphi. It's been there since late June, in the ST Programs database, and can be located by searching for the keyword *CUSTOMGEM*. If you tried before and couldn't find it, sorry. It's there now, I promise! ///

---

*Allergic to all things Commodore, Maurice Molyneaux is an author and artist who—when not writing for **ST-Log**—continues to struggle with a recalcitrant 8-year-old science fiction novel, paints, illustrates and uses his ST for "every conceivable task." His interests include classic cel animation as well as the computer variety, and he draws the meanest "Star Trek" pictures on microcomputers. His Delphi username is MAURICEM.*

Make Your Own Magic.



Speed, great graphics and a little bit of magic: that's what you expect from your Atari® ST. And that's what you get from programming in True BASIC.

It's a structured language that's easy to use. A compiler, editor and debugging tools rolled into one. It's the latest from Kemeny and Kurtz, the inventors of BASIC.

If you crunch numbers, you'll appreciate floating-point math that's very fast. Support for large, dynamic arrays, and built-in syntax for matrix algebra.

And True BASIC makes graphics easy. Define your own coordinate schemes and color palettes. Plot entire arrays with a single statement. Apply built-in 2D transforms. Or use the 3D Graphics Library to put some depth in the picture.

You can change line styles, add pull-down menus and get at all the GEM and AES routines. But if portable code is important, there's True BASIC for the PC, Macintosh™ and Amiga® as well.

True BASIC lets you build your own libraries. They can be separately compiled, in BASIC, C or assembly. Debug with breakpoints and immediate mode. Visually trace program execution. Create macros and script files to complement True BASIC's fantastic mouse-based editor.

But if you're just learning, True BASIC will coach you with on-line HELP and friendly error messages. If you need more, we have textbooks and videotapes.

Find out why thousands of engineers, developers, and students use True BASIC to make a little magic. Visit your Atari ST dealer, or call us today at **1-800-TRBASIC**.

Now Shipping!

Language System	$ 79.95*
Runtime	79.95*
Both	149.95
Libraries	49.95 each

* Special introductory price

LANGUAGE
- multi-line, user-defined functions
- local, global variables
- SELECT CASE, DO/LOOP
- recursion, modules
- GKS graphics, 2D transforms
- access to GEM, AES features

COMPILER
- compiles to fast, compact b-code
- parameter type-checking
- optional LET statement
- optional declared variable names
- Runtime package creates stand-alone file

LIMITS
- strings, program lines to 1 megabyte each
- floating-point 5e-309 to 1.8e+308
- 14-digit floating-point accuracy
- unlimited program size, data space
- variable names up to 31 characters

EDITOR
- full-screen, mouse-based editor
- global search, replace
- block copy, move, delete
- menu, function key, command line interfaces
- keyboard macros, script files

DEBUGGING
- set breakpoints and continue
- immediate mode execution
- visual trace, single-step execution
- cross-reference utility

True BASIC™ inc.

39 South Main Street
Hanover, NH 03755
(603) 643-3882

Floyd the Droid

on the Run

A SPECIAL INCLUSION

Our 8-bit gladiator makes his move to the ST, with plenty of action.

by Paul Lay

We're proud to present this, the first-prize effort in our ST Programming Contest. Readers of ***ANALOG Computing*** *have met* ***Floyd*** *before, in issue 53, where he bravely battled mutants on the Atari 8-bit computers. Next month, we'll bring* ***ST-Log*** *readers the high-resolution image of* ***Floyd. . .on the Run****.*

Readers will find the Listings for ***Floyd the Droid on the Run*** *on the magazine's disk version, since the program was far too long to print in these pages. Our Atari Users' Group on Delphi will also carry the complete program.*

You control the latest in robot technology, a cute little droid called **Floyd**. You're surrounded by hordes of evil beings which are out for **Floyd**'s blood (sorry, oil). They shoot past you from side to side in all directions, and, if you collide with them, they explode, zapping your strength down.

Luckily, you're a highly maneuverable droid. Armed with the latest in photon weaponry, you can simultaneously fire four photons in any direction. Also flying around are Atari logos which, if collected, will increase your strength. In all, one hell of a blast!

Floyd the Droid on the Run represents my first project of any real significance on the ST. It has taken almost two months to complete, from start to finish, and it's been great fun exploring a whole new machine.

Thanks must be expressed to all those programmers whose programs and utilities were used in the creation of **Floyd**. Very special thanks go to Harvey A. Kong Tin, who did most of the graphics design work (incredible, really, as he only borrowed an ST for a couple of weekends).

The source code was written with Metacomco's Macro Assembler. All screens are held in straight **DEGAS** format, which made their editing simple.

Note that there's a significant difference between the low- and high-resolution programs, particularly the graphics data and the high-score screen material.

Using the program.

Floyd can be loaded by clicking on the DROIDLO.PRG icon. On loading, an **ANALOG** logo screen will fade in, followed by the title screen. The program will then be loaded and executed within a few seconds.

The keyboard controls are used for various levels and functions of the game. Keys F1 through F6 will begin the game, at levels 1, 10, 20, 30, 40 or 50. Hitting *C* lets you continue from the previous level. Press *D* if you want to see a crazy demo. The *V* key toggles volume on or off, while *P* dumps the screen to your printer. To abort a game, press ESCAPE, and use the SPACE BAR to pause or resume the game.

A joystick in the second port is used to control the droid. Moving the joystick in any direction moves **Floyd** in that direction. Pressing the trigger fires a photon in the direction of the joystick. Four photons can be fired at once.

If your score is high enough, you'll be able to enter your name in the high-score table. All alphanumeric characters are available, plus the SPACE and BACKSPACE. To terminate the entry of a name, press RETURN.

Game play.

You begin each game with a strength level of 75 percent, which is reduced by 4 percent every time you collide with a creature and increased by 6 percent every time you collect an Atari logo.

Your strength level cannot exceed 100 percent, and the game ends when it falls below 0 percent. For every creature that you kill, you score 10 points—but you don't score anything for killing Atari logos. You advance onto the next wave once all the creatures in the current wave have been killed. Once you reach wave 40, all the creatures start moving at double speed.

A line at the bottom of the display during the game shows the current status of your strength, the wave which you're on and your score (or demo indicator).

Notes.

It would take far too long to explain in detail what features of the ST the program uses. Basically, everything is used, apart from the GEM graphics library routines for which there was no real need in this type of program. I will, however, try to outline some important aspects of the program.

Floyd runs entirely in the supervisor mode of the M68000, which allows the interrupt vectors and the hardware addresses at the top and bottom of memory to be accessed directly. The GEMDOS, BIOS and XBIOS routines were very useful, providing easy access in machine language to quite complex operations (such as file handling; and so on). I didn't use any of the Line-A graphics operations; I found it better to code the drawing routines and so on exactly, as I needed them.

A lot of work had to be done in writing a keyboard interrupt routine to allow the use of the joystick. As far as I can make out, there isn't any support for a joystick in the operating system (in other words, you have to write your own interrupt routine). Note that I used a really simple vertical blank interrupt (used only for synchronization to the display), so, once the program's running, there's no checking for a new monitor. Also, the disk drive "busy" light remains on.

One interesting point concerning the hardware: a couple of the chips are accessed via only two addresses. One is used for control, the other for data. Thus, a register within the chip is accessed by writing the required control code followed by writing (or reading) the data. This is quite different from the 8-bit machines, where all the register addresses are hardwired into memory.

All the creatures, in both resolutions, are drawn on a 32-by-32 grid, using two colors in high resolution and sixteen in low. I used my own text routines, because I wanted them to be as fast as possible during the game. The high-resolution font is based on a 16-by-16 grid, and the low-resolution font is based on an 8-by-8 grid, using the 8-bit character set. The star-scrolling in the background uses a simple algorithm, made possible by the fact that the screen is redrawn every vertical blank period. Also, as collision checking is done by comparing positions, it isn't necessary to worry about overwriting things on the screen.

The pause display feature, although quite simple, does show that fine scrolling is possible on the ST. Basically, by using auto-increment and longword addressing, memory can be moved at a very fast rate. The screen dump was easy to implement, because the routine is provided as an XBIOS routine. Note, however, if no printer is connected, the timeout period is drastically decreased. This was done by increasing the rate of the 200Hz system clock.

The demonstration uses completely random movements, with constant firing. The result is impressive and, surprisingly, gets through quite a few levels. The sound is white noise for the missiles, normal control for the creatures' movements and an envelope for their explosions. I found the sound chip difficult to use, probably because I haven't got any good documentation on it.

I could go on and on, but I think I've covered all the major points. It's been great getting to know a new machine, like starting with an 8-bit all over again. At first, programming the ST in machine language looked like it was going to be a nightmare, with hundreds of different routines available, hundreds of different interrupts, the complicated GEM operating system, and so on.

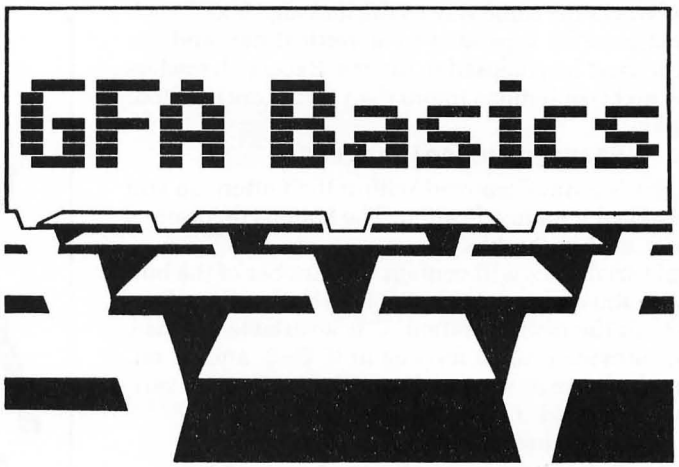
My first task, then, was to get into the ST and turn everything off so that I had a simple machine to work with, and full control. Then I was able to start using various interrupts and operating system routines as I needed them. I strongly advise this route to anyone wanting to program the ST in machine code. The M68000 processor really sets the ST apart, not just because of its incredible speed, but also because of its fantastic architecture (addressing modes and instruction set).

The only reference I've used in writing this program is *The Anatomy of the Atari ST* by First Publishing. It contains almost everything you'll need to know about the ST, other than the GEM graphics library routines. The only drawbacks are that the English is pretty bad (it was written by Germans), and the explanations aren't always clear. I used the Metacomco Macro Assembler to write the program, but can't really recommend it for anyone who wants to do serious machine language programming on the ST. I bought it because it was available at the time, and I'll be upgrading to a different assembler for my next project. Harvey used **DEGAS** for all the graphics designs. I'll let them speak for themselves.

In summary, I'm really pleased with **Floyd the Droid on the Run**. I did not convert the program over from the 8-bit version (in fact, it wouldn't have been possible, because of special features of the 8-bits), but rewrote the entire program, based around the same algorithm and design principles. I've been pretty dedicated to the 8-bits over the years, but the ST has, for me, aroused new excitement and a new challenge! //

---

*Paul Lay studied Computing Science at Imperial College in London. He's owned his Atari 800 for about four years, and has had several programs published in various English magazines (*Personal Computer World, Page 6 Magazine *and* Atari User*). He has also written an arcade game,* ***Sprong,*** *released by Bignose Software in the U.K.*



Alert boxes: what, why and how.

by Ian Chadwick

First, the good news: MichTron has released its second edition of the GFA BASIC manual, and it's about 1000 percent better than the original. Ninety-nine percent of the typos, example errors and misinformation have been cleared up, thanks, I'm told, to the efforts of David Plotkin. Hats off to his efforts—he's helped bring the manual up to an acceptable level.

Now for the bad news: it's no longer in a binder. Instead, it's perfect bound, which makes it difficult to use without a bookstand.

While Plotkin's Herculean efforts helped produce a considerably better piece of documentation, it isn't perfect yet. The manual retains its awkward, gangly structure of splitting commands into several appendixes, when they should have been combined into one complete guide. The index still lacks references to characters such as *, ? and +. An unfortunate typo in the description of the alert box command creates more wrong information about its use.

Which brings me to the subject of this column—alert boxes and how to use them *properly*, no matter what the manual says!

What is an alert box? Well, it's a small boxed area that, under certain conditions, is displayed by GEM to give program users information, warn them of something, or request that they make a choice before continuing. The choices are represented by two or three buttons that can be clicked on or, if outlined with a thicker border, can be selected by pressing RETURN.

To make it more interesting, you can specify which of several icons you want to accompany the alert box, thus providing a visual clue to the alert's purpose and helping to separate one from another if you're using several in a program. Finally, you can display up to four lines, each with thirty characters, in an alert box—so you have the opportunity to provide a reasonable amount of information.

The manual's typo lies in the character used to define the separations between lines. It appears in the first version of the manual as a backslash (\) and in the second as a bullet (•) or small sphere. It isn't either: it's supposed to be a vertical bar (|).

The syntax for an alert box is:

```
ALERT A,MESSAGE$,B,BUTTON$,C
```

A is the variable that defines what, if any, icon will accompany the message. The choices are:

A=0 . . . . . no icon
A=1 . . . . . exclamation mark (!)
A=2 . . . . . question mark (?)
A=3 . . . . . stop sign

MESSAGE$ defines the text that appears in the box (up to four lines of thirty characters each). Anything longer than thirty characters in a single line is truncated, and any more than four lines is omitted. The lines are not justified or centered—if you want them to look a bit classier, you may have to tinker a bit, adding or deleting spaces as necessary. Each line must be separated by the vertical bar, so the program knows where to break them, and the entire string must be enclosed in quotes. Here's an example:

```
" You are about to exit|
     the program.|Save o
r abandon changes?"
```

Note the spaces—they're necessary if the message is to line up properly on the screen.

B is the button variable. It defines which of the three possible buttons to highlight with a thicker black border. This is the one you can select by pressing RETURN. Obviously, if you give B the value of *3* and there are only two buttons in the box, you won't get an outlined button! For A, numbers higher or lower than *0-3* simply "cycle through" the same icons again—e.g., *4* is the same as *0*, *5* the same as *1*, and so on. For B, values outside the range result in no button being outlined. BUTTON$ is the text for the buttons themselves: up to three buttons with eight characters each.

Button text works the same way as the message text—each button's text must be separated by a vertical bar, and the entire string must be enclosed in quotes. Excess characters (more than eight) or buttons (more than three) are omitted. For example:

```
"Save|Abandon|Cancel"
```

Button text is always centered within the button, so you don't need to pad it for justification. The buttons themselves are centered within the box.

The final variable, C, will contain the number of the button selected. You can use this variable to determine where to branch to in the next operation. C is a variable you use, rather than provide a value for—as in IF C=3, and so on. Even if you don't use it, you need to include a dummy variable in the command, or you'll get a syntax error.

So the entire command would look like this:

```
ALERT 3,"  You are about to exit|
       the program.|Save or abando
n changes?",1,"Save|Abandon|Cance
l",C
```

Fairly simple, right? It's the easiest of GFA's GEM commands. The only place you're likely to run into problems is in aligning the text. Message text always begins at the first character to the right of the first quotation mark or of the subsequent vertical bars.

The longest line of text determines the basic alert box width. The alert box borders are adjusted to suit the message text, but if your lines are too short, the buttons may not be displayed properly. Three eight-character buttons require one of the message lines to be at least twenty-eight characters long, if they are to fit properly in a box (with no icon). So pad the line with blank spaces if necessary, to accommodate the buttons, or use an icon. With an icon, twenty-two characters are required to fit three buttons comfortably. Here's the number of characters you need in at least one line of a box, with or without icons, depending on the number of buttons:

BUTTONS	1	2	3
Icon	0	11	22
No icon	6	17	28

Of course, fewer characters in a button mean that less horizontal space is required. Three one-character buttons need only seven characters in any line in order to fit, if no icon is present.

Another consideration is resolution: thirty characters make up less than half a medium-resolution screen width, but the same box will almost completely cover the low-resolution screen. (Remember that, due to the oddities of the program in low resolution, you'll have to use the function keys for commands to the right of the screen center, since the mouse pointer won't reach there!)

A final note: MichTron has released **GFA Vector**, a program to add 3D wireframe graphics to GFA BASIC programs. I'll report on this later, after I've had a chance to use it and ask MichTron why I keep getting error messages in the examples (all in German) and why the sample comments are all in German, too. ///

C-manship

Looking into windows.

by Clayton Walnum

So far, we've talked about dialog boxes (including alert boxes and file selector boxes) and menu bars. That leaves us with one important area of GEM we've yet to touch upon: windows. This is a complex subject, one that we'll need to take several columns to cover. The subject of windows can get as complex as you'd like. There's almost no end to the ways we can use them.

What are windows *really?*

We've all used them, but how many of us have sat down and thought about what a window really is? From the user's point of view, a window truly lives up to its name, allowing us to move a transparent opening over information that may be too lengthy to fit on our screen, giving us a glimpse at data stored somewhere beyond the borders of our desktop.

But this "windowness" is just an illusion, the result of some programmer's tedious and careful work. A window is not a magical creation; it's just a box.

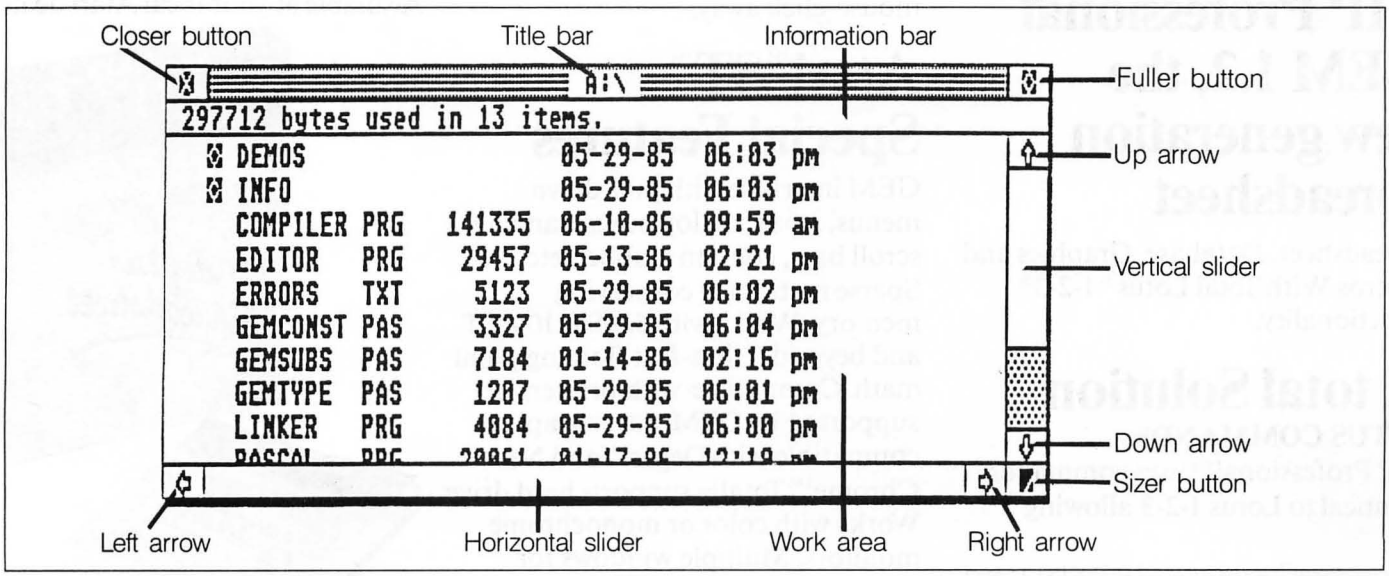
Imagine, if you will, a child's slate on which you've written (What color chalk? Does it really matter?) as many film titles as will fit. Now some guy comes up to you and says, "The movie I'm looking for isn't on that list. Let me see some more." So you take out your eraser, restore the slate to its clean condition, and chalk (all right, all right; the chalk is blue) some more titles onto its surface. The man shakes his head, mumbles something like "Maybe it wasn't a movie after all," and tells you to put the slate down on a table with several others. He then points to a different slate and asks you to pick it up. On this one are written (in yellow chalk, okay?) book titles. The man smiles (Gee, look! There's a piece of spinach in his teeth!) and points to the title *Foundation and Earth* by Isaac Asimov. You erase the slate, set it back on the table, then go to the library and retrieve the book. The end.

Who's the bossy guy in the story? The user, of course. And "you" are the programmer, manipulating the "windows" in the manner the user requests.

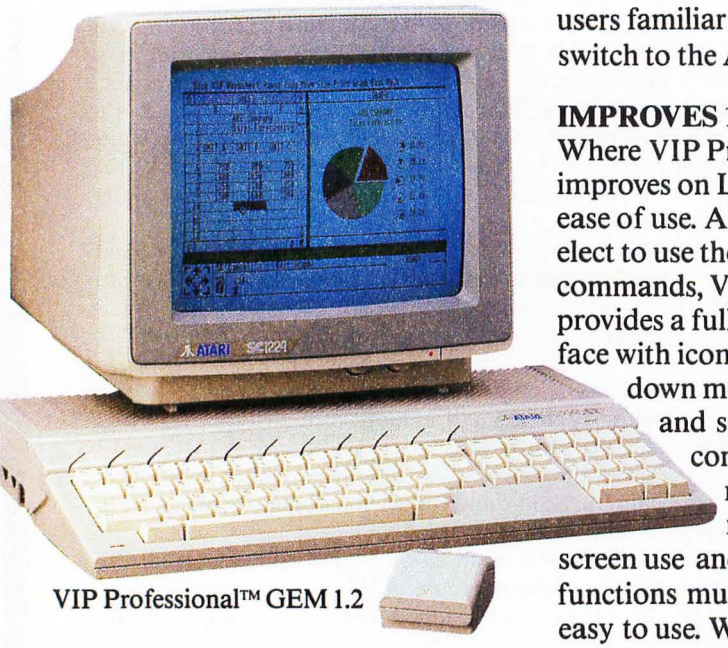
Okay, maybe windows are a *little* fancier than a chalk slate. They do have some extra parts (if we want to use them), such as sliders, movers, fullers, closers, etc., and GEM does provide a small amount of help with handling windows. But, for the most part, a window is just what I said before: a box—a box that you, the programmer, have to maintain in accordance with messages received from your program's user.

What makes up a window? Figure 1 shows you all the components contained in a complete window. You can use any or all of these parts, depending on your application's needs.

Figure 1.—Window components.



Make the PROFESSIONAL™ choice.



VIP Professional™ GEM 1.2

VIP Professional™ GEM 1.2, the new generation spreadsheet

Spreadsheet, Database, Graphics and Macros With Total Lotus® 1-2-3® Functionality.

A total Solution

LOTUS COMMANDS

VIP Professional™ uses commands identical to Lotus 1-2-3 allowing users familiar with Lotus to easily switch to the Atari ST.

IMPROVES 1-2-3

Where VIP Professional™ really improves on Lotus is its attention to ease of use. Although the user may elect to use the keyboard for all commands, VIP Professional™ also provides a full-featured mouse interface with icons, dialog boxes, pull-down menus, mouse ranging and scroll bars. This makes command selection, movement in the sheet, range selection, split-screen use and a score of other functions much more intuitive and easy to use. What were once complex commands with Lotus are now a mouse-click away.

Atari™ ST™ Special Features

GEM interface with pull-down menus, icons, dialog boxes, ranging, scroll bars, column grabber, etc. Sparse matrix for conserving memory. Works with 520ST, 1040ST and beyond. Ultra-fast floating point math. Compatible with printers supported by GEM. Saves graphs compatible with Degas™ and Neo Chrome™. Totally supports hard drive. Works with color or monochrome monitors. Multiple windows for viewing graphs and worksheet at the same time.

Enhanced Text Version 1·2 now available.

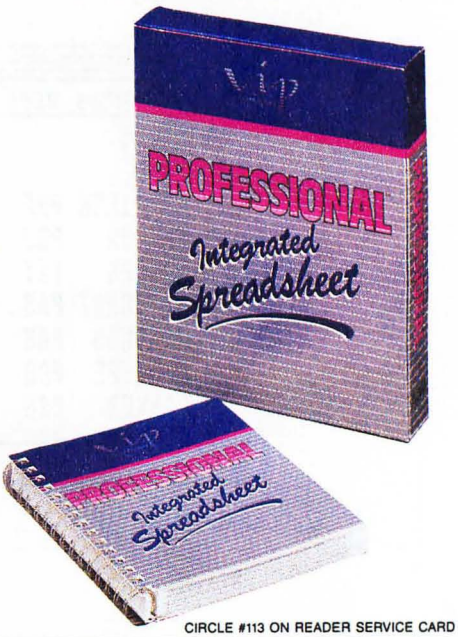
Professional available for:

APPLE® IIGS
APPLE® IIe/c
ATARI ST™
AMIGA™
XENIX®/UNIX®

FLEXIBILITY

VIP Professional™ gives the user, no matter the computer, a uniform, powerful program that conforms to the Lotus standard and which can exchange data with any other program using the 1-2-3 file format. VIP Professional™ is available for the Atari ST, Apple IIe, IIc, IIGS and Commodore Amiga.

Available at your local Atari dealer.



ISD Marketing, Inc., 2651 John Street, Unit 3, Markham, Ontario, Canada L3R 2W5. (416) 479-1880. Fax # 1-416-479-1882.

The window demo.

Listing 1 is this month's sample program. Type it in and compile it. The listing was developed with Megamax C, so if you have a different compiler, you may have to make some slight changes to the code.

When the program is run, a simple window will come up on the screen. This window won't contain all of the parts shown in Figure 1; it will have only a title bar, an information bar, a mover bar (actually the same as the title bar), a fuller button and a closer button—the parts we're going to cover in this month's column. We'll save the others for future **C-manship**s.

Clicking on the fuller button will cause the window to fill the entire desktop workspace. Clicking it a second time will return the window to its previous size. If you place the mouse pointer on the mover bar, then press and hold the left button, you can drag the window to any location you like on-screen—even off the screen, if you want. However, if you *do* move part of the window off the screen, when you move it back again, you'll notice that the window's workspace isn't redrawn. This is one of those details we'll be covering in another column.

When you're through experimenting, click on the closer button. The window will be closed and you'll be returned to the desktop.

Drawing a window.

Now let's take a look at Listing 1 in detail and see how all this window stuff works. The function do_wndw() is where most of the fun takes place, so we'll start there.

The first thing we have to do is decide what our window's maximum allowable size will be. We can limit the size to anything we want, but, in most cases, a window's maximum size is equal to the desktop workspace. The desktop is actually a window itself, the workspace of which is all the area of the screen, excluding the menu bar. The size of this workspace, measured in pixels, varies with the resolution, so we need a way to find out what the actual coordinates are. Luckily, GEM provides us with a function that'll supply the information we need. The call below will return requested information about a window:

```
wind_get(w_h,flag,&x,&y,&w,&h);
```

Here, w_h is the window's handle (in the case of the desktop, the handle is always *0*), the integer flag is a flag telling the function what information we want, and &x, &y, &w and &h are the addresses where the returned information will be stored. What information is actually placed in these locations depends on the value of flag. To get the work area's rectangle, we need to make flag equal to WF_WORKXYWH, which is defined in the Megamax header file, GEMDEFS.H.

The function wind_get() can provide us with a lot of information about our window, including the size of the work area, the size of the entire window, the window's maximum allowable size, the previous window's size, the position or size of either the vertical or horizontal sliders, and the coordinates of the first or next rectangle in the rectangle list (something we'll cover in another column). All the flags you need to request any of this information are already set up for you in the GEMDEFS.H header file that came with your compiler.

Now, where were we? Oh, yes! To get the size of the desktop's work area (which we'll use as our window's maximum allowable size), our call to wind_get() should look like this:

```
wind_get(0,WF_WORKXYWH,&fullx,&fully,
        &fullw,&fullh);
```

Remember: window handle *0* is always the desktop.

Now we know the maximum allowable size for our window, and we've stored that information in fullx (X-coordinate of the window's upper left corner), fully (Y-coordinate of the window's upper left corner), fullw (the window's width) and fullh (the window's height). Next, we need to generate the window, as well as get its handle. (A window's handle is its name; that way we can differentiate it from other windows that may also be in use.) We do this with the call:

```
w_handle=wind_create(PARTS,fullx,
        fully,fullw,fullh);
```

Here, the integer w_handle will receive the window's handle (a negative value indicates that the window couldn't be opened), PARTS is a flag representing the components we want included in the window, and fullx, fully, fullw and fullh are the window's maximum allowable size. A call to wind_create() does not actually draw the window; it only sets up the window in memory.

In our sample listing, PARTS is defined as:

```
NAME|CLOSER|FULLER|MOVER|INFO
```

The definitions for these labels (and all the others needed for a complete window) are defined in the Megamax header file GEMDEFS.H as follows:

Label	Value
NAME	0x0001
CLOSER	0x0002
FULLER	0x0004
MOVER	0x0008
INFO	0x0010
SIZER	0x0020
UPARROW	0x0040
DNARROW	0x0080
VSLIDE	0x0100
LFARROW	0x0200
RTARROW	0x0400
HSLIDE	0x0800

As you can see, each of the above values sets a particular bit in the flag. To select the parts you wish included in your window, you OR the appropriate values together. Though you may include as many or few of the parts as you need for your application, you should never include in a window parts you don't plan on handling in your code. It tempts the user to play around with things he shouldn't, even though, in most cases, it won't do any harm; only your program can actually change a window.

Since we've included the title and information bars in our window, we need to tell GEM where the associated strings can be found. If we neglect to do this, we'll get unpredictable results; we may even end up staring at a row of bombs across our screen (nasty old things). The call below fits the bill nicely:

```
wind_set(w_handle,WF_NAME,title,0,0);
```

Here, w_handle is the window's handle, WF_NAME (defined in the Megamax header file GEMDEFS.H) is a value indicating the field we wish to change, title is the address

of the string we want displayed, and the two 0s are dummy arguments.

Just like wind_get(), wind_set() has many possible values for its flag (represented by WF_NAME in the above call), each of which lets you change one of your window's attributes, including the title or information text, the window's position, the window's components, the sliders' size or position, and whether or not the window is the topmost (active) window. All the flags for this function are defined in your GEMDEFS.H file.

The above call takes care of the title bar. We must make another call to wind_set() for the information line. The call is exactly as above except you would replace WF_NAME with WF_INFO and title with info.

Now we're ready to actually bring the window up on the screen. First, we make a call to draw the animated, expanding box:

```
graf_growbox(startx,starty,startw,
   starth,endx,endy,endw,endh);
```

Here, the integers startx, starty, startw and starth are the X- and Y-coordinates of the upper left-hand corner and the width and height, respectively, of the box's starting rectangle. The integers endx, endy, endw and endh are the equivalent values for the box's ending rectangle.

The call below opens and draws a window:

```
wind_open(w_h,x,y,w,h);
```

Here, w_h is the window's handle, and the integers x, y, w and h are the X-and Y-coordinates of the upper left corner and the width and height of the window, respectively. You can open the window to any size less than or equal to the maximum you set with the wind_create() call.

Next, we call our own function, draw_backgrd(), to fill in the new window's work area. The call to wind_open() actually draws only the window's borders and whatever parts we requested when we created the window. The work area is the programmer's responsibility. It's there for us to do with it what we like.

Let's follow the flow of the program now by taking a look at the function draw_backgrd(). First, we turn off the mouse so the pointer doesn't interfere with any of our drawing. Then we call wind_get(), as we did before, to get the coordinates and size of the work area of the window we just opened. Now that we know this information, we simply draw a filled rectangle at the coordinates returned. A piece of cake!

Handling a window.

Okay, our window's on the screen. Now what do we do with it? We get information about what the user is doing with our window the same way we did with menu bars—through messages.

Since we're only interested in one type of message in the sample program, we're not going to bother with that bulky evnt_multi() call. There's an easier way:

```
evnt_mesag(msg_buf);
```

This call allows us to wait for messages without all of evnt_multi's extra and burdensome baggage. Here, msg_buf is the address of a 16-byte buffer where the message will be stored. Every time the user does something with our window, evnt_mesag() will notify us.

The messages we'll receive are limited to those generated by the parts we included in our window when we created it. More specifically, in Listing 1, the only actions we're looking for are: the window was moved, the fuller button was clicked or the close button was clicked.

When one of these actions occurs, a message is written to our message buffer. The first word, msg_buf[0], will contain the message type received. We'll use this value in a switch statement to choose the appropriate action.

Window moved.

If the window is moved, we'll receive a *WM_MOVED* (defined in the Megamax GEMDEFS.H file) message, telling us we have to reposition the window. The handle of the window moved will be found in msg_buf[3]. The coordinates and size will be found in msg_buf[4] through msg_buf[7] (X, Y, W and H, respectively). We move the window with the call:

```
wind_set(msg_buf[3],WF_CURRXYWH,
msg_buf[4],msg_buf[5],msg_buf[6],
msg_buf[7]);
```

The label *WF_CURRXYWH* is defined in the Megamax GEMDEFS.H file and tells wind_set() that we want to change the current window's coordinates, automatically moving the window to the new position.

What if the user moved the window and we ignored the message by not calling wind_set()? The user would be able to move the window's outline around the screen all he wanted, but as soon as he released the button, the outline would vanish, leaving the window in its original location.

Full-size or previous size?

Another message we might receive in our sample program is *WM_FULLED* message. We get this message when the user clicks on the fuller button, at which time we must either expand the window to its maximum size or, if it's already at its maximum, return it to its previous size. It's up to the programmer to figure out which size is the right one. The first thing we do is call our own function, full_wind(), to set full_flag to its proper state.

All full_wind() does is get the coordinates of the current and full-size windows and compare them. If any of the current coordinates don't match the full-size coordinates, we know that the window is not at its maximum, and we return a value of TRUE. If all the coordinates match, we're already at maximum and need to set the window back to its previous size. We signal this by setting our flag to FALSE.

If we need to reduce the window to its old size, we first need to know the original coordinates. A call to wind_get(), where the second argument is WF_PREVXYWH takes care of that. Once we have the old coordinates, we call:

```
graf_shrinkbox(startx,starty,startw,
   starth,endx,endy,endw,endh);
```

to animate the shrinking box (the parameters are the same as for graf_growbox()), then reposition the window with wind_set().

The process of setting the window to its full size is similar, except we draw an expanding box instead of a shrinking one and use the full-size coordinates for the call to wind_set(). Also, when expanding the window to its maximum, we have to perform a window redraw (in this case,

it's just a matter of drawing that rectangle in the work area) for reasons we'll learn about in a future installment.

Closed for business.

Now, all we have to do is provide a way for the user to get out of our program. The window's close button is perfect for this. When the user clicks it, we'll receive a *WM_CLOSED* message, which will cause us to exit our do... while loop.

When we exit the loop, we find the coordinates of the current window with a call to wind_get(), then we use those coordinates in a call to graf_shrinkbox(). To get rid of the window, we must first close it with the call:

```
wind_close(w_handle);
```

Here, the integer w_handle is the window's handle. Then we must remove the window from memory with the call:

```
wind_delete(w_handle);
```

More to come.

In future months, we'll learn what to do with redraw messages, how to handle sliders and arrows, and how to deal with multiple (gasp!) windows. Betcha can't wait, huh? //

Listing 1.
C listing.

```
/*************************************************/
/*              C-manship, Listing 1             */
/*                    ST-Log #16                 */
/*           Developed with Megamax C            */
/*************************************************/

#include <osbind.h>
#include <gemdefs.h>
#include <obdefs.h>

#define TRUE  1
#define FALSE 0
#define PARTS NAME|CLOSER|FULLER|MOVER|INFO

/* The usual required GEM global arrays */
int work_in[11],
    work_out[57],
    pxyarray[10],
    contrl[12],
    intin[128],
    ptsin[128],
    intout[128],
    ptsout[128];

/* Global variables */
int handle, dum, fullx, fully, fullw, fullh,
    curx, cury, curw, curh, oldx, oldy, oldw, oldh;

int msg_buf[8];

char *title = "C-manship - Issue 16";
char *info = "Learning about windows";


main ()
{
   appl_init ();        /* Initialize application.        */
   open_vwork ();       /* Set up workstation.            */
   do_wndw();           /* Go do the window stuff.        */
   v_clsvwk (handle);   /* Close virtual workstation.     */
   appl_exit ();        /* Back to the desktop.           */
}


open_vwork ()
{
   int i;

   /* Get graphics handle, initialize the GEM arrays and open  */
   /* a virtual workstation.                                   */

   handle = graf_handle ( &dum, &dum, &dum, &dum);
   for ( i=0; i<10; work_in[i++] = 1 );
   work_in[10] = 2;
```

```
   v_opnvwk ( work_in, &handle, work_out );
}


do_wndw ()
{
   int w_handle, full_flag;

   /* Find the size of the desktop's (handle 0) work area. */
   wind_get ( 0, WF_WORKXYWH, &fullx, &fully, &fullw, &fullh );

   /* Create window in memory. */
   w_handle = wind_create ( PARTS, fullx, fully, fullw, fullh );

   /* Set the window's title and info text. */
   wind_set ( w_handle, WF_NAME, title, 0, 0 );
   wind_set ( w_handle, WF_INFO, info, 0, 0 );

   /* Draw the window on the screen. */
   graf_growbox ( 10, 10, 10, 10, 50, 50, 250, 200 );
   wind_open ( w_handle, 50, 50, 250, 150 );
   draw_backgrd ( w_handle );

   /* Change mouse to arrow. */
   graf_mouse ( ARROW, 0L );

   /* Receive event messages until the window closer is clicked. */
   do {
      evnt_mesag ( msg_buf );
      switch ( msg_buf[0] ) { /* msg_buf[0] is message type. */

         /* If window is moved, set window at new location found */
         /* in msg_buf[4] through msg_buf[7].  The handle of the */
         /* window moved is in msg_buf[3].                       */

         case WM_MOVED:
            wind_set ( msg_buf[3], WF_CURRXYWH, msg_buf[4], msg_buf[5],
                       msg_buf[6], msg_buf[7] );
            break;

         /* If the fuller button has been clicked, set window to */
         /* appropriate size based on full_flag.                 */

         case WM_FULLED:
            full_flag = full_wind ( w_handle );
            if ( !full_flag ) {
               wind_get ( w_handle, WF_PREVXYWH,
                          &oldx, &oldy, &oldw, &oldh );
               graf_shrinkbox ( oldx, oldy, oldw, oldh,
                                fullx, fully, fullw, fullh );
               wind_set ( msg_buf[3], WF_CURRXYWH,
                          oldx, oldy, oldw, oldh );
            }
            else {
               wind_get ( w_handle, WF_CURRXYWH,
                          &curx, &cury, &curw, &curh );
               graf_growbox ( curx, cury, curw, curh,
                              fullx, fully, fullw, fullh );
               wind_set ( msg_buf[3], WF_CURRXYWH,
                          fullx, fully, fullw, fullh );
               draw_backgrd ( w_handle );
            }
            break;
      }
   }
   while ( msg_buf[0] != WM_CLOSED );

   /* Get current size of window for use in graf_shrinkbox, */
   /* then close and delete the window.                     */

   wind_get ( w_handle, WF_CURRXYWH, &curx, &cury, &curw, &curh );
   graf_shrinkbox ( 10, 10, 10, 10, curx, cury, curw, curh );
   wind_close ( w_handle );
   wind_delete ( w_handle );
}


/* This function calculates if the window should be drawn to */
/* its maximum size or reset to its previous size.           */
```

YOUR ATARI 520ST SUPPORT CENTER

Atari 520 ST Mono System	Call
Atari 520 ST Color System	For
Atari SF 314 Disk Drive	Best
Atari SH 204 Hard Drive	Price
Atari Hardware	

ABACUS
- Paint Pro 33
- Text Pro 33
- Data Trieve 33
- Powerplan 54
- PC Board Designer 139
- Assem. Pro 43
- Paint Pro Data Lib. 21
- Forth M/T 33

ACADEMY
- Typing Tutor/ Word Invaders 23

ACCESS
- Leaderboard Golf 26
- Tournament Disk 14
- Tenth Frame 26

ACCOLADE
- Mean 18 Golf 29
- Sundog 26
- Mean 18 Course Dsk 14

ACTION
- Action Pak 28
- Solapak 28

ACTIVISION
- Mindshadow 33
- Borrowed Time 33
- Hacker 29
- Music Studio 39
- Paintworks 47
- Little Comp. People 33
- Hacker II 33
- Tass Times/Tonetown 29
- Champ. Basketball 29
- Shanghai 29
- Champ. Baseball 27
- Champ. Football 29

ANTIC
- C.O.L.R. Obj. Editor 23
- Maps & Legends 27
- Macro Assembler 60
- Lattice C 114
- Disk Doctor 23
- A-Calc 45
- Meta Pascal 75
- CAD-3D 37
- A-Ram 15
- A-Seka 27
- GST C Compiler 60
- GST-ASM 45
- Expert Opinion 75
- Flash 28
- Star Struck 19
- Red Alert 19
- Kermit & Remote Co. 19
- Crystal 19
- Fonts, Primitives 19
- Cartographer 27
- G.I.S.T. 27
- K-Switch 27
- K-Resource 38
- Rambrandt 15
- A-Chart 30

ARTWORX
- Bridge 4.0 19
- Compubridge 19
- Maillist 14
- Strip Poker 25
- Peggammon 12
- Hole in One Golf 19

ATARI
- DB Man 99
- DB Master 33
- Joust 19
- Star Raiders 19
- Crystal Castles 19

BECKEMEYER
- Micro C-Shell 39
- Micro C-Tools 19
- Micro Make 27
- RTX 53
- Micro MTC Shell 99
- Hard Disk Accelerate 30
- Ansiterm 19
- Hard Disk Tool Kit 23

CENTRAL POINT
- Copy II 26

DAC
- Easy Accounting 47
- Easy Payroll 33

DATAPACIFIC
- Magic Sac 119

DATASOFT SOFTWARE
- Mercenary 25
- Alternate Reality 23

EIDERSOFT
- ST Protector 22
- ST Karate 22

ELECTRONIC ARTS
- Financial Cookbook 34
- Ultima III 39
- Coloring Book 15
- Golden Oldies 23
- Arctic Fox 29
- Skyfox 33
- Starfleet I 32
- Ogre 29
- Quizam 25
- Chessmaster 2000 33

EPYX
- Winter Games 26
- Rogue 26
- Temple of Apshai 26
- World Games 26
- Champion. Wrestling 26
- Super Cycle 26

FIREBIRD
- The Pawn 29
- Starglider 29

FIRST BYTE
- Kid Talk 32
- Speller Bee 32
- Math Talk 32
- First Shapes 32

HYBRID ARTS
- EZ Track 39
- CZ Android 59

LOGICAL DESIGN WORKS
- LDW Basic Compiler 47

MARKSMAN TECHNOLOGY
- Phaser 69

MASTERTRONICS
- Renegade 14

MEGASOFT
- A Copier 27
- X-Rated Library 17

OTHER:
- Certificate Maker 33
- Psion Chess 39
- Microleague Baseball 39
- Aegis Animator 54

MICHTRON
- DOS Shell 26
- The Animator 26
- Personal Money Mgr 33
- Utilities 39
- M-Disk 26
- Softspool 26
- Flip Side 26
- Calendar 19
- Mi-Term 33
- Time Bandits 26
- Cornerman 33
- Mighty Mail 33
- Cards 26
- Business Tools 33
- Major Motion 26
- Mi-Dupe 20
- BBS 2.0 54
- Pro Football Wizard 27
- Pinball Factory 27
- Your Finance Future 26
- Mi-Print 20
- Trivia Challenge 27
- GFA Basic Compiler 54
- Trim Base 69
- Match Point 27
- Raid 27
- Super Directory 27
- GFA Basic. Interpret. 54
- Tech Mate 26
- Perfect Match 26
- Space Shuttle II 26

MICRODEAL
- Karate Kid II 29

MICROLEAGUE
- Baseball 39

MICROPROSE
- Silent Service 26

MI-GRAPH
- Easy Draw 52
- Fast 33

MILES
- Harrier Strike 33

MINDSCAPE
- Brataccus 33
- S.D.I. 33
- Balance of Power 33

NAVARONNE
- Astrology 29
- Timekeeper 34

OSS
- Personal Pascal 50

PROCO
- Pro-Copy 24

PROGRESSIVE COMP. APPL.
- Graphic Artist 132
- Font Editor 54
- Font Pak I 34
- Desktop Publ. Lib. 34

PROGRESSIVE PERIPHERALS
- Picasso's Revenge 40

PSION
- Chess 39

PSYGNOSIS
- Deep Space 34
- Arena 26

QUANTUM
- ST-Talk 14

QUICKVIEW
- Zoomracks II 99

REGENT
- Regent Spell 34
- Regent Base 67
- Regent Word II 67

SHANNER
- Soundwave SW-1 15
- ST-Key 10
- Macro-Manager 15
- Macro-Desk 10
- DB Calc II 15
- Colorwriter 10

SHELBOURNE
- Pool 23
- Shuffleboard 20

SIERRA
- Kings Quest II 33
- Ultima II 20
- Winnie the Pooh 17
- Black Cauldron 26
- K/Q Hint Book 6
- B/C Hint Book 6
- Donald Duck 17
- Cash Disbursements 67
- Space Quest 33
- Kings Quest III 33
- Kings Quest I 33

SOFTLOGIK
- Publish. Partner 99

SPECTRUM HOLOBYTE
- Gato 27

SPINNAKER
- Homework Helper:Math 34
- Treasure Island 27

SPRINGBOARD
- Certificate Maker Call

SSI
- Phantasie 26
- Phantasie II 26

SUBLOGIC
- Flight Simulator 35

TDI
- Modula-2 75
- Modula-2 Develp.Kit 99
- UCSD Pascal 59

TIMEWORKS
- Datamanager 53
- Swiftcalc 53
- Wordwriter 53

UNISON WORLD
- Printmaster 26
- Art Gallery I 19
- Art Gallery II 19

VIP
- VIP Professional 149
- VIP Lite 69

X-LENT
- Typesetter Elite 29
- Rubber Stamp 25
- Music Box 32
- Megafont 25
- Write 90 Degrees 19
- PM Interface 19
- Typeset Color Icon 13

PRINTERS

STAR MICRONICS
- NP-10 179
- NX-10 229

PANASONIC
- KX-P 10801 199
- KX-P 10911 259
- KX-P 3131 259
- KX-P 1592 399
- KX-P 1595 579
- KX-P 10921 329

MODEMS
- Supra 300 ST 59
- Supra 1200 ST 149
- Avatex 1200 99
- Avatex 1200 HC 129
- Compuserve Start Kt 24
- Microstuffer 59
- Omega Terminal 20
- Atari 301 Modem 49

ST ACCESSORIES

Disk Cleaning Kit	14
Dust Covers	7
Monitor Stand	12
Mouse Pad	8
Shanner Planner	29
The Easel	14
Disc Directory	20
Pocket Pak	10
The Library	34
Flip N File Micro(25)	6
Flip N File " (50)	17
Flip N File II (40)	20
Flip N File Micro (5)	2
Flip Sort Micro (50)	7

ST BOOKS

Atari ST GEM Reference
Atari ST Internals
Atari ST Machine Language
Atari ST Tips & Tricks
Atari ST from Basic to C
Atari ST Basic Training
Atari ST Graphic & Sound
Atari ST Logo
Atari ST Peeks & Pokes
Atari ST for Beginners

All books are $16 ea.

CABLES

ST to Printer	15
ST to Modem	15

COVERS

130XE Computer
800XL Computer
1050 Disk Drive
520ST Computer
SF354 Disk Drive
SF1224 Color Monitor
SM124 Mono Monitor
1040ST Computer

CALL

To order call TOLL FREE
1-800-824-7506
ORDER LINE ONLY

No extra charge for MasterCard or Visa

COMPUTER CREATIONS, Inc.

P.O. BOX 493 - DAYTON, OHIO 45459

For information, order inquires, or for Ohio orders (513) 435-6868

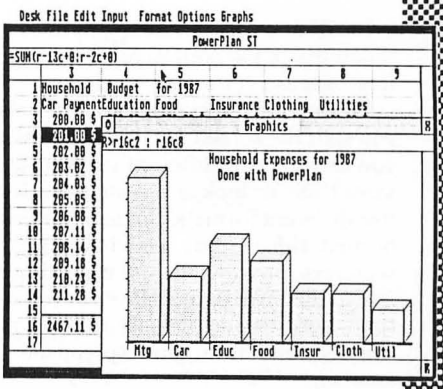
VISA® MasterCard

Order lines Open 9:00 a.m. to 8:00 p.m. Mon.-Fri.; 10 a.m. to 4:00 p.m. Sat. (Eastern Standard Time). Minimum $15 per order. C.O.D. (add $3.00). Please specify computer system. Call toll free number to verify prices and availability. Prices and availability are subject to change without notice. We ship C.O.D. to Continental U.S. addresses only! Please include 4% shipping on all Hardware orders (min. $4.00). Software and accessories add $3.00 shipping and handling in continental U.S. Actual freight will be charged outside U.S. to include Canada, Alaska, Hawaii, Puerto Rico and APO. Ohio residents add 6½% sales tax. Canadian orders add 5% shipping, (min. $5.00). All other foreign orders, please add 15% shipping, (min. $10). For immediate delivery send cashier's check, money order or direct bank transfers. Personal and company checks allow 3 weeks to clear. School purchase orders welcome. Due to our low prices, all sales are final. NO CREDITS. All defective returns must have a return authorization number. Please call (513) 435-6868 to obtain an RA# or your return will not be accepted for replacement or repair. FOR YOUR PROTECTION WE CHECK FOR CREDIT CARD FRAUD. We do not bill until we ship.

```
full_wind ( w_h )
int w_h;
{
   int c_x, c_y, c_w, c_h,
       f_x, f_y, f_w, f_h;

   wind_get ( w_h, WF_CURRXYWH, &c_x, &c_y, &c_w, &c_h );
   wind_get ( w_h, WF_FULLXYWH, &f_x, &f_y, &f_w, &f_h );
   if ( c_x != f_x || c_y != f_y || c_w != f_w || c_h != f_h )
      return ( TRUE );
   else
      return ( FALSE );
}


/* This function draws a white background in a window's  */
/* work area.  w_h is the window's handle.               */

draw_backgrd (w_h)
int w_h;
{
   int wrk_x, wrk_y, wrk_w, wrk_h;
   int pxy[4];

   /* Turn off mouse for all drawing operations.  */
   graf_mouse ( M_OFF, 0L );

   /* Get the size of the window's work area. */
   wind_get ( w_h, WF_WORKXYWH, &wrk_x, &wrk_y, &wrk_w, &wrk_h );

   /* Set the color and fill style. */
   vsf_interior ( handle, 1 );
   vsf_color ( handle, WHITE );

   /* Draw the rectangle in the window work area. */
   pxy[0] = wrk_x;
   pxy[1] = wrk_y;
   pxy[2] = wrk_x + wrk_w - 1;
   pxy[3] = wrk_y + wrk_h - 1;
   vr_recfl ( handle, pxy );

   /* Drawing over, so turn mouse back on. */
   graf_mouse ( M_ON, 0L );
}
```

●

A Spreadsheet Comparison



A look at *EZ-Calc, SwiftCalc ST, Power Plan, A-CALC Prime* and *VIP Professional.*

by David Plotkin

The Atari ST is a powerhouse of a computer. Equipped with double-sided drives or a hard drive and a megabyte of memory, it is a suitable candidate for use in a business environment, as well as for advanced applications at home. One of the primary pieces of software needed if you want to use a computer for business is a spreadsheet. Spreadsheets are handy for many home tasks, too. The purpose of this article is to take an in-depth look at the five ST spreadsheets currently available.

What is a spreadsheet?

A spreadsheet is a grid of intersecting vertical columns and horizontal rows of "cells," into which data, text or formulae may be entered and the results calculated. Each cell is identified by the row and column designation of its position (the cell at the intersection of row A and column 1 would be called *A1*). The width of each cell can be adjusted to fit the information that will be entered into it. Rows and columns can be inserted and deleted for flexibility. The formulae in the cells can refer to numbers or the values in other cells, and can use a wide variety of built-in mathematical functions. Text in cells can be used as labels, to provide information about the numeric values. A spreadsheet looks very much like the paper ledger long used by accountants. Its advantage over such a tool: when any value in any cell is changed, the whole spreadsheet can be quickly recalculated, to show the new results. It is in this automatic recalculation that the power of the spreadsheet lies. The user can test a whole series of different values of data, trying out large numbers of possible scenarios, or "what-ifs."

Spreadsheets usually contain data covering a much larger area than can be shown on-screen. Using a variety of techniques, you scroll your data across the screen, looking at various pieces of your sheet. Your screen is a window onto the spreadsheet.

Spreadsheets are among the most popular software, and are available for just about every computer ever made. The first commercial spreadsheet was **VisiCalc**, developed for the Apple II in 1979. The largest selling spreadsheet is Lotus 1-2-3 for the IBM PCs and compatibles; it's a standard of sorts. Spreadsheets are used for project planning, tax records, loans, financial modeling, mathematical calculations, checkbook balancing and even, occasionally, for word processing.

Basis of comparison.

The spreadsheets compared in this article are quite different from one another. I've tried to identify factors which would be important to a frequent user.

Interface. All of the programs included here are GEM based, using the mouse, windows and menus to a certain extent. How well the advantages of GEM are integrated into the spreadsheets is important, not only for ease of use, but because a well-designed interface makes a program much more powerful—as well as more fun to use.

Functions. The heart of any spreadsheet is how many mathematical functions it can perform. More powerful programs have advanced financial functions, such as Net Present Worth and IRR, built right in. Further, the "look-up" functions are very important, since these allow you to set up tables where values from a cell can be matched up to values in a row or column, then adjacent values from the table can be selected. This is imperative for programs like tax tables. The ease with which "relative" and "absolute" cell references can be specified is also important. Relative and absolute cell references refer to the cells which appear in a formula. When you copy or move a formula with an absolute reference to a different place on the spreadsheet, that formula will still refer to the same cell as it did previously. If the formula has a relative reference, then it will refer to a cell in the same *relative position*. For example, if the formula in cell B1 refers to cell A1, and you move the formula to cell C6, then, if it's an absolute reference, it will still refer to cell A1; but if it's a relative reference, it will now refer to cell B6.

Copying. One of the most-used operations in a spreadsheet is the copying of contents of one cell (or a range of cells) to a destination. This is especially useful when the source cells contain formulas, because then you don't have to enter the formulas into each cell separately. The

flexibility of the copy operation is important because it's used so often.

Windows. Even simple spreadsheets let you split the screen into two windows, so you can see two different sections at the same time, or look at the same section in two different formats. Some programs go beyond this, letting you have multiple windows for viewing, and even allowing you to load two spreadsheets at the same time—presenting the possibility of moving sections from one spreadsheet to another.

Size. The overall size of a spreadsheet you can build may depend not only on the limits built into the program, but also on how large the program is and how efficient it is with memory usage. Recently, a memory allocation technique called the "sparse matrix" has become popular. With a sparse matrix, only cells actually *used* take up space in memory. There are variations in how a sparse matrix is implemented, with some schemes being more efficient than others.

Speed. The speed with which a spreadsheet recalculates after you've made a change—as well as the speed with which you can scroll data across the screen—can have quite an impact on how much work you get done. To be perfectly honest, every product I looked at could use *some* improvement in scroll speed, although a few are pretty good. Some are so slow as to seriously inhibit your efficiency. Benchmarks which give some indication of the speed of each spreadsheet are included in the table at the end of this article. The speeds at which each program can copy 1000 cells, recalculate the contents of 1000 cells, and scroll across 100 cells containing data are shown. Use these numbers only as rough estimates.

Graphing. A built-in graphing function can be exceptionally useful in viewing data, as a pictorial representation can often tell you more than simple rows of numbers. Several of the programs here have extensive graphing capabilities, and can use labels, titles, legends and multiple data sets.

Database. You enter a lot of data in the course of building a spreadsheet. The ability to manipulate this data with typical database functions, such as sorting and searching, gives you a lot more flexibility and brings more utility to your spreadsheet.

Macros. This is the ability to automate your keystrokes. You can assign a whole series of keystrokes and commands to a single key. Then, pressing this key will activate the series of keystrokes, just as if you'd typed them. Macros have the potential to improve your efficiency considerably, saving time and allowing you to design simple, easy-to-remember command sequences for use by those not familiar with the spreadsheet.

Help. Even with the GEM interface, spreadsheets can be pretty complex to learn, especially if they aren't used frequently. You *can* look things up in the manual (if you can find it), but a good help facility within the program—to tell you a command's format or just general information—is far quicker.

Accessories. Extra accessories are often included. A calculator (for intermediate computations), a sideways printing program (for those extra-wide spreadsheets), and a notepad—all can be useful. The notepad is an especially good idea. It lets you attach notes to a cell, documenting what assumptions were made and other information about the cell. In the IBM PC world, people pay $80 extra for a program that lets you to attach notes to cells in Lotus 1-2-3.

General. Spreadsheets need to be able to distinguish between numbers and labels entered into a cell. Under many circumstances, it will be obvious—the cell will start with a number or letter, and this will determine the treatment. But what if you want to use a number for a label, or you use a formula that starts with a cell reference (a letter)? The various spreadsheets accomplish this in different ways. "Freezing titles" is a technique whereby a column or row (or both) is frozen, so that when you scroll the spreadsheet, they remain on-screen. This helps you remember what numbers to enter in each row or column. "Named ranges" allow you to assign a name to a range of cells. You can then refer to that range of cells by name in formulas.

EZ-Calc

ROYAL SOFTWARE
710 McKinley
Eugene, OR 97402
(503) 683-5361
$69.95

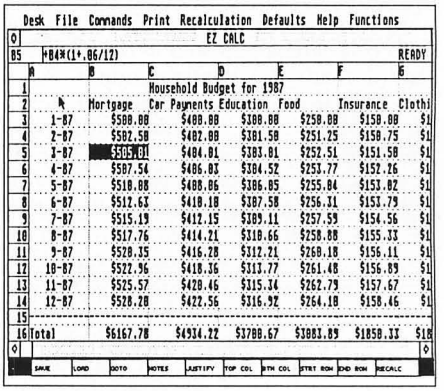
EZ-Calc is a fully GEM-based spreadsheet, with menus across the top. A work area pops up at the bottom of the screen, to provide a template for entering information on various commands. An edit line shows current contents of the cell highlighted. Columns have letter designations (*A, B, C. . .*), and rows are numbered in "standard" format.

The apostrophe (') is used to indicate when a number's to be treated as a label, and the plus sign (+) indicates that cell contents starting with a letter actually designate a formula (such as *+A1-3/B2*). Cell entries can only be finished with RETURN (arrows won't work to finish entry and move a cell in the chosen direction). Only the top row and/or leftmost column can be frozen on-screen. The program uses a dollar sign ($) in front of a row number or column letter in a cell designation, to indicate an absolute reference.

The GEM interface is good. Almost all commands are available from menu selections, and you can select a range for copying or moving via "click-and-drag," although this only works *before* the command is selected. The range you select in this manner must be completely visible on-screen; you can't select a range extending past the window border. If you do select a range and a command, the dialog box which pops up at the bottom of the screen will have the selected range already in it. Once the dialog box is on-screen, however, you can only select a range by typing in cell coordinates. Widening of cells is possible only from a menu selection; you can't click-and-drag to set cell width. When entering a formula, you can click on a cell with the mouse, and the cell reference will appear in the formula, along with proper punctuation. The most frequently used commands are available from the function keys, and appear on-screen to remind you of their functions.

The maximum **EZ-Calc** spreadsheet size is 300 columns by 999 rows. It's actually one of the smaller spreadsheets, but consider this: I did my 1986 tax return, both state and federal, on **EZ-Calc**, and didn't run out of room.

EZ-Calc features a wide range of mathematical functions, including: trig, log,


Desk File Commands Print Recalculation Defaults Help Functions

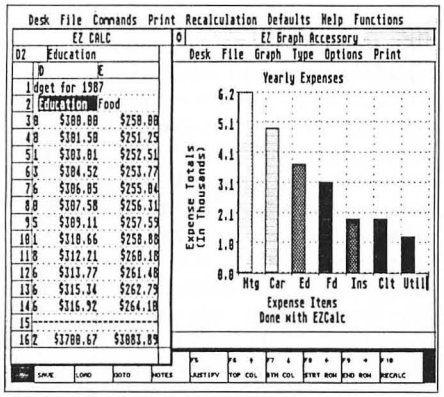
EZ CALC

B5 +B4*(1+.06/12) READY

	A	B	C	D	E	F	G
1			Household Budget for 1987				
2		Mortgage	Car Payments	Education	Food	Insurance	Clothi
3	1-87	$500.00	$400.00	$300.00	$250.00	$150.00	$1
4	2-87	$502.50	$402.00	$301.50	$251.25	$150.75	$1
5	3-87	$505.01	$404.01	$303.01	$252.51	$151.50	$1
6	4-87	$507.54	$406.03	$304.52	$253.77	$152.26	$1
7	5-87	$510.08	$408.06	$306.05	$255.04	$153.02	$1
8	6-87	$512.63	$410.10	$307.58	$256.31	$153.79	$1
9	7-87	$515.19	$412.15	$309.11	$257.59	$154.56	$1
10	8-87	$517.76	$414.21	$310.66	$258.88	$155.33	$1
11	9-87	$520.35	$416.28	$312.21	$260.18	$156.11	$1
12	10-87	$522.96	$418.36	$313.77	$261.48	$156.89	$1
13	11-87	$525.57	$420.46	$315.34	$262.79	$157.67	$1
14	12-87	$528.20	$422.56	$316.92	$264.10	$158.46	$1
15							
16	Total	$6167.70	$4934.22	$3700.67	$3082.89	$1850.33	$18

SAVE LOAD GOTO NOTES JUSTIFY TOP COL 8TH COL STRT ROW END ROW RECALC

EZ-Calc.



EZ-Calc.

SUM, MAX, MIN, COUNT, LOOKUP, AVG and five financial functions. It also has logical functions like IF, AND, NOT and OR. The IF function only tests and outputs numbers or formulas; you can't use it to put text or labels into a cell.

EZ-Calc's copy function allows column-to-column and row-to-row copies. A single cell or a range can be copied or moved by click-and-drag, but copying a cell or range on top of itself erases the cell contents. A single cell can also be copied to a range using the menu "Replicate" command. Database commands set a range and sort the range, on either the top row or left column of the range. **EZ-Calc** does not allow named ranges.

You can read in a spreadsheet from disk and have it merged into the existing sheet, beginning at the point where the cursor's located. Partial sheets can be saved, and sheets can be saved in ASCII format for use with word processors. **EZ-Calc** does not support the DIF format for data interchange with other spreadsheets.

EZ-Calc isn't a speed demon, but is fast enough not to get in your way. The window scrolls at an acceptable speed, and the cursor doesn't keep trying to overshoot the spreadsheet's edge if you accidentally hold down the arrow key too long and get ahead of the scrolling.

You're allowed two windows to view your spreadsheet. These are true GEM windows, and can be moved, sized and closed when you're done. Formatting isn't very powerful. You can set justification (center, right or left alignment for numbers or labels) over any specified range, but formatting for commas is global (the whole spreadsheet) only. You can set precision (number of decimal places) globally, then change it for selected ranges. The spreadsheet can be set as dollar format, then certain ranges can be protected from this format. You can print out the whole spreadsheet or just a selected range, set the page length and width, turn off the headings, and select from several print styles (NLQ, bold for noted cells, compressed, Pica and Elite).

Both a notepad and a calculator are provided. The notepad is four lines, and a separate note can be attached to each cell. Any cell with a note attached can be shown in bold on-screen and printed out in bold, as well. The calculator is provided as a desk accessory, so you must boot with it to have access to the function. It's a standard four-function calculator, with a neat feature: you can click-and-drag the result to any cell in the spreadsheet.

A very nice graphing package is available from a desk accessory. You have control over general labels (axis labels, graph titles, etc.) and can edit any text in the graph. To make a graph, you drag a cell or a cell range to the graph window, and the graph is constructed automatically. To label the data, you can drag a cell containing a label (or a range of label cells) to the graph window.

Data and labels can be inserted and deleted, and types of graphs include: line, bar and pie. The line and bar graphs can include up to four sets of data (bar graphs and line graphs can be stacked), and you can pull a slice from the pie graph.

The on-line help facility is available, but you must choose the subject, and the help isn't very specific—many topics aren't covered. Luckily, you don't need much help with this simple package. The documentation is a small book which *does* cover all essential information. However, parts of the manual are on the disk (the graphing program and file for converting Lotus 1-2-3 files to **EZ-Calc**), and the included tutorial really doesn't illustrate many of the program's features.

To sum up, **EZ-Calc** is a good simple spreadsheet. It has financial functions, a good notepad and graphing functions, and is very easy to use. It also supports many of the same conventions as Lotus 1-2-3, so you'll have little trouble moving to it if you're familiar with that package.

SwiftCalc ST

TIMEWORKS
444 N. Lake Cook Rd.
Deerfield, IL 60015
(312) 948-9208
$79.95

SwiftCalc is another standard-looking spreadsheet, with menus across the top and a work area at the bottom. The work area shows current cell contents and type (label, value, formula, etc.) of each cell. Columns and rows are designated in the same way as **EZ-Calc**. The quotation mark (") is used to signify labels, and the plus sign (+) indicates a formula. Cell entries can be finished either with the RETURN or the arrow keys. You can freeze any number of rows and/or columns for borders. The program uses the case (upper or lower) of the letter in the cell designation to distinguish between relative and absolute cell references. Columns can be widened automatically if your data overflows, or can be set using a menu. There is no click-and-drag option for setting column width.

The user interface is, in my opinion, not as convenient as **EZ-Calc**'s. Some operations are available from the drop-down menus, but most must be accessed by pressing the slash (/), which brings up a row of letters in the work area. To select one of the commands, you either press the letter key or use the arrow to move the cursor through the letters and hit RETURN when you reach the one you want. As each letter is highlighted, a short explanation of the function appears in the work area.

This method of selection is not as straightforward as drop-down menus or even Lotus 1-2-3's menus, but you can get used to it. You can also select some functions with the function keys.

SwiftCalc doesn't support click-and-drag for range selection. In fact, for some functions, you can't even type in the limits of the range, but must set limits via drop-down menu items ("Start Block" and "End Block"). For most formulas, clicking on a cell enters it into the formula, but you must then enter the proper punctuation by hand. If you make a syntax error in entering a formula, the program will no longer accept cell input by use of the mouse. Instead, you *must* type it in.

The maximum **SwiftCalc** spreadsheet size is 8192 rows by 256 columns, so space will never be a problem. A full range of functions, including four financial functions, IF, CHOOSE, SUM, MAX, MIN, AVG, INT, trig, log, IF, OR, AND and LOOKUP are available. The IF function does let you put text into a cell as a result of the function.

The copy function is powerful. You can copy a cell to a row, a cell to a column, a row to a range of columns, or a column to a range of rows. Strangely, you can't copy one cell to a range of cells, and moving cells is only allowed by row or col-

umn, not by range. As mentioned earlier, no click-and-drag is allowed.

Database functions include the ability to specify a range and sort on any column in that range (but not by row). You can fill a range with data, using a chosen starting value and increment. It's also possible to find, delete and extract records (rows) to another block of cells which meet the stated criteria. Cells and blocks of cells can be named.

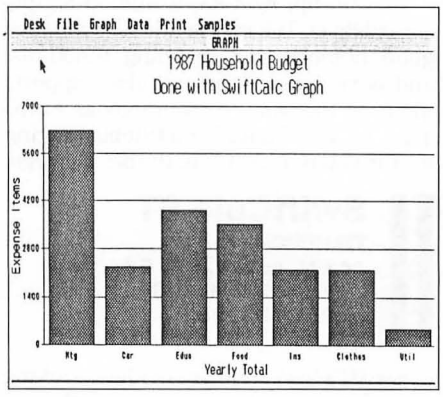
A spreadsheet from the disk can be merged into the existing sheet, but the merge is of limited use because the new spreadsheet can only be merged at cell A1 (the upper left corner). **SwiftCalc** is one of an integrated series from Timeworks, and can exchange data and files with **WordWriter** (word processor) and **Database Manager** (database). Files can be imported from and exported to these programs. For example, I imported my tax receipt records from **Database Manager** into **SwiftCalc**. Also supported is the DIF format, for file exchange with other spreadsheets.

SwiftCalc is about average at copying cells and doing recalculations. Its weakest point is its window-scrolling speed. This definitely needs work. In fact, the window scrolls so slowly that the cursor can actually move *off* the screen, so you don't even know where the cursor is (except by watching the cell indicator in the work area).

When you've stopped moving the cell pointer, the window jumps to the proper cell. This is inconvenient if you need to move to a cell holding a particular set of contents, but you don't know the cell's designations. Further, the scroll speed is so slow that the keyboard buffer can fill up if you hold down the arrow key—so that, if you bump up against the edge of the spreadsheet, the window will keep trying to move, and it can be a while before the buffer is emptied so that you can proceed.

SwiftCalc can split the screen into two windows, oriented horizontally or vertically. These are not GEM windows; you must press a special key to switch between them. Though only a single spreadsheet can be loaded at any time, you can "link" sheets, so that a named cell can be pulled into the current sheet from one stored on disk.

You can set formats (precision, column width and justification) for the whole sheet (global), for a row or a column, but *not* for a range of cells. As an alternative, **SwiftCalc** contains up to eight user defined formats, where you choose such things as commas, dollar signs and scientific notation. Once you've defined these, you simply choose one for the specified column, row or sheet. You can print the whole spreadsheet or just the specified range, choose characters per line and lines per page, include printer codes at the beginning to put your printer in compressed, NLQ mode, etc., turn off the headings, and print borders and titles on each page.



SwiftCalc.

SwiftCalc does not include a notepad, calculator or macro capability.

Graphing is handled via an external program. You must use **SwiftCalc** to create a special graph file, then exit the program and load the graph software. This program is complex but versatile. It's capable of vertical bar (overlaid, stacked, hi/lo), line (overlaid, summed), horizontal bar (overlaid, stacked, Gannt), scientific (overlaid) and pie graphs. You can import rows, columns or labels from the file, and add data from within the graph program. You have a high degree of control over page layout, and can even scale your output. You can also edit the graph and axis titles and divisions, as well as colors and patterns for the output. You can even pull out pie sections, add legends and put in a footnote.

The on-line help facility is *extensive*. Pressing HELP brings up help screens which vary, depending on what you're trying to do. For example, if you press HELP after you've hit the key for a built-in function, some screens with general information on that function will come up, as well as a screen showing how to enter data into the function. This "template" would be even more helpful if it came up automatically when you called up the function, but the help facility is still very good.

The manual is also quite good. In fact, it's one of the best I've seen. It comes in a professional binder, with an extensive tutorial (including graphing and a financial model), a quick reference section and an excellent index.

SwiftCalc has many powerful functions, including graphing and database functions. It suffers from slow scrolling and an interface which I found difficult to master, and which contains less GEM-related helpful functions than do other spreadsheets.

Power Plan

ABACUS SOFTWARE
P.O. Box 7219
Grand Rapids, MI 49510
(616) 241-5510
$79.95

Power Plan has a different look from other spreadsheets in this overview, because both its rows and columns are numbered. To refer to a given cell, you would use *R1C2* for row 1, column 2. You'll have to remember the row/column order, because specifying *C2R1* won't work. Cell

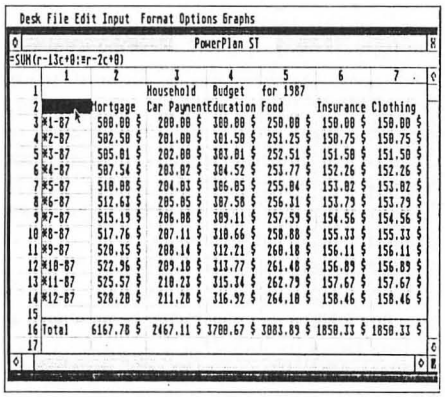
references are absolute by default (also the reverse of other spreadsheets), and relative references are clumsy: to refer to a cell two rows down and one column to the left, you'd use *R+2C-1*. Thus, you must count over to the cell to find out how far away it is. You can also point and click on the cell you want, then press INSERT to enter that cell into a formula in absolute format, or press the HELP key to enter it in relative format. Formulas must be preceded by the equals sign (=), and numbers can never be labels. To use a number as a label, you have to have a nonnumeric character in it somewhere. Cell entries can be finished with RETURN (the cursor moves down) or with TAB (the cursor moves right). You can't freeze columns or rows to serve as page borders.

The interface features all commands available from drop-down menus, with twenty of the most frequently used present on the function keys (used with the SHIFT). You can point and click on a cell or select a range by click-and-drag, although you must press INSERT or HELP to enter the cell or range into the formula in the proper format (absolute or relative). You can also select the cell you want by moving the cell pointer to it, using the SHIFT and arrow keys. Widening of cells is available from both the menu and via click-and-drag of the column boundary. If you select a range by click-and-drag, the window will scroll automatically, so you can choose a range larger than what can appear in the window.

The maximum size of a **Power Plan** spreadsheet is 65535 rows by 65535 columns. Functions include trig, log, MAX, MIN, SUM and AVG, but the important IF and LOOKUP functions are missing. And there are no financial functions at all.

Editing is done by specifying a range, then using the menu items "Cut," "Copy" and "Paste" to move, copy or delete a range of cells. You can only replicate a single cell (not a range), and the block of cells you replicate *to* must be immediately to the right and/or below the source cell—you cannot replicate to a nonadjacent block. Database functions include ascending or descending sorts on specified key rows or columns. Named ranges are not supported, but you *can* name a single cell and refer to it by name in a formula.

Power Plan doesn't allow for merging of spreadsheets, nor does it support DIF format. As part of an integrated series, it can import data from **DataTrieve** (database) and **TextPro** (word processor). It can also export data to **TextPro** by either



Power Plan.

printing to disk in ASCII format or by saving a special section of memory known as the "Scrapbook."

Recalculation and scroll speeds are adequate, and the program recovers quickly when you try to scroll past the edge of the spreadsheet. It's slowed down by the fact that the sheet appears to be recalculated after *every* entry, including labels which don't change any of the calculations. I recommend turning off the automatic recalculation.

Power Plan allows up to seven GEM windows to view your single spreadsheet. Formatting flexibility is very poor. You can set the precision (number of decimal places) or scientific notation only on a global basis, and neither comma format nor justification control are available at all.

You can choose a range for currency units (there are four available, including dollars), but the symbol appears *after* the number. When printing out the spreadsheet, you can't turn off column and row indicators (numbers), nor is it possible to print titles on every page to identify columns and/or rows. No special printer functions are supported.

The so-called calculator is just an empty window for entering formulas and getting results. You can't paste these results into the spreadsheet.

There's something called a notepad, that's actually a sort of crippled word processor. It consists of a window for entering notes and is not attached to any cell. It also does not support word wrap, so you must either accept fragmented lines or press RETURN to cause the word break. You can cut and paste between areas of the notepad, or move a block of cells into the notepad, with each cell occupying one line. You may also move a section of the notepad to the spreadsheet, but you must specify how many cells wide the data is. You can print the notepad's contents, and there is a way to embed printer codes in contents to customize the printout. You can specify a column or row of cells to use, and each time the notepad is printed, the contents of the next cell in the chosen range will be inserted at the indicated position, giving rudimentary form-letter capabilities.

Graphing functions are available from the drop-down menu. You select a range of values to plot (rows or columns), then open a graphics window. Graphs available include bar (3D and stacked), line, area and pie charts. Changing spreadsheet data will change the graph. Symbols are available to distinguish between sets of data. You can add text at any point in the window, delete text or move entered text. You can't import labels from the spreadsheet and, once text has been added, you can't resize the graphics window.

Power Plan has no on-line help facility. The manual assumes you know nothing about your ST—and wastes a lot of pages explaining menus, dialog boxes, windows, even what a mouse is. A full explanation is included on what a spreadsheet is; but the manual contains no list of functions, the tutorial is very sketchy, and the index is so incomplete as to be virtually worthless.

Power Plan has an unusual interface which tends to be clumsy, lacks many essential functions and has little formatting capability. It supports a large number of cells, has a good graphics package and a "notepad." Unfortunately, these good peripheral features don't compensate for the shortcomings in the basic spreadsheet package.

A-CALC Prime

THE CATALOG
Antic Publishing
544 Second Street
San Francisco, CA 94107
(415) 957-0886
$79.95

A-CALC is another unusual-looking spreadsheet, but, unlike **Power Plan**, this uniqueness makes it *easier* to use. Even those who've never used a spreadsheet should have no trouble learning **A-CALC**'s system. Users familiar with Lotus 1-2-3 will take some time to adapt to the differences.

Cells are specified by numbered rows and lettered columns, as usual. **A-CALC** features a "desktop," with icons for a disk drive (loading and saving files), spreadsheet (copying), clipboard and trashcan.

Boxes for selection a wide range of functions are at the top of the desktop, as is a line for the current cell contents.

A small box containing a letter is also present at the top of the screen. This box represents the first major difference between **A-CALC** and other spreadsheets. Instead of using a symbol (such as a plus or equals sign, or an apostrophe) to indicate that cell contents are formulas, values or labels, you change the letter in the box. This can be done either by pressing TAB or clicking on the box to cycle through the possible values. You *do* need to get the type correct, or you could get unexpected results. The four possible letter choices are (V)alue, (F)ormula, (T)ext, and (L)abel. The last two are similar; the difference is only apparent when using named cells.

Cell entries are finished by pressing RETURN, at which time the cursor moves in the direction indicated by the arrow in the box near the top of the screen. The arrow direction always reflects the last direction you moved the cursor, but can be changed by clicking on the box or pressing an arrow key. Only a single row and/or column can be frozen on-screen for titles. Absolute references are designated by use of square brackets.

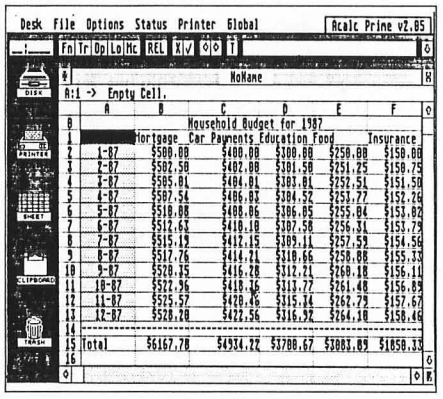
The interface has some features that make it easier to use, as well as some that make it a bit cumbersome. Click-and-drag can be used to select a range, and the window will scroll automatically for selection of a range bigger than that seen on-screen at any one time. Cells can be entered into formulas by clicking on them, and the proper syntax will even be inserted in the formula if you hold the button down. You can also enter a cell into a formula by using SHIFT and the arrow keys to move the pointer to the cell you want.

A-CALC is different from other spreadsheets in its use of the edit mode. In most spreadsheets, you enter this mode automatically when you start typing in a cell. You usually also have the option to enter edit mode to edit cell contents, and pressing RETURN exits edit mode.

In **A-CALC**, pressing RETURN enters the cell contents, but *does not* exit edit mode. You must explicitly exit edit mode by clicking on the edit icon or pressing UNDO. While in edit mode, various keys work differently: you must use SHIFT and the arrow keys instead of just the arrow keys to move the cell pointer, and clicking on a cell enters that cell into the formula—rather than moving the cell pointer to the indicated cell.

All of this takes some adjusting to, especially if you're used to other spreadsheets. Any reference to a cell containing text must be prefaced by $, and numbers less than one must still have a zero in *front* of the decimal point. One very nice feature is the ability to shrink the cells to half their former size, showing four times as many cells on-screen at once. This is especially effective in high resolution.

Maximum size of an **A-CALC** spreadsheet is 8192 rows by 256 columns. Built-in functions include trig, log, ROW, COL, MAX, MIN, SUM, AVG, IF, AND, OR, NOT, date and LOOKUP. IF allows text to be entered in a cell as the result if you preface the IF function with $.



A-CALC Prime.

No financial functions are included, but COPY facilities are very powerful. You can copy a cell to a column or row (or part of each) by clicking and dragging the cell to the row or column header, then filling in the range in a dialog box. One cell may be copied to another or to the clipboard, via click-and-drag. You can also drag a range to the clipboard, which can hold ten separate entries.

Dragging the clipboard icon to a target cell causes it to pop open so you can select the entry to copy. If multiple windows are open, you can drag a cell from the active (top) window to an inactive window. You can replicate one cell to any rectangular range by dragging the cell to the spreadsheet icon, then specifying the range to copy to when the range dialog box pops up. Ranges, rows, columns and cells can be deleted simply by dragging them to the trashcan icon.

Database functions include FIND and REPLACE for values and text, both globally and over a specified range. You can replace all occurrences, move to the next occurrence, or have the program query you at each occurrence. You can sort a range by key row or column, both ascending and descending. **A-CALC** does not allow named ranges. However, if you have cells with data of type *label* (as opposed to data type *text*) in both a row and a column, the cell at the intersection of the row and column can be referred to by a name made up from the text contained in the two label-type cells.

You can read in a spreadsheet from disk and merge it into the resident sheet at the current cursor position. You can load and save DIF files, to communicate with other spreadsheets.

A-CALC's speed is very good. The fastest of the programs looked at here when it comes to copying and recalculating, it's screen scroll speed is satisfactory, too. It recovers quickly from any attempt to move past the edge of the screen.

You're allowed five GEM windows for viewing your single spreadsheet. Formatting flexibility is also quite good. You can set text color, background color, pattern, number of decimal places, format, justification, comma format, degrees/radians for trig functions, and a two-character header or trailer over any range, or globally.

Print formatting is powerful. This program uses printer drivers and supports setting of page width, all borders, titles, single-sheet printing, plus headers and footers. You can embed printer control codes right in a cell, so that different cells can have different print formats. You can even choose the print format you want (including bold, italics, condensed and double-width) from a dialog box; the program will insert the codes for you.

No notepad or calculator is provided, and there's no built-in graphing. You *can* export data to Antic's **A-CHART**, a powerful graphing package.

A very limited macro function is available. Using a dialog box and the cell edit line, you can build your own functions—with up to nine variable parameters, including absolute cell references (but not relative cell references). Macros are stored under the "Macro" box at the top of the screen. You must remember to save your macros, as they aren't saved with the spreadsheet. To use a macro, you drop down the macro menu and click on the one you want, finish filling in the parameters, then press RETURN.

There is no on-line help available, but the manual includes a very good explanation of what a spreadsheet is, an adequate reference section (though there are some confusing parts) and a small tutorial. There is *no* index. This is a real prob-

lem for novices, as you need to search the manual for your answers.

A-CALC Prime lacks financial functions and built-in graphs, so it may not be suitable for certain applications. Aside from this, it's a relatively powerful program with good screen and print formatting, and a somewhat nonstandard interface.

VIP Professional

ISD MARKETING
20 Steelcase Rd., Unit 12
Ontario L3R IB3, Canada
(416) 479-1880
$149.95

The first spreadsheet to arrive for the Atari ST was the non-GEM version of **VIP**. The GEM version is now out and combines more ease of use than the original with remarkable power. The outstanding feature of **VIP Pro** is its total compatibility with Lotus 1-2-3. Users of that program will feel right at home, even as they begin to explore the enhancements.

VIP features the standard GEM menus across the top of the screen, with a control panel at the bottom. This panel has various icons and some boxes for frequently used commands, as well as status indicators for the various spreadsheet functions.

The control panel shows the current cell and its contents. Cell designations are the standard numbered rows and lettered columns. Various symbols (^, ' and ") not only specify that a cell's contents are a label, but also indicate whether the label is to be centered, or right or left justified. The plus sign indicates that a cell whose contents begin with a letter contains a formula. Cell entries can be finished by a RETURN, the arrow keys or any other method of cursor movement—including mouse clicking or use of the scroll controls on the window. Any number of rows or columns can be frozen on-screen to serve as borders. Absolute cell references are indicated by use of the $.

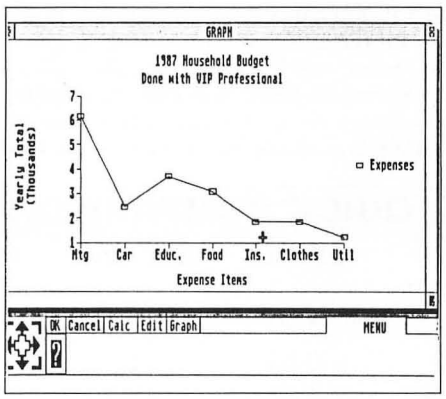
The interface for **VIP** allows an excellent range of choices when it's time to input data. All selections are available from both drop-down menus and the keyboard. With the mouse, clicking on a menu item either selects the item or brings up a submenu. The "desk" and "VIP" menus are only accessible via mouse.

Key selections are virtually identical to 1-2-3. Pressing the / activates menu selection, after which you use the left and right arrow keys to highlight an item. As each menu item is highlighted, submenus belonging to that item appear in the control panel, to let you know what the rest of the choices are. Pressing RETURN selects the item, causing the submenu to appear, if there is one, or causing execution of the selected command.

Once you learn the menu system, it's even faster to simply press the first letter of the menu item you want (*W* for Worksheet, etc.) This causes any submenu to appear, and you just continue pressing the key that corresponds to the first letter of each item you want.

The function keys are assigned ten of the most used functions. These assignments are identical to those of 1-2-3. A cell can be selected for a formula by clicking on it with the mouse, moving the cell pointer to the cell using the arrow keys, or by typing it into the template which appears when the function is chosen.

Selection of a range can be done by click-and-drag, either before or after a command is selected. The window will scroll automatically if you drag the pointer outside its boundaries, so that you can specify a range larger than what will show in the window. You can also select a range by choosing the first cell by one of the methods mentioned above, typing in a period, then selecting the second cell. This, again, is identical to 1-2-3.



VIP Professional.

The maximum size of a **VIP** spreadsheet is 8192 rows by 256 columns. There are a full range of functions, including trig, log, five financial functions, IF, AND, NOT, OR, CHOOSE, STD (standard deviation), VAR (variance), ERR, COUNT, SUM, AVG, MIN, MAX, RAND, LOOKUP, INT, ABS, ROUND, SQUARE ROOT and date functions. In fact, there are more functions built into this program than any of the others looked at here.

The IF function does not allow text. Cells can be copied to a range of cells, columns to a range of columns, and rows to a range of rows.

VIP has a wide range of database functions. The database is set up in columns, each row being a record and each cell a field. The first row of the range holds the field names. You can use all the standard commands for formatting, copying, moving, etc. You can sort on two columns (primary and secondary), in either ascending or descending order. You can find (cursor moves only to selected records, skipping those in between), delete or extract (move to a selected cell range) records which meet the multiple criteria set up in a cell range. Criteria can even use wild cards for more general matching.

You can extract partial records. You can also generate statistics on records matching your criteria—how many there are; their sum, average, standard deviation and variance; and maximum and minimum of a selected field.

There are two types of data tables you can set up. The first calculates the effect of a series of values for an input cell on a series of formulas which depend on the cell. The second calculates the effect of two series of input values on a single formula which depends on both. Data fill will fill a table with values from a chosen start point, using the specified increment or decrement. Data distribution will tell you how many items fall between the stated limits, and how the data is distributed in that range.

VIP will allow named ranges. Working with these is made simpler by the fact that, when you call up the range names, a list of the already defined names is provided.

Two spreadsheets can be merged together with either a direct replacement of one by the other where they overlap, or with the values added or subtracted. No offset is available; the second spreadsheet is loaded at the cell where it was created (normally, A1). **VIP** can export ASCII files for use with a word processor, and can also print to disk. The ST version does not support DIF files.

VIP suffers from a slow-scrolling window, though the problem isn't as severe as in **SwiftCalc**. Hold down an arrow key and bump against the border of the sheet, and the cursor will hang there, beeping madly until the keyboard buffer is exhausted. Hitting CONTROL-UNDO empties the buffer and stops the bell.

VIP has a very fast copy, and its recalculation speed is also respectable. You can set up two windows to view your spreadsheet. As GEM windows, these can be moved, sized, etc.

Any of the multitude of formats can be set up as global or on a chosen range. Formats include: precision, commas, dollars, justification control and format for negative numbers. You can even format text cells to fit within specified margins (with word wrap), allowing you to generate simple text files.

You can print the whole spreadsheet or a range. Borders can be printed on every page, and the headers can be turned off. Printing is only available on Epson-compatible printers, and the initial set-up string is the only printing control available. You can set headers and footers (and number pages in them), and all margins and page length.

No notepad or calculator is provided, though graphing is built in. The graph can be put on-screen in a GEM window, although controls on the window only work if it was brought up via mouse. Graphs include pie charts, bar, stacked bar, line and XY graphs. Labels for graphs are taken from a range on the spreadsheet (you can't type them in), and you can graph up to six different ranges of numbers, complete with six different symbols for legends. You can enter the X, Y and overall graph titles, and scale the Y-axis if you want. Graph definitions are saved along with the spreadsheet data. To print a graph you have created, you must save a special file and use a utility included with the package.

The macros portion of **VIP** is extremely powerful. Complete applications for use by people not familiar with **VIP** are quite possible. Almost every command you can type in can be represented in a macro, as well as cursor controls and the RETURN key. Jumps (similar to GOSUBs and GOTOs in BASIC) add to the flexibility of the macros. You can even test conditions, then branch depending on the result.

VIP makes it possible to set up menus of choices for the user to step through. These work just like the program's regular menus, so they'll be familiar to anyone who's used the program. The macros are activated by a single keypress. For example, to print, you must set PRINT/PRINTER/SET RANGE/SET MARGINS/SET OPTIONS/ALIGN/GO. With a macro, all this could be set up just once, and then you could (for example) simply press ALT-P. You can single-step through macros for debugging purposes, but there is no "learn" mode, where your keystrokes are recorded. Instead, you must program the macro using the macro language.

Help is available from the HELP key, the help icon, or by pressing F1. Aid consists of condensed explanations of commands and directions for use, including templates. The help is context sensitive—what appears depends on what you're trying to do at that moment. If what appears isn't what you want, you can move to other help screens or exit the help facility at any time.

The manual has both good and bad points. There are good explanations of the concept of a spreadsheet, screen layout and the scrolling window. There's also a good tutorial and a flowchart of available commands at the beginning of every chapter, plus an excellent index. However, the manual appears to have been written from a list of proposed features (many of which were never included or were changed), rather than from the finished product.

A whole extra manual is included to note corrections to the main segment, but the bulk of the manual is the introductory chapter, and the correction manual isn't especially accurate or complete. It doesn't include all the errors, nor all references to the errors it does pick up. The main manual makes many statements which are not true or don't work as stated. For example, up/down arrows can't be used to select submenus (which also don't drop

A QUICK COMPARISON

	EZ-CALC	SWIFTCALC	POWER PLAN	A-CALC	VIP PRO
Benchmarks:					
Copy 10x100	1:29	1:54	1:54	0:28	0:23
Recalc 10x100	0:53	0:09	0:08	0:06	0:12
Scroll 100 cells	1:07	4:58	0:43	0:53	3:55
Maximum Rows x Columns	999x300	8192x256	65535x65535	8192x256	8192x256
Number of spreadsheets	1	1	1	1	1
Number of windows	2	2	7	5	2
Notepad	Y	N	Y(4)	N	N
Calculator	Y	N	Y(5)	N	N
Graph built-in	Y	Y	Y	N	Y
Macros	N	N	N	Y(6)	Y
Sideways print	N	Y(3)	N	Y	N
Memory Requirements (1)	(2)	628	806	703	(7)

Notes:
(1) Memory available on a 1-meg machine after load (in K).
(2) No memory status indicator is included in the program.
(3) Must save a special file and exit spreadsheet program to use.
(4) Empty window for notes—not attached to cell.
(5) Empty window for formulas.
(6) Single keypress can substitute for multiple keypresses.
(7) Memory status indicator is relative (percent available).

down when you're using arrows to move among the menus). The mouse cannot be used to select menu titles, only the submenu items themselves, as in a regular GEM menu. While none of these are serious, you do waste time trying to sort out how things actually work.

VIP Professional includes all the features of most powerful spreadsheets, and is 1-2-3 compatible. It's a large program, which really can't be used effectively in a 512K machine.

Conclusion.

So what should you do? Normally, at this point, perhaps in order to keep from antagonizing advertisers, a lot of reviewers will get wishy-washy (I've done it myself) and say something like "only you can decide what's best for you." But this is **ST-Log**; we aren't going to do that.

If you need the power to run a business, don't mind the requirement for 1 megabyte of memory and the relatively high price tag, then I have to recommend **VIP Professional**. The powerful macros, extensive database capabilities and multitude of functions make it an ideal choice for the power user, especially if you're familiar with 1-2-3.

For home use, or for those who must make do with an unmodified 520ST, I recommend either **A-CALC Prime** or **EZ-Calc**. **A-CALC** is a very efficient spreadsheet once you learn its interface, and it is quite fast. **EZ-Calc** is the better choice if you're familiar with other spreadsheets and don't want to learn a new system—or if you must have built-in graphing. It also includes financial functions missing in **A-CALC**. But you really can't go wrong with either of these programs. //

David Plotkin, with his M.S. in Chemical Engineering, is a Project Engineer for Chevron USA in Offshore Engineering. From his first computer, a 32K 400, he has stuck with Atari. He now has a 130XE and a 520ST, which his wife uses in writing children's books, while David enjoys programming and beta-testing.

SCHEDULED ATARI FAIRS

JULY 25 & 26, 1987
CHICAGO, ILLINOIS
—CANCELLED—

AUGUST 28, 29 & 30,1987
DETROIT, MICHIGAN
Southfield Hilton, Southfield, MI
Sponsored by MAGIC, GLASS & GAG
Contact MAGIC at 28111 Imperial Drive, Warren, MI 48093
(313) 751-8291

SEPTEMBER 19 & 20, 1987
GLENDALE, CALIFORNIA
Glendale Civic Center
Sponsored by ACENET
Contact John King Tarpinion at
(818) 760-1831

OCTOBER 10 & 11, 1987
WORCESTER, MASSACHUSETTS
Worcester Centrum
Sponsored by the BCS — contact Alan Glick at (617) 296-8286 or ANALOG Publishing at (617) 892-3488

OCTOBER 24 & 25, 1987
WASHINGTON D.C.
Fairfax High School, Fairfax, VA (near Rt. 50)
Sponsored by NOVATARI
Contact Geogia Weatherhead at
(703) 938-4829

 continued from page 18

and vertical bold, underline, variable italics (from 0 to 75 degrees), even change font styles multiple times on the same line (eight fonts are included). You can create multiple columns or blocks of text that can flow around your pictures, with text automatically flowing from one block to the next. All text and graphics are shown on the screen in WYSIWYG (What You See Is What You Get) fashion.

You may be familiar with GEM's fonts, which are bit-mapped. In contrast, **TGA** uses software fonts that are object oriented. This allows **TGA** to print the text in any size and rotation, and to the best ability of the output device. In addition, the availability of the Font Editor (to create or modify your own fonts, or modify existing ones) means you have a virtually unlimited number of available fonts.

TGA also has an integrated spreadsheet and business graphics in the same environment. The spreadsheet is a type of database for graphic and text information.

The final objective.

TGA can print not only on dot-matrix printers, but also on laser printers and color plotters. It produces a high-resolution output on a dot-matrix printer not available with any other Atari ST package I've seen. This is possible because **TGA** uses a custom technique of overlapping dots not only across, like many others, but also up and down. This results in a dark print, with smooth lines and no streaks. It takes advantage of the best resolution available on laser printers and plotters, also. Thus, even curves and diagonals look smooth on output from any device.

Another unique feature of **TGA** is the device driver editors. If there is no driver included for your device, you can use a device driver editor to make one. You simply have to answer questions about the codes for your device, and the editor makes a driver for you—*no* programming is required. With the release of version 1.52, **TGA** can support virtually any printer or plotter, including wide carriage and 24-pin, except those using a page description language (such as Postscript) and color printers that print in multiple colors (coming later this year).

Fundamentals of the program.

If you have not purchased a copy of **The Graphic Artist**, you can still get the feel of the program by obtaining a copy of the **TGA** demonstration disk from your local Atari dealer or user group. The demo disk is a full-functioning version of the software, except you can't save and print. Any Atari dealer can obtain a *free* copy of the demo disk, either from a PCA distributor or from PCA directly.

From the desktop, double-click on GA.PRG, as you would to start any program. In case you do not have a copy of the program, Figure 1 is a dump of the screen.

The screen is divided into three sections: the Coordinate Location (and message) Line at the top, the Menu and Help Lines at the bottom, and the area in the middle with a grid of dots called the Work Area. You should understand the purpose of each section to get the best use of **TGA**.

When creating a design, you'll be working in a world of X- and Y-coordinates. The top line of the screen changes to show the numeric coordinates of the current location of the cross hair (or pointer). Therefore, we call this line's information the Coordinate Location.

You operate **TGA** through commands (a command tells **TGA** what you want to do) with a "pull-across" menu for selecting these commands. All menu choices are a single word representing a specific command's action(s). The Menu Line displays seven of the possible sixty-one menu choices. The highlight indicates the menu choice to be executed by pressing RETURN. The Help Line displays a description of the highlighted command. Therefore, automatic help is provided for each command.

Menu selection is easy, using the arrow keys to browse and select. Once you're familiar with the menu choices, you can type the first few letters of the item you want, and the highlight will automatically move to that choice. All menu choices are significant within three characters. Once you've selected, press RETURN to execute that menu choice. Press UNDO at any time to abort execution of any command.

The remainder of the screen is the Work Area. Think of this as a movable window or camera lens view over a large piece of paper. You can make the piece of paper essentially any size you require. You view and draw on any portion of your design with an extensive zoom capability. You need a method to determine precisely what portion of the design you are viewing in the Work Area, and to draw items in exact sizes and proportions. To accomplish this, the entire drawing area is mapped to an X,Y-coordinate system. A drawing position is thus expressed as some number of units across and some number of units up. All measurements in **TGA** are in generic units: a foot, an inch, a meter, a kilometer, or whatever you need. Each unit is represented by a pair of numerical coordinate values, seen on the Coordinate Location line and, optionally, as a dot on the grid.

The grid is the set of dots you see in the Work Area. Think it as graph paper under your drawing paper. The grid's sole purpose is to provide a visual guideline to the Work Area. The grid won't appear on your printed output. By default, each dot on it represents one unit. The Grid command lets you change grid spacing, or turn it on or off. Type G and RETURN. Type 1.5 and RETURN; press RETURN again to leave the grid on. Now, a grid dot appears every 1.5 units. You can change the spacing to any value you want.

Last but not least, notice the dotted rectangle in the Work Area. That dotted rectangle outlines your current "view." The current view has a primary function: when you want to print your design, it guarantees that everything inside the dotted rectangle is included in the output. The default view is from position 0, 0 to 8.5, 11, the size and shape of the most commonly used paper. If you turn off the grid, this rectangle outlining your view will disappear, also.

Will I ever start drawing?

As is suggested in this article's title, this has just been the first of a three-part series about **TGA** and how to use it. The information covered in this first part gives only fundamentals. Part 2 will show you how to start creating CAD designs, while Part 3 will guide you through the desktop publishing basics.

This short series of tutorials can only reveal a tiny portion of the power available in **The Graphic Artist**. Therefore, PCA has produced *Tips & Techniques*, which contains an in-depth tutorial and many advanced techniques to help you get the most power (with the least work) from **The Graphic Artist**. Other auxiliary products for **TGA** are available from PCA, including a Font Editor (to create or modify font styles), Electronics Library and Desktop Publishing Library.

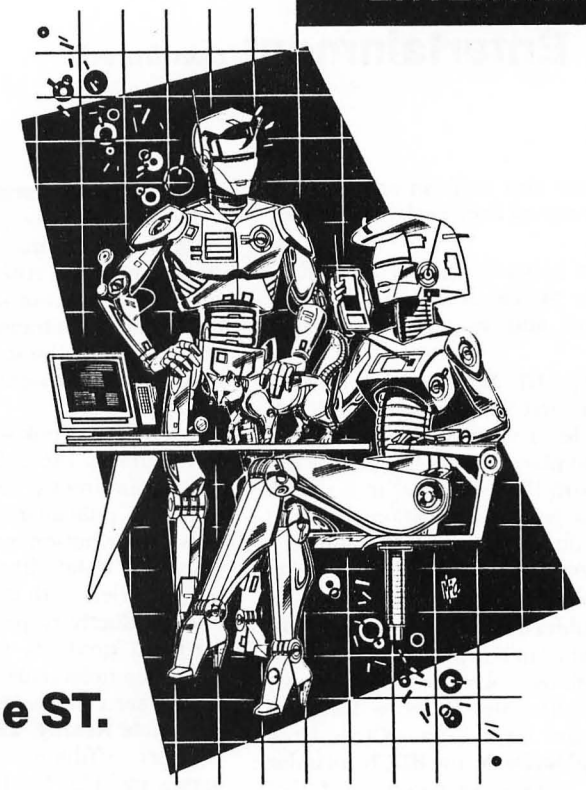
The Graphic Artist (Progressive Computer Applications, Inc., 2002 McAuliffe Drive, Rockville, MD 20851— (301) 340-8398) retails for $199.95. ///

---

*Deborah Elder is currently PCA's Vice President of Marketing and the author of **The Graphic Artist** manual.*

*Peter Naleszkiewicz is PCA's President and designer of **The Graphic Artist**.*

Things to come!



A preview of upcoming entertainment software for the ST.

by Bill Kunkel, Arnie Katz and Joyce Worley

Everyone *loves* coming attractions. How many times have a pair of zippy promotional "trailers" proved far more exciting than both parts of a double-feature; or a 30-second ad for a new TV show turned out to be hotter than the entire series? There's something about a "sneak preview" that gets our juices flowing.

Keeping this in mind, let's cast our gaze forward for a look at some of the ST entertainment software scheduled for retail shelves within the next six months.

MichTron.

In the past, MichTron led the pack in bringing U.S. users the cream of the British ST software crop. Though they now face stiff competition from heavy players like Mindscape, the Michigan-based publisher continues to cherry-pick some extraordinary products from the English orchard.

The hottest new number in the MichTron catalog is Edward Scio's **Airball**, a visually breathtaking variation on **Marble Madness**, in which the player maneuvers a bouncing, rolling ball through some 300 rooms stocked with all manner of eye-popping traps and treasures. Alas, the ball has a slow leak, and it's necessary to refill it periodically from strategically positioned pumps—but be careful not to fill it *too* much, or it goes boom.

The game offers arcade action, strategy (you'll want to locate the spellbook, for example, or at least find a torch to explore dark rooms) and excellent audio, in addition to the visual splendor.

Epyx.

Epyx, now releasing all its games in ST format, has become one of the Big Three publishers in the ST entertainment universe (with MichTron and Mindscape). This means a number of older Epyx titles (**The World's Greatest Football Game**, **The World's Greatest Baseball Game**, **The Movie Monster Game**, **Destroyer**, and **Jet Combat Simulator**) will soon be joining the ST library.

The World's Greatest Baseball Game offers both a stat-based replay and an arcade action game; **The World's Greatest Football Game** is a very sophisticated strategy contest, in which users can design as well as implement a wide variety of offensive and defensive plays.

The Movie Monster Game should be a delight on the ST; users pick a favorite movie monster from an impressive menu that includes a giant tarantula, a flying insect, a gigantic robot, and even old Godzilla himself. Via joystick, the player then guides his titanic troublemaker through any of several major cities (New York, Tokyo, London, etc.), while the army does its best to eliminate the menace. Each monster has its own special power (Godzilla's flame breath, the tarantula's web, etc.), in addition to the usual people-stomping abilities that are generic to all behemoths.

Destroyer and **Jet Combat Simulator** are a pair of high-quality (not to mention self-descriptive) tactical combat contests that can only benefit from enhanced ST graphics.

In addition, several *new* titles will be appearing on Our Favorite System, including **Street Sports Baseball**, **Street Sports Basketball** and **California Games**. The "Street Sports" series is a new line of athletic simulations that eschews the pro-sports milieu for the sandlot (**SS Baseball**) and schoolyard (**SS Basketball**) environments. **California Games**, the latest entry in the "Games" series, brings surfing, skateboarding, roller skating, hackey-sack and flying disc (known as a "frisbee" in the world of registered trademarks) to the ST.

Epyx has also announced two new lines: "The Masters Collection" and the "Maxx Out" series. The Masters Collection currently includes **Omnicron Conspiracy** and **Sub Battle Simulator**, and is geared toward "advanced" game players. **Omnicron** casts the player as "Ace" Powers, Captain in the Star Police, a group of interplanetary cops under the direction of **Omnicron**, a planet-spanning Artificial Intelligence.

The player uses the resources of **Omnicron**, as well as P.A.L., his companion droid, to save the universe. The game, which spans three planets and eight scenario levels, was produced by First Star, written by your humble authors and programmed by Jim Nangano.

Sub Battle Simulator is a circa WWII contest, in which the player opts to command any of six U.S. submarines or Ger-

man U-boats and perform any of sixty missions, ranging from seek-and-destroy to patrols.

The Epyx Maxx-Out line is oriented toward 10- to 14-year-olds, has a high action content and very accessible play features.

Spy vs. Spy III: Arctic Antics, third in the popular First Star series, is chock full of cartoon-level violence and diabolical traps, as the players contest with one another (or with the computer) in a search for the space helmet, navigation chart and uranium cube they need to launch a single-seater rocketship and escape the frozen wastes.

The **BoulderDash Construction Kit**, also by First Star, lets players run wild in creating their own playfields for Rockford, the player-surrogate, to move through. Based on the Peter Leipus-Chris Grey classic, **BoulderDash**, the **BDCK** includes fifteen new, preprogrammed mazes full of rocks, diamonds, butterflies and magical bricks. **Road Warrior**, meanwhile, takes the look and feel of an earlier Epyx classic, **Impossible Mission**, and brings it into the future. Here, users must wade through mutated forests, brave a radioactive volcano and locate a secret generator room, in an ongoing battle with those ever-popular radioactive mutants.

SSI.

SSI, a publisher once known primarily for its wargames, has developed an impressive reputation for high-quality fantasy role-playing contests in recent years. These games have been especially rewarding in their ST incarnations, with titles like **Phantasie** and **Phantasie II** taking special advantage of the ST's superior graphic capabilities.

Fans of those games will be delighted to learn that the third title in the "Phantasie" series, **The Wrath of Nikademus**, should be available as you read this. Fantasy/role-playing mavens can also look forward to ST versions of **Wizard's Crown** and **Rings of Zilfin**, role-playing adventures which share the same magic, mazes and monsters genre, as does **Shard of Spring**, an ST version of which is also in the works.

Car-war fanatics who were driven to ecstasy by **Roadwar 2000** can look forward to a sequel, currently in development, set in a post-armageddon Europe. And on the licensing front, SSI's recent acquisition of computer rights to TSR's "Dungeons & Dragons" may well be the software publishing Deal of the Year.

Electronic Arts and Affiliated Labels.

EA has been the quirkiest of the major publishers in terms of its ST releases. In its wholehearted rush to embrace the Amiga, EA's turned out *very* few ST products, and when it has translated programs from other formats, the selections (like **Financial Cookbook**) were curious, to say the least.

EA recently took steps to plug this hole in its catalog, through direct and indirect means. Indirectly, EA added a couple of major ST publishers to its "Affiliated Labels" distribution network: IntelliCreations (aka DataSoft) and Batteries Included. Batteries, with its excellent creativity and productivity products (**DEGAS**, **PaperClip**) products for Atari computers nicely complements DataSoft's first-rate XE/ST entertainment catalog (**Mercenary**, **Alternate Reality**, **221B Baker Street**).

Other Affiliated Labels software releases include: **Gridiron**, a strategy oriented football simulation; **Quizam**, an offbeat trivia contest from Interstel; and **Mad Libs**, an education/entertainment program based on the popular word-insertion game from the 60s.

But EA isn't leaving the field *entirely* to its affiliates. The vastly popular **Bard's Tale**, a role-playing quest contest, will be turning up in ST format, as will the red hot **Autoduel**, from Origins. **Autoduel** is the computer version of "Car Wars," the popular nonelectronic contest in which players earn money (by contesting in local arenas or taking dangerous courier runs) to build bigger, better and *badder* vehicles with which to rule the road. Also coming: **Ring Quest** and the **Music Construction Set**, as EA gets down and gets funky with the ST!

Mindscape.

Mindscape is solidly behind the ST, even supporting the computer with a pair of products that will be published exclusively for the system. **Plutos** is an action-arcade game of historical interest as one of the very few (only?) computer games ever to feature a dedication (programmer/designer "Dr. J" created this game "for Sarah"). This SF shootout is otherwise pretty pedestrian, but it does boast coin-op quality visuals.

Mindscape's second ST-only title is a sprite of a different color. **Q-Ball** is one of only a handful of genuinely original concepts to turn up in the action-strategy genre. **Q-Ball** is, put simply, three-dimensional pool. The balls hang suspended inside a large cube, which the player can rotate freely, even when the balls are in motion! The player aims the cue ball, sets the power of the shot and determines the "friction" (the amount of resistance which exists within the cube), then shoots. The balls bounce madly in all directions, like rolling jelly beans in a jug of glycerine, and might even occasionally fall into one of the pockets, located at the upper and lower edges of the cube. This is one *hard* game. As two-dimensional thinkers in a three-dimensional universe, it takes most players quite a while to adjust to this game's "depth," but the struggle is worth it. **Q-Ball** is refreshing and innovative, with beautiful graphics and an easy-to-use play system.

Mindscape will also produce an ST version of **Into the Eagle's Nest**, a state-of-the-art update of a theme explored several years ago in the Muse classic, **Castle Wolfenstein**. The gamer is cast in the role of WWII commando, attempting to rescue a trio of Allied saboteurs from the infamous **Eagle's Nest**, a Nazi castle fortress deep behind Axis lines. The **Nest** has four floors, connected by elevator, and the player maneuvers his on-screen surrogate via joystick. The stunning, over-head perspective graphics and super sound effects would look at home in any arcade, but **Eagle's Nest** also adds a strategic depth through objects, salted about the castle, which are invaluable in completing the mission.

Accolade.

Accolade will be releasing a new entertainment program, and an old favorite, in ST format within the coming months. **Hardball!**, a visually impressive recreation of the batter vs. pitcher confrontation (the rest of the teams are there, too, but they hardly matter) will finally do its ST turn, as will **Test Drive**, one of the year's more unusual entertainment programs.

Test Drive puts you behind the wheel of one of several super bad sports cars (a Ferrari 398GTS or Lamborghini) and lets you work it out on a treacherous mountain road. Great sound and graphics make this a possible sleeper hit.

Activision.

Tass Times in Tonetown, a cult classic on the Amiga and Mac earlier this year, will be turning up on the ST, as will **Championship Baseball** (the revamped version of Gamestar's **Star League Baseball**).

Tass Times uses an experimental player interface with lots of icons and audio effects, to create an offbeat adventure. **Championship Baseball** is an excellent

action-strategy hardball simulation that employs a split screen to display both behind-the-plate visuals for pitcher-batter confrontations and full field views for fielding.

Firebird.

Firebird, who astonished the ST universe last year with **The Pawn**, may do it again this year with **Universal Military Simulator** (**U.M.S.**) by American designer Ezra Sidrah.

A whole new approach to strategy oriented computer wargames, **U.M.S.** takes such contests in a totally different direction from the standard SSI, Avalon-Hill approach. The game has a three-quarters visual perspective, and makes inspired use of icons and user-friendly GEM-style menus. This results in games which are more accessible to both grizzled veterans and green recruits.

Miscellany.

Gunship, the excellent fighter helicopter simulation from MicroProse, will be turning up any day now; a company named Capcom is promising a line of sports games for the ST; and Hybrid Arts is offering a special product from Xanth F/X, **MIDI-MAZE**, perhaps the ultimate in maze-chase games. The players move through a first-person perspective macromaze, obliterating floating smiley faces. What's *really* special about **MIDI-MAZE** is that up to sixteen people can play the game simultaneously, on sixteen different STs, through the MIDI interface. Now this is, admittedly, a somewhat eccentric idea, and it's hard to know how often the opportunity to link up sixteen STs will come along, but when it does, you'll be ready.

Heck, any game that uses smiley faces for targets is worth owning, even if you play it solitaire. //

Wanderer

by Beatrice and Jean-Luc Langlois
PYRAMIDE/EIDERSOFT USA INC.
P.O. Box 288
Burgettstown, PA 15021
(412) 947-3739
Low resolution $39.95

by D.F. Scott

The following is a scenario for an actual game, **Wanderer**. None of the events depicted here have been exaggerated or falsified.

You're traveling through the galaxy, subsisting on the basic material for survival —money. Space has been mapped conveniently into a 7x7 rectangular grid, wherein the planets have been made to hold still. In order to gain fame and acquire money, you must be victorious in battle against countless alien space vessels: starships, pyramids, cloverleafs, battering rams, box-kites, dogs, bicycles and Spy-vs.-Spy-type characters.

Having achieved a modicum of fame, you make your way to a hostile planet. It sends out its best defenses (champagne-glass-shaped missiles) to atomize you, and you blow them away with your torpedo cannon. The planet, convinced of your intentions, is willing to deal.

It has a pair of fours, a king, an ace and a seven. You have a king and a three (as a lowly pilot, you're only allowed two cards), but you'd really like that ace. You convince the planet that, in order to achieve its goal of global democracy and peace, it would be much better off with two kings—it'd have two pair. Since you were kind enough to help make the planet a better, safer, happier world—and to give it a decent hand—the planet pays you a bonus, in the accepted monetary currency of the galaxy: cats.

Live cats. It has, understandably, been difficult for the galaxy to develop a wallet or purse in which to keep the live cash, and galactic banks have had to make pleas for food donations. Still, cats were the only items everyone could agree to place a value on. So you're flying around, shooting up box-kites, trading kings for aces and collecting cats.

Why? Folks, let's face it, this isn't a job we'd write home about. There are really two reasons for continuing: first, in ways psychiatrically inexplicable, you're having fun; second, you must earn enough ability points to fly through the black holes, while risking your rear to obtain jokers. The galaxy loves jokers, and will pay you plenty in exchange for them.

If you can earn 8000 cats—or hold a five-of-a-kind with aces—you win the right to do battle with "The Ark" and "The Sphinx." You want to defeat them, so you can retrieve your landlady's prize cat before she evicts you.

And that is the premise of **Wanderer**, a most appropriately titled game. Besides being the first space game to use "live" cash, it also holds the distinction of being the first 3D ST game. Since it uses the red/blue method made famous in the 50s by movies with names like "King Kong vs. Raymond Burr," a color monitor is required.

If you've ever viewed the world through rose-colored glasses, you'll remember how normally-red roses appear white, and red printing on paper becomes invisible, whereas blue printing shows up as extra dark. A similar effect occurs when you're wearing the famous red/blue 3D glasses, though the eye with the blue filter sees red printing as extra dark and filters out all blue printing.

So, when viewing a picture made with only red and blue inks—or, in **Wanderer**'s case, graphics produced with only the red and blue beams of the screen's electron gun—each eye sees separate images, since half of the picture has been filtered out. The two resulting images are similar, but in different positions, forcing the eyes to converge to see one clear picture. In this convergence the three-dimensional effect is simulated.

Consider undertaking a programming feat such as this: plotting two simultaneously simulated vector-graphics images, where the three-dimensional coordinates of the vertices of *every* object *and* the stars in the background are calculated and plotted (by my estimate) sixteen times per second. Each view is 32K in pixel length, and intersecting pixels are replotted in magenta. All this is taking place while the game scenario is continually updated. And still the program's length is kept to *just over 32K of code.*

Unfortunately, there's a bad side: the three-dimensional effect doesn't work for everyone. To simulate depth, **Wanderer** employs a window frame effect, so that the floating targets appear far away. Along the edge of this frame is the radar, an absolute necessity because you must see

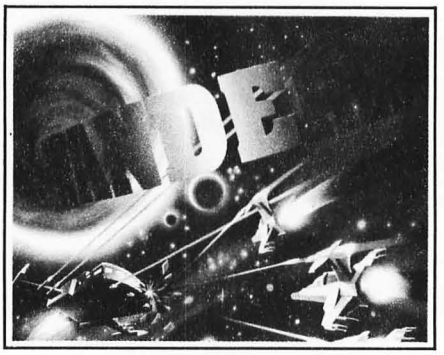
what's coming up behind you. The red and blue images of both the frame and the radar were plotted on the *wrong sides*, so no one's eyes—mine included—will be able to see the entire screen as one clear picture. Some people's eyes won't be tricked into seeing any 3D effect at all.

Wanderer is entirely joystick controlled; you can sit at a distance so your eyes don't burn, since the keyboard need not be touched.

The galaxy map shows planets, black holes (I call them "vortices," since they're no blacker than the rest of space), and the sector where the prize cat lies. Blank sectors on the map actually contain mystery targets, which can be defeated to increase your ability level. Such rewards, besides improving your movement potential, are the keys to entering the vortices and earning jokers. With every joker earned, your hand increases by one card (remember, you need five to have a hand of your own).

As you may have surmised, being a **Wanderer** pilot requires a frame of mind that's a cross between Luke Skywalker and Wink Martindale. After defeating a planet's defenses, you'll see a list of locations of all the cards in the galaxy. There are ten planets with five cards each; add your initial two and that makes fifty-two. (No wonder the galaxy is in turmoil; there's only one deck of cards to go around!) The best-developed flight plan is a sort of "whistle-stop" tour, trading high value cards for those another planet in the tour might want. I say "might" because some planets are stubborn and would rather hold one ace and garbage than three-of-a-kind.

Cats are used (sounds cruel, but it's true) to buy fuel and shield power from a planet. You can hold a maximum of nine shields, good for one weapon hit or collision each. If you're hit with no shields, naturally, you die. Here, **Wanderer** gains yet another distinction: it's the first to offer an *afterlife*.



Wanderer.

In the afterlife—"limbo," as the handbook has it—you see your wedge-shaped ship, a hint of its former self, floating in a cramped *white* tunnel. The object is to destroy every dark target you see. If you do, you're *resurrected* at one lower ability level; if you're hit, you'll have died twice, this time for good. The extended form of dying is bad for your ego.

As for the promised final battle between you, The Ark and The Sphinx, there isn't much to say. . . I haven't gotten there yet. Sorry, folks, I tried for hours to reach 8000 cats or ability level 10 with five aces—with absolutely no luck. In fact, the scoring system is such that, if you start to do badly, you *recede* to lower ability levels, sort of a reverse Peter Principle: the more you lose, the more ground you lose toward your final goal.

The sound effects during battle and movement phases are quite adequate; but if the authors were interested in realism throughout (and not in conserving RAM), they could've added some equally realistic sound during the card-trading portion of the game, preferably an excited audience shouting, "Go for the *king!* Give up the *ace!*"

Having played **Wanderer** for several hours, I have yet to see all the enemy targets; there may be hundreds, some of which can be modified by the program to counter the pilot's ability. Each enemy vessel possesses its own unique, often strange, strategy—whether it be using a cloaking device, splitting into several targets, ejecting smart shuttles, having one side that's indestructible, deploying electric force screens, laying mines, or simply heading kamikaze-style for your nose.

Even with its undeniable silliness of premise, this is a brilliantly executed game. It begins looking like a warm-up excursion, then, like a sea anemone, it closes in on the mind, trapping it into the need to press on. The pilot soon feels he's systematically disarming a maniacal galactic dictator, whether it be a Blofeld seated in a safe steel enclosure petting the prize white Persian cat (the object of envy throughout the galaxy), or an Alex Trebek categorically decimating your patience and willpower till you'd risk life and pixel limb for five easy aces.

Prospective galactic freelancer, *come on down!* Pick up your phaser, your pilot's license; pick any two cards from the deck; have a seat; put on those dumb-looking glasses; and lose yourself in a vortex searching for a cat. After all, what's an ST for, anyway? ///

Harrier Strike Mission

by Tim Hays
MILES COMPUTING
7741 Alabama Ave., Ste. 2
Canoga Park, CA 91304
Low resolution $39.95

by D.F. Scott

The closest I've ever been to a Harrier was when I heard one over the BBC via shortwave, attacking Argentine fortifications during the Falklands War. I assure you, **Harrier Strike Mission** hasn't taken me any closer than that.

Here's the objective: you're piloting an AV-8B Harrier aircraft which begins its existence atop an aircraft carrier. Three miles away is an island whose main geographical feature may as well be called the "Blue Glass Mountains." Its defenses consist of one airbase and a thing that looks like a tent, which is called the "tank barrack." Your mission is simple: blow everything up.

When I say, "Your mission is simple," I'm in no way overstating the facts. You take off, go over to the island, blow everything up and come home. The casual game player, I estimate, will be able to master this game in expert mode in *one hour's time;* and when she does so, she'll notice it took her all of 90 seconds to complete the mission. In fact, it's quite possi-

ble to blow everything up *without leaving the safety of the aircraft carrier.*

Aircraft fans of all ages will instantly recall that the Harrier is the jet able to take off straight up. They'll also recall that a mouse isn't used to fly the jet (many are quite thankful for that), and taking off is a bit more difficult than being seated and telling the jet to take off. But, unless the Army has seriously deployed an arsenal with its own telepathic guidance system, this weaponry in **Harrier** is *unreal*, possessing a tendency to hit from *miles* away.

The game's graphics are in first-person format, with a view from the cockpit window and the instrumentation along the window frame. The opening selection menu allows you to choose game options: novice or expert flying mode (virtually identical); day or night mission (night offers a black sky, day a *white* sky); and the choice of whether or not to add shading to the ground and enemy planes—in other words, whether to sacrifice objectivity for speed.

In comparison to many simulated vector graphics games, which use their plotting algorithms very effectively, and with respect to the small amount of objects on **Harrier**'s screen at any time, the graphics here are needlessly slow, with a frame-redraw rate that at times approaches 1.5 frames per second. In a sense, the shade/no-shade option is a shy computational apology for this fact.

The aircraft is equipped with what's described in the "Operations Manual" as a 30mm Aden Cannon—what might better be described as a one-hit-takes-all object incinerator. It also has three AIM-9 Sidewinder heat-seeking missiles which, despite their title, aim themselves. Once deployed at any angle, they *will* blow up the nearest enemy plane, without fail. Thus, we have the classic 1980s dogfight: there's the enemy plane, press the *M* key and blow it up.

It's only fair to point out that the documentation does go into a fair amount of detail about the operation of a Harrier, including what the different parts are called, how they work, and the game procedure. You could even say the manual was educational; I only wish the game itself went into as much detail as the docs.

The only real challenge in **Harrier** is evading the enemy's heat-seeking missiles. Since they can't be hit with your cannon fire, you have the choice of either outrunning the missile—and wasting time—or deploying a flare and watching the missile seek it like a blind, Italian tenor in an opera dissuaded by a whiff of provolone emanating from the orchestra pit. You have sixteen flares, which, in themselves, could start a war.

The island has five planes and ten tanks, but only deploys one of each at any time. Each uses missiles you could fool with a flare bought one July 4th at a corner fireworks stand; your ammunition could pierce a nuclear fallout shelter *after* blowing apart the walls of Fort Knox.

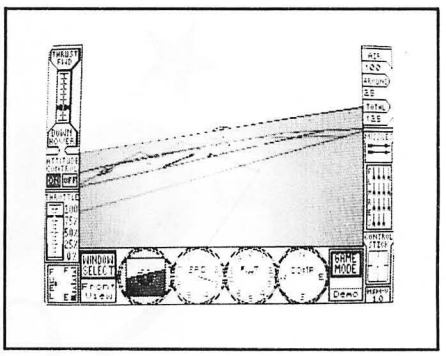
One earns points for each item blown up. Since there can only be one tank on the ground and one plane in the air, the rest of the planes wait inside the hangar—which is one target—and the tanks, within their barracks. Both barracks and hangar could easily be decimated for points in the first 30 seconds; but all the remaining undeployed tanks and planes are then destroyed *without* points awarded. So, if you like big numbers, the best strategy is to wait around and make sure a stray bullet doesn't hit the enemy base just yet.

The game is also equipped with an overhead "map" view with adjustable magnification via the plus and minus keys *à la* **Flight Simulator II**. You could fly with map mode on all the time and judge your altitude with the artificial horizon—without sacrificing an iota of what realism there is.

Flying the plane via mouse is more like rowing. The screen instrumentation includes a simulated control stick that moves as your plane does. You have to wonder why the author went to the trouble of simulating a control stick when most computer owners have instantaneous access to the real thing.

The sound in this game does indeed simulate the earth-shattering and exhilarating whine felt when switching on the power of the mighty Electrolux vacuum cleaner.

Computer games have reached a development stage where originality of theme and brilliance of execution qualifies the game as "good." **Harrier** is a throwback to older games we've mastered already, become too skilled for. I get the feeling a mere hour's discussion over the phone with Bruce Artwick or Jez San would've improved this game's concept and execution. As it stands now, the game is too simple and too common. Instead of offering commodities to game consumers who can "use a game up" in a day and still be ready to spend fifty bucks on what's next, Miles Computing should consult the computer artists—then take a few months off, clear its heads, and meditate upon the concepts of uncommonness and brilliance. //



Harrier Strike Mission.

D. F. Scott is an artist, writer, educator and programmer living in Oklahoma City. He is currently engaged in the study of quantum physics, computing and other ways in which elementary particles interact with each other. Otherwise, he fills infinite pieces of paper.

FOUR-STAR Software

Our choices for the ST.

compiled by Lee H. Pappas

The Best. That's what this is all about. I asked some of the leading experts in the Atari world what their favorite software is—not their favorite game, not their favorite language, and not their most prized utility. I asked each of them to give me a *list* of their favorite games, languages, utilities, telecommunications software, spreadsheets, databases, word processors, you name it.

The result is a list of products with accompanying remarks. What makes a good piece of software? That all depends on what an individual is looking for. Certainly, being bug-free is very important. Other necessary features seem to include: ease of use, usefulness (the fun factor or endurance, for games), and plenty of options or capabilities.

There's a goodly amount of software support for the ST, and some heavy-hitting products are on the way. Desktop publishing on the ST is still in its infancy. Some of the more well known applications software seen in the IBM market should start to find their way into the ST world, also.

Finally, as we mentioned in the comparable software picks in **ANALOG Computing**, we would be *most* interested to hear your opinions on this list—in agreement or not. We hope you get as much enjoyment out of reading through this list as we did composing it.

The categories are:

Applications . . . . . Spreadsheets, databases, desktop publishing, etc.
Entertainment . . . . . Our favorite games.
Graphics . . . . . Products to bring out that special creativity.
Languages . . . . . Main tool of programmers.
Telecommunications . . . . . Terminal software.
Word Processors . . . . . **ST-Log**'s writers' mainstays.
In addition . . . . . Assorted software.
Hardware . . . . . What we like for add-ons.

The players are:

Charles Bachand (CB) . . . . . Atari programmer for eight years; ANALOG Publishing staffer for seven years.
Ian Chadwick (IC) . . . . . Author of *Mapping the Atari* and software documentation; **ST-Log** Contributing Editor.
Frank Cohen (FC) . . . . . Atari programmer; author of **Regent Base**.
Michael DesChenes (MD) . . . . . Atari user for seven years; a Publisher of **ANALOG Computing** and **ST-Log**.
Tom Hudson (TH) . . . . . Atari Programmer for seven years; author of **DEGAS**.
Charles F. Johnson (CFJ) . . . . . Professional musician and our new West Coast Editor.
Arthur Leyenberger (AL) . . . . . Human factors psychologist, five-year Atari user and our East Coast Editor.
Maurice Molyneaux (MM) . . . . . **ST-Log** Contributing Editor and computer artist.
Lee H. Pappas (LHP) . . . . . Atari user for eight years; a Publisher of **ANALOG Computing** and **ST-Log**.
Steve Panak (SP) . . . . . Frequent contributor to **ST-Log** and author of **Panak strikes!** every month in **ANALOG Computing**.
Matthew J.W. Ratcliff (MR) . . . . . Electrical Engineer for McDonnell Douglas; our Midwest Editor.
D.F. Scott (DFS) . . . . . Contributing Editor and ace reporter.
Clayton Walnum (CW) . . . . . Veteran Atari programmer and Technical Editor extraordinaire.

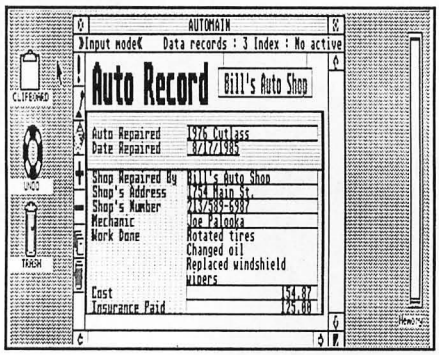
Four-Star Software *continued*

APPLICATIONS

☆**DataRetrieve** (Abacus) — An easy-to-use database (formerly called **FilePro ST**), which offers drop-down menus, search and sort, the ability to access up to four files at once, and screen templates.

"Although not without problems, this program is very easy to use, especially if your database needs aren't particularly demanding." —IC

"A database program that doesn't require a degree in engineering with a minor in mathematics to use." —CW



DataRetrieve.

☆**DCopy** (Shareware) — This program by Ralph Walden allows you to archive and copy files, get a directory, erase or format disks, hide or extract files, perform a directory search, and lots more.

". . . programmer's answer to GEMophobia. For people who loved 8-bit DOS 2.5." —DFS

☆**Easy Draw** (Migraph) — Design artwork or complex illustrations with this powerful program—or add text, to create your own desktop publishing system, for newsletters, ads, flyers, and more.

*"**Easy Draw** provides more flexibility [than **Publishing Partner**] in creating line drawings with your documents, and it has more fonts available, since it uses GDOS." —MR*

☆**EZ-Calc** (Royal Software) — A fully implemented, GEM-based spreadsheet, with easy-to-use mouse control and built-in calculator.

*"While you're waiting for **VIP** to redraw the screen, **EZ-Calc** has finished the same functions in half the speed. **EZ-Calc** doesn't support all of **VIP**'s functions, but is easy to learn and very versatile." —FC*

"It's a well-designed, bug-free, GEM-based spreadsheet with many features and options. I did my 1986 taxes with this one." —IC

☆**GFA Vector** (MichTron) — This program from West Germany supports GFA BASIC by permitting the user to create three-dimensional images and add them into programs. Animation via rotation on any of three axes is one of its many features.

*"**Starglider**-like graphics offered to the public on a silver platter; how could we refuse?" —DFS*

☆**HabaView** (Haba Systems) — File manager with GEM compatibility and the capability to arrange data in any readable format you choose, as well as sorting, user-definable list and label formats, and easy changes to data.

"Of all the file managers out there, this was one of the first and remains one of the easiest to use, with the least overhead involved in creating and manipulating data." —FC

☆**Publishing Partner** (SoftLogik) — The first and most popular desktop publishing system for the ST, this product is excellent for producing newsletters, forms—anything and everything requiring text and graphics.

*"The first fully-implemented desktop publishing program for the ST, **Publishing Partner** does everything the Mac programs do, and more." —CFJ*

"It's truly impressive that this program was developed in such a relatively short period of time. And what you can do with it is even more so. If you want to get into desktop publishing, this is the essential program." —IC

"This is a great program, especially since they had to do everything themselves (no GEM BIOS). . ." —AL

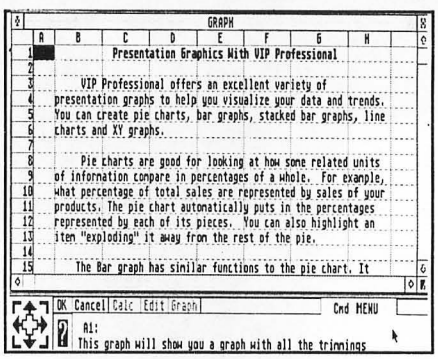
"A very impressive piece of desktop publishing software. . . particularly considering it is primarily the work of one person (Mr. Deron Kazmaier)!" —MM

". . . the perfect application for mixing text and graphic art simultaneously." —MR

"Newsletter editors never had it so good." —CW

☆**Regent Base** (Regent Software) — A full-function relational database with mail merge, GEM compatibility, checkbook manager, and more.

"When people realize just what SQL is, their opinion of the ST skyrockets. This program itself justifies the ST in business." —DFS



VIP Professional.

☆**VIP Professional** — Spreadsheet with many easy-to-use features, and a version with full GEM interface.

"A good Lotus 1-2-3 spreadsheet clone." —AL

"The GEM version is easy to use and fills a large variety of my needs." —LHP

"There are so many features and options, I hardly even notice that it beats me at the lowest level." —IC

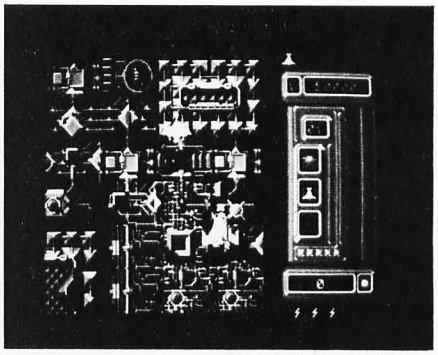
☆**Flight Simulator II** (subLOGIC) — The most famous of the flight simulators, with all the features you expect and can imagine. In addition, twelve extra disks are available, containing mapped-out regions of the entire continental U.S.

*"By far the best version of **FSII** ever released on any computer system." —IC*

"This may be as close as some of us will ever come to flying a plane (or a Learjet for that matter!) The graphics of the ST version make its 8-bit cousin look anemic and slow." —CFJ

☆**GATO** (Spectrum Holobyte) — Sub warfare in dangerous waters of enemy freighters, destroyers and patrol boats. The goal is to wipe out all the ships in the large database, using mines and torpedoes. Also, you must protect your sub tender and venture out on the many assigned missions, day or night.

"A good combination of strategy and action, with some neat features and options. The ST version is several leagues beyond the IBM version." —LHP



Goldrunner.

☆**Goldrunner** (MichTron) — Battle the Ring Worlds of Triton in this mega-arcade game. This game only promises great arcade shoot-'em-up action . . . and delivers.

*"Finally, the cure for fanatic **Galaga** players. It also gets my vote for best scrolling." —DFS*

☆**High Roller** (Mindscape) — A combat simulation that wisely spends more time on combat than on simulation.

"At last, an airplane game that remembers how to go on maneuvers." —DFS

☆**Home Casino Poker** (Dubl Dubl Funware) — Choose between Draw Poker, that contains the best features of the coin-op machines, or a one- to four-player version of Stud Poker.

"I can't believe the work they went through for a card game. The options, menus and graphics are the best I've seen." —MD

☆**Infocom Games** — These products have almost universally become the standard, leaders in text "adventures." Their advanced text parsers, complex plots and innovative packaging (usually essential to game play) have

ENTERTAINMENT

☆**Chessmaster 2000** (Electronic Arts) — A tough chess game, packing a ton of features.

made them favorites for years. Over twenty titles now exist in the series.

Bureaucracy: *"This work of interactive-fiction-plus turns the power of your ST against you, tying a frustrating Gordian knot so complex only 'restart' can release you." —SP*

"Infocom's fine text adventures give your imagination a real workout. So who needs graphics?" —CW

Hitchhiker's Guide to the Universe: *"I've never had so much fun looking into a screen full of plain ol' text. No graphics, no sound effects, just challenging fun." —FC*

☆**International Karate** (Andromeda, System 3) — Like ***World Karate Championship*** from Epyx; basically, this is the version sold in the U.K.

"Takes you through four continents of high-contact karate. The excellent blend of animation, sound effects and music rivals most arcade karate games (and each game doesn't *cost 25 cents!)" —FC*

☆**Mean 18** (Accolade) — Real golf simulation using colorful graphics and allowing up to four players. Select from many courses based on real ones thoughout the world. You can also design your own fairways—with sand traps, water hazards, and much more.

"From the first tee to the eighteenth green, as real as it gets. And it never rains." —SP

"Another fine product from Accolade. . . even a help to me in the real game (I'm down almost 100. . .on nine holes!)" —LHP

☆**MIDI-MAZE** (Hybrid Arts) — A new game that gives up to sixteen players, on sixteen separate STs connected via MIDI ports, the ability to play each other in an arena/maze. Different-colored "happy faces" roam through the maze in search of others. You're one of these, the object being to shoot any of the other guys. One-player mode is also possible, against computer-controlled faces, and many other options are available, including designing your own mazes.

"The most exciting game available on the ST. This program has so many possibilities in terms of play combinations, it's unbelievable." —TH

"I don't even have a copy of this (yet), but in my limited acquaintance it was love at first byte. . .so to speak. Original and intriguing multi-ST/player game." —MM

"The ultimate in game play. . .you aren't just annihilating images on a screen, but the person sitting beside you!" —LHP

☆**Monopoly** (public domain) — This computer version of the most famous board game comes to you via compiled GFA BASIC. Everything is there, in the proper places and the right colors. One player against up to three computer opponents.

"I never liked the board game because of the time to set it up. This is perfect!" —MD

"Great fun to play. . .you'd never think this game was written in BASIC—and public domain!" —LHP

☆**The Pawn** (Rainbird/Firebird) — A complex tale, told and played with innovative graphics and text. Colorful, detailed graphics enhance the advanced language parser.

"WOW!" —AL

☆**Phantasie** (SSI) — A graphics and text multicharacter role-playing game, where you command a group of one to six characters—a wizard, thief, priest, fighter, and so on. These entities need not be human, but could be elves, orcs, lizard men, or whatever.

"The fact that the original already has two sequels is the best testament to the addictive nature of this game. The best D&D-type game so far." —SP

☆**Shanghai** (Activision) — This 3000-year-old game is brought to you on the computer. From 144 random picture-tiles of seven different suits, you must match tiles in a certain way and plan out a strategy up to twenty moves ahead.

"This gets loaded at 6 o'clock each and every night." —CB

☆**Silent Service** (MicroProse) — World War II battle simulation aboard a submarine in the South Pacific. Varied screens display the control room, periscope, engine room, maps, and more.

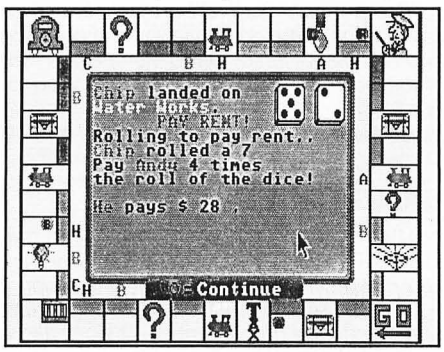
"The best submarine simulation I've ever played. You know a sub simulation is realistic when you can imagine yourself on the ocean bottom, looking up at the ceiling of your sub while being depth-charged by enemy destroyers. A real white-knuckler." —TH

☆**Skyfox** (Electronic Arts) — Pilot your fighter against enemy craft.

"Although we've only seen the tip of the iceberg as far as arcade games on the ST—dynamite graphics, fast action and complexity will keep this one around for a long time." —SP

☆**Starglider** (Rainbird/Firebird) — A high-quality 3D aerial combat flight simulator, where you must wipe out a multitude of ground and air forces, as well as keep an eye on your energy and defense systems. Music, voice synthesis, 3D wireframe graphics, color or monochrome options are all here.

"The ST equivalent of the 8-bit ***Star Raiders****. A realistic spaceflight simulation, I sometimes find myself trying to dodge missiles while sitting in my chair." —TH*



Monopoly.

"Amazingly fast vector-style 3D graphics, very responsive controls. A tour-de-force, definitely one of the best ST games." —CFJ

"This has got to be one of the hottest action games I've ever seen! The closest thing to a real arcade game yet." —MM

"One of the incredible games on this machine that stands out far ahead of the rest. My only gripe: it's too difficult in the early levels." —LHP

"The new king of games; every bit as good as, if not better than any vector-graphics coin-op anywhere." —DFS

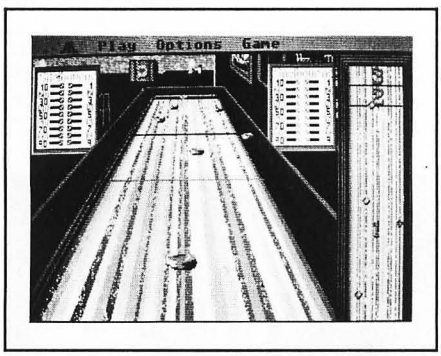
"At last! My own airborne Ground Attack Vehicle. Boy, are the neighbors jealous now!" —CW

☆**Star Raiders ST** (Atari) — The ST version of the most famous Atari 8-bit game. Battle Zylons in deep space, avoiding asteroids and docking with friendly starbases. Graphics-rich game, with strategy and tactics.

"Not the Neubauer original, but a good strategic space game, nonetheless." —DFS

☆**Strip Poker** (Artworx) — Two lovely computer beauties are out to beat the pants off you in this strategic card game.

"Does the fact that it's the most popular party game I have say something about the company I keep?" —SP



ST Shuffleboard.

☆**ST Shuffleboard** (Shelbourne) — A colorful version of tabletop shuffleboard with some neat features, including player's point of view and an overhead look at the playfield.

"Sounds boring, but once you've tried it, you'll be hooked. Good graphics and playability." —MD

☆**Sundog** (FTL Games) — A star freighter is at your command in this graphics adventure. Visit many planets, wheel and deal.

"An adventure that has the best user interface of any game of its kind. Lots of action and adventure, great graphics, and a good story." —IC

"The first game on the ST that made my eyes bulge. A definite winner in the adventure department." —CW

☆**Time Bandits** (MichTron) — The old west, a fantasy world, deep space and thirteen other arcade worlds await you in this graphics action and adventure game. Not just your everyday shoot-'em-up.

Four-Star Software *continued*

*"The best arcade game yet. . . something for everybody—adventure, action, **Pac-Man**, puzzles, mazes, fabulous graphics." —IC*

"This is a sprawling, many-leveled masterpiece that combines elements of text adventure, arcade and strategy games." —CFJ

"[The] best shoot-'em-up on the ST" —AL

"A work of art. Games within games within games. . ." —CW

☆ **Wanderer** (Eidersoft) — Action in space, as you investigate the cause of strange disappearances on Earth. This game features true 3D graphics with supplied glasses.

"The first time a borderline-ridiculous game has ever been great. Get rich by saving cats." —DFS

☆ **World Karate Championship** (Epyx) — Furious black-belt action against a backdrop of landmarks around the world. Use up to fourteen different moves against your computer or real-life opponents.

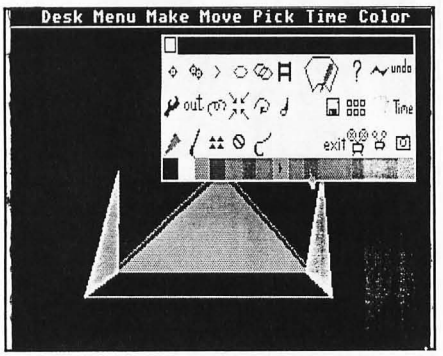
"Its playability is just a little slack in the joystick response area, but it makes up for this in superbly detailed graphics, animation, and addictive music and sound effects." —MR

GRAPHICS

☆ **Aegis Animator** (Aegis) — Turn your computer into an animation machine; design the shapes and animate them, changing the size, color, shape or position.

"Until you start trying to create your own animated scenes, you don't realize what this program can do. It's demanding but worth it; the results will surprise and delight you." —IC

*"**Aegis Animator** is excellent." —AL*



Aegis Animator.

☆ **Art Director** and **Film Director** (Broderbund) — A very powerful, easy-to-use paint and animation program duo.

*"This pair of programs is digital heaven for me (at least for this year). **Art Director** packs more powerful features than any other paint program I've used; **Film Director** gives me the tools to do quality animation with more ease than any of the other ST animation programs available." —MM*

☆ **CAD 3D 2.0** (Antic/The Catalog) — A 3D modeling package with shading, wireframe graphics and the ability to view objects from any angle.

"Sets a new level in technology. . ." —FC

"A professionally produced package, an unbelievable program, great documentation—what more can I ask for?" —IC

☆ **DEGAS Elite** (Electronic Arts/Batteries Included) — The follow up to the popular drawing/art program **DEGAS**, **Elite** adds many powerful features, including object stretching and rotation, as well as multiple screens in memory simultaneously.

"To my mind, still the best paint program on the market; well supported and with a wealth of printer drivers. If only it did full-page screens, it would be perfect." —IC

"Easy, fun and useful. It's always one of the first things I boot up when I'm trying to impress people with the ST." —MD

"I have yet to see a better graphics program for the ST." —CFJ

"I use it for drawing pictures, schematics, posters, and just about everything else that can be better represented visually than with words." —MR

"This one is sure to be on everyone's four-star lists." —CW

☆ **NEO-Chrome** (Atari) — Plot circles and other shapes with this program, as well as full-blown color images (all in low resolution), and any artwork or designs you can imagine.

". . .quick, efficient and features the best zoom window I've seen. . .as well as sporting the best circle/oval-drawing routine of all ST paint programs." —MM

"Yet another model of proper programming. THE utility for the computer artist." —DFS

☆ **Paintworks** (Activision) — This easy-to-use "paint" program permits drawing in all three ST screen modes, color rotation, animation, magnification modes, and much more.

". . .the closest thing to MacPaint on the ST." —FC

*"Although I'm no artist, the **Paintworks** integration of sight and sound (via music studio) make it my favorite drawing program." —SP*

☆ **Tinyview** (public domain) — Take **DEGAS** and **NEO-Chrome** files (or compatible) which have been saved with **Tiny**, and display them one after the other in this "slideshow." And images are compressed on disk to a fraction of their previous memory size.

*"Being a collector of ST artwork, I have to say that the **Tiny** format viewer and converter programs written by David Mumper are two of the best utility programs to come around in a long time. Pictures are stored in a compressed format, but you are not penalized for this with reduced loading speed." —CB*

LANGUAGES

☆ **Alice** (Looking Glass) — This "fill in the blanks" Pascal editor and interpreter won't let you make syntax errors; full GEM support.

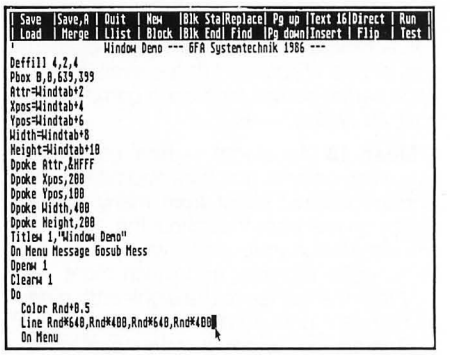
"A great learning tool for beginning structured language programmers—it won't allow you to make mistakes. One of few languages that I've even bothered to put onto my hard disk." —CB

☆ **AS68 Assembler** (Digital Research Inc., included with the Atari Developer's Kit) — The standard DRI 68000 assembler—the only one available for quite a long time.

*"While not a macro assembler, and not particularly fast, I've found **AS68** to be a very capable workhorse assembler. And all of the other ST assemblers I've looked at have had drawbacks of one sort or another, so **AS68** wins top honors for the ST assembler." —CFJ*

☆ **AssemPro** (Abacus) — A machine language development package offering a fast 68000 assembler, full screen editor, debugger and utilities.

"The editor and debugger have minor problems, but its user interface is well defined and its speed is quite impressive." —MR



GFA BASIC.

☆ **GFA BASIC** (MichTron) — Fast, compilable and structured, this language is becoming a real favorite among ST users everywhere.

*"A hands-down winner for me—I like BASIC, and **GFA**'s does just about everything I need, as well as compiling into stand-alone code. I've grown very fond of structured code since I began using this program." —IC*

". . .easiest to use, well thought out structured version of BASIC. This should have been the BASIC released with the ST." —FC

"Fast and powerful, this could be used for serious program development." —TH

*"Although the first release had some serious bugs, **GFA BASIC** is a definite step forward in user-friendly programming languages for the ST." —CFJ*

"At last, a working BASIC for this machine; it can remind you a little bit of all other ST languages." —DFS

"Clean, lean and muscular. This hybrid just may become the standard." —CW

☆ **Lattice C** (MetaComCo) — Full-blown C language with full screen editing, linker and 270-page manual.

*"Okay, so it isn't **Megamax**, but it's easy to use and it works." —DFS*

☆ **Mark Williams C** (Mark Williams) — A C compiler with access to GEM AES and VDI libraries, microshell command processor and full documentation.

*"Well done implementation of C, with many ST-specific areas supported. . .my only gripe with **MW C** is their assembler, which needs*

to be reworked or replaced with a more standard 68000 assembler." —TH

☆**Megamax C** (Megamax) — C development system with compiler, linker, resource construction program, editor, disassembler, and much more.

"A very fast and flexible C development system, with everything you need to get going in C very quickly. (Forget about the included text editor, however. . .)" —CFJ

*"Its power and speed astounded me. . . the **Megamax** manual provides more complete, concise and accurate documentation than any other reference manual I have seen for the ST." —MR*

"Arguably the fastest high-level language ever to use so little RAM. Also the best docs." —DFS

"About as user-friendly as a compiler can get; it taught me that banging your head against the wall is not part of K&R's definition of C." —CW

"Not ST BASIC (Atari) and not ST Logo (Atari)." —AL

TELECOMMUNICATIONS

☆**Flash** (Antic/The Catalog) — A large capture buffer, plus autodialer, macros and help screens, are some of many features in this popular program.

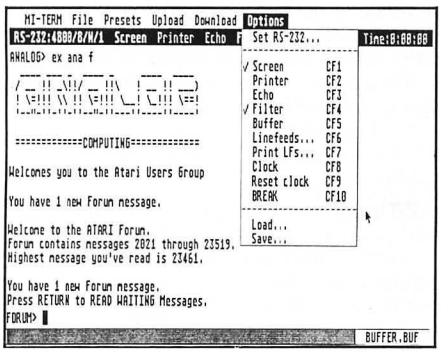
"The program uses the ST's abilities to the fullest, with GEM for convenience on the editor screen and TOS for speed on the main terminal screen." —TH

"One of the most-used programs in my software library." —CFJ

". . .a model of responsiveness and utilization." —DFS

☆**Kermit** (public domain) — File transfer protocol gaining wide acceptance among both mainframe and micro users. Allows use of wildcards in filenames, so more than one file can be transferred at a time.

"The GEM version has made running our SIG on Delphi a lot easier. I can send over a dozen or so files without worrying about specifying filenames." —CB



Mi-Term.

☆**Mi-Term** (MichTron) — This terminal program supports uploads and downloads, has a capture buffer, user-definable preset keys, and much more.

". . .More complete and slightly more costly. You decide." —AL

☆**ST-Talk** (QMI) — Full-function package with capture buffer, help screens, date and time reporting; now available as a new, updated version (**ST-Talk Professional**).

"Relies on simplicity to perform all the functions you need in a terminal program." —FC

"The best value in an ST terminal program. Also, I like the way the company does business—upgrades for two bucks!" —AL

"Inexpensive, very simple, enough power to keep the average or occasional modem user happy." —MM

"A simple-to-use program that packs a lot for so little." —DFS

WORD PROCESSORS

☆**1st Word** (Atari) — The premier ST word processor, packaged free with the first models.

"It does everything I need. The price wasn't too bad, either." —MD

"It's slow in places; it's buggy everyplace—but it's still more applicable than its competition." —DFS

☆**Regent Word II** (Regent) — Contains a 30,000-word spell-checker; display text will change instantly on-screen.

". . .of course!" —FC (President of Regent)

*"Shades of **AtariWriter**." —DFS*

☆**ST Writer** (public domain) — Word processor based on the popular 8-bit **AtariWriter**; lets users move up with little retraining.

". . .fastest word processor I've ever used." —CB

*". . .extremely fast and flexible, allowing me to write program documentation, articles and other creative writing with the minimum headaches. Instead of being what-you-see-is-what-you-get, **ST Writer** focuses its power on the task of writing. The way I see it, that's what a word processer is for. I let **Publishing Partner** do the fancy stuff at print time." —TH*

"It's fast! It's not GEM-based, and you don't see what your text will look like on the printer as you type, but it's more full-featured than most of the commercial word processors available." —CFJ

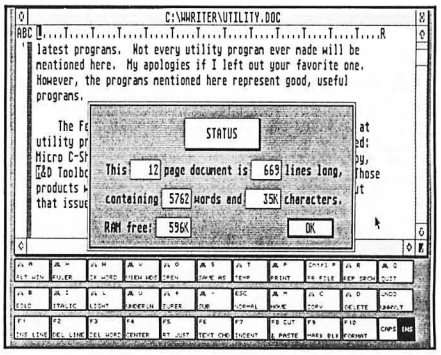
*"It was a simple transition from **AtariWriter** on the 8-bit. . .and that's good enough reason for me." —LHP*

*"I've found that **ST Writer** is easy to use if you're already happy with **AtariWriter** on the 8-bit." —MR*

☆**Word Writer ST** (Timeworks) — Some of this word processor's many features include: three spelling checkers with 90,000 words; an integrated outliner; and on-screen bold print, underlining, italics, and more.

*"The built-in spelling checker has saved my bacon on many an occasion! A vast improvement over **1st Word**." —CB*

"This is the best GEM-based word processor I've encountered to date, and it deserves credit for being a competently designed and produced package. If you must have a GEM word processor, this is the best buy." —IC



Word Writer ST.

*"Pretty much a clone of **1st Word**, but with most of the annoying quirks eliminated. A usable, although not ideal, GEM-based word processor." —CFJ*

*"I find it every bit as powerful as **1st Word**, with more power and features than I've had time to learn." —MR*

"The finest GEM-driven word processor available. It does just about everything." —CW

And a general quote on the subject:

*"No first choice. Until we get Macintosh-like fonts and other capabilities, I will be underwhelmed. If you have to choose, **Text Pro** (by Abacus) and **Regent Word II** (from Regent) are two to choose from, unless you get **1st Word** free with your ST." —AL*

IN ADDITION

☆**ALT** (MichTron) — A desk accessory that allows you to redefine macro keys; you can assign a string to any of the ALT-key combinations.

"Really eases repetitive typing, especially while programming." —CFJ

*"**ALT** is a handy utility that goes a long way to making programs such as Micro Emacs user-friendly." —MR*

☆**ARC.TTP** (public domain) — ST translation of a popular file compression program found on other computers.

"Being able to compress a file in order to save disk space is a SYSOP's dream come true." —CB

*"**ARC** stands for ARChive. . .[This program] allows you to compress and group many files together into one; since ST programs tend to be quite large (and often need other support files, as well) **ARC** is an indispensable ST utility." —CFJ*

☆**The "Bee" series** (First Byte) — Educational programs with voice synthesis.

"All these fine programs get my vote, especially because they speak to the child in a kind voice." —DFS

☆**Copy II ST** (Central Point) — Back up your ST disks.

"Use it, don't abuse it." —AL

☆**Cornerman** (MichTron) — A desk accessory with auto phone dialer, clock, games, calculator, notebook, and more.

Four-Star Software *continued*

"Best and most useful ST software add-on." —AL

☆ **CZ Android** (Hybrid Arts) — Editor and librarian program for the Casio CZ series. Performs graphic and numeric editing.

*"**CZ Android** takes a lot of the hassle out of creating new sounds. . . it does it automatically, letting you tweak parameters." —TH*

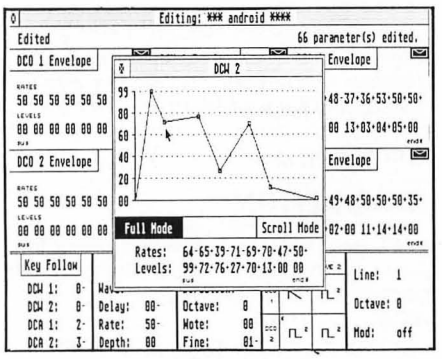
"This program will actually come up with new sounds for your Casio synthesizer all by itself. And a surprisingly large percentage of these "artificially" generated sounds turn out to be very usable. Also acts as an excellent GEM-based patch librarian." —CFJ

"This program is excellent for experienced users and those with a casual interest in computer music just getting into MIDI." —LHP

☆ **DeskCart** (QMI) — Cartridge which gives your computer a calendar, RAMdisk, calculator, note pad, and more—without taking up too much of your ST's RAM.

"Should've been built into the ST." —FC

"Gives all the little side programs needed, without consuming much RAM." —DFS



CZ Android.

☆ **Disk Doctor** (Antic/The Catalog) — Disk utility which allows for file examination and editing.

"A great utility for editing disks, recovering files, examining sectors." —IC

"A GEM-based disk sector editor for the ST; very easy to use, with comprehensive help files. The only thing it lacks is a disassembly feature." —CFJ

☆ **DiskMan** (MichTron) — A shareware program that offers most GEM desktop functions from the desk menu.

". . . it's better than some commercial accessories that try to do the same job." —MM

☆ **Fast RAM Disk** (Gold Disk) — Ian's description says it all.

"A fast, easy to install, configurable RAMdisk and a separate print spooler. A good solid product that performs well." —IC

☆ **K-Switch** (Kuma) — Allows you to have two separate applications in memory at the same time, and to switch quickly between them.

"Great way to run two applications at one time on the ST, by splitting [it] into two computers . . . if expanded to handle more than two partitions, this would be a spectacular utility on the Mega STs." —TH

☆ **LogiKhron Clock Module** (SoftLogik) — Give your ST the ability to tell the time with this battery-powered clock module.

"Gee, it's nice to have an ST that knows how to tell time." —CW

☆ **Magic Sac** (Pacific Data) — Turn your ST into an Apple Macintosh cartridge.

"I won't vouch for its ultimate usefulness, but it makes the ST monochrome run like a Mac. Just the look on a Mac user's face when they see it running is worth the price! It doesn't run everything from the Mac, but David Small is working on improving it and fixing the code, all the time." —IC

☆ **MichTron Utilities** (MichTron) — Modify sectors, repair damaged sectors and change data.

"More functions than you'd expect in this very, very useful program." —FC

☆ **MIDIPlay** (Electronic Music Publishing) — Very easy-to-use music program, which uses the monitor sound channel or a MIDI port. Create and play back your own music or tunes offered by the manufacturer.

". . . an inexpensive tape recorder that does what the boxtop promises: record and playback of your music, or playback of prerecorded music." —FC

"A big help when I started experimenting with MIDI. Very easy to use, and offers canned music which I can demo to others." —LHP

☆ **MIDITrack** (Hybrid Arts) — Full-range editing and auto patch librarian are some of this MIDI recorder's abilities.

"On the ST, [this program] takes on a whole new dimension. Thanks to the mouse, commands can be accomplished by pointing and clicking . . . impeccable timing, as well; something I can't say for the 8-bit version, I'm afraid. Highly recommended." —CFJ

☆ **Music Studio** (Activision) — Compose and play music, utilizing the built-in ST sound through the color monitor or a MIDI port. Up to fifteen channels may be created.

"Another on my list of best programs ever written. The creative music utility that doesn't get in your way." —DFS

☆ **Thunder!** (Electronic Arts/Batteries Included) — Spelling checker which works with any GEM product, such as word processors, databases, communications software, and more. An over-50,000 word database (user accessed), real-time and smart.

"A great spelling checker I use all the time. It needs some updating, however (are you listening Mark?), to improve some minor limitations." —IC

"Okay, so I'll admit that I'm not great at spelling. That's why this sure beats opening a dictionary every minute." —MD

"An asset to any writer; a must have." —AL

"The best spelling checker I've ever had the pleasure of playing with. A few features lacking, but they're so miniscule, the average user will never miss them." —MM

"This RAM-resident spelling checker is just like an annoying grade-school teacher, looking over your shoulder, catching your every mistake." —SP

☆ **Turtle** (public domain) — A multi-faceted hard disk backup program.

*"Simply the best program to back up your hard disk. It's hard to believe that **Turtle** is free, when retail programs don't support half of its functions." —FC*

HARDWARE

☆ **CZ-101 Digital Synthesizer** (Casio) — For right now, *the* low-end synthesizer for ST users. Lots and lots of features, all at a low cost.

"Okay, so it wasn't originally made for the ST. But it plugs in, receives data, and makes great output, does it not?" —DFS

☆ **Haba 1200 Modem** (Haba Systems) — Inexpensive and built to stay that way.

"Haba delivered what Atari and others promised: a $99, 1200-baud, Hayes compatible modem." —FC

☆ **Monitor Master Switchbox** (Practical Solutions) — Change from your monochrome monitor to the ST color monitor with a push of a button. And it has outputs for stereo connection.

*"If you have two monitors, you need **Monitor Master**" —AL*

"Maybe it's not an engineering miracle, but it sure is handy!" —CW

☆ **QMS Big Kiss Laser Printer** (QMS) — The most inexpensive laser printer available, offering very quiet, reliable, high-quality service.

"Fourteen builit-in fonts, Epson emulation, proportional and variable pitch printing, and other functions, make it well worth the price tag." —FC

☆ **SH204 Hard Disk Drive** (Atari) — Atari's 20-megabyte hard disk.

*"My Atari **SH204** (although larger than other 20-meg drives available) has performed flawlessly through a lot of use; I'm satisfied with it. Knock on wood." —CFJ*

☆ **Supra 20-Meg Hard Disk** (Supra) — A compact hard disk with 20 megabytes, offered in versions up to 60 megabytes.

"Only choice for an ST hard disk; the company stands behind its products, too." —AL

"Reliable and fast, these drives also manage to be compact." —CW

". . . fast, quiet and reliable, it has turned my ST into a serious development system." —MR

☆ **Z-Time** (Terrific Corp.) — A real-time clock add-on, which installs inside the ST.

"I don't think that I could go back to a time before file date/time stamping, since I installed this clock in my ST." —CB

☆ **5¼-inch Disk Drive** (First Byte) — Runs 5¼-inch disks which can be ported over to an ST for alteration, or any use you desire.

"Useful for transferring files from an MS-DOS machine to ST, and vice versa." —AL

Lasers do not buzz.

by D.F. Scott

In August of 1986, Atari's Leonard Tramiel told us he thought "desktop publishing" was merely a computer industry buzzword, and, essentially, that Atari is a company which addresses new technology, not new terminology. Since then, Apple Computer, after hiring its new advertising agency, launched an aggressive marketing blitz based on use of the Macintosh for nothing other than generating high-quality printed output. Although not named specifically, those ads implied the presence of an item called the Apple LaserWriter. It was the most revolutionary laser since the Strategic Defense Initiative.

This year, ironically, Leonard Tramiel is in charge of producing desktop publishing software for Atari. Leonard has appointed Dave Staugas, veteran programmer of the "old" Atari, to write the software necessary to drive its new laser printer. If that name seems familiar, it may be because you've seen it beneath the term *Programmer:* on the title screen of **NEO-Chrome**. Staugas has also been a member of the TOS development team and probably knows more ways to circumvent Digital Research's GEM than any other programmer alive. Knowing Line-A routines by heart, Staugas is also familiar with the blitter chip, which we'll discuss later, when we'll actually see it work.

Driving a laser.

Unlike current, ordinary appliances for sale, which use the term *laser* in their names—the Chrysler Laser, Lazer-tag, or the LaserBeam wristwatch—laser printers do indeed have lasers in them. In essence, they are photocopy machines without some of the "photo."

A standard photocopier beams high intensity light over an original page, whose reflection is shone onto a light-sensitive photo plate. At the darkest areas, where less light is reflected, the printing ink collects by means of chemical attraction. The inked plate is then pressed against a blank page, and the copy is made.

With a laser printer, since there's no "original," the image to copy is actually conceived in rasters, or consecutive memory, just as if the image were to be sent to a very large monochrome monitor with ultra high resolution. With standard laser printers, including Atari's (but not Apple's), a bit-mapped image of the item to be printed is sent from the host computer to the printer, where the internal laser acts like a slow version of the electron gun on the monitor.

The beam scans from left to right, one pixel at a time, from top to bottom, onto the photographic plate. The laser thus replaces the high intensity light source of the photocopier. After the image is fully scanned onto the plate, several "copies" may be run off in succession, without having to rescan the image each time.

If the amount of memory in a **DEGAS** monochrome screen image saved to disk is 32K, and with at least 32K of video RAM necessary to maintain the screen image, imagine how much memory it would take to hold an 8½x11-inch image with a resolution of 300 dots per inch—roughly thirty-three times as much, or just over a megabyte of memory for *one page!* Since our microcomputer can't operate with that much overhead (remember, it has to hold printed fonts in RAM, as well) programmers have no choice but to find ways to send images to the laser printer in smaller chunks.

Apple's solution—a popular one, at that—was to use an interpretive language called PostScript. Instead of communicating the letters from a font as a raster set, PostScript uses algorithms which geometrically interpret each letter's outline. The straight line in the letter *P*, for instance, would have a starting and end point, and a width; the arc would have a focal point and a certain number of degrees in curvature. This information consumes considerably less memory than the entire letter represented as a string of pixels.

What's required of the laser printer using PostScript is the capability to interpret that language and develop a raster set from the geometric information provided by the computer. To do this requires a computer in and unto itself; the Apple LaserWriter is equipped with one. It isn't too difficult, as SoftLogik's **Publishing Partner** has proven, to adapt the LaserWriter to the ST, using PostScript.

Yet if Atari had chosen to use PostScript in its laser printer, it would have had to develop a separate interpretive computer device for the printer—which would have driven up the cost of the hardware. Still,

had Atari developed a lower cost PostScript based printer, it probably would've placed itself in the position of being one of the leading Macintosh peripheral manufacturers—not exactly what Atari had in mind.

Enter NeoMan!

So Dave Staugas, the author of Atari's laser printer drivers, describes the situation he's faced with: "The laser printer we're going to produce has no smarts, has no RAM, has no processor, no operating system. It's going to be dirt cheap."

To Atari, the use of virtually two computers to publish one document is overkill. All the "smarts" can easily be software based within the controlling unit (preferably a Mega ST). Without a computer unto itself, the only machine that can currently drive the printer *is* an ST. Yet with prices as low as they are, one component may very well sell the other. The cost of an entire Mega ST4 system *with* laser printer will still be lower than the cost of an Apple LaserWriter *alone*.

Perhaps the easiest task for a laser printer driver is screen dumps. Staugas explains: "I use the theoretical minimum band size, which is one pixel. In a 4X screen dump, every pixel is actually four laser pixels on a side. It's a four-by-four enlargement. If you were to take the data on the screen and just splat it down on the laser printer, you'd get a little postage-stamp-sized picture. You might *want* to embed a postage-stamp-sized picture in your text."

A true photograph, for instance, could be scanned into **DEGAS** format with a converted dot-matrix printer, using SoftLogik's **IMG Scan** by Tad Painter. The resultant screen image could then be laser printed in a selection of sizes using the Staugas screen dump driver, with a final result nearing AP Wirephoto quality.

Next on Staugas's list is Diablo 630 emulation. A Diablo 630 is a serial daisywheel printer, manufactured by a division of Xerox, which uses the accepted serial printer command sequence. The laser printer under the 630 driver will accept those commands as if it were a Diablo, but will print using whatever raster-based font has been loaded into it. The results can look like a page from a book.

"I'm going to allow an extension," says Staugas, "where you can put in a little piece of a bit-mapped picture, say, or a filled rectangle—the only other thing it'll allow." Please note that Staugas, like most dedicated artist/programmers, speaks of his work—even his code—in the first person. "I have to build this on the fly. You turn the laser printer on, it starts to request data. I can't have the whole image sitting in memory, waiting to go. I have to be building it as it's requested. So I have a little band buffer that's already built for graphics, and I turn on the DMA. As the DMA is shipping the data, I'm building the next band, and I'm keeping up with it. As long as it's not too complicated, I'll do just fine."

One disadvantage of using the 630 emulator with a "low-RAM" computer like a 520 or a 1040, is that there's not much RAM left for the screen after it's completed a page layout—so WYSIWYG (or What You See Is What You Get) page processing is out of the question. You can, however (or as Staugas would say, "I can . . .") use two different font sizes (two separate fonts) with super- and subscripting, and automatic margin justification or line-centering to the microdot.

"It'll be just a matter of time," predicts Staugas, "As soon as I get the Diablo 630 emulator done, we'll easily convert to HP [LaserJet] emulation and Epson. I can see it doing Epson real easy. A full Epson with full graphics. The only limitation is that you won't get the full resolution on the laser printer; you'll get 150 dots per inch instead of 300. But you can use a 520."

It's not really the printer driver, admits Staugas, that takes up so much space—the 630 emulator driver should consume all of 2K—but the data it uses. Each different size of font, for instance, requires its own separate file. "We just have a bit-map of the font," says Staugas, "the exact placement of the pixels. You just copy it over to the laser printer—it's a matter of *blitting* it, basically. With PostScript, they don't record the letters in that form. It's just lines, circles. It's described algorithmically. You don't have to store the *picture* of the circle; you just store the *formula*. It's a description of a letter in algorithmic terms, so that it can be scaled and rotated. Ours is very difficult to rotate, and its scaling looks ugly. You just do some crude approximations.

"We're going to supply font sizes that will work with most applications. The letter-quality printing will be just great. My driver for page layout will not be so great. You'll have a word processor which doesn't show you what the page layout will look like, exactly."

So the laser printer may have an economical price tag, but it certainly won't be economical in its memory consumption. The same can be said of a Formula I race car. It can also be said that both printer and car make proper use of *speed*. A LaserWriter will take a few minutes to set up the page; the Atari printer will be a Ferrari, beginning the print process almost immediately.

The reason, Dave Staugas explains, is that his software takes full advantage of a programming feat you may not have thought the ST could achieve: multiprocessing. "When you're using the Apple LaserWriter, you've got three things that have to be done serially: you have to send the PostScript program over the serial link from the host to the LaserWriter; then the PostScript interpreter has to build the image up into memory—that's another process. Then you start up the laser engine and start sending data to the CRT scan. Those are three serial processes; they can't start until the previous one's fin ished."

Staugas describes his solution—and again, he's speaking *as* the printer driver: "All that I have to do is wait until I get a whole page's worth of data from the word processor. It's all internal; I'm sitting [in memory], and the word processor is sitting there at the same time, and if the word processor is slow, then I'm slow. It can take less than a *tenth of a second* if the word processor is decent.

If Atari's final version of GDOS (Graphic Device Operating System) is used "literally," or by the book, a full-page layout would be interpreted, from conception to printout, as if it were one huge raster-display image—those thirty-three **DEGAS** pictures laid side-by-side, as we mentioned earlier. If the laser is to work with any other computer besides a 4Mb Mega, Atari's programmers must conceive of some way to overcome the memory intensity of GDOS.

Staugas's solution for his emulator-drivers is to have the memory for the printed font set up in such a way that each character is pointed to *at the time* the single "scanline" of data is being conceived. This way, instead of having over a megabyte of memory set up for one page and extra memory for the font, the page contents become composites of data blocks from font memory—thus, the same memory serves two purposes.

"As soon as I've got the full page's worth of text," Staugas explains, "I convert it to an intermediate form so that I can have all the pointers set up for each character on the page. It's ready to go, once I start the beam on the laser printer. I'm build-

ing these graphics as the beam is scanning, so I don't have to build the whole page, then start up the printer."

Assuming all goes according to schedule, Staugas' printer drivers should be completed by press time. The printer hardware, however, awaits the Mega, whose completion depends upon the fabulous bouncing blitter we spent most of last month's edition discussing.

The blitter bounces back.

Last month, we listed what was then considered to be all the improvements made to TOS to account for the inclusion of the blitter. Leave it to Atari to come up with more. Here, we present some more TOS additions and changes, from official Atari documents:

The internal real-time clock on the Megas may now be set or read through XBIOS calls 22 and 23, settime and gettime.

There is a new XBIOS call. As yet unnamed, it is 64, and its purpose is to read or set the blitter configuration. The parameter used is a 16-bit word referred to as flag. If flag is set to -1 beforehand, the call will only read the configuration, not set it. For any value of flag above -1, the least significant bit (bit 0) turns on or off (1 or 0) the hardware blitter. The configuration, whatever that may be, is returned in machine register D0.

The mouse handler can now be set to draw the mouse in XOR mode via Line-A. In this mode, if a "set" pixel is overwritten by the mouse pointer's "set" pixel, that pixel becomes "reset." Thus, an all-white pointer doesn't disappear against a white background, but becomes dark.

The AES now has two more working functions, appl_trecord and appl_tplay. It may be assumed their purposes are to capture and reutilize AES event pipelines.

Previous hard disk drivers had to enlarge the BDOS folder pool on their own; although they no longer have to, Atari docs now state that, as a result, long programs may get *longer*. No reason is given for this mysterious phenomenon. The 40-folder hard disk drive limit *remains*, despite reports to the contrary.

VDI arrays are also being enlarged, although, as of now, we know not how or why.

The alert box, called through AES 52, form_alert, now has a fixed maximum number of characters per line, *30*.

The timing in AES calls 21 and 25, evnt_button and evnt_multi, has been fine-tuned so all single, fast button clicks will be returned *as* single button clicks at all times, and not "buffered out."

The "Open Application" dialog box that appears after choosing a .TTP (TOS Takes Parameters) program will no longer translate lowercase characters to uppercase.

Machine registers D0, D1, D2, A0, A1 and A2 have often been automatically "destroyed," or written over, after placing machine-language trap calls; now they'll be destroyed *more* often.

The OS header—the first memory address in TOS—*may* move from its current location at $FC0000.

Last month, we mentioned the addition of a fast-format routine for use in XBIOS call 10, flopfmt. Though that routine isn't being used in current TOS prototypes, the routine is said to use -1 as its skew value; and the longword unused in the parameter table in that call is now a pointer to a one-word-per-sector skew table.

There's one more major change to be noted—a rather unfortunate one, but, since we're dealing with hardware, it could not be avoided. When the Line-A blit was software based, it was easy (even "legal") to start one blit procedure while another was in progress. This was called a "re-entrant blit." Most hardware, by nature, is not re-entrant; it would tie knots in itself if it were. Thus, those programs that invoke re-entrant Line-A blits, or those interrupt routines that invoke a blit while another's in progress in the main routine, go nuts and usually bomb.

Dave Staugas, a veteran of the blit, explains: "With the software blit, basically, you could stack up as many blit calls as you want—you could interrupt a blit call and finish up a blit, and have no problem with it. With the blit chip, though, we can't save the status of the hardware and execute another blit. Once you initiate a blit, you have to let it complete. If you're in an interrupt routine, you can't interrupt the proceedings with a blit in progress—there's a horrible interaction that causes it to bomb. It's only a matter of time before **Leader Board** just bombs."

There are reports that, with the blitter engaged, the **Leader Board** golfer moves madly across the screen, as if the putting green is, in fact, hot coals.

***Status report* interviews a blitter and its new TOS.**

Our other primary source this month is a 1040ST, with blitter/TOS enhancement, courtesy Atari's Darryl May, representing the corporation at its Dallas AtariFest. With us was Mark Bruttell, MichTron's Customer Service Director.

Our tests with the machine suggest the enhancement will have a greater effect on *future* software than on the current line. Many programmers—like Jez San for **Starglider** and Dave Staugas for **NEOChrome**—were able to write their own software-based blits which were considerably faster than the TOS Line-A software blit. As a result, many programs will show little or no speed improvement.

Apparently, TOS has also been changed in some undocumented way, so that the CONTROL-ALT-SHIFT key sequence used as a break in MichTron's GFA BASIC no longer registers; software-based stops and ends simply don't.

With graphically oriented programs, the degree of change using the blitter can be anything from imperceptible to uncontrollable. If there was any speed increase in **Aegis Animator**, neither I nor Darryl May noticed it. Tom Hudson's **CAD-3D 2.0** shows signs of a minor "turbo" boost. Of the games tested, **Goldrunner** displayed the most obvious speed increase. What once was a minor tap of forward thrusters now seems as if a brick was hurled onto the full-throttle switch. At least in one corner of the universe, warp speed may be enjoyed in safety.

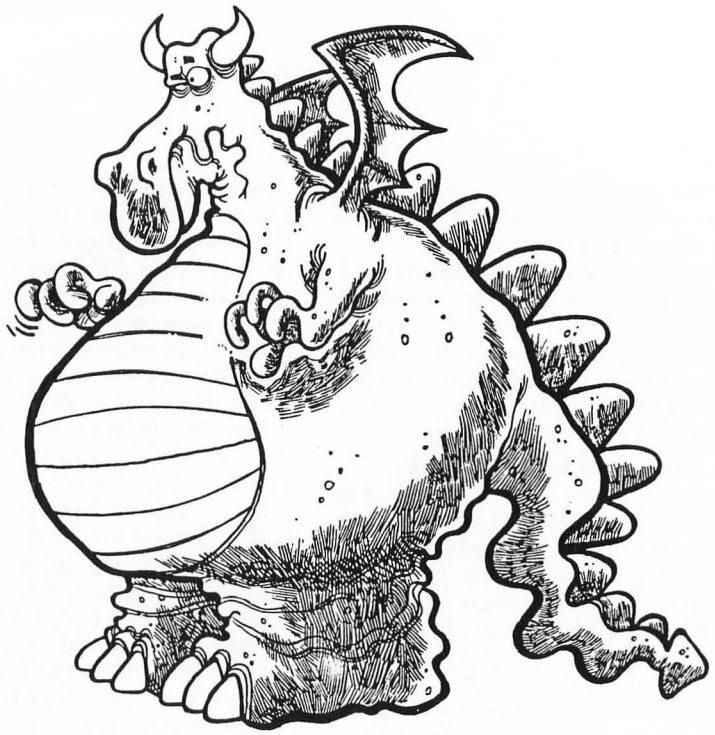
Dave Staugas tells us Atari's reluctance to ship the blit may have been due to an embarassing underestimate of the final price. "The design is good," he says, "the mask is good. The vendors we've been using have problems with the yield. It's not economical to produce the chip if the yield isn't high enough, so we couldn't offer it to you for $125. If we have a 5 percent yield, we couldn't sell it to you for $125—which is what we think it should go for, what the market will go for."

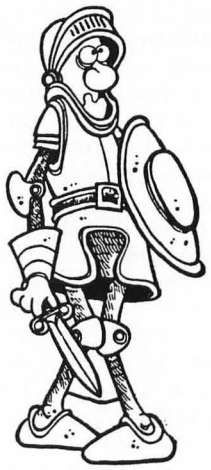
He confirms Atari has chosen the "piggybacking" method of installation, soldering the 64-pin chip directly onto the CPU.

The power of the presses.

We often speak of the desktop publishing revolution as if it related to the evoution of the microprocessor. To ignore its potential effect on the principle of free speech would be like describing the first printing press as a mere mechanism.

With the printing press, information became a commodity like farmland, and the voice of the ruler became only one of many. People learned to *disagree*, to challenge with the word as they had with the sword. Now, microelectronics has given the people the *press*. Now, whatever is said by the common person can be printed as beautifully as it was conceived. Perhaps this important vessel of reason will lend importance to the word it carries. ///





ST news, information and opinion.

by Ian Chadwick

Lee Pappas called me not long ago and asked me to do a column on what makes good software. Sure, I thought, fish in a barrel. After all, that's what I'm always on about, right? Should be easy.

Well, it's not. The more I thought about it, the more I realized that—like a lot of things in our lives—*good* and *bad* are subjective terms when applied to software. Basically, I came up with: "Software is good when it does well that which you expected of it."

Sound a trifle wishy-washy? Perhaps, but I couldn't come up with any overall laws governing something that's as wide-ranging as software. It's a lot like asking "what makes a good book?" or "what makes a good meal?" I tried two approaches: first, I assembled my own list of good and bad software, and tried to determine what made me put each piece in its category. Then I called a few friends and asked them to name pieces of software they thought good or bad—not just for the ST, although I paid somewhat more attention to it than to other machines. Bear with me, kind reader, because the issue is not confined to one machine. I'll have to use examples from several systems to illustrate my points. However, at the end of this diatribe, I *do* try to make a point.

The answers I got didn't surprise me; after all, I've heard a lot in the same vein before—and I've been arguing the point for years. But they didn't clarify the question of what determines good or bad.

A writer friend told me he thought **WordStar** was the best piece of software he'd ever used. To me, **WordStar** is one of the worst—it's difficult to learn and use, has nothing even vaguely intuitive about it, has few mnemonic commands, has bad documentation and produces files that can't be read by most other word processors without difficulty. He thinks it's great and won't even try another word processor—it took him so long to learn **WordStar** that he doesn't want to venture into that process again. And now, having mastered every nuance and quirk, he's quite proficient and efficient with it. He swears he *likes* the way the program works.

Another friend, who recently bought an ST after years working as a systems analyst in the mini- and mainframe computer environment and with an old CP/M system as his main home computer until this spring, sang the praises of **1st Word** endlessly—another program I detest, for different reasons. But to him, the simplicity of GEM and the ability to perform reasonably complex actions with a minimum number of keystrokes was simply amazing. After all, he was accustomed to command-driven, text-oriented programming. He waxed rhapsodic over the use of the mouse alone.

Both these people are professionals and both are well acquainted with the computer field. Both use computers in their daily lives to perform regular job-related tasks. Yet both approach the tasks differently and want their software to perform much the same task in almost radically different manners. But Bill, the writer,

Ian's Quest

wants a program that does *everything* a writing tool might do—while Bob, the programmer, uses his to write letters and shopping lists, and lets the kids do their homework on it. So it comes down to subjectivity, based on need for performance again.

Ah, but are there not some essential elements, some basic blocks we can use to measure quality? I wondered. Can we develop a scale, say one to ten, to measure certain features and determine, *in vitro*, as it were, the quality of a product? Break a program down to its molecular structure and analyze each component against an absolute scale? Not so easy, I learned.

Take, for example, the user interface. I've said before that I like to do a lot of things through the keyboard rather than the mouse, and having to remove my hands from the keyboard to activate a command, rather than being able to do it by key combinations, breaks the momentum of my activity. Something like saving a file is a task I pause to perform—the interruption to write to disk forces it, so I don't mind having to add into that pause the time to use the GEM file selector to set the drive and filename. But something like moving the cursor to the top of a file and searching for a word is a task I want to control through the keyboard. If I have to stop and play around with a scroll bar, then pull down a menu to get at the search command, I find I'm wasting my time and losing the thread of my work.

To me, therefore, a good ST program offers an integrated dual interface—a keyboard command structure *and* a GEM mouse/menu structure for those who prefer it. In some programs, GEM is such an integral part of things that I can't imagine the program doing the same job even remotely well without it. Look at Tom Hudson's fabulous **CAD 3D** 2.0. On the other hand, in **DEGAS Elite** he combines GEM and TOS (non-GEM) functions so well that you can't really tell they're not one and the same interface. And he supports the whole structure with a series of keyboard commands for "power" users.

In the last version of **PaperClip Elite** ever seen by BI, you couldn't use the arrow keys to move from the start of one line to the end of the previous one. Not a programming flaw, this was something the programmers wanted and felt was the way a word processor should work. In order to go backward this one space, you had to go to the beginning of the line above, then move the cursor over to the end. That sort of design is what I call seriously flawed. It's about as friendly to the user as a hungry crocodile—and about as pretty. This is, of course, a problem that often occurs when one lets programmers, rather than end users, design the specifications.

Another thing to consider is how well a program handles the activities in the interface. How many games have you encountered that don't have a "quit" button or command? Seems incredible, but many don't—you have to shut the machine off in order to exit. It's like having to turn off your car every time you come to a stop light. This is not only silly, it shows a critical lack of awareness of the user's needs.

In my column in *Strategy and Tactics* magazine, I've regularly raked companies over the coals for selling games in which cursor positioning is an integral part of play—and yet not providing a joystick routine to accomplish it. Instead, you must play with hands continually poised over the keyboard, to perform a task simply and easily handled by a natural joystick movement.

When computers come with built-in joystick ports and handling routines, this omission says a lot about the lack of development and testing, not to mention the abilities of the programmers. SSI, of wargame fame, has produced a lot of good games crippled by this glaring oversight, and they continue to do so—**Roadwar 2000** on the ST requires keypad cursor movement when a simple joystick control would make the game 100 percent friendlier.

Imagine buying a word processor to find that it wouldn't save a file. It might format, search and print, but not save. Sounds crazy, right? You'd never buy it; it would offer little that a typewriter couldn't provide. But what about a game, say one that could take four, five, maybe as long as eight hours to complete? What would you think if you couldn't save a game in progress, but had to start from the beginning each time, and play either to the end or until you ran out of chances and "died"? Crazy? Broderbund's popular **Lode Runner** was thus flawed. It was otherwise a great game, but one that I (like many others I know) never played after the first two or three times—because we

couldn't save it midway through the game. To my mind, that makes for bad software, no matter what other strengths it has. MichTron had the sense to revise **Time Bandits** to permit saving, much to their credit.

The original version of a popular IBM program, Ability, offered a telecommunications program, but it wouldn't let you download files—just capture text. Think of how useful that would be on Delphi or CompuServe!

Let's look at some other items. Ever use a spreadsheet? I've used several, on a lot of different machines. When I got my ST, I transferred all my income/expense templates over to **A-CALC** from Macintosh Multiplan format. But **A-CALC**, while being reasonably good, is plagued by a simple, annoying and avoidable data entry flaw. In it, you need to double-click on a cell to enter data in it—unnecessary, but you can learn to live with it.

The real irritation is that each cell accepts only one data type (value, text or formula), and, in order to specify which, you have to press TAB to cycle through the choices. It's an extra step that shouldn't be required of the user. After all, 1-2-3, Multiplan, Supercalc, VisiCalc and Perfect Calc never required it—they all have more intelligent methods to determine the data type. In Multiplan, anything that starts with an equals sign is a formula; anything that starts with a number or an arithmetic sign is a value; the rest is text. If you need, for some reason, to force an exception to this rule, it's easy to do so with a special character prefixed to the data. (Not having seen it, I can't answer whether or not the new **A-CALC Prime** continues this bit of folly.)

Another area I constantly harp on in word processors is the ability to search and replace carriage returns and other characters outside the normal number/letter range. I use this simple feature constantly, to remove excess CRs from text files I download or port from another computer. Anyone who has ever done this knows the frustration of trying to reformat text without this function. It's the main reason I continue to use **ST Writer**. But, while easy to implement, it seems to have been ignored by far too many people designing word processors. Personally, since it's so essential to my daily writing, I'll never use a word processor that doesn't have this capability, nor can I consider one without it a "professional" program.

Documentation is, as some of you may know, rather close to my heart, since I write and edit it for a living. I've seen a *lot* of good programs hampered by poor documentation. The PC version of **Word Perfect** is a shining example of a great program with an abominable manual. It's arcane, explains things poorly, is awkwardly organized, and, in order to learn how to use a feature, you have to read about it in the tutorial, then the reference and

ATARI USERS' GROUPS

Pasadena Area Local Atari Computer Enthusiasts (PALACE)
220 N. Encinitas, Monrovia, CA 91016
Meetings; BBS; newsletter.

Pikes Peek Poke Atari Computer Enthusiasts (^3ACE)
P.O. Box 9282, Colorado Springs, CO 80935
Meetings; newsletter. President: Eric Kowalski.

Gainesville Atari Group (GAG)
6733-B SW 5th Place, Gainesville, FL 32607-1706
Meetings; newsletter. President: Darin L. Delegal.

Mid-Florida Atari Computer Club (MFACC)
887 Benchwood Drive, Winter Springs, FL 32708-5114
Meetings; newsletter. President: Craig L. Kaplan.

Wells Atari Computer Owners (WACO)
505 Meadow Lane, Bluffton, IN 46714
Meetings; BBS; newsletter. President: James Davis.

Capital Hill Atari Owner's Society (CHAOS)
P.O. Box 16132, Lansing, MI 48901
Meetings; BBS; newsletter. President: Leo Sell.

Long Island Atari Computer Enthusiasts (LIACE)
P.O. Box 382, Bellport, NY 11713
Meetings; newsletter. President: Mike Ferrara.

Atari Computer Enthusiasts (ACE)
3662 Vine Maple Drive, Eugene, OR 97404
Meetings; PD library; BBS; newsletter.
President: Kirt Stockwell.

White Rose Atari Computer Enthusiasts (WRACE)
2413 Carllyn Drive, York, PA 17403
Meetings; BBS.

West Texas Atari Users' Group (WTAUG)
8215 Goodnight Trail, Amarillo, TX 79110
Meetings. President: Pete Coombes.

San Angelo Atari Computer Enthusiasts (SAACE)
P.O. Box 60993, San Angelo, TX 76906
Meetings; newsletter: *Hardcopy!!*
President: Joanne C. Miller.

Milwaukee Area Atari Users Group (MIL-ATARI)
P.O. Box 19858, West Allis, WI 53219-0858
Meetings; BBS.

Nanaimo Atari User Group
3326 Rock City Road #113
Nanaimo, British Columbia, V9T 3H6, Canada
President: Andy Barclay.

Merserside Atari Club (MAC)
24, Oakdene Rd., Liverpool
Merseyside, England L4 2SR, U.K.
ST Group; contact Mike Lynch.

ATTENTION USERS' GROUPS

If you would like your organization to be listed here, send information (and newsletter, if appropriate) to **ST-Log** Group Listing, P.O. Box 23, Worcester, MA 01603. **ST-Log** is not responsible for errors in copy.

(usually) the special features section—three places in which, too often, only a part of the explanation is found.

My wife told me about the problems she and her fellow workers had when they first got their PC, and tried to learn to use it and the software they were given with it. One program was Lotus's famous 1-2-3. The tutorial tells you to type the data exactly as shown, in order to follow their lessons. So they were reading through and working on it, and came to a line like *ENTER 100.* This meant, of course, to type *100* in the selected cell and press ENTER. But nothing told them that—and, being accustomed to typewriters with a RETURN key, they assumed the line meant press ENTER then type *100.*

Of course, nothing worked right—and it was a long time before they figured out why. That's an example where the docs should have been made more explicit, assuming unfamiliarity with jargon.

The initial release of MichTron's GFA BASIC suffered from what might easily be described as the worst manual for an ST product ever. If it hadn't been for the quality of the program and the stamina of its users, the package would have died right away. Again, MichTron had the sense to rewrite, update and revise the manual (although they foolishly bound the pages, rather than using the more practical binder style).

Of course, the flip side to the documentation issue is that the worse it is, the greater the demand for books on the program. Thus, there's a greater opportunity for writers like myself to enter the field. Of course, that too often ends up producing an overwhelming glut of books on a particular topic—and the books also serve as manuals for any pirated copies of the program.

The efficient use of computer resources is a key issue in determining how good a program is. I've already mentioned the idiocy of not providing a joystick routine in a computer designed for it—a great example of poor resource use. Another is memory management.

The earliest versions of MacWrite could only manage a document up to about ten or twelve pages long. Professional? Not even close. (The latest versions of MacWrite, without the size restrictions, are still clumsy, weak and ungainly word processors, ill suited for anything professional or important.) Perfect Writer, on my 64K Kaypro 4, can manage a text file as large as a disk (400K), by loading only segments into memory at a time. But the first IBM translation of the program (the PC was then also a 64K machine, remember. . .), couldn't even do that. After 40K, it choked and crashed.

DataSoft has produced some pretty good programs, with simple idiocies in them that, if avoided, would make them a lot better. Take **Mercenary**, for example. Every time you play—even returning to a saved game—you have to go through the agonizingly long intro. The first time it's cute. After that, it's merely annoying. I find it irritates me so much that I won't play the game any more. At least in **Alternate Reality**, they let you get out of it by pressing the spacebar. But **A.R.** has one of those stupid endgame situations where you have to reboot the computer if you die—something which again reduces its repeat play value for me.

What else makes a program good or bad? Well, the visual interface for one. Good graphics get me to appreciate a game more than mediocre efforts. **King's Quest II** uses graphics that seem identical to the 8-bit versions—something that annoys me no end. After all, I paid good money for a machine with higher graphics capability, so why should I pay for 8-bit graphics in my software? Not that they have to be art masterpieces, but elegance can be had simply.

Sundog from FTL has better graphics than its 8-bit versions, and, when seen side by side, the difference makes the ST version stand out. Look at the terrific graphics in **Time Bandits**. It's a superb example of what can be done. (MichTron's latest, **GoldRunner**, is a good arcade game with nicely crafted graphics, marred by a protection scheme that makes it impossible to save high scores. The program hangs and requires a system reboot —at least on my copy, and on those of two other people I know.)

Good graphics don't necessarily make a good game, however; Mindscape's **SDI** has a lot of good graphics, but the flight controls are sluggish, the action slow and the plotline, well, juvenile. Electronic Arts translated **FireFox** for the ST from other computers. The result is bad graphics *and* bad action, making it one of the worst games of the year.

Activision's **Little Computer People** has graphics that look a lot like Commodore 64 screens, which they probably are. The program is stunning in another manner: its total lack of purpose. It struck me as the software incarnation of the word *pointless*. Thinking it might be just my adult point of view, I gave it to a friend with two girls, 8 and 11. They lost interest in it within a few minutes, declared it stupid and went off to play elsewhere. I was gratified to have my impression seconded by experts.

Why are there so many little nuisances in otherwise good software? In part, because programmers do a lot of the design. Many are simply not qualified for this; they may program like geniuses, but when it comes to creating specifications, they're fish out of water.

Another nuisance is that too many programmers and small software companies work in a void. They never check out the competition, especially that on other systems. There's nothing wrong with taking a good idea or a good implementation and writing your code around it—why reinvent the wheel?

Good design *demands* that a program be developed by a broad spectrum of people—users, programmers, marketing experts. Who can best design a tractor: a race car driver or a farmer? Who can best design a word processor: a writer or someone whose writing is mainly done with a text editor for source code? With all due respect for their talents, programmers seldom have the breadth of knowledge, the user experience or the practical experience in the business world to design a variety of products.

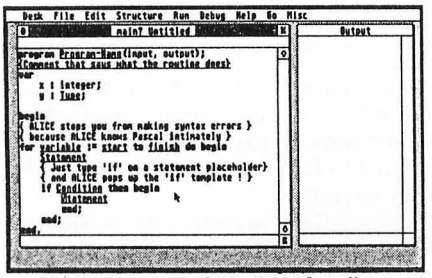
So what makes software good? Design, I suppose, is the key issue. After all, it, more than any other single thing, determines the quality of software. Without proper, well developed design work, you end up with software in the discount sale bins a few months after its release—or, less fortunately, on buyers' shelves gathering dust after one or two attempts to use it. And I, for one, think we're long past having to put up with that sort of thing. //

We were going to tell you all about

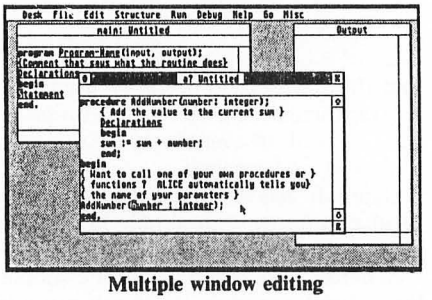
ALICE

The Personal Pascal™

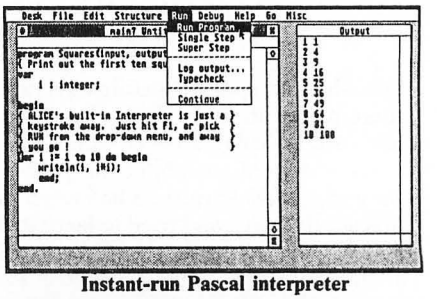
but we'd rather show you . . .



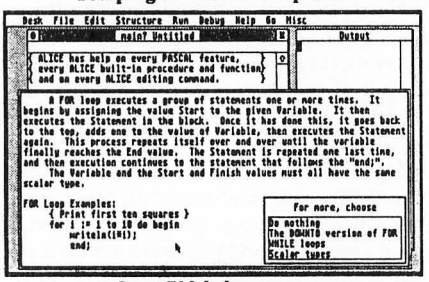
You program from "templates"



Multiple window editing



Instant-run Pascal interpreter



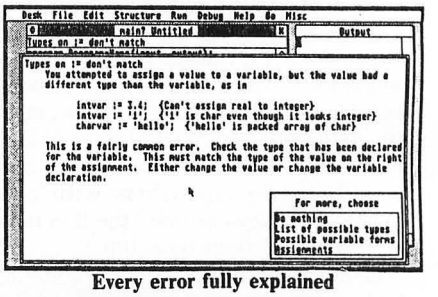
Over 700 help screens

Programming Made Easy

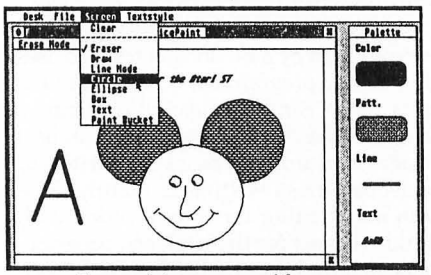
If you're into programming, or if you *want* to be, you won't find a better system for your Atari ST than ALICE: The Personal Pascal. Already popular on the IBM-PC, ALICE now brings easy, interactive programming to the Atari ST.

ALICE is, without question, the best way around to learn about computers and programming. Beginners can even order our ALICE based textbook for only $19.95 with ALICE.

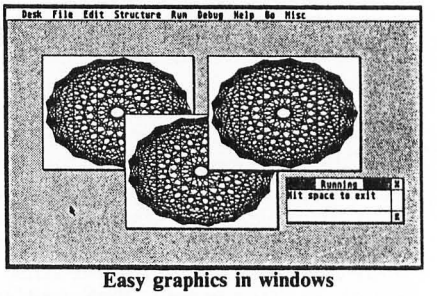
Even if you're an expert, ALICE makes it easier to write, test and especially *debug* programs. ALICE's extensive Pascal improvements include a GEM interface even beginners can use and most of the extensions of Turbo Pascal. The surrounding examples just give a glimpse at how easy it is to put programs together using ALICE.



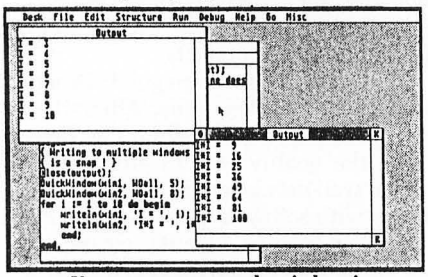
Every error fully explained



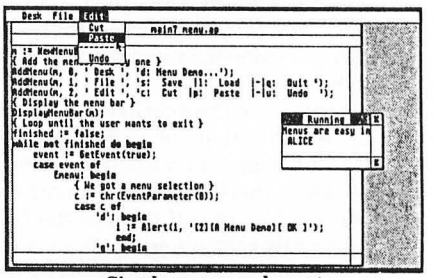
Free paint program with source



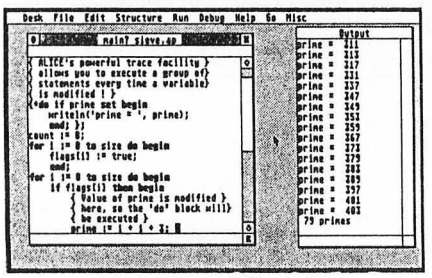
Easy graphics in windows



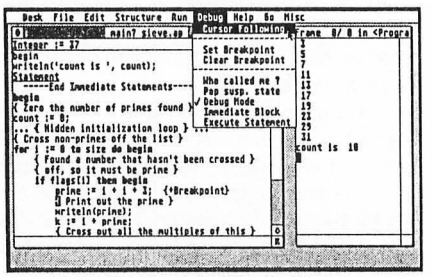
Your programs can do windows!



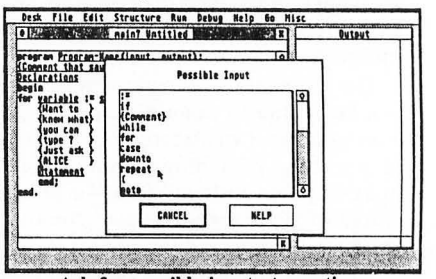
Simple menus and events



Powerful variable trace



Breakpoints, Single Step



Ask for possible input at any time

"If I needed to learn Pascal all over again, or were going to teach a course in the language, I can't imagine using any program other than ALICE."

— Adam Green, **Infoworld**

"ALICE may be the most advanced programming environment currently available for the PC."

— Michael Covington, **PC World**

"If you enjoy programming languages, this comes pretty darn close to being as much fun as a video game."

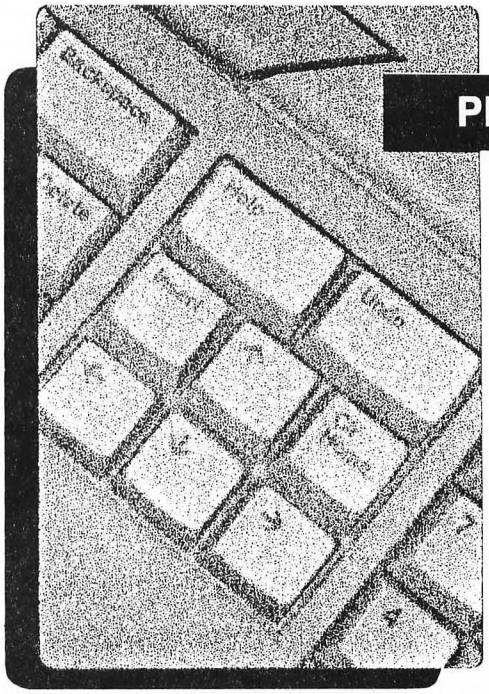
— Gene Wilburn, **Computing Canada**

Visit your dealer or send us $79.95 (USD) plus $5 for shipping and handling. Major credit cards. Call 1-800-265-2782. Dealer inquiries invited. IBM-PC version $95.

Looking Glass Software

Looking Glass Software Limited
124 King St. N. Waterloo, Ontario
N2J 2X8
519/884-7473

Assembly line



Further instructions.

by Douglas Weir

There are supposedly three types of statements necessary in any useful programming language. First, you must be able to execute sequential statements. Second, you must be able to branch from one part of a program to another, depending on a specified condition. Third, and finally, you must be able to execute a group of statements again and again any arbitrary number of times. BASIC, for example, implements the first type by consecutively numbered program lines, the second by GOTOs and GOSUBs, and the third—looping—in various ways: an IF statement followed by statements to be executed and a GOTO back to the IF, or, in more recent implementations, WHILE or FOR loops.

So far, we've learned how to use sequentially executed instructions in 68000 assembly language, and we've had a hint of how to branch or jump to other sections of executable code. This time we'll learn a common way of doing both kinds of jumps—"returnable" and "nonreturnable"—and we'll also learn how to write a simple loop.

Loops and, usually, branches both require a capability to test a condition and make a decision on the basis of that test. There are times when an unconditional branch is useful, as any BASIC programmer knows. Loops, however, must always come to an end, and the only way they can do so is by continually testing a condition which is at first true, while the loop continues, but later becomes false, when the loop ends. So we must first learn how to test conditions with the 68000.

Condition codes.

To do so, we must introduce a new area of the chip: the "Condition Codes Register." Although this register is a full byte in size, only the lower—rightmost—5 bits are used. Unlike all the other registers we've looked at so far, the CCR (as it's abbreviated) is not used to store numeric values, neither data nor addresses. Instead, each single valid bit is used as a separate "yes/no" flag to record certain specific results of various 68000 instructions. These results are automatically monitored by the 68000 whenever any instruction causes them to change, and they are then (also automatically) recorded into the CCR bits.

Let's take a simple example to see how this works in practice. Bit 2 of the CCR, the Zero flag, is used by the 68000 to record whether or not the "result" of the last instruction was "zero." I've put these two words in quotes because they have a somewhat broader meaning in this context than you might expect.

If, for example, you were to execute an instruction which subtracted the number *2* from register d0, which also contained *2*, then as a result register d0 would contain the value *0*. In this case the Zero flag in the CCR would be "set"—i.e., would have the value of *1*, meaning "yes, the last instruction had a result of zero." On the other hand, if d0 had contained *3*, then the result of subtracting *2* would have been to leave d0 containing the value *1*. The Zero flag would in this case be automatically "reset"—given the value of *0*, meaning "no, the last instruction did not have a result of zero."

This is just what one would expect. However, the Zero flag is affected by many other instructions. Among them, perhaps surprisingly, is move. If the 68000 executes an instruction that simply moves a zero into a data register, the Zero flag is set to *1*; if a nonzero value is moved into a data register, the flag is set to *0*. This allows a programmer to both move a value and set up a condition with one instruction.

How do you know which instructions affect which flags in the CCR? By looking up the definitions of the instructions you want to use. Part of the definition of each instruction consists of a statement of which condition codes are

affected by the instruction, and when. It's very important in assembly language programming to be sure about what's happening to the condition codes. For example, moveing data into data registers affects the Zero flag, but the flag is not affected by moves into address registers. Knowing these rules well is just as important for a programmer as knowing the tax code is for a corporate lawyer, and it can be just as advantageous.

Branching . . .

Fine—that's how a condition is recorded by the 68000, but how does the programmer go about using this feature? There are in fact several 68000 instructions that allow one to tinker directly with the CCR's contents, but they aren't often needed. Conditions are used, most of the time, to determine flow-of-control decisions. So the 68000 provides the programmer with a set of branch instructions, each of which tests one or a combination of CCR flags and then branches—or doesn't branch—to a location, depending on the result of the test. Whenever a branch is not taken, the 68000 continues execution with the next sequential instruction, just as if nothing happened.

For example, consider the following series of instructions:

```
        move.w    #4,d0     counter=4
loop:   nop                 do nothing
        subq.l    #1,d0     decrement counter
        bne       loop      if not zero, loop
start:  move.l    #8,d0     new important activity. . .
```

We start by moveing 4 into the register d0. The next instruction, which is labelled loop, does nothing. That's right—nop means "no operation." Almost every assembly language has a nop. For a long time programmers were unsure just what to do with such an instruction, until someone discovered that it made a great placeholder in memory. And that's what it's been used for ever since. Whenever you want to reserve—for any reason—some space in executable code, you can simply fill it up with nops. I've put one here just so I could say something is happening in this line of code.

The next instruction subtracts 1 from the contents of d0. The subq instruction, as you might expect, means "subtract quick," in the same way that addq means "add quick," as we learned last time. The next instruction does two things. First, it tests the Zero flag in the CCR. Then, if the flag indicates that the last instruction executed did not have a result of zero, meaning that our loop counter, d0, has not been exhausted, it loads the program counter with the address represented by the label loop. Otherwise, the program counter is loaded, as it normally would be, with the next sequential instruction, which also happens to be labelled in our example.

In other words, the bne instruction branches back to the label loop if the loop counter has not been decremented to zero. As soon as the count is zero, execution "drops through" to the next instruction following the bne, as if there were no branch at all.

The result of all this is that the "body" of this loop—namely, nop—is executed just as many times as the number originally put into the loop counter. There could have been many more instructions between the nop and the subq-bne pair which terminates the loop, and they would all have been executed as many times. This then is a nearly exact equivalent to a WHILE or FOR loop construct of a typical high-level language.

What does the bne stand for? "Branch not equal"—not equal to zero, that is. Branch to the label indicated if the result was not zero; otherwise, don't branch.

As I mentioned, there is a whole series of branch instructions, designed to test the five separate conditions and various combinations of them, and branch or not branch to an indicated destination depending on the result. We'll be learning the rest of the condition codes and the branch instructions as time goes on. They all work in exactly the same way; only the conditions are different.

. . .and branching to a subroutine.

You've probably noticed that I've said nothing about return addresses yet. That's because the branch instructions we've been discussing do not save a return address—they either branch or don't branch, and that's all there is to it. There is, however, another kind of branch instruction that does save a return address—bsr, "branch to subroutine."

Bsr has nothing to do with the CCR. It simply loads the program counter with the address represented by the label in its destination field, and, as a result, the 68000 starts executing instructions from this new location. At the same time, however, it saves the address of the instruction immediately after the bsr, and pushes it on to the stack. Now, all we need to do to "return" from the subroutine is pop this address off the stack and load it into the program counter, and presto—the 68000 will be back where it would have been if it had "fallen through" the bsr instruction instead of branching. As we learned last time, there is an instruction that does precisely that: namely, rts, "return from subroutine."

Although bsr is not affected by the condition codes, it's easy to set up subroutine calls based on conditions. You simply branch "around" the bsr using a branch instruction that tests for the opposite condition. Or you can label the bsr itself, and branch to it or not, depending on the original condition.

The program.

Having done most of the necessary theoretical work, it's time to take a look at this month's program. Type it in, then assemble and link it under COMMAND.TOS, following the same steps as for our first program from installment 2; however, you should give the file a different name (try *X*—I'm fond of it). As before, the source code is compatible with the AS68 assembler from the Atari Developer's Kit. Now run the program either as a .TOS file from the desktop or by typing its name from COMMAND.TOS.

The program begins by printing a *Hello* message, then waits for you to strike some keys. It will display the keys you type, but pressing RETURN will only move the cursor back to the beginning of the same line, without moving it to a new line. However, if you press the ESCAPE, F10, up arrow or down arrow keys, a message is displayed identifying the key you pressed. If you press F1, the program prints *Good-bye* and terminates.

As you know, pressing the ESCAPE or cursor or function keys will normally not display anything at all. What's more, the two latter groups don't even return ASCII values. These

keys are detected by means of their "scan codes." Scan codes are values unique to each separate key on the keyboard, which low-level system routines use to identify keys. The ASCII values that are typically returned to, say, a BASIC statement are built by the system from the original scan codes. It turns out that the same GEMDOS routine that returns ASCII keycodes also returns scan codes, and we'll soon see how to access them.

The first four instructions should be familiar—they use GEMDOS function 9 to print the sign-on message, just as in our first program. The next instruction—bsr keycodes—branches to the label keycodes and saves the return address of the call. Some time after the branch is made, an rts instruction will be executed. When that happens, the 68000 will "return" to where it would have been if the bsr had never executed—namely, the move instruction immediately following. The next five instructions print the sign-off message and execute a GEMDOS 0 call, which, as we know, ends the program and returns to the operating system.

The subroutine.

Right after this comes the label keycodes. This is where the 68000 "comes" when the bsr above is executed, and everything between this location and the concluding rts (inclusive) is, therefore, a subroutine which we can call "keycodes." Let's look at what it does.

The first three instructions call GEMDOS function 1. If you look this function up in one of the standard references available, you'll find that it's called "Conin" (from "Console In"). Calling GEMDOS 1 gets you the ASCII code of the next key pressed. This code can be found in the low byte of register d0. (This is a pretty standard rule on the ST—subroutines, if they return a value in a register, return it in d0.) The documentation also tells us that the scan code of the same key is returned in the low byte of the high word of register d0. It's easy to get the low byte of a register—just specify a .b data size field and move the value. But how can we isolate and retrieve byte 2 (counting from 0 from the right)?

Since individual bytes within a register do not have addresses, we have to tackle this problem in a different way. There are at least three ways of going about this in 68000 assembly language; I've chosen to introduce a useful but simple new instruction.

The logical shift.

The 68000 provides a set of instructions that allow us to "shift" the contents of registers, bit by bit, in either direction. The lsr instruction—"logical shift right"—moves the contents of the destination data register as many bits to the right as is specified in the source field. The number specifying the shift count is, of course, an immediate value.

As the register's contents are shifted, the bits that "fall out" of the right "end" of the register are lost (there is no truth to the rumor that a company in California is developing a gadget that will save these discarded bits, reassemble them into bytes, and allow you to use them as extra memory). On the other end, zeros are shifted in to take the place of whatever data has been moved over. That's why this is called a "logical" shift—nothing special happens, only what you'd expect. (An "arithmetic" shift, on the other hand, does have such a special property.)

One restriction on the form of the instruction we're using is that the shift count specified cannot be greater than 8. So, in order to move byte 2 over to byte 0's position, we have to repeat the operation, thus moving d0's contents a total of 16 bits to the right, and gaining access to the scan code, which is now the low byte of d0.

With the scan code ready to be examined, we can now load registers a0 and a1 with the base addresses of a couple of tables, and d1 with a loop count. Now we can find out what scan code we have—or, at least, what codes we don't have.

The tables.

First, the tables. There are two of them: codes and keys. Codes can be found in the label field, where you'd expect it. In the next field is a new assembler directive, equ. The directive equ stands for "equals" or "equates to." For example, the sequence F1 equ $3b (see the very beginning of the data segment) assigns the hex value 3b to the label F1. I can now use this label in my program instead of the number. There's one important thing to remember. Equ does not reserve any space in the assembler's work area, and no object code is generated when it is used. Instead, the assembler simply "remembers," as it assembles your source file, that you want F1 to stand for $3b. Once the assembly is over, and $3b has been used in all the places you specified with F1, all other trace of the equ "operation" has disappeared.

For this reason, directives like equ (most assemblers have them) are sometimes called "pseudo-operations," because they look like "genuine" machine operations in the source code, and yet they don't generate any object code.

At the beginning of the codes table you'll find the line codes equ*. The * is, in this case, a special symbol that stands for the current value of the assembler's location counter (which we discussed last time). In other words, this line of code assigns to the label codes the assembler's current working location in memory. This lets us record in an easy way the starting, or base, location of the table of values that follows.

Four lines follow, each reserving a word of memory, associating it with a label, and initializing it with a hex value. These latter happen to be the scan codes for each of the keys identified in the comment field. Note that F10, for example, is a label associated with a real address. F1, however, can't strictly be called a label at all, since it doesn't represent an address—rather, it's just a name associated with a value. Note also that the addresses "labelled" by codes and ESC are the same—codes is used as a convenient reference point for the programmer. At the end is another equate I've inserted to record the number of items in this table, four. The reason I didn't use equs in this table will become clear in a moment.

The second table, keys, looks more complicated. Its base is labelled as was the codes table. The last four items here are strings containing the messages to be displayed when one of the four "special" keys is pressed. Each string is preceded and followed by a linefeed/carriage return sequence, to separate the message from what else may be on the cur-

rent line at the time, and terminated, of course, by a zero. I've also labelled the "end" of the table—really the first byte after the end of the table—with k_end, although I don't use this label in the program.

What about the first four items? Each consists of a "define constant" code, with a longword size specifier, followed by the label of the next of the four strings. This has the effect of inserting the addresses of the strings, in order, as the first four items of this table. In effect, we now have an array of four strings which we can index almost as easily as we could in C or BASIC.

Note that address items are not labelled—there is no requirement to label a line of code using the dc directive. I labelled keycodes in the first table, but purely for reference—I don't use the labels in the program. The equ pseudo-op, however, for obvious reasons, does require a label.

Using the tables.

Now let's see what happens in the rest of the subroutine keycodes. As noted, the table base addresses are loaded into two address registers. The number of items in codes is loaded into d1; this will be our loop counter. Now we're ready to go.

You'll recall that the scan code we want to look at is still in register d0. We now want to see whether this code matches any of the special codes we're looking for. In high-level languages we determine whether one value is equal to another by "comparing" them with various Boolean operators. At the assembly language level, however, we now find out how a computer really compares two values: it subtracts one from the other. If the result is zero, the values are equal.

Obviously, performing a subtract operation each time we want to do this is going to be somewhat laborious: we'd have to save a copy of the operand subtracted from, in order not to lose it as a result of the operation. The cmp—"compare"—instruction rescues us from this situation. It operates like a subtract. The source operand is subtracted from the destination, and the condition codes are set just as before. The only difference is that cmp has no effect on the operands. So we can perform all the compares we want without worrying about what's happening to the values being compared.

The first line of code after the label loop (the assembler considers this to be one single line, although I've broken it up in the source listing, for clarity) compares the first item in the table with the contents of d0. Both items are word size: the scan code in d0 is in the low byte, but since there's nothing in the next byte (we know that GEMDOS 1 returns byte values), there's no harm in looking at the two bytes together. The address mode used to access the table is Address Register indirect with Postincrement. We learned it last time in connection with stack operations. Here its use has nothing to do with a stack. It simply reads the current table item for us and then automatically increments the value in a1 so that it points to the next item.

If the two values are not equal, then the bne instruction causes a branch down to the label next. Here we add 4 to the value in a0; this register now points to the address of the next message in keys. Then we subtract 1 from the loop counter, d1. If the result of this subtraction is 0, then we've looked at all the values in our table and the loop is finished. We now, as a last resort, compare the scan code to the value we equated to F1. If they're equal, then our special "exit" key has been pressed, and we return from the subroutine by executing an rts. Otherwise, we branch back to the line labelled keycodes, wait for the next keypress, and try again.

In the body of the loop, what happens if one of the codes does match the value in d0? In that case, all we have to do is push the current value of a0 onto the stack—we know that it points to the address of the message string that "goes with" the current compared scan code—and execute a GEMDOS 9 call, displaying the string.

Now you should be able to see why we had to use the dc directive to reserve actual space in memory for the table items, but not for the F1 value. Our tables work just like arrays. Instead of looking through them with subscripting operations, however, we have to use the addresses of the table items. But in order to have addresses, the items accessed must exist in memory. On the other hand, our final comparison for F1 is a one-shot affair. There's no reason not to code the value directly into the instruction, since there's only one value. I could have declared F1 and accessed it the same way as I did the table items, but there would have been little reason for doing so.

Also note that, because I am storing word and longword values in these tables, it is necessary to use the even directive to make sure the tables begin at even addresses.

There's one last trick at the end of the program, which I should mention. Note that I didn't bother to pop the arguments to the last GEMDOS 9 call (to display the sign-off) before executing the GEMDOS 0 call. That's because GEMDOS 0 doesn't use a return address from the stack to return to the operating system, and it automatically restores the stack situation as it existed before the present program began execution. In this case, then, I can get away with neglecting the stack, since I won't be using it any more.

The codes and keys tables could be made more efficient by being combined into one table. To take full advantage of such an arrangement, though, we need to learn a new addressing mode. That will be one of the items on our agenda for next time. In the meantime, play around with the table organization and see what you come up with. //

(Listing starts on page 80)

UTILITY

ALL RESOLUTIONS

Raster Sprite Editor

A SPECIAL INCLUSION

Create your own sprites and icons with this easy-to-use program.

by Peter Beery

The **Raster Sprite Editor** (**Raster**, hereafter) is a general-purpose sprite-editor/graphics-shell. Nested menus and a variety of help messages make this program easy to use in all three ST resolutions.

The C listing is too lengthy to publish in these pages, so our readers will find it on the disk version and on the Atari Users' Group SIG on Delphi. Additionally, the Megamax C source code contains a number of routines that may be of general usefulness to other C programmers—including the management of menus, raster copies, and disk I/O (with folder access allowed). Here is a list of program features.

Sprite Editor.

Cut and save sprites from any **NEO-Chrome** or **DEGAS** file. Save as hex code for direct inclusion within your program, or save in the SPR binary form that can be loaded directly into screen memory by your program.

Picture Editor.

Convert pictures from **NEO-Chrome** format to **DEGAS**, and vice versa. Working with two files simultaneously, you can paste images from one to the other using any of the sixteen standard raster copy modes. There's also an additional, "masked" mode, which lets the background show through the clear portions of the sprite. Save the result in either of the two formats.

Demo Mode.

This works in the same manner as the paste function, but doesn't alter the original picture.

Graphics Shell.

Run **DEGAS**, **NEO-Chrome** or any other GEM program from within the shell. When you return, the contents of all six picture buffers will be as you left them.

Touch Up.

Compose new artwork or touch up the old with the built-in drawing and fill functions. An additional feature allows you to include your own custom drawing routines. The program LINE__FAN.PRG (on the disk) serves as an example around which you can design these routines.

Picture Buffers.

There are two primary picture buffers, two working screens (one shown, one hidden), one sprite buffer and one sprite mask buffer.

Top Menu.

This is the user's I/O interface. All disk and shell operations are handled here.

Bottom Menu.

This is the "work" menu, where all cutting and pasting functions are located, including the demo routine. Many of these routines prompt for the "next logical" menu selection following some action; for example, "show sprite" after a cut operation. This provides a shortcut to using the menus in many situations. Additionally, most operations can be repeated by simply pressing the mouse button anywhere on the screen. This mode, which is *not* the default, can be selected with the "toggle auto repeat" option under the "miscellaneous" selection on the work menu.

Touch-up Menu.

This is the level that allows you to create or modify any of your art, including masks, sprites and/or picture files.

On the menu.

Most of the menu items are self-explanatory, so I'll limit myself to explaining those that may confuse you.

*LOAD. . .**DEGAS** Picture* or *LOAD. . .**Neo-Chrome** Picture* will flip up a file selector box showing the default drive. Choose another drive or folder if you need to. The picture will load into the oldest picture buffer, then display it. Thus, if you load two pictures, both will remain in memory.

SAVE-Append Pic allows you to add the current picture to the end of any other file. It only adds the screen information (32000 bytes), and ignores any palette or resolution data. This allows you to have several screens of data in one file and still keep the **NEO-Chrome** color-rotation information, for example.

EDIT-Horizontal Flip creates a mirror image of the current screen.

EDIT-Vertical Flip flips the screen about the other axis, so that words, etc., become inverted.

DEMO-Exchange Buffers will flip the active and inactive picture buffers. The active file is the one that is shown. It can be altered via the paste routine.

MODE-Masked selects the masked mode. See below for a quick word about masks. The important thing here, however, is that this mode allows the use of solid sprites which have any shape.

MODE-(The rest) sets the current paste and demo modes. These are the logical operation modes used by the vro_cpyfm VDI routine.

EDIT-Restore is a boo-boo rectifier which will only work immediately after a paste or touch-up operation. It's that "one more" chance you've always hoped for.

DEMO-Ghost is a demonstration that ignores the current mode and uses the source "ORed" with the destination (in C: S | D) in its place.

DEMO-Solid is the same as the above, but yields a "masked" copy.

EXTRA-User Functions. When you select this menu item, you'll get a file selector that lets you load any program you have stored on disk. It puts you back into the touch-up buffer and then runs the program. Thus, any changes made on this screen by the program will be permanent. This allows each user to add custom routines written in any compiled or stand-alone language.

MODE-Spill Bucket selects and/or does contour fill.

DESK,MISC.,EXTRA-.acc* are desk accessory access points. Note: use the control panel accessory to edit colors.

A word about masks.

The mask used in this program is a simple one. It sets all pixels of color index 0 to color index 1. All other pixels are set to color index 0. This creates what looks like a black-and-white negative of the sprite. This mask is ANDed with the screen. Everywhere the mask was white, the background scene shows through. Everywhere the mask was dark, the background is blanked out. When the sprite is ORed with this result, it fits nicely into these blank areas.

Note that there's nothing special about using color index 0 in the first part of this mask. You, the programmer, can use another—or several others—so that some of your sprites have colored, translucent sections. Your imagination is the limit.

Using it.

First of all, the best way to understand the many features and shortcuts in **Raster** is to play with it. For now, however, let me give you a few hints, and also outline the procedure to follow for some of the most common uses.

(1) Read the alert boxes. They'll tell you how to use each feature. When they say *press this button* or *hit this key*, it generally means *press and hold* for a fraction of a second. I haven't used single-or double-clicking in this program; if you're too quick on the button, the command will be ignored.

(2) If you use a desk accessory, you may find the screen's dirty and needs to be redrawn. Press and hold the mouse button anywhere on-screen. When you let go, the screen will redraw itself. This will work on either of the top two menus. For the touch-up menu, the redraws are mostly automatic. The exception is if *you* drag the accessory too far from the middle of the screen. If this happens, it will be overwritten. While the problem can be overcome with a little patience and a good memory, it isn't a whole lot of fun. Hence, I suggest that you not drag desk accessories around on this level. Note that you can still drag away to your heart's content on the top two menus.

(3) The alert boxes with the buttons <<, Ok and >> will automatically reverse direction when you reach the end of the of the valid range. For example, if you keep hitting RETURN while the touch-up SPILL alert box is showing, you will cycle through all the fill patterns continuously.

NEO to *DEGAS* conversion.

Select the "Neo" option under "Load" in the topmost (main) menu. Read the file selector that pops up. All the .NEO files, as well as all the folders on the current disk, should be displayed. If you want to see inside a folder just click on it. If the drive is *not* the one you want, then *click* on the topmost line of text in the window. It should look something like: *A:*.NEO.* Backspace until you've cleared the line and type in the new drive, for example: *D:*.*. *Don't* hit RETURN at the end. Rather, click on the little square in the top left corner of the inner window.

This window should now show the contents of the new drive. Once you see the file you want, click on it and hit RETURN. You can also double click on it. The file will load into the next buffer, which will now become the current buffer and be displayed. Now, go back to the menu bar under "Save" and select "Picture." You will get an alert box asking which format. Choose **DEGAS**.

Now, you'll see another file selector. Deal with it in the same manner as before. But note that you can also type the filename in explicitly, in the space provided on the right-hand side.

Cut and save a sprite.

Load a picture as in the above section. Now go over to the run menu title and select "work menu." This will drop you down to the next menu level. Go to edit and select "Cut sprite." You will now get an alert box giving instructions on how to cut the sprite. Basically, the the idea is to move the mouse to the top left corner of the region you want to copy. Then press and hold the mouse button while you move the mouse to the bottom right corner of this region.

You should see a box form outlining the current region. When you let go of the mouse button, this box will disappear—and you'll get a new alert box, asking you if you wish to see the sprite. Go ahead and look at it. The screen will switch to the sprite buffer. Press the mouse button, and you'll get another alert box that asks if you wish to save the sprite.

Let's say that it isn't what you want; answer *no*. You'll go back to the normal screen, ready to recut your sprite. This time, you get it right and thus answer *yes* to the save question. You'll be given the choice of storing the sprite in hexadecimal format (which can be used within your source code), or in binary format (which can be written directly

to the screen). This last can also be reloaded from disk by **Raster**. Note that this alert box tells you how the resolution, height, width and actual data are written to the disk.

As a final note, the sprite can also be saved via the "Sprite.SPR" and "Sprite.HEX" items under "Save" in the top (main) menu.

Paste a sprite.

Cut the sprite as above, but this time click on "Exit" when asked to view the sprite. Go over to the mode title and select a paste mode. These may very well be confusing; the only way to understand what they'll do is to try them. This is why I added the demo routine. Under this title, "Current" refers to the current mode selection.

Again, you'll get a dialog box explaining how to use the demo. The masked mode will cause a mask to be generated if one doesn't already exist. I find this the most useful mode. Once you've decided on a mode, go over to edit and select "Paste."

You'll receive directions which will be the same as those in the demo—except the exit buttons are reversed. This will minimize the number of mistakes you'll make. If you do make a mistake, select the "Restore" item the very next thing. This option will remain valid for only one menu-related action after each paste or return from the touch-up menu. Also note that you can modify this sprite and/or mask via the routines of the touch-up menu. You can also switch the picture buffer you'll paste to with the "Exchange Buffer" item of the demo title. The return to the main menu is found here, too.

Other programs from within *Raster.*

Go to the top menu and select either NEO.PRG, DEGAS.PRG or OTHER from under the run title. **NEO-Chrome** or **DEGAS** will be loaded from drive A if there. If they're on another disk or within a folder, or if you want to run something else, then use the OTHER option. This will give you a file selector. When you return from the other program, the screen will be redrawn, with things as they were.

If you want the screen to be altered, you'll have to access your program from within the touch-up menu via the user function interface located under "User Functions" of the extra title. LINE__FAN.PRG is an example of what one of these looks like and how to write one.

So here you are, ready to go. But, before you leave, let me thank the creators of Megamax C. Overall, I'm extremely impressed with the quality of this development package. The support staff at Megamax is friendly and knowledgeable. By the way, the Library source code disk for $50 is a good deal, for those of you who need more information on these routines, especially the floating-point math package.

I would like to offer special thanks to the following friends for their help and suggestions: Cash McManus, Dan Shibata, Don Turnock, Ravi and Vandana Chuckravarti, and my brother Andrew Beery. //

Peter Beery did his graduate studies in physics at the University of Notre Dame. The ST is his first personal computer. He chose C, on the advice of friends at AT&T, as a new language to learn for use in development.

STs in the UK

A link to Atarians in the British Isles and on the Continent.

by Dale Hughes

Greetings to all readers of **ST-Log**, from across the big pond. In this article and occasional updates, I'll be bringing you a view of ST computing in Britain and Europe. We'll also look at British-written software from GST, Fleet Street, Psygnosis and Kuma, and we'll visit the Atari spring show in London.

In the business world, the ST is making great inroads in Britain. In these pages we'll investigate this welcome development, in regard to desktop publishing, hardware and business applications. As a great believer in user groups, I'll be listing the best of the British groups, and I hope U.S. groups will respond, to keep our global village in touch.

In future articles, we'll also check out communication possibilities between Europe and the U.S., personalities in the ST world, and the range of ST books and other publications.

It's an exciting prospect, and I hope you enjoy it as much as I do. So sit back, fasten your seat belt and relax. It's first class on the Concord, so why not enjoy a wee dram of highland malt whiskey, feast on Scottish smoked salmon, and enjoy the journey; after all, we are cousins.

A history lesson.

This is a dedicated ST magazine, and I wouldn't want it any other way. Would you, the reader, please indulge me for a bit, while I explain the British computer market? I promise it's important—and necessary for an understanding of the phenomenal success of the ST here. Besides, you might learn something.

In the 8-bit market, the British doggedly continue to favor the old standards. The long-lived Commodore 64's alive and well, with new blood being added by GEOS, a wimp environment which provides an ST-like desktop at the expense of a tired and much overworked 8-bit processor.

The Sinclair Spectrum began life in the U.S. as the Timex Spectrum, with a "massive" 3K of memory. In Britain, it was just the Spectrum. These days, it's 128K of 8/16-bit computer with micro drives, a large user base and thousands of games. The headlines about Sinclair are, more often than not, about its eccentric developer, Sir Clive Sinclair, than about any hardware developments.

Sir Clive (the "'Sir" was added after he revolutionized the computer industry in Britain) is what I would call one of the last great inventors. After developing the Sinclair, he went totally bonkers with two financial failures. The first, a hand-held color TV, was brilliant in design—and totally devoid of any sales. His next invention was a three-wheel car which met with a similar fate. Since then, he has wisely gone back to computers and recently announced a new Sinclair, the Z88 portable.

The BBC is the last of the 8-bits to dominate the home market. Following Apple's example, the BBC was sold to the school and education market. Although just a renamed Acorn PC, by adopting the name and backing of the British Broadcasting Corporation, the BBC has become a programming and educational standard. So, Apples=Acorns.

IBM and its clones are the corporate standard here—as they are everywhere else—and that's enough said about that. The recent release of the Atari PC (more on that later) should give the ST design a piece of this corporate market.

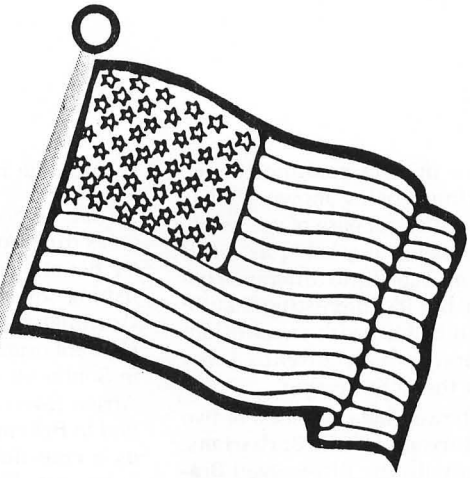
The 16-bit market is a different world altogether. The Apple Macintosh is so blatantly overpriced in Britain that it's virtually unobtainable for the home user. This costliness has always been a feature in Apple's European market, and the Mac maintains the tradition with a vengeance. The prices are even more remarkable, as the Apple II+ was assembled in Ireland.

The Amiga, although partly a British development, is as elusive as the Loch Ness monster. Only now are units being sold in the retail marketplace. Until recently, it's been hyperbole and promises—a very difficult way to sell a computer system in a competitive market.

Other PC manufacturers include the business oriented Amstrad, Apricot and Olivetti machines. All these 16-bit computers are aimed at a special segment of the computing public, whether it be the business, publishing, educational or home markets.

Power minus price.

Into this environment came the ST, in 1985. The ST brought to Europe something rarely been seen before—a quality prod-



uct that was affordable. Seeing that it has sold over 25,000 units, mainly by mail order in Britain, and is set to triple this figure in the next year, the ST has been a phenomenal success story.

Remember, we are a country of only 20 million, one-tenth the size of the U.S. Yet there are three monthly ST publications, and dozens of user groups, bulletin boards, retailers and mail-order houses. . . plus an ever-growing number of software houses and hardware manufacturers, and a fanatical following of users which would rival the early days of the Apple II.

With Atari's recent price reductions and the splitting up of the 520 and 1040 lines, more and more often the disenchanted British 8-bit owner is deciding the ST is the machine to own—and at a price that leaves a bit of change.

Distribution problems.

This is not to say that all's rosy in the British ST market. There are distribution problems to be solved before the ST becomes a household item. As I stated earlier, most STs are purchased through the mail. This isn't because of any financial savings or convenience; it's because there are remarkably few retail outlets stocking the ST. British chain stores that regularly carry competing PC brands have yet to realize the ST's potential. Thus we have a situation which parallels that of a great new film or exotic new restaurant—advertisement by word of mouth.

There are also some software problems. The British pound is worth, on average, about one and a half times a U.S. dollar; so, in theory, a pound sterling should purchase about $1.50-worth of product. The reality, though, is that both imported and homegrown software products are priced, on an average, 30 percent higher than those purchased in the U.S. It would be most welcome if the software houses would use a pricing policy like that of Atari. A warning to the summer traveller planning on visiting Europe to pick up some cheap software: find another reason to make the trip. Fortunately, there *are* a few other things to do.

On the whole, the ST range is alive and flourishing in Europe. The Mega STs are now starting to appear, and the IBM clones will, this summer, grudgingly begin working into the European business world. Desktop publishing fever is rising here, since the release of **Publishing Partner** and **Fleet Street Publisher**. With the imminent release of Atari's laser printer, this market can only improve.

Yet it is in the games market that the ST has captured the imagination of the British public. Calling a computer a game machine has, in the past, demeaned its character, but I totally disagree with this thinking. The ST, with its stunning graphics and GEM desktop, has brought into computing a whole new generation of British users who would have stuck with their 8-bit machines forever. The ST has changed all that, by being programmable, affordable and lovable. If games are the medium that get the message across, I say: more power to them. Besides, I like games.

The London Show.

Late April brings two major events to Britain: spring and the Atari show. Spring has a way of postponing its arrival till well into August, but the Atari show is always on time—and well worth the wait. This year's show was the biggest and best ever, with almost one hundred hardware and software vendors exhibiting. Such importance was placed on the hardware announcements that "the man" himself, Jack Tramiel, attended to give the assembled press a pep talk and a send-off.

Atari Corp. chose London for the world debut of two Atari PCs. The Entry Level System is a standard clone, featuring an 8088 microprocessor with a switchable clock speed of 4.77 or 8 MHz. Graphics mode is built in, and the unit comes with one 5¼-inch disk drive. The Atari PC-Expandable System differs from the Entry Level offering, in that it can be configured as either a dual disk drive or a single drive and hard disk system. Also standard are five expansion slots on the Expandable System.

Another product launched in the U.K. was the Atari SLM Laser Printer. With a resolution of 300 DPI and a speed of 8 pages per minute, the SLM is a winning addition to the Atari lineup. We can only hope that prices for the SLM continue to follow Atari's "Power Without the Price" concept. If so, even we normal people might be able to buy a laser printer soon.

STs in the UK *continued*

Also on view was the Mega ST, which, it goes without saying, is the machine I covet above all others. As you may know, the Mega ST comes with 2 or 4 megabytes of RAM, detachable keyboard, one D/S disk drive and a battery-backed clock/calendar. The unit is designed with open architecture for easy expansion.

For dedicated game players, the XE Computer Games System was launched. All cartridge game players should give this machine a look.

The latest in software.

British software houses had a bundle of new issues and upgrades on display. GST released version 2.02 of **1st Word Plus**. This new version, a definite improvement over **1st Word**, comes with a built-in dictionary, mail merge, better printer drivers and a graphics snapshot accessory for text with graphics documents. There are a host of other minor changes, all designed to improve on the success of **1st Word**, and sure to set the standard for *others* to improve on.

Both **Fleet Street Publisher** and **Publishing Partner** (**PP**) were represented, and both were up and running on the Atari Laser Printers being shown. **PP** has a growing reputation for ease of use, while **Fleet Street** is being marketed as a full business system. I see little difference between the two, but it's interesting that **PP** is being sold in Britain with an additional graphics library which currently is not bundled with the U.S. version.

Psygnosis showed beta versions of two new games: **Terropods** and **Barbarians**. Those familiar with the ill-received **Brataccas** will see a similarity in screen design in both new games, but, fortunately, Psygnosis has solved the movement problem by using mouse-controlled icons at the bottom of the screen. I've always had a soft spot for Psygnosis, because they're based in Liverpool, a city with one of the highest unemployment rates in all of Europe. If they can create jobs *and* give ST users some extra enjoyment, more power to them.

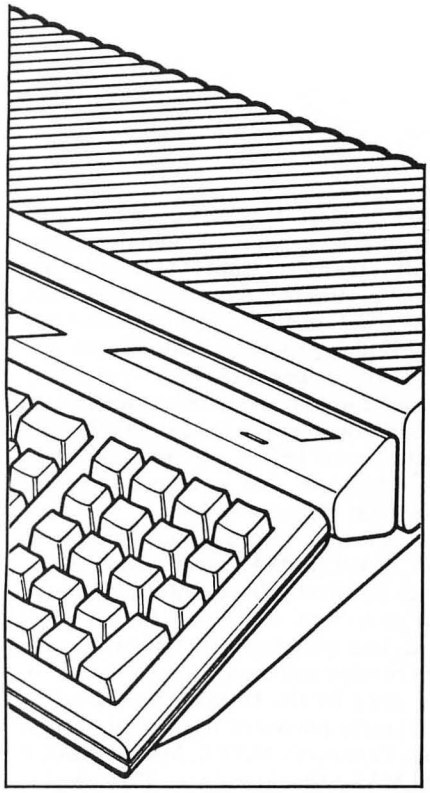
There's more—much more—but I've tried to concentrate on the British contribution to the Atari revolution. I hope you have all enjoyed this first installment.

Until the next time, cheers.

Dale Hughes holds B.S. and M.A. degrees from California State University. After his stint at CSU, he spent two years fishing for salmon on a reservation in Washington and driving a garbage truck. He went overseas in 1973, to live and work in Southeast Asia, the Middle East, North Africa (even Libya) and Europe. He settled in Britain in 1978 and currently works as a consulting engineer and free-lance journalist. He's been a dedicated "STer" since 1985.

Assembly line *continued from page 74*

```
*** X.S
        text                            code segment
        move.l  #sign_on,-(a7)          get start message
        move.w  #9,-(a7)                code=display string
        trap    #1                      do it
        addq.l  #6,a7                   fix stack

        bsr     keycodes                go look at keyboard

        move.l  #sign_off,-(a7)         get end message
        move.w  #9,-(a7)                code=display string
        trap    #1                      do it
        move.w  #0,-(a7)                code=exit program
        trap    #1                      do it

keycodes:
        move.w  #1,-(a7)                code=conin
        trap    #1                      do it
        addq.l  #2,a7                   pop arg
        lsr.l   #8,d0                   get scan code
        lsr.l   #8,d0

        movea.l #keys,a0                messages
        movea.l #codes,a1               scan codes
        move.l  #nrcodes,d1             number of good keys
loop:
        cmp.w   (a1)+,d0                good code?
        bne.s   next                    if not
        move.l  (a0),-(a7)              else get message
        move.w  #9,-(a7)                code=display string
        trap    #1                      do it
        addq.l  #6,a7                   pop args
        bra     keycodes                get another key
next:
        addq.l  #4,a0                   point to next message
        subq.l  #1,d1                   dec loop count
        bne     loop                    if more special keys
        cmpi.w  #F1,d0                  else: F1 key?
        bne     keycodes                if not
keys_exit:
        rts                             else leave

        data                            data segment
F1      equ     $3b                     F1 key equate

sign_on  dc.b   'Hello',10,13,0         start message
sign_off dc.b   'Good-bye',10,13,0      end message

        even                            align table
codes   equ     *                       table of scan codes
ESC     dc.w    $01                     escape
CUP     dc.w    $48                     cursor up
CDN     dc.w    $50                     cursor down
F10     dc.w    $44                     function 10
nrcodes equ     4                       manually entered

        even                            align table
keys    equ     *                       table of messages
        dc.l    k1                      address of first msg
        dc.l    k2                      of 2nd message
        dc.l    k3                      of 3rd message
        dc.l    k4                      of 4th message
k1      dc.b    10,13,'Escape key',10,13,0
k2      dc.b    10,13,'Cursor Up',10,13,0
k3      dc.b    10,13,'Cursor Down',10,13,0
k4      dc.b    10,13,'F10 key',10,13,0
k_end   equ     *
```



ST user

THIS MONTH:

Rescuing(?) Batteries Included, suit-ability, suppression, and more.

by Arthur Leyenberger

By now, you're probably aware that Batteries Included, the Canadian software company which published a number of excellent programs for both the 8-bit and ST computers, is no longer with us. You probably also know that Electronic Arts has purchased all the existing BI programs and will continue to market most of them as a separate line of products under the EA name. For complete information as the story continues to unfold, keep reading **ST-Log**.

What's interesting about this purchase is that EA also bought all the artist's (authors') contracts for existing and future products. Most notable is the still forthcoming (maybe?) **PaperClip Elite** word processor for the Atari ST. **PaperClip Elite** has been near completion for quite some time, according to BI officials, and much of the ST community has been eagerly awaiting its publication.

I've seen preliminary versions of the program; it was super. It definitely pushed the envelope on ST word processors. According to an EA spokesperson, **PaperClip Elite** is undergoing close scrutiny, to determine if it's viable. As I write, this program and a number of others are being evaluated. We should know the outcome in a few months. I, for one, hope **PaperClip Elite** makes it to market.

Out-of-Control Department.

This is getting ridiculous. More and more computer software companies are suing each other over the "look and feel" issue. I've discussed it before. Apple computer threatened to sue Digital Research, creators of the GEM operating system we all know and use, claiming that GEM's windows, icons, etc., infringed on Apple's Macintosh OS. In other words, GEM "looked and felt" too much like the Mac interface. But, hey, where did Apple get the idea for the Mac? Xerox's Palo Alto Research Center (PARC), that's where.

I've also mentioned that Broderbund Software, makers of the excellent and popular **Print Shop** for the Atari 8-bits, sued Unison World for their ST program, **PrintMaster**. Although **PrintMaster** contained some improvements (it was actually a fuller implementation of the program concept than was Broderbund's product) and the program code was different, the "look and feel" of it was almost identical to that of **The Print Shop**. Broderbund won, in effect, as the two companies settled out of court. Both Apple and Broderbund were successful in requiring their imitators to change the way their software looked to the user.

This lunacy continues. Lotus Development Corporation makes an excellent spreadsheet program for the MS-DOS computer, called Lotus 1-2-3. It's been around for a few years and is the predominant spreadsheet program in the IBM (and clone) market. If you're familiar with the non-GEM-based **VIP Professional**, you know what Lotus 1-2-3 is—because **VIP** is an exact visual clone. Although the code is different (since the ST and IBM PC use different microprocessors), the program looks and works the same.

Arthur Leyenberger is a human factors psychologist and free-lance writer living in New Jersey. He's been writing about computers for over five years and continues to be an Atari enthusiast. When not computing, he enjoys playing with robotic toys.
CompuServe — 71266,46
Delphi — NJANALOG

There are two companies who make MS-DOS clone versions of 1-2-3. Their names aren't important; the fact that Lotus is suing them for copyright infringement is. Lotus claims that the command names and the hierarchical horizontal menu at the top of the 1-2-3 screen are owned by Lotus Development. But wait a minute, where did Lotus get the idea for their spreadsheet? Why, from **VisiCalc**, of course. In fact, Mitch Kapor, who started the Lotus firm, is an ex-employee of Software Arts, makers of **VisiCalc**. Sounds pretty fishy to me.

On and on it goes; where it will stop nobody knows. I know where it *should* stop: here and now. Unless this nonsense is halted, innovation will definitely be discouraged. What is innovation if not, often, the further improvement of existing products or the creation of new (albeit familiar) products. The computer industry needs standardization. Knowing Atari DOS does not necessarily mean you know how to use Apple DOS, MS-DOS or CP/M. But knowing GEM will probably mean you can use a Macintosh without much help. And knowing how to navigate your way around Lotus 1-2-3 means you can use **VIP Professional** or the other clone programs.

If these lawsuits continue to proliferate, standardization will be further away than ever. Rather than create new programs using existing interfaces and commands users are familiar with, software companies will have no choice but to produce completely new programs—programs we will have to learn how to use all over again. And that's the *last* thing this industry needs.

I admit that the legal questions underlying these copyright issues are thorny indeed. Who is to say where the line is drawn between an idea and the expression of that idea? Further, the authors and publishers of software programs deserve to be compensated for their work. No one should be allowed to steal ideas and make money from them without royalties or some other arrangement.

But I am a *user*. And, as a user, I don't want to have to learn new ways of running software. Moreover, the very nature of separate, unique user interfaces will drive up the cost of software. They'll cost more to create and more to train users. And this kind of system will prevent good programs, such as Lotus 1-2-3, from becoming (as the Army says) "all that [they] can be." Improving existing products only benefits one person: the user—the most important component in the human-computer equation.

Wanna buy some protection?

No, this isn't a review of *The Untouchables*. If you want violence, see the movie. We're talking about *hardware* protection—surge suppressors. Surge suppressor is a generic name for a device inserted between the AC outlet and your computer. By limiting extra-high or extra-low voltages, it keeps your computer's electronic components from being fried.

These products come in a variety of shapes and sizes. Some simply offer protection against temporary power surges, often called "spikes." Others offer protection from undervoltage which might occur during a brownout. Still others protect against overvoltage. Of course, there are products that combine several of these features, and condition the telephone-line connection to your modem and computer, too. Some offer power-line protection features in addition to a multi-outlet strip.

Many computer owners don't use a power-line conditioner when they should. How do you know if you need one? Only when it's too late—like insurance. If you have hundreds of dollars (or even more) tied up in your computer equipment, then an under-$100 investment in one of these products makes a lot of sense.

The problem is that there are dozens of companies manufacturing these products, each with a multitude of offerings. How do you choose a power-line conditioner that meets your needs? More importantly, how do you choose one that *works*?

A leading MS-DOS computer magazine recently commissioned an independent laboratory to test power-line conditioners. Dozens of these protectors were tested by zapping them with high voltage, gradually decreasing the voltage to determine the effect on the attached computers, and performing other tests. Several products stood out from the crowd, by consistently performing as they were supposed to. One particular brand that did so: Panamax (150 Mitchell Blvd., San Rafael, CA 94903 — (800) 472-5555).

Not only did Panamax products pass these rigorous tests, but having used several of their models for over a year, I can vouch for their survival of lightning storms and power outages. Although I may *never* know if I've been the victim of any of these power problems, I do know that my computers still function. I also have peace of mind due to my "power insurance." Here are details on some of the models I can recommend.

My Atari ST, Supra hard disk and modem are plugged directly into the **SuperMAX**. The fourth **SuperMAX** outlet has an outlet strip plugged into it, for the disk drives and monitors. The telephone line is also plugged into the **SuperMAX**. For $149, the **SuperMAX** gives me surge, noise, brownout and data-line protection. In addition, there's a lighted on/off toggle switch, reset button, circuit breaker, replaceable fuse, and separate surge and brownout indicator lights.

If you want to protect primarily your hard disk, **UltraMAX** is the answer. It provides transient voltage, noise and brownout protection. It comes with a six-foot cord, four receptacles, on/off switch, circuit breaker and undervoltage alarm. Retail price for the **UltraMAX** is $149.

If you're interested in surge protection only, Panamax's **MAX 6**, **MAX 4**, **MAX 2** and **MAX 1** should suit you. They provide six, four, two or one receptacle, respectively. In addition, the **MAX 6** and **MAX 4** have circuit breakers and on/off switches.

For data-line protection, the **TeleMAX** is specifically designed to handle voltage spikes on phone lines, to protect RJ11- and RJ45-compatible devices like phones, answering machines and, of course, our modems. The one-receptacle **TeleMAX** costs $89 and the four-receptacle costs $129. Both also provide surge protection.

Panamax has over a dozen models to choose from, ranging in price from $69 to $149. Although choosing a particular Panamax surge suppressor for your needs may seem overwhelming, it's well worth the effort. I know of several people who did *not* have power-line protection and got "burned," as did their equipment. They all use surge suppressors now.

Computer insurance.

Speaking of insurance, I've got computer insurance. Do you?

It's inexpensive, easily obtainable, and covers your computer hardware and software against a variety of problems. As far as I know, the only insurance agency offering microcomputer equipment insurance is Safeware, 2929 N. High Street, Columbus, OH 43202 (800-848-3469; Ohio residents call 614-262-0559).

I've had insurance with Safeware for four years. About $65 buys up to $5000 of coverage against fire, theft, explosion and other problems (excluding war, nuclear accidents and such), with a $50 deductible. They'll even cover you if you're robbed while carrying your equipment. The actual policy is underwritten by the

WHY SETTLE FOR 2ND PLACE WITH BASEBALL GAMES?

HIT A GRAND SLAM!

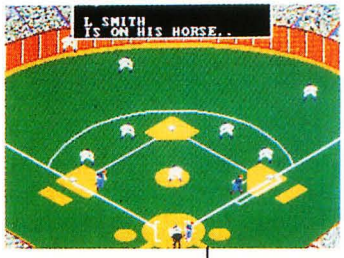
MICRO LEAGUE Baseball® II

The "Exclusive" strategy simulation of Major League Baseball.

By popular demand, in its *SECOND* great edition, MicroLeague Baseball is a winner again! And it's so good, it's the MVP (Most Valuable Program) of baseball simulation games.

New levels of baseball strategy. Now you can enjoy an all new level of baseball strategy. Because MicroLeague Baseball II features everything the award-winning first edition had, including: dynamic, high-quality graphics ■ real players and stats ■ true-to-life performance ■ powerhouse teams from the past and present ■ and exclusive endorsement by Major League Baseball and the Players Association. Whether you play a game or a season, if you're looking for managerial excitement, you can't lose.

Natural vs. Artificial Turf



We made it even better! MicroLeague Baseball II goes beyond the first edition, with phenomenal new visuals and sophisticated features such as: stadium dimensions, grass and artificial turf factors, box score and stat compiler *built in*, so you can keep track of your players and their season ■ expanded statistical factors if you're a baseball purist ■ injuries, arguments with the ump and even rain delays. Plus a *"Quik-Play Option"*, if you like to get to the bottom of the ninth inning *fast*.

Major League Excitement Year-Round!

MicroLeague Baseball is *not* a joystick game. Your *mind* — not your reflexes — are in control, and that's what keeps you challenged, game after game. The sights, sounds and stats create a level of involvement no other product can approach. Add to this MLB Season disks and the unique General Manager/Owner disk for drafting & trading players — even creating *your* own fantasy dream teams, and you've got a League Champ!

OFFICIAL LICENSEE MAJOR LEAGUE BASEBALL®

MLB

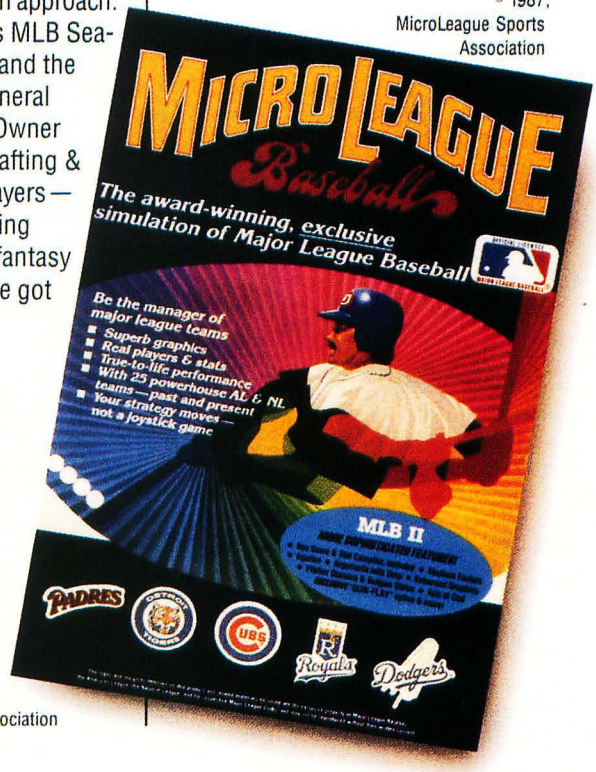
Official Licensee Major League Baseball, Major League Players Association

MicroLeague Baseball II is available for the Atari ST systems, and will be coming for the Amiga and MAC this summer.

Available at computer and software stores everywhere; or for direct VISA/MC orders simply call **1-800-PLAYBAL**.

MicroLeague I is still available for Apple II Series; Atari 65/130 XE, 800/XL; Commodore 64/128; and IBM PC, PC Jr. and compatibles.

MicroLeague Sports
2201 Drummond Plaza
Newark, DE 19711
(302) 368-9990



INDEX TO ADVERTISERS

This index is an additional service. While every effort is made to provide a complete and accurate listing, the publisher cannot be responsible for inadvertent errors.

ST user *continued*

Fireman's Fund Insurance Company of San Francisco, a top-rated company.

Regardless of how much or how little computer equipment you have, it should be insured. This is especially true since many homeowner policies do *not* cover computers. And, if you live in an apartment or other rental housing, your equipment is virtually uncovered.

Game-of-the-Month Dept.

For the past couple weeks, I've been playing a new game from Microdeal (MichTron, 576 S. Telegraph, Pontiac, MI 48053), called **Goldrunner**. It's a *fast* arcade-style game, similar to **Galaxian** except that your fighter isn't stationary. The entire playfield scrolls vertically, forward and backward, and you fire in the direction you're headed. Controls can be joystick, mouse or keyboard, and there's no need to select—all are active at once.

A particularly clever aspect of the game is that *you* control the scrolling speed—the longer you hold the joystick (or move the mouse) in one direction, the faster you move, and the more points you get for hitting enemy targets. I won't describe the scenario, since I personally enjoy the game for its main purpose: blasting the aliens. In fact, at the beginning, the title instructs you: "If it moves, shoot it. If it doesn't move, shoot it." That's as good a description as I could give.

If you enjoy shoot-'em-ups, you'll probably like **Goldrunner**. It retails for $24.95 and is available most anywhere.

They're here. . .

Several months ago, I mentioned a new ST hardware product from Practical Solutions, called **Monitor Master**. Like many startup companies, I wasn't sure if these guys would make it. Well, the company is still around, and so is their product.

Monitor Master is a little gray box that connects to the ST's monitor output jack, letting you hook up an RGB and a monochrome monitor at once. Using either is just a pushbutton away, rather than disconnecting and reconnecting cables. But wait; there's more.

Monitor Master also has audio and composite video output jacks. The audio RCA jack can be used to connect a cable to your stereo system, for better quality sound than that either monitor can produce. The video jack can link your ST to a composite video monitor or a VCR (your ST must have a built-in RF modulator to use the video output jack). If you're using a monochrome monitor, there's no need to disconnect the monitor to use your TV or composite monitor.

Monitor Master sells for $49.95, Practical Solutions, Inc., 1930 E. Grant Road., Tuscon, AZ 85719. The company also sells 13-pin DIN connectors and video cables for the ST. Contact them for details and pricing information.

Coming soon to an *ST-Log* near you.

Next month, we'll be looking at hard disk utility programs. In particular, disk-cache and backup programs from several software publishers will be evaluated and compared. Stay tuned. //

"Ordinary online services left me flat..."

$10.00+ per hour

"With GEnie™ I discovered vast new horizons."

$5 per hour

There's only so far you can go with most ordinary online information networks. But with GEnie—the General Electric Network for Information Exchange—there's virtually no end to the exciting new interests you can explore. Discover the value of GEnie's vast range of services and explore new territory with guides from our Special Interest Groups.

Join the friendly crew with GEnie's Atari† ***RoundTable™ Special Interest Group*** for exciting and informative discussions. Benefit from the expertise of experienced individuals. Compare notes on hardware, software, games or any other topic you choose. And, for the simple cost of downloading, collect thousands of valuable public domain software programs for your personal computer. Over 1000 new programs are added to our libraries every month. And remember, UPLOADS ARE FREE on GEnie!

No extra charge for 1200 Baud.

Make new friends, schedule meetings, gather facts and figures, set up travel plans, get the news, be entertained and even shop online for a fraction of what other information services charge, because GEnie's non-prime rate for 300 or 1200 baud is only $5 an hour. GEnie always comes out ahead of the competition in savings, in some cases by as much as 60%.

You get even more for your money.

In addition to user support groups, you can meet new people with ***GEnie's Live Wire™ CB Simulator.*** Exchange messages with ***electronic mail*** service. Find the best vacation deals and make your reservations with ***American Airlines travel service.*** Shop at home with ***American Express Merchandise Shop.*** Learn the fun way with a subscription to ***Grolier's electronic encyclopedia.*** Play classic and ***multiplayer games.*** Use GEnie's ***Financial Services*** to track stock market quotes, check market indicators and maintain an automatically updated personal portfolio online. All this ***plus*** new services added each and every month!

Map out your own voyage of discovery with GEnie as your guide...enrich your tomorrows by signing on with GEnie today.

Backed in the GE Tradition

GEnie is brought to you by General Electric, whose reputation is built on attention to detail and customer satisfaction. We're continuing that tradition with GEnie.

Compare & Save*	Services						Pricing			
	Travel & Shopping	SIGs/User Groups	CB & Mail	Financial Services	News	Games	Registration Fee	Monthly Minimum	Non-prime time rates 300 baud	Non-prime time rates 1200 baud
The Source	X	X	X	X	X	X	$49.95	$10.00	$8.40	$10.80
CompuServe	X	X	X	X	X	X	$39.95	none	$6.00	$12.50
GEnie†	**X**	**X**	**X**	**X**	**X**	**X**	**$18.00**	**none**	**$5.00**	**$5.00**

**Basic rates and services shown in effect 2/87. †Non-prime time applies Mon.–Fri., 6pm–8am local time, all day Sat., Sun., and nat'l holidays. Subject to service availability. Additional charges apply for 2400 baud and financial services.*

Sign up from your keyboard today.
4 Easy Steps:

1. Have your major credit card or checking account number ready.
2. Set your modem for local echo (half duplex)—300 or 1200 baud.
3. Dial **1-800-638-8369.** When connected, enter HHH
4. At the U# = prompt enter **XJM11877,GEnie** then RETURN.

Need help or more information? No modem yet? We can help. In U.S. or Canada call **1-800-638-9636.**

† *Atari is a registered trademark of Atari Corporation*

ATARI CORP. NOW ONLINE!

GEnie™

Stay online longer, for less.

GE INFORMATION SERVICES

General Electric Information Services Company, USA

GEnie rates and services subject to change. Uploads are free during non-prime hours at 300 or 1200 baud. Some services offered on Genie may include additional charges.

Kill ~~Put on~~ a Happy Face!

New from Xanth F/X! The game that lets up to 16 people play at the same time on 16 different STs. All networked together using MIDI.

Design your own mazes using First Word™.
Select up to 15 happy face opponents in single player mode.
Requires 1 MIDI cable for each computer in network.

Only $39.95 suggested retail.

For a free product catalog and newsletter call or write us at:

Hybrid Arts, Inc.™

11920 W. Olympic Blvd., Los Angeles, CA. 90064
Computer BBS: (213) 826-4288 **Main Office**: (213) 826-3777 **Telex**: 4931231 (HYBRDUI)

Hybrid Arts offers a full range of MIDI products for the Atari ST including:
EZ-Track™: A 20 track polyphonic MIDI recorder.
EZ-Score™: A sheet music printing program.
CZ-Android™: A patch librarian-editing system for the Casio CZ keyboard series.
DX-Android™: A patch librarian-editing system for the Yamaha DX-7 and TX series.
GenPatch ST™: A generic patch librarian system for all MIDI devices.
MIDITrack ST™: A professional MIDI recorder availlable with and without SMPTE sync.
ADAP SoundRack™: A 16 bit digital audio workstation for music and broadcast professionals.