

The #1 Magazine For Atari Computer Owners

ANALOG

**NOW
INCLUDES
ST LOG**

C O M P U T I N G

T.M.

NOVEMBER 1989
ISSUE 78

U.S.A. \$3.95
CANADA \$4.95

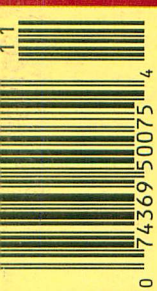


**ATARI ASSEMBLER/
EDITOR REFERENCE**

TYPE-IN SOFTWARE:
SURVEY SWEEPSTAKES
MEMORY MATCH
BASIC UTILITIES

REVIEWS:
MT-32 CONTROLLER
TALESPIN
BATMAN
AIRBALL

HOLIDAY GIFT GUIDE!





The magnitude of Calamus and its family of related products is difficult to describe in multiple pages of advertising. This one page is our attempt to give you a very general feeling of this powerful product line.

"How does Calamus stack up to the 'big guns' on other computers? Option for option, Calamus is comparable to, or surpasses, the desktop publishing competition."

"Without a doubt, Calamus is the most daring entry into the desktop publishing market in years. It's a powerful program that incorporates all the best features of existing publishing products into a fast, easy to use product."

Start Magazine May 1989

Calamus is a powerful, professional desktop publishing solution at an affordable price.

A partial list of features include;

- Outline font technology that produces identical results on both the printer and the screen, for 100% true "WYSIWYG"
- Fonts can be scaled from 1 to 999.9 points in 10th of a point increments
- Text can be rotated 360 degrees
- Printer output to the Atari SLM804 300 DPI laser printer is *very, very fast*
- Multiple clipboards throughout the document
- Complete MACRO capabilities
- A built-in Text Editor
- User definable view of up to 999%
- Multiple Master Pages (Style Sheets), Multiple Page Layouts and 300+ additional features

☆☆☆☆

A partial list of Compugraphic fonts for use with Calamus

Antique Olive

ITC Avant Garde Gothic Book

Bauer Bodoni

Brush

CG Century Schoolbook

Coronet

Dom Casual

English Times Extra Bold

Futura II Book

Garamond

Isabella

ITC Korinna Regular

Lucia Script

Modern Blackletter

Old English

CG Palacio

Park Avenue

Raphael

Revue Regular

Schneider Black

Signet Roundhand

ITC Souvenir Medium

CG Triumvirate

Typewriter Large Elite

Uncial

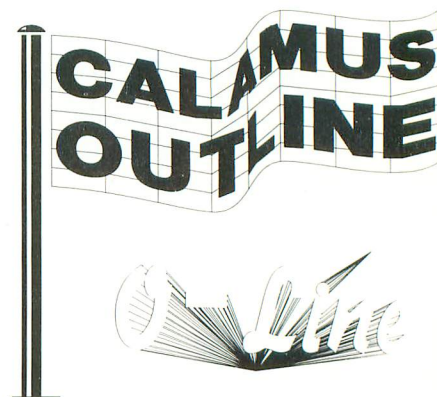
Univers Medium

Univers Extra Bold

Wedding Text

ITC Zapf Chancery Medium

Zapf Dingbats ☐ ☒ ☓ ☔ ☆ ✈



Calamus Outline Art

This vector graphic editing program offers you nearly unlimited possibilities for the manipulation of text, letter and font attributes. A few examples include: sending text along any path including waves, circular text, enlarge, compress, bend, stretch and distort any of our outline fonts. Generate smooth raster gradients in any direction, ray objects and project lines as rays. Project text onto a globe, cylinder or cone. Lines, Bezier curves and control paths are easily definable.

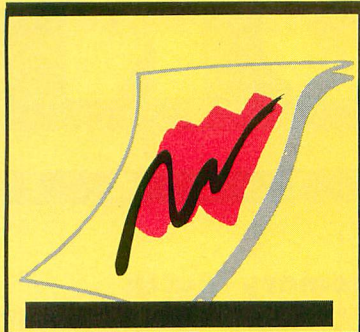


For more information please contact your local Computer Dealer or contact us directly

ISD Marketing, Inc.

Tel: (416) 479-1880 Fax: (416) 479-1882

Note: This page has been created entirely within Calamus and Calamus Outline and output at 1270 DPI on a Linotype L300 Imagesetter. Compugraphic® is the registered trademark of the Compugraphic Corporation. Calamus® is the registered trademark and tradename of DMC/Ditek International, represented exclusively in North America by ISD Marketing.



EDITORIAL

BY CLAYTON WALNUM

Well, here we are: the first issue of the new ANALOG Computing, your one-stop magazine for coverage of all Atari computers. We've worked hard to put together what we feel is the best Atari magazine value available anywhere, and we hope that, once you have had a chance to look through it, you'll agree.

Because we are now the largest Atari-specific magazine, we have plenty of room to stretch out. We've managed to cram into these pages all of the features and columns that both ANALOG Computing and ST-LOG readers have come to expect. Nowhere else can you get over 130 pages of reviews, programs, tutorials, columns and features designed to help you get the best from your Atari computer. No matter which Atari computer you own—an 8-bit or an ST—the new ANALOG Computing is one of your best investments.

Of course, we can't claim this issue is perfect. We've done the best we can, but we need your input in order to help us fine-tune ANALOG Computing so that it is exactly what you'd like it to be. So please, sit down and read through this issue; then, while it's fresh in your mind, jot us a quick note. Let us know what you like and what you don't like. If there's something new you'd like to see, tell us. Your opinions are important. This magazine is, after all, for you.

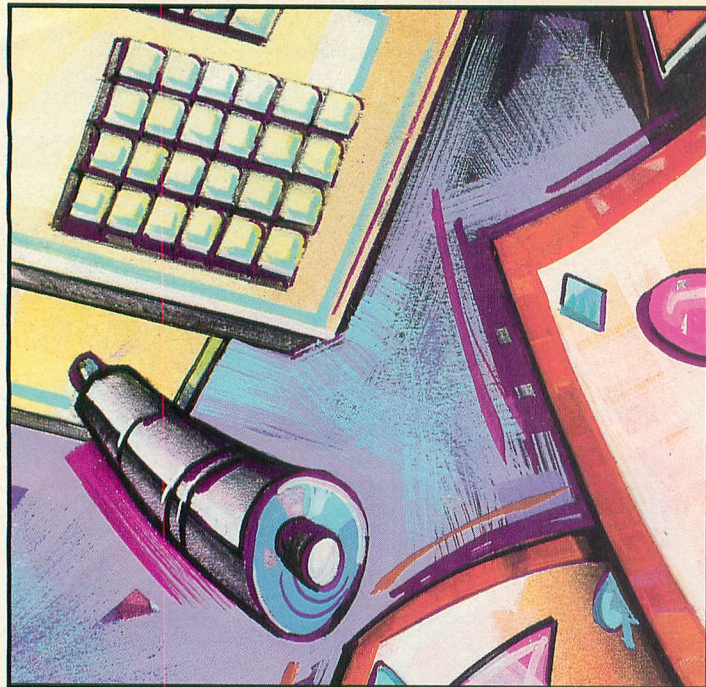
Those of you with subscriptions may be wondering exactly how the combining of the magazines will affect you. As you know by now (you are holding this magazine in your hands, after all), subscribers to either ST-LOG or the old ANALOG Computing will now get the new ANALOG. Those who subscribe to both magazines will have their subscriptions combined and extended. For example, if you have six months left on your subscription to ANALOG and nine months left on your subscription to ST-LOG, you will now get 15 months of the new ANALOG. In the months to come, we will continually improve ANALOG Computing, incorporating your suggestions and making sure all of you feel that you are getting the best possible magazine for your money. All we ask is that you write that letter. Write it today. Fill our mailbox to overflowing. We want to know what you think.

Responses to this editorial should be sent to ANALOG Computing, P.O. Box 1413-M.O., Manchester, CT 06040-1413.



CONT

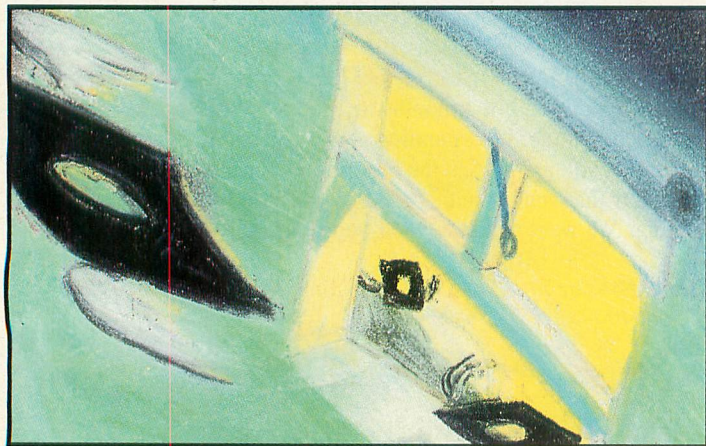
FEATURES



MEMORY MATCH 9



SURVEY SWEEPSTAKES 26



DIALOG BOXES IN GFA 119
SYMBOL GUIDE



This program runs in both color and mono.
It is available in type-in form & on the disk.



This program runs in color only.
It is available in type-in form & on the disk.



This program runs in mono only.
It is available in type-in form & on the disk.



This program runs in both color and mono.
It is available only on the disk.

(No Image) This article does not have an associated program.



This program runs in color only.
It is available only on the disk.



This program runs in mono only.
It is available only on the disk.

9



Memory Match

A sound-enhanced version of the ever popular concentration-type matching game.

by Joe Brzuszek

12



Title Maker

This utility will help you create attractive title screens that remain on your display as your main program is loading.

by Larry Black

15



BASIC Utility Package

BASIC programmers will enjoy this set of RAM-resident programming utilities.

by Barry Kolbe

17

Atari U.K. Show

Come with us on a short tour of a recent ST show in merry old England.

by Marshal M. Rosenthal

26



Survey Sweepstakes

Here's your chance to play a version of a popular TV game show on your Atari.

by Albert Baggetta

60

The Assembler/Editor Reference

For those who purchased the Atari Assembler/Editor cartridge without a manual, help is on its way.

by Matthew J.W. Ratcliff

65

The 1989 ANALOG Gift Guide

Need some gift ideas for that special computer person?

by Arthur Leyenberger

76

The Animation Stand: Special Effects

Maurice explains how to add some unusual effects to your computer animations.

by Maurice Molyneaux

114

Dialog Boxes with GFA BASIC 3.0

Part 1 of a complete tutorial on programming dialog boxes in this popular language.

by David Plotkin

REVIEWS

49 Changing Patterns
by Matthew J.W. Ratcliff

50 MT-32 Controller
by Michael Friesen

51 Airball
by Matthew J.W. Ratcliff

52 Talespin and STAC
by D.A. Brumleve

48 The ST Gameshelf
This month Ballistix, Batman, Captain Fizz, Twilight's Ransom and Speedball are reviewed
by Frank Eva and Steve Panak.

COLUMNS

7 C-Manship
by Clayton Walnum

14 The Personal Publisher
by Donavan Vicha

33 Ian's Quest
by Ian Chadwick

36 The End User
by Arthur Leyenberger

40 Database DELPHI
by Michael A. Banks

45 Boot Camp
by Tom Hudson

80 From Over the Big Water
by Marshal M. Rosenthal

107 The Compukid Connection
by D.A. Brumleve

112 BASIC Training
by Clayton Walnum

39 PD Parade
by George L. Smyth

118 Assembly Line
by Frank Cohen

121 The ST Macintosh
by Clayton Walnum

DEPARTMENTS

3 Editorial
by Clayton Walnum

6 Reader Comment

10 News Clips

109 Disk Contents

110 M/L Editor
by Clayton Walnum

128 Footnotes
by Karl E. Wieggers

ANALOG COMPUTING STAFF

Publisher
LEE H. PAPPAS
Executive Editor
CLAYTON WALNUM
Art Director
LEW BRYANT
Associate Editor
ANDY EDDY
Managing Editor
DEAN BRIERLY
East Coast Editor
ARTHUR LEYENBERGER
West Coast Editor
CHARLES F. JOHNSON
Contributing Editors
MICHAEL BANKS, FRANK COHEN,
MATTHEW J. W. RATCLIFF
Contributing Artist
CATHY KOSTREBA
Cover Photography
GARRY BROD
Illustrations
DALE CORY
STEPHEN STERLING
JONATHAN D. SMITH
CATHY KOSTREBA
Copy Chief
SARAH WEINBERG
Copy Editor
ALYSON GOULD
Editorial Assistants
PATRICIA KOURY
CATRINA MASON
ROBIN THOMPSON
Chief Typographer
DAVID BUCHANAN
Typographers
B. MIRO JR.
QUITA SAXON
LIGAYA RAFAEL
LARRY GANNON
Contributors
JOE D. BRZUSZEK
TOM HUDSON
BARRY KOLBE
BRYAN SCHAPPEL
BRAD TIMMINS
Vice President, Production
DONNA HAHNER
Advertising Production Director
JANICE ROSENBLUM
Advertising Production Coordinator
MAGGIE CHUN
National Advertising Director
JAY EISENBERG
(213) 467-2266
(For regional numbers, see right)
Subscriptions Director
IRENE GRADSTEIN

Analog Computing Published
By L.F.P., Inc.

President
JIM KOHLS
Vice President, Sales
JAMES GUSTAFSON
Vice President, Client Relations
VINCE DELMONTE
Corporate Director of Advertising
PAULA THORNTON
Corporate Editorial
TIM CONAWAY

Where to Write

All submissions should be sent to: **ANALOG Computing**, P.O. Box 1413-M.O., Manchester, CT 06040-1413. All other editorial material (letters, press release, etc.) should be sent to: Editor, **ANALOG Computing**, 9171 Wilshire Blvd., Suite 300, Beverly Hills, CA 90210.

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

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

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

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

Advertising Sales

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

Permissions

No portion of this magazine may be reproduced in any form without written permission from the publisher. Many programs are copyrighted and not public domain.

Due, however, to many requests from Atari club libraries and bulletin-board systems, our new policy allows club libraries or individually run BBSs to make certain programs from **ANALOG Computing** available during the month printed on that issue's cover. For example, software from the July issue can be made available July 1.

This does not apply to programs which specifically state that they are not public domain and, thus, are not for public distribution.

In addition, any programs used must state that they are taken from **ANALOG Computing Magazine**. For more information, contact **ANALOG Computing** at (213) 858-7100, ext. 163.

Subscriptions

ANALOG Computing, P.O. Box 16927, North Hollywood, CA 91615; (818) 760-8983. Payable in U.S. funds only. U.S.: \$28-one year, \$54-two years, \$76-three years. Foreign: Add \$10 per year. For disk subscriptions, see the cards at the back of this issue.

Authors

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

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

JE Publishers Representative

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

Los Angeles	— (213) 467-2266
San Francisco	— (415) 864-3252
Chicago	— (312) 445-2489
Denver	— (303) 595-4331
New York City	— (212) 724-7767

READER COMMENT

Still #1

This is a note to thank you for your excellent magazine. I have always been amazed by the high-quality utilities and applications, not to mention the arcade-quality games. What is most impressive is your continued support of the 8-bit Ataris. I've found other magazines losing quality lately, but yours has certainly kept up its standards. I'm sure all 8-bit Atari users thank you as I do.

—Oscar Fowler
Tucson, AZ

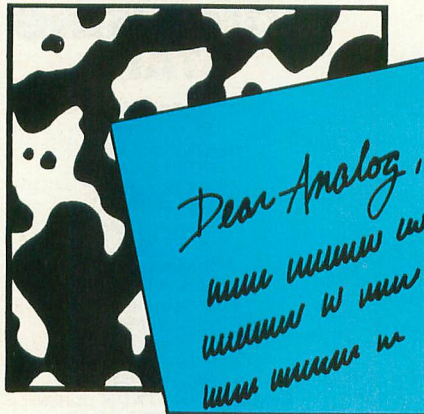
Reviews, Old Versions and Piracy

I want to give my support for Ian Chadwick's style and tone in reviewing software. I agree that Ian is often stern and critical, but I have used some of the software he has been critical of and I agree with him. At least when he says a piece of software is good, I know it is. When Ian is critical, he gives his reasons and suggests how the software could be improved. If I want to read gushing praise, I'll read the back of the box the program comes in.

Furthermore, I want to know why many software publishers think that beta-test copies and old versions of software are good enough for Atari users. There may not be as many of us, but our money is just as green.

I am also tired of having software piracy waved in my face by people like Gilman Louie. If the software publishers are so angry about the pirate BBSs, they should sue the SYSOPs. I do not pirate software, and I resent being grouped in with those who do simply by virtue of the brand of computer I own. The software publishers have every right to be angry, but they should not take their anger out on those of us who buy the software and keep them in business. They should not use threats to fight piracy; they should use the law.

—Travis Capener
Waterdown, Ontario



Show Me You Care

In "Show Me Your Wares" (July 1989 *Footnotes*), Karl Wieggers states that "shareware" is "the best 'ware' word because both syllables rhyme." Mr. Wieggers is apparently not familiar with the concept of careware. Four of my KidProgs are distributed as careware; I ask that satisfied users send a contribution to an ST-related charity (the Leal School ST Fund). I would propose that "careware" is a far better "ware" word than "shareware": Not only do both syllables rhyme, but both syllables also have the very same number of letters, and careware contributions are tax-deductible.

—D.A. Brumleve
Urbana, IL

A Correction . . .

ST-LOG's COMDEX Report (August '89, Issue 35) made mention of *PageStream PC* in a comparison of desktop-publishing systems for the IBM PC. Although we are certain the authors of *PageStream* for the ST would like to see an IBM version of their product, there is no such animal. The report intended to compare *PageMaker PC* to *Calamus*.

. . . And an Apology

It seems that in our World of Atari show report, which appeared in the September '89 issue of ST-LOG, we inadvertently left out mention of Migraph's attendance, and more importantly, the unveiling of their new hand scanner for the ST. We feel that the hand scanner is an important product and that it should have been included in the report. Our apologies to Migraph for this oversight.

ST would like to see an IBM version of their product, there is no such animal. The report intended to compare *PageMaker PC* to *Calamus*.

The PC Question

I thoroughly enjoyed the August '89 issue of ST-LOG and thought I'd make a few comments. I liked Todd Threadgill's *Footnotes* and would like to share my answers to Fred's questions. I sell Ataris in a mostly PC store and have been dealing with these questions for some time.

"Can it run Lotus 1-2-3?"

It runs a spreadsheet better than Lotus. (*LDW Power*, which is faster and has a greater capacity.)

"Can it run WordPerfect 5.0?"

It runs a superior version of *WordPerfect* that includes windows, drop-down menus and point-and-click operation.

"Can it run Microsoft Word?"

It runs Microsoft *Write*, which is the same thing. (I just won't mention that it's Version 1.0 instead of 5.0.)

What Fred was really asking was whether the ST performed PC *functions*, and the answer is a resounding "Yes, and better."

Finally, applause to Maurice Molyneaux for the best explanation of piracy I have ever read. From the comparative economies of scale to the moral impact on our society, he really hit it on the head.

—John Kolak
Carmichael, CA



ALL LETTERS TO BE
CONSIDERED FOR PUBLICATION
SHOULD BE ADDRESSED TO:
ANALOG COMPUTING READER COMMENTS
P.O. BOX 1413 M.O.
MANCHESTER, CT 06040-1413

C-manship:



A Complete GEM Application, Part 4

by Clayton Walnum

Believe it or not, there are many people who prefer to do things the old-fashioned way, people who despise such newfangled contraptions as menu bars and other mouse-driven devices that make them lift their chubby fingers from the keyboard. You and I, of course, are great fans of GEM, but whenever we design a program, we have to remember that not everyone shares our good taste. Put simply, programs should have keyboard alternatives whenever possible, especially for selecting functions from a menu bar.

Needless to say, **MicroCheck ST** provides keyboard selection for every function in the menu bar. The user who considers the ST's mouse a furless rodent unworthy of his touch may type a Control-key combination to select any function he desires. This month, we'll be looking at the portion of *MicroCheck ST*'s source code that handles the Control-key selections. We'll also look at the routines that allow the user to begin a new account.

Compiling

Listing 1 is this month's portion of the *MicroCheck ST* source code. Add it to the combined source code from the previous installments, then delete the *handle_keys()* and *do_newacct()* stubs from the resulting file.

After compiling the program, you'll find that you can now select functions from the menu bar by pressing a Control-key combination. For example, pressing Control-Q will exit the program.

Pressing Control-N, on the other hand, will activate the "New" selection of the File menu. A dialog box will appear, asking for your name, address and account balance, after which a second dialog box will ask for the filename you wish to use for the account. This filename can be up to six characters long. When you've typed in the filename, *MicroCheck ST* will save the information you typed in the new-account dialog box to a file with an .MCK extension and will create all the monthly data files for your new account.

Now let's take a closer look at how all this works.

Function *handle_keys()*

If you look at the selections in the menu bar, you'll see that each has a single letter next to it. This is the key to press along with Control in order to select that function from the keyboard. But just having the letters on the menu isn't enough, of course. We have to retrieve the key presses from the keyboard (which we do by watching for MU_KEYBD events with *evnt_multi()*), and when we get a Control-key combination, we have to route it to the right portion of the program.

This is handled by the function *handle_keys()*, which does much the same work as *do_menu()*, except that we're using Control-key values in the *switch* statements rather than a menu message. Another major difference is that we're using the *loaded*, *search* and *canceled* flags to determine which menu functions are active. We didn't have to do this in *do_menu()* because inactive menu selections are grayed-out and are not selectable by the user. Notice that in *handle_keys()*, we are highlighting the appropriate menu title with *menu_tnormal()*, just as we did in *do_menu()*. This tells the user which menu he is working with.

All of the values for the Control-key combinations in *handle_keys()* are defined at the top of the program. (That portion of the code was presented in the July '89 installment of *C-manship*.)

Function *do_newacct()*

Whenever the "New" selection of the File menu is selected or a Control-N is typed, the function *do_newacct()* is called. This function brings up the dialog box for entering the information needed to start a new *MicroCheck ST* account.

First, we call *clear_newacct()* to make sure any information that was previously entered into the dialog is erased. Next, we call up the dialog box in the usual way (see the *C-manship* in the May '87 ST-LOG) and activate it with a call to *form_do()*.

The variable *choice* will contain the number of the object used to exit the dialog. If this object is the OK button (NEWOK), we call the function *check_newacct()* to make sure all the information in the dialog is filled in. If the dialog has empty fields (*okay* is

FALSE), we redraw the dialog (the dialog is still on the screen, but the buttons haven't been redrawn to their deselected state) and loop back to *form_do()* in order to let the user try again.

If the user clicked the OK button and the dialog was properly filled in, we call *newacct_file()*, which will get a filename from the user for the new account and call the functions that will write the new-account information out to the disk.

If the user clicked the cancel button (NEWCANCL), we clear the dialog and close it, then go back to wait for another event.

Function *check_newacct()*

This function simply checks to see that none of the fields in the dialog has been left empty. We use the flag *okay* to communicate this information back to the calling function. First, we set *okay* to TRUE. Then we use a *for* loop to cycle through the editable text objects in the dialog, checking that none is empty. (An empty field will begin with a "@").

Note that should you ever use this method in any of your own programs, the objects you're checking must have been created sequentially, all at the same time. Otherwise, you can't use a *for* loop to check the objects. If an empty field is found, *okay* is set to FALSE and an alert box warns the user that he must complete the form.

Function *newacct_file()*

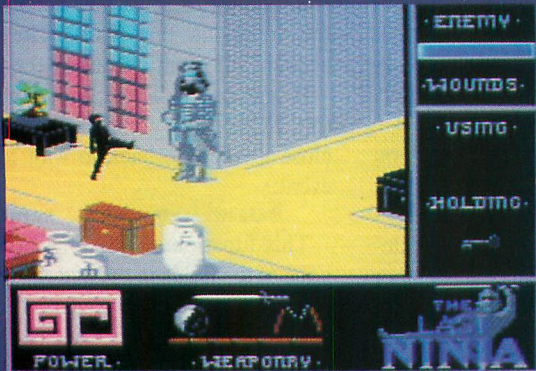
Most of this function is dedicated to getting a filename from the user and combining it with the right path specification.

First, we get the address of the string in the filename dialog box and clear it with a null. (With dialog boxes, you can clear a string with the "@" or by the usual null character.) Next, we zero out the string *filename* and call up the dialog box. If the user exits the dialog with the OK button (FILEOK), we call *check_file()* to be certain a filename has been entered. If it hasn't, we loop back to let the user try again.

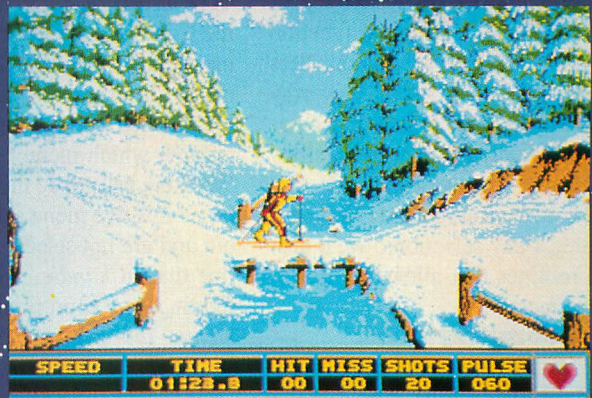
If the user entered the filename properly, we retrieve it from the dialog, copy it into *acct_name* (a string that will be used in the window's title bar), then add the complete pathname and the .MCK extension.

(CONTINUED ON PAGE 44)

Video Games & Computer Entertainment™



12
ALL COLOR ISSUES
ONLY \$19.95
Save over \$15 off the cover price!



- GAME REVIEWS
- ARCADE ACTION
- STRATEGY GUIDES
- TECHNICAL REPORTS
- COMPUTER SOFTWARE

VideoGames & Computer Entertainment™

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

☐ **Yes!**

Sign me up for 12 issues for only \$19.95—I'll save over \$15!

☐ Payment Enclosed — Charge My ☐ VISA ☐ MC

_____ EXP _____

NAME _____

ADDRESS _____

SIGNATURE _____

CITY _____ STATE _____ ZIP _____

Money back on unused subscriptions if not satisfied!
Foreign—add \$10 for postage.

Your first issue will arrive in 6 to 8 weeks.
WATCH FOR IT!!

Offer expires December 27, 1989 9.

CJXYA



Memory Match

BY JOE D. BRZUSZEK

MEMORY MATCH IS A COLORFUL COMPUTER ADAPTATION OF A SIMPLE CARD GAME THAT HELPS TO DEVELOP THE MEMORY SKILLS OF YOUNG PLAYERS. AS AN ENTERTAINING BONUS, THE PROGRAM PLAYS A UNIQUE SOUND EFFECT FOR EACH DIFFERENT PICTURE.

A DECK OF CARDS IS SHUFFLED AND LAID FACE DOWN ON A TABLE. PLAYER 1 TURNS TWO CARDS OVER TO REVEAL THEIR FACES. IF THE PICTURES ON THEIR FACES MATCH, PLAYER 1 REMOVES THE TWO OVERTURNED CARDS FROM THE TABLE. PLAYER 1 REPEATS HIS TURN UNTIL HE FAILS TO FIND A PAIR, AT WHICH TIME BOTH CARDS ARE TURNED BACK OVER AND THE TURN PASSES TO PLAYER 2. WHOEVER FINDS THE MOST MATCHES WINS THE GAME.

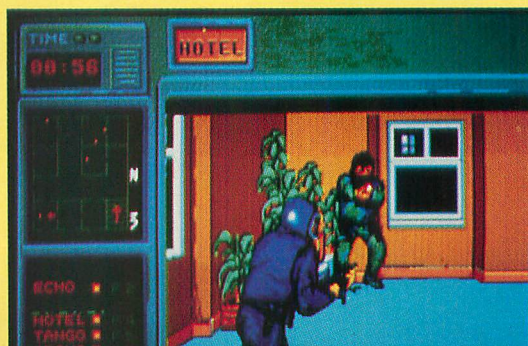
(CONTINUED ON PAGE 25)

NEWS

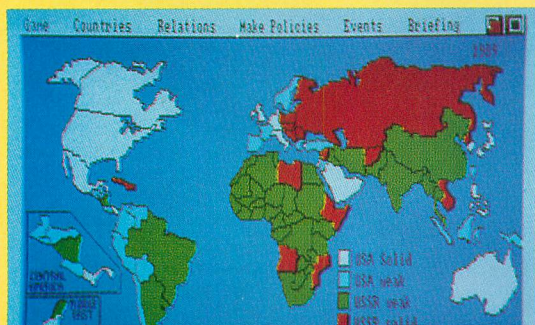
New Games From Mindscape



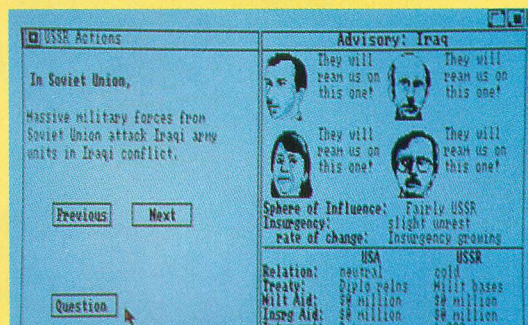
Déjà Vu II: Lost in Las Vegas



Hostage



Balance of Power: The 1990 Edition



Balance of Power: The 1990 Edition

Mindscape, one of the largest producers of games for the ST, has several new titles ready to go. *Déjà Vu II: Lost in Las Vegas* is the latest graphics adventure from the creators of *Uninvited*, *Shadowgate* and, of course, *Déjà Vu II*. In this sequel, the notorious mobster Tony Malone has threatened you with cement shoes if you don't come up with \$100,000. Lucky for you, Las Vegas is a great place to raise large sums of money... or is it? Like its predecessors, all commands in *Déjà Vu II* are entered by clicking the mouse. The game is priced at \$49.95.

In *Hostage*, your six-man strike team must

rescue hostages from a besieged embassy. This arcade-style game has you blasting out the embassy's windows, descending by rope from the roof and then searching for the hostages inside. Suggested retail is \$44.95.

Finally, Chris Crawford's award-winning simulation, *Balance of Power*, gets a revamping in *Balance of Power: The 1990 Edition*. In this version, you choose to become the president of the United States or the general secretary of the U.S.S.R. In either case, it's up to you to find the right balance between power and diplomacy in order to gain global prestige and avoid nuclear holocaust. Ac-

cording to Mindscape, "*Balance of Power: The 1990 Edition* includes an updated database covering 1989 to 1997, additional countries, an on-screen advisory council to help resolve diplomatic crises and a much-requested multipolar level, which allows even minor countries to make foreign-policy decisions." Retail price: \$49.95.

Mindscape, Inc.
3444 Dundee Road
Northbrook, IL 60062
(312) 480-7667

CIRCLE #117 ON READER SERVICE CARD.

Educator's Software

Educational Management Systems has announced a series of educator's programs for the Atari ST. *School Scheduler I* is a keyboard and mouse data-entry scheduling program used to create schedules by sections or by teacher. *School Scheduler II* uses Scantron forms that have been filled out by students to automatically set up the student's schedule. The Scantron reader is connected to the ST via the RS-232 port, allowing all student data to be entered into the computer almost instantly. Charles Gauthier, president of EMS, says, "Our goal is to get the ST into schools. We are working closely with established dealers, giving presentations to local school systems to demonstrate this power at a price at which there is no competition—one that Apple can't compete with because the programs aren't available, and one that IBM can't compete with because their system is not as fast or as easy to use as our software for the ST."

The programs are due for release in 1990. Prices are not currently available.

Educational Management Systems
P.O. Box 153
Huntington, MD 20639
(301) 535-0062

CIRCLE #120 ON READER SERVICE CARD.

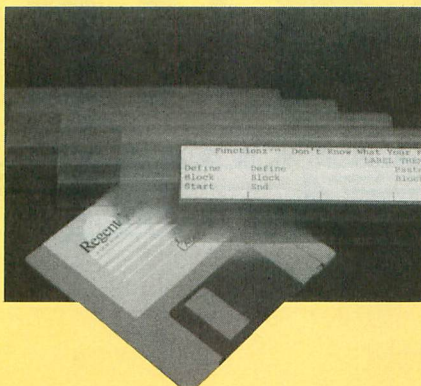
RAM-Disk Utilities

Helpways has just released *RAMPACK*, a "unified system" of 12 Atari 8-bit utilities. The *RAMPACK* loader automatically installs all 12 utilities into your RAM disk at boot-up time and then erases itself from memory. Any of the utilities may be called up almost instantly, including a RAM-resident help screen. Because each of the utilities can be loaded independently, they do not require a RAM disk and may be run on any Atari 8-bit computer. The utilities include, among others, MENU, which displays a disk's directory; PRNTFILE, which prints a LISTed file directly from disk; RAMMAP, which displays the current memory limits and free RAM; and SCROLLIT, which scrolls program lines both forward and backward. *RAMPACK* sells for \$19.95, plus \$2.50 shipping and handling.

Helpways
P.O. Box H
Rochester, NY 14623
(716) 334-2928

CIRCLE #122 ON READER SERVICE CARD.

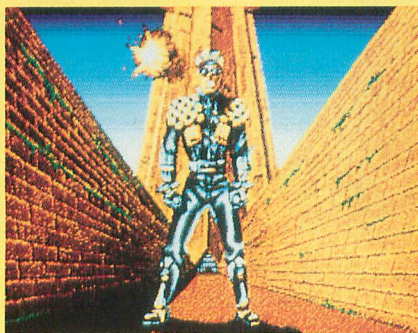
Function-Key Template



Regent Software is now shipping *Functionz*, a function-key template maker for the ST. Included in the package are six clear-plastic label stands and the label-maker program. The labels are printed onto ordinary paper and then inserted into the stands, which slide into the keyboard immediately above the function keys. Eight additional plastic label stands may be ordered for \$13.95 a set. *Functionz* is priced at \$24.95.

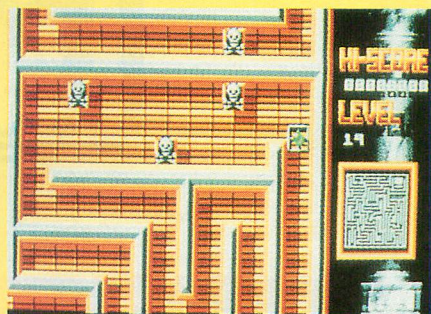
Regent Software
P.O. Box 14628
Long Beach, CA 90803-1208
(213) 439-9664
CIRCLE #119 ON READER SERVICE CARD.

Titus's Titan



New from Titus Software, producer of *Crazy Cars*, *Fire and Forget* and *Off Shore Warrior*, is *Titan*, a fast-scrolling strategic puzzle game. Professor Hybris, a resident of Vegapolis in the year 2114, has created a new

game consisting of 80 maze levels from which you must escape while avoiding the many death icons. All who have tried previously have died horrible deaths.



Anyone brave enough to take on *Titan* may do so for \$44.95.

Titus Software
20432 Corisco Street
Chatsworth, CA 91311
(818) 709-3693

CIRCLE #121 ON READER SERVICE CARD.

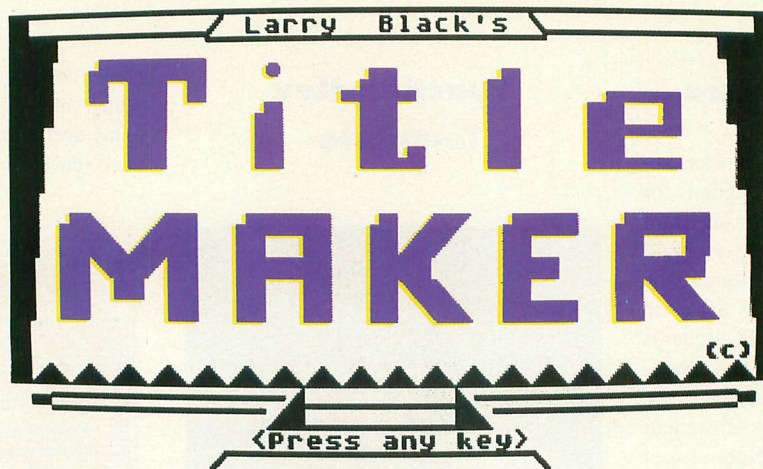
Print Shop Drivers

Owners of the 8-bit version of *Print Shop* from Broderbund Software might want to check out a series of new printer drivers from Innovative Concepts. Drivers are now available for the Atari 1020 plotter/printer, the Epson LQ-500/800 24-pin printers and the Okimate 10 color printer. The drivers may be used with both *Print Shop* and the *Print Shop Companion*. These programs do not modify your original disks. Each driver is priced at \$14.95, plus \$2.00 shipping and handling.

Innovative Concepts
31172 Shawn Drive
Warren, MI 48093
(313) 293-0730

CIRCLE #118 ON READER SERVICE CARD.

All product announcements to be considered for News Clips should be sent to News Clips, ANALOG Computing, P.O. Box 1413-M.O., Manchester, CT 06040-1413. Photos, screen shots and product samples should be included whenever possible.



Once the title has been loaded, the main program can load right over the top of it.

by Larry Black

Here is a utility to create a machine-language title screen that can be used with any binary-load program. When a binary program is appended to the title file, you can binary-load the resulting compound file. The GRAPHICS 0 title screen will display the characters you entered with whatever colors you selected while the main program continues to load. This method of loading a title screen does not require any additional mem-

ory in the main program. Once the title has been loaded, the main program can load right over the top of it.

Type in the DATA from Listing 1 using the *M/L Editor*, found elsewhere in this issue, saving the file to disk as TITLMAKR.EXE. *Title Maker* is a machine-language program written with the MAC/65 assembler, which works with just about every DOS configuration. When binary loaded, it executes auto-

matically (address \$3700).

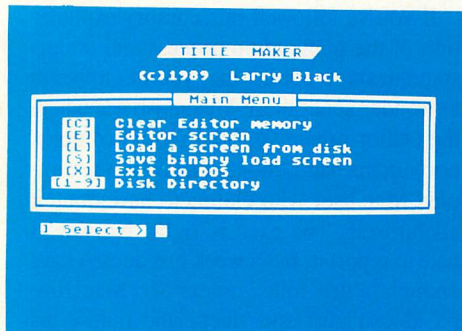
The main menu is simple to understand: [C] Clear editor memory—Erases the title screen from memory and puts you in the editor. Be sure you don't hit this before first saving your work. There is no warning prompt.

[E] Editor screen—Switches from the main menu to the editor so you can continue editing the title screen. Select will return you from the editor to main menu again.

[L] Load a screen from disk—Normally used to load a previously saved title screen so you can edit it further. It will also load the title from a program that has already been appended to the title file. However, do not save a screen with the same name as a program file, or it will erase the original program. Keep a backup copy of any program you append to a title screen. If the file you load is not in the proper binary format used by *Title Maker*, it assumes you are loading an ASCII text file and will automatically load the first 24 lines, converting the ASCII to screen format. Your ASCII text file should have an EOL (RETURN character) at the end of each line. Each line must be no longer than 40 bytes, including the EOL. If the file is more than 24 lines, you can expect a load of the first 24 lines only. The ASCII load routine sets the color registers to default (black and white). You can then set your desired screen colors.

[S] Save a screen to disk—Saves the title screen to disk in a binary-load format. Actually, it's a small machine-language program along with the screen data that will binary load and display the title on your screen. It can be reloaded into the editor later, using the "L" function, to either change it or finish an incomplete screen. The saved file will also contain the screen colors you have selected. In order to keep the screen in view after the load has finished, a graphics call must either be delayed by your main program, or a simple cursor position with a prompt and a get-key routine should be used. Of course, if your program is long, this method probably would not be necessary.

[I-9] Disk directory—Type the number of the desired drive, and the filenames will be displayed. Press any key, and you return to main menu.



THE EDITOR SCREEN

The editor uses 40 columns and all the standard Atari editor keys to display 254 of the 256 ATASCII characters, just like the BASIC editor. The return key simply moves the cursor to the left column, as usual, but no

character is displayed. If the cursor is on the bottom line, it will return to the top-left corner. The screen will not scroll. The clear key works the opposite from the BASIC editor. Pressing Shift-Clear or Control-Clear will only display the ATASCII "arrow" character, and will not clear the screen. This is a precaution in case you accidentally hit it while trying for a nearby key. To actually clear the screen while still in the editor mode, just press Escape first, then Shift-Clear. This is helpful, should you want to start over and retain the same screen colors. If you want to clear the screen with default colors, use the "C" option from the main menu.

Using the GRAPHICS 0 Atari character set, create the desired title screen. You will notice that when you print a character in the bottom-right corner of the screen, the character is printed and the cursor jumps to the top-left of the screen, rather than scrolling the entire screen. You can save your work from time to time by pressing the start key from the editor screen or with the "S" option from the main menu.

THE HELP MENU

Control-arrow key: responds for cursor placement, just like the BASIC editor.

Shift-Control/up-down arrows: changes the screen color register. You can cycle back and forth through all 128 colors.

Shift-Control < or >: changes the screen border colors.

Shift-Control-O or -P: changes the text luminance. If you should happen to find a blank screen after loading a file, try changing the text luminance to read the screen. It might have been set to the same value as the screen color, hiding the text from view.

Option: displays the help menu. Press any key to return to editor.

Select: returns to the main menu from editor.

Start: returns to the main menu and sets up for a binary save of the screen.

HOW THE TITLE SCREEN WORKS

Using DOS, copy with /Append the binary program to the title file. For example, let's say your title file is TITLE.EXE, and the binary program is DATABASE.EXE. With a backup copy of each on the same disk (Drive 1), and using Atari DOS, type the following:

```
C [RETURN]
DATABASE.EXE,TITLE.EXE/A [RETURN]
```

The DATABASE.EXE program will be appended to the TITLE.EXE program. Then delete the backup copy of DATABASE.EXE


and rename the TITLE.EXE to DATABASE.EXE. The new program will be about eight single-density sectors longer than the original.

When you binary-load the new compound program, the title screen will load and display, and the main program will automatically load and execute. For best display purposes, your main program should display a "Press Any Key" type of prompt on the screen before clearing the title. Then it should look for a keypress before making a graphics call. A good example of this is the *Title Maker* program itself. Watch the screen as it loads. You can even load the title screen from the program itself, as a sample.

BASIC programmers who would like to use the *Title Maker*, don't fret. There is a way you too can use this utility for BASIC AUTORUN-type files. Listing 2 is a small but efficient means for you to do so. To access this method, type in the TITLMAKR.BAS program (Listing 2), using *BASIC Editor II* to verify your work. Once you have done this, save it to disk for backup use. Now run the BASIC program. What it does is create a binary file (AUTOBOOT.TM) that you can append to any of your *Title Maker* screen files to be used with a saved BASIC program. Once you do this and have appended the AUTOBOOT.TM file, rename it to AUTORUN.SYS and rename your BASIC program to AUTORUN.BAS. Next time you reboot that disk, it will load your title screen and automatically run the AUTORUN.BAS program.

For an added touch of show, here is a neat trick you might want to try. Make several screen files and append them all together. When you binary-load a compound screen file, each screen will be displayed, one at a time, in the order they were appended to the main file.

CONCLUSION

Using *Title Maker*, you can make your programs look as professional as the commercial ones. Try it. I know you'll be pleased with the results. 

LISTINGS
CONTINUED ON PAGE 86



Larry Black got his first computer, an Atari 400, six years ago and now hopes for a career in computers. *Title Maker* is his first published program.

The Personal Publisher:



P A G E S T R E A M G E T S S E R I O U S

by **Donavan Vicha**

Last month I described a book project I put together using the Migraph "Art Department" (*Easy Draw* and *Touch Up*). This month I will continue the story, concentrating on SoftLogik's *PageStream*. But first, let's look at what's involved in the editorial production of a book.

Once an author submits his manuscript to a publisher and it's accepted, it's no longer his or her labor alone that puts the finished book on the shelves of a B. Dalton or Waldenbooks store. The manuscript goes from the acquisitions editor to a project editor (there are other titles used: content editor, associate editor), whose responsibilities include fact checking, evaluation of contents for consistency and comprehensiveness (as well as comprehension!) and ascertaining whether all illustrative material (photos, charts, graphs, line art, etc.) is included and properly keyed to its place in the manuscript.

Once this editor has completed his or her work, the manuscript is copyedited. The copy editor ensures that the manuscript is grammatically correct, that captions for illustrations are correct and appropriate, and that the author's style remains consistent. The copyedited manuscript then goes for design, either by a book designer or the production editor.

The production editor (PE) is given the responsibility to oversee the manuscript through the rest of the publication process. Usually, the first-pass galleys (the typeset manuscript) are read by a proofreader and the PE to catch typos and dropped words/lines/paragraphs/etc., and to see that all the type specifications have been correctly carried out. The author also sees this set of galleys to make any changes, which the PE is obliged to transfer to the proofread set that goes back to the typesetter. Second-pass galleys are also proofed, but another set is given to a layout artist who creates a dummy of the book. (As a production editor myself, I have often been the layout artist as well.)

The dummy shows, page by page, where text and graphics go, usually in two-page spreads. All elements of the book are accounted for here: running heads (or footers), part and chapter openers, front matter (copyright, title pages, contents, dedications, epigraphs and so forth), back matter (glossaries, appendices, index), and the body of the book (text and graphics). The PE goes over the dummy to ensure that it has been handled correctly, as does the acquiring editor or the project editor. The PE

You might begin to wonder about those Macintosh ads where the senior executive, holding the complex book in his hand, looks askance at the dimpled Mac user who claims to have done it all himself.

also keys patches to the dummy pages—patches being the corrected portions of revised (second-pass) galleys caught by the second go-round of proofreading. (Instead of rerunning a third complete set of galleys, corrections are made only of the offending line or lines and then pasted over the incorrect passage.)

Once the dummy has been approved, the layout goes to a keyline artist who does the actual paste-up of the camera-ready type and other elements, if any, on "boards." These boards are generally of a lighter weight than run-of-the-mill cardboard and are preprinted with blue lines, which are invisible to the camera used by printers, to create the plates used to print the pages of the book. These blue lines indicate where text should go and where running heads and folios should go, making the keyliner's task

much simpler. In some cases, these boards are printed with black ink for elements that are supposed to be captured on the (negative) plate and printed.

The printer usually provides the publisher with a trial printing of the book, usually called a "blueline" due to the blue ink used in its creation. Depending on the kind of press used, these pages are grouped in "signatures" of 12, 16 or 32 pages of very cheap, slick paper, which I've always found hard to write on except with a grease pencil. The PE must check every page for instances of broken type or blobs of ink, note that photos and graphics are cropped and positioned correctly and watch for irregularities in print quality of text and graphics. Oh, yes, they must also verify pagination (make sure the pages are in the correct order).

As you can see, a complicated process and a team effort lies behind the production of any book you purchase. A similar process occurs for most publications, but because newspapers and magazines generally follow the same format, the process is simplified. The deadlines for newspapers and magazines are stricter and shorter, necessitating the simplified process, but certainly not making the job for the production editor any easier.

If you've followed this cursory explanation of the publication process and understand desktop publishing, you might begin to wonder about those Macintosh ads where the senior executive, holding the complex book in his hand, looks askance at the dimpled Mac user who claims to have done it all himself. The exec is right: No way! I hate to report it, but a week just doesn't have enough hours, folks, unless Mr. MacDimple doesn't eat and sleep, and unless his Mac is equipped with a time machine.

It's unlikely a laser printer—especially the much-vaunted Lino 300—is going to give you several hundred text pages (let alone text *and* graphics) in less than two or three days. That leaves four days for writ-

(CONTINUED ON PAGE 22)

THE BASIC UTILITIES PACKAGE

by Barry Kolbe

With the *BASIC Utilities Package*, you can program in BASIC and still access the disk functions like: directory, locking files, deleting files. Normally, to do these functions you would have to save your program, type DOS and wait until DUP.SYS loaded in. Then it was back to BASIC, reload your program, etc. With BUP.SYS you merely type "DIR" for a disk directory. That's all there is to it. Your program remains intact. And that's not all. You also get automatic line numbering and renumbering, not to mention decimal and hex conversions at your fingertips. And there's more! List the variables in your program and their line numbers, and trace your program while it is running.

Typing it in

BUP.SYS is a machine-language program and must be typed in using the M/L Editor, found elsewhere in this issue. Listing 1 is the data that will create your copy of the *BASIC Utilities Package*. You should type it using the M/L Editor, found elsewhere in this issue, and name the resulting file AUTORUN.SYS.

Loading BUP.SYS

The *BASIC Utilities Package* will load automatically when you boot your computer with a disk containing DOS and the AUTORUN.SYS file you created from Listing 1. The message "BASIC Utilities Package" will appear, followed by the BASIC prompt "READY." BUP.SYS remains in memory until you type DOS or turn the computer off.

BUP.SYS Commands

BUP.SYS supports 13 commands, shown in format form below:

```
DIR
PRO "D:\filespec.ext
UNP "D:\filespec.ext
NAM "D:\filespec1.ext, filespec2.ext
ERA "D:\filespec.ext
```

```
REN [nnnn,nnnn]
NUM [nnnn,nnnn]
TRA
OFF
DEC nnnn
HEX [$]nnnn
LVA
DOS
```

All commands are three letters in length, with no spaces allowed in the first three letters. After the first three letters, spaces are removed. For example, HEX \$ A0 00 is crunched down to HEX\$A000.

DIR gives a listing of the disk directory.

PRO, UNP, NAM and ERA are file-management commands. They lock (protect), unlock (unprotect), rename and erase files on disk. Think carefully before using ERA—you are not asked if you are sure! Once you hit Return, your file is erased.

REN is the renumber command. The first number following REN is the new starting line number and the second is the increment between lines. For example, REN 100,10 makes the first program Line 100, the next 110, the next 120 and so on. More on REN later. Typing REN [Return] gives default values of 10 and 10.

NUM is the auto line-numbering command. As in REN, the first number is the first line number and the second is the increment between lines. The default values are 10 and 10 if you type NUM [Return]. If a program is in memory, NUM [Return] will start numbering at the last line (plus the increment, of course). To stop auto line-numbering, hit the break key. Do *not* press Reset! Disastrous things may happen if you do.

NUM will not allow you to type over an already existing line. For example, if Line 100 exists and you type NUM 100,10, no auto line-numbering will be done. No message appears, just the READY prompt. Even while NUM is in operation, it will quit if you are about to type over an existing line.

TRA allows you to trace a BASIC program. When you type TRA [Return] an ex-

tra display line is set up above the normal screen. The message "EXECUTING LINE: 032768" is displayed on this line. When you run a BASIC program, this message is updated 60 times a second during the vertical blank. Consequently, it may be a blur at times. Infinite loops are easy to spot. Trace works in any graphics mode. It will stay in effect until you remove it by typing OFF [Return]. Trace uses the immediate vertical blank interrupt, so don't use Trace if your program also uses that interrupt vector.

OFF turns off the Trace function and removes the extra display line. I would recommend turning it off before performing disk commands. Whenever BUP.SYS uses the disk drive, it automatically turns TRACE off.

DEC converts any decimal number from 0 to 65535 to its hex equivalent. If you type in anything other than a whole number in this range, the result should not be trusted.

HEX converts any four-digit hex number to its decimal equivalent. A "\$" is optional. For example: HEXA000 and HEX\$A000 are both acceptable.

LVA lists the variables in your BASIC program and the line numbers in which they appear.

Now that we've covered the basic commands, we must turn our attention back to renumber.

REN, in addition to renumbering the program lines, renumbers internal line references. There are nine program statements that can use line numbers. They and their BASIC tokens are:

Statement	Token
LIST	4
GOTO	10
GO TO	11
GOSUB	12
TRAP	13
RESTORE	35
IF(THEN)	7(. . .27)
ON(GOTO)	30(. . .23)
ON(GOSUB)	30(. . .24)

De Re Atari has a nice explanation of tokenized BASIC. ANALOG issues 25 and 26 also have articles on this subject. Each statement can be followed by a line number(s), variable(s) or algebraic expression(s). For example:

```
13 GOTO 1050
23 LIST 1040,1100
33 GOSUB M4
43 RESTORE 240+Q
53 TRAP 15*P
63 ON Z GOSUB 20,M,25,6*T,30
73 ON T GOTO 1000,1100,1200,1300
```


The numbers or expressions following the statements are internal line references; REN cannot decipher all of these. REN will only renumber an internal line reference if it is an actual number. Only in examples 1, 2 and 7 would all internal lines be renumbered. The others would cause a message to be sent to the screen.

The three conditions and their messages are shown in Table 1.

CONDITION	MESSAGE
1. A renumbered line would be ≥ 32768 . No renumbering done.	Line ≥ 32768 .
2. An internal line number cannot be found. It is not renumbered.	NF-100
3. An internal line reference is a variable or contains a variable.	VR-540

Table 1

To remedy the first condition, choose a smaller first line number and/or a smaller increment. The second condition means that you are missing a program line. For example, Line 100 might be "100 GOTO 2000" when Line 2000 doesn't exist.

The last type is the most frustrating to deal with. Examples 3, 4, 5 and 6 from above would result in a VR-line number message. Whenever REN encounters one of those special nine tokens, it checks whether it is followed by a line number. If it is a line number, that line number is renumbered. If not, a VR message is given.

If a list of line numbers, separated by commas, is possible (types 2, 6 and 7), these are renumbered until a variable is encountered. If a variable is found, all renumbering on that statement is halted and the next statement on the line or the next line is checked for one of the nine tokens. Again, the VR message would be sent to the screen. In example 6, the 20 would be renumbered, but no others, not even the 25 or 30.

A TRAP statement with a line greater than 32767 would also give an NF message, but this type of TRAP is used to shut off a previous TRAP and so doesn't need to be renumbered.

If LIST is not followed by a line number, it is listed as a VR. This doesn't represent a problem.

Except for the exceptions just noted, all lines printed on the screen with either NF or VR need to be examined and any necessary changes made. Save a copy of the un-

renumbered program in case you need to check line numbers.

The message "Program renumbered" is displayed on the screen when the process is complete. It should not take more than a few seconds.

Anomalies

Some unusual things have happened to me while using BUP.SYS. An IF-THEN statement, such as "20 IF I\$='R' THEN POP:GOTO 240" resulted in a VR-20 message when I renumbered it. Even though Line 240 existed, the message VR (NF) was sent to the screen. I'll try to explain why. In tokenized BASIC, following the token for THEN is a number (zero-255), which gives the offset to the next statement on the line. REN checks the token after the THEN token to see whether it is a line number. The token for a number is 14. Guess what the offset to the next statement is in Line 20. Yes, it is 14! The same as the token for a number. Since POP is not a number, the VR message was printed.

Originally, I used UNL as the command to UNLOCK files. I changed it to UNP so it is more like DOS XL. One Saturday my son, Philip, was trying out BUP.SYS. Needless to say, he was impressed! Then Phil decided to work with Clayton Walnum's programs from "Adventurous Programming." He got to the point where he was to "UNLOCK DOOR." The screen looked like this:

```
WHAT NOW? UNLOCK DOOR
ERROR # 146
READY
WHAT NOW?
```

Remember that BUP.SYS parses (scans) all inputs before passing them on to BASIC. BUP.SYS thought he was trying to UNLOCK a file! We changed the verb to KEY to make the program work with BUP.SYS. It was also interesting to watch the program work with the TRACE function.

Technical Notes

BUP.SYS is a BASIC Utilities Package, so the BASIC cartridge must be present to use it. BUP.SYS loads at \$2000 and moves LOMEM to \$2AFB, using 2,811 bytes of memory. After loading, BUP.SYS clears the stack and jumps to the BASIC cartridge. If BASIC is not there, there will certainly be a lockup. In the initialization process, it wedges itself into the editor's Get-Byte routine. By doing this, an immediate-mode command is scanned by BUP.SYS before sending it to BASIC. In this way, BUP.SYS

can take control if the command is one it recognizes. The Put-Byte routine is used to print messages on the screen.

BUP.SYS works fine with DOS 2.0 and 2.5, but has serious problems with DOS XL and perhaps others. Two vertical blanks are used. One is for the TRACE command. The other is used by auto line-numbering. BUP.SYS is "protected" from Reset to prevent its erasure. But if I were you, I would not press Reset while using auto numbering, renumber or TRACE. I shudder to think of a program that is partially renumbered. Any other time, pressing Reset should present no difficulty. If you do get stuck, typing X=USR(8192) will probably fix things by reinitializing BUP.SYS.

Conclusion

I think you'll agree that a program like BUP.SYS has been needed for some time. I hope it will make your programming easier. **A**



Barry Kolbe is part of the BBK programming team, which is responsible for such ANALOG programs as The BBK Artist, The BBK Monitor and The ANALOG Database. He is a teacher in Wisconsin.

LISTINGS (CONTINUED ON PAGE 102)

Public Domain Software

#57 - Tease Me Adult Animation (Color Only)
#393/394/533 - PrintMaster Graphics
#400 - 7 Disk Labeling Programs (w/100 Labels \$6.95)
#443 - Intersect RAM Baby (RAM Disk/Print Spooler)
#475 - Werty's House of Horror (Adult Game, Color)
#500/600 - Publishing Partner Fonts
#511 - Dungeon Master Maps for Levels 1-7
#512 - Dungeon Master Hints/Character
#555 - The Assistant Chef - Electronic Cookbook
#557 - Children's Programs (Color Only)
#567/728 - Accessories
#575 - Sheet V2.0 - Shareware Spreadsheet
#588 - Pac Man, Hangman and 5 others (Color Only)
#599 - PageStream Fonts, Font Converter
#655 - ST Writer V3.0 w/Spell V2.8
#690 - Opus V2.10 GEM Spreadsheet (1 Meg/DBL)

Introductory Offer - Above Disks Just
\$2.99 Each

ST Xformer Cable \$19.95

Run 8 Bit programs on your ST! Use 5 1/4" or 3 1/2" Disks. Loads & Runs protected disks. Receive the latest version of ST Xformer **FREE** with purchase of cable when you mention this ad.

Call or Write for **FREE** Catalog
(800) 347-6760

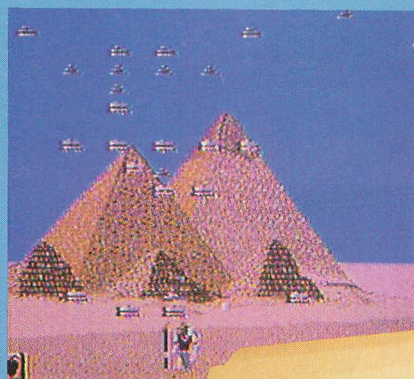
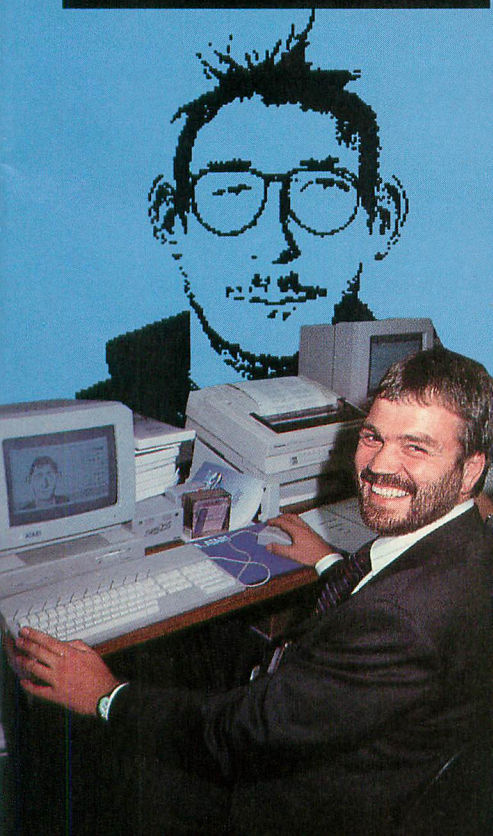
BRE Software Dept. STL
352 W. Bedford, Suite 104
Fresno, CA 93711
Customer Service (209) 432-3072
Shipping \$2.50 Ground / \$4.00 2nd Day Air



CIRCLE #110 ON READER SERVICE CARD.

ATARI UK SHOW

by Marshal M. Rosenthal



Europe is known for culture, and since we all love culture, what could be better than going to England to explore many centuries' worth of history and art? I can think of only one thing: an Atari ST show just outside of London, in the Alexandria Palace Conventional Hall.

The first thing I noticed was the crowd waiting to get in, something we're not used to seeing in the States (the ST is doing well overseas). Inside the show there were many young people—some as young as ten years old—dragging their “mums” from booth to booth. Many adults were in attendance too. And, of course, vendors everywhere were selling software and hardware.

Atari Innovation Award

Immediately apparent was a complex just inside the entrance where a number of exhibitors were showing their entries in the Atari Innovation Award Contest. The purpose of the contest was to show how an ST can be used in new and innovative ways. One exhibitor in particular caught my eye: David Jones and his Comfortable Wheel de-

sign. This award-winning invention (it took second place) is a resilient wheel with a built-in suspension system. The wheel is ideal for wheelchairs, as it “pushes” up and over obstructions. Jones bypassed the necessity of producing handmade models of the wheel by using his ST. Because the design program is written using FastBasic, changes in the design can be easily implemented, with the computer drawing and animating the result.

Hardware

Another product that attracted my attention was the Hawk Colibri hand-held scanner, which will read graphics as large as half a standard page and convert them into 32 gray levels. The unit plugs into the ST's cartridge port and can adjust the level of scanning to 100, 200, 300 and 400 dots per inch (dpi). The scanner includes sophisticated software with zooming capabilities, and a Tools and Options menu. Images are saved in *DEGAS*, *IMG* and *GEM* formats, with many laser and dot-matrix printers supported. Optical Character Recognition

(OCR) can be purchased in ROM, allowing the unit to translate text into ASCII characters at a speed of about 50 characters a second.

When it comes to a monitor, big is always better. Signa's Matscreen M110 is a fat, 19-inch paper-white monitor with a bright, non-interlaced screen that has a 1,280 × 960 resolution. The controller board fits into the processor bus of a Mega unit and will work unmodified, provided the application being used was written according to GEM rules (some examples of programs that work with the monitor are *Fleet Street Publisher*, *Timeworks Desktop Publisher*, *1st Word Plus v1.12* and *Gem Draw*). This monitor is expensive, however (£1,750, nearly \$3,000.00), but seeing is believing.

How often have you heard the old saw, "Everybody talks about the weather, but nobody ever does anything about it"? Well, here's your chance to get a jump on the weather. *Meteosat* (Signa Publishing) turns your ST into a weather receiving station that picks up images from the Meteosat and GOES satellites. The images are initially processed in West Germany, where land-mass outlines are added, along with the Whole Earth image being split into 11 distinct subimages. These are then retransmitted to satellite and can then be picked up by your ST, which can save and animate 11 images at a time.

Not exactly your typical weekend project, the entire system can be up and running in a few hours. Hardware consists of a satellite dish, pre-amp, receiver and demodulator. The software enables infrared, visible light and cloud-vapor images of Europe and the U.S. Any ST will do the trick, provided there's a double-sided drive and color monitor.

Part of the fun of these shows is trying to find the hot products "buried" amidst the more ordinary displays. I was attracted to one of these "gems" by a small television superimposing titles over scenes from *Miami Vice*. I could see a small black box, slightly larger than a hardcover book, attached to an ST and a VCR under the table. I managed to get director Jeremy Rihl of Digita to pull it out for a closer look.

"Our Multigen genlock is designed to work with either PAL or NTSC [U.S. standard] systems automatically," he says. "You can also see that it's been designed to run off a nine-volt battery rather than AC." Using custom chips designed in the States, the genlock attaches to the ST's RGB port, with a pass-through for the monitor. Multigen uses standard BNC plugs and outputs to RGB or RF. Digita expects to retail it in the

£500-£600 range (around 1,000 U.S. dollars) and is discussing a more professional unit that will take advantage of the 400 lines of resolution possible with the Super VHS system.

In another corner could be found more monitors, this time displaying images of the crowd. These images were manipulated by VIDI ST, a real-time digitizer that can grab an image from a moving video in 1/50th of a second in a full 16 shades.

VIDI ST's software includes many power-



ful tools, such as full palette control over each frame and eight levels of controllable brightness. Frame registration is accessible through an interrupt feature (interrupt live frames with existing frame), and the interface uses drop-down menus and single-key shortcuts. Cut and Paste is fully supported as well. The device plugs into the expansion port and works with both PAL and NTSC images. Frame sequences can be stored into memory faster than 12 per second, and animation can be displayed at up to 25 frames per second.

Now we come to one of those products that sound too good to be true. The ParSec graphic interface is a hardware device that plugs into the Atari ST without modification and turns it into a super graphics machine. Resolution with a non-interlaced monitor reaches up to 1,024 × 768 (low being 640 × 480). You want colors? Try handling 16 per line, 4,096 per screen, using four-color planes. Utilizing a 32-bit 50-MHz processor with memory capabilities up to 5.5 megabytes, ParSec is about the size of a 1040ST and has video, RS232 and SCSI (for connecting CD ROMs, video and hard-disk drives) connectors.

A full-featured art package is supplied, which supports pixel processing and conventional drawing/manipulation features as

expected. A professional artist package is being designed to include chalk, oil and water emulation, plus support for ray tracing and animation. (Designs are under way to upgrade this package further.) Most conventional programs can be modified using a GEM-to-ParSec converter program so that they can run in conjunction with the ParSec and realize the advantages of superior resolution and color palette, plus high-speed operation. Add-ons will include a genlock/digitizer for external mixing and overlaying of video signals and the input and manipulation of stored images. All of the above can be had for a retail price of approximately \$1,550.00.

Frontier Software is known mainly as an importer of Supra products, but they've got a new line of their own hardware as well. Their disk duplicator is just the thing for the small software house, as it enables duplication of up to 32 disks per minute. The interface board plugs into the disk drive of the ST and has ports for drives in both 3½-inch and 5¼-inch formats. Frontier's Forget-Me-Clock II plugs into the cartridge port and has a pass-through for additional uses.

Games

Before I had a chance to look at the many game booths, Mungo Leir of Software Horizon grabbed me by the shoulder, pulled me over to a corner (what happened to the restrained British character?) and started pulling out boxes containing Software Horizon's new budget line of software.

"Games have a definite life," Leir says. "They must keep in step with the expectations of the player and give value based on their price. We have new titles that are both original and inexpensive."

First up is *Twilyte*, a 3-D fighting simulation that owes much to the *Star Wars* coin-op. *Triton 3* is a vertically scrolling shoot-'em-up with a space theme, while *Stomp* (great title) turns the ST into a drum machine with 60 editing tracks. *Metropolis* is



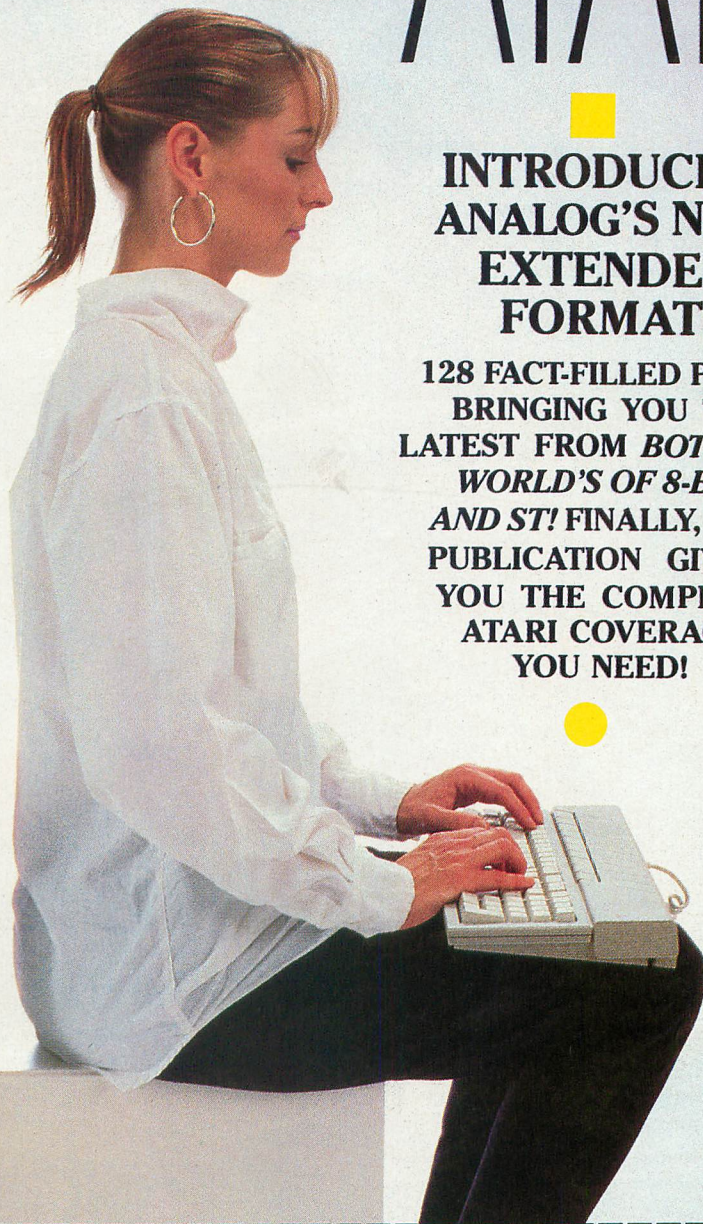
The Complete

ATARI

12 ISSUES
ONLY \$28

INTRODUCING
ANALOG'S NEW
EXTENDED
FORMAT!

128 FACT-FILLED PAGES
BRINGING YOU THE
LATEST FROM *BOTH THE
WORLD'S OF 8-BIT
AND ST!* FINALLY, ONE
PUBLICATION GIVING
YOU THE COMPLETE
ATARI COVERAGE
YOU NEED!



ANALOG
COMPUTING

P.O. Box 16927
North Hollywood
CA 91615

☐ **12 Issues \$28** CJWYY

GET THE NEW ANALOG COMPUTING ON DISK!

☐ **12 ISSUES OF ANALOG COMPUTING WITH
8-BIT DISK—ONLY \$79!** (CDJXY)

☐ **12 ISSUES OF ANALOG COMPUTING WITH ST
DISK—ONLY \$79!** (CDJXW)

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

☐ PAYMENT ENCLOSED ☐ BILL ME

CHARGE MY ☐ VISA ☐ MC

_____ EXP _____

SIGNATURE _____

MONEY BACK ON UNUSED PORTIONS OF SUBSCRIPTIONS IF NOT SATISFIED! FOREIGN—
ADD \$10. MAKE CHECKS PAYABLE TO LFP, INC. YOUR FIRST ISSUE WILL ARRIVE IN 6 TO
8 WEEKS. WATCH FOR IT! OFFER EXPIRES JANUARY 31, 1990

an arcade/adventure game using an overhead 2-D view.

Interceptor Ltd. showed their upcoming new title, *Outland*, to be released on their Pandora Software label. Giving the player near-arcade quality, it features two-plane full-parallax scrolling, a 380K digitized sound track and a 380K animated end sequence. The aliens may have come in peace, but once Earth accepted them... dogmeat time! Now it's up to those Earthlings left on Titan to plan their revenge.

Anco's soccer simulation, *Kick Off*, is one of the fastest-moving games we've ever seen on the ST. It's traditional European soccer, coupled with a U.S. tournament. Even if you don't care one whit for soccer, you'll want to see it just for the speed. An overhead view presents both teams, with a map in the upper-left corner that shows an overview (the screen scrolls vertically). Sound effects are excellent, but it's the animation that really takes the cake here, making for a fast-paced game of highly addictive quality. The goalies are computer-controlled at all times, and they slide and jump all over as the ball approaches them. Control of the team is handled with the joystick and fire button, and takes some getting used to.

Other Stuff

Hat Software addressed the young people's market with *First Paint* and *First Type*. Each is easy to use, and *Paint* contains many of the standard drawing features, such as fill, lines and circles. *Type* helps children learn to use the keyboard and *Jigspell* encourages word recognition through text and pictures.

For a wider audience, there's *Blow Up*, which enlarges images for printing, and *Colour Strip*, a color-separation and sequence-dump utility. *Sprinter* is a package that helps in the designing of patterns for silkscreen printing. It also handles color separations and includes drawing tools. *That's Fun Face* is a hoot. Designed in Holland, *Face* lets you create your own "people" by using the parts supplied. Parts can be shifted left, right, up, down, inwards and outwards. A paint menu allows for airbrushing, bas-relief and text insertion. Five screens can be held in memory at one time, and images can be saved/loaded in formats that include .IMG, .PIx, .RGH, .CLP and .ICN. Most interesting is that printing can be done with the included Postscript driver, as well as with the expected dot-matrix.

At the End of the Day

The show was winding down for the day when I heard an exhibitor mutter, "Two



Marshall Rosenthal and Jeff Minter

more days of this?" As for me, I was off to see the new *James Bond* film at Leicester Square. **A**

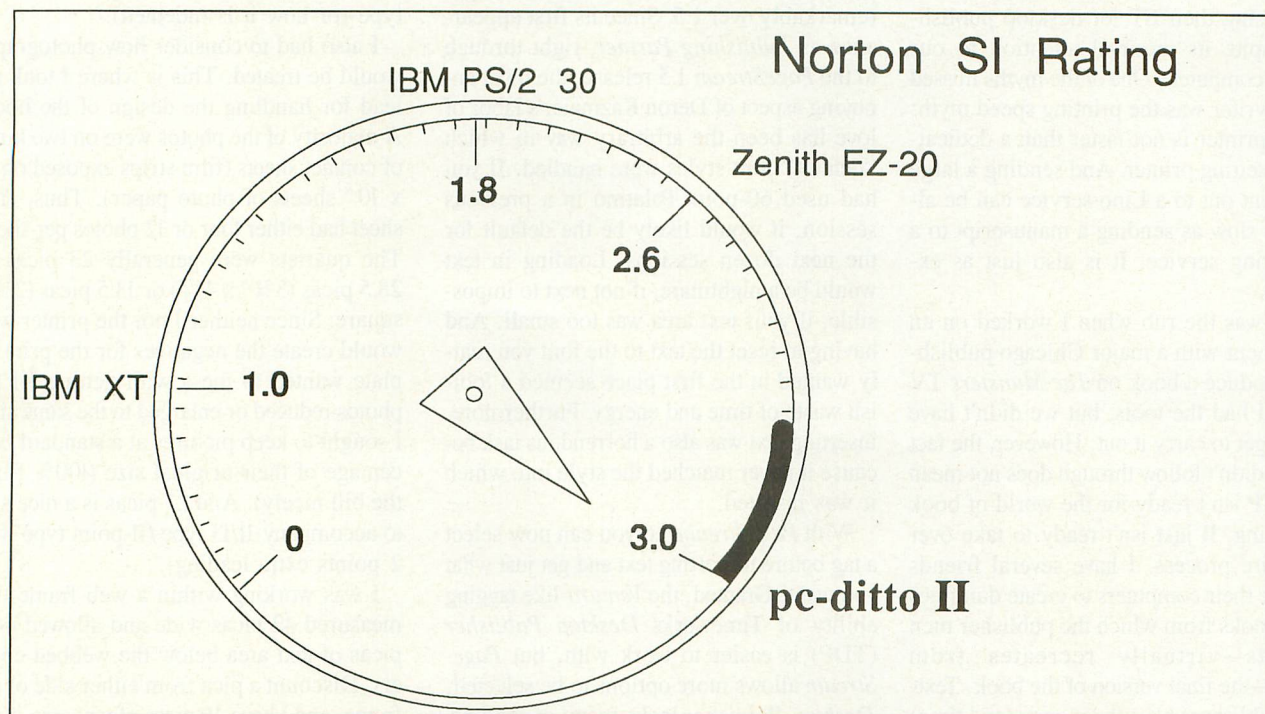


Marshall M. Rosenthal has been involved as a photographer and writer in the overseas market since the early days of Atari. His features and pictorials can be found in major computer- and entertainment-related publications throughout England, France, Germany, Sweden, Mexico and the U.S.

Products mentioned: CONTINUED ON PAGE 32

pc-ditto II

By Avant-Garde Systems



IBM XT COMPATIBLE !

IBM AT PERFORMANCE !

ATARI ST PRICE !

\$299.95

Now! Run the most popular IBM software on your Atari ST...

FAST !

See pc-ditto II at your local Atari dealer or write for free information.

Yes. Please send me more information !

Avant-Garde Systems, 381 Pablo Point Drive
Jacksonville, Florida 32225

Name

Address

City St Zip

(CONTINUED FROM PAGE 14)

ing, editing, compiling notes, creating graphics and laying out the material. (And assumes no corrections are needed.) Also, most corporations require a multitude of executives to make their own additions and subtractions, as do book editors.

Personal Publishing magazine ran an excellent article in their August 1989 issue debunking eight desktop myths. (I highly recommend this magazine to anyone seriously using their ST for desktop publishing despite its minimal attention to our favorite computer.) One of the myths missed by the writer was the printing speed myth: a laser printer is not faster than a dedicated typesetting printer. And sending a large document out to a Lino service can be almost as slow as sending a manuscript to a typesetting service. It is also just as expensive.

That was the rub when I worked on an experiment with a major Chicago publisher to produce a book on *The Munsters* TV series. I had the tools, but we didn't have the budget to carry it out. However, the fact that we didn't follow through does not mean that DTP isn't ready for the world of book publishing. It just isn't ready to take over the entire process. I have several friends who use their computers to create dummies of textbooks from which the publisher then typesets—virtually recreates from scratch—the final version of the book. Textbook publishers have the money (and time) to do this; but slowly, trade-book publishers are beginning to jump into the DTP arena as well.

Enter PageStream

The sample pages included with this column were created using *PageStream* and were employed to guide the design of the *Munsters* book we talked about last month. (The book should be now available at a bookstore near you.) I used *PageStream* because of its ability to wrap text around irregular objects.

Also included in the figures is a page from the working dummy of the book to give you a rough idea of how the typeset book would turn out. Scans of photos were created using PictaScan (E. Arthur Brown) and ST Scan (Navarone), but were for position only, meaning that the printer would use the actual photo for creating a halftone picture.

Another reason for using *PageStream* was that I am very familiar with it.

I am indebted to beta-tester Bill Rehbock and SoftLogik's Mark Wetzel for use of beta

copies of *PageStream* (1.55b and, most notably, 1.57b), which vastly improve upon the performance of 1.5. By now, Version 1.6 is available, and anyone who has been hanging on since the horrendous beta release of 1.0 knows that, at last, *PageStream* is a serious, state-of-the-art publishing program.

To me, the most significant improvement has been the solidification of tagging operations, although most people will find the speed of its loading and redrawing increased remarkably over 1.5. Since its first appearance as *Publishing Partner*, right through to the *PageStream* 1.5 release, the most annoying aspect of Deron Kazmaier's labor of love has been the arbitrary way in which "default" font styles were handled. If you had used 60-point Palatino in a previous session, it would likely be the default for the next dozen sessions. Loading in text would be a nightmare, if not next to impossible, if your text area was too small. And having to reset the text to the font you really wanted in the first place seemed a foolish waste of time and energy. Furthermore, inserting text was also a horrendous task because it never matched the style into which it was inserted.

With *PageStream* 1.6 you can now select a tag before importing text and get just what you want. Granted, the *Ventura*-like tagging ability of *Timeworks Desktop Publisher* (TDP) is easier to work with, but *PageStream* allows more options to be selected. *Desktop Publisher* lacks many of the one-shot typestyle options that *PageStream* offers, so it gives with one hand while it takes away with the other. The main point is that, to master *PageStream*, you must master its tagging capabilities and learn to live with its present limitations.

Setting Up a Template

Thus, once you've set the document size (by choosing New Document under the File menu), your next step in using *PageStream* should be to create tags. These tags can be modified later, but first begin with the "default" tag, which is usually the first tag you create. Name the tag, then go down the list of options. Unless you are getting creative and know what you are doing, leave the Fill and Color options at the bottom of the list alone. I cannot begin to relate the anguish and frustration that dogged me once I'd set a white fill to print black and saw nothing printed despite what appeared correct on screen. You cannot go back to an uncommitted Fill or Color choice.

Let's look at what I had to do after I created a new document for my *Munsters* sam-

ple. In the copyediting process, 16 distinct text treatments were identified. Along with the usual chapter titles, heads and subheads and body-text styles, there were several kinds of sidebars indicated, each requiring a head. This did not mean I had to use 16 different typefaces or sizes. Generally, good design sticks to one or two font families, so the rest of the distinctions are broken down into using bold and italic styles and changing sizes and the measure of a line of type (or how it is indented).

I also had to consider how photographs would be treated. This is where I took my lead for handling the design of the book. A majority of the photos were on two kinds of contact sheets (film strips exposed on 8" x 10" sheets of photo paper). Thus, each sheet had either four or 12 photos per sheet. The quartets were generally 23 picas by 28.5 picas (3 3/4" x 4 5/8") or 13.5 picas (2 1/4") square. Since neither I nor the printer who would create the negatives for the printing plate wanted to mess with getting all the photos reduced or enlarged to the same size, I sought to keep pictures at a standard percentage of their original size (100% filled the bill nicely). And 23 picas is a nice size to accompany 11/13 type (11-point type with 2 points extra leading).

I was working within a web frame that measured 42 picas wide and allowed 44.5 picas of text area below the webbed corners. Discount a pica from either side of the frame, and I have 40 picas of text area. Subtract 23 picas for the main text, plus a 1.5-pica space for breathing room between main text and the sidebar area, and that leaves 15.5 picas. At 115%, the 13.5-pica pictures fit in the sidebar area. If I included the photo inside a box in the 15.5-pica sidebar area, the photo could stay at its original size. Thus, I would generally work with three photo widths: 15.5, 23 or 40 picas.

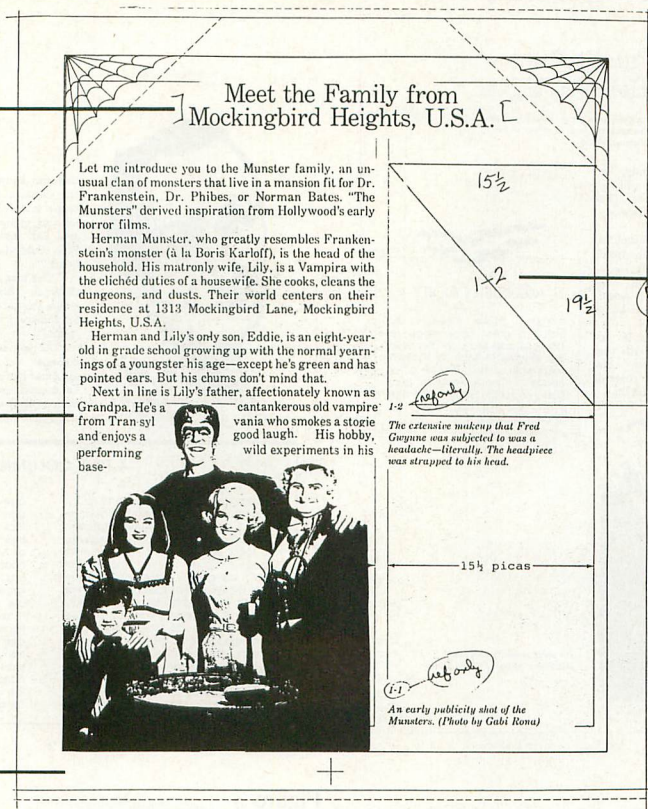
I could also go against this asymmetrical text arrangement with two 18-pica columns of sidebar text with two picas between the columns and a pica on the outside margins. Optimum layout flexibility was achieved using a pica rule and calculator even before I turned on the computer!

Now the tagging operation could begin. Here are a few examples:

~ *Body text*: Schoolbook, 11 point; 13 fixed leading (or 2-point Auto); no change in character spacing; no baseline settings; no margins; 12-point (1 pica) paragraph indent; word justification (never use character justification for books); no fill or color.

~ *A head*: Schoolbook, 15 point; 18 fixed

FOLIOS GO HERE



**BLEED AREA,
WHERE PAGE
WILL BE TRIMMED
AT PRINTER**



Figure 2

My first-pass galleys probably would have followed my page format, but without the web frame to save file overhead (and thus printout time), since I could easily add the frame to the master page of each chapter document later, when camera-ready printouts were needed. After proofing, I would have made corrections within *PageStream* and then began the layout by adding "photos" in the form of boxes drawn to the size of the photo. I would still have to crop and mark the photos for size outside the program. Only for photos where I would have outlines (the background cut away, leaving heads or other prominent foreground features of a photo) breaking into the text would I need to use scans. And those scans, however nice they might look using halftone mode in Navarone's ST Scan (special thanks to Coz Computers in Chicago for its use), would be for position only. Again, to avoid file-size problems, I would have likely foregone halftones and gone no further than 150 dpi (although my sample did use a 300 dpi scan of the Munster family—just showin' off!).

Doing page-sized galleys from the start—unlike what we traditionally do with first-pass galleys—would have put us ahead of the game by having an idea of the page count before photos were inserted. And there's always somebody looking over my shoulder, wondering about the length of the book...

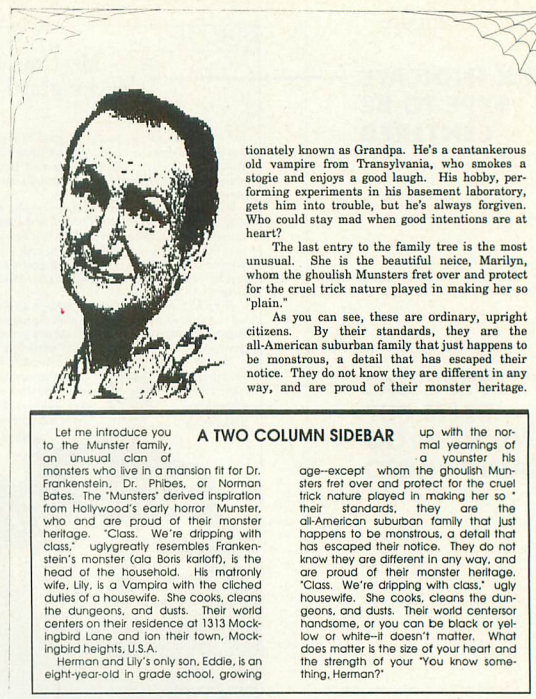


Figure 3

Clicking Off

Except for the problem of 300 dpi PS output being unacceptable and the Lino service being too expensive, it would have been possible to do this book using *PageStream* and the other ST programs, like *WordPerfect*, *Touch Up*, *Easy Draw* and *ST Scan*. It still would have required some traditional hands-on work with scaling and cropping photos, but I've been doing that stuff for years. The point is, we *do* have the tools to do serious, professional publishing with Atari hardware and software.

I'm probably going to return to this book project one last time in order to illustrate the power and applicability of the ST to real-life DTP tasks. Then I'll be looking into various other areas of personal-publishing topics. Next month, however, Christmas comes upon us, and I've always wanted to play Santa's helper and discuss stocking-stuffers and gifts for the personal publisher in your life. I'll try to give you my choices from among word processors, drawing programs, accessories, utilities and hardware, as well as DTP programs and an odd game or two.

1 This is a page from the layout for the *Munsters* book. You can see the web frame as well as all the blue lines (the arrows and "15½ picas" are not meant to print, but are there as a guide for the keyliner). The galley was cut up in order to do the "wrap"

around Herman Munster (note the bit of hyphen in "Transyl," the lack of hyphen after "transyl," and "vania" on the other side of his head). The broken diagonals below the webs are also non-reproducible to the printer's camera and mark off an area where text should not go, as well as marking where the text area begins. Photo 1-2, like the majority of photos in the book, is drawn for position, with measures for the keyliner to reproduce on the boards. The printer will place the halftones (the printable versions of the photos) into position. (All illustrations of the *Munsters* in these samples are copyright 1989 by MCA Publishing Rights, a Division of MCA Inc. Used with permission.)

2 This page shows many of the elements that had to be considered in designing the *Munsters* book: display text, opening quote, opening paragraph and body text, sidebar, folio and caption. There's also some fun stuff with the webs and bats, to say nothing of the text wraparound and photos. One thing lost from the return to traditional typesetting was the shadowed text since the publisher did not have the font. On the other hand, the bats will be treated to a 40% gray screen and print under the larger "MUNSTER ODDITY" head. (All illustrations of the *Munsters* in these samples are copyright 1989 by MCA Publishing Rights, a Division of MCA Inc. Used with permission.)

(CONTINUED ON PAGE 64)

TYPING IT IN

Type in Listing 1, using BASIC Editor II to verify your work, and save the program as MEMATCH.BAS. Then Type in Listing 2, again using BASIC Editor to verify your work, and save a copy to disk. Listing 2 creates a file named MEM.LST, which contains some hard-to-type lines that should be merged with Listing 1. To merge the lines: LOAD "D:MEMATCH.BAS", ENTER "D:MEM.LST" and finally SAVE "D:MEMATCH.BAS".

PLAYING THE GAME

Memory Match can be played by one or two players. A one-player game requires a joystick plugged into port 0. Plug another joystick into port 1 for a two-player game. Press Select to change the number of players. Press Option to turn the sound effects off, if you like. Press Start to play.

The goal is to match all the cards in the fewest number of turns. Rank yourself according to the number of turns it takes you to complete the game.

12-22 turns: You must have ESP!


23-29 turns: Not bad.

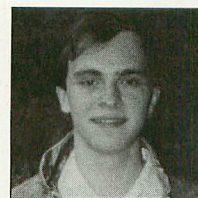
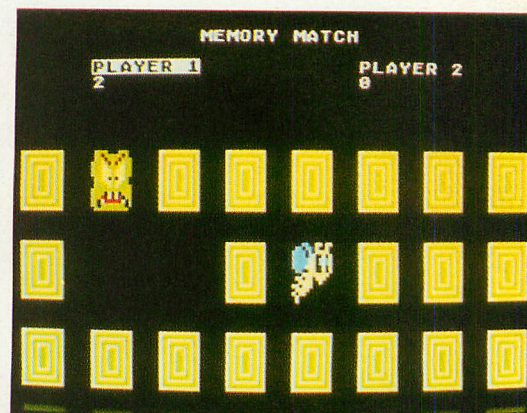
30+ turns: Are the sound effects distracting you?

ABOUT THE PROGRAM

Memory Match is written in BASIC and uses a machine language USR routine contained in the string MM\$. The routine copies a string of memory to a new location. The call format is X=USR(ADR(MM\$), MEM1, MEM2, NUM), where MEM1 is the address of memory to copy from, MEM2 is the address of memory to copy to and NUM is the number of bytes to copy. This routine is used to move the modified character-set data (lines 9010-9070) to an area of memory that begins four pages (each page is 256 bytes long) below the last usable address of RAM. The routine also copies the DLI (display-list interrupt) data in line 9080 to page 6 and copies the cursor data in PL\$ to player/missile memory (line 20).

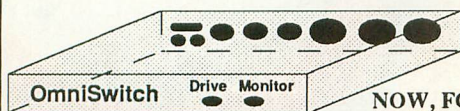
CONCLUSION

I hope you enjoy *Memory Match*. It's a fun game in which the graphics and sound effects enhance the playing experience. What better way to improve your memory? 



Joe Brzuszek is majoring in computer science at the University of Pittsburgh and has owned his Atari 800 since 1983. He uses his Atari as a VT-100 terminal to communicate with a VAX mainframe computer system.

Multi-Sync with Atari ? YES!!!



NOW, FOR THE FIRST TIME YOU CAN HOOK-UP YOUR ATARI TO A MULTI-SYNC MONITOR!

	Suggested retail	*NOW !
OmniSwitch	\$129.95	\$89.95
Samsung CN4551 Multi-sync Monitor	\$799.00	\$499.95

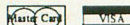
ORDER TODAY:

Inside CA. (619) 792-6511
Outside CA. (800) 243-0443

TALON TECHNOLOGY GROUP

U.S.
243 S. HIGHWAY 101
ste. 308
Solana Beach, CA 92075
phone (619) 792-6511
FAX (619) 792-9023

U.K.
31 Palace Street
London, SW1E 5HW
phone (01) 828-9755
FAX (01) 630-7343



COD's

* plus shipping & handling

- **Multi-sync monitor port:** Gives your Atari the ability to use a multi-sync monitor. Color or B/W with the touch of a button. (low, medium, and high resolutions)
- **Color and monochrome monitor ports:** If you do not have a multi-sync monitor, Omniswitch allows you to plug in color and B/W monitors and switch between them. NO MORE CABLE SWAPPING!
- **Floppy drive ports:** Enables you to use two external floppy drives instead of one! (uses 3.5" and/or 5.25")
- **Audio Jack** to plug in to your amplifier.
- **Video Jack** for composite TV plug in.
- **Dimensions:** W 7.75" X D 4.25" X H 1.25"

TECH WAY SALES

P.O. BOX 605 WARREN, MI 48093

1-800 USA-8832

IN MICHIGAN CALL 1 (313) 751-8807

WE SPECIALIAZE IN ATARI & THE ST LINES!

SOFTWARE & HARDWARE
WITH A FULL LINE OF ACCESSORIES

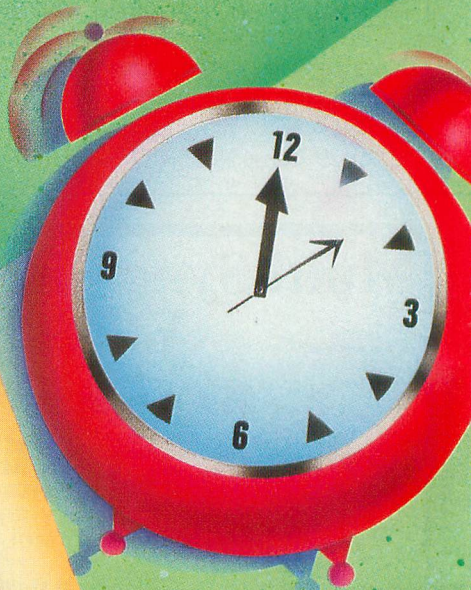
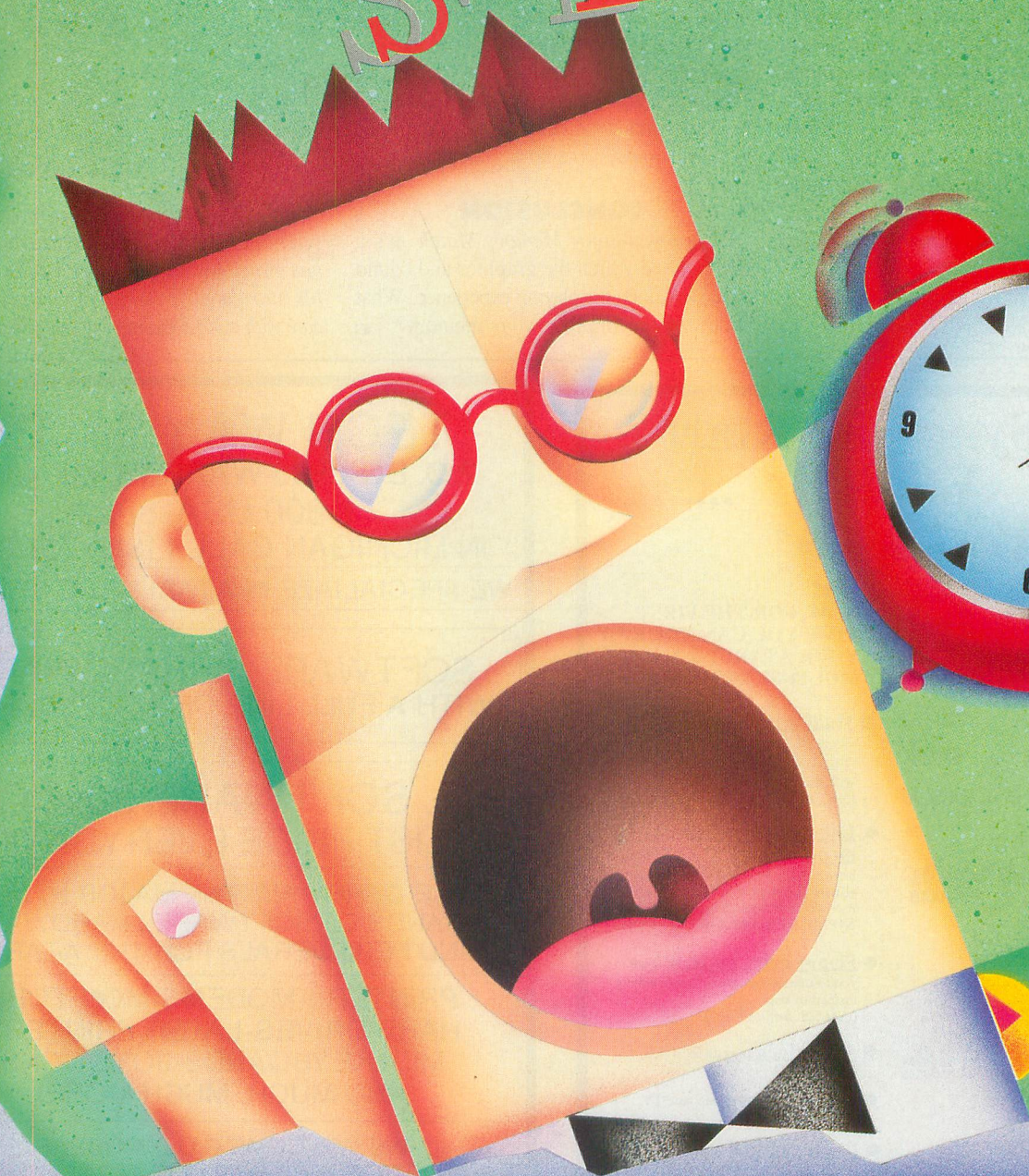
ALL SOFTWARE 30% OFF
LIST PRICE EVERYDAY!!

WE CARRY ALL THE MAJOR NAME
BRANDS OF SOFTWARE, HARDWARE
AND PERIPHERALS FOR THE ATARI'S

PRINTERS-MODEMS-MONITORS
HARD DRIVES-LASER PRINTERS
MIDI KEYBOARDS-JOYSTICKS
AND MUCH, MUCH MORE!

WE WELCOME C.O.D. ORDERS
MOST ORDERS SHIP OUT IN 24 HOURS!

SURVEY SWEEPSTAKE



BY ALBERT BAGGETTA

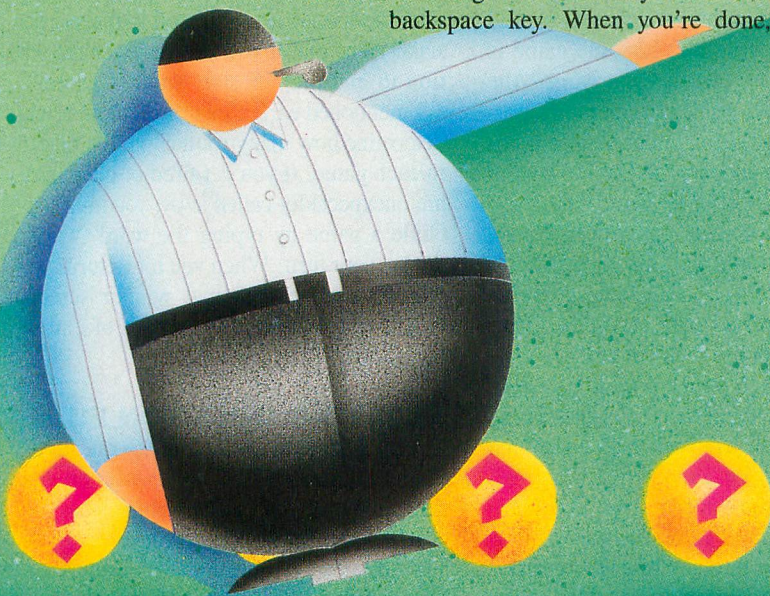
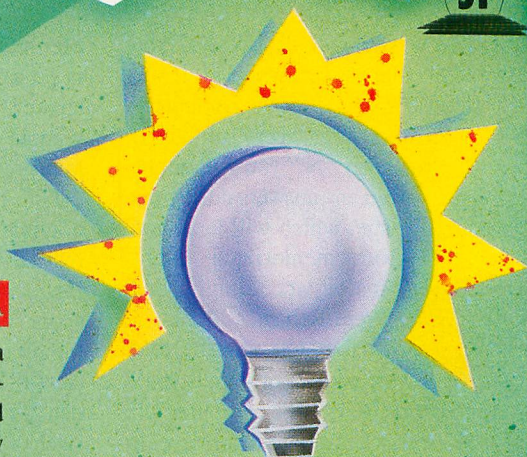
In *Survey Sweepstakes* you play against an egotistical adversary—a built-in obnoxious opponent who affectionately calls himself “Knowing Norman.” He has a carnival-barker personality and is the kind of player you’d like to kick in the pants. There is also a third party in this game: the computer. The computer is in charge of the game. It selects the questions, acts as referee and keeps track of all the calculations and displays.

The computer has conducted a survey in order to discover such things as, for example, the five most popular drinks. (Oh, you didn’t know that computers take a little nip once in a while?) The Atari pits you against Norman and challenges you to name three of the favorites in each round. Pretty easy, huh? Correct answers get you points; incorrect answers get points for the opponent.

PLAYING THE GAME

First, run the file SWEEP.PRГ (use low resolution), and you will see a file selector from which you may choose a question file. (Even Vanna and Pat don’t let you do this.) A sample file of 75 questions is on this month’s disk. Later in this article, I will tell you how to create your own. For now, select the Click QUEST.SWP file. You will then see the questions loading into memory.

After the questions have loaded, you’ll see the set-up screen, where you’ll find Knowing Norman and be asked to enter your name for this game. In subsequent games you can just hit Return, because your name will already be entered. Otherwise, you may enter any name up to 25 characters long. To correct any mistakes, use the backspace key. When you’re done, press



Return. Now you will be asked if you would like the answers revealed. If you select "Y," you'll see the computer's answers after every round. If you enter "N," you will not be given any answers during the game, which will keep you from memorizing them. Answer accordingly, and Press Return.

You will now be told the number of rounds you'll be playing. The computer will select anywhere from five to ten rounds. Don't worry. If you get tired of playing, there's a way to leave the game before it's over. I'll get to that later.

The screen will pause for a few seconds, and then you'll be sent to the main play screen. Before starting play, take a close look at the screen. You will see two black windows outlined in white. The left window is yours (see your name over it?). The right window belongs to Norman. You can see his picture and name over this box. The long white window toward the bottom of the screen is where you'll read the polled questions. These questions have been created and secured by the mythical Brice-Waterhose, Inc. You can begin play by pressing the return key.

When you press Return, you'll see a new display appear at the bottom of the screen. It consists of four indicators arranged as follows: <RIGHT> <WRONG> <ILLEGAL> <TIME INDICATOR>. I will explain these as we progress through the game. At the top of the screen (in the center) you'll find the number of the current round. This number will increase throughout the game until you have played all of the allowed rounds.

The Brice-Waterhose window clears, and a question appears. The question format is the same throughout the game. You'll be asked to identify three of the computer's five most popular selections.

You will always play first. (This advantage will, of course, be offset by an advantage Norman has.) The border around your box will pulse, signaling you to type in your first answer. There are several restrictions you need to be aware of here. First, you may use only 16 characters to answer (including spaces between words). Second, you may enter your answer in all lowercase, or use uppercase for the first character only. Acronyms are the only exception here. These must be entered in uppercase only. For example, you should enter "Young

Men's Christian Association" as "YMCA" to get a correct answer. After you enter your answer, press Return. If you press Return without entering anything, the move will be interpreted as a wrong answer.

Let's take a look at those indicators at the bottom of the screen. When you give a correct answer, the appropriate indicator will flash and ten points will be added to your score. (Your score is displayed at the bottom of your window.) If you give a wrong answer, points will be added to your opponent's score. If you try to give the same answer twice, or if you try to use an answer given by your opponent, the "illegal" indicator will flash. There's no penalty for an illegal answer; you will simply be reprimanded, your answer will be cleared and you will have to enter something else.

To create some suspense, I have added a timer. You'll notice that when your window starts pulsating, the time in the lower-right of the screen starts dropping from ten to zero. Finish typing your answer before the timer reaches zero; otherwise, you will lose ten points to your opponent.

After judgment has been passed on your answer, the computer switches control to Norman's window. Norman will enter his answer after he has had time to "think." (While Norman is a state-of-the-art opponent, he is a slow and sloppy typist. However, he does make every attempt to be accurate, so he will correct any typos he makes. Being a novice actually makes him a little more human.)

There are a few other facts you should know about scoring and how the game is played. There are two types of games: the normal game and the bonus game. In a normal game the award is ten points. In a bonus game (announced by a flashing sign in the upper-left corner of the screen) the prize is 20 points for each correct answer. Points, by the way, are multiplied by a random figure and accumulated to arrive at a sweepstakes prize, awarded at the end of the game.

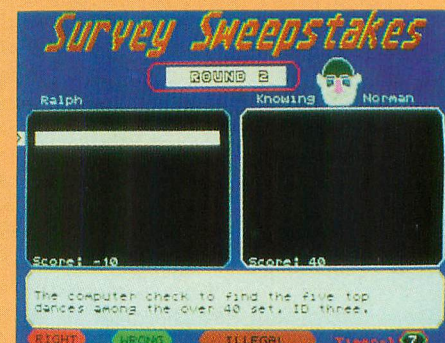
Each question has three of five possible answers. But each player does not have an infinite number of guesses. Four guesses are allotted each player, with one additional guess awarded for each correct answer. Wrong answers are costly, even in a regular game. If a player cannot answer a question correctly, he not only loses the awarded points, but the lost points are added to the opponent's score.

Remember the timer I mentioned earlier? When the meter runs out for you, good ol' Norman receives ten points (20 in a bonus round). Unfortunately for you, Norman

is not subject to this penalty when his timer runs down. ("Not fair!" you say. Well, remember that little advantage you had by starting the game?)

Each round will eventually come to an end. Either one of the players will guess three of the correct answers, or the round allotment will be used up by each player. Whoever finishes the round will gain an additional 20 points.

One other rule of play must be mentioned. Each player has the right to challenge at least once in each round. If, for example, you are not sure of an answer, simply type in the word "Challenge." Norman will then have to come up with the answer. If he answers correctly, however, he gets the points. If he does not, you will receive them.



At the end of each round, applause will be given in the form of a vibrating screen, identifying the leader up to that point. At the close of the game, another colorful screen will announce the final winner, along with the money accumulated in the sweepstakes. The player will also be given the option to end play or start another game.

I mentioned earlier that the computer will determine how many rounds will be played in each game. If you get tired of playing or are unexpectedly called away, you may conclude a game by typing the word "Quit" during your turn. When you hit Return, you will be sent to the final winner screen, where you will have the option of ending the game.

CREATING QUESTION FILES

After playing with the default questions (found in the QUEST.SWP file on the magazine disk), the game becomes easy to beat, so you might like to make up some new questions of your own. The games can be tailored in this way toward specific subject areas or degrees of expertise and may be played by your friends.

A question file is an ASCII file that can be created with most word processors or

(CONTINUED ON PAGE 30)

RIO DATEL COMPUTERS

800-782-9110

*ORDERS ONLY

MON-SAT 8AM-6PM (PACIFIC TIME)

AUTHORIZED ATARI SERVICE CENTER

COMPLETE LINE OF HARDWARE & SOFTWARE IN STOCK

CUST SERVICE
TECH SUPPORT

702-454-7700
TU-SA 10AM-6PM

FAX: 702-454-7700



GENISCAN ST

*Easy to handle Scanner featuring 4 1/8" scanning width & 400dpi resolution enables you to reproduce graphics & text on your computer screen *Optical Character Recognition Mode will capture text images from the page & save them in format ready for conversion to text file suitable for word processor *Package includes GS-4000 scanner, interface & Scan Edit software *Powerful partner to Desk Top Publishing *Printout for Epson compatibles *Unmatched range of edit/capture facilities simply not offered by other scanners at this unbeatable price *Adjustable switches for brightness & contrast *Cut & paste editing of images *Save images as IMG, Degas, Neo, etc - **ONLY \$299.95**

OCR SOFTWARE - Teach your Atari to read!

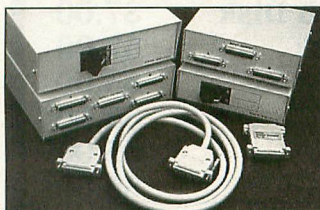
*Optical character recognition software converts scanned text to ascii*Use converted text in your word processor*No more manual transcribing of text*Uses special scanning mode of the Geniscan ST*Learn mode lets you teach your computer any font - **ONLY \$89.95**

TRACKMASTER ST



*Trackmaster track display unit is a must for the disk hacker*Simply plugs into disk drive port of the ST*Displays track number, disk side and read/write status as the head moves along the disk*Two-digit LED display*Displays up to track 85*Works on both internal and external drives (switchable)*Complete hardware solution - no software required - **ONLY \$59.95**

ACCESSORIES



AB Switchbox	24.95
ABCD Switchbox	39.95
Printer Cables, 6 ft.	14.95
Modem Cables, 6 ft.	14.95
Serial Cables, 6 ft.	9.95
Parallel Cables, 6 ft.	9.95
Gender Changer, M/M, F/F	4.95
Null Modem Adaptor	4.95
RS232 Surge Protector	9.95
RS232 Wiring Box	14.95
Diskette Holder, 3.5", 50 disk capacity...	6.95
Bulk 3.5" Disks, 50 Pack	39.50
Printer Ribbons, (Star, Panasonic)	4.95
3.5" Head Clearing Kit	4.95

EPROM BOARD

*Complete Eeprom Board & case *Accepts up to 4 X 27256 Eeproms *Mapped into the cartridge area - **ONLY \$24.95**

SAMPLER STUDIO

*A complete sound sampling package *100% machine code for real-time functions *Saved files are compatible with other sampling systems *Sequencer for playback via keyboard or MIDI input - **ONLY \$99.95**

ADVANCED ART STUDIO

*All the facilities needed for computer artwork, sprite & game design in one powerful but simple to use system *Graphic tools for home & professional use *Animated sprite designer & world map creator *Icon driven toolbox with on-screen help *16 color to res mode operation - **ONLY \$29.95**

SKYLOT

*Most advanced features of any astronomy prg. *Covers 17,000 stars, planets & nebular objects *On line help *Zoom display thru 12 orders of magnitude *Several star chart formats supported *3D views *Search command *Compute Eclipse, transits, & conjunctions *supports med & hi res display *requires 1 meg mem - **ONLY \$49.95**

ST COPY DEEP SCAN BURST NIBBLER

Copy even highly protected programs in under 60 seconds!!

Copy MS DOS protected programs with equal ease!!

*Works with one drive or two *Copy between 5.25" and 3.5" drives *Multiple copy option allows many backups from one original *Copy 1 or 2 sides up to 85 tracks *Special format parameters for non standard formats *Unique "INFO ANALYSER" displays vital disc parameters including sector distribution, dispersion, interleave, etc *Easy to use Icon driven program takes the mystery out of Disc backup - **ONLY \$39.95 Upgrades ONLY \$12.00**

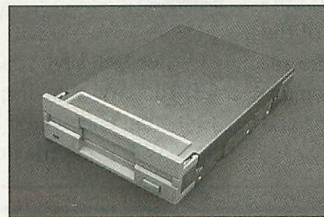
3.5" & 5.25" EXTERNAL DRIVES



3.5" - *Slimline extra low profile unit *Top quality drive mechanism *Fully compatible - 1 meg unformatted capacity *Can be formatted to 85 tracks and 11 sectors - **ONLY \$159.95**

5.25" - *Operate in 40 or 80 track mode *Rear panel switches to configure ALL combinations of computer model and track modes *Perfect for use with PC Ditto- **ONLY \$179.95**

1 MEG INTERNAL DRIVE UPGRADE



*Replace internal 500K drive with a full 1 Meg unit *Top quality drive unit - Easily fitted - Full instructions *Direct plug in replacement *No special skills or tools required *Minor modification to enclosure is required *When considering a drive replacement remember that quality is most important - **ONLY \$119.95**

REPLACEMENT MOUSE



*Twice the tracking speed of the Atari mouse!

*No more mouse doublers required *Softer more positive button clicking *More consistent double click *Extra long 6 foot connecting cord *highest quality mechanical parts for for longer life and consistent positioning - **ONLY \$44.95**

ST SUPERTOOLKIT

*Whole disk Track & Sector Edit *Read up to 85 tracks and 255 sectors! *Super HiRes display works with Mono monitor only *3 different editor modes-Drive, Disk or File *Direct select function for Bootsector, Directory sector *Fully Menu/Icon driven *HEX / ASCII edit modes directly on the disk *Perfect for the disk hacker - **ONLY \$24.95**

ST TIMEKEEPER

*Simple plug-in battery-backed calendar clock cartridge *Fits into cartridge port *On-board clock chip *Lithium battery backed to last up to 5 years *Automatically adjusts for leap year etc *Complete with install disk and corner clock display *Exact time, date etc available on power up - **ONLY \$49.95**

PC DITTO II - \$249.95

PC DITTO II + 5.25" DRIVE - **\$419.95**

PC DITTO II + 5.25" DRIVE + DRIVE SWITCH - **\$451.95**

AUTHORIZED ATARI SERVICE CENTER

LET US REPAIR YOUR EQUIPMENT - 5 DAY REPAIR OR THE SHIPPING IS FREE - \$ CALL

ALL THE LATEST EUROPEAN SOFTWARE

CALL FOR PRICE AND AVAILABILITY

AERCO MEMORY BOARD UPGRADES

ABSOLUTELY THE EASIEST INSTALLATION - UNPOPULATED BOARD - \$139.95
* 1/2 MEG RAM - \$80.00 * 2MEG RAM - \$240.00 * 4MEG RAM - \$480.00

COMPLETE LINE OF MIDI PRODUCTS

MANY OTHER ITEMS IN STOCK - IF YOU DON'T SEE WHAT YOU WANT CALL FOR LOW, LOW PRICES

RIO / DATEL

3430 E. TROPICANA AVE. #67
LAS VEGAS, NV 89121

*Add \$4.00 shipping/handling in the continental U.S.: \$7.00 - PR, AK, HI, FPO, APO: \$10.00-Canada/Mex: C.O.D. orders add \$3.00 to above charges: SPECIFY COMPUTER MODEL WITH ORDER: VISA/MC/Checks/Money Orders/C.O.D. Accepted: Please call for return authorization number - returns subject to a 20% restocking fee: We reserve the right to substitute products of different appearance but equal quality and function for items pictured above: Prices subject to change without notice.

800-782-9110
702-454-7700
IN NEVADA

(CONTINUED FROM PAGE 28)

text editors. There is, however, a definite format to the file that must be followed.

To make a file, the first item you must type is the number of questions you will create. You may put this information in later if you are not yet sure of the number of questions. After this, press Return only once. Next, type your first question. This must be two lines of no more than 40 characters each. Press Return at the end of each question line. It is best to organize each question to fit on two lines. However, if you do not need two lines for your question, just press Return to add a blank second line.

Now you need to enter ten possible answers for this question. The first five must be acceptable answers. These can be anything you want. The second set of five answers must be wrong and will be used by Norman for alternative guesses. None of these should be the same as the first five entered, and none of them should be correct answers. The first letter of each of the answers should be capitalized and the remainder in lowercase. Each of these ten answers must be entered on a separate line and followed by a Return.

To make this a little clearer, let's take a look at the setup for the beginning of the default file on the magazine disk:

```

75
Our computers have polled for the five most
popular colors. Identify three to win.
Red
Green
Blue
Orange
Yellow
Black
Purple
Pink
Burgundy
Tan
Which three of five board games have proved
most popular in our polls?
Monopoly
Life
Scrabble
Clue
Checkers
Chess
Backgammon
Beat the Clock
Boggle
Othello
    
```

When you save your questions as an ASCII file, you may use any filename for the first eight characters. But the extender must be .SWP. If the file does not load properly,

go back and check that you have followed the format carefully. Creating the questions can be time-consuming and requires some invention, but it can be fun and educational. I found the almanac and encyclopedia to be of great help. Within a short time, you can have an informative collection of questions and a whole new game to offer your friends and relatives. **A**



Albert Baggetta is an English teacher and a professional guitarist. He lives in Agawam, Massachusetts, with his wife, Beverly, and his two children. He frequently can be found wandering the ST SIG on DELPHI.

LISTINGS

SIDELINE SOFTWARE

850 N.W. 57th Court - Ft. Lauderdale, Florida 33309 - (305) 938-9755

IMPORTERS OF FINE EUROPEAN SOFTWARE



1-800-888-9273



(305) 771-2161

Falcon Mission Disk	28.00
Rick Dangerous	30.00
Bloodwych	32.00
Sleeping Gods Lie	30.00
Waterloo	30.00
Wierd Dreams	30.00
KULT	30.00
New Zealand Story	30.00
AARGH!	30.00
Rodeo Games	30.00
Garfields Winter Tail	30.00
High Steel	30.00
Asteroth	32.00
Spherical	30.00
The Story So Far	30.00
<small>Ikari Warrior, Buggy Boy, Beyond Ice Palace, Battleship</small>	
Triad 3 Pack	20.00
<small>Barbarian, Defender of Crown, Starglider</small>	

CALL US - NEW TITLES DAILY !!

MAGAZINES

ST ACTION	- Now With Disk	\$7.00
ST USER	- Now With Disk	\$7.00
ST FORMAT	- With Disk	\$7.00
ST WORLD		\$5.00
THE ONE		\$5.00
THE GAMES MACHINE		\$5.00
ACE		\$5.00

Yearly Subscriptions Also Available

**ATARI ST
SERVICE**

**520, 1040, MEGA, XL & XE
UPGRADES & PARTS
ALSO AVAILABLE**

Increase your publishing power.

Migraph Hand Scanner.



Add scanned graphics to all your projects quickly and easily with the Migraph Hand Scanner.

The Hand Scanner provides a scanning window over 4" wide with four scanning resolutions: 100, 200, and true 300 and 400 dots per inch. It also offers adjustable contrast, three dither settings for scanning halftones, and a special setting for line art.

Build your own library of images. Logos, photographs, books, illustrations . . . provide you with an unlimited supply of art.

Touch-Up,[™] Migraph's high resolution image editor, lets you scan and edit your images. Powerful editing tools include crop, invert, resize, flip, cut and paste, multiple zoom levels, a full array of paint tools, and much more.

When your images are "pixel perfect", import them into your favorite ST publishing programs like Calamus, Fleet Street, PageStream, and Publisher ST. In addition, several

save formats let you use your images on the PC and Mac.

The Migraph Hand Scanner and Touch-Up. Powerful tools for professional publishing.

See your dealer today for more details or call us direct. Dealer inquiries invited.

For all Atari ST and Mega systems with monochrome or color monitors, 1MB memory and 3.5" DS disk drive.



200 S. 333rd St., Suite 220

Federal Way, WA 98003

(206) 838-4677

(800) 223-3729 (Pacific Standard Time)

© Copyright 1989 Migraph, Inc. The Migraph logo is a registered trademark and Touch-Up is a trademark of Migraph, Inc.

CIRCLE #106 ON READER SERVICE CARD.

(CONTINUED FROM PAGE 20)

Products mentioned:

FIRST PAINT

FIRST TYPE

JIGSPELL

COLOUR STRIP

SPRINTER

BLOW UP

Hat Software

21 Wyle Cop

Shrewsbury, Shropshire

England SY1 1XB

FORGET ME CLOCK II

DISK DUPLICATOR

P.O. Box 113

Harrogate, North Yorkshire

England HG2 0BE

HAWK COLIBRI

MATSCREEN-M110

METEOSAT

Signa Publishing Systems Ltd.

3rd Floor, Alexander House,

Station Road

Aldershot, Hants

England GU1 1BQ

KICK OFF

Anco Software

Units 9-10, Burnham Trading Estate

Lawson Road, Dartford, Kent

England

MULTIGEN

Digita International

Black Horse House

Exmouth, Devon

England EX8 1JL

PARSEC

Elmtech Research

6 Witherford Way

Selly Oak, Birmingham

England B29 4AX

TWYLYTE

METROPOLIS

TRITON 3

STOMP

Software Horizons

5 Oakleigh Mews

London, England N20 9QH

OUTLAND

Pandora Software

Mercury House, Calleva Park

Aldermaston, Berks

England RG7 4QW

VIDI ST

Rombo Ltd.

6 Fairbairn Road, Kirkton Road

Livingston, Scotland EH54 6TS

THAT'S FUN FACE

Compo/Holland

Cavendish Distributors

85 Tottenham Court Road

London, England W1

SPECIAL CHARTER SUBSCRIPTION RATES

**6 ISSUES
ONLY \$13.95**

**SAVE 40% OFF
THE COVER PRICE**



INTRODUCING

PC LapTop
COMPUTERS MAGAZINE

☐ SIGN ME UP FOR 6 ISSUES OF LAPTOP MAGAZINE FOR ONLY \$13.95!

☐ PAYMENT ENCLOSED ☐ BILL ME

CHARGE MY ☐ VISA

☐ MC

No. _____

EXP _____

SIGNATURE _____

NAME _____

ADDRESS _____

CITY _____

STATE _____

ZIP _____

Money back on unused subscriptions if not completely satisfied. Foreign add \$7. Make checks payable to LFP, INC., Your first issue will arrive in 6 to 8 weeks. Watch for it.

CJWYV

Offer expires November 26, 1989.

I think the game could be richer and more appealing to a wider audience if there was more to do than kill anyone who approaches.

QUEST

BY
IAN
CHADWICK

Maybe I'm just getting old.

I was in Electronic Playworld a few weeks ago asking what's new. Richard pointed out a game called *Battletech*, a graphic adventure from Infocom. Now, that interested me, because I've always had a high regard for Infocom's games.

"What's it about?" I asked, always a dangerous question.

"It's based on the Japanese cartoon series," he replied, as if that was all I needed to know.

Unfortunately, it stumped me. Japanese cartoon series? I wasn't aware that any Japanese cultural figures had made enough impact on these shores to be recognized by more than a few film aficionados. Toshira Mifune is modestly well known enough, thanks to his role as Toranaga in the *Shogun* mini-series, but mention Kurosawa to Rambo-level moviegoers, and all you get is "huh?" When you've got pop comic stars, you know we're at an intense level of cultural exchange. Does this mean they're reading *Batman* over there?

I'm not a television watcher, so I haven't the foggiest idea if "Battletech" is a show. For me, a heavy week of TV watching is four hours, and that's probably two or three hours of PBS specials and a couple of M.A.S.H. reruns during dinner. Anything more strikes me as bordering on addiction. Our set never goes on before 7:00 p.m. and is often off for days at a time. If *Battletech* has come on the boob tube, it's slipped by me.

But it's obviously a popular comic-book series. I managed to see a cover in a local comic shop. I'm not a comics fan; I think comics are literature for the hard of thinking. Sure, there's a cult image and a cer-

tain amount of fun in the *Fabulous Furry Freak Brothers* or *Dr. Atomic*, but kids get carried away. One of the local bookstores is going over from literature to comics (where the money is, I'm told), and it fills me with a cold dread to be snooping around for a collection of short stories by Gogol and hear these kids talking about comics as if they're important contributions to culture—serious discussions on characters like Silver Surfer, Dr. Doom and a host of others. These kids analyze the characters' attitudes, motives, attire and so on. It's much like soap operas, I suppose—a fascination with the trivial details of a meaningless pastime. But they act as if comics and soap operas actually have meaning and depth.

Maybe it's a factor of age. Maybe you have to be young to appreciate comic books. When I was young (back in the Paleozoic), I liked things I have no interest in pursuing now: sweet wines, Mickey Spillane novels, the radical left, cigarettes—all the follies of youth. Maybe when the comic-readers get to be my age, they'll grow into more relevant things. Like sweet wines, Mickey Spillane novels. . . .

Of course, when I was a kid, I read a few comics, though there wasn't a fraction of the number available today, and no one seemed to take them seriously. Mostly I chose the Classic Comic series, in part because they helped get me through school without having to read such tedious tomes as *The Red Badge of Courage* and *Wuthering Heights*. Instead, I had discovered Orwell, Kawabata, Darwin and Golding and wanted to save my reading time for them.

Anyway, I bought the game, figuring it would be a challenge since I had no precon-

ceptions about the theme or the characters, and also because I wanted to see what Infocom was up to these days.

Battletech reminds me a lot of *Sundog*, an older graphic adventure from FTL and still one of the best ST games around (if you can find it). The interface is similar. Because there's a specific purpose in the game, *Battletech* is more limited than the freewheeling *Sundog* in its available options, but overall, it proved quite an interesting game. I think graphic adventures is a direction in which Infocom ought to continue.

One of the rules of an adventure game is to discover what the rules are, so I won't go and spoil your fun in discovery. However, I will give you this: The beginning is a peanut-butter sandwich game. Got that? No? Here's the theory: There are some games where *things* happen without your intervention. With these games, you should best walk away and make yourself a peanut-butter sandwich and enjoy it while the system goes about its merry way and does what it must. In this case, the hint is: money. 'Nuff said.

Of course, the documentation you get with the game gives a few hints, but nowhere near enough. You'll learn, but it takes a few tries before you uncover some of the tricks and secrets. I've only learned a few—like how to get out with an intact mech suit—but they sure count! Mostly, the docs provide a map and mech suit diagram, both of which get used a lot.

The game isn't copy protected, but has no provisions for using a hard disk (at least none stated in the docs). This is a pain because it does a lot of disk I/O, and nothing grinds my teeth more than having to wait

while the disk whirls.

For a long time I thought the game involved little more than careening about the countryside in your mech suit, mapping the terrain and shooting anyone who came close. That sort of activity soon wears thin. I finally discovered that the point of the game is revealed over a period of time and by certain actions you undertake (and influenced by the order in which you take them). It actually becomes somewhat involved, like a complex scavenger hunt.

The user interface is well-designed and easy to learn and use. It's very smooth, but there are places I wish I had a simple alternative. For example, in combat, one mech has nine weapons, which can be individually targeted. However, if I want to fire everything at one target (the usual case), I still have to go through the process of clicking each weapon, then each target. An "all weapons/one target" button would be nice.

Battletech is nowhere near as polished and sophisticated as an Infocom text adventure. The options are severely limited and the available range of actions is pretty narrow; but of course, it has graphics, and they count for a lot. Still, I would prefer to see them apply the same approach to a game like *Deadline*, rich in characters and options. Also, the logic in the earlier Infocom games had more internal consistency than *Battletech* exhibits, although this is at most a minor annoyance.

The main drawback and the point that irritates me the most is that the interaction with 95% of the nonplayer characters has to do with fighting—to the death. It's not as violent as, say, *Techno Cop*, but once you get your character into a suitable mech suit, you go about blithely killing anything that moves. A major battle between your side and the enemy, resulting in eight or ten deaths, is referred to in the game as "an invigorating scuffle." Wonder what they'd call a nuclear war?

There's no wildlife in *Battletech*. I infer from this that the wildlife on Pacifica (what a misnomer!) was previously wiped out by gangs of mech-suited hunters before they turned their weapons on each other.

It's also a male-dominated game, not surprising given the origin and the topic. Comics and violence seem to appeal more to young males than females. Even so, a few female characters would be pleasant. Not only more realistic, but maybe romantic as well. Yes, female warriors too. *Dungeon Master's* handling of the sexes shows a more mature approach. But since I've never seen the comic, I don't even know if there are women in the "Battletech" mythos. May-

be it's all guys. Like the Spartans. I guess that appeals to some males.

I've railed on about violence in games before, and here I go again. I think the game could be richer and more appealing to a wider audience if there was more to do than kill anyone who approaches. Once I got three mechs together and upgraded, I had an almost invincible party. I started to avoid combat, not because I was afraid of losing, but because it became boring. Click the weapon, target, next weapon, target, begin fight. Sigh.

I'm of the notion that there's too much violence around us already. I do believe that it affects us when we see it on TV, in movies or in games. I think we become desensitized to the real fact of other people's pain. It brutalizes the human spirit to be constantly barraged by violence. A few years back there was a controversy over several video games (one or two of which were for the Atari 800XL) that graphically depicted sex, and in one particular case the rape of an Indian woman by a cowboy. A lot of people howled in outrage over these games, saying they'd affect their kids' perception of sex. Some were even perceptive enough to protest the way the Indians were depicted. The same goes for violence. You can't see it, hear it, read it, play it in your games without being affected by it.

We ought to have a game-rating system like they have with movies—PG, PG-13, R and so on. Parents should try anything rated R or higher first, to determine if it's suitable for their kids. And maybe stores should be asked to refrain from selling to young kids any game that's not suitable for them. I'd rate *Techno Cop* XX or maybe even XXX, obscene violence. *Battletech* would be R. I wouldn't give it to a kid under 14.

Much to my surprise and delight, I have recently been sent a C interpreter, from HiSoft and marketed on this continent by Michtron. No, I haven't tried it, but it's the next thing on my list. My only worry is that it doesn't come with a compiler. The whole point of the exercise seems to me to create compiled code from whatever language you're using. The back cover promises the code compiles with "any standard compiler", without bothering to name any with which it has been proven. How do I know whether what the English think of as a "standard compiler" is even available over here? If you buy the package and pursue the matter, you'll find mentions of Lattice, Aztec and MegaMax C. However, the information about problems using these compilers, differences between various im-

plementations and recommendations are mighty few.

A quick glance through the manual revealed several typos. But by now I don't have to warn you about Michtron manuals, do I?

I have no idea yet how the interpreter works with things like the Resource Construction Set. However, despite the documentation, if the program proves workable, it will be a tremendous boon to a lot of inspiring programmers who want to sink in the C swamp but simply don't want to wade through the compiler morass to get there.

Michtron also sent me *K-Graph3*, a graph-construction program reminiscent of the old and cherished *B/Graph* for the 800 series. On page 15 of the manual, it says, "More information on formats is available in the REFERENCES section of this manual." Needless to say, there isn't a REFERENCES section. And the screen shots were entirely left out and had to be included together on a separate errata sheet. Page 38 says this about the Wilcoxon Rank Test: "... computes the 'W' value for the test."

Hope you understood that explanation, because there ain't no more coming. Any idea what skewness and kurtosis are? Even though they're inadequately explained on page 36, the index doesn't mention them. Or slope, intercept, quartiles and other concepts that appear in the docs. Another Michtron manual.

Non-sequitur Interruptus

I adopted two ferrets in June. The lack of information about these animals is appalling. If anyone has any names of books, magazines, ferret clubs, articles, reports or other information I can use, please contact me at: 47 Oakcrest Ave., Toronto, Ontario Canada M4C 1B4 or by FAX at (416) 698-8880. ☐



Ian Chadwick is a Canadian freelance writer who also does volunteer work at the Toronto Humane Society. He and his wife share their small house with dogs, cats and ferrets, and, at odd times, a time-traveling Stenonychosaurus.

(CONTINUED FROM PAGE 126)

```

{
    int x, i, d, len;
    char s2[20], s3[20];

    strcpy ( s2, s );
    len = strlen ( s2 );
    i = 0;
    d = FALSE;
    for ( x=0; x<len; ++x )
        if ( d && s2[x] != ' ' )
            ++i;
        else
            if ( s2[x] == ' ' )
                d = TRUE;

    if ( i == 0 && len < 6 )
        strcpy ( &s2[i], "00" );
    else
        if ( i == 1 || len == 6 )
            strcpy ( &s2[i], "0" );

    i = 0;
    for ( x=0; x<strlen(s2); ++x )
        if ( s2[x] != ' ' & s2[x] != '.' )
            s3[i++] = s2[x];
    s3[i] = 0;
    strcpy ( s, s3 );
}

long str_to_long ( s )
char *s;
{
    int x, len, factor;
    long num;

    num = 0;
    len = strlen ( s );
    factor = len - 1;
    for ( x=0; x<len; ++x )
        num += (long) ( s[x] - '0' ) * pwr5[factor--];
    return ( num );
}

write_new_info ()
{
    int len, x;
    char s[10], tmpfile[64];
    FILE *f;

    string = get_tedinfo_str ( newacct_addr, NEWNAME );
    fwrite ( string, 1, 26, acctfile );
    string = get_tedinfo_str ( newacct_addr, NEWADDR );
    fwrite ( string, 1, 26, acctfile );
    string = get_tedinfo_str ( newacct_addr, NEWCITY );
    fwrite ( string, 1, 26, acctfile );
    string = get_tedinfo_str ( newacct_addr, NEWSTATE );
    fwrite ( string, 1, 3, acctfile );
    string = get_tedinfo_str ( newacct_addr, NEWZIP );
    fwrite ( string, 1, 10, acctfile );
    string = get_tedinfo_str ( newacct_addr, NEWBALNC );
    no_decimal ( string );
    balance = str_to_long ( string );
    fwrite ( &balance, 1, 4, acctfile );
    if ( fclose ( acctfile ) != 0 )
        form_alert ( 1, "[1][File close error!][OKAY]" );
    for ( x=0; x<13; ++x ) {
        sprintf ( s, "%d", x );
        strcpy ( &s[strlen(s)], ".dat" );
        ob_tedinfo = (TEDINFO *) newfile_addr[FILENAME].ob_spec;
        tmpfile[0] = Dgetdrv () + 'a';
        strcpy ( &tmpfile[1], "" );
        Dgetpath ( &tmpfile[strlen(tmpfile)], DFLT_DRU );
        strcpy ( &tmpfile[strlen(tmpfile)], "\\\" );
        strcpy ( &tmpfile[strlen(tmpfile)], ob_tedinfo->te_ptext );
        strcpy ( &tmpfile[strlen(tmpfile)], s );
        if ( ( f = fopen ( tmpfile, "bw" ) ) == NULL )
            form_alert ( 1, "[1][Error creating file!][OK]" );
        else
            fwrite ( &zero, 2, 1, f );

        if ( fclose ( f ) != 0 )
            form_alert ( 1, "[1][File close error!][OK]" );
    }
}

```


NOW ST-LOG AND
ANALOG HAVE BEEN
COMBINED.
ONCE AGAIN,
THE END USER
WILL COVER
BOTH THE
8-BIT AND ST
COMPUTERS
AND THE
SOFTWARE THAT IS
AVAILABLE FOR THEM.

THE END USER

by Arthur Leyenberger

It seems like a long time since I sat here at the keyboard letting my thoughts flow through my fingers and into the computer. Actually, it hasn't been that long, just a week. But having been overseas for the past week, I feel like a portion of my life somehow got misplaced.

You see, I have just returned from the Philippines. It was a good trip from a business standpoint. And personally, it was interesting to see another culture and meet people who are both different and yet the same as myself. But 20 hours of flight time (not counting the multi-hour layovers) and crossing the international date line twice can leave a fellow tucked out, not to mention confused about what day it is.

I brought a laptop computer with me to get some work done on the plane and in the hotel. No, not the Atari ST laptop; it isn't out yet. Anyway, after about three hours, the laptop's batteries gave up the ghost, leaving me to find some way to entertain myself for the remainder of the journey to Manila. No problem. After a couple of movies, several issues of *Time* and *Newsweek* (read cover to cover), some Heinekens and about 14 pounds of honey-roasted peanuts, I finally arrived

at the Ninoy Aquino International Airport.

By this time I felt fuzzy, needed a shower and was having difficulty keeping my eyelids in the raised position. I then had a long wait to get my luggage, during which I imagined all kinds of horrors, such as being stranded in a foreign land with no clean underwear. The luggage arrived and I eventually made it through the heat, humidity and traffic-congested streets of Manila to the hotel.

Wanting to charge the laptop when I arrived at my room, I asked the hotel receptionist what AC voltage was used in Manila. She politely told me that 220 volts at 50 hertz was the standard. Great, I thought. I've lugged a 15-pound boat anchor halfway around the world and won't be able to use it. She then told me that the hotel would be happy to supply me with a transformer if I needed one.

When the bellman brought the transformer to my room, I immediately hooked it up. I had the laptop up and running in no time. The transformer itself must have weighed almost 20 pounds. It was enclosed in a large brown wooden box that had obviously seen better days. In fact, it looked like something out of

a 1950s-style science-fiction movie. Nonetheless, it worked fine during my stay at the hotel.

Two and a half days later I found myself again on a plane heading west, back to New Jersey. I also discovered that security is much tighter at foreign airports compared to domestic ones. At both Narita Airport in Tokyo and Aquino Airport in Manila, I had to pass through several security checkpoints before entering the gate area. These included the security procedures customary in the United States, as well as being frisked by a security officer and having to be checked by another guard with a hand-held metal detector. My carry-on bags were also carefully inspected and I was asked to turn on the laptop computer.

All the people I met in Manila were friendly and polite. In addition to their native Filipino language, most of the residents also speak English. The standard of living there is much lower than what we are used to in the United States. Most of us don't realize how much we have to be thankful for until we see how other people live in another part of the world.

Well, it's now Sunday afternoon. It is raining in this part of New Jersey and

WELL, IT'S NOW
SUNDAY AFTERNOON.
IT IS RAINING
IN THIS PART OF
NEW JERSEY AND
KRAFTWERK
IS PLAYING ON
THE CD PLAYER.
I'M TIRED, BUT
GLAD TO BE HOME.

Kraftwerk is playing on the CD player.
I'm tired, but glad to be home.

Déjà Vu

In April 1985 I started writing *The End User* column in *ANALOG Computing*. For the second half of 1986, I covered both the 8-bit and ST computers. When *ST-LOG* became a separate monthly magazine in January 1987, I began writing the *ST User* column in that magazine. For about the last three years, *The End User* has focused on the Atari 8-bit computers and computing in general, while *ST User* has focused (understandably) on the Atari ST hardware, peripherals and software.

Now *ST-LOG* and *ANALOG* have been combined. Once again, *The End User* will cover both the 8-bit and ST computers and the software that is available for them. In addition, I will still keep you abreast of what is happening in the computer industry as a whole—and especially the Atari community.

I will continue to bring you news, information, application tips, short reviews of useful or significant products and whatever else seems of interest to me or you. My goal is to continue to make this column a place where you can find information on how to get the best from your Atari computer (whichever one you have), regardless of how experienced you are or how much equipment you have.

As always, I welcome your input on what I have discussed or what you would like to see covered in the coming months. You can contact me electronically on DELPHI (ARTL) or CompuServe

(71266,46). Other correspondence (U.S. mail) should be sent to: *The End User*, *ANALOG Computing*, P.O. Box 1413-M.O., Manchester, CT 06040-1413. If you would like a reply, please include a self-addressed, stamped envelope.

ST TV

There is a rather good computer show on television that I try to catch whenever I can. It's called *Computer Chronicles* and in New Jersey is broadcast (irregularly) on public station WNYC out of New York City. The program is hosted by Gary Kildall and Stewart Cheifet and usually contains news, information and short reviews concerning computers and software. Imagine my surprise when, on a recent show, the entire half hour was devoted to the Atari ST.

This particular show consisted of four segments in which ST software and applications were demonstrated. Naturally, the MIDI interface was discussed. In this segment, Passport Design's *Mastertrack Pro* software was shown being used by a university music studio. The ST and this particular software were described as new tools that have revolutionized music education.

Next, *Cyber Paint* was demonstrated by its author, Jim Kent. Jim stated that he chose the ST as the platform for his software because of its color capabilities and fast 68000 processor. When asked to compare the ST to the Amiga, Jim also said that the ST was more reliable, had similar color capabilities and was cheaper.

A local Berkeley computer store specialist, Vince De Felippo, then described and demonstrated the Spectre 128 Macintosh emulator. Vince demonstrated several Mac applications, including HyperCard, and stated that these programs run about 20% faster on the ST. Vince also mentioned PC Ditto, which can run MS-DOS programs.

The following segment featured the Steinhart Aquarium in San Francisco. Here, the ST is used to help dolphins and seals communicate with humans. Marine researcher Eric Carlson said he writes GFA BASIC programs that allow the ST to control the experiments with these marine mammals. Eric also stated that he likes the sound-generating and graphics capabilities of the computer.

The final segment featured demonstrations of *Calamus* and *DynaCADD*. Using a Mega 4 with a Moniterm Monitor,

30-megabyte hard disk and SLM804 laser printer, president of ISD Marketing Nathan Potechin demonstrated *Calamus*. He described it as an entry-level yet professional desktop-publishing program with over 350 features. He showed how it is truly a WYSIWYG (what you see is what you get) program by holding the printed output of his demo next to the screen.

When asked to compare *DynaCADD* to the industry standard, *AutoCADD*, Potechin said that it has a better interface, runs faster and would be equivalent to some future version of *AutoCADD*. Gary Kildall wrapped up the show by saying that the above-mentioned \$3,995 system was less than you might pay for a laser printer alone.

Although many ST users would not find much new information presented in this show, it is heartening to see the ST getting some widespread exposure. In the absence (so far) of any major national advertising by Atari, this kind of media coverage may help spread the good word about the Atari ST.

As they say, check your local listings for time and availability of *Computer Chronicles*. Chances are it will be aired by your local Public Broadcasting System (PBS) station. Atari computers are rarely mentioned, but viewing the show is a good way to keep up with what's happening in the computer industry.

Hand-Held Wonder

I generally use *WordPerfect* for word processing on my ST. As you probably know, it is a powerful program that has just about any feature you'd want. That includes a spelling checker and a thesaurus. However, I usually edit my articles and columns from a hard-copy output after a piece is written.

Being away from the computer, I obviously cannot take advantage of *WordPerfect's* features. Still, I often need to check the spelling of a particular word or rephrase a sentence. In addition, I typically need to find another word that has just the right nuance or a subtle variation in meaning. So I decided to get a pocket spelling checker and thesaurus.

There are a number of hand-held spelling checkers currently on the market from a variety of vendors. Some have more features than others and they range in price from about \$50 to over \$300. I looked at a number of them and settled on one that I thought was particularly good. It is the

WordFinder 220, made by SelecTronics of Minneapolis, Minnesota.

The six-ounce WordFinder has a tiny QWERTY keyboard and measures 4.9" × 3" × 0.8". It has a one-line, 20-character LCD display, contains a 100,000-word spelling dictionary and a 220,000-word thesaurus. It can also help you with jumble-type games, crossword puzzles and Scrabble.

One problem with using a traditional dictionary to check the correct spelling of a word is that you first must have a pretty good guess at how the word is spelled in order to find it. This problem is avoided with the WordFinder because it allows you to enter words phonetically. For example, you could type "crocodial", and it will respond with the correct spelling, "crocodile." To check the spelling of any word, you just type it in and press the "spell" key.

If the WordFinder can't determine exactly what you typed, it will display its best guess and a flashing Right-arrow on the screen. The arrow tells you that WordFinder has other alternatives for your word. If its first attempt was wrong, you can then scroll through these other words until you find the correct spelling of your original word. Sometimes the WordFinder simply cannot recognize the word you typed because either it is so misspelled (occasionally) or is

just not in its memory (infrequently). When this happens, the screen displays "unknown."

The synonym function works quickly and intuitively and almost makes the task of finding alternate words fun. To find a synonym, you just press the "synon" key after you enter the word. At that point, the WordFinder will display its first synonym. However, when you type in a word, the WordFinder has no way of knowing which of the many possible meanings a word can have. Fortunately, it has an elegant solution to this.

For example, say I wanted to find a synonym for "tease." Wordfinder's first synonym is "seductress," which it displays on the screen. Also displayed are a Down-arrow and Right-arrow indicator. "Seductress" is considered a "concept" word. Other words with similar meaning can be viewed by pressing the right-arrow key. In this example, "temptress," "coquette," "harlot," "vamp," etc., are words that are related to one particular meaning of "tease."

Pressing the down-arrow key reveals "mock," "sneer," "toy," and "annoy"—additional concept words that represent other possible meanings for "tease." When each of these concept words is displayed, you can scroll through various synonyms

with that central meaning. For example, I was originally looking for a word to describe "making fun of someone." But I wanted a word that had a little softer implication. Pointing to "mock," some of my alternatives were "joke," "needle" and "rib." "Rib" was just the word I wanted.

The synonym function can also be used to determine the meaning of a word. This is done just by reviewing the concept words that the WordFinder finds for a particular word. You can even find synonyms for synonyms. Proper names, cities, states, countries and corporate names are also included, complete with capitalization.

Microlytics developed the linguistic technology that allows a 9:1 word compression ratio for storing several megabytes of information on the single 256K memory chip used in the WordFinder. SelecTronics sells the WordFinder 220, under license from Microlytics, for \$99. If you are interested, SelecTronics can be reached at 701 Decatur Avenue North, Building 204, Minneapolis, MN 55427. Their phone number is (612) 545-6823. **A**

Arthur Leyenberger is a freelance writer who lives in beautiful New Jersey. He can be reached on CompuServe at 71266,46 or on DELPHI as ARTL.

"The most useful program for the Atari since Print Shop!"

FORMS GENERATOR

for the Atari 800, 800XL, 65XE, 130XE

Designed by Jeff Brenner, columnist for *Computer Shopper* magazine, of "Applying The Atari" fame, and author of book and magazine articles in *COMPUTE!*, *ANALOG* and others.

LOOK WHAT YOU CAN DO WITH FORMS GENERATOR:

Purchase merchandise by mail? Next time, send a customized **purchase order form**! Does your home or business ever need **statements, invoices, proposals, job work orders, gift certificates**, etc.? No problem! Use **FORMS GENERATOR's scrolling spreadsheet-style screen** to design almost any form to suit **your exact needs**. What you see on-screen is what you get on paper! Use the text mode with any 80-column printer, or the high-res graphics mode with the Epson, Gemini/Star, Okidata, Panasonic or Prowriter for **remarkably realistic forms**. BUT THAT'S JUST THE BEGINNING: Once you've designed a form, you can program **FORMS GENERATOR to make all calculations automatically!** Imagine: after you enter quantities, descriptions and prices, **FORMS GENERATOR** moves about the form calculating extended prices, subtotals, and even the sales tax! Like magic! (Sample invoices included). You can also use **FORMS GENERATOR** for record keeping, since you can save filled forms to disk!

NAMED A "BEST BUY" IN 8-BIT SOFTWARE BY ANTIC MAGAZINE, JANUARY 1988.

Our "down to planet Earth" price: Only \$23.95 (product #ATA611).

FOR C.O.D. ORDERS CALL (516) 932-5330

Send coupon to:

Twenty-Fifth
Century™

Software Division

Dept. AT 2

234 Fifth Avenue
Suite 301
New York, N.Y. 10001

☐ YES! Please rush me **FORMS GENERATOR** (product #ATA611) with complete documentation, 90-day free replacement warranty, full customer service support and 20-page Atari software catalog. I am enclosing \$23.95 + \$3.50 shipping and handling.

☐ Check/Money Order enclosed ☐ C.O.D. (add \$2.50)

Name _____

Address _____

City _____

State _____

Zip _____

DEALER INQUIRIES INVITED.

New York State residents add 8% sales tax.

*The Print Shop and Atari are registered trademarks of Broderbund Software and Atari Corp., respectively. — Prices and availability subject to change without notice.

CIRCLE #108 ON READER SERVICE CARD.

YOUR SOFTWARE RENTAL CENTER

THIS IS THE PLACE

Life on easy street means picking up your phone and ordering the software of your choice. We carry software for Atari, IBM, Apple, 520ST, MAC, Amiga & Commodore.

Call us now, toll free, for a complete list.

Call toll-free outside Texas: 1-800-433-2938

— Inside Texas call: 817-292-7396

WEDGWOOD RENTAL

5316 Woodway Drive
Fort Worth, Texas 76133



CIRCLE #107 ON READER SERVICE CARD.

PD PARADE

ST

by George L. Smyth

About five years ago I taught programming and was in charge of the school's AlphaMicro minicomputer. As part of my job I attended a conference where one of the main topics centered around the potential proliferation of a self-replicating program called a virus. The speaker explained how its possible threats, including the alteration of high-level security programs, had been demonstrated on the Pentagon's computer by a person with the lowest level of access. I was alerted to the security needs of my company's computer, but I never thought that this type of problem would filter down to the microcomputer level.

When downloading a program from a bulletin board or merely exchanging disks with a friend, how are you to know if the program is virus-free? Precautions can be taken, such as installing programs that restrict write activities to specified drives, keeping the write-protect window open, and booting the system with a newly formatted disk, but viruses can take shapes that are subtle or delayed, making them difficult to detect.

The first point of assault against computer viruses is to gain knowledge—the ability to recognize the existence of a virus; the second is to gain control—the ability to destroy it. At the vanguard of this attack is George R. Woodside, who has written two programs to help us recognize virus symptoms and remove them.

FLU.PRГ is a program that demonstrates the symptoms of several viruses known to exist on programs written for the Atari ST series of computers. These examples include symptoms that attack the sound chip and display, not those that are destructive to the disk. By clicking on one of the selector boxes, the chosen virus symptom is demonstrated. The delayed and/or slow-infesting routines are sped up to make the demonstration more visible. The different examples are:

- **Blot:** After the system has been on for three hours, the screen is slowly blanked out, one pixel at a time.

- **Screen:** Similar to the blot virus, this one works more quickly and is invoked after only 30 minutes.

- **Green goblin:** Every 16th disk access causes a portion of the screen to be flipped upside-down; after 128 accesses

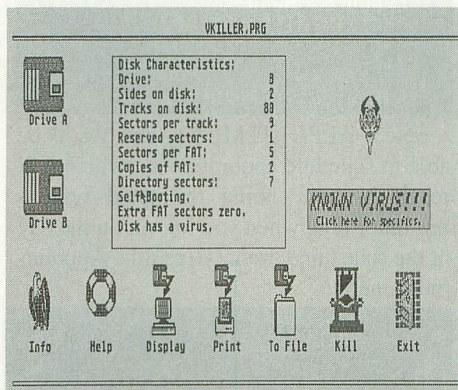
the virus's "signature" is displayed.

- **Label:** Every tenth disk access causes the screen to completely blank out.

- **Mad:** This category involves eight different attacks, including reverse video, horizontal and vertical scrolling, and scrambling.

- **Maulwurf:** After a specified period of time, the perpetrator's signature is displayed at the cursor position.

- **Mouse inversion:** The vertical orientation of the mouse is inverted after a certain number of new disks are detected,



i.e., pushing the mouse forward moves the screen pointer down.

- **Pirate trap:** Possibly installed by a software vendor or developer, this virus copies itself to other disks and every 50th copy displays a message claiming that the user is being watched.

So what can we do about this? How can we know if a newly downloaded program is virus-free? What can we do if it is infected? There are no guarantees, but VKILLER.PRГ is able to recognize the current crop of viruses and erase them from the disk.

Viruses inhabit the boot sector of a disk. The boot sector allows a program to be automatically executed by resetting the computer or powering it off and then on, a scheme employed by many software developers within copy-protection devices. This type of disk cannot be checked, since VKILLER.PRГ searches this boot sector for the presence of any information. There is no way to tell if the information found in an executable boot sector is a virus or represents legitimate instructions. Simply stated, if a program requires a system reset or shutdown in lieu of double-clicking its icon to be run, VKILLER.PRГ should not be used to check the disk.

The existence of a virus is searched by selecting the A or B drive icon. The pro-

gram displays a happy face if the examined disk appears to be *clear of viruses*. A devil's face appears if the boot sector contains executable code, characteristic of a virus. If the disk does not contain an executable boot sector, but instead contains another form of data where the disk should be blank, a question mark appears. A virus warning box is displayed if the characteristics of a known virus are detected.

If the user detects a possible virus, the "Kill" icon should be selected. This operation erases the unused portions of the boot sector and FATs. Mr. Woodside states that all currently known viruses can be eliminated in this manner. But, as mentioned above, if the disk has a legitimate use for the executable boot sector, the routine will not recognize this and will "Kill" the disk just as if it contained a virus, rendering it inoperable.

The program has the ability to detect the presence of a known virus in the system while the program is running. If this happens, instructions are displayed as to the action which should be taken.

The program displays information about the disk, such as the number of reserved sectors, sectors per FAT, directory sectors and other information.

The user can also send the information contained in the boot sector and FATs, the binary values and their ASCII equivalents, to the screen, disk or printer. This is useful to the knowledgeable individual who wishes to learn more about virus placement on the disk, or to the novice who merely wants to learn more about the software structure of Atari's disk format.

To date, 15 known virus types can be identified by this program. Mr. Woodside urges anyone finding a virus that this program cannot detect to send him a copy so that an update can be implemented. This is an excellent way to support your friendly local public-domain programmer. ☐



George L. Smyth holds a degree in psychology from West Virginia University and is currently employed as a programmer. He is the author of a series of tutorials on programming in Forth.

DATA BASE DELPHI



by Michael A. Banks

(If all the foregoing is "Greek" to you, type FORUM at the ATARI or ST SIG main menu and see what you've been missing! The SIG Forums are your main Atari information channels.)

New Forum Feature

If you use the Forum a lot, you'll want to know about a recent change that adds a lot of power to the DIR and READ commands.

As you know, you can add a number of qualifiers to either DIR or READ to specify the exact message(s) you're seeking. For example, typing DIR TO KZIN FROM ANALOG4 will display a list of all messages posted to user KZIN by (from) ANALOG4.

You can now use AND/OR logic to further define messages to scan with DIR or read with READ. If you type DIR TO KZIN AND FROM ANALOG4, it is the same as typing DIR TO KZIN FROM ANALOG4, but using OR is a different matter. If you type DIR TO KZIN OR FROM ANALOG4, you will see a listing of all messages that are addressed to KZIN, as well as all messages posted by (from) ANALOG4.

There is one limitation to all this: You cannot use AND and OR together in the same command.

If you're a bit confused, just remember this basic rule of thumb: Using AND (or no qualifier) decreases the number of messages selected, while using OR increases the number of messages selected. (It's the same as using NARROW and WIDEN in the databases.)

Your Personal Online Schedule and Notepad!

Want to try something really different—something that's not even covered in the DELPHI manual (yet)? From the Conference menu, or from within a Conference group, type /DIARY. This leads to DELPHI's SCHEDULER feature—actually an appointment calendar and diary/notepad.

After you type /DIARY, type "?", and you'll see this menu:

```
SCHEDULER Menu:
APPOINTMENT-CALENDAR
DIARY
EXIT
```

```
SCHEDULER> (APPOINTMENT, DIARY)
```

Select APPOINTMENT, and you'll be able to schedule appointments, which will result in your being prompted with a reminder note when you log-on on the day of the scheduled event. Here's the Appointment menu:

```
SCHEDULER APPOINTMENT-CALENDAR Menu:
ADD
DELETE
EXIT
HELP
LIST
```

```
APPT> (Add, Delete, List)
```

The features on this menu allow you to add or delete events, and to list all events.

Select DIARY, and you'll see this menu:

```
SCHEDULER DIARY Menu:
```

```
ADD
DELETE
LIST
HELP
EXIT
```

```
DIARY> Add, Delete or List? list
```

This menu allows you to enter notes, which nobody else can see.

When you type /DIARY in a Conference group, it is like typing /MAIL in the sense that only 1,000 characters of what is typed while you are gone will be saved for redisplay when you return to Conference.

Whenever you are prompted for a date while using the Appointment or Diary feature, enter it using the format MM/DD/YY (Example: 03/21/90). If you don't enter a date, the current date will be used. (Note: When you view a date's schedule or note with the LIST command, you can step through entries for all dates following that date by merely pressing Return at the NEXT> prompt that appears.)

To exit the Diary (or to move up through its menu levels), type EXIT or enter Control-Z.

Need Help?

For help with the Forum, type DA GENERAL at the SIG main menu; then type READ KZIN1 to see a complete tutorial on using the SIG Forum message base. (And look for more tutorials in the near future!)

TQ Tournaments Pay Off!

DELPHI's TQ Trivia Tournaments are still going strong. Fred Boughter (FREDHB) was the TQ Trivia Tournament champion for May, winning the month's grand prize, a desk reference set valued at over \$200, provided by Grolier. Lots of other prizes, including free time, are offered in TQ. Type GO ENT TQ at the ATARI or ST SIG main menus for more info.

More Prizes

DELPHI's newest online real-time game, *FlipIt!*, also features tournaments. Participation in Monday-night games builds a jackpot, which is divided among the top scorers in each of three categories: best on 8-by-8 board, best on 10-by-10 board and best overall. Type GO ENT FLIP at the SIG main menu for more info.

That's it for now. See you in Conference! (Tuesday evening, 10:00 p.m., Eastern time; be there, or be an obtuse rectangle!)

In addition to science fiction novels and books on model rocketry and other topics, Michael A. Banks is the author of DELPHI: The Official Guide and The Modem Reference, both from Brady Books. You can write to him via E-mail on DELPHI to membername KZIN. ☐

DATABASE DELPHI

Make the DELPHI connection

As a reader of ST-LOG, you are entitled to take advantage of a special DELPHI membership offer. For only \$19.95, plus shipping and handling (\$30 off the standard membership price!), you will receive a lifetime subscription to DELPHI, a copy of the 500-page *DELPHI: The Official Guide* by Michael A. Banks, and a credit equal to one free evening hour at standard connect rates. Almost anyone worldwide can access DELPHI (using Tymnet, Telenet or other networking services) via a local telephone call.

To Join DELPHI

1. Dial 617-576-0862 with any terminal or PC and modem (at 2400 bps, dial 576-2981).
2. At the Username prompt, type JOINDELPHI.
3. At the Password prompt, enter STLOG.

For more information, call DELPHI Member Services at 1-800-544-4005, or at 617-491-3393 from within Massachusetts or from outside the U.S.

DELPHI is a service of General Videotex Corporation of Cambridge, Massachusetts.

ATTENTION PROGRAMMERS !

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

All submissions for publication, 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 **ANALOG Computing**, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of **ANALOG Computing**. If not accepted for publication, the articles and/or programs will remain the property of the author. If submissions are to be returned, please supply a self-addressed, stamped envelope. All submissions of any kind must be accompanied by the author's full address and telephone number.

Send your programs and articles to:

ANALOG COMPUTING

P.O. BOX 1413-M.O., Manchester, CT 06040-1413

ATARIWRITER 80

- 80 COLUMN EDITING
- EDITABLE PRINTER DRIVER
- 30,000 WORD DICTIONARY
- MAIL MERGE
- ON 5 1/4 DISKETTE

\$49**ATARI XEP 80**

- CONNECTS TO ANY COMPOSITE MONITOR
- 80 COL. DISPLAY
- PRINTER INTERFACE

\$79**ATARI 80 Col. PACKAGE**

- 12" MONO MONITOR
- ATARI XEP 80
- ATARIWRITER 80

\$179**EVEN BETTER W/ Printer**

- 12" MONO MONITOR
- ATARIWRITER 80
- ATARI XEP 80
- 1025 PRINTER

\$229**1020 COLOR PRINTER PLOTTER**

COMPLETE WITH:

- 2 PEN SETS
- 1 ROLL PAPER
- POWER SUPPLY & CABLE

\$14.95

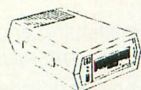
BRAND NEW

EXTRA PEN SETSCOLOR \$3.98
BLACK \$8.99**1200XL 64K COMPUTER****\$49** 256 COLORS
XEGS COMPATIBLE **\$69**

RECONDITIONED FREE PAC-MAN FACTORY NEW!

1027 PRINTERS 1025LETTER QTY 1027 IS FACTORY NEW! DOT MATRIX
\$79 FOR YOUR 8-BIT **\$69**
1025 IS RECONDITIONED**HARDWARE & ACCESSORIES**

850 PRINTER INTERFACE	\$79.95
1010 PROGRAM RECORDER	\$29.95
ATARI TWO JOYSTICK SET	\$9.95
SPACE AGE JOYSTICK	\$14.95
WICO COMMAND JOYSTICK	\$19.95
WICO THE BOSS JOYSTICK	\$19.95
ATARI TRACKBALLS	\$9.95
ATARI SX212 MODEM	\$89.95
AVATEX 1200HC MODEM	\$89.95
DOS 2.5 W/ MANUAL	\$4.95
600XL (NO Transformer)	\$19.95
400,800,850,1200XL Transformer	\$14.95
XL/XE Transformer	\$19.95

COMPLETE WITH:
POWER SUPPLY
I/O CABLE
DOS 2.5 W/
MANUAL**DISK DRIVES FOR 800/XL/XE**ATARI 810 **\$129.00**ATARI 1050 **\$169.00**

RECONDITIONED

**THE BEST LIGHT GUN**

FOR YOUR 800/XL/XE

\$35OR THE PACKAGE **\$109**LIGHT GUN
CRIME BUSTERS
CROSSBOW
BARNYARD BLASTER**THE NEWSROOM**Desktop Publisher By
SPRINGBOARD **\$15****CARTRIDGES FOR THE 800/XL/XE**

BASIC CARTRIDGE	\$4.95
BASIC TUTOR (2 BOOKS)	\$4.95
QIX	\$4.95
TURMOIL	\$4.95
PAC-MAN (no box)	\$4.95
DONKEY KONG (no box)	\$4.95
GORP (400,800)	\$4.95
CHICKEN	\$4.95
SLIME (400,800)	\$4.95
CLAIM JUMPER	\$4.95
DELUXE INVADERS	\$4.95
JOURNEY TO THE PLANETS	\$4.95
STAR RAIDERS	\$4.95
MISSILE COMMAND	\$4.95
GALAXIAN	\$4.95
DEFENDER	\$4.95

ET	\$4.95
FACE MAKER	\$4.95
MATH ENCOUNTER	\$7.95
DANCE FANTASY	\$8.95
LOGIC LEVELS	\$8.95
MEMORY MANOR	\$8.95
LINKING LOGIC	\$8.95
DELTA DRAWING	\$9.95
HEY DIDDLE DIDDLE	\$9.95
GRANDMA'S HOUSE	\$9.95
FRACTION FEVER	\$9.95
ALPHABET ZOO	\$9.95
ALF	\$9.95
DIG DUG	\$14.95
MILLIPEDE	\$14.95
SKY WRITER	\$14.95
FOOTBALL	\$14.95

ROBOTRON	\$19.95
TENNIS	\$19.95
FINAL LEGACY	\$19.95
MARIO BROS	\$19.95
DONKEY KONG JR	\$19.95
JUNGLE HUNT	\$19.95
MOON PATROL	\$19.95
BATTLEZONE	\$19.95
FOOD FIGHT	\$19.95
HARDBALL	\$19.95
FIGHT NIGHT	\$19.95
ONE ON ONE BASKETBALL	\$19.95
DESERT FALCON	\$19.95
NECROMANCER	\$19.95
RESCUE ON FRACTALUS	\$19.95
BALLBLAZER	\$19.95
BLUE MAX	\$19.95

STAR RAIDERS II	\$19.95
DAVID'S MIDNIGHT MAGIC	\$19.95
ARCHON	\$19.95
KARATEKA	\$19.95
CHOPLIFTER	\$19.95
GATO	\$24.95
ACE OF ACES	\$24.95
LODE RUNNER	\$24.95
BARNYARD BLASTER (LG)	\$24.95
DARK CHAMBERS	\$29.95
AIRBALL	\$29.95
SUMMER GAMES	\$29.95
CROSSBOW (LG)	\$29.95
EAGLES NEST	\$29.95
CRIME BUSTERS (LG)	\$29.95
MICROFILM (database)	\$39.95

DAVID'S MIDNIGHT MAGIC	\$4.95
REPTON	\$4.95
BANDITS (48K 400,800)	\$4.95
CLAIM JUMPER	\$4.95
TIME WISE	\$4.95
DECISION IN THE DESERT	\$4.95
CRUSADE IN EUROPE	\$4.95
KENNEDY APPROACH	\$4.95
CONFLICT IN VIETNAM	\$4.95
NATO COMMANDER	\$4.95
CROSSCHECK	\$4.95
MISSION ASTEROID	\$4.95

DISK SOFTWARE FOR THE 800/XL/XE

COMBAT	\$4.95
PREPPIE I	\$4.95
PREPPIE II	\$4.95
THE COUNT	\$4.95
FREAKY FACTORY	\$4.95
LASER HAWK	\$4.95
CRYSTAL RAIDERS	\$4.95
ADVENTURELAND	\$4.95

PIRATE ADVENTURE	\$4.95
SECRET MISSION	\$4.95
VOODOO CASTLE	\$4.95
STRANGE ODYSSEY	\$4.95
REPTON	\$4.95
HULK	\$4.95
ALIEN AMBUSH	\$4.95
GATEWAY TO APASHAI	\$7.95

JAWBREAKER	\$7.95
TOP GUNNER COLLECTION (3 GAMES)	\$9.95
SILENT SERVICE	\$9.95
DISPATCH RIDER	\$9.95
SILICON DREAMS	\$9.95
F-15 STRIKE EAGLE	\$9.95
THE NEWSROOM(1050-64K)	\$15.00
VISCALC	\$29.95
BOOKKEEPER	
W/ num keypad	\$29.95
HOME ACCOUNTANT	\$29.95

1010 PROGRAM RECORDER

SAVE AND LOAD YOUR OWN PROGRAMS

\$29.95

RECONDITIONED

LIGHT SPEED C DEVELOPMENT KIT**\$35.95****COMPOSITE MONO MONITORS****\$59****NEW ARRIVALS**

DECISION IN THE DESERT	\$4.95
CRUSADE IN EUROPE	\$4.95
KENNEDY APPROACH	\$4.95
CONFLICT IN VIETNAM	\$4.95
NATO COMMANDER	\$4.95
F-15 STRIKE EAGLE	\$9.95

WICO JOYSTICK SALE!

COMMAND CONTROL 3-WAY	\$14.95
BLACK MAX	\$9.95

ASTRA 2001 DUAL DRIVE**\$199**DOUBLE DENSITY
DUAL DISK DRIVE
(2 DRIVES IN ONE CASE)

MAC EMULATOR FOR YOUR ST

MAGIC SAC PLUS	\$59.95
SPECTRE 128 V1.9	\$169.95
MAC ROMS 64K	\$19.95
MAC ROMS 128K	\$129.95



SONY DISK SPECIAL

3.5 BULK PACKS WITH LABELS
1 - \$99 25 - \$24.75 1000 - \$750.00

ST COLOR MONITORS

3 TO CHOOSE
FROM!

\$299

MEMORY UPGRADES

TECH SPECIALTIES	2/4Meg	\$149.95
NEW WORLD UPGRADE	2/4Meg	\$139.95
ARCO BOARD	1/2/4Meg	\$169.95

ST TURBO SPEED!!

J.A.T.O. BOARD 8/16Mhz	\$99
CMI PROCESSOR ACCELERATOR	\$299
Math Co-processor for CMI 12Mhz	\$99

STAR PRINTERS

STAR NX1000	9 Pin	\$199
STAR NX1000 Rainbow	9 Pin	\$299
STAR NX2400 4 Fonts	24 Pin	\$399

HARDWARE ITEMS

Mega 4ST Computer	\$1899
JRI GENLOCK SYSTEM	\$650
RC AEROCHOPPER	
RC Flight Simul. w/Futaba control	\$199
ST/IME MODULE	
Battery Backed up Clock	\$59
IMG SCAN Printer Scanner	\$89

FAX US YOUR ORDER! FOR FASTER SERVICE PLEASE INCLUDE:

• NAME	VISA / MASTERCARD
• BILL TO ADDRESS	ORDERS ONLY
• SHIP TO ADDRESS	
• PHONE #	CARD ADDRESS MUST MATCH
• ITEM(S) YOU WISH TO PURCHASE	BILL TO ADDRESS

(408) 224-8574

1200XL PACKAGE \$99

- 64K 8-bit COMPUTER
- 1020 COLOR PRINTER PLOTTER
- 1025 DOT MATRIX PRINTER (80 COL)
- PAC-MAN CARTRIDGE
- BASIC WITH TUTOR SET

Give the kids something else to play with besides your ST!

HIT DISK PACKAGE ST

- SLAGON
- GOLDRUNNER
- JUPITER PROBE
- KARATE KID PART II

\$29.95

ST HARD DRIVE SAVINGS

ATARI MEGAFILE 30	\$599
AWESOME 60+ (60Meg 28ms)	\$899
AWESOME 80+ (80Meg 28ms)	\$999

PCBoard Designer \$49

Reg. \$199.95

DESIGN YOUR OWN PC BOARDS!!

THE BEST MOUSE

REPLACEMENT MOUSE FOR
THE ST!



\$49

RAINBIRD TITLES ST

Knight Orc	\$9.95
Tracker	\$9.95
Advanced Art Studio	\$9.95
Guild of Thieves	\$9.95
The Pawn	\$9.95
Golden Path	\$7.95
Universal Military	
Simulator	\$14.95
Carrier Command	\$29.95
Jinxter	\$9.95
Sentry	\$9.95
StarGlider	\$9.95

ALL ITEMS REG. \$39.95 - \$49.95

SPECIAL SOFTWARE SALE ST

HIT DISK (4 games)	Reg \$59.95	NOW! \$29.95
GOLDRUNNER II	Reg \$39.95	NOW! \$19.95
GRAIL	Reg \$19.95	NOW! \$12.95
INVASION	Reg \$29.95	NOW! \$19.95
SOCCER	Reg \$39.95	NOW! \$19.95
ZERO GRAVITY	Reg \$29.95	NOW! \$19.95
TETRA QUEST	Reg \$29.95	NOW! \$19.95
TEXT PRO	Reg \$49.95	NOW! \$19.95
PAINT PRO	Reg \$49.95	NOW! \$19.95
DATA RETRIEVE	Reg \$49.95	NOW! \$19.95
H & D BASE	Reg \$99.95	NOW! \$19.95

SONUS ST SPECIALS

SST SEQUENCER V2.0	\$99.95
SONUS MIDI PORT B	\$69.95
SONUS MASTER PIECE	\$249.95
SONUS SUPER SCORE	\$229.95

SAN JOSE COMPUTER

T H E A T A R I S T O R E

640 BLOSSOM HILL RD. SAN JOSE, CA 95037

STORE (408) 224-8575 • BBS (408) 224-9052 • FAX (408) 224-8574

SHIPPING: ADD \$5.00 TO ALL ORDERS. AIR AND INTERNATIONAL SHIPPING EXTRA. THAT'S IT.

WARRANTY: 90 DAY WARRANTY ON ALL ITEMS. TAX: CALIFORNIA RESIDENTS ADD 7% SALES TAX.

PREPAYMENT: USE VISA, MASTERCARD, MONEY ORDER, CASHIER'S CHECK OR PERSONAL CHECK. PERSONAL CHECK MUST CLEAR PRIOR TO SHIPMENT. C.O.D.: CASH, CASHIER'S CHECK OR M.O. ONLY.

Prices subject to change without notice.

Brand and/or product names are trademarks or registered trademarks of their respective holders.

Ad produced on an ATARI ST using CALAMUS and printed on an ATARI SL M804 PostScript compatible laser printer.

(CONTINUED FROM PAGE 7)

We then open the file in binary-write mode, and a call to `write_new_info()` writes all the account information to the file, as well as creates the monthly data files.

A call to `open_acct()` actually opens the account so the user can enter checks if he wishes. (Note that `open_acct()` is represented only by a stub at this point; so when you run this month's portion of the program, although you can create a new account, the account cannot actually be opened.)

If the user exited the dialog by clicking on the CANCEL button (FILECANC), we exit the dialog without starting a new account.

Function `clear_newacct()`

This function simply steps through each of the editable text fields in the new-account dialog box, clearing them of whatever information they may already contain.

Function `get_tedinfo_str()`

In order to handle dialog boxes with editable text fields, it's necessary you know how to handle TEDINFO structures (see the *C-manship* in the May '87 ST-LOG). This function simplifies the process of getting the address of a string in an editable text field. Simply pass it the address of the tree and the number of the object, and it'll return the address of the string.

Function `check_file()`

In order to open a file for a new account, we must, of course, have a legal filename. The validation string for the dialog's editable text field will make sure the user enters only legal filename characters, but it's up to us to make sure that the field is filled in. This function simply checks the object FILENAME to be sure it's not empty. If it is empty, we scold the user with an alert box.

Function `no_decimal()`

In *MicroCheck ST*, we have to deal with numbers in two ways. The user, since he is entering amounts as dollars and cents, needs to use decimal numbers; that is, he needs to enter his values with dollars on the left of a decimal point and cents on the right. Unfortunately, this means using floating point numbers, which are infamously inaccurate due to the rounding operations performed when calculating with these numbers.

In order to avoid rounding errors, *MicroCheck ST* calculates not with both dollars and cents, but only with cents. For *MicroCheck*, \$100 isn't 100.00, but rather 10,000. So every time the user enters a value, we need, as the first step to making the

conversion from dollars to cents, to combine both the dollar and cent portions of the number.

The function `no_decimal()` accomplishes that task using conventional string-handling techniques. The string to be converted will have one of three formats: dollars and no cents, dollars and cents, or cents and no dollars. The strings, as retrieved from the dialog, will take one of the forms shown in the first column of the chart below (none of the strings contains trailing spaces):

String	Actual value	After <code>no_decimal()</code>
99	\$ 99.88	9988
99 99	99.99	9999
99999999	99999.99	99999999
9	.98	98
99	.99	99

The strings, when they are retrieved from a dialog box, contain no decimal points; instead, dollars and cents are separated by spaces. If the entire field is full, there are no spaces at all. Without spaces delimiting the two numbers, you have to know the field length in order to separate the dollars and cents; in this case, dollars can be up to five digits, and cents, of course, can be up to two.

The chart above shows the string obtained from the dialog, the value it represents and the value that `no_decimal()` will return, respectively. Note that the value returned is the number of cents, but it's still in string—not numeric—form.

Function `str_to_long()`

Once we have the number of cents in string form, we need to convert it to long-integer form. The function `str_to_long()` handles this by multiplying each digit in the string times its corresponding value in the `pwr[]` array. Each product is added to `num` until the end of the string is reached. The long integer `num` will contain the final result.

Function `write_new_info()`

In order to begin a new account, *MicroCheck ST* must write out the information retrieved from the new-account dialog to the .MCK file, as well as create each of the 13 monthly data files. These files will be named file0.DAT through file12.DAT, where "file" is the filename that was entered into the filename dialog box.

All this is accomplished by `write_new_info()`. First, we write out the user's name and address to the file we opened back in `newacct_file()` (`acctfile`). Then we retrieve the account balance, convert it to long integer and write it out to the same file.

The last step is to create the 13 monthly data files. (Yes, I know there are only 12 months. The extra file, the one numbered "0," is storage for uncanceled checks from the previous year, necessary whenever the "new year" function is used.)

We use a `for` loop to repeat the file-creation process 13 times. The loop variable, `x`, is used as the file number and is tacked onto the end of the filename the user typed into the filename dialog box. The complete pathname is tacked onto the filename, as is the extension ".DAT." When the filename is complete, we open the file (binary) and write out a zero, after which the file is closed. (The first word [two bytes, that is] of the monthly files is the number of transactions in the file; since a new file has no transactions, we start off with a zero.)

Conclusion

That finishes things up for another month. If anything I've described in this or the past few installments of *C-manship* doesn't make sense to you, you should review the topic in question. If you want to write full-GEM applications, you need to know all this stuff! ☐

LISTINGS CONTINUED ON PAGE 124



Clayton Walnum is the executive editor of ANALOG Computing, as well as the associate editor of VIDEOGAMES & COMPUTER ENTERTAINMENT.

THE JUDGE \$39.95

- Decision making
- Contest judging
- Classroom grading

LOFI \$19.95

- Loan financing calculations

TURTL \$19.95

- Graphics fun for kids 8-80
- Educational

MORSEY \$29.95

- Morse code tutor

Atari 48K 800/800XL/65XE/130XE Disk
(NM Tax 5%, US S&H \$2)
Write for product flyer.

Mead Micro Ware
10 Bonito Pl.
Los Alamos, NM 87544

CIRCLE #126 ON READER SERVICE CARD.

BY TOM HUDSON

BOOT CAMP

It's hard to believe, but here we are in the seventh installment of *Boot Camp*. We've only got a few more 6502 operation codes to cover before we begin writing full-scale programs, so hang in! The best is yet to come.

Old Business

Last issue's assignment asked you to solve eight bit-manipulation problems. You were given "before" and "after" bit patterns and asked to find what operation codes and operands were used to get the results. **Figure 1** shows the completed assignment. Some of the problems had two possible answers. These are so noted, with both solutions.

Clever readers have probably noticed that the fourth problem actually has far more than two possible answers. In fact, by using the ORA instruction, Byte 2 could be any value with bits 1, 3, 5 and 7 set! Try it yourself with a short program.

BYTE 1	OPN	BYTE 2	RESULT	ANS
01000011	AND	01000001	01000001	(1)
01000011	EOR	00000010	01000001	(2)
11001011	EOR	01101001	10100010	
11110000	AND	01000000	01000000	(1)
11110000	EOR	10110000	01000000	(2)
01010101	ORA	10101010	11111111	(1)
01010101	EOR	10101010	11111111	(2)
11001000	EOR	10110100	01111100	
11111111	AND	11110001	11110001	(1)
11111111	EOR	00001110	11110001	(2)
00100100	EOR	10011100	10111000	
01000111	EOR	01010011	00010010	

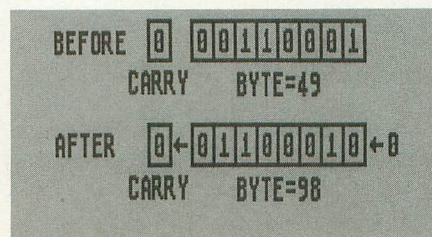
Simple Multiplication

As you may recall, by shifting a binary number left one bit, we effectively multiply it by two. Shifting it left two bits multiplies it by four. This principle is very handy, allowing us to multiply integers quickly and easily.

How do we perform this left-shift operation in 6502 assembly language? With the ASL (Arithmetic Shift Left) instruction, of course. This operation shifts the contents of the accumulator or a selected memory byte left one bit, and has the following formats:

ASL A (ACCUMULATOR)
ASL nn (ABSOLUTE)
ASL n (ZERO PAGE)
ASL n,X (ZERO PAGE INDEXED X)
ASL nn,X (INDEXED X)

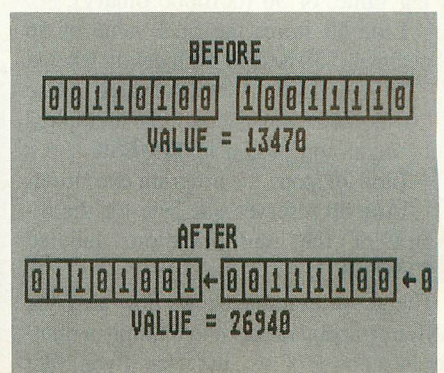
When an ASL instruction is executed, the accumulator or memory byte is shifted one bit to the left. **Figure 2** shows how the operation is handled internally.



As you can see from the "before" and "after" images in **Figure 2**, each bit of the selected byte is shifted to the left one place. Since Bit 8 has no other place to go, it is shifted into the 6502 Carry flag. This is done to allow for multiple-byte shifts, which we'll look at in a moment. A 0 is shifted into the 1 bit. As you can see, the value of the byte has been multiplied by two!

As long as the results of your shift-multiples do not exceed 255 decimal, you will find the ASL instruction works fine. Problems begin, though, when you get into multiple-byte values.

Figure 3 shows an example of a multiple-byte shift. As you can see, the contents of Bit 7 of the low byte must shift into Bit 0 of the high byte. In order to do this, we must see the LSR instruction to shift the low byte, and a new instruction, ROL (Rotate Left through carry), for the high byte. ROL has the following formats:



ROL A ACCUMULATOR)
 ROL nn (ABSOLUTE)
 ROL n (ZERO PAGE)
 ROL n,X (ZERO PAGE INDEXED X)
 ROL nn,X (INDEXED X)

The ROL instruction performs the same function as ASL, except that it puts the contents of the Carry flag in the low-order bit instead of a zero.

Both ASL and ROL set the Sign, Zero and Carry flags according to the result of the operation.

Let's look at a few examples of multiplication using the ASL and ROL instructions.

```
10 *= $0600
20 LDA #507 ;PLACE 7 IN ACCUM.
30 ASL A ;TIMES 2
40 ASL A ;TIMES 4
50 ASL A ;TIMES 8
60 STA TIMES8 ;SAVE RESULT
70 BRK ;AND STOP!
80 TIMES8 *=*+1
90 .END
```

The above shows an example of single-byte multiplication using the ASL instruction. In this example, we're multiplying the contents of the accumulator (7) by eight and storing the result in the location labeled TIMES8.

Line 20 loads the accumulator with the number 7 (00000111 binary). You can try different values here to test the multiply. Remember that since this is only a single-byte multiply, the result cannot exceed 255. Therefore, don't use any values greater than 31 decimal here.

Line 30 shifts the accumulator to the left one bit, multiplying the accumulator by two. After this instruction executes, the accumulator will contain 14 decimal (00011110 binary).

Line 40 shifts the accumulator left another bit. At this point, the accumulator is four times the starting value of 7, or 28 (00011100 binary).

Line 50 shifts the accumulator left a third time, giving us eight times the starting value, or 56 (00111000 binary).

Line 60 stores the final value of 56 decimal (\$38 hex) in the location labeled TIMES8. If you change the value in line 20, the value you enter will be multiplied by eight and placed in TIMES8.

Line 70 stops the program execution.

Line 80 reserves one byte for the result of the multiplication, labeled TIMES8.

The above example shows how easy the ASL instruction makes it to multiply a number by a power of two, but what if you want

to multiply a number by five?

In such cases, it's good to break the multiplier down into "bite-sized" pieces. For example, a multiply by five can be broken down into:

$$\begin{array}{r} \text{(NUMBER * 4)} \\ + \text{(NUMBER * 1)} \\ \hline \text{(NUMBER * 5)} \end{array}$$

The 6502 code required for this operation is shown below.

```
10 *= $0600
15 LDA #23 ;PLACE 23 IN ACCUM.
20 ASL A ;TIMES 2
25 ASL A ;TIMES 4
30 CLC ;CLEAR CARRY FOR ADD
35 ADC #23 ;ADD 23 = TIMES 5!
40 STA TIMES5 ;AND STORE RESULT
45 BRK ;ALL DONE!
50 TIMES5 *=*+1
55 .END
```

Similarly, a multiply by ten can be broken down to:

$$\begin{array}{r} \text{(NUMBER * 8)} \\ + \text{(NUMBER * 2)} \\ \hline \text{(NUMBER * 10)} \end{array}$$

With its 6502 code shown here:

```
10 *= $0600
15 LDA #23 ;PLACE 23 IN ACCUM.
20 ASL A ;TIMES 2
25 STA TIMES2 ;SAVE #2 VALUE
30 ASL A ;TIMES 4
35 ASL A ;TIMES 8
40 CLC ;CLEAR CARRY FOR ADD
45 ADC TIMES2 ;*8 + #2 = *10!
50 STA TIMES10 ;SAVE TIMES 10
55 BRK ;AND STOP!
60 TIMES2 *=*+1
65 TIMES10 *=*+1
70 .END
```

As you can see, you can multiply a number by almost any value through a combination of left-shifts and add/subtract operations. It's just a matter of careful planning when writing a program.

Multi-Byte Multiplication

Now that we've looked at single-byte multiplication, we can go on to bigger and better things, such as multiplying two-byte values. The figure below shows the procedure for multiplying the two-byte value TOTAL by 16. Note that the low-order byte is always SHIFted, and the high byte is always ROTATED.

```
10 *= $0600
15 LDA #502 ;PLACE 02...
20 STA TOTAL+1 ;IN TOTAL HI BYTE
25 LDA #54F ;PLACE 4F...
30 STA TOTAL ;IN TOTAL LO BYTE
35 ASL TOTAL ;SHIFT LOW,
40 ROL TOTAL+1 ;ROTATE HI = TIMES 2
45 ASL TOTAL ;SHIFT LOW,
50 ROL TOTAL+1 ;ROTATE HI = TIMES 4
55 ASL TOTAL ;SHIFT LOW,
60 ROL TOTAL+1 ;ROTATE HI = TIMES 8
65 ASL TOTAL ;SHIFT LOW,
70 ROL TOTAL+1 ;ROTATE HI = TIMES 16
75 BRK ;ALL DONE!
80 TOTAL *=*+2
85 .END
```

Lines 15-30 initialize the variable TOTAL to \$024F (0000001001001111 binary). Note that the label TOTAL is the low-order byte and TOTAL+1 is the high-order byte.

Line 35 shifts the low byte of TOTAL left one bit, multiplying it by two. This operation places the contents of Bit 7 of the low byte in the Carry flag so that it can be shifted into the high byte by the next operation.

Line 40 rotates the high byte of TOTAL left, placing the Carry flag's contents in Bit 0. Like the shift operation, the rotate places the contents of the high byte's Bit 7 in the Carry flag. After this instruction executes, TOTAL contains \$049E (0000010010011110 binary), or two times the original value.

Lines 45-50 multiply TOTAL by two a second time, resulting in a value of \$903C (0000100100111100 binary), or four times the original value.

Lines 55-60 multiply TOTAL by two again, giving a value of \$1278 (0001001001111000 binary), or eight times the original value.

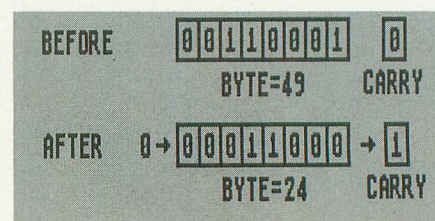
Lines 65-70 multiply TOTAL by two a final time, giving a final result of \$24F0 (0010010011110000 binary), which should be \$024F*16. Checking, we find that \$024F is 591 decimal. 591 times 16 is 9456 decimal, or \$24F0, and our answer in TOTAL is correct.

These examples show the basics of 6502 multiplication, but don't stop here. Study the above code and try creating your own programming puzzles. I've given you the ball, now run with it!

Divide and Conquer

Now that we've covered simple multiplication, let's look at basic division. You know how bit-shifting works, so picking up the finer points of binary division should be easy.

Remember how shifting the value 49 decimal (00110001 binary) left one bit gave us 98 (01100010 binary)? What happens if we shift the value RIGHT one bit? **Figure 4** gives us the answer.



As you can see, we've just discovered the

first limitation of binary division—we can't handle decimals! Using real numbers instead of integers, $49/2 = 24.5$. Shifting the value 49 right one bit divided it by two, all right, but we lost the decimal portion of the result. We'll look at real-number division in later installments of *Boot Camp*, but for now the loss of the precision does not matter. I mentioned the problem because it's good for you to be aware of this limitation.

In the 6502 instruction set, the operation that performs this right-shift is the LSR (Logical shift right) instruction. Its formats are:

```
LSR A    (ACCUMULATOR)
LSR nn   (ABSOLUTE)
LSR n    (ZERO PAGE)
LSR n,X  (ZERO PAGE INDEXED X)
LSR nn,X (INDEXED X)
```

As **Figure 4** shows, the LSR instruction shifts all the bits of the indicated byte right one position. A zero is placed in the high-order, or 128, bit. The low-order, or 1, bit is shifted into the Carry flag. This allows us to perform multi-byte right-shifts, similar to multi-byte left-shifts.

Before we look at multiple-byte division, let's look at a single-byte example.

```
10  * = $0600
20  LDA #184      ;PUT 184 IN ACCUM.
30  LSR A         ;DIVIDE BY 2
40  LSR A         ;DIVIDE BY 4
50  LSR A         ;DIVIDE BY 8
60  STA DIV8      ;SAVE RESULT
70  BRK          ;AND STOP!
80  DIV8 *=*+1
90  .END
```

The above shows an example of dividing a single-byte value by eight. Like multiplication by eight, this operation requires three shifts, but in the opposite direction. In this example, we divide the number 184 decimal by eight, placing the result in the location DIV8.

Line 20 places the number 184 (0111000 binary) in the accumulator.

Line 30 shifts the accumulator contents right one bit, dividing the value there by two. After this instruction, the accumulator contains 92 (01011100 binary).

Line 40 shifts the accumulator right another bit, dividing the value by two again. At this point the accumulator is divided by four and contains 46 (00101110 binary).

Line 50 shifts the accumulator right a final time, leaving the accumulator containing the original value divided by eight. At this point it contains 23 (00010111 binary).

Line 60 stores the contents of the accumulator in the location labeled DIV8. If you examine this location after the program executes, you will see that it contains 23 decimal (\$17 hex). Checking, you will find that this is 184 divided by eight.

Line 70 BREAKS the program, stopping execution.

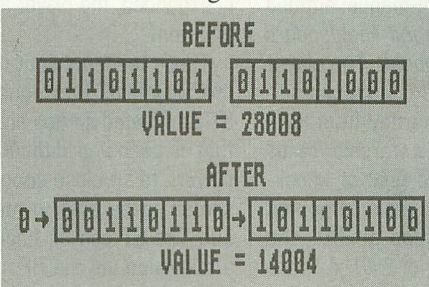
Line 80 reserves one byte for the value DIV8.

Now you see how simple single-byte division is. If you want to divide any integer up to 255 by a power of two, this process works fine.

Shifting Into High

Up till now, we've limited ourselves to simple, single-byte division. Now let's see how we do it with more than one byte.

Figure 5 shows the division of the two-byte value 28008 by two. As you can easily calculate, the result is 14004. If you compare this example with the multi-byte multiplication shown in **Figure 3**, you will notice an interesting difference.



In multiplication, the low byte is shifted and the high byte(s) is (are) rotated. This is because the bit-shift proceeds from right to left.

In division, however, things are reversed. Since we are shifting all the bits to the right, the highest byte is shifted and the remaining bytes are rotated. This allows the low-order bits of the bytes being divided to shift into the lower-order bytes.

Let's look at an example of the three-byte value SCORE being divided by four. The code necessary is shown below:

```
10  * = $0600
15  LDA #$49      ;SET UP...
20  STA SCORE+2   ;3-BYTE...
25  LDA #$23      ;VALUE...
30  STA SCORE+1   ;IN SCORE...
35  LDA #$F8      ;= $4923F8
40  STA SCORE
45  LSR SCORE+2   ;DIVIDE...
50  ROR SCORE+1   ;SCORE...
55  ROR SCORE     ;BY 2
60  LSR SCORE+2   ;DIVIDE...
65  ROR SCORE+1   ;SCORE...
70  ROR SCORE     ;BY 4
75  BRK          ;AND STOP!
80  SCORE *=*+3
85  .END
```

Lines 15-40 initialize the three-byte value SCORE to \$4923F8. Remember that multi-byte values are always stored in low-byte/high-byte order. In this case SCORE is the lowest-order byte and SCORE+2 is the highest-order byte.

Line 45 shifts the highest-order byte of SCORE right one bit. The 1-bit of SCORE+2 is placed in the Carry flag, ready to be rotated into the next byte of SCORE.

Line 50 rotates the middle-order byte right one bit. The bit carried from the highest-order byte is shifted into SCORE+1's 128-bit, and the 1-bit of SCORE+1 is placed in the Carry flag for the next rotate.

Line 55 rotates the low-order byte of SCORE right one bit. Once again, the Carry status is placed in the 128-bit, and the 1-bit is shifted into the Carry. This final Carry is not used, but is ignored. After this instruction executes, the value in SCORE is divided by two and contains \$2491FC. As an exercise, you can calculate the binary and decimal values.

Lines 60-70 perform the same function as Lines 45-55, leaving SCORE with the original value divided by four, or \$1248FE. Calculate the decimal and binary values for this result, and you will see that the original value has been divided by four.

Line 75 BREAKS the execution of the program. At this point, you can examine the three bytes of SCORE and see that they contain the proper result.

Line 80 reserves three bytes for the variable SCORE.

Well, now you have the basics of integer-binary multiplication and division under your belt. The principle is simple; you just have to work with it until you feel comfortable. In order to do that, create your own problems to solve.

Here it Comes

For those of you who need some prompting to get started with problems, here's one that shouldn't be too hard if you've read carefully.

Write a program that multiplies the value 5 by 27. Use any of the techniques we have discussed so far. There are several possible solutions to this problem, so give it your best shot.

Next issue, we'll look at a couple of possible solutions. We'll also find out what the stack is and how it helps us write subroutines.

GAME SHELF

Captain Fizz Meets the Blaster-Trons

Reviewed
by
Frank Eva

Psygnosis, Ltd.
Distributed by Computer Software Service
2150 Executive Drive
Addison, IL 60101
(800) 669-4912
\$29.95; Color only

Captain Fizz, an arcade adventure, is another of Psygnosis' independently produced titles, released under the Psychapse label. The single-disk program, packaged in the familiar square hard case, is accompanied by a six-page mini-manual entitled *The Queen's Cloned Highlanders Regimental Handbook*. The player assumes the role of a cloned soldier who must journey from level to level inside the starship *Icarus* with the ultimate goal of knocking out the craft's main computer. In the best tradition of HAL, the berserk computer of *2001: A Space Odyssey*, the computer aboard the *Icarus* has gone haywire, and the ship is hurtling uncontrollably toward the sun.

Captain Fizz is a simultaneous two-player split-screen contest. (One person can play alone, but is warned that it will be impossible to win.) The right and left sides of the screen are filled with status readouts that will be mostly gibberish until the game's learning curve has been crested. The numerical readouts at the left display the player's score, personal health, armor (how well-shielded from enemy weapons he is), damage (how destructive his blaster is), charge (how many times his ultimate weapon can be used), credit (points earned that can be exchanged for other things) and cards (keys used to unlock color-coded doors).

Also, a vertical meter indicates how much energy is left in the blaster. Green is the safe level, but if the power should drop into the red, Captain Fizz is in mortal danger. After his energy is completely expended, the player is virtually impotent.

These readouts limit the size of the playfield. Consequently, the ST's animated sprites are so small that it becomes difficult for two players to sit close enough to the screen to be able to make them out. The ST can be hooked up to a television via the RF modulator. This might be a good idea in order to keep eyestrain to a minimum.

Because the game's sprites are so tiny and are viewed from above, there is little animation. The main characters move their arms and legs, but this can just barely be seen. With two humans connected to the computer via separate joysticks, *Captain Fizz* shows no tendency (except in one minor case) to bog down in its execution. With so many droids running around loose in the *Icarus*, it's nice to see that the game stays up to speed.

The function keys allow for some configuration of the game. F1/F2 select player 1 or 2. F3 pauses the game. F4 is supposed to restart the game, but this does not work as most gamers would expect it to. F5/F6 allow players to

commit suicide, a surefire restart function. F7/F8 detonate a smart bomb. F9/F10 are difficulty levels.

Sound plays a major role in untangling *Captain Fizz's* adventure-type web. The many sound effects are easily distinguishable; however, it's quite a trick to identify an accomplished objective by audio signals alone, especially since the documentation says nothing on the matter.

Various icons give the players extra powers. The blue icons marked with an "A" add to your personal armor. The "gun" icons increase the damage you can inflict. The dynamite icons give the player the ability to use a "blitter bomb," which destroys all enemy weaponry on the screen.

Not only is each game timed, but a player is allowed to continue only as long as his health readout stays above zero. In order to stay in play as long as possible, a player is allowed to exchange earned credits for improvements in personal status.

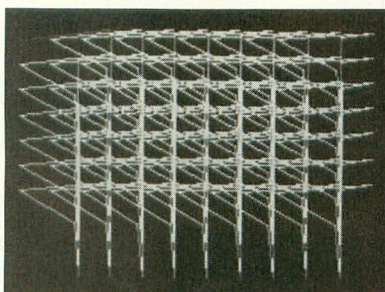
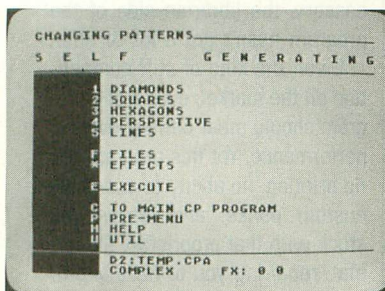
To save you some frustration, I offer the following hint: At the right side of the screen is a four-LED readout labeled "Switches." Each LED must be lit in order to disable a force field for a limited period of time. The player must have already destroyed all of the "L" icons (I'm still not sure what the "L" stands for). When this has

(CONTINUED ON PAGE 59)

Changing Patterns

Reviewed
by
Matthew
J.W.
Ratcliff

Stewart Software
11323 Blythe Street
Sun Valley, CA 91352
(213) 875-2012
800/XL/XE Disk; \$35



Changing Patterns is billed as an "art and idea generator." But this collection of four disks presents many electronic "Spiro-Graph" simulations, many of which can be varied with keyboard inputs by the user. Hard-copy and file-save features provide some creative outlets besides the show seen on the monitor as the designs are drawn.

Disk one contains the main program for *Changing Patterns*, referred to as *CP*. It also provides a "pre-menu" that serves as a gateway to the other disks, each of which may be booted separately as well. After presenting the title and revision screens, *CP* delivers the primary *Changing Patterns* main menu. To get an idea of what this program is about, selecting the self-generating function sends control to the "self-generating" art program, the main creativity module of the *CP* package. Selecting the Execute option immediately musters an interesting display of diamonds. As they are drawn, console- and keyboard-key presses change the size, shape, colors, margins and

numerous other features of the design.

Multiple columns of objects, such as hexagons, 3-D perspective boxes, octagons and lines are created with this module. The sizes and overlapping of the objects create many different, sometimes hypnotic, patterns. Certain controls applied during the creation of the art add an asymmetric twist to the screens. A particularly fascinating design can be saved to a temporary file by pressing all the console keys—Start, Select and Option—simultaneously.

Once a masterpiece is created and saved, *CP* is used to load the picture and print it. Only Epson and Star Micronics Gemini printers are supported. *CP* defaults to "none," instead of Epson, which requires the user to specify a printer every time the *CP* is run. The manual recommends using some advanced graphic-printing utilities, such as those from Alpha Systems.

CP does a good job of generating a quick plot of a design at the top-left of a page of paper. This is ideal for folding the page lengthwise, then widthwise, to create a nice-looking greeting card, ready to be filled in by hand on the inside. This is probably the most practical use of *Changing Patterns*. Its ability to save screens allows the user to make use of some interesting design results in other programs, with some additional programming efforts.

The second disk is called the *Exploration Disk*. It contains 34 separate short graphics programs, all tied together by a main menu. Each has a "canned routine" for creating the art. Some create intricate tile work or pictures, very much like those done with a child's Spiro-Graph. Any pattern is easily saved to disk for

later viewing and plotting with the main *CP* module.

All the programs are written in BASIC and are unprotected. This allows the user to back-up the files, as well as study and learn from them. The programs work well overall, but I did crash one of the *Exploration* programs while adjusting the size of the patterns. Because all the software is in BASIC, the pattern-generation process itself can be tedious at times. The manual documents many different "macro-key" commands that are employed by the various art modules.

The third disk is the *Help* disk. It is a nicely detailed tutorial that helps the user through some of the more esoteric macro functions. The manual also provides a short tutorial that contains a series of step-by-step exercises to create some pictures and help teach the menu and macro-key functions.

The manual, nearly 50 pages, is complete, with many details. It is not well organized, however. The hardest part of using the manual is locating the right set of macro-key commands for each art generator.

Changing Patterns is not a spectacular program, but it will provide a lot of interesting artistic interaction with the computer on those rainy days. Its inconsistent user controls provide for a bit of adventure in experimenting with cause-and-effect. When it comes time to get caught up with letter writing, the printouts make nifty little greeting cards. There are infinite designs to be discovered with *Changing Patterns*.



Matthew Ratcliff, a frequent contributor to *ANALOG Computing*, lives in St. Louis, Missouri, with his wife and two children.

MT-32 Controller

Reviewed
by
Michael Friesen

What defines a useful piece of software? That's a good question to throw onto the bulletin board next time things get boring. The *MT-32 Controller* from Polysoft prompted me to ask myself this question. For although the program does some genuinely useful things, it has enough idiosyncracies to occasionally hamper the music-making process.

This is a desk accessory, but you may want to beware of what kind of software you team it up with. Because the program grabs many interrupts, you might find that some screen functions are slower than normal. Programmer David Barr assures me that sequencer timing is not affected.

Along with the main program are three additional files. The first two are banks of 64 sounds. The other file is a "translibrarian," which allows you to load (from disk) and store (to synth) banks of sounds. This program is compatible with any synth supported in the *Controller* series: Roland's D-110, MT-32, Yamaha TX81Z and FB-01 and the Kawai K1.

The program is not copy protected and must be installed in the root directory of your boot drive. It will run either in medium or in high resolution.

The program's functions can be divided into two distinct areas: patch librarian and front-panel expander. Both functions reside on a single screen.

The Front Panel

The panel expander is where this program scores most of its merit points. The expander takes up the left-hand portion of the screen and gives you quick and easy access to Master and Part controls.

Reverb, pan, trigger mode and fine-tuning are some of the parameters this program gives you access to. Even though many of

the parameters are already available from the front panel of the synth, it's nice to have them all laid out in an easy-to-see format.

Parameter values are changed by left-clicking on Up or Down arrows to either side of a parameter. Left-clicking inside the parameter box sets the value to a sensible default. If you're accustomed to a quick mouse, you may have to reset the speed to fine-tune some of the parameters. Sometimes it's hard to stop the mouse just at the value you want.

The Right Stuff

The right mouse button is put to good use. Clicking on the right button gates the sound selected for the current part. Holding the button down sustains the sound, and removing your finger from the button starts the release cycle.

Here's the neat part: Pitch and velocity are determined by y and x positions of the mouse. Right-clicking while the pointer is in the upper-right corner will gate a high-pitched note at a high velocity. Lower-left will give you a note with a low pitch at a low velocity. This is a great idea, since it makes use of the oft-ignored right button.

Librarian?

Whatever the ads and the packaging might say, this is not a patch librarian. It simply does not conform to the standards of what a patch librarian can reasonably be expected to do.

On the good side, the program will give you access to both banks of ROM sounds (the presets), as well as one additional bank of 64 sounds. Each bank is displayed as a whole, all 64 patches at a time. Even though the font is necessarily small, it's nice to be able to pick

from 64 on-screen sounds.

New sounds can only be loaded into the *MT-32 Controller* if the file of 64 sounds has already been saved in the *Controller's* unique format. And it is impossible to swap sounds between banks. These two factors weigh heavily against the program. Here's why:

You can only get new sounds into the *Controller* if they have already been saved in the proprietary format. If you buy a bank of sounds, they will either need to have been saved in *Controller* format, or you'll have to find a way to send them to your synth. Once the sounds are resident in the MT-32, you can use the Retrieve function to get the sounds into the *Controller* and then save them to disk.

This means that, unless you're content to play around with the two new banks and the two banks of presets, you'll need to buy a patch editor or librarian before you can start messing around with more sounds.

And because you can't shuffle sounds between banks (or even within a bank), there's no way to remap sounds or create performance banks. You'll need to buy a patch librarian to be able to do that.

The Sound Banks

The two new banks of sounds supplied with the program represent the usual mix of weird effects, the requisite attempts at a piano, some nice strings and a few absolute killers. My own favorite is the Rock Organ patch in the D50 bank.

Some of the sounds appear to be either misnamed or have had their data grossly scrambled. All in all, though, they're a nice addition, giving the user an alternative to

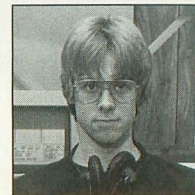
ROM presets.

The Bottom Line

It's hard to pin down exactly where this program should fit into the scheme of things. The Master and Part controls are useful, but at least partly duplicated on the front panel of the synth. And the librarian can only deal in terms of whole banks of patches that have been stored in the *Controller's* proprietary format. Unless your librarian is a real monster, which won't work as a desk accessory, the librarian side of the program hasn't got a lot to offer.

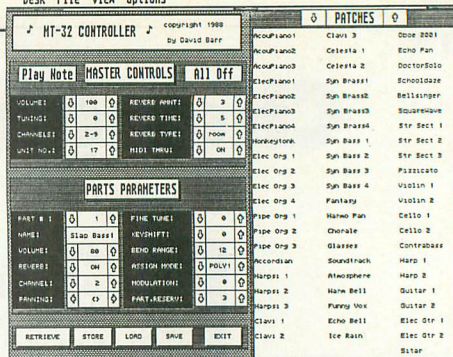
Given the kind of stiff competition on the market, even a \$50 program should meet certain levels of performance. Yet this program has no printing, no ability to assemble custom banks, and leaves you stuck with that proprietary file format, requiring you to have another patch editor or librarian before you can do any serious work with your synth.

In strictly economic terms, this program doesn't offer enough to justify its price. Fifty dollars is too much to pay just to access a few new functions.



Michael Friesen has worked as a bookbinder, a chicken-shed cleaner, a cow milker, a ditch-digger, a jingle composer and a synthesist for a touring theatre company. A MIDI fanatic, he likes to annoy retailers by asking them about products that haven't yet been released.

Desk File View Options



Airball

Atari Corp.
1196 Borregas Avenue
Sunnyvale, CA 94086
(408) 745-2000
XL/XE cartridge; \$39.95

**Reviewed
 by
 Matthew J.W.
 Ratcliff**



Airball is a fascinating three-dimensional graphic adventure for one player, with superb graphics and entertaining musical sound effects. It is one of the few games Atari has released recently, with a copyright of 1987 by Microdeal.

The instruction leaflet does not provide much detail, other than the basics of game play. It seems you have upset an evil wizard. No one knows exactly why, but he is so angry that he conjures, "Mortal human, weak and small, turn into a bouncing ball." *Poof!*

Your goal, as a deflating ball, is to navigate the evil wizard's mansion to recover his spell book. There are more than 150 peril-filled rooms to search. What's worse, the wizard has given you a slow leak. Scattered about the labyrinth of rooms in the mansion are air pumps. Whenever you are nearly flat, you must bounce upon one so it can pump you up. Wait too long and the ball explodes in a shower of rubber. You are given only four balls to complete the adventure, so you need to learn where the pumps are.

Each room and hallway is viewed from above. The rooms are tilted and rotated at a 45-degree angle. This makes joystick control confusing at first, as is typical with all games of this type. I found that turning my joystick 45 degrees to the left made it much easier to navigate the rooms.

The rooms are finely detailed,

with excellent perspective and shading for a nice illusion of depth. The floors and walls appear to be made of a brown sandstone. The floors are tiled; the safe paths are smoothly colored. Some tiles are impregnated with lethal spines, which you must roll around or hop over. Wicked spikes protrude from the floors in nearly every room. Bouncing on any of these spikes is certain destruction. Staircases, which are easy to roll down or bounce up, can be found in many rooms.

The total number of air balls remaining is shown at the bottom-left of the game screen. The score is at the bottom-right. In the center is a horizontal bar that acts as a gas gauge for total air remaining in the ball. When you hop on an air pump, this line increases. You must hop off before overinflating and bursting.

Each room has at least one entrance and exit, depicted as a stone archway. As you progress in the game, it is wise to draw a map. Some rooms require instant selection of the proper move upon entry.

Along the way you will encounter, scattered about the mansion, precious stones and gold bricks, which may be exchanged for points. Apparently, the wizard is following you closely and dropping these valuables as he goes. Sometimes, you will enter an empty room, exit and immediately re-enter it to find more gold. This is entirely random, so you must keep

a sharp eye.

There are also special objects, essential to progress and ultimate completion of the game, that must be collected along the way. For example, some rooms are dark and thus unsafe to travel without proper lighting. You must gather the wizard's candle before you can safely enter this kind of room. Various obstacles are impassable until you move some crates into a position that allows you to safely bounce over them.

Strategy is important in *Airball*. Only one object can be carried at a time. As you learn the mansion's mazes, planning where to place these objects for most efficient progress is vital.

Airball will provide many hours of addictive fun. The graphics detail, depth, playability and musical effects reinforce the fact that the 8-bit Atari is an excellent game-playing machine. Its only flaw, due to the fact that it is a cartridge game, is that you cannot save a game in progress. My hat is off to Atari and Microdeal for bringing such a fine and timely video game to the Atari XL/XE/XEGS computers. Note that *Airball* is not compatible with older Atari 400/800 computers.

The author wishes to thank Randall's Home Computers of St. Louis, Missouri, for their valuable assistance in preparing this review.

Talespin

Microdeal
476 S. Telegraph
Pontiac, MI 48053
(313) 334-5700
\$49.95; Color only

STAC

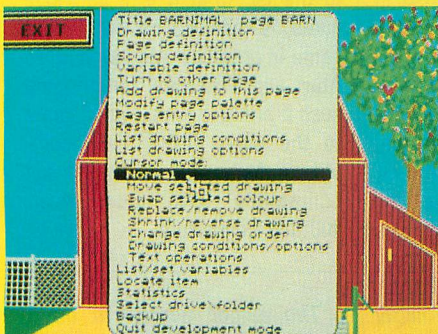
Terrific Software
544 Second Street
San Francisco, CA 94107
(415) 957-0886
\$69.95; Color only

**Two
 Adventure-
 Creation
 Programs
 From England
 Reviewed
 by
 D.A. Brumleve**

Figure 1: The opening scene from a *Talespin* adventure. When the player moves the mouse over text that can affect the progress of the game, the text is highlighted. Text balloons that cannot affect the course of the game have the same background at all times. The mouse is represented here by a stylized pointer in the lower right-hand corner of the text window.



1

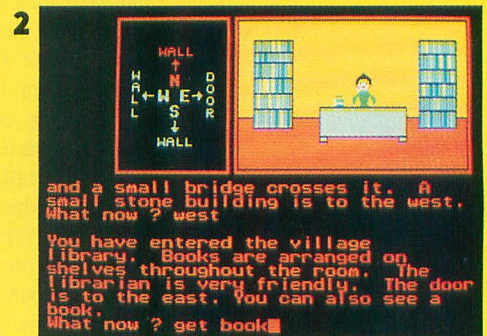


5

Figure 5: *Talespin*'s main menu and submenus are mouse-controlled.

The development of my interest in creating programs did not coincide with a willingness to spend the time it takes to learn a programming language. I was looking for easier, faster results, so I ordered a public-domain disk containing an "adventure creator." This program, as it turned out, was difficult for a novice to use and the product it created was a simple text adventure, whereas I wanted my programs to make ample use of graphics. I finally admitted that there were no shortcuts, loaded up ST BASIC and plodded away. With the help of an excellent little book, *Write Your Own Adventure Programs for Your Microcomputer* (by Jenny Tyler and Wes Howarth, Usborne Computer Books, Tulsa: EDC Publishing, 1983), I

Figure 2: *STAC*'s product has a more traditional adventure format. The text bar at the top of the screen is provided automatically by *STAC*, but may be eliminated if the programmer doesn't want it. Most of the rest of the top portion of the screen can be filled with a picture. The text shown is in the default font, but other fonts can be used.



2

and a small bridge crosses it. A small stone building is to the west. What now? west
 You have entered the village library. Books are arranged on shelves throughout the room. The librarian is very friendly. The door is to the east. You can also see a book. What now? get book

6

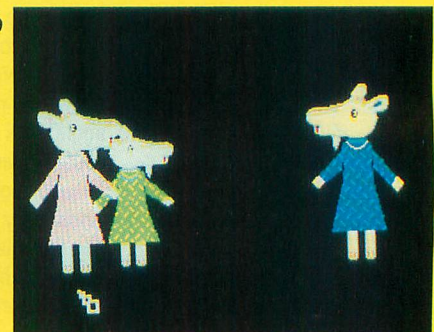


Figure 6: Each of these images is a duplicate of the original, shown at the left. The left-to-right orientation of the goat on the right has been reversed. The goat in the middle is reduced in size. Different colors are used for the clothing of each copy, and the goat at left has a darker coat.

did write an adventure game in ST BASIC.

Lately the ST magazines have contained ads with lines like these: "Adventure-game writing made easy!" and "Requires no programming skills!" The ads promote two very different graphics/text adventure game-creation programs. Since this was precisely the kind of application I had yearned for before learning to program, I was intrigued. Would these programs have met my needs two years ago? Would they allow a non-programmer to produce high-quality products?

The two programs, *Talespin* and *STAC*, do create graphics/text adventures, but there are many differences both in the products they create and in the means by which these products are developed.

Talespin, "An Adventure Game Creation Program," distributed by MichTron/Microdeal and written by Mark Heaton, creates a mouse-controlled program. Graphics are absolutely essential to the use of the product, as the player must click the mouse on a graphic before any consequence (appearance of text, movement to another screen, appearance or disappearance of graphic images, playing of digitized sound) can occur. Text is displayed in tiny type in *Talespin*'s own unique windows. Once text is displayed, the user may also be able to click on the text in order to make choices that yield specified results. The *Talespin* programmer determines whether or not clicking on a particular graphic will cause any results. For example, in Figure 1, an opening scene from a *Talespin* adventure is shown. As the scene begins, clicking on the kids will have no effect. Clicking on the mother goat will cause a text balloon to appear. Clicking on

that text balloon, in turn, changes the value of a variable. If you click on the kids at this point, text balloons will appear for each of them.

STAC, "The Atari ST Adventure Creator," distributed by Terrific Software and written by Sean T. Ellis, creates a more traditional graphics/text adventure. Graphics, if desired, can occupy most of the top half of the screen, and text is displayed below it. Responses by the player are typed; the mouse has no role in game play. Graphics are used merely to enhance the player's understanding of the text. A cursor (of adjustable style) shows the player where the next letter he types will appear.

Products created with both programs can be distributed in runtime versions, so that they may be used by people who do not own the respective development package. A *STAC* program can be saved as a .PRG file; the data unique to the *STAC* programmer's adventure is saved to a .LNK file and the needed programming instructions are saved in a .PRG file. Double-clicking on the .PRG causes the computer to automatically load the related data and begin the game.

To run a *Talespin* program, the user loads the public-domain runtime module, TELTale.TOS, and then selects the data file from *Telltale*'s menu. Use of *Telltale*'s short menu is not difficult to learn. In *Talespin* programs, there is a major delay right before the display of a new graphic or the transfer to a new page. This occurs even when the computer does not need to read information from disk, and it annoyed all of my play-testers at least a bit. Products created with both programs can occupy more than one disk. The player's place in the game can be saved to disk.

Both *Talespin* and *STAC* can be used to create a slick, professional-looking program. While *STAC*'s literature promotes its use only for the development of adventure games, it is worth noting that *Talespin*'s author recommends other uses for the program, including service manuals and educational Hypercard-like files. Indeed, this reviewer has used *Talespin* to create an adventure game, a catalog, a sound-recognition program and even a somewhat successful matching game. *Talespin* is suitable for many kinds of applications in which graphics and text must be linked.

Adventure games are mapped out in "rooms," which may or may not represent actual rooms indoors. An adventure game is really a kind of database. There is a filing system that keeps track of the contents in each room, the connections to other rooms, the graphics used, room descriptions, etc. The program keeps track of a variety of conditions—and of the results to be executed if those conditions are met. For example, if the user chooses to "Go north" and a connecting room is to the north, the program will move the user to the connecting room.

A great deal of planning must be done before the adventure-creation program is finally loaded to begin actual development. Adventure-game development does not lend itself to the kind of seat-of-the-pants programming to which I am accustomed, but both programs do allow for changes in a basic plan, once it is in place. Both the *STAC* and *Talespin* packages provide sample adventures to play, and the manuals lead the programmer through the development of another sample.

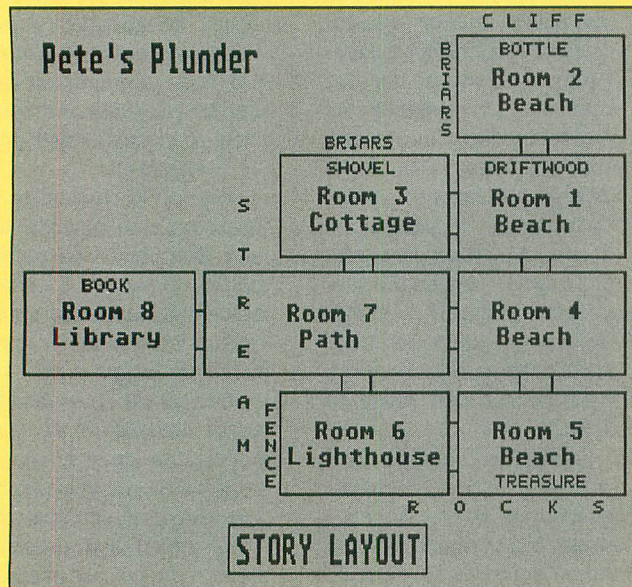
Both programs come with

a paperbound manual and two disks. *STAC*'s manual is bound with staples through the spine; it lays open on your table while your hands are busy at the keyboard. *Talespin*'s manual is "perfect-bound" (a misnomer if ever there was one) and will not stay open to a particular page unless you break the spine. Neither manual is as helpful as it should be.

The *STAC* manual's 72 pages are crammed with tiny and faint text. The sample type-in program is clearly presented, but the presentation is interrupted by chapters dealing with the command set, the graphics editor and the music commands, making it difficult to pick up where you left off. The sample adventure is text-only, so the reader is not led through the process of adding graphics to a preplanned adventure. The chapters are otherwise well organized, but the treatment of some features is weak. There are several options in the editor's menu in which commands can be entered; some types of commands are only effective when used in a particular option.

The organization and distribution of the commands in the *STAC* manual is not identical to their use in the editor. In addition, the listing of commands in an appendix is by category as presented in the book; these categories do not necessarily correspond to the menu options in the editor. Commands are frequently presented without examples of their use within the context of the editor's menu. For example, the command "colour," which determines the color values of the lower half of the screen, is clearly presented and an example is shown, but no information is offered that really tells you how to implement the command in a program.

Figure 3: Development of an adventure game requires careful planning. The manuals for both programs have helpful suggestions for adventure planning and design.



I finally determined which menu option to use for the "colour" command by trial and error—a process that took nearly half an hour.

The book attempts to teach adventure construction and design, and some of the suggestions are helpful. Sample flowcharts and maps help you visualize adventure construction. A major complaint for me, personally, is the writing style selected by the author. The instructions are friendly, but attempts at humor distract the reader while the difficult content requires concentration. There is no alphabetical listing of *STAC*'s commands, and no item-by-item presentation of the menu choices. The index is virtually useless. For example, "palette" and "colour palette" both refer the user to page 30, where the color palette of the top half of the screen is discussed, but the example of the "colour" command on page 14 is not mentioned. After a series of such experiences, I learned not to

bother consulting the index.

The 133-page manual for *Talespin* has readable type, illustrating graphics, boldface headings and a slightly more useful index. The booklet contains helpful information about planning programs and an item-by-item explanation of the use of the main and secondary menus. The writing style is clear and straightforward. A weak link is the walk-through demonstrating the creation of a sample adventure. This tutorial does demonstrate the use of many menu options, but I found some instructions confusing, and as with *STAC*, the author interrupts the walk-through with a brief explanation of the drawing program.

These manuals are of minimal assistance. Fortunately, *Talespin* users will find that minimal assistance is more than enough to get them up and running. *STAC* users, however, should be prepared to learn to use the program largely by experimentation and to spend a great deal of

time doing so.

Even if it were graced with a top-notch manual, *STAC* would still be by far the more difficult program to learn to use, but along with that complexity comes far greater control of the finished product. *STAC* has a huge command set. There is a variety of commands to link particular conditions with particular results (if...then), to set the color palette, to play three-voice multi-octave melodies, to control movement and actions (repeat...until, GOTO) and to keep track of the value of variables. The program can generate random numbers for use in determining conditions. Variables are limited to two values (true/false, on/off, etc.). Variables, rooms and even vocabulary items are assigned a number, and that number is used in commands to identify the feature.

A basic vocabulary file, including typical responses by the computer ("You can't do that!") is provided with the program; vocabulary and instructions unique to the programmer's own adventure can be added to this list. The *STAC* programmer can select or design alternative fonts for text displays. Even the resolution of the text portion of the screen is under the programmer's control: While the top half of the screen is always displayed in low resolution, the lower half can display 40- or 80-column text in the font style of the programmer's choice (a neat trick), thus accommodating lengthy room descriptions. The color palettes of the top and the bottom halves of the screen are manipulated separately.

Nearly all activity in the *STAC* editor is controlled by the keyboard. The menu system is complicated, but actions within each of the menu choices are designed for con-

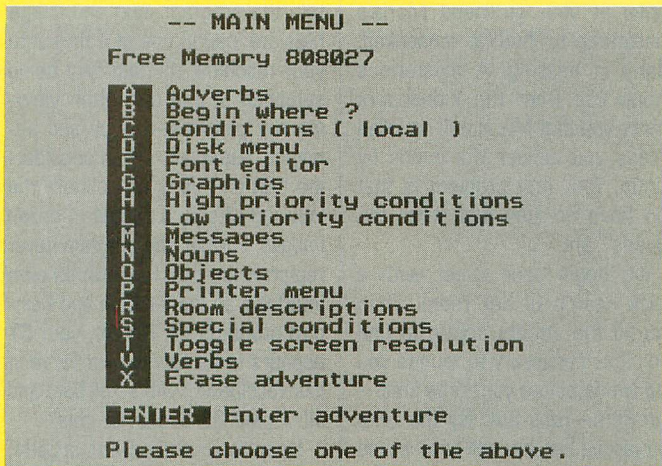
sistency, so it isn't terribly hard to use. A useful feature is the ability to print out information from the various editor categories; this helps you keep track of the numbers associated with vocabulary, rooms, etc. The most difficult aspect of program development is the planning and entering of condition-related commands; this task is made no easier by its minimal treatment in the manual. *STAC* contains a pared-down drawing program, but pictures can be imported from *Neochrome* or *DEGAS*.

By contrast, *Talespin* is easy to use and easy to learn to use, but there is a corresponding decrease in control of the finished product. *Talespin* provides a mouse-controlled development menu. Choices from the limited command set are selected by mouse. *Talespin* refers to results to be executed if conditions are met as "options." I found this terminology confusing. It would have helped me in getting started if results had been called "results" or "consequences."

One feature I missed was the ability to link particular conditions to particular responses. *Talespin* allows you to say that "if <variable> = 1 or if <variable> = 2, then play <sound> and go to <page>," but it doesn't let you say, "(if <variable> = 1, then go to <page>) or (if <variable> = 2, then play <sound>)." This small difference decreases the programmer's power and control considerably; there are sometimes ways to get around this deficiency, but it is nevertheless limiting. On the other hand, variables and their value limits can be defined by the programmer, and a limited use of probability is possible.

One thing that makes using *Talespin* so simple is that

Figure 4: STAC's main menu's features are selected by pressing the corresponding key.



it allows the programmer to name rooms, variables, objects, etc., with real names. Because the finished adventure is mouse-controlled, no vocabulary file is needed. The editor for typing in text to appear in windows is straightforward and easy to use. But the small font size and its style cannot be altered.

A major weakness of *Talespin* is that sound is only possible through the use of *ST Replay* or *Replay 4* (also from MichTron/Microdeal) digitized sound files. (You can also import prerecorded sounds from another *Talespin* adventure.) This is a nice enhancement to an adventure, but unfortunately, even short .SND files take up huge amounts of disk space and memory. I would like to have been able to access the ST's sound chip more directly, as in BASIC. If an upgrade to *Talespin* is produced, I hope it will provide the ability to play simple melodies without undue use of memory, just as *STAC* does.

Talespin automatically adds changes to the adventure file on your disk as they are made, a real convenience for a programmer. It is, of course, also possible to save the file under a new name.

Graphics are important to

the success of *Talespin* adventures, and the creation program provides many useful features to allow the programmer to make the most of whatever graphics are available. Each object on the screen (for example, each of the kids, the mama goat and the door in Figure 1) is controlled separately in memory. A background color is chosen, and when loading a picture, whatever is not part of the background is considered by the computer to be part of the object.

Objects can be readily repositioned. Their position relative to each other is adjustable; with a few clicks of the mouse, the programmer can place one object behind another and in front of yet another.

A different color palette can be used for each page, if desired. Multiple versions of the same object can appear on the same screen, and yet they can each be changed so much that the player is not aware that they are actually duplicates of the same image. For example, a duplicate can be made shorter (or narrower) than the original image, or the left-to-right orientation of the duplicate can be switched, or colors used in the object can be swapped

for other colors.

Since text appears only in windows, graphics may fill the entire screen, not just the top half. A simple drawing program is provided within *Talespin*, and graphics can be imported from *Neochrome*, *DEGAS* or another *Talespin* adventure. The drawing program is suitable for touching up graphics.

Talespin is not copy-protected, and it runs from a hard drive. *STAC*, on the other hand, is heavily copy-protected. The particular copy-protection scheme used prevented the disk from running on my Mega ST4. This is the first experience I have had with copy protection of development software, and I was shocked. This plan may prevent the program from being pirated, but it also prevents legitimate users from enjoying the benefits of their purchase. When I've used the phrase "*STAC* programmer" in this article, I haven't used it lightly. Using *STAC* is programming, and it is hard work. A programmer with a program in progress doesn't want to wait six weeks or more for the return of a working version of his damaged development software. Because *STAC* files can grow to be quite large, a programmer is likely to want to work from a hard drive, where multiple saves can be easily accommodated.

STAC isn't cheap, and purchasers have a right to expect a good value for their money. Preventing legitimate, convenient use of a development package by purchasers is, in my opinion, absolutely inexcusable. Before purchasing the program, serious developers should consider whether they are willing to put up with the hassles created by *STAC*'s copy-protection scheme.

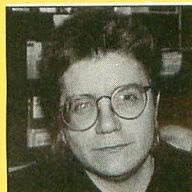
Two years ago I wished for a program that would allow

me to create high-quality graphics/text adventures relatively effortlessly. *Talespin* does. *Talespin* can be used by almost anyone; certainly children in the later elementary grades can learn to use it with some adult guidance. It has the added advantage of having multiple uses, as it can create a variety of programs linking text and graphics.

The complexity of *STAC* would have precluded my use of the program two years ago. *STAC* programmers will benefit from extensive experience with a programming language; I definitely would not recommend its use by novices, especially because the manual is so weak. For the experienced, *STAC* does have some advantages over programming an adventure game "from scratch"; these advantages must be balanced, however, against the inconvenience of copy protection.

Both of these programs can be used to create professional-looking adventures. The products created are very different, however, and if you have a strong preference for one over the other, you may want to let that be your guide.

Scenes in the sample pictures come from public-domain programs designed by the reviewer. *The Wolf and the Seven Kids* and *Barnimals* were developed with *Talespin*, and *Pete's Plunder* was created with *STAC*. ☐



D.A. Brumleve is a programmer who prefers to do things the easy way, as long as it's easy to learn to do things the easy way.



Twilight's Ransom

Paragon Software
Distributed by Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
\$34.95; Color only

**Reviewed
 by
 Steve Panak**

Upon loading *Twilight's Ransom*, you assume the role of Ron Mulligan, an artist living in Liberty City. This graphics-text interactive novel opens as you arrive at your girlfriend Maria's apartment to find it ransacked. Maria is missing. A mysterious phone call from the kidnappers warns you that Maria will be killed unless you deliver the goods by sunup. Your first problem is that you have no idea just what the "goods" are.

You begin your quest with a quick search of her room. Then you hit the streets, trying to find the clues necessary to reunite you and Maria before your time limit—and hers—runs out. By the time you rescue her, you will have visited almost 200 locations and interacted with a slew of unsavory characters, including prostitutes, con men, drug dealers and outcasts.

Unfortunately, I was disappoint-

ed upon opening the box—a feeling that remained unchecked as I examined every aspect of the package. The three disks that contain the program, text and image files are loaded one at a time into your machine, the last two being swapped in and out when viewing the pictures—sloppy and annoying, especially when considering that all this is to facilitate the viewing of only a couple of dozen images. Further, the program never replaces the desktop with its own interface, as the Infocom and Rainbird products do. Rather, your ST appears to be running *1st Word* in low resolution, with scroll bars and all. I paid 35 bucks for this?

Despite this apparent crudity, functionally the program works, sports a large vocabulary and contains vivid descriptions of the people and places you visit on your search for your beloved Ma-



Ballistix

Psynopsis, Ltd.
Distributed by Computer Software Service
2150 Executive Drive
Addison, IL 60101
(800) 669-4912
\$34.95; Color only

**Reviewed
 by
 Frank Eva**

There seems to be an interest in sports games as of late, and *Ballistix* takes advantage of this by adopting features from several sources and combining them into a new genre of indoor sports contests. Play mechanics have been borrowed from hockey and pinball, the results resembling a combination of *Blastaball*, one of three games from the *Awesome Arcade Action Pack*, and *Marble Madness*, the first coin-operated arcade game to be programmed on a 68000-based computer.

The action screen is actually two screens connected by a vertical scroll that is fast and as smooth as silk. The playfield is presented in simulated 3-D, with a top-down viewpoint that is angled toward the human player's goal. Surrounding the playfield is a gruesome audience and the skulls of would-be *Ballistix* challengers.

At the beginning of a game, a gargoyle slips down from the top of the screen and in a resonant voice cries, "Let the game commence!" He exits the screen, and

from the left, an alien referee drops the *Ballistix* puck into the action area.

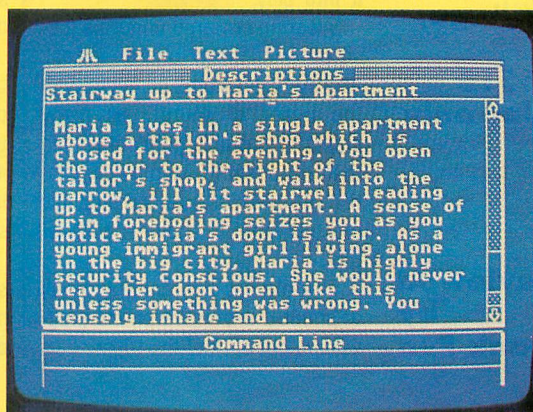
The contest revolves around the player's ability to direct the puck into the opponent's goal. This sounds similar to *Blastaball*, but the similarity ends here, for *Ballistix* has much more to offer than the aforementioned title. It's the variety of playfields that sets *Ballistix* apart from other games. In the one-player mode there are 50 playfields, and in the two-player mode, 80.

Once the game commences, *Ballistix* takes on the characteristics of pinball. There are bumpers, pipes, acid pools, troughs, magnets, hyperspace tunnels, splitters and accelerators to contend with.

The pipes are initially covered by sewer grates, but occasionally the grates pop open, and if a puck falls into the pipe (the fall is also represented in simulated 3-D), it will reappear from another pipe, located elsewhere on the playfield. The green acid pools stop the puck in its tracks, absorbing it, requiring

ria. As long as a disk swap is not necessary to view an image, response time is relatively quick. The only options that are supported allow you to save your game, control the amount of text displayed and turn the images on and off. The documentation contains a number of clues, but is otherwise uninspired.

All things considered, I cannot recommend *Twilight's Ransom* for two reasons. First, it doesn't give me what I want to see in a piece of commercial software. I would expect to see the many shortcuts and seemingly halfway-finished aspects of this game in public-domain software. Which brings us to the second problem: the price. For two bucks, or the price of a phone call, I might like this one. But at almost \$35, I can't help but feel ripped off. Steer clear of *Twilight's Ransom*. ☹



the referee to restart the match with a new puck.

One of the finest features of *Balistix* is the program's ability to be reconfigured by the player. Because of this, a player can start out at the easiest difficulty level, slowly and gradually improving his abilities as he learns the game, while never feeling the frustration inherent in so many other sports games that have such long and baffling learning curves. Unfortunately, there is no handicapping feature. Therefore, in a two-player game, both contestants experience the same effects when changing the game's configurations.

At the Level Selection screen, many levels of progressive difficulty can be chosen. Here, also, the choice can be made to play against the computer or another human. Since this is a simultaneous two-player mode, two joysticks are required. Pressing the space bar brings you to the first Options screen.

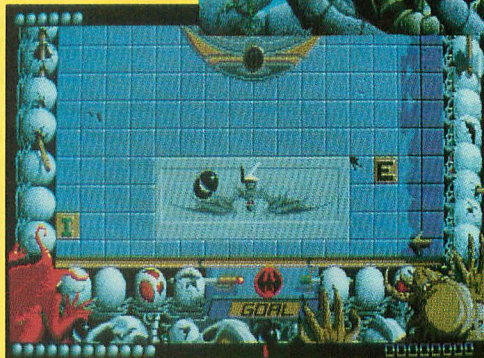
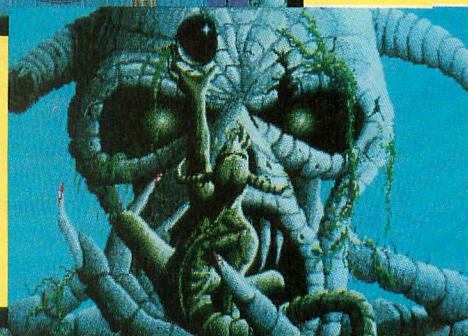
Some factors that can be altered include ball speed, number of balls

available and auto-repeat firing level (a real thumb-saver). Pressing the space bar from the main Options screen sends the contestants to the second Options screen. (Note that at any time, an Options screen can be exited by pressing the joystick trigger.)

At Options screen 2, the following features can be configured: ball lifespan, accelerator value, number of goals to win a game and number of games required to win a match.

There are a number of ways for a player to improve his score. Icons randomly appear that, when hit by a ball shot by a human player, award bonus points or shields that protect the player's goal for a limited time. Letter icons can be shot, and if the word "ricochet" is spelled, 10,000 points are added to the player's total.

But the most important characteristic of any game is its ability to hold its audience. I'm happy to report that *Balistix* is a totally captivating game, and its price is money well spent. ☹





Speedball

Cinemaware
4165 Thousand
Oaks Blvd.
Westlake Village,
CA 91362
(805) 495-6515
\$39.95; Color only

Reviewed
by
Frank Eva

Speedball is a futuristic combination of several popular sports. The game is held on a steel playfield with a goal at each end and a warp tunnel in each side wall. The ball launcher rises up from under the floor, rotates, and then shoots out the ball—a solid, stainless-steel riveted affair weighing five pounds—in a random direction, after which any player can retrieve it and put it into play.

Wearing armored uniforms (covered with razor-sharp spikes), two five-man teams compete with the intent of scoring as many points as possible (while maiming their opponents), using a combination of speed, muscle and strategic employment of power tokens/tiles.

The players can carry the ball or shoot it anywhere on the playfield. A fast click of the trigger but-

ton causes a straight shot from the hip, while a longer click results in an alley-oop.

The two-player mode requires two joysticks. However, a single player can engage in league competition, and opposing teams are drawn randomly by the computer.

The playfield is smoothly scrolled vertically. Each player—a larger than usual, outfitted human sprite—is viewed semi-top-down, allowing superb animation.

The controllable players are identified by a cursor-type pointer. The player controlling the ball is identified by arrows that point toward it from each corner of the cursor. It is not difficult to identify the sprites that are being controlled.

If your player does not have possession of the ball, but is near it, pressing the trigger button will cause him to jump and attempt to

catch it. If the button is pressed while the sprite is not capable of gaining possession of the ball, he will attempt to tackle the player facing him. This is not quite like football, since there are no fouls and no rules; just down-and-dirty fun.

Tackling is an important game dynamic. A successful tackle will result in taking control of the ball. The stronger the team, the more stamina they have against their opponents' tackling. When all else fails, you can use tokens to cheat by bribing the official, the timer, the trainer or the referee to gain an advantage over your opponent.

Speedball is superbly programmed. The animation and game play are astounding, and the variations in game mechanics and the league options all point to a high level of addiction.

Buy it. 



Batman, The Caped Crusader

Data East
470 Needles Drive
San Jose, CA 95112
(408) 286-7074
\$39.95; Color only

Reviewed
by
Steve Panak

By the time you read this, the nation will have been swept away in a wave of Bat-mania. In fact, just today I read about the slew of merchandising products poised to feed parasitically on the movie's release. *Batman, The Caped Crusader* from Data East is the first (but probably not the last) computer product based on the Dark Knight to cross my desk.

Batman resides on two disks, each containing a classic confrontation between the caped crusader and one of his arch-criminal rivals. Choose disk one to battle the Penguin, disk two to match wits with the Joker. Regardless which opponent you choose, the rules of play and goals are fixed. Batman moves about the city, collecting items he needs to quash his adversary, all the while fighting a time limit. At various points in the

game, you will be challenged by henchmen or diabolical traps set by the master criminal. Play gets progressively more difficult as you wind your way through a maze of rooms on your way to the final confrontation between you and your arch nemesis.

The command interface is a little difficult to master, but performs nicely once learned. While you are on the main screen, which shows Batman in his current location, the joystick controls movement along all three possible axes. Holding the button while moving the stick allows you to attack with high and low kicks and punches. Holding the button and pulling back on the stick accesses the status screen, which displays your lifeline, your inventory, and allows you to turn off the *Batman* theme music and restart the game. It is also through

this screen that you get, use and drop objects.

The graphics can't be described as anything less than superb. The cartoonlike images are mesmerizing to watch, with successive windows opening as you wind through the various rooms. Unfortunately, I found the puzzles to be less of a challenge than was the need to complete them in a limited timespan.

One example of a puzzle is the necessity of a light to enter a dark room. Tougher puzzles are usually accompanied by a short clue, which helps you hone in on the correct strategy. Once the puzzles are solved, the rest of the strategy revolves around two rules. The first rule to learn is that picking up and eating food increases your energy; the second is that a good kick to the head or a deftly thrown

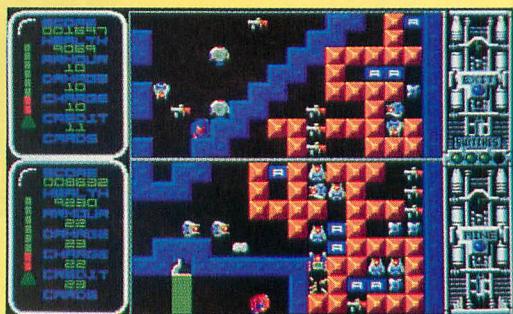


(CONTINUED FROM PAGE 48)

been accomplished, a gong will be heard. Find the room containing four multi colored devices. Touching one of these devices while pressing the joystick button produces a sound effect and activates an LED. Repeat the procedure until all four LEDs are lit and then run for the exit. If you fail to get there in time, the whole process must begin again.

As I said, *Captain Fizz* is probably too difficult to win alone. However, two players will find themselves enjoying the blasting of *Blaster-Trons* and solving the game's puzzles.

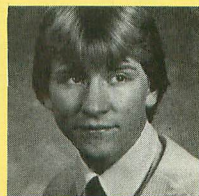
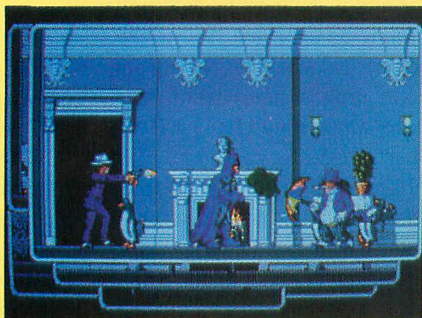
Recommended. A



batarang disposes of troublesome hoods. In the end, careful conservation and timely use of the food pellets spell the difference between a victorious *Batman* and a dead bird.

The six-page manual provides the basics, although you'll suffer from confusion in your first few attempts as you struggle to decide what to do. Actually, the worst aspect of *Batman* is that its two single-minded scenarios limit playability, which I put at only a couple of weeks.

Overall, *Batman* has merit. Its comic book-like graphics do justice to the legendary caped crusader and it is enjoyable to play. The main problem is that once you're done with the two scenarios, you'll be longing, with nowhere to turn, to play *Batman* again. Oh, well; there's always the sequel... A



Steve Panak has written more game reviews for ANALOG Computing than anyone on the face of the earth. He lives in Ohio, where he plays games on his ST and, with the time remaining, practices law.



In addition to being fascinated by computers, **Frank Eva** is an avid Star Trek fan. He has been interested in computer games ever since the release of Pong and has written several text adventures, as well as some educational game software.

THE

ASSEMBLER

/EDITOR REFERENCE

by Matthew J. W. Ratcliff

Iteethed on the Atari Assembler/Editor (Asm/Ed) cartridge way back in 1982. Now, nearly seven years later, it is becoming more popular than ever. Why? Well, I have seen several mail order ads over the past year offering Asm/Ed for only \$10, and sometimes even as little as \$5. But there's a hitch: no documentation. A lot of Atari cartridges were left over from the Warner days when the Tramiel family took over the company. It seems that thousands of cartridges were sold "by the pound," with no boxes or documentation, just to clear out the warehouses. Now many of you Atarians have decided that Asm/Ed was too good to pass up at such a low price, because you just know that someday you will learn assembly language programming.

This article is for you, the Atariian who took the plunge and bought the Asm/Ed cartridge without documentation. I won't pretend to teach you how to write your own

assembly language programs (although I'm bound to toss in a bit of free advice along the way); our *Boot Camp* series will do that job well enough. I hope to give you a quick comparison between BASIC and assembly language to illustrate the major speed differences. I will also cover the mechanics of writing a USR routine to help speed up BASIC programs and a stand-alone assembly program that may be executed from DOS.

The Asm/Ed cartridge will work with any DOS (disk operating system) from Atari DOS 2.0s all the way through DOS-XE and even the SpartaDOS X cartridge. The Asm/Ed cartridge itself is made up of three basic components: the Editor, Assembler and Debugger. I will present a quick reference for all the commands of each section of the cartridge and then lead you through the creation of your first program.

After booting your Atari with Asm/Ed installed and control is sent to the cartridge;

the EDIT prompt will appear. From here you can begin entering your assembly "source code" with line numbers, assembly mnemonics and comments. The editor is line oriented, requiring line numbers. They are used for reference while editing only and are not used as part of the program itself, as they are in BASIC.

The Editor

NEW—The NEW command clears all assembly source code from memory, providing a clean slate for entry of a new program. NEW will irrevocably erase your program from memory, so always be certain to save important code before using this command. I highly recommend using a comment as the first line of every program, similar to the following:

```
10 ;LISTHD:USRTEST1.ASM
```

It's easy to forget what file you are work-

ing on, or make a typographical error when using the LIST command. Saving a file to the wrong place can ruin a lot of work in a big hurry. By placing the LIST command, followed by the correct file name, in a comment, you may simply display that line and use the full-screen editor to eliminate the line number and comment character (the semicolon), press Return, and execute the proper save command every time. This good habit has saved me countless hours of frustration. Use it!

DEL—This command deletes lines of code. The format is *DEL xx*, where *xx* is the line you wish to delete, or *DEL xx,yy*, where *xx* is the first line of a block you wish to delete and *yy* is the last line. Example: *DEL 50,100*.

NUM—Number lines automatically, for fast entry of source code. If no starting number is specified, the program begins with the last line number currently in your program, plus 10. Type your code and comments, pressing Return when each line is complete. Press Return on a blank line to stop auto-entry of code. Do *not* use full-screen editing functions to change lines previously entered while still in the NUM mode. This will give unpredictable results. The command format is *NUM*, to increment by 10 after each line; *NUM nn*, to begin line entry at Line *nn* with an increment of 10; or *NUM nn,mm*, which forces the next statement number to be *nn* and the increment to be *mm*. Example: *NUM 100,20*.

The last command format may be used to insert new lines between code that already exists (e.g. *NUM 11,1* to enter up to nine lines of code between Lines 10 and 20).

REN—Renummer the file. It will resequence all the line numbers of the source file. The command format is *REN*, to renumber starting with 10 and using an increment of 10; *REN nn*, to renumber with an increment of *nn* starting with 10; or *REN nn,mm*, to renumber with an increment of *mm* starting with *nn*. Example: *REN 100,10*.

When inserting a lot of new code, REN may be used to space the line numbers wider apart, thus allowing entry of more new code with the NUM entry method between consecutive lines of source code.

FIND—The FIND command can be used to help locate any string of text anywhere in the program. This command format is *FIND/string/*, to find the first occurrence of

string, and display the line it is found on; *FIND/string/A*, to find all occurrences of *string*. Each line is listed to the display as it is located. Pressing Control-I will halt screen scrolling; *FIND/string/xx* to find *string* on Line number *xx*. The line is listed, if found; or *FIND/string/xx,yy,A*, to find all occurrences of *string* from Lines *xx* through *yy*, inclusive. Example: *FIND/LABEL1/1000,5000,A*.

Note that when searching, the line numbers themselves are ignored (you don't search the line numbers, when looking for a particular number). In the examples above, the string of interest is delimited by the slash (/) character. Any matched pair of characters may be used as delimiters. The following would be used to find all occurrences of the slash character in your program:

```
FIND */*,A
```

REP—Replace strings in the file. The command format is *REP/oldstring/newstring/*, to replace the first occurrence of *oldstring* with *newstring*; *REP/oldstring/newstring/A*, to replace all occurrences of *oldstring* with *newstring* (use the "all" option with care); *REP/oldstring/newstring/xx,yy*, to replace the first occurrence of *oldstring* with *newstring* in the line number range *xx* to *yy*; *REP/oldstring/newstring/xx,yy,A*, to replace all occurrences of *oldstring* with *newstring* in Lines *xx* through *yy*; or *REP/oldstring/newstring/xx,yy,Q*, to replace with query. You will be prompted to press "Y" for each replace. Example: *REP/LDA #\$02/LDA #\$04/100,200,Q*.

LIST—The LIST command is used to display, print or save assembly source code. The format is *LIST*, to list the entire program to the screen; *LIST nn,mm*, to display Lines *nn* through *mm*; *LIST #P:*, to send the entire source program to the printer; *LIST #C:*, to list the source code to cassette in ASCII form; *LIST #D:filename.ext* to list the entire program to the file *filename.ext* on Disk Drive 1; *LIST #D:filename.ext,nn,mm* to list Lines *nn* through *mm* to the file *filename.ext*. Example: *LIST#D:MYPROG.SRC,100,1000*.

Any filename or device specification may be followed by the line number range specification.

PRINT—The PRINT command functions exactly like LIST, except the line numbers

THE Asm/Ed CARTRIDGE

WILL WORK WITH ANY

DOS (DISK OPERATING

SYSTEM), FROM ATARI

DOS 2.0s ALL THE WAY

THROUGH DOS-XE AND

EVEN THE SPARTADOS X

CARTRIDGE.

are not output. Since assembly source files require line numbers, it won't be very useful to PRINT your program to disk and attempt to ENTER it later. This would hopelessly confuse Asm/Ed. Always LIST source code to disk.

ENTER—The ENTER command is used to retrieve a previously listed source file. A valid input device must be specified such as *ENTER#D:MYFILE.ASM* or *ENTER#C:*.

If you wish to merge a program, append a "M" to the ENTER command, like *ENTER #D:ROUTINES.ASM,M*. This merge works the same as the following sequence would in Atari BASIC:

```
LOAD "D:MYPROG.BAS"
ENTER "D:NEWSUBS.LST"
```

The lines are merged. If any line numbers in the file to be merged match those of the file already in memory, the merge file takes precedence. If you wish to append a file to a current working program, it may

be best to ENTER the merge file first, renumber it with some large range such as 20000,1 and then LIST it out to a temporary file. Then ENTER your main program, and finally ENTER with the merge option this renumbered file.

SAVE—Use the SAVE command to write a block of memory, such as an object program, to a file. Let's say your program begins at \$4000 (with an $\text{=}\$4000$ at the top of your assembly code). After the ASM command, you see the final address was \$4IFE. Then, to create a binary image of this file, which may be loaded and run later, enter **SAVE #D:MYFILE.OBJ \times 4000,4IFE**.

Note that the addresses are always assumed to be in hexadecimal, and you do not specify a dollar sign (\$) to indicate this on the SAVE command line. You may also SAVE to the cassette (#C:). With the proper ASM command, your object files may be created automatically, as we will see. Note that you may assemble your program in memory, and then go to DOS and use the DOS memory save command to create this object file as well. (The advantage of the DOS memory save command is that you can specify the RUN address as well, so that your program automatically executes when you perform a binary load. There are ways to set this up with the ASM command as well.)

In BASIC, filenames are enclosed by quotes, such as **SAVE "D:MYFILE.BAS"**. In Asm/Ed, the filename is preceded by the pound sign (#); no quotes are used. Filenames you may use are shown here:

E: The screen editor, used by default with some commands such as LIST.

#P: Refers to the printer.

#C: This is used in reference to the Atari program recorder.

#Dn:filename.ext This is a disk file. The *n* is the drive number, which may be from 1 to 8, depending on the DOS and drive configuration you employ. If no drive number is specified, drive 1 is assumed. The name of the file may be up to eight alphanumeric characters, followed by a period and an optional three-character extender. The extender may be anything you wish. **ASM** or **SRC** is generally used for assembly source files, and **OBJ** or **COM** for executable object files.

ASM—Once you have created your assembly source code and listed it to a file for safe keeping, it is time to assemble it. This is Asm/Ed's primary function, to convert your source code into executable object code. When you issue the ASM command, the current file in memory is

scanned for syntax errors. If it understands all your source code, all the assembly mnemonics are converted into equivalent binary codes and written to memory or a file.

Care must be taken that your code assembles to an area of memory that does not conflict with your source code. Before assembling a program for the first time always enter the SIZE command. Three hexadecimal numbers will be displayed, such as:

```
10F4 1345 9C1F
```

The first number indicates where in memory your source code begins, just above DOS's basic memory requirements. The second number is where, in memory, your source code ends. The final address is the top of usable RAM. At the top of your program will always be a statement similar to the following:

```
10  $\text{*=}\$4000$ 
```

This tells the assembler to start building your object code at memory location 4000 hexadecimal, the program origin. This address may be any number between the second and third numbers reported by the SIZE command, with some notable limitations, when assembling to memory. It may be any value above the first number, so long as you assemble to a file.

If you wish for your program to assemble into lower memory, you may use the LOMEM command. This must be the first command entered, once you start up Asm/Ed and receive the EDIT prompt. The format is **LOMEM xxxx**, where *xxxx* is the hexadecimal address to set the new low memory value. This is the first address value reported by the SIZE command detailed above. For example, if you want your program to load at address \$2400, and you know the object code will be 4K or less, then use **LOMEM \$3400** (\$1000 is 4K bytes). Then ENTER your program, and use $\text{=}\$2400$ at the top of the file to set the origin. Then the program may be assembled in RAM to RAM safely, so long as your object code does not grow beyond 4K.

If you plan to write stand-alone assembly programs, which may be loaded from DOS with the binary-load command (option "L"), I recommend an origin of \$3400. This will set the start address of your code above both DOS.SYS and DUP.SYS RAM in Atari DOS, any version through DOS-XE, as well as any version of SpartaDOS.

Unlike BASIC, you must manage memory yourself. If your program origin is too close to the second number from the SIZE

command, the assembler may get confused. The assembler must build a symbol table and assign some temporary storage as it processes your source code. It starts building this information from the end of your source code and grows upward. If the symbol table runs into the area where the object code is being stored in RAM, the assembler is likely to generate a lot of erroneous phase errors. If your origin is set too high, your object code will run into display memory and eventually run out of room.

These problems may be avoided in several ways. The general form of the ASM command is **ASM #D:SOURCE,#D:LIST,#D:OBJECT**.

The first filename in the ASM command represents the file your assembly source code is stored in. This allows you to assemble from disk (but not cassette, since Asm/Ed requires multiple passes through the file). If this field is empty, simply place a comma immediately after the ASM command; then the source code is assumed to be in memory. The second filename specifies a listing file, where a complete "assembled listing" is routed. This will usually be the printer (#P:). If this field is left empty (you must still include the comma, though), the listing goes to the screen. The listing always goes somewhere; it cannot be turned off as it can in MAC/65. However, assembler directives may be used to control the output of a listing, as we shall see. The third field is the filename where the object code will be stored. If this field is not specified, your program is assembled to memory. Always make a current listing of your program on disk or cassette before issuing the ASM command. If you have set up memory mapping improperly, the source code will get clobbered in a big hurry.

Assembler Directives

Directives, or pseudo operations (pseudops), are special instructions to the assembler. They can be used to control listing format, program title for listing, allocation of memory, and more. In general, assembler directives begin with a period (.), followed by some key word and associated parameters.

OPT—The options directive controls assembler output. They are **.OPT NOLIST**, to suppress the output of the listing during assembly; **.OPT LIST**, to output assembly listing (default); **.OPT NOOBJ**, to not generate object code during assembly; **.OPT OBJ**, to output object code (default); **.OPT NOERR**, to not display error messages while assembling.

bling (there is no good reason to ever use this option); *.OPT ERR*, to display error messages when assembling (default); *.OPT NOEJECT* for no margin at the bottom of each page when outputting the listing; *.OPT EJECT*, to skip four lines at the end of each page (default).

More than one option may be placed on a single line, such as *.OPT NOLIST,NOOBJ*. Note that the MAC/65 assembler defaults to *.OPT NOOBJ*; it does not generate object code unless explicitly told to with the *.OPT OBJ* directive. Asm/Ed is just the opposite. Whenever you are assembling your program frequently, working out syntax and undefined label errors, it is generally wise to have a *.OPT NOOBJ* near the top of your program. When you are ready to generate code and start test running it, then change it back to *.OPT OBJ*.

TITLE and *PAGE*—The title and page directives are designed to make your assembly listings easier to read. The title directive is generally used to specify the name of your program, revision and date. The page directive can be used to force a page break and optionally output some text. For example:

```
10 .TITLE "Attack Of The Dweebies"
20 .PAGE "Program equates"
:
:
200 .PAGE "Graphics Routines"
:
:
300 .PAGE "High Score Routine"
```

TAB—The TAB directive is used to set the spacing of the fields of your assembly code for listings. The command format is *.TAB 12,17,27*. This example illustrates the defaults used by Asm/Ed. These may be set to any position you find most suitable for your printer listings. The first number indicates the field where the mnemonics (assembly op codes) will appear, the second for the operands, and the third for the comment field. For example, suppose your program has a lot of long labels with a maximum of 15 characters. Then you may wish to set the tabs as *.TAB 20, 25, 40*, which would make for a prettier listing on the printer.

BYTE, *DBYTE* and *WORD*—The *BYTE*, *DBYTE*, and *WORD* directives are used to reserve storage in memory, similar to variables in BASIC. Labels may be associated with these directives for easy reference. For example:

```
100 LDA STORAGE
110 LDX STORAGE+1
:
:
```

500 STORAGE .BYTE 34,\$45

In the above the statement, Line 100 will fetch the first value at location *STORAGE*, which is the number 34 following the *BYTE* directive. In Line 110 the X register will receive the data value 45 hexadecimal. Note that the assembler will perform the address arithmetic *STORAGE+1* automatically. The *BYTE* directive may also be used to reserve storage for strings:

```
100 LDA #STRING/$100
110 LDX #STRING&FFF
:
:
320 STRING .BYTE "This is a test",155
```

In Line 320 the *BYTE* directive reserves storage for a string initialized to "This is a test" followed by a 155 (ATASCII carriage return). The code in Lines 100 and 110 fetches the address of the label *STRING*, placing its address high byte in A and the low byte in X. This technique is commonly used to pass the address of data collections (such as strings or data tables) to subroutines.

The *DBYTE* directive reserves two consecutive memory locations, generally used for numbers greater than 256, in high-byte low-byte order. For example:

1000 DATA .DBYTE 258

The above will result in two bytes of memory being reserved at location *DATA*, with the values 1 and 2 respectively (1256 + 2 equals 258). Addresses are stored in low-byte high-byte format as expected by the 6502 microprocessor. The *WORD* directive is used for this, such as:

```
100 *=$3400
110 START LDA #45
:
:
290 RTS ;End of program
300 *=$2E0
310 .WORD START
```

In Line 100 the origin of the program, or program counter, is set to 3400 hexadecimal. The first line of code, with the label *START*, will then be assembled into your computer's memory at \$3400. At Line 300 the program counter is reset to \$2E0. At Line 310 we have the *WORD* directive, immediately followed by the label *START*. The assembler will 'backtrack' as it processes your source code, realize that *START* refers to memory location \$3400, and place this value (low-byte high-byte order) in memory at \$2E0, \$2E1 respectively. This is a special location, commonly referred to by name as *RUNAD* in Atari

memory maps. When you assemble a program to disk which will be loaded and run from DOS, you use the above technique to set the run address of your program. When the program ends with an *RTS*, control is returned to DOS. Most game programs do not end, but you will use this technique for many utilities. As we will see later, a *BRK* instruction is used, instead of *RTS*, when testing programs from Asm/Ed's debugger.

Label Directive—You do not have string, integer and floating point variables in assembly language, the way you do with Atari BASIC. As we saw above, you must set up your own storage and interpret it properly. There are no automatic mechanisms in assembly language for managing variables. To make life easier, you will want to attach meaningful labels to constant values, such as:

```
10 RUNAD = $2E0
:
:
1000 *=RUNAD
1010 .WORD START
```

It is much easier to tell from this example that the intended run address of our program is defined at the label *START*. In the previous example for the *WORD* directive, we simply had the number \$2E0. Unless you want to memorize a lot of memory locations, employ meaningful labels wherever practical.

Labels are used for reference when you want to *GOTO* (*JMP*) or *GOSUB* (*JSR*) in assembly language. For example:

```
10 PROMPT .BYTE "PRESS RETURN
TO CONTINUE",155
:
:
100 LDA #PROMPT/$100
110 LDX #PROMPT&FFF
120 JSR PRINTSTRING
:
:
500 PRINTSTRING STA ICBADR+1
510 STX ICBADR
```

Labels may also be used in branch instructions such as:

```
50 CONTINUE LDA TABLE,X
:
:
100 DEX
110 BEQ EXITLOOP
120 BNE CONTINUE
130 EXITLOOP STA RESULT
```

In the above example, we have set up a loop, similar to a BASIC *FOR/NEXT* loop, between the labels *CONTINUE* and *EXITLOOP*. In Line 100 the X register is decremented by 1 (we assume it was initialized by some code previous to Line 50). If the result of the *DEX* instruction is zero (*BEQ*) then control will be passed to *EX-*

ITLOOP. If the X register has not gone to zero (BNE) the control is sent back up to CONTINUE. As a result of DEX, the zero flag can only be set (BEQ) or cleared (BNE), so we have exhausted the possibilities. It would have been equally valid to use:

```
120 JMP CONTINUE
```

Generally, whenever you have the choice between a JMP and Bxx (branch) instruction, use the branch. It will require less memory and work faster. The problem is that a branch is limited to plus or minus 127 bytes from the current position. If you try to branch too far, you will get an assembly error. Then JMP instructions, or combinations of JMP and branch instructions may be required.

Origin Directive—We have already used the origin directive “[]=” in many of the previous examples. This tells the assembler “set the program counter to the following address.” The address may be some number, or a label, or some expression (so long as the assembler may resolve it to a fixed value). Some examples are:

```
100 *=$3400
300 START = $4400
310 *=START
:
```

```
:
500 *=START + 439
:
600 HERE *=*+45
```

Take note of the spacing used in all of our examples. Any label always begins one space after the line number, referred to as the label field. The op code field begins at least one space after the start of the label field. If a line of code has no label on it, then your assembly mnemonics may begin two spaces after the line number. At least one space after the op code field will begin the operand field. This field is optional since not all assembly mnemonics have an op code (such as DEX or INY). Anything after the operand field is ignored by the assembler and assumed to be the comment field. A comment can take up an entire line, when the label field begins with a semicolon.

IF—The IF directive is used for “conditional assembly.” This may be used to enable or disable the generation of some test code, for example, based on the value of a number, label or expression. For example:

```
10 DEBUG = 0 ;Enable test code
:
300 .IF DEBUG @ENDOFDEBUG
310 ; Debugging test routines
```

```
:
500 ENDOFDEBUG
```

If the expression (DEBUG above) is equal to zero, then everything from the line following the IF directive to the specified label (ENDOFDEBUG) is assembled. When you are satisfied that your code works, don't throw away all that useful testing code. Simply change Line 10 to *DEBUG=1* and reassemble your program. If you do not understand the use of conditional assembly, don't worry. I have only used it a few times in the past seven years, and generally you don't need it at all.

END—The Asm/Ed manual recommends that every program have one .END directive, as the last line. It really isn't necessary, since the assembler knows when to stop (it runs out of source code to assemble). If you place an .END in the middle of your program inadvertently, all code after it will be ignored and not assembled. I seldom use a .END in any of my assembly code.

More to Come

Next month, we'll finish up our quick reference to the Atari Asm/Ed cartridge by discussing error codes, expressions and the debugger, among other things. ☐

(CONTINUED FROM PAGE 24)

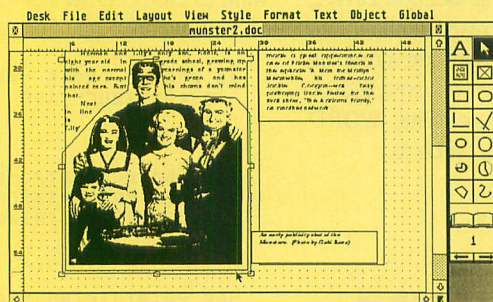


Figure 4

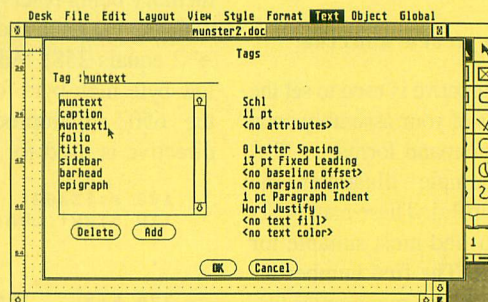


Figure 5

3 Not all of my design suggestions were carried out. For example, the “TWO COLUMN SIDEBAR” head shown here didn't make it into the finished book because I forgot to write up the specs. The scan of Al Lewis was done with PictaScan at screen resolution, the only resolution available with this otherwise excellent scanner. For this project, such a scan would have been acceptable since it was “For Position Only.” (All illustrations of the Munsters in these samples are copyright 1989 by MCA Publishing Rights, a Division of MCA Inc. Used with permission.)

4 Here, I've just set the text wraparound for the polygon I created around the Munster

family. I made it large so I could click on either the picture or the shape to help position them. The next step is to call down the Object menu and choose zero for the Line choice, leaving the polygon invisible except for the graphic itself. Open polygons can be very useful for creating unusually shaped text.

5 Here's a snapshot of my tags for the *Munsters* book sample pages. Note the “Muntext1” entry: To get the first paragraph of every chapter to run without an indent, I made a duplicate of the “Muntext” tag, without the paragraph indent. Get your tags set up before working, and life with *PageStream* is much easier. ☐



Donovan Vicha has been writing about desktop publishing on the Atari ST for three years. He uses a Mega ST2 system for his freelance editorial service, adapting 12 years of book-publishing experience to this rapidly growing field. He lives in Chicago with his wife and two sons.

ATARI

GIFT GUIDE!

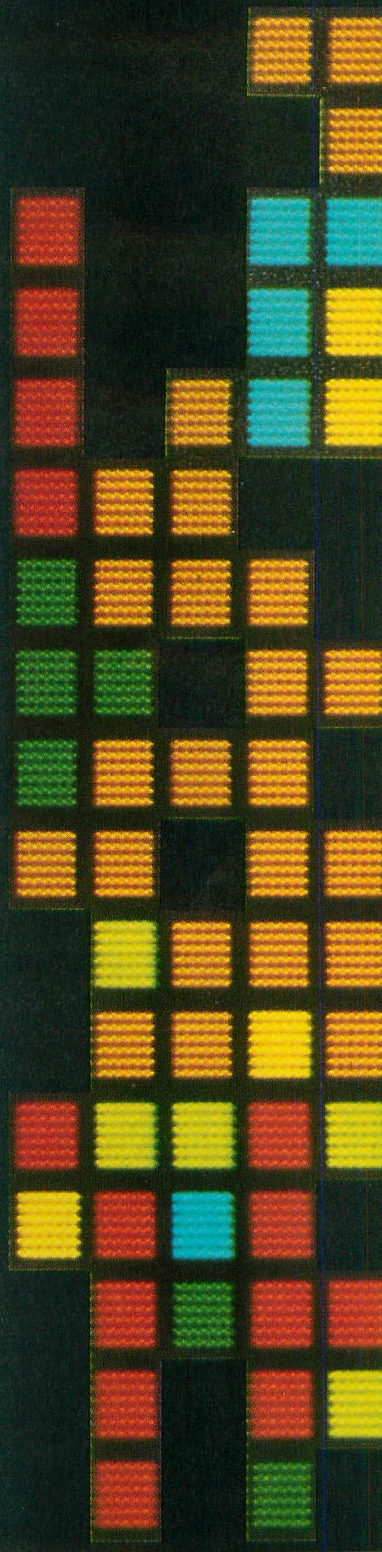
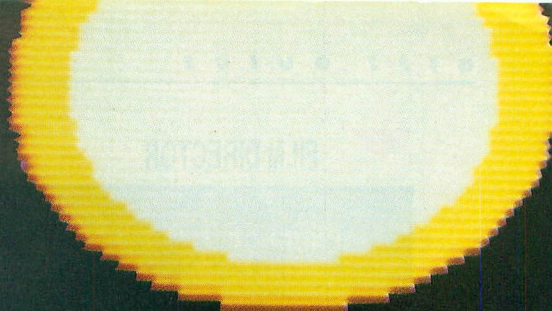
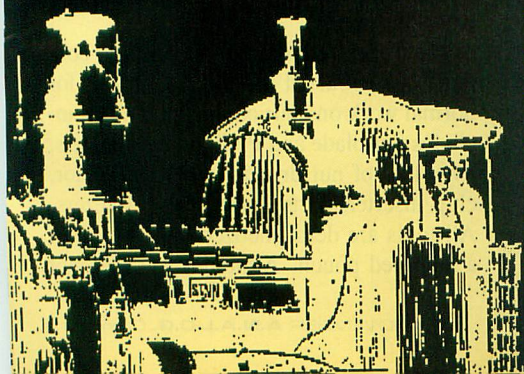
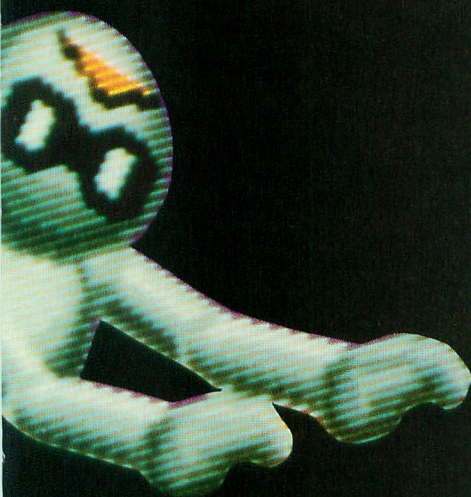
BY ARTHUR LEYENBERGER

It's that time of year again. Atari 8-bit and ST users everywhere are starting to think about what program or peripheral will make the perfect gift, either for themselves or for someone else. There are literally thousands of currently available products that will make any Atari system complete. These products will help you get the most out of your Atari computer.

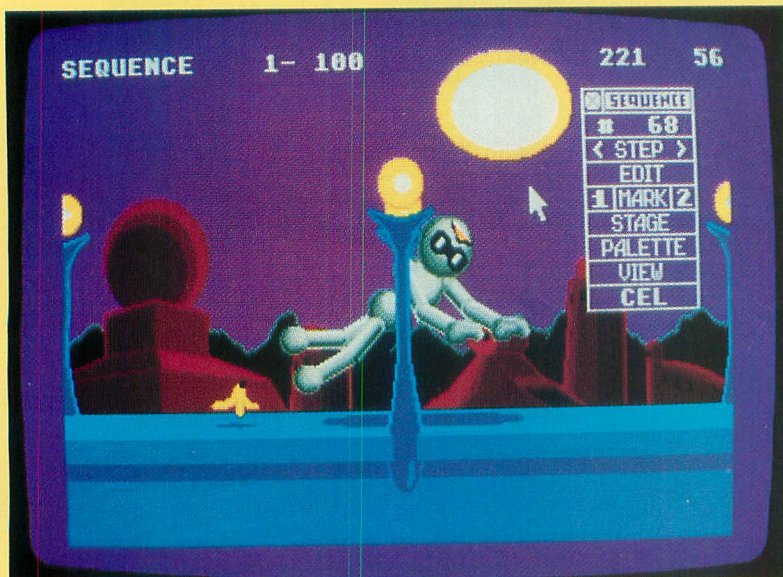
The following, then, is ANALOG's annual gift guide. Here you will find the best and most useful products for your computer. Some products are mentioned specifically by name, while other product categories are discussed generally. A list of the manufacturers and their addresses can be found at the end of the article.

Space does not permit me to mention every Atari-related product ever made. Nor can I even mention all of the good Atari products I use or know about. My apologies in advance if I leave out your favorite. However, I do know what the needs of the typical user are, since I too am a user. Further, having written about Atari computers for the past five years, I am well-informed about existing products.

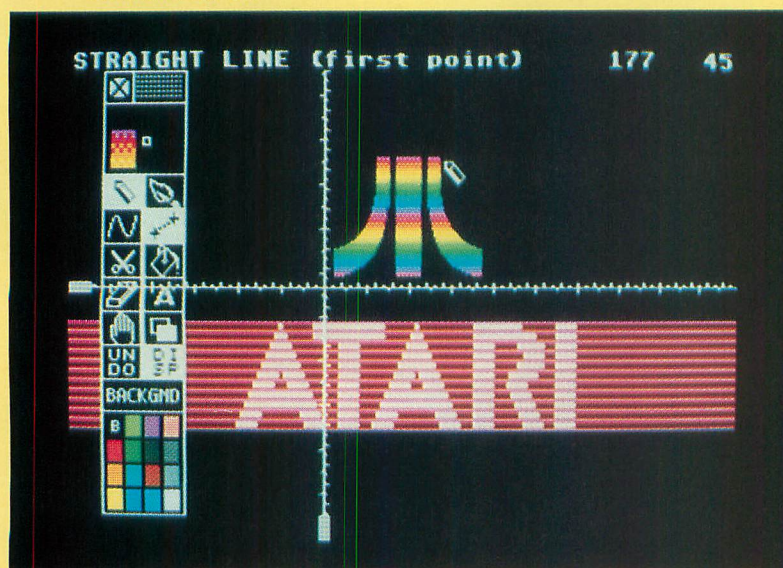
Products are listed in order of occurrence: that is, the order in which they occurred to me. First, generic products and services that are useful to both 8-bit and ST users are described. Next, 8-bit software is mentioned, followed by ST software. I guarantee that you can use any of these products and still respect yourself in the morning.



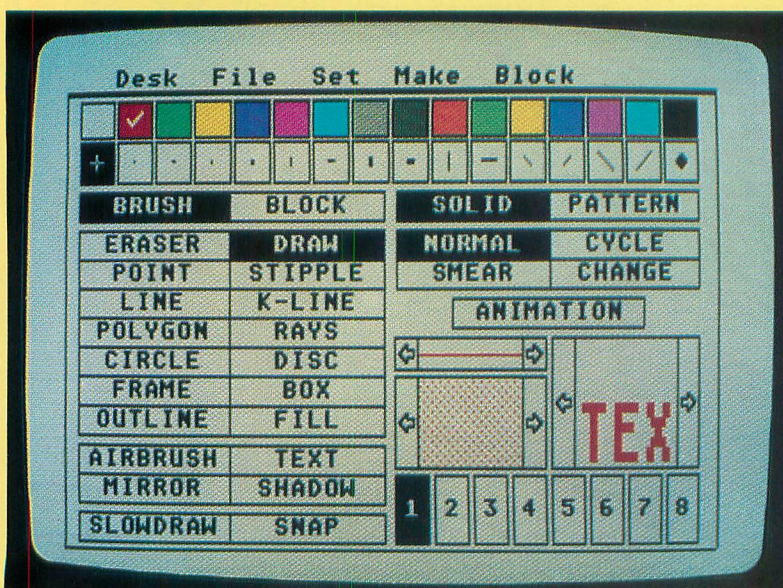
FILM DIRECTOR



ART DIRECTOR



DEGAS ELITE



GENERIC GIFTS FOR ALL ATARI USERS

There are many generic gifts useful to an Atari computer user. One of the best you could give or receive is a membership to an Atari Users' Group. There are many fine groups around the country, and most clubs offer both 8-bit and ST special-interest groups. They also have knowledgeable people who are willing to help the other guy out.

Atari Users' Groups have monthly meetings, publish newsletters and often have libraries of public-domain software. Meetings typically include question-and-answer sessions, hardware and software demonstrations and even guest speakers from Atari and various software companies. Many also have electronic bulletin boards for their members' use.

Membership to an Atari Users' Group typically costs about \$20 and provides one of the best sources of information about using your ST and 8-bit computer. To locate the nearest Atari Users' Group contact Atari Corporation directly. Whether you are just starting out with your new Atari computer or are a seasoned veteran, joining a local user group is worthwhile. The Washington Area Atari Computer Enthusiasts publishes one of the best user-group publications in the country. Their newsletter is called *Current Notes* and looks more like a magazine in size and content. Each issue features product reviews, special reports and tutorials. It has a glossy cover, is well-written and looks professional. A ten-issue subscription costs \$24 and can be obtained by sending a check payable to: Editor, *Current Notes*, 122 N. Johnson Road, Sterling, VA 22170. *Current Notes* is an example of a high-quality user-group publication.

Curtis Manufacturing has a number of products for users of any computer. Two products that I use are the Curtis Clip and the Computer Tool Kit. The Curtis Clip is a simple but useful plastic paper holder that attaches to either side of your monitor. The \$6 product will hold a single sheet of paper at eye level for easy typing or reference. In addition, the clip swings out of the way when not in use.

The Curtis Tool Kit lists for \$30 and contains just about all the tools you need for normal work on your computer. The kit includes flat-blade and Phillips screwdrivers, two sizes of nut drivers, a chip extractor, chip inserter, parts grabber and tweezers. All tools are demagnetized and housed in a zippered pouch. Although you may not

need every tool, it's good to have them available and in one place.

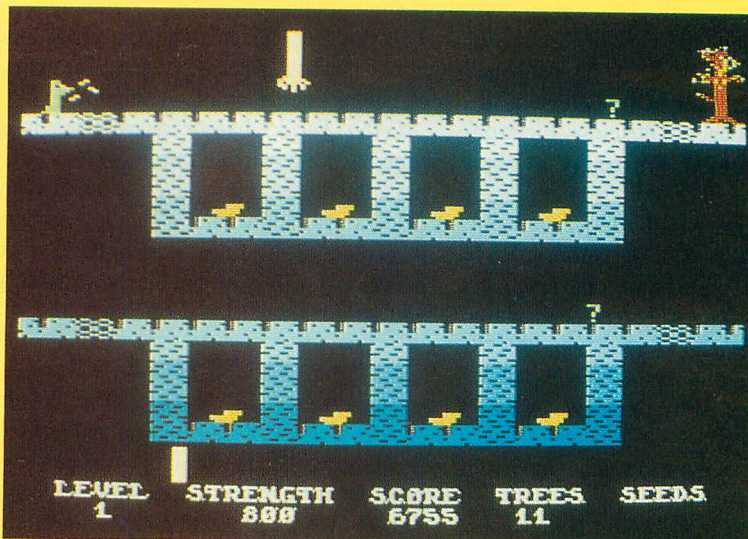
Your Atari computer system represents a sizable investment in hardware. Unless you protect it with a surge protector, your equipment is susceptible to the power fluctuations of your local power company. Temporary power surges, spikes, too-high or too-low voltage can seriously damage your hardware. A number of devices are sold to offer protection from these kinds of catastrophes, but their quality ranges from poor to excellent.

Panamax offers a complete line of high-quality products that work as advertised. Their products include multiple-outlet strips, power-line conditioners, telephone-line protectors and under-voltage protectors. Various features are offered on the different models, such as circuit breakers, alarms and on/off switches. Panamax's models range from \$50 to \$150.

One of the best ways to participate in the world of Atari computing is by using the DELPHI telecommunications service. DELPHI is an online information service which, among others things, hosts two Atari Forums: one for 8-bit users and one for ST users. Both forums are operated by the editors and staff of ANALOG.

A host of services and features is available on the DELPHI Atari Forums. Electronic mail can be sent and received by Atari 8-bit and ST users worldwide. Thousands of free programs can be downloaded from the database areas, including the programs presented each month in ANALOG. The conference feature allows you to meet electronically with other Atari users, and there are regular formal online conferences where you can discuss important Atari-related topics. Access charges average as low as ten cents per minute from most parts of the country, and there is no extra charge for high-speed access.

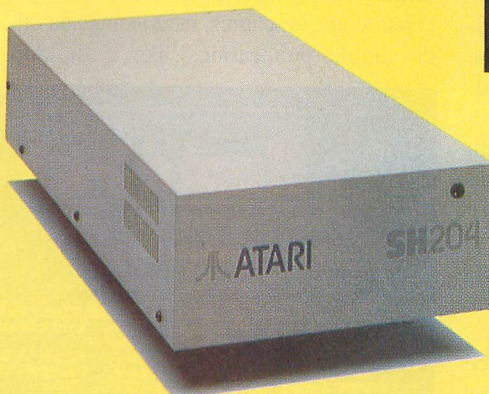
Other popular information services include CompuServe and GENie. To use DELPHI, CompuServe, GENie or any of the other information services, you will need a modem and a terminal program. An excellent choice for a modem is the Supra 2400. This is a fully Hayes-compatible 2400-baud modem that features auto-dial and auto-answer capabilities. Configurations are set via commands from the keyboard instead of by using DIP switches. The Supra 2400 has a list price of \$140. You'll need an RS-232 interface, such as the Atari 850 or P/R: Connection if you have an 8-bit computer.



NECROMANCER



HOME TERM



ATARI SH204

For the 8-bit Ataris, the terminal program included with HomePak is an excellent choice. Good public-domain terminal programs include *Express!* and *Amodem 7.2*. If you have an ST, I recommend *Interlink* by Intersect Software. This program offers an enhanced type-ahead buffer that allows you to create and edit text on the screen before it is sent. In addition, you can capture your entire online session (Autolog), with the capture being saved to disk automatically.

8-BIT SOFTWARE

It seems there are thousands of products

made for the Atari 8-bit computers. Although the supply of new products has dwindled in recent years, there are many quality programs still available. Regardless of whether you have a 400/800, XL, XE or XE Game System, these programs allow you to use your computer for education, entertainment and productivity purposes. Here are a few suggestions for your consideration.

APPLICATION PROGRAMS

Word processing remains the foremost category of software for any computer, and there are several excellent choices for the

Atari 8-bit computers. One of the best of the bunch is *PaperClip*, originally published by Batteries Included. Although Batteries Included is no longer in business, Electronic Arts has bought the rights to their products.

PaperClip runs on all Atari 8-bit computers with at least 48K of memory. Since it was written entirely in machine language, it is fast. Two different files can be displayed on the screen at the same time, and blocks of text can be moved between the windows. In addition, each editing window can be set to a different size. Further, a full page (greater than 40 columns) can be previewed on the screen.

PaperClip has many useful features that assist the user. The program has one-key macros that can be used to add a word or phrase with the touch of a single key, an auto-save feature that automatically saves the text file you are currently working on after a predefined number of keystrokes, and the ability to use DOS commands (directory listings, LOCK, UNLOCK, ERASE and RENAME) from within the word processor. Other features include extensive HELP, word-count display, UNDO command, global replace on chained files and printer-configuration files for most popular printers that are used with Atari computers. (You can also create your own configuration file.)

Electronic Arts distributes another original Batteries Included product that may be one of the best 8-bit software values around. Called *HomePak*, it is three programs in one: a word processor, terminal program and database. *HomeText* is an excellent entry-level word processor, but does not have the dozens of features contained in *PaperClip*. *HomeTerm* is one of the best terminal programs for the 8-bit computers. *HomeFind* is an electronic filing system.

Other word processors to be considered include *Atariwriter Plus*, *The Writer's Tool* and *Letter Perfect*. *Atariwriter Plus* is a descendant of the original, excellent *Atariwriter* word processor and maintains command similarity with its predecessor. It's a full-featured word processor that offers block functions, spelling correction (via a separate program), predefined and customizable printer drivers, text-formatting and word counter. In addition, a scrollable window is used to show text that is wider than 40 columns.

The Writer's Tool was originally published by OSS but is now being distributed by ICD. It is a high-end word processor that has more features than any other similar

8-bit program. Although it takes a little time to master, it is especially useful for someone who does a lot of writing.

Letter Perfect has been around a long time. Although LJK is no longer in business, the program can still be found at many dealers and represents a quality product with plenty of features. Any of these word processors will give you all the capability you could want in an 8-bit program.

GRAPHICS

The Print Shop from Broderbund Software is a classic 8-bit program that lets you create letterheads, posters, banners and greeting cards. Once you finish your design (either from canned samples or customized), the output can be printed on a variety of dot-matrix printers. The program contains all of the elements (graphic symbols, fonts and forms) you need to produce any design and is easy to use. All you have to do is step through the various menus and choose whatever options you like. The package also includes a supply of colored paper and envelopes.

Broderbund also sells several companion products for the *Print Shop* called the *Graphics Library*. There are three volumes available, each containing additional clip art, fonts and designs.

Another excellent graphics program is *The Newsroom* by Springboard. It lets you create signs, brochures, newsletters and other simple publications like a desktop-

can be purchased separately. *The Newsroom* is not a desktop-publishing program per se, but can be used like one. It features automatic text flow around graphics, a logical menu-based interface and support for over 50 printers.

If you have used more than one of the 8-bit graphics programs, you have no doubt realized that many of them have their own file format. Files created by one program cannot generally be used by another. Now there is a way to convert graphics files from one format into another. The program is called *Graphics Transformer* by Alpha Systems. The \$25 program can read *Koala Pad*, *Atari Light Pen* and *Touch Tablet*, *Print Shop*, *Paint*, *Graphics Master*, *Movie Maker*, *B/Graph*, *Super Sketch*, *MicroPainter*, *MicroIllustrator*, text modes 0, 1 and 2, *Computer Eyes* and many more graphics format files. It can then convert these files into one of the following graphics files: *Typesetter*, *MicroPainter*, *MicroIllustrator*, *Print Shop*, *Koala*, *Atari Touch Tablet* and *MagniPrint II+*.

There is another handy product from Alpha Systems called *MagniPrint II+*. This print-utility program lets you print all graphics modes files (including GTIA modes with 16 shades), enlarge or shrink pictures, print in normal or inverse and print selective areas of a picture. In addition, the program will print horizontally or vertically, read a variety of graphics files and allow you to change graphics modes for



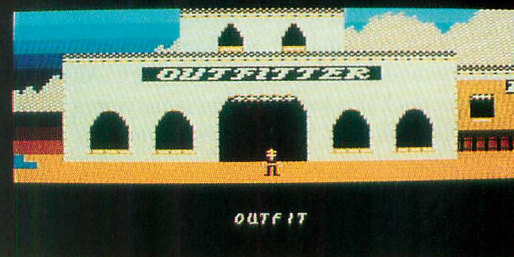
HOME PAK

SEVEN CITIES OF GOLD

publishing system. As such, you must create and assemble each of your pages with text, clip art and graphics. Over 600 graphics are included with the package, and more

special effects.

An additional supplied program will print BASIC program listings complete with graphics and inverse characters. *MagniPrint*



II+ supports Epson, Panasonic, C. Itoh, NEC, Gemini and Star printers and sells for \$20.

GAMES

My list of recommended 8-bit games would probably fill this entire magazine. However, there are a number of outstanding titles of which any would make an Atari user happy. Many games are no longer sold by the original publisher either because they have gone out of business or have sold the rights to someone else. In addition, a number of games have been repackaged as "budget" titles on disk or as cartridges.

One of the most enduring and endearing 8-bit games is *Flight Simulator II* by subLOGIC. This excellent flight-simulation program is one of the most complete examples of the genre I have seen for any computer. The game consists of a single-engine plane simulation where you can either fly prescribed courses or practice the many aspects of flight. Excellent documentation accompanies the program and additional course disks can be purchased. Highly recommended.

My favorite text-adventure game is still *Planetfall* from Infocom. This game combines adventure, science fiction and humor into an engrossing example of what is called interactive fiction. For a child or someone new to the text-game genre, I would recommend *Seastalker*, also from Infocom. It is not as difficult or complex as *Planetfall*, but offers a good challenge and, like all Infocom games, has excellent documentation. Every Atari owner should own at least one Infocom text-adventure game. Prices range from \$30 to \$50.

Electronic Arts has lowered the price on many of their 8-bit games and now markets them under the "Classics" category. They now typically sell for under \$15. One of the most creative and innovative games they ever produced was *Pinball Construction Set*. Written by Bill Budge, this game lets you create your own pinball games from a video parts box. You can then edit and save those games for future play.

Another classic from EA is *Seven Cities of Gold*. It is an excellent example of what an 8-bit game can be. You play the role of a 16th-century Spanish explorer, searching for new worlds, peoples and glory. Your own personality and belief system determine how you play the game, and the hidden feature of the game is self-awareness.

Atari has bought the rights to a couple of dozen 8-bit games and repackaged them in cartridges. These XE Game cartridges are

marketed toward the XE Game System, but they will work in any 8-bit Atari computer. Some of the better titles include *Archon* (Electronic Arts), *Necromancer* (Synapse), *Mario Brothers* (Atari), *Choplifter* (Broderbund), *Ball Blazer* (Lucasfilm Games) and *Rescue on Fractalus* (Lucasfilm Games). The XE Game carts sell for \$25 to \$35.

MISCELLANEOUS GOODIES

The P:R: Connection from ICD represents one of the best ways to satisfy your 8-bit interfacing needs. It allows you to connect a Centronics parallel printer and two RS-232 devices to your 8-bit computer. Both the parallel and serial ports are virtually identical to that of the standard Atari 850 interface. Since the P:R: Connection takes its minimal power needs from your computer, no separate power supply is necessary. The P:R: Connection sells for \$70 and includes a free terminal program with the package. However, one of its main advantages is that it allows you to use industry-standard products, such as RS232 modems and Centronics printers, which can be used with other computers as well.

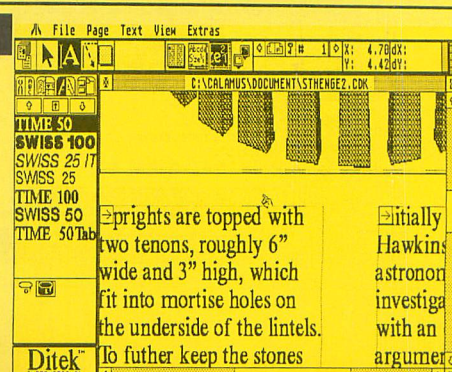
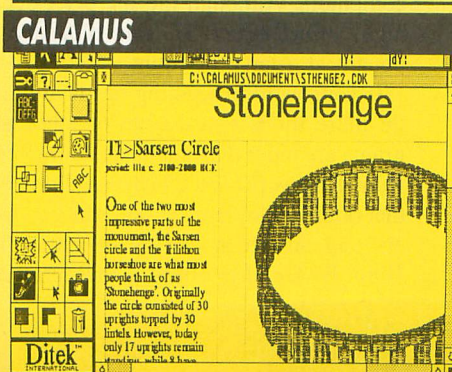
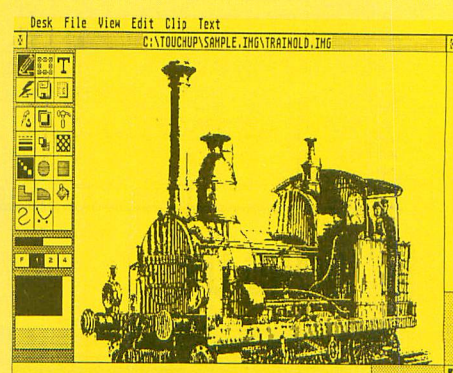
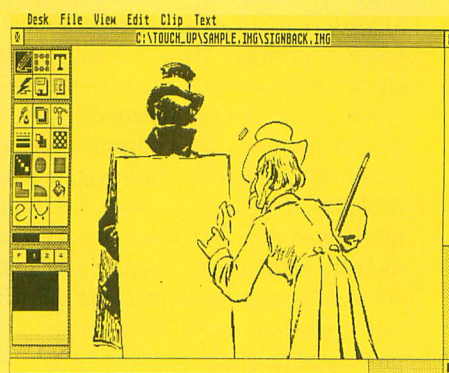
There are a number of 8-bit disk operating systems currently available. One of the most powerful is *SpartaDOS X* from ICD. *SpartaDOS X* comes in a special 64K cartridge and gives the Atari computer a full complement of DOS commands. The cartridge consumes none of the computer's memory and also has the provision to accept another cartridge.

Features of *SpartaDOS X* include the ability to create subdirectories, directly change the status any of a number of file attributes, display the contents of the file and set the time and date of the *R-Time 8* cartridge (if you have it) or the system clock. It also provides other file commands, such as COPY, FIND, ERASE, RENAME and DIR. Other features include a batch file and archive utility, the ability to switch to internal BASIC on an XL or XE computer and I/O redirection. *SpartaDOS X* sells for \$80.

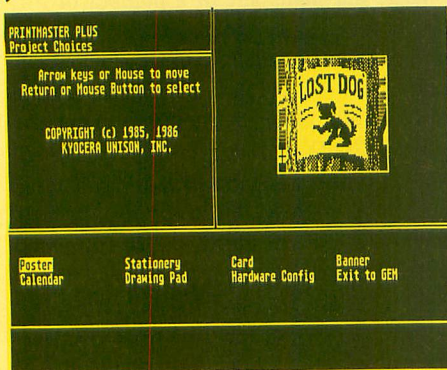
How would you like to have a graphics interface for your Atari 8-bit computer? It is possible with *Diamond GOS* from Reeve Software. *Diamond GOS* (Graphics Operating System) is a "piggy-back" cartridge and disk software that provides a GEM-like Desktop on your 8-bit screen. The GOS allows you to have multiple windows open at once, where you can run programs and perform maintenance activities, such as copying and erasing files, formatting disks, viewing the contents of disks, etc. Also included with *Diamond GOS* is *Diamond Paint* and utility disk. The entire package sells for \$80 and requires an ST mouse.

Educational computing is another category in which the Atari 8-bit excels. Although it has been available for several years, the *AtariLab* science kit allows you to perform experiments with the computer. The *AtariLab Starter Set* consists of a hardware interface device that plugs into a joystick port, a 16K ROM cartridge contain-

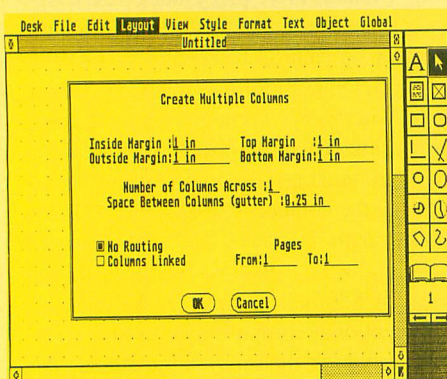
TOUCH-UP



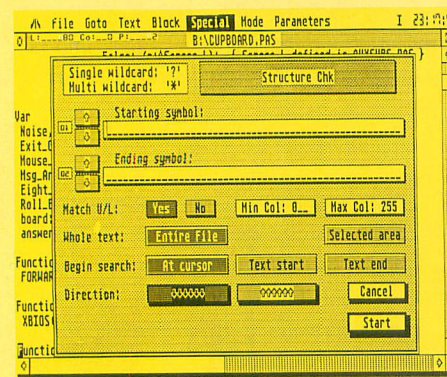
► PRINTMASTER PLUS



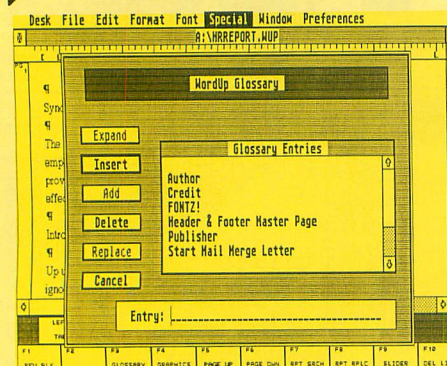
► PAGE STREAM



► TEMPUS



► WORD UP



ing the temperature program, a temperature sensor, a standard thermometer and a 140-page manual. Users can make observations, take temperature measurements and then analyze the information.

The *AtariLab Light Module* kit is a companion product that requires the starter kit. It contains a cartridge, manual, light sensor, polarizer, filters, light stick and other items. The *Light Module* contains experiments to observe the light level of a candle, measuring response time of the light sensor, measuring how light passes through various materials and how polarizers work.

The documentation included with both modules stresses the scientific method and the importance of such things as calibration, repeated observation and proper data analysis techniques, such as graphing. The manuals also contain detailed information about the hardware interface itself and how it may be used with your own BASIC or Logo programs to measure real-world information or control outside devices, such as relays and switches. The *AtariLab Starter Set with Temperature Module* sells for about \$50. The *Light Module* costs about \$40.

ST SOFTWARE APPLICATIONS

The most frequently used application program on any computer is word processing, and there are several excellent word-processing programs for the ST, the most sophisticated of which is *WordPerfect* from WordPerfect Corp. This \$250 program provides all of the normal functions one would expect, such as BLOCK/MOVE, SEARCH/REPLACE, headers/footers, column printing (two kinds), spelling checks, a thesaurus and much more. In addition to the normal functions, such advanced features as automatic backups, paragraph numbering, outlining, automatic hyphenation, multiple undo levels, multiple on-screen windows (up to four), footnoting/endnoting, indexing and table of contents generation are included. *WordPerfect* competes easily with major word-processing programs available for MS-DOS computers. Although the list price seems high, *WordPerfect ST* can typically be bought at a large discount.

If your word-processing needs are a little more modest, a couple of other programs are more than adequate. *WordWriter* from Timeworks is a GEM-based program that offers multiple windows, headers, footers and on-screen display of text styles. *Word Writer* also offers a thesaurus and

spelling checker. The spelling checker provides multiple personal dictionaries and can check an entire document or individual words as you type.

WordWriter sells for \$80 and is easy to learn, thanks to the drop-down menus with equivalent key commands. In addition to being a stand-alone word processor, *WordWriter* interfaces with other Timeworks products, such as their database program (*Data Manager*) and spreadsheet (*Swiftcalc*). Text files from *1st Word* can be imported, and printer drivers are provided for a variety of printers.

Another excellent ST word processor is *Word Up* from Neocept. *Word Up* uses GDOS to produce high-quality output containing multiple fonts and different type sizes. Up to four windows can be open at once, each containing a different file. Other features include extensive block functions, headers, footers, endnotes, automatic file backup and the ability to incorporate graphics (*DEGAS*, *Neochrome* and *IMG* files).

Although *Word Up* doesn't contain all of the features of *WordPerfect*, it will print the highest resolution your printer is capable of. Further, it is an easy program to use. At \$80, *Word Up* should definitely be considered when you are looking for a word processor to call your own.

A number of desktop-publishing programs are available for the ST. By combining the three elements of word processing, page layout and drawing into one program, you can easily design, compose and paste up the contents of a printed page for forms, newsletters, charts, etc. You normally see exactly what you will get on the screen be-

Another good game from Electronic Arts is Zany Golf. More like an arcade game than a golf simulation, Zany Golf is like a miniature golf course from hell.

fore it is printed. These programs range from simple-to-use entry-level products to more complicated programs intended for professional users.

One of the most popular desktop-publishing programs is *Desktop Publisher ST* by Timeworks. An easy-to-use program, it includes fully integrated word processing, page layout, typesetting and graphics functions all in one package. Some of the major features of the program include a full GEM interface with pull-down menus; icons; scroll bars and dialog boxes; flexible page layout to let you overlap, reposition and resize your text; columns and graphics; built-in fonts that range from seven to 72 points; and high-quality output to a dot-matrix or laser printer.

Text can be imported from *1st Word Plus*, *Word Writer ST* or any ASCII file so the program can work with existing word processors. Graphic files can be imported from *DEGAS*, *Neochrome* and *Easy Draw* file formats. Further, you can view, edit and layout pages in actual, double or half-size windows. *Desktop Publisher ST* sells for \$130.

A professional-level desktop-publishing program is *Calamus* from ISD Marketing. The program provides text and document processing, design tools and page-layout capabilities that rival and often surpass those found in MS-DOS desktop-publishing programs. It uses a full GEM interface with mouse control and drop-down menus, or it can be used with keyboard commands. Vector fonts rather than Postscript-type fonts are used, and the program provides many features, such as outline fonts, printer preview, automatic kerning and stretching.

Other features of *Calamus* include automatic text flow from one page to another; automatic kerning in all fonts and sizes; word, character and line spacing adjustments; page and chapter numbering; text import and export; built-in macros and much more. *Calamus* can also create output in PDL, DDL, interpress and linotype formats. It sells for \$300.

PageStream from SoftLogik is another high-level desktop-publishing program. It incorporates typographical features, word processing, page layout and graphics. It also offers such capabilities as color-separation of pictures within documents and color printing. Word-processing features include spell checking, search and replace by attributes, automatic kerning, scalable fonts and text flow around irregularly shaped graphics. Text rotation can be performed in one-degree increments and text point sizes range

from 1/50th to 1,310 points. *PageStream* sells for \$200.

If you have any programs that use Atari's GDOS (graphics device output system), such as CAD or desktop-publishing programs, you know how slow the output can be. Further, you also know that GDOS is a memory hog, is temperamental (it must be unloaded when a non-GDOS program is run) and is unwilling to let you change fonts and drivers after it has loaded. Thankfully, CodeHead Software has created *G+Plus* as a replacement to the original GDOS.

G+Plus is an entirely distinct product that remedies the problems users have had with GDOS. Although it acts like GDOS (in terms of function), it offers better compatibility with non-GDOS programs, increases the speed of output and increases the speed of program execution (with those that need GDOS functions). In addition, it allows you to switch ASSIGN.SYS files from the desktop without having to reboot the computer. *G+Plus* sells for \$35.

GRAPHICS

DEGAS Elite is one of the best values in ST graphics software and has become a standard. The program allows you to paint in any of the three ST graphics modes and save your files to disk. *DEGAS Elite* contains a multitude of features, including: the use of eight work screens; the ability to cut and paste all or parts of these screens to each other or to disk; the ability to create and use clip art; block operations, such as stretch, rotate, horizontal or vertical skewing and distortion. Of course, the neat features involve color animation. Complete control of the direction, speed, colors and other aspects of the animation is at the user's command. *DEGAS Elite* retails for \$50 and is distributed by Electronic Arts.

If you use any of the ST graphics programs, such as *DEGAS* or *Neochrome*, you may be interested in an animation program that makes your artwork come to life. Called *Make It Move*, this \$50 program from Michtron is really more of an easy-to-use slide-sequencing and -manipulation program than an animation program. Nonetheless, *Make It Move* is great for such tasks as creating video titles, making presentations and polishing your collection of ST graphic images. The program is mouse-driven, easy to use and doesn't require any special programming skills. Features include screen wipes in any of four directions, fades and quick cuts, each with

user-specified durations. Further, objects may be zoomed, panned, hidden and moved.

Another graphics program that allows you to be creative is *PrintMaster Plus* from Union World. With it, you can easily create calendars, fliers, stationery, banners, invitations, signs and greeting cards with many types of printers. The \$40 program is menu-based, allows you to preview your design before you print it and permits you to save your graphic to disk. In addition, *PrintMaster Plus* will work with add-on font and graphic disks.

Still another excellent graphics program is *Art and Film Director* by Epyx. This \$80 package includes *Art Director*, a full-featured paint program that uses all the power of the Atari ST to create dazzling works of art. Menus and icons appear on-screen that allow you to choose among a variety of shapes, lines and colors from the palette, as well as perform such actions as stretch, bend, bulge, spin and rescale to enhance the pictures.

Film Director then uses true cel animation to transform artwork into stunning graphic presentations. It's easy to automate many of the repetitious steps required by traditional types of animation. The program also includes a library of music and sound effects to create just the right ambiance.

Printers are capable of printing in resolutions of 100 dpi (9-pin printers) to 300 dpi (laser printers). However, graphics programs, such as *Neochrome* and *DEGAS Elite*, can only create graphics as good as the screen image (a 640x400 monochrome screen resolution translates to about 80 dpi when printed). With *Touch Up* from Migraph, you can edit and print these graphics images with as high a resolution as the printer can handle.

One of the main features of *Touch Up* is the ability to convert 16-color low-resolution pictures into image files (color mapping). Six different ways to do this are provided and each produces better results at the expense of requiring more time to accomplish. For example, this procedure allows you to first convert a low-resolution graphics image created in *DEGAS Elite* into an image file and then import that image file into a desktop-publishing program. Graphics file conversion is also available to and from *DEGAS*, *DEGAS Elite*, *Neochrome*, *MacPaint*, .TIF, .PCX, .IMG and .GEM formats.

The \$180 *Touch Up* program also contains full-featured drawing capabilities for bit-

mapped images. A variety of drawing tools, such as geometric shapes, lasso, fill patterns, line sizes, ten fonts (with attributes), etc., is provided to create and edit images. Advanced functions, such as Spines and Bezier curves, are also included. All in all, *Touch Up* is a sophisticated yet easy-to-use graphics utility program.

UTILITY PROGRAMS

Utility programs are the glue that holds various ST application programs together. They also provide a means to make your computer more productive. Whether you need to convert files from one format into another, perform routine maintenance on your system or just want to be able to use all of the capabilities of your ST, utility programs are a necessary component of your software library.

MultiDesk allows you to use any desktop accessory program on your ST at any time without rebooting the computer. In addition, you are not limited to the six desktop accessories (.ACC files) loaded at once when the computer boots up. MULTI-DESK.ACC is loaded as one of the six possible accessories when the ST boots up in the usual manner. However, from within *MultiDesk*, you can access up to 32 additional accessories.

MultiDesk works as either a desktop accessory or as a regular GEM program. When run as a program, it gives you access to any desktop accessory whether it has already been loaded or not. *MultiDesk* sells for \$32 and is an excellent utility.

If you have a hard disk, the \$60 Michtron *Hard Drive Turbo Kit* will be an essential

utility program. The *Turbo Kit* contains three hard-disk products (upgraded versions of previously released separate programs) in one: backup, optimize and disk cache. Each of the programs has a variety of features and options to simplify and enhance its use. The backup utility allows you to perform either a file-by-file or image (sector-by-sector) backup of the hard disk. You can produce a full backup of the entire hard disk or an incremental backup of only those files that are new or that have changed since a specific date. A restore program allows you to copy the files from the floppies back onto the hard disk in the event of a hard-disk failure. *Tuneup* is the optimize program, which improves the file-access time of the hard disk by unfragmenting (rewriting all of the data to contiguous sectors) files. Performance is also increased by moving all of the hard disk's free space to one contiguous section. The hard disk can be optimized either for read access or read/write access. The third part of the *Hard Drive Turbo Kit* is *M-Cache*: a disk-cache utility. A disk cache is another way to improve the performance of a hard disk by holding recently accessed disk sectors in memory, much like a RAM disk.

NeoDesk is an alternative desktop for the ST that both replaces and extends its functions. Although it looks similar to the normal ST Desktop, there are a number of significant and subtle differences that vastly improve the use of the desktop. Some of the features of *NeoDesk* include arranging file icons to fit the size of the window, displaying the total size of a selected group of files, being able to format disks in differ-

ent formats, displaying program files as icons on the desktop in addition to files within a window and not changing the date/time stamp when copying a file.

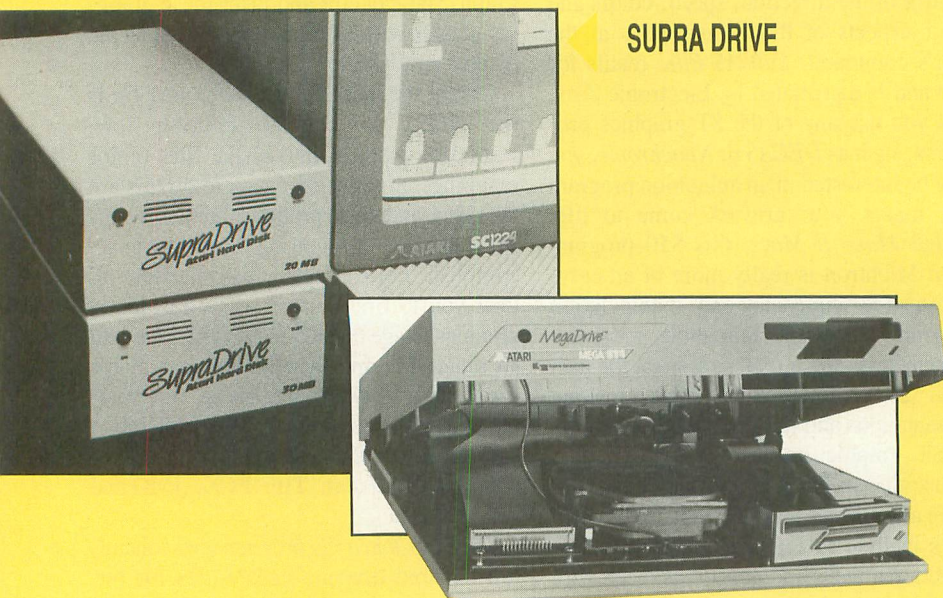
NeoDesk allows you to exit the program and return to the normal Desktop at any time, regaining all of the memory that the program used. In addition, the *NeoDesk* desktop can be saved to disk, complete with program icons, as well as auto-executed when you boot the ST. *NeoDesk* costs \$30 and is an example of what the ST Desktop should be.

Tempus is a GEM-based text editor that was originally published by Eidersoft. As a text editor rather than a true word processor, it includes medium- and high-resolution support, mouse or keyboard operation, auto indentation for structured programming, online help menus and automatic line-number generation. The most outstanding feature of this product is its speed, which is blindingly fast. Compared to a word processor, *Tempus* is four to ten times faster in loading files, scrolling the document from top to bottom and searching and replacing on a single character. Compared to another text editor, such as *MicroEMAX*, *Tempus* is from one-and-a-half to five times faster for the same operations.

Now MichTron has bought the rights to the program and has released *Tempus II*. The new version increases the speed of the original and adds new commands and keyboard shortcuts. Features of *Tempus II* include the ability to edit up to four files simultaneously, optional word wrap with justification, search-and-replace options with wild cards, an assortment of block commands, 20 user-definable function-key macros, a sort utility, automatic file save, and cross-reference generation. *Tempus II* sells for \$80.

Still one of the best ST accessory products is *DeskCart!* from Quantum Microsystems Inc. (QMI). *DeskCart!* contains a battery-backed-up real-time clock and a plethora of desktop accessories on one cartridge. Only 75K of precious RAM is consumed by the accessory programs.

The 14 accessories include a calendar good to the year 2040; an appointment book with alarm; a notebook, which is really a mini-word processor that allows you to create, edit and save multiple 12-page notebook files; a card file that allows nine lines of data; a good calculator; a typewriter that allows you to use the ST to address envelopes or print other short, quick pieces; an ad-



◀ SUPRA DRIVE

TYPHOON THOMPSON



dress book; a VT-52 terminal emulator; keyboard macros; a RAM disk of any size on any drive; disk utility functions; a print spooler; a control panel similar in function to the ST's control panel; and a screen dump and memory test. *DeskCart!* retails for \$100.

GAMES

Ah, games. The ST is unquestionably a premier game-playing computer, and the number of excellent games now available attest to this fact. I might as well first describe one of my favorite games of all time: *Tetris*.

Tetris is a puzzle game with a simple concept. You rotate and position into a solid row at the bottom of the screen various-shaped blocks that fall from the top. Points are awarded as each solid row is formed then disappears. As gaps are left within a row, the rows build up line by line until they reach the top of the screen. When there is no more room for blocks to fall, the game is over. You can either let the pieces fall at their own speed or drop them yourself for more points. *Tetris* is a Soviet-designed game and offers nine levels of addictive play. At each higher level, the pieces fall at an increasingly faster rate and there is a different background screen. Also, you can begin the game with varying numbers of pieces already on the screen for an even greater challenge. Other features include turning the sound effects on or off, displaying statistics on your progress and previewing the next piece that will fall. *Tetris* sells for \$40. If you are going to buy one ST game this year, *Tetris* should be it.

Another excellent ST game is *Arkanoid* from Taito Software. Similar in many ways to *Breakout*, *Arkanoid* consists of several rows of "bricks" of different designs, a ball and a sliding paddle. However, unlike *Breakout*, when certain bricks are hit, they release capsules with special powers. The capsules must be captured with your pad-

dle to gain their special properties. Capsules release additional balls, allow your paddle to capture the ball, award extra lives, double the size of your paddle, temporarily slow the ball down, advance you to the next screen and give you a laser weapon to shoot the bricks. *Arkanoid* has 33 screens, each more difficult than the last, and sells for \$40.

Another of my favorite games is *Typhoon Thompson* from Broderbund Software. Written by Dan Gorlin of *Choplifter* fame, *Typhoon Thompson* offers excellent animation, 3-D graphics and challenging game play. In the game, you control a Jet-Sled (a water craft) around the uncharted ocean planet Aguar. Your mission is to rescue a missing sea-child from the Sea Sprites, which live in these tropical waters. To do so, you must overcome the many defenses of the Sea Sprites. Further, you must capture all of the Sea Sprites from all seven islands within a village in order to retrieve one of the four artifacts from the Spirit Guardians. Each succeeding mission is more difficult until you are destroyed or save the sea-child. Your final score is based upon the number of tasks you complete, the number of islands won, the number of Sprites captured and the amount of time it took to complete them.

What makes *Typhoon Thompson* so good is the quality of the graphics and the attention to detail. The Jet-Sled has a very fluid motion to it, and control of it feels natural. You can see the expressions of the captured Sprites, and the game has an overall dreamy look. *Typhoon Thompson* sells for \$35.

Tower Toppler from Epyx is an interesting variation of the *Q-bert*-style "hopping game." The goal of the game is basically to avoid the monsters and get to the top. But like many games with a simple concept, completing the game is more difficult than it looks. In *Tower Toppler*, eight giant tow-

ers rise from the depths of a toxic ocean. The goal is to destroy each tower after you have reached the top. Each tower has obstacles, such as deadly rolling boulders, disintegrating ledges, indestructible mutant molecules, flying phantoms and flashing blockades. You can hop over some of the obstacles or ride emergency elevator lifts, and a snowball gun can be used to freeze or destroy the enemies. As you climb the towers, they rotate with you so your vantage point changes.

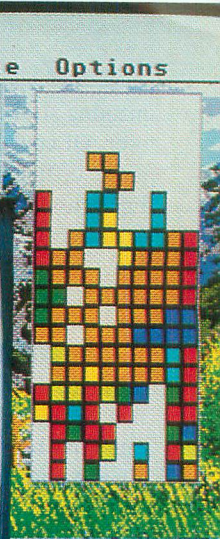
If you fail to reach the top of a tower in the given time limit, the game ends. Extra points can be earned by catching fish between the completion of each of the towers. *Tower Toppler* sells for \$40 and allows one or two players to play. In addition, there are two missions to choose from.

Lords of Conquest by Electronic Arts sells for \$20 and can be compared to the board game *Risk*. It's a classic strategy game in which the action takes place on a world map. Up to four players can play at once and each opponent chooses home territory, then tries to protect his holdings while conquering territories belonging to other players. Games can take anywhere from a half hour to several hours to play.

Another good game from Electronic Arts is *Zany Golf*. More like an arcade game than a golf simulation, *Zany Golf* is like a miniature golf course from hell. Like traditional miniature golf, there are nine holes, each presenting a particular obstacle to overcome. However, the obstacles themselves make the game so challenging (and interesting): Flying carpets, bouncing hamburgers, pinball machines and "rubber birdie bogey bumpers" are not found on your typical golf course.

Each hole has a par value and allows a certain number of strokes. Once you make the putt, any remaining strokes are carried over to the next hole. When all of your re-

TETRIS



maining strokes are depleted, the game is over. Of course, each hole is increasingly more difficult and the final hole, "Dr. Frankenstein's Private Country Club," is exceedingly challenging. *Zany Golf* sells for \$40 and offers top-notch graphics and music.

ODDS AND ENDS

Practical Solutions offers several products that, as their company name implies, solve minor problems encountered with using an ST and peripherals. If you happen to have both a monochrome and a color monitor for your ST, attaching and re-attaching the monitors can be a pain. Monitor Master allows you to simultaneously connect monochrome, RGB and composite monitors to your ST (composite video available only on STs with internal composite video). Then, by simply pressing a button you can switch from one to the other. In addition, a separate RCA audio-out jack is provided on the switch box. Monitor Master is a passive device and sells for \$55.

Another useful product from Practical Solutions is Mouse Master. If you have a 1040ST, you know that it is difficult to attach a mouse or joystick since the ports are located underneath the computer. Mouse Master plugs into the joystick ports beneath the computer and allows you to easily hook up the mouse or joysticks. In addition, both a mouse and joystick can be permanently attached and switched from one to another. Mouse Master requires no separate power supply and sells for \$40.

Neither the 1040ST nor the Mega computer contains an internal RF modulator. Therefore, these machines do not provide composite video output. If you want to record your graphics designs on a VCR or play games on a big-screen TV, you're out of luck. Unless you have VideoKey.

VideoKey is a small box from Practical Solutions that converts the RGB signals from the ST into color-composite video that is superior to even that available on the STs that do have an RF output. VideoKey is designed primarily for low-resolution color, used by the majority of games and animation. A separate external power supply operates the unit, but since it is invisible to your system, you need not unplug your existing monochrome or color monitor. VideoKey sells for \$100. Another product from Practical Solutions is the Tweety Board. The ST computer actually has three separate audio channels, which are accessed by the Tweety Board. It provides true stereo output from the audio generated by the ST. Instead of the tinny sounding audio you

normally hear through your monitor speaker, the full audio capability of the ST can be heard through your stereo or sound system. The Tweety Board is compatible with all ST software and installs inside the ST without any soldering. It sells for \$60.

If you have been searching for the ultimate joystick, your quest might be over. I know of two joysticks that offer state-of-the-art performance at a reasonable price. Wi-co's Ergostick is one. It is a short "throw" stick with a steel shaft that has a slightly sticky soft covering that is easy to grip. Quality microswitches are used for durability and to provide better feedback to the user. The Ergostick sells for \$25.

Another excellent stick is the Epyx 500XJ. This too is a small stick, but the base conforms to the shape of your hand and is less fatiguing. There is a groove for the thumb on the base, and the trigger button is strategically located under the tip of the index finger. This stick is made to be held with the left hand while the right hand's fingers perform the action. The Epyx 500XJ has a sturdy design and feels solid to the grip. It sells for \$25.

FOR THE ST USER WHO HAS EVERYTHING

Several years ago, David Small designed the Magic Sac cartridge that gave Macintosh compatibility to the Atari ST computer. In his latest effort, and from his own new company, Gadgets By Small, he has designed a new, improved product called the Spectre 128. It is a Macintosh Plus emulator cartridge and software that has a faster screen redraw and file-access time compared to the Magic Sac.

The Spectre 128 can use either Mac 128K or 64K ROMs (that you must buy elsewhere). Although it does not read Mac disks directly (a forthcoming product called GCR will do that), the \$180 product runs dozens of Mac software products (such as *HyperCard*, *Adobe Illustrator* and *PageMaker*) at up to 20 percent faster, and it can use the larger ST monochrome monitor.

Digital Vision was the first to develop and market a video digitizer for the 8-bit Ataris. Their video digitizer for the ST is called Computereyes. It includes both hardware and software, and it supports all of the graphic capabilities of the ST. The hardware interfaces easily to any source of standard NTSC color or black-and-white composite video. These sources include video cameras, VCRs, video laser discs and other computers. All capture and display functions are mouse-controlled under GEM,

and images may be captured in either monochrome or full color.

Automatic calibration of brightness, contrast and color balance is performed by the Computereyes software. Once the image has been captured, the brightness, color content and contrast can be easily adjusted to suit your needs. Further, the images saved by Computereyes are compatible with *Neochrome*, *DEGAS*, *DEGAS Elite* and other graphics programs. Computereyes ST retails for \$250.

One of the major upgrades for the ST is a hard disk. There are a number of vendors, drive sizes and options to choose from. All hard-disk drives mentioned here are reliable, differing only in their price, physical dimensions and length of connecting cable.

Atari offers two hard drives that differ only in price and the size of the case. The SH204 is a shoe box-size model that has a 19-inch cable and 20 megabytes of storage. It sells for \$700. The Atari MegaFile 20 sells for \$900 and can serve as a monitor stand or fit under a Mega ST. Both drives come with boot, format and autoboot programs. In addition, a disk-cache program is also provided.

Supra drives range in size from 20 megabytes to 60 megabytes. They are about the size of a shoe box and feature a 19-inch cable that connects to the DMA port on the ST. In addition, all Supra drives come with a boot program, a utility for mapping bad sectors and a formatting program. The drives are available from either the factory or retailers. The Supra 20-, 30-, 45- and 60-megabyte drives sell for \$700, \$795, \$995 and \$1,295, respectively.

ICD also makes hard disks for the ST. Designed to fit under your monitor, the ICD FA20ST provides 20 megabytes of storage and features a 31-inch cable. Innovative features, like a built-in clock with battery back-up, adjustable legs for positioning the monitor and providing cable clearance, and the ability to daisy-chain additional DMA devices are also included. Software consists of boot, format, autoboot and utility programs. The FA20ST retails for \$700. Other ICD hard drives are available: 30-megabyte, \$950; 50-megabyte, \$1,100.

THAT'S ALL, FOLKS!

That concludes this year's gift guide. I hope you'll manage to find that perfect gift for your favorite ST owner. Of course, this article makes a great "want" list too. Just circle items you'd like to find under the Christmas tree and leave the magazine open in a strategic place. You might just get lucky. **A**

Companies Mentioned:

- | | | |
|--|--|---|
| Alpha Systems
4435 Maplepark Rd.
Stow, OH 44224
(216) 374-7469 | ICD, Inc.
1220 Rock Street
Rockford, IL 61101-1437
(815) 968-2228 | Softlogik Corp.
11131F South Towne Square
St. Louis, MO 63123
(314) 894-8608 |
| Atari Corp.
1196 Borregas Ave.
Sunnyvale, CA 94088
(408) 745-2000 | Intersect Software
2828 Clark Road, Suite 10
Sarasota, FL 34231
(813) 923-8774 | Spectrum Holobyte
2061 Challenger Drive
Alameda, CA 94501
(415) 522-3584 |
| Broderbund Software
17 Paul Drive
San Rafael, CA 94903-2101
(800) 527-6263 | ISD Marketing, Inc.
2651 John Street, Unit #3
Markham, Ontario
Canada L3R 2W5
(416) 479-1880 | Springboard Software
7808 Creedridge Circle
Minneapolis, MN 55435
(612) 944-3915 |
| CodeHead Software
P.O. Box 4336
North Hollywood, CA 91607
(213) 386-5735 | Microprose
120 Lakefront Drive
Hunt Valley, MD 21030
(301) 667-1151 | subLOGIC
713 Edgebrook Drive
Champaign, IL 61820
(217) 359-8482 |
| Curtis Manufacturing Company, Inc.
30 Fitzgerald Drive
Jaffrey, NH 03452
(603) 532-4123 | MichTron
576 S. Telegraph
Pontiac, MI 48053
(313) 334-5700 | Supra Corporation
1133 Commercial Way
Albany, OR 97321
(503) 967-9075 |
| CompuServe Communications
5000 Arlington Center Blvd.
Columbus, OH 43220
(614) 457-0802 | Migraph, Inc.
200 S. 333 St., #220
Federal Way, WA 98003
(206) 838-4677 | Taito Software, Inc.
11715 North Creek Parkway South, Suite 110
Bothell, WA 98011
(604) 984-3040 |
| DELPHI
General Videotex Corp.
3 Blackstone Street
Cambridge, MA 02139
(800) 544-4005
(617) 491-3393 (in MA) | Mindscape
3444 Dundee Rd.
Northbrook, IL 60062
(312) 480-7667 | Timeworks
444 Lake Cook Rd.
Deerfield, IL 60015
(312) 948-9200 |
| Digital Vision
14 Oak St., Suite 2
Needham, MA 02192 | Neocept
547 Constitution Ave., Unit A
Camarillo, CA 93010
(805) 482-4446 | Unison World
2150 Shattuck Ave., Suite 902
Berkeley, CA 94704
(415) 848-6666 |
| Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
(800) 245-4525 | Panamax
150 Mitchell Blvd.
San Rafael, CA 94903
(800) 472-5555 | Wico Corporation
6400 West Gross Point Rd.
Niles, IL 60648
(312) 647-7500 |
| Epyx, Inc.
600 Galveston Drive
Redwood City, CA 94063
(415) 366-0606 | Practical Solutions, Inc.
1930 E. Grant Rd.
Tuscon, AZ 85719
(602) 884-9612 | WordPerfect Corp.
288 West Center St.
Orem, UT 84057
(801) 225-5000 |
| Gadgets By Small
40 W. Littleton Blvd. #210-211
Littleton, CO 80120
(303) 791-6098 | Quantum Microsystems, Inc.
P.O. Box 179
Liverpool, NY 13088
(315) 451-7747 | Xlent Software
P.O. Box 5228
Springfield, VA 22150
(703) 644-8881 |
| Gribnif Software
P.O. Box 350
Hadley, MA 01035
(413) 584-7887 | Reeve Software
29W 150 Old Farm Lane
Warrenville, IL 60555
(312) 393-2317 | |

T H E A N I M A T I O N S T A N D

SPECIAL EFFECTS

FIGURE 1

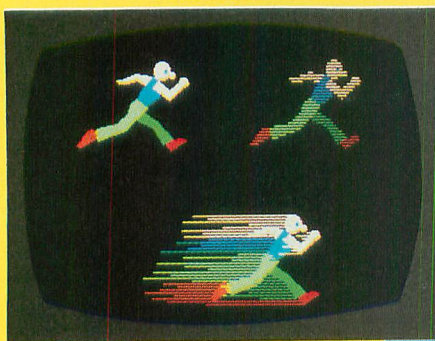


FIGURE 2

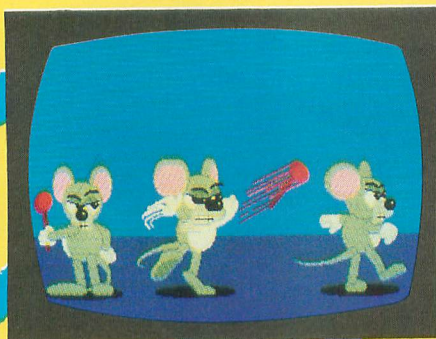
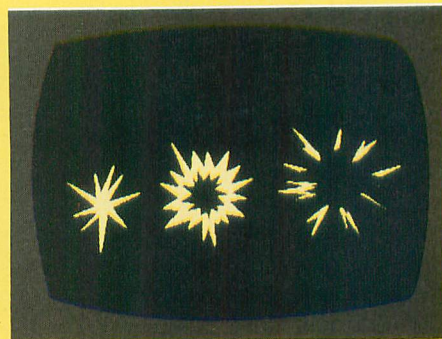


FIGURE 3



MAURICE MOLYNEAUX

Byond the important aspects of such animation basics as squash-and-stretch, anticipation, characterization through motion and so on (discussed last time), there is another realm of techniques that fall under the general heading of "special animation effects." These range from such simple things as speed lines to the bizarre world of distortion, double and triple images and so forth. If you want to animate something moving quickly, exploding, being disintegrated, flattened, you'll want to take advantage of these techniques.

LESS IS MORE

Before we proceed, I feel it's important to once again stress that the following techniques should not be overused. They should be used only where necessary. The reasons for this sparing usage are twofold: First, they are meant to add energy and/or life to a scene or action, and their overuse tends to make them less special, and, consequently, they have less impact. Second, since most of these techniques involve some kind of distortion of the image to attain a certain effect, continual use gives the viewer a chance to see the effect repeatedly, allowing them to see through the effect to what you're actually doing. Therefore, they start

seeing the effect as an effect, rather than as part of the action.

I've said it before, and I'll say it one more time: Nothing exceeds like excess.

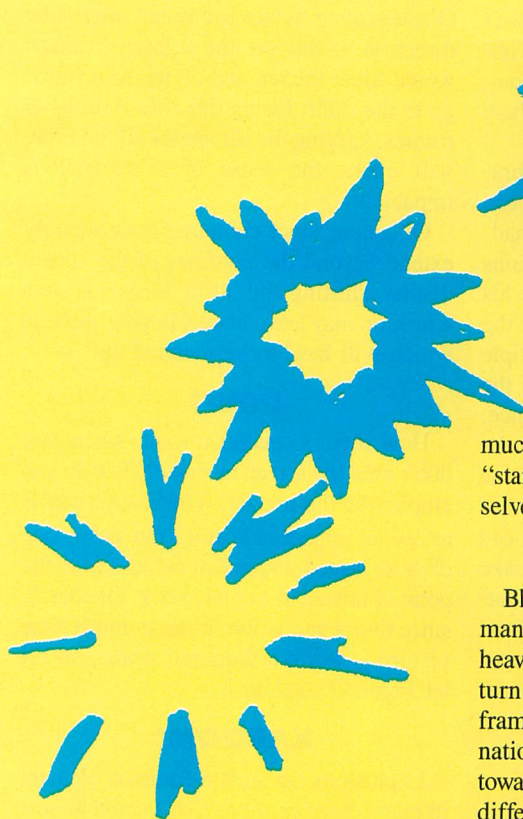
SPEED!

Try making some relatively large object move across the screen in a span of ten frames or less. Go ahead, try it. No matter what animation program you're using, chances are you're going to notice a sort of "jumping" effect. The object doesn't seem to actually move smoothly across the screen. It gets there in a series of small "hops". No matter how fast the playback is, you can see each individual image. And, because these images are so few and the changes in each frame so great, it seems there's almost nothing you can do about it. Adding more frames would smooth out the action, but it would also slow it down. Even at full-speed playback, the object wouldn't seem to be moving all that fast. So, how can you make an object move quickly from one position to another without encountering this staccato movement problem or slowdown? There are a number of techniques that can be used to cover this effect, but the three basic types are speed lines, blurs and distortions.

SPEED LINES

Everyone knows these. They appear everywhere from animated cartoons to comic books. When you want to imply speed, you simply do not fully draw the trailing edge of the fast-moving object/character, but instead draw a lot of parallel lines in its wake. In animated film this effect is usually accomplished via a technique known as "dry brushing," wherein a brush with very little paint (nearly dry) is used to streak the colors of an object, creating speed lines. How to do this with your ST varies with the software you're using. Since you have to deal with pixels and limited resolution, the easiest speed lines to draw are those that are perfectly horizontal or vertical. In such cases you might manually draw the lines on a specific frame or frames, or, if the program you're using allows it, cut out a copy of the object, then paste it down elsewhere and edit it. You would then knock out every other line with the transparent background color, cut out this new brush and then paint a trail behind the object. To make this effect less mechanical, you would make the lines of uneven length and perhaps even let some of the blur spill over parts of the object. Figure 1 demonstrates these steps.

Please note that to use this technique, you



may have to save a "safety" copy of the original frame, so that if the speed lines obliterate important details, you can retrieve them from this backup screen. Also, if an object has many details and colors, you may have to touch up the speed lines. The rule of thumb is that each speed line should be the same color as the pixel from which it originates on the object. If a speed line comes off a brown belt, it should be brown. The exception to this is speed lines over a color where the actual color wouldn't work, such as a brown sleeve moving in front of the jacket it's attached to. In such a case, you will probably have to use a different color or the lines won't show up.

Speed lines have some limitations. They don't work well if the background features a lot of cluttered colors or lines that parallel the speed lines. In such instances the speed lines can get lost. Furthermore, while speed lines can be successfully maintained for a considerable period of time in a traditional animated cartoon, they don't tend to work well on computers if used for more than a handful of frames. This is because the drybrush technique used on cels provides a certain amount of randomness to each frame, so that each line doesn't stand out for long. On a computer, the screen resolution tends to make the lines heavier, and such randomness is much harder to achieve. Therefore, the lines themselves are

much more obvious and have a tendency to "stand still," drawing attention to themselves.

BLURS

Blurs are often accomplished in the same manner as speed lines, but they are usually heavier. For instance, having a character turn his head suddenly might involve one frame where the head is almost to its destination pose, but all the colors "leak" back toward the previous position. The key difference between a blur and a speed line is that a blur isn't a bunch of parallel lines, but a mass of color, occasionally featuring a few speed lines atop it for added effect.

Blurs don't always convey as much velocity as speed lines. Speed lines lend a look of great speed to a movement because they are somewhat transparent, making it seem as if the object passed by so fast that you could see through it. Because blurs tend to be more opaque, they seem to have more weight and thus less speed. They are better for shorter actions, like the aforementioned head movement, rather than something like *Road Runner* bolting down the highway at Mach 2 (which is better accomplished with speed lines).

DISTORTIONS

To the average viewer of animated film, this is probably the least-noticed speed effect. It is the "stretch" part of squash-and-stretch taken to ridiculous extremes. Rather than some mild warping of form to lend a little more *oomph* to an action, a distortion is a wild deformation of an object for a frame or two (no more). It is commonly used to accomplish nearly instantaneous transitions from point/position A to B.

For instance, in Bob Clampett's *The Great Piggy Bank Robbery* (1946), Detective Duck Twacy (Daffy Duck) is on the phone, and in the span of three frames he goes from leaning on one side of a desk to laying atop it, on his back, on the other side.

**I've said it
before, and I'll say
it one more time:
Nothing exceeds
like excess.**

There is a frame where he starts moving, one where he stops, and the single frame in between them features a drawing of Daffy that is hardly recognizable as the duck. The animator has created a pose that features one half of Daffy in the starting position and the other half in the destination position, with the duck's details stretched in between, as if he were made of Silly Putty! The drawing is so severely distorted that when you look at it your mind refuses to believe such a wild drawing would work at all. You think, "Surely, that bizarre image would stick out like a sore thumb." But, no, when you view the cartoon, you don't see it. You see Daffy zip into another pose, but you don't actually see the distorted drawing. Chances are, even if you've watched it via freeze-frame, you'll have a tough time seeing it during normal playback.

A few months back, I loaded one of the scenes from the *Art & Film Director* video into my ST and decided to try a little experiment on it. At the end of the shot, Megabit Mouse spun around and hurled a mallet at and through the *Art Director* toolbox. I had never been satisfied with this action, because I was unable to make him throw it as fast as I wanted and have it look right. To get a pleasing action, I had to use more frames than I wanted, and the throw lacked the force I sought. As an experiment, I deleted a few frames of Megabit spinning around with the hammer and replaced them with a single frame featuring a wildly distorted mouse. Figure 2 shows the start and end positions, with the distorted image in the middle. Like the Daffy Duck scene described above, when I played it back I couldn't really see the distorted image, but wow! did that throw have force! That single frame of distortion accomplished what I thought was impossible.

Such wild distortion can be used to strengthen a lot of other types of action. Just freeze-frame through almost any door-slam in an old theatrical cartoon. The door "bends," its top and bottom trailing. In the

middle of its action, it will often appear six times as thick as normal. The doorknob may even lag behind, stretched out, and actually sink into the door when it catches up. Cars bunch up and change shape as they "wind up" to zip away, buildings bulge and radically change shape and volume, and characters' limbs become massive arcs for a single frame.

And it's all virtually invisible to the viewer.

MULTIPLE AND AFTERIMAGES

Another technique for enhancing movements is the use of multiple images. These can sometimes be used in lieu of speed lines and blurs and are often useful for effects involving movements too rapid to animate with single images on each frame.

Afterimages have a distinct look. Specifically, they tend to be either transparent or, at the very least, dimmer than the object in its current position. If not, it becomes difficult to discern which image is the actual object. The easiest way to create an afterimage for a frame is to take the previous two or so frames, combine them into a new image, dim the colors (if your palette permits) and add them to the current frame. The best program for such effects has to be *Cyber Paint*, where you can overlay and underlay pictures and whole animation sequences. If you do this to each frame, the selected moving objects will have "trails" that follow them. If done properly, they can add the illusion of speed, but more often they are used for different effects.

(Using TRACE ON in *Film Director* automatically leaves afterimages, but it leaves all afterimages for the range of frames selected, and no color-dimming is possible.)

While afterimages are composed of multiple items, what I refer to as multiple images is a different animal. This means creating a frame that has two or more images of the same item or character. For instance, a frame that has a character looking both left and right at the same time (in essence, having two heads). Multiple images are mostly used for extremely rapid motions where more poses must be struck than there are available frames. For instance, you want a character to look up, down, left and right numerous times in a matter of only a few frames. You do this by having one frame where the character is looking one way, another frame where he is looking in three directions at once (perhaps enhanced with speed lines), another with him looking in

two directions and so on. The eye doesn't necessarily perceive these as the same image. You see him looking wildly about, striking more poses per second than there are frames.

This same technique is used for vibrations. Let's say your character has just been hit on the head, which vibrates like mad. Drawing his head in a series of positions going left to right would look just like his head going from left to right. To get the chaotic feel of a real vibration, multiple images are used. Like the example of the guy looking in multiple directions at once, here we draw the character's head in the same way on each frame, but on some frames we have two, others three, another one and so on (you may add some speed lines to heighten the effect). If you make sure not to put any specific or repeating pattern into it, the effect will be one of his head rapidly vibrating. If you make the images fall into a repeating cycle, such as one head, three, two, one, two, and then repeating the cycle again, the viewer will pick up on this, and the illusion will be destroyed.

VIOLENT SOLUTIONS

In classic cartoons, the most common "effects" were violent actions, such as explosions, crashes and other catastrophes. Most of us are consciously aware of such obvious effects as fiery explosions, clouds of smoke and cartoon characters imprinted in steel doors. Those are pretty obvious. But there's a whole range of techniques used to make these acts of violence have more impact.

For instance, let's say you want to drop a safe on a bulldog. How do you do it? You animate the safe falling into the picture, covering up the bulldog as it lands on him and embeds itself into the sidewalk. If you animated such an action exactly as I've just described it, the action would lack any sense of weight and impact. How would you make the safe "feel" heavy?

The easiest way is to jar the whole scene, as if the impact of the safe had shaken the camera. This can range from a slight quivering of the entire frame to severely shifting the image. Sometimes enlarging and shrinking the image on a few frames will achieve the same effect. If this is done over a series of frames, and it doesn't fall into a pattern, the shot will have a lot more impact than it would without this special effect.

To make the impact even more severe, you could try tilting the entire frame; that is, rotating the image so it is at an angle.

On alternating frames you would reverse the direction so that on one frame the image would tilt to the left and on the next frame go to the right. Doing this over a series of frames, varying the tilt angles all the time, will create the sense of a tremendous impact.

Of course, since computer images rarely extend beyond the confines of the screen display, shifting and tilting images in such a fashion may leave blanks in your picture, which will have to be touched up.

POW!

How about a smaller impact, where you don't want the screen to shake? How do you emphasize the impact? The easiest way is to use an impact indicator. You know, one of those jagged shapes that radiates from the point of impact in a comic book. Of course, since this is animation, it has to move. One picture is worth a thousand words, so I'll let Figure 3 explain it.

KABOOM!

Explosions are such a common element in many older cartoons that I could almost write an article on that subject alone. For the sake of brevity, I'll keep it to a small section.

Explosions are usually bright, quick affairs. Like the violent impacts discussed above, they will often involve camera-shaking techniques. The easiest explosions to animate are of the ultra-cartoony variety, with lots of jagged, starlike polygons of bright color filling the screen. These aren't often effective, so most explosions are of the fire-and-smoke variety, featuring either a flash followed by swirling smoke or just fiery-colored smoke.

Fiery explosions are difficult to animate properly, particularly when you have a palette as limited as the ST's. Usually, such explosions feature a lot of swirling clouds of reds and blacks, which need to move like real smoke in order to look right. They also have to be dissolved out, which isn't easy to do with only 16 colors. Unless you're feeling really ambitious, I'd recommend not using such explosions.

The most violent-looking of all explosions is the flash variety. First, a point of bright light briefly emanates from the exploding object, then the entire screen goes white. The problem with this effect is that the white alone seems violent for a split-second. If you hold it for longer than that, it becomes a white screen instead of an explosion.

I learned the trick to this years ago when

I animated the *Balance of Terror* *Star Trek* game proposal. I wanted the starships to explode in a way not usually seen in a computer game. No little sparks flying or animated fireballs. I wanted it to look like their power sources had been violated and the vessels were instantly vaporized. I selected the white-flash technique for this and solved the inherent problem by changing the screen colors to black on every third or fourth frame. The effect was instantly transformed from a static field of white to a flashing, strobing spectacle. People tended to comment that the explosion was almost too violent. But it works. It's one of the simplest effects, and at the same time, one of the most effective.

LIGHTNING AND ELECTRICITY

Let's face it, the stereotyped zigzag lightning bolt won't cut it anywhere but in the most cartoony of circumstances. Lightning doesn't look anything like the cartoon symbol of it. It's much more complicated and, therefore, a lot tougher to animate.

I recently suffered through the movie *Star Trek V*, and of the myriad bad special effects that film contained, one of the worst

violators were the lightning bolts fired by the "un-God" at the film's climax. These things looked like beams with little electrical frayings around the edges. No go. They didn't work. The way to do electricity is to think of a Jacob's Ladder. You know, one of those gizmos in the labs of mad scientists where a little electrical arc runs up between two metal rods. The electrical arcs seen in those writhe and twist as they go. You never see frayings. The electricity follows the path of least resistance, which is rarely a straight line.

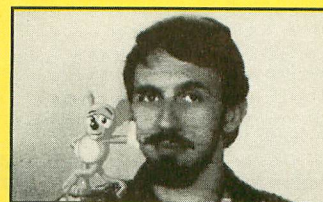
The best-looking lightning-bolt effects are simply a bunch of thin, scrawly lines connecting two points. Make those lines change frame after frame, and you'll get a much more interesting effect. If you want to freeze-frame through some well-executed lightning effects as reference, study either the opening scenes of *Star Trek—The Motion Picture* (where the Klingon ships get zapped) or *Return of the Jedi* (where the emperor lets Luke have it).

By the way, such electrical effects can benefit greatly from occasional one-frame flashes where you brighten up the palette. Heck, even the white/black alternating effect discussed for explosions comes in

handy. Make an occasional one-frame negative image, and it works quite well.

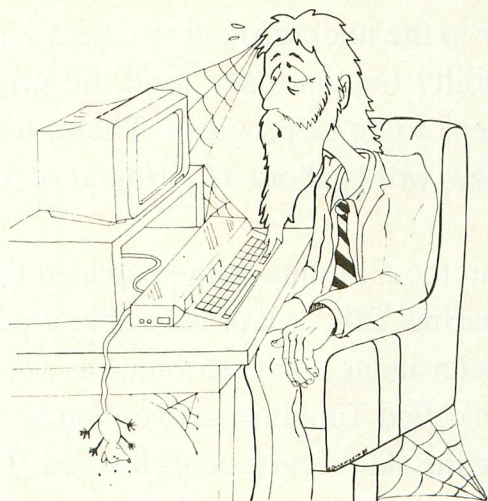
THE SECOND-TO-THE-LAST ROUNDUP

I was planning to wrap up this series with this article, but it seems I've once again run out of space. Next time it's on to post-production, where we'll discuss putting it all together, from adding sound to putting it on videotape. See you then. **A**



Blissfully ignorant of the realities of time and space and plain old common sense, Maurice Molyneaux hopes someone will someday discover retroactive reincarnation so that when he dies he can come back in a previous life as animation director Chuck Jones. His greatest fear would be to come back as Wile E. Coyote and, in the process, have to learn some humility.

How much is YOUR time worth?

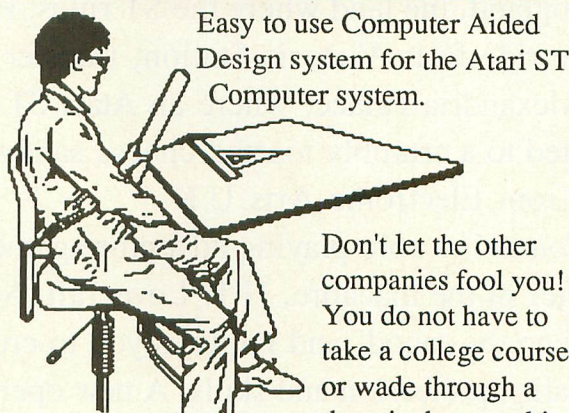


Turbo ST turns your 520/1040 ST into a speed demon. Check out **Turbo ST** at your local dealer or download our free demo from your favorite bulletin board.

Upgrades to version 1.6 are only \$5 with your original disk. For more information, call (407) 657-4611, or write to SofTrek, P.O. Box 5257, Winter Park, FL 32793.

CIRCLE #113 ON READER SERVICE CARD.

ATHENA II



Easy to use Computer Aided Design system for the Atari ST Computer system.

Don't let the other companies fool you! You do not have to take a college course or wade through a three inch manual in order to use a CAD package! Within minutes you can be up and running with Athena II.

See what Athena II can do for you at your local Atari Dealer!!

Only **\$99.95**

iliad
Software Inc.

1470 South State, Orem, Utah 84058 (801) 226-3270

CIRCLE #114 ON READER SERVICE CARD.



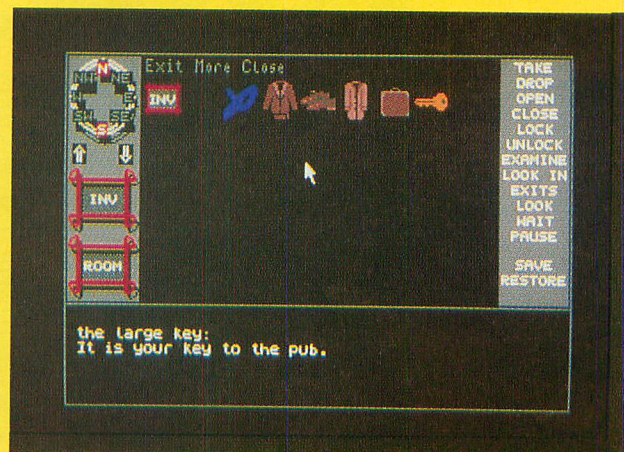
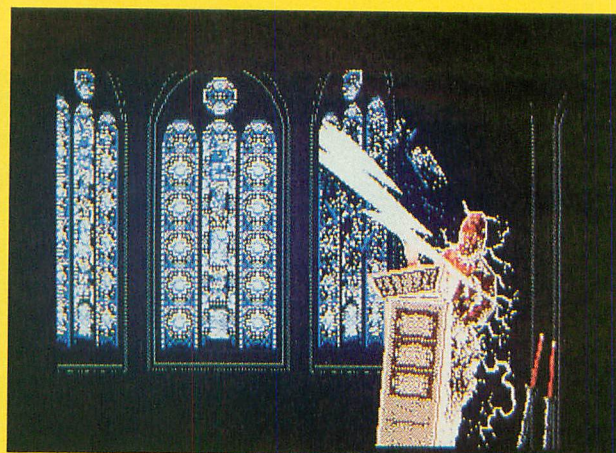
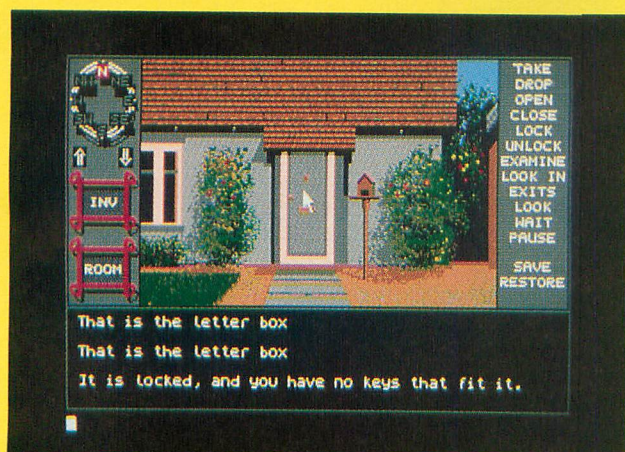
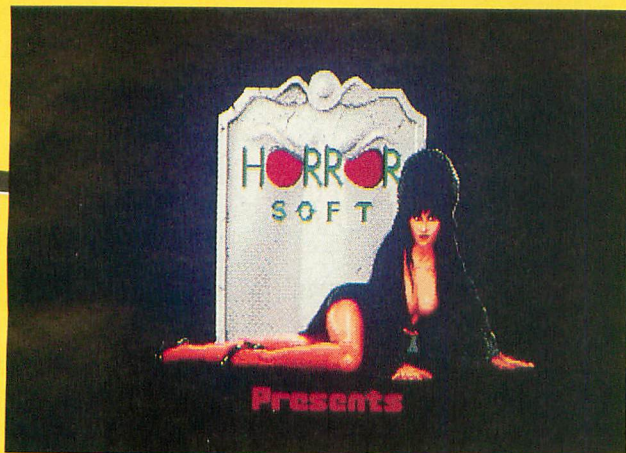
By **Marshall M. Rosenthal**

FROM OVER THE BIG WATER

England, the land where the ST rules. Popping over to the underground, we take a quick Tube ride from Victoria Station, transfer to the Picadilly line and take it all the way out to Alexandria Palace, where an Atari ST show happens to be in progress. There, we are treated to a crumbly toasted-cheese sandwich and a few words about *The Hound of Shadow* from Electronic Arts U.K.

Hound is a role-playing adventure game that adds to the Cthulhu mythos created by the master of the macabre, H.P. Lovecraft. Set in the Roaring '20s, it will use sepia graphics to keep the mood, and requires you to create a character using such attributes as sex, nationality, profession and skills. A new operating system called Timeline enables you to move your character to subsequent adventures while retaining the skills and lessons learned. Timeline also affects the game in response to the character and attributes you've chosen. England may seem genteel and tame, but, in *Hound*, evil lurks just below the surface (although, I doubt anything can compare with the horror of the cheese sandwich I just consumed).

Jeff Minter is nearby. He's one of the first game programmers in England, having created games for the (gasp!) VIC 20, then moving over to the Atari 8-bit and now the ST. His *Super Grid Runner* maintains the excellence of fun graphics, nifty sound effects and



PERSONAL NIGHTMARE

Graphics are bright and cartoon-like, with the program utilizing the entire 512-color palette. Sound, as in all Minter games, is loud, effective and entertaining. (Why can't other developers handle the sound chip like Jeff does?) Full documentation includes notes and personal comments by Minter, a nice job, except for the Tom Thumb-sized print.

Supernova releases two budget arcade games, *Classic Invaders* and *Pharaoh III*. New art-work and characters, but they're still just *Space Invaders* and *Galaxian*. *Invaders* is exactly what it should be: those ugly faces zipping back and forth, as your blaster dodges their fire from beneath three bunkers. Then it's your turn to blast them to bits. *Pharaoh III* adds some digitized music and an opening story sequence before placing you outside of the Pyramids of Gizeh. Then it's rock-'em, sock-'em time as Egyptian Gods of the Dead try to wipe out descending spacecraft. No surprises here, just solid gameplay.

Personal Nightmare makes its appearance, and it turns out that all the hype noted last time was mostly justified. *Nightmare* is a graphic/text adventure game with some

playability. *Grid*'s premise is simple: You move your spaceship around a geometric grid pattern, avoiding the falling bombs and aliens. Not too complex, eh? Of course, there are the kamikaze spaceships, the replicating snakes which leave pods that turn into bombs, and the invulnerable moving turrets that send blasts along the grid.

Control of the ship is easily handled by mouse movements, coupled with a unique firing mechanism. A special nose-cone can be called over to the ship by pressing the

right mouse button. Once there, a tap of the left mouse button fires blasts into the cone, which cause them to exit as multiple rays of greater strength and in planned directions (accessed through Left Shift/Alternate key). The cone can be called over to another area or left where it is. Help also appears in the form of shield tokens which must be grabbed, and a falling Llama, which adds an extra life (Minter has this thing about Llamas, you see). Blasting these little fellas is a no-no.

SUPER GRID RUNNER



added perks, most notably the animation in the viewing window.

Set in England, you find that your father, the local vicar, has mysteriously vanished. Arriving at your boyhood home of Tynham Cross, you find the sleepy little village almost unchanged, except for this nagging feeling that something is wrong.

The local pub and hotel is the Dog and Duck, run by the nice (?) Mr. and Mrs. Jones. Plenty of other townsfolk are around, including the local constable, all friendly, going about their business—and concealing a dark secret.

Nightmare is touted as good, clean, in-offensive horror containing “adult themes,” meaning there’s some violence and general mayhem not suggested for the young (but probably less damaging than a *Rambo* film). The game progresses well and is easy to access through on-screen icons and mouse. You can click on the compass at the upper-left to move about, and in most cases, clicking on objects and areas accesses them in an intelligent manner (clicking on a door means you want to open it, not put it in your pocket).

Objects picked up are stored in an inven-

tory window that is “turned on” when desired. Sound effects are present throughout and the digitized ones enhance the eerie mood as night descends. The graphics are quite nice as well, even Elvira’s appearance with the logo is worth seeing (at least once; then you can bypass all of this to continue a saved game or start a new one).

Perhaps the only complaint is that five disks require a lot of swapping, since no RAM cache is supplied, nor is multiple drive use supported. This can get irksome. Copy-protection is all right, but why can’t we go to a hard disk, or at least use another drive? These exceptions aside, *Personal Nightmare* scores well, an above-average adventure.

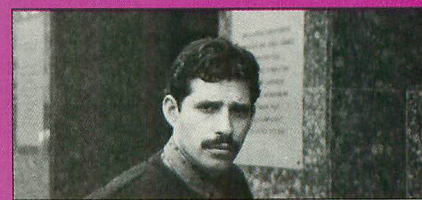
For a bit more insight, we turn to the game’s designer, Mike Woodruff. “There are a number of things that make *Personal Nightmare* different,” he begins. “Game logic is handled by our AGOS interpreter, a powerful interface that makes interacting easy for the user. Moving around and typing in responses is part of these games, a bit of a boring part since we’re so used to it by now, but there’s also some 500 sequences of animation and 600K of digitized

sound. Events are ongoing and occur as time passes on. So you will see different things depending on where you are each time you play.”

Not wanting to give too much away, Woodruff concedes that supernatural forces are at work here, and scenes that are “very nasty, indeed” lie in store for the player.

The next game out will feature Elvira, Mistress of the Night. You play an average human caught up in the strange and weird as you attempt to rescue Elvira from the evils haunting her medieval castle.

Well, it’s time for another sandwich, which means a trip out of the convention hall and on to a McDonald’s (such a decadent American, indeed). See you next time.



Marshal M. Rosenthal is a New York-based writer and photographer whose work takes him throughout the world. Working extensively in the computer/electronic entertainment field, his only known fear is the brown-out.

Product Information:

CLASSIC INVADERS

Pharaoh III
Supernova Software
Black Horse House
Exmouth, England EX8 1JL

THE HOUND OF SHADOW

Electronic Arts UK
Langley Business Centre
11-49 Station Road
Langley Nr. Slough, Berkshire
England SL3 8YN

PERSONAL NIGHTMARE

HorrorSoft
Addison Industrial Estate
Blaydon, Tyne & Wear
England NE21 4TE

SUPER GRID RUNNER

Llamasoft
49 Mount Pleasant
Tadley, Hants
England



AUTHORIZED SERVICE
CENTER FOR ALL
ATARI PRODUCTS

MICROTYPE

A DIVISION OF MICRO PERIPHERALS, INC.

4049-51 MARSHALL RD. • KETTERING, OHIO 45429



HARDWARE ST'S...IN STOCK!!!

Color Monitors	CALL
Mono Monitors	CALL
SF 314 Drive	CALL
GTS 100 Drive	CALL
IB 5 1/4 Drive	199
Navarone Scanners	CALL

PRINTER'S DEVIL
HI-RES GDOS
FONT & CLIP
ART PACKS
IN STOCK!
(Great for
Desktop Publishing!!)

MODEMS

SX-212 300/1200 bps	CALL
Avatex 1200E	79
Zoom 2400	125

**ATARI ST
SCANNERS,
SOUND &
VIDEO
DIGITIZERS
In Stock!**

SUPRA 2400

HAYES® COMPATIBLE

\$124.95

6' ST MODEM CABLE.....9
(With Purchase Of Supra)

HARD DISK DRIVES FOR ST'S

SUPRA 30 MB HARD DISK

\$619.95

OTHERS.....CALL

ICD 20 MB HARD DISK

\$599

OTHERS, INCLUDING TAPE BACK-UP...CALL

LARGEST SELECTION IN THE U.S.

★ OVER 1000 TITLES IN STOCK ★

IF YOU SEE IT CHEAPER IN ANOTHER AD
CALL US ANYWAY...WE'RE PROBABLY ALREADY
LOWER...THESE ADS TAKE 3 MOS. TO GET OUT.

10th Frame Bowling	26	Cracked	21
220 ST (Terminal Emulator)	15	Crazy Cars	25
3D Breakthru	26	Cyber Control	39
AB Zoo	21	Cyber Paint	48
Advanced OCP Art Studio	31	Cyber VCR	45
Air Ball	26	Dark Castle	27
Air Ball Construction Set	17	Data Manager ST	46
Algebra 1, 2, 3	ea 14	Datatrieve	33
Aliants	19	DB Man	149
All About America	41	Death Sword	13
Alt	21	Deep Space	31
Alternate Reality-The City	32	Defender of the Crown	31
Alternate Reality-The Dungeon	32	Degas Elite	38
America Cooks Series	ea 9	Demon's Winter	25
Architectural Design	19	Desk Cart	67
Arctic Fox	26	Digi Drum	14
Art Gallery 1, 2, 3	ea 19	Dive Bomber	19
Assem Pro	37	Dr. Drums (DR T)	19
Autoduel	24	Dr. Keys (DR T)	19
Award Maker	27	Drax	129
Balance of Power	32	Dungeon Master 2	18
Bally Hoo	27	Dyna Cadd	429
Barbarian	25	Easy Draw (Regular)	63
Bards Tale	31	Easy Draw W/ Supercharger	95
Base Two	45	Easy Tools	32
Basketball (Two on Two)	26	Elite	22
Battle Droidz	25	Empire	34
Battlezone	19	Expert Opinion	72
Beyond Zork	34	EZ Score Plus	95
Biology 1, 2, 3 or 4	ea 14	EZ Track Plus	43
Bismarck	28	F15 Strike Eagle	24
Black Lamp	17	Falcon ST (Low, Low)	CALL
Blockbuster	27	Fast Basic	59
Boulderdash Construction Kit	17	Fast Basic M Compiler	19
Brataccas	15	Fire and Forget	25
Breach	27	First Cadd 2.0	33
Bridge 5.0	24	First Letters & Words	25
Bubble Ghost	24	First Shapes	25
Business Tools	26	First Word Plus	59
Cad 3D (Ver. 2.0)	57	Flash (Great!)	18
Calamus	175	Flight Simulator 2	32
Calamus Font Editor	62	Scenery Disks	ea 18
Captain Blood	31	Font Disks (Pub Part) 1-6	ea 20
Carrier Command	29	Fonts and Borders	24
Certificate Maker	25	Fontz ST	22
Championship Baseball	27	Foundations Waste	26
Championship Wrestling	26	Fraction Action	24
Chartpak	34	G + Plus	21
Chessmaster 2000	29	Gateway	31
Chrono Quest	29	Gauntlet	31
Circuit Maker 2	63	Genesis (Molecular Modeler)	59
Clip Art 1, 2, 3, 4, 5, 6	ea 13	GFA Basic 3.0	59
Club Backgammon	23	GFA Basic Book	27
Color Computer Eyes	169	GFA Companion	32
Colorburst 3000	25	GFA Compiler	38
Copyist Level 2	158	GFA Draft Plus	49
Cosmic Relief	26	GFA Quick Reference Manual	12

PRINTERS

PANASONIC	call for latest
1180	CALL
1191	CALL
1124	CALL
Panasonic Brand Rib	CALL

STAR	Call for latest
NX-1000	NEW! CALL
NX-1000 Color	CALL
1000 Ribbon (Bk)	6
1000 Ribbon (Color)	8

OLYMPIA	simply, the best!
NLQ modes use 18 x 24 matrix!	
NP-30	130 CPS...199
NP-80s	240 CPS...changeable
font cards	349
NP-136	15 inch...529

ACCESSORIES

ST Dust Covers	from 8
Mouse Mat	5
Power Strip w/ Surge	15
Deluxe Power Strip w/ Surge	24
Drive Master	32
Monitor Master	32
Mouse Master	29
EPYX 500 XJ Joystick	15
WICO Ergo Stick Joystick	17
Printer Stand-Heavy Duty	13
Mail Labels 3.5x15/16-500 pk	4
1000 pk	6
Compuserve Starter Kit	24
On-Line Encyclopedia Kit	36
Printer Cable 6'	14
Modem Cable 6'	14

MIDI

Midi Cables 5' to 25'	CALL
Software (Hybrid Arts etc.)	CALL

★ ST SOFTWARE ★

N Vision	29	Spy vs Spy 3 (Arctic Antics)	19
Neo Desk 2	33	ST Disk Drives Inside & Out	18
New Tech Coloring Book	15	ST Gem Programmers Ref Man	15
Night On The Town	22	ST Internals Book	15
Ninja	14	ST Intro to Midi Book	15
Obliiterator	25	ST Machine Language Book	15
Ogre	27	ST Pool	21
Oids	24	ST Talk Pro	17
Omnires	23	STAC	44
Orbiter	25	STOS	39
Page Stream	115	Star Fleet 1	37
Paint Pro	33	Star Raiders	19
Paintworks	15	Starblitzer 2	26
Paperboy	24	Stellar Crusade	36
Partner Fonts	21	Strip Poker 2	25
Partner ST	43	Sub Battle Simulator	25
PC Ditto 2	Low	Sundog	25
Perfect Match	27	Super Base Professional	174
Personal Pascal	66	Super Star Ice Hockey	31
Phantassie 1, 2 or 3	ea 26	Swift Calc St	46
Phasar 3.0	58	Take Note	52
Pinball Wizard	24	Tanglewood	25
Pirates of the Barbary Coast	17	Terror Pods	25
Planetarium	33	Test Drive 1 or 2	24
Platoon	25	Test Drive 2 Extra Disks	ea 14
Police Quest 1, 2	32	Three Stooges	34
Pool of Radiance	25	Thunder	26
Prime Time	27	Time Bandit	24
Print Master Plus	26	Top Gun	11
Prison	25	Trailblazer	32
Pro Copy (Latest Ver.)	28	True Basic	52
Publisher ST	79	Tune Smith (DR T)	95
Q Ball	21	Tune Up	31
Quantum Paint Box	31	Turbo ST	32
Quink	11	TV Sports Football	31
Rastan	25	Typhoon Thompson	21
Read & Rhyme	24	Uninvited	31
Renegade (Outcast)	14	Universal Item Selector	12
Road Runner	26	Universal Military Sim.	31
Roadways	22	Vampires Empire	20
Rockford	39	Vegas Craps	24
Santa Paravia	19	Vegas Gambler	23
Scan Art	32	Video Titleing	22
Scraples	29	Vip Professional (Gem)	129
SDI	13	War Ship	38
Shadow	18	Wargame Construction Set	22
Shadowgate	34	Winter Challenge	11
Shard of Spring	27	Wizards Crown	26
Shuffleboard	19	Word Perfect	159
Silent Service	24	Word Up	47
Sinbad	19	Word Writer ST	46
Sky Fox	31	World Games	12
Space Quest 1 or 2	ea 34	World Karate Championship	19
Space Quest 3	37	WWF Microleague Wrestling	29
Spectrum 512	41	Xevious	21
Spelling Bee	19	Zak McKracken	27
Spiderman	7	Zany Golf	26

HOURS: M-F 9 a.m.-9 p.m. EST
SAT 9 a.m.-5 p.m.

ALL 50 STATES CALL TOLL FREE
1-800-255-5835

For Order Status or
Tech. Info, Call (513) 294-6236

TERMS AND CONDITIONS

• NO EXTRA CHARGES FOR CREDIT CARDS! • Minimum order \$15 • C.O.D. Yes, if all Shipping Charges are PRE-PAID • SHIPPING: Hardware, minimum \$4; Software and most accessories, minimum \$3 • Next day shipment available at extra charge • We ship to Alaska, Hawaii, Puerto Rico (UPS Blue Label Only), APO and FPO • Canadian orders, actual shipping plus 5%, minimum \$5 • Ohio residents add 6% sales tax • Please allow 2 weeks for personal or company checks to clear • All defective products require a return authorization number to be accepted for repair or replacement • No free trials or credit • Returns subject to 15% re-stocking charge • Due to changing market conditions, call toll free for latest price and availability of product. FOR YOUR PROTECTION, WE CHECK ALL CREDIT CARD ORDERS FOR FRAUD.


```

WO 1 REM *****
PB 2 REM * MEMORY MATCH *
PS 3 REM * by Joe Brzuszek *
ZD 4 REM *
IN 5 REM * COPYRIGHT 1989 *
PM 6 REM * BY ANALOG COMPUTING *
WU 7 REM *****
NN 8 REM
HT 10 POKE 566,143:POKE 567,231:RESTORE 9
000:GOSUB 8000
PT 15 POKE 53277,6
ZA 20 POKE 53248,X:T=USR(ADR(MM$),ADR(PL$),
P1+Y,V4):T=USR(ADR(MM$),ADR("♥♥♥♥"),
P1+Y,V4):T=STICK(5)
WS 30 X=X+20*((T=7)-(T=11)):Y=Y+24*((T=13)
-(T=14)):T1=STRIG(5):IF X>53 AND X<19
5 AND Y>61 AND Y<111 AND T1 THEN 20
WI 40 IF Y<62 THEN Y=62
XA 50 IF Y>110 THEN Y=110
XG 60 POKE 53278,Z:T=USR(ADR(MM$),ADR(PL$),
P1+Y,V4):IF X<54 THEN X=54
KT 70 IF X>194 THEN X=194
IE 80 IF NOT T1 THEN POKE 53277,V2:IF PE
EK(53252) THEN 100
TO 90 GOTO 15
QL 100 SOUND Z,V1,V2,V12:SOUND Z,Z,Z,Z:X1
=INT(.25*(X-54))+V1:Y1=.125*(Y-62)+6
:K=((X1-V1)/5+V1+V8*(Y-62)/24)
NZ 110 POKE 77,Z:I2=ASC(MIX$(K)):I=V1+6*I
2:T=USR(ADR(MM$),ADR("♥♥♥♥"),P1+Y,V4):
IF MIX$(K,K)=" " THEN 15
RD 120 POSITION X1,Y1:POKE(I,I+2):"↓↑↑
↑":CARD$(I+3,I+5)
SC 130 TN=TN+V1:IF TN=V2 AND K1=K THEN TN
=V1:GOTO 15
RD 140 ON NOT 50 GOSUB 500+I2*20:IF TN=V
1 THEN K1=K:I1=I:X2=X1:Y2=Y1:GOTO 15
JL 150 N=N+V1:TN=Z:IF I1=I THEN 300
SB 160 FOR I=Z TO 425:NEXT I:POSITION X2,
Y2:POKE C1$:POSITION X1,Y1:POKE C1$:ON NOT
PL GOTO 15:5=5+V1:IF 5=V2 THEN 5=Z
XG 190 POSITION 6,V2:IF 5=Z THEN ? "PLAYE
R 1"
DV 200 ? "PLAYER 1"
YI 300 RESTORE :MIX$(K,K)=" ":MIX$(K1,K1)
=" "
DE 310 TRAP 320:READ I:FOR K=14 TO 5 STEP
-V1:SOUND Z,I,V10,K:NEXT K:GOTO 310:D
ATA 60,60,47,40,47,40,F
TA 320 FOR K=Z TO 30:NEXT K:SOUND Z,Z,Z,Z
:51=51+V2:IF 5 THEN 52=52+V2:51=51-V2
VC 330 POSITION 6,3:POKE 51,52
DA 340 FOR I=Z TO 300:NEXT I:POSITION X1,
Y1:POKE 51,52:POKE 52,51:POKE 51,52
NV 350 FOR I=Z TO 250:POKE 709,I:POKE 532
79,Z:POKE 53279,V8:NEXT I
UR 360 POSITION Z,Z:POKE 53279,V8:NEXT I
RV 370 ON PEEK(53279)=7 GOTO 370:GOSUB 80
60:GOTO 15
EA 500 FOR J=V12 TO V4 STEP -V1:FOR H=J T
O Z STEP -1:SOUND Z,200,V10,H:NEXT H:N
EXT J:RETURN
RA 520 FOR J=15 TO Z STEP -.5:SOUND Z,V2
,V2,V8:SOUND Z,Z,Z,Z:NEXT J:SOUND Z,Z,
Z,Z:RETURN
HN 540 FOR J=V12 TO Z STEP -2:FOR H=V12 T
O Z STEP -1.5:SOUND Z,19+H*2,V10,J:NEX
T H:NEXT J:SOUND Z,Z,Z,Z:RETURN
QS 560 SOUND V2,19,V4,V4:FOR H=V1 TO V2:F
OR J=Z TO 30:SOUND Z,V2,V2,V12:NEXT J:
SOUND Z,Z,Z,Z:NEXT H
JP 570 SOUND V2,Z,Z,Z:RETURN

```

```

ZC 580 FOR J=15 TO Z STEP -1.5:SOUND Z,J,
V10,J:NEXT J:SOUND Z,Z,Z,Z:RETURN
AZ 600 FOR J=V12 TO Z STEP -.4:SOUND Z,1
05,V10,J:NEXT J:FOR J=V12 TO Z STEP -.
45:SOUND Z,132,V10,J:NEXT J:RETURN
HQ 620 FOR H=15 TO Z STEP -V1:SOUND Z,H,V
10,H:SOUND Z,Z,Z,Z:NEXT H:RETURN
JB 640 FOR J=50 TO 60 STEP 6:SOUND Z,J,V1
0,V8:NEXT J:FOR J=60 TO 30 STEP -V4:50
UND Z,J,V10,V8:NEXT J:SOUND Z,Z,Z,Z
ZM 650 RETURN
YC 660 SOUND Z,49,V8,V8:SOUND V1,50,V8,V8
:FOR J=1 TO 300:NEXT J:SOUND Z,Z,Z,Z:5
OUND V1,Z,Z,Z:RETURN
FR 680 RESTORE 690:TRAP 690
TM 685 READ I2,J:FOR H=Z TO J:SOUND Z,I2,
V10,V12:NEXT H:GOTO 685
FQ 690 RETURN :DATA 91,8,108,8,136,20,108
,20,91,20,68,30,0,0,F
GL 700 FOR J=15 TO Z STEP -V2:SOUND Z,V2,
V4,V12:FOR H=Z TO J:NEXT H:SOUND Z,V2,
V8,V12:NEXT J:SOUND Z,Z,Z,Z:RETURN
GY 720 FOR J=15 TO Z STEP -.5:SOUND Z,29
,V10,J:NEXT J:SOUND Z,Z,Z,Z:RETURN
UJ 8000 READ V1,V2,V4,V8,V10,V12:P=PEEK(1
06)-V8:POKE 106,P:DIM MM$(63),PL$(9),C
ARD$(100),MIX$(24),C1$(13)
SZ 8010 GRAPHICS Z:POKE 559,Z:DL=PEEK(560
)+PEEK(561)*256:POKE DL+V8,130
GY 8020 POKE DL+19,65:POKE DL+20,PEEK(560
):POKE DL+21,PEEK(561):P1=P*256+512:FO
R K=Z TO 128:POKE P1+K,Z:NEXT K
EZ 8040 FOR K=Z TO 6:READ CARD$:Y=USR(ADR
(MM$),ADR(CARD$),(P+V4)*256+K*100,LEN(
CARD$)):NEXT K
PF 8050 READ CARD$:Y=USR(ADR(MM$),ADR(CAR
D$),1536,LEN(CARD$)):X=USR(1536)
IO 8060 POKE 53248,Z:POKE 1563,224:FOR X=
9 TO 18:POKE DL+X,7:NEXT X:POKE 708,36
:POKE 709,42:POKE 710,V2:POKE 711,40
CO 8070 POKE 82,V1:POKE 752,V1:POKE 753,
V1:POKE 754,V1:POKE 755,V1:POKE 756,V1
:POKE 757,V1:POKE 758,V1:POKE 759,V1
:POKE 760,V1:POKE 761,V1:POKE 762,V1
:POKE 763,V1:POKE 764,V1:POKE 765,V1
:POKE 766,V1:POKE 767,V1:POKE 768,V1
:POKE 769,V1:POKE 770,V1:POKE 771,V1
:POKE 772,V1:POKE 773,V1:POKE 774,V1
:POKE 775,V1:POKE 776,V1:POKE 777,V1
:POKE 778,V1:POKE 779,V1:POKE 780,V1
:POKE 781,V1:POKE 782,V1:POKE 783,V1
:POKE 784,V1:POKE 785,V1:POKE 786,V1
:POKE 787,V1:POKE 788,V1:POKE 789,V1
:POKE 790,V1:POKE 791,V1:POKE 792,V1
:POKE 793,V1:POKE 794,V1:POKE 795,V1
:POKE 796,V1:POKE 797,V1:POKE 798,V1
:POKE 799,V1:POKE 800,V1:POKE 801,V1
:POKE 802,V1:POKE 803,V1:POKE 804,V1
:POKE 805,V1:POKE 806,V1:POKE 807,V1
:POKE 808,V1:POKE 809,V1:POKE 810,V1
:POKE 811,V1:POKE 812,V1:POKE 813,V1
:POKE 814,V1:POKE 815,V1:POKE 816,V1
:POKE 817,V1:POKE 818,V1:POKE 819,V1
:POKE 820,V1:POKE 821,V1:POKE 822,V1
:POKE 823,V1:POKE 824,V1:POKE 825,V1
:POKE 826,V1:POKE 827,V1:POKE 828,V1
:POKE 829,V1:POKE 830,V1:POKE 831,V1
:POKE 832,V1:POKE 833,V1:POKE 834,V1
:POKE 835,V1:POKE 836,V1:POKE 837,V1
:POKE 838,V1:POKE 839,V1:POKE 840,V1
:POKE 841,V1:POKE 842,V1:POKE 843,V1
:POKE 844,V1:POKE 845,V1:POKE 846,V1
:POKE 847,V1:POKE 848,V1:POKE 849,V1
:POKE 850,V1:POKE 851,V1:POKE 852,V1
:POKE 853,V1:POKE 854,V1:POKE 855,V1
:POKE 856,V1:POKE 857,V1:POKE 858,V1
:POKE 859,V1:POKE 860,V1:POKE 861,V1
:POKE 862,V1:POKE 863,V1:POKE 864,V1
:POKE 865,V1:POKE 866,V1:POKE 867,V1
:POKE 868,V1:POKE 869,V1:POKE 870,V1
:POKE 871,V1:POKE 872,V1:POKE 873,V1
:POKE 874,V1:POKE 875,V1:POKE 876,V1
:POKE 877,V1:POKE 878,V1:POKE 879,V1
:POKE 880,V1:POKE 881,V1:POKE 882,V1
:POKE 883,V1:POKE 884,V1:POKE 885,V1
:POKE 886,V1:POKE 887,V1:POKE 888,V1
:POKE 889,V1:POKE 890,V1:POKE 891,V1
:POKE 892,V1:POKE 893,V1:POKE 894,V1
:POKE 895,V1:POKE 896,V1:POKE 897,V1
:POKE 898,V1:POKE 899,V1:POKE 900,V1
:POKE 901,V1:POKE 902,V1:POKE 903,V1
:POKE 904,V1:POKE 905,V1:POKE 906,V1
:POKE 907,V1:POKE 908,V1:POKE 909,V1
:POKE 910,V1:POKE 911,V1:POKE 912,V1
:POKE 913,V1:POKE 914,V1:POKE 915,V1
:POKE 916,V1:POKE 917,V1:POKE 918,V1
:POKE 919,V1:POKE 920,V1:POKE 921,V1
:POKE 922,V1:POKE 923,V1:POKE 924,V1
:POKE 925,V1:POKE 926,V1:POKE 927,V1
:POKE 928,V1:POKE 929,V1:POKE 930,V1
:POKE 931,V1:POKE 932,V1:POKE 933,V1
:POKE 934,V1:POKE 935,V1:POKE 936,V1
:POKE 937,V1:POKE 938,V1:POKE 939,V1
:POKE 940,V1:POKE 941,V1:POKE 942,V1
:POKE 943,V1:POKE 944,V1:POKE 945,V1
:POKE 946,V1:POKE 947,V1:POKE 948,V1
:POKE 949,V1:POKE 950,V1:POKE 951,V1
:POKE 952,V1:POKE 953,V1:POKE 954,V1
:POKE 955,V1:POKE 956,V1:POKE 957,V1
:POKE 958,V1:POKE 959,V1:POKE 960,V1
:POKE 961,V1:POKE 962,V1:POKE 963,V1
:POKE 964,V1:POKE 965,V1:POKE 966,V1
:POKE 967,V1:POKE 968,V1:POKE 969,V1
:POKE 970,V1:POKE 971,V1:POKE 972,V1
:POKE 973,V1:POKE 974,V1:POKE 975,V1
:POKE 976,V1:POKE 977,V1:POKE 978,V1
:POKE 979,V1:POKE 980,V1:POKE 981,V1
:POKE 982,V1:POKE 983,V1:POKE 984,V1
:POKE 985,V1:POKE 986,V1:POKE 987,V1
:POKE 988,V1:POKE 989,V1:POKE 990,V1
:POKE 991,V1:POKE 992,V1:POKE 993,V1
:POKE 994,V1:POKE 995,V1:POKE 996,V1
:POKE 997,V1:POKE 998,V1:POKE 999,V1
:POKE 1000,V1:POKE 1001,V1:POKE 1002,V1
:POKE 1003,V1:POKE 1004,V1:POKE 1005,V1
:POKE 1006,V1:POKE 1007,V1:POKE 1008,V1
:POKE 1009,V1:POKE 1010,V1:POKE 1011,V1
:POKE 1012,V1:POKE 1013,V1:POKE 1014,V1
:POKE 1015,V1:POKE 1016,V1:POKE 1017,V1
:POKE 1018,V1:POKE 1019,V1:POKE 1020,V1
:POKE 1021,V1:POKE 1022,V1:POKE 1023,V1
:POKE 1024,V1:POKE 1025,V1:POKE 1026,V1
:POKE 1027,V1:POKE 1028,V1:POKE 1029,V1
:POKE 1030,V1:POKE 1031,V1:POKE 1032,V1
:POKE 1033,V1:POKE 1034,V1:POKE 1035,V1
:POKE 1036,V1:POKE 1037,V1:POKE 1038,V1
:POKE 1039,V1:POKE 1040,V1:POKE 1041,V1
:POKE 1042,V1:POKE 1043,V1:POKE 1044,V1
:POKE 1045,V1:POKE 1046,V1:POKE 1047,V1
:POKE 1048,V1:POKE 1049,V1:POKE 1050,V1
:POKE 1051,V1:POKE 1052,V1:POKE 1053,V1
:POKE 1054,V1:POKE 1055,V1:POKE 1056,V1
:POKE 1057,V1:POKE 1058,V1:POKE 1059,V1
:POKE 1060,V1:POKE 1061,V1:POKE 1062,V1
:POKE 1063,V1:POKE 1064,V1:POKE 1065,V1
:POKE 1066,V1:POKE 1067,V1:POKE 1068,V1
:POKE 1069,V1:POKE 1070,V1:POKE 1071,V1
:POKE 1072,V1:POKE 1073,V1:POKE 1074,V1
:POKE 1075,V1:POKE 1076,V1:POKE 1077,V1
:POKE 1078,V1:POKE 1079,V1:POKE 1080,V1
:POKE 1081,V1:POKE 1082,V1:POKE 1083,V1
:POKE 1084,V1:POKE 1085,V1:POKE 1086,V1
:POKE 1087,V1:POKE 1088,V1:POKE 1089,V1
:POKE 1090,V1:POKE 1091,V1:POKE 1092,V1
:POKE 1093,V1:POKE 1094,V1:POKE 1095,V1
:POKE 1096,V1:POKE 1097,V1:POKE 1098,V1
:POKE 1099,V1:POKE 1100,V1:POKE 1101,V1
:POKE 1102,V1:POKE 1103,V1:POKE 1104,V1
:POKE 1105,V1:POKE 1106,V1:POKE 1107,V1
:POKE 1108,V1:POKE 1109,V1:POKE 1110,V1
:POKE 1111,V1:POKE 1112,V1:POKE 1113,V1
:POKE 1114,V1:POKE 1115,V1:POKE 1116,V1
:POKE 1117,V1:POKE 1118,V1:POKE 1119,V1
:POKE 1120,V1:POKE 1121,V1:POKE 1122,V1
:POKE 1123,V1:POKE 1124,V1:POKE 1125,V1
:POKE 1126,V1:POKE 1127,V1:POKE 1128,V1
:POKE 1129,V1:POKE 1130,V1:POKE 1131,V1
:POKE 1132,V1:POKE 1133,V1:POKE 1134,V1
:POKE 1135,V1:POKE 1136,V1:POKE 1137,V1
:POKE 1138,V1:POKE 1139,V1:POKE 1140,V1
:POKE 1141,V1:POKE 1142,V1:POKE 1143,V1
:POKE 1144,V1:POKE 1145,V1:POKE 1146,V1
:POKE 1147,V1:POKE 1148,V1:POKE 1149,V1
:POKE 1150,V1:POKE 1151,V1:POKE 1152,V1
:POKE 1153,V1:POKE 1154,V1:POKE 1155,V1
:POKE 1156,V1:POKE 1157,V1:POKE 1158,V1
:POKE 1159,V1:POKE 1160,V1:POKE 1161,V1
:POKE 1162,V1:POKE 1163,V1:POKE 1164,V1
:POKE 1165,V1:POKE 1166,V1:POKE 1167,V1
:POKE 1168,V1:POKE 1169,V1:POKE 1170,V1
:POKE 1171,V1:POKE 1172,V1:POKE 1173,V1
:POKE 1174,V1:POKE 1175,V1:POKE 1176,V1
:POKE 1177,V1:POKE 1178,V1:POKE 1179,V1
:POKE 1180,V1:POKE 1181,V1:POKE 1182,V1
:POKE 1183,V1:POKE 1184,V1:POKE 1185,V1
:POKE 1186,V1:POKE 1187,V1:POKE 1188,V1
:POKE 1189,V1:POKE 1190,V1:POKE 1191,V1
:POKE 1192,V1:POKE 1193,V1:POKE 1194,V1
:POKE 1195,V1:POKE 1196,V1:POKE 1197,V1
:POKE 1198,V1:POKE 1199,V1:POKE 1200,V1
:POKE 1201,V1:POKE 1202,V1:POKE 1203,V1
:POKE 1204,V1:POKE 1205,V1:POKE 1206,V1
:POKE 1207,V1:POKE 1208,V1:POKE 1209,V1
:POKE 1210,V1:POKE 1211,V1:POKE 1212,V1
:POKE 1213,V1:POKE 1214,V1:POKE 1215,V1
:POKE 1216,V1:POKE 1217,V1:POKE 1218,V1
:POKE 1219,V1:POKE 1220,V1:POKE 1221,V1
:POKE 1222,V1:POKE 1223,V1:POKE 1224,V1
:POKE 1225,V1:POKE 1226,V1:POKE 1227,V1
:POKE 1228,V1:POKE 1229,V1:POKE 1230,V1
:POKE 1231,V1:POKE 1232,V1:POKE 1233,V1
:POKE 1234,V1:POKE 1235,V1:POKE 1236,V1
:POKE 1237,V1:POKE 1238,V1:POKE 1239,V1
:POKE 1240,V1:POKE 1241,V1:POKE 1242,V1
:POKE 1243,V1:POKE 1244,V1:POKE 1245,V1
:POKE 1246,V1:POKE 1247,V1:POKE 1248,V1
:POKE 1249,V1:POKE 1250,V1:POKE 1251,V1
:POKE 1252,V1:POKE 1253,V1:POKE 1254,V1
:POKE 1255,V1:POKE 1256,V1:POKE 1257,V1
:POKE 1258,V1:POKE 1259,V1:POKE 1260,V1
:POKE 1261,V1:POKE 1262,V1:POKE 1263,V1
:POKE 1264,V1:POKE 1265,V1:POKE 1266,V1
:POKE 1267,V1:POKE 1268,V1:POKE 1269,V1
:POKE 1270,V1:POKE 1271,V1:POKE 1272,V1
:POKE 1273,V1:POKE 1274,V1:POKE 1275,V1
:POKE 1276,V1:POKE 1277,V1:POKE 1278,V1
:POKE 1279,V1:POKE 1280,V1:POKE 1281,V1
:POKE 1282,V1:POKE 1283,V1:POKE 1284,V1
:POKE 1285,V1:POKE 1286,V1:POKE 1287,V1
:POKE 1288,V1:POKE 1289,V1:POKE 1290,V1
:POKE 1291,V1:POKE 1292,V1:POKE 1293,V1
:POKE 1294,V1:POKE 1295,V1:POKE 1296,V1
:POKE 1297,V1:POKE 1298,V1:POKE 1299,V1
:POKE 1300,V1:POKE 1301,V1:POKE 1302,V1
:POKE 1303,V1:POKE 1304,V1:POKE 1305,V1
:POKE 1306,V1:POKE 1307,V1:POKE 1308,V1
:POKE 1309,V1:POKE 1310,V1:POKE 1311,V1
:POKE 1312,V1:POKE 1313,V1:POKE 1314,V1
:POKE 1315,V1:POKE 1316,V1:POKE 1317,V1
:POKE 1318,V1:POKE 1319,V1:POKE 1320,V1
:POKE 1321,V1:POKE 1322,V1:POKE 1323,V1
:POKE 1324,V1:POKE 1325,V1:POKE 1326,V1
:POKE 1327,V1:POKE 1328,V1:POKE 1329,V1
:POKE 1330,V1:POKE 1331,V1:POKE 1332,V1
:POKE 1333,V1:POKE 1334,V1:POKE 1335,V1
:POKE 1336,V1:POKE 1337,V1:POKE 1338,V1
:POKE 1339,V1:POKE 1340,V1:POKE 1341,V1
:POKE 1342,V1:POKE 1343,V1:POKE 1344,V1
:POKE 1345,V1:POKE 1346,V1:POKE 1347,V1
:POKE 1348,V1:POKE 1349,V1:POKE 1350,V1
:POKE 1351,V1:POKE 1352,V1:POKE 1353,V1
:POKE 1354,V1:POKE 1355,V1:POKE 1356,V1
:POKE 1357,V1:POKE 1358,V1:POKE 1359,V1
:POKE 1360,V1:POKE 1361,V1:POKE 1362,V1
:POKE 1363,V1:POKE 1364,V1:POKE 1365,V1
:POKE 1366,V1:POKE 1367,V1:POKE 1368,V1
:POKE 1369,V1:POKE 1370,V1:POKE 1371,V1
:POKE 1372,V1:POKE 1373,V1:POKE 1374,V1
:POKE 1375,V1:POKE 1376,V1:POKE 1377,V1
:POKE 1378,V1:POKE 1379,V1:POKE 1380,V1
:POKE 1381,V1:POKE 1382,V1:POKE 1383,V1
:POKE 1384,V1:POKE 1385,V1:POKE 1386,V1
:POKE 1387,V1:POKE 1388,V1:POKE 1389,V1
:POKE 1390,V1:POKE 1391,V1:POKE 1392,V1
:POKE 1393,V1:POKE 1394,V1:POKE 1395,V1
:POKE 1396,V1:POKE 1397,V1:POKE 1398,V1
:POKE 1399,V1:POKE 1400,V1:POKE 1401,V1
:POKE 1402,V1:POKE 1403,V1:POKE 1404,V1
:POKE 1405,V1:POKE 1406,V1:POKE 1407,V1
:POKE 1408,V1:POKE 1409,V1:POKE 1410,V1
:POKE 1411,V1:POKE 1412,V1:POKE 1413,V1
:POKE 1414,V1:POKE 1415,V1:POKE 1416,V1
:POKE 1417,V1:POKE 1418,V1:POKE 1419,V1
:POKE 1420,V1:POKE 1421,V1:POKE 1422,V1
:POKE 1423,V1:POKE 1424,V1:POKE 1425,V1
:POKE 1426,V1:POKE 1427,V1:POKE 1428,V1
:POKE 1429,V1:POKE 1430,V1:POKE 1431,V1
:POKE 1432,V1:POKE 1433,V1:POKE 1434,V1
:POKE 1435,V1:POKE 1436,V1:POKE 1437,V1
:POKE 1438,V1:POKE 1439,V1:POKE 1440,V1
:POKE 1441,V1:POKE 1442,V1:POKE 1443,V1
:POKE 1444,V1:POKE 1445,V1:POKE 1446,V1
:POKE 1447,V1:POKE 1448,V1:POKE 1449,V1
:POKE 1450,V1:POKE 1451,V1:POKE 1452,V1
:POKE 1453,V1:POKE 1454,V1:POKE 1455,V1
:POKE 1456,V1:POKE 1457,V1:POKE 1458,V1
:POKE 1459,V1:POKE 1460,V1:POKE 1461,V1
:POKE 1462,V1:POKE 1463,V1:POKE 1464,V1
:POKE 1465,V1:POKE 1466,V1:POKE 1467,V1
:POKE 1468,V1:POKE 1469,V1:POKE 1470,V1
:POKE 1471,V1:POKE 1472,V1:POKE 1473,V1
:POKE 1474,V1:POKE 1475,V1:POKE 1476,V1
:POKE 1477,V1:POKE 1478,V1:POKE 1479,V1
:POKE 1480,V1:POKE 1481,V1:POKE 1482,V1
:POKE 1483,V1:POKE 1484,V1:POKE 1485,V1
:POKE 1486,V1:POKE 1487,V1:POKE 1488,V1
:POKE 1489,V1:POKE 1490,V1:POKE 1491,V1
:POKE 1492,V1:POKE 1493,V1:POKE 1494,V1
:POKE 1495,V1:POKE 1496,V1:POKE 1497,V1
:POKE 1498,V1:POKE 1499,V1:POKE 1500,V1
:POKE 1501,V1:POKE 1502,V1:POKE 1503,V1
:POKE 1504,V1:POKE 1505,V1:POKE 1506,V1
:POKE 1507,V1:POKE 1508,V1:POKE 1509,V1
:POKE 1510,V1:POKE 1511,V1:POKE 1512,V1
:POKE 1513,V1:POKE 1514,V1:POKE 1515,V1
:POKE 1516,V1:POKE 1517,V1:POKE 1518,V1
:POKE 1519,V1:POKE 1520,V1:POKE 1521,V1
:POKE 1522,V1:POKE 1523,V1:POKE 1524,V1
:POKE 1525,V1:POKE 1526,V1:POKE 1527,V1
:POKE 1528,V1:POKE 1529,V1:POKE 1530,V1
:POKE 1531,V1:POKE 1532,V1:POKE 1533,V1
:POKE 1534,V1:POKE 1535,V1:POKE 1536,V1
:POKE 1537,V1:POKE 1538,V1:POKE 1539,V1
:POKE 1540,V1:POKE 1541,V1:POKE 1542,V1
:POKE 1543,V1:POKE 1544,V1:POKE 1545,V1
:POKE 1546,V1:POKE 1547,V1:POKE 1548,V1
:POKE 1549,V1:POKE 1550,V1:POKE 1551,V1
:POKE 1552,V1:POKE 1553,V1:POKE 1554,V1
:POKE 1555,V1:POKE 1556,V1:POKE 1557,V1
:POKE 1558,V1:POKE 1559,V1:POKE 1560,V1
:POKE 1561,V1:POKE 1562,V1:POKE 1563,V1
:POKE 1564,V1:POKE 1565,V1:POKE 1566,V1
:POKE 1567,V1:POKE 1568,V1:POKE 1569,V1
:POKE 1570,V1:POKE 1571,V1:POKE 1572,V1
:POKE 1573,V1:POKE 1574,V1:POKE 1575,V1
:POKE 1576,V1:POKE 1577,V1:POKE 1578,V1
:POKE 1579,V1:POKE 1580,V1:POKE 1581,V1
:POKE 1582,V1:POKE 1583,V1:POKE 1584,V1
:POKE 1585,V1:POKE 1586,V1:POKE 1587,V1
:POKE 1588,V1:POKE 1589,V1:POKE 1590,V1
:POKE 1591,V1:POKE 1592,V1:POKE 1593,V1
:POKE 1594,V1:POKE 1595,V1:POKE 1596,V1
:POKE 1597,V1:POKE 1598,V1:POKE 1599,V1
:POKE 1600,V1:POKE 1601,V1:POKE 1602,V1
:POKE 1603,V1:POKE 1604,V1:POKE 1605,V1
:POKE 1606,V1:POKE 1607,V1:POKE 1608,V1
:POKE 1609,V1:POKE 1610,V1:POKE 1611,V1
:POKE 1612,V1:POKE 1613,V1:POKE 1614,V1
:POKE 1615,V1:POKE 1616,V1:POKE 1617,V1
:POKE 1618,V1:POKE 1619,V1:POKE 1620,V1
:POKE 1621,V1:POKE 1622,V1:POKE 1623,V1
:POKE 1624,V1:POKE 1625,V1:POKE 1626,V1
:POKE 1627,V1:POKE 1628,V1:POKE 1629,V1
:POKE 1630,V1:POKE 1631,V1:POKE 1632,V1
:POKE 1633,V1:POKE 1634,V1:POKE 1635,V1
:POKE 1636,V1:POKE 1637,V1:POKE 1638,V1
:POKE 1639,V1:POKE 1640,V1:POKE 1641,V1
:POKE 1642,V1:POKE 1643,V1:POKE 1644,V1
:POKE 1645,V1:POKE 1646,V1:POKE 1647,V1
:POKE 1648,V1:POKE 1649,V1:POKE 1650,V1
:POKE 1651,V1:POKE 1652,V1:POKE 1653,V1
:POKE 1654,V1:POKE 1655,V1:POKE 1656,V1
:POKE 1657,V1:POKE 1658,V1:POKE 1659,V1
:POKE 1660,V1:POKE 1661,V1:POKE 1662,V1
:POKE 1663,V1:POKE 1664,V1:POKE 1665,V1
:POKE 1666,V1:POKE 1667,V1:POKE 1668,V1
:POKE 1669,V1:POKE 1670,V1:POKE 1671,V1
:POKE 1672,V1:POKE 1673,V1:POKE 1674,V1
:POKE 1675,V1:POKE 1676,V1:POKE 1677,V1
:POKE 1678,V1:POKE 1679,V1:POKE 1680,V1
:POKE 1681,V1:POKE 1682,V1:POKE 1683,V1
:POKE 1684,V1:POKE 1685,V1:POKE 1686,V1
:POKE 1687,V1:POKE 1688,V1:POKE 1689,V1
:POKE 1690,V1:POKE 1691,V1:POKE 1692,V1
:POKE 1693,V1:POKE 1694,V1:POKE 1695,V1
:POKE 1696,V1:POKE 1697,V1:POKE 1698,V1
:POKE 1699,V1:POKE 1700,V1:POKE 1701,V1
:POKE 1702,V1:POKE 1703,V1:POKE 1704,V1
:POKE 1705,V1:POKE 1706,V1:POKE 1707,V1
:POKE 1708,V1:POKE 1709,V1:POKE 1710,V1
:POKE 1711,V1:POKE 1712,V1:POKE 1713,V1
:POKE 1714,V1:POKE 1715,V1:POKE 1716,V1
:POKE 1717,V1:POKE 1718,V1:POKE 1719,V1
:POKE 1720,V1:POKE 1721,V1:POKE 1722,V1
:POKE 1723,V1:POKE 1724,V1:POKE 1725,V1
:POKE 1726,V1:POKE 1727,V1:POKE 1728,V1
:POKE 1729,V1:POKE 1730,V1:POKE 1731,V1
:POKE 1732,V1:POKE 1733,V1:POKE 1734,V1
:POKE 1735,V1:POKE 1736,V1:POKE 1737,V1
:POKE 1738,V1:POKE 1739,V1:POKE 1740,V1
:POKE 1741,V1:POKE 1742,V1:PO
```


8
BIT

```

IL  F555500FFFFF4D4500000033CFFFFF
   750 DATA FFFF3CC2F23238E8A8288A08A82
   2EEEE222202280AA2A8A20280AA2A080000
   000A8A080000000000003F3D3FFFEFEFE
MC  760 DATA 3C00FF7FDDBF AEAE8C9B393037304
   4415441FCDC7FEFAFAFAFC3F3FFFFDF2F33
   FACEEFF226655FFFFBCFFCF3F4F63F3FC
NI  770 DATA 010707070707070755F7F7F7F7F
   7F754F5F5F5F5F5F51595955140444011555
   55555555410055555545011101449B39
AD  780 DATA 3038304441544168A9068D0102A91
   18D0002A9C08D0ED6408A9868D0AD48D18D0A
   9F08D09D408408B

```

MEMORY MATCH
PROGRAM LISTINGS

8

BIT

```

1000 DATA 255,255,0,112,195,115,0,0,0,
72,128,161,142,174,142,161,6926
1010 DATA 142,172,142,175,142,167,142,
128,163,239,237,240,245,244,233,238,96
00
1020 DATA 231,128,176,242,229,243,229,
238,244,243,128,200,0,0,0,0,7547
1030 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,1030
1040 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,1040
1050 DATA 0,0,0,0,0,0,78,78,78,78,78,7
8,78,78,78,78,20
1060 DATA 78,78,78,78,78,78,78,78,78,7
8,78,78,78,78,78,1668
1070 DATA 78,78,78,78,78,78,78,78,78,7
8,78,78,78,78,128,78,2428
1080 DATA 78,78,78,78,78,78,78,78,78,70,0
,44,97,114,114,121,0,1059
1090 DATA 0,34,108,97,99,107,7,115,0,7
1,78,78,78,78,78,78,1004
1100 DATA 78,78,78,78,78,128,128,128,8
9,0,0,0,0,0,0,5759
1110 DATA 77,77,77,77,77,77,77,77,77,7
7,77,77,77,77,77,1582
1120 DATA 77,77,0,0,0,0,0,0,128,
128,128,128,128,311
1130 DATA 86,0,204,128,128,128,128,128
,0,0,0,84,0,0,0,6676
1140 DATA 73,85,0,0,0,0,0,73,79,0,0,0,
0,0,0,0,2678
1150 DATA 0,0,0,217,128,128,128,128,86
,0,128,0,128,128,0,128,3032
1160 DATA 0,0,0,0,0,0,0,85,128,128,85,
0,0,0,66,128,8245
1170 DATA 89,0,0,0,0,85,85,85,85,0,0,2
17,128,128,128,128,3837
1180 DATA 86,0,0,0,128,128,0,0,0,73,
128,0,0,0,0,5013
1190 DATA 128,128,0,0,0,0,66,128,89,0,
0,0,128,128,0,0,7317
1200 DATA 128,0,0,217,128,128,128,194,
0,0,0,0,128,128,0,0,9508
1210 DATA 0,0,217,128,0,0,0,128,128,
0,0,0,0,66,128,7843
1220 DATA 89,0,0,0,128,128,213,213,213
,0,0,0,128,128,128,194,6309
1230 DATA 0,0,0,0,128,128,0,0,0,217,
128,0,0,0,0,6561
1240 DATA 128,128,204,89,0,0,66,128,89
,0,0,0,128,128,85,85,970
1250 DATA 85,0,0,0,128,128,128,194,0,0
,0,0,0,0,0,5191
1260 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,1260
1270 DATA 0,0,0,0,0,0,0,0,0,0,0,128,
128,128,89,8070
1280 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,1280
1290 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,1290
1300 DATA 0,0,0,0,217,128,128,89,73,12
8,74,0,0,72,128,89,1864
1310 DATA 0,73,128,128,128,128,89,0,73
,128,0,72,128,200,0,73,2816
1320 DATA 128,128,128,89,0,73,128,128,
128,128,89,0,217,128,128,89,6170
1330 DATA 217,128,128,74,72,128,128,89
,0,217,128,0,0,128,89,0,1924
1340 DATA 217,128,72,128,200,0,0,217,1
28,0,0,0,0,217,128,0,1387
1350 DATA 0,214,89,0,217,128,128,86,21
7,128,202,128,128,200,128,89,281
1360 DATA 0,217,128,128,128,128,89,0,2
17,128,128,194,0,0,0,217,5162
1370 DATA 128,128,89,0,0,217,128,128,1
28,128,86,0,66,128,128,86,4567
1380 DATA 217,128,0,214,194,0,128,89,0
,217,128,0,0,128,89,0,1992
1390 DATA 217,128,202,128,74,0,0,217,1
28,0,0,0,0,217,128,202,4429
1400 DATA 128,74,0,0,66,128,128,86,217
,128,0,0,0,0,128,89,935
1410 DATA 0,217,128,0,0,128,89,0,217,1
28,0,202,128,74,0,217,5448
1420 DATA 128,128,128,89,0,217,128,0,2
02,128,74,0,66,128,128,0,3224
1430 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,1430
1440 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,1440
1450 DATA 0,0,0,0,0,128,128,0,0,0,0,0,
0,0,0,0,3114
1460 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,1460
1470 DATA 0,0,0,0,0,0,0,0,0,0,8,99,9,1
28,128,72,7727
1480 DATA 74,72,74,72,74,72,74,72,74,7
2,74,72,74,72,74,72,1400
1490 DATA 74,72,74,72,74,72,74,72,74,7
2,74,72,74,72,74,72,1410
1500 DATA 74,72,74,72,74,128,0,85,82,8
2,82,82,82,82,82,82,2246
1510 DATA 82,82,82,82,72,78,78,78,78,7
8,78,78,78,74,82,82,2196
1520 DATA 82,82,82,82,82,82,82,82,82,8
2,82,82,85,0,0,0,9021
1530 DATA 77,77,77,77,77,77,77,77,77,7
7,77,72,128,78,78,78,2650
1540 DATA 78,78,78,78,128,74,77,77,
77,77,77,77,77,77,2312
1550 DATA 77,77,77,77,0,0,0,0,0,0,0,
0,0,0,0,2320
1560 DATA 0,0,70,0,44,47,33,36,41,46,3
9,14,14,14,0,71,5731
1570 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,1570
1580 DATA 0,0,0,0,0,0,0,0,0,70,77,77,77,
77,77,77,77,9217
1590 DATA 77,77,77,77,77,77,77,77,77,7
1,0,0,0,0,0,0,5765
1600 DATA 0,0,0,0,0,0,242,252,0,252,25
5,255,102,73,212,73,391
1610 DATA 32,170,73,120,160,0,140,47,2
,132,203,169,112,133,204,165,9454
1620 DATA 88,133,207,165,89,133,208,16
2,2,177,203,145,207,200,208,249,5606
1630 DATA 230,204,230,208,202,16,242,1
77,203,145,207,200,192,192,208,247,821
6
1640 DATA 162,0,177,203,157,197,2,200,
232,224,4,208,245,169,34,141,1911
1650 DATA 47,2,88,96,162,96,169,12,157
,66,3,32,86,228,162,96,5780
1660 DATA 169,3,157,66,3,169,28,157,74
,3,169,0,157,75,3,169,3446
1670 DATA 210,157,68,3,169,73,157,69,3
,76,86,228,83,58,155,226,7645
1680 DATA 2,227,2,102,73,255,255,203,0
,205,0,132,128,64,104,54,6472
1690 DATA 99,55,67,69,76,83,88,9,56,5,
56,250,59,21,62,39,727
1700 DATA 60,27,28,29,30,31,126,127,15
6,157,158,159,254,255,246,247,5382
1710 DATA 200,202,206,207,200,2,255,20
0,2,0,197,2,255,197,2,0,6469
1720 DATA 198,2,255,198,2,0,0,0,0,0,68
,49,58,102,105,108,310
1730 DATA 101,110,97,109,101,46,101,12
0,116,155,68,73,58,42,46,42,2148
1740 DATA 155,96,112,66,176,123,0,66,6
4,132,2,2,2,2,2,2,2,6891
1750 DATA 2,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,2022
1760 DATA 2,65,183,54,32,128,75,58,155
,160,0,0,255,255,0,112,6246
1770 DATA 195,115,179,163,178,165,165,
174,128,128,172,175,161,164,162,169,37
58
1780 DATA 174,161,178,185,128,128,179,
161,182,165,165,106,201,136,16,3,8679
1790 DATA 108,10,0,141,223,54,165,12,1
41,91,60,165,13,141,92,60,4474
1800 DATA 173,36,2,141,224,54,173,37,2
,141,225,54,169,0,133,206,7605
1810 DATA 32,23,63,76,93,60,32,205,55,
32,0,64,32,162,55,162,2754
1820 DATA 2,32,144,55,162,16,32,123,59
,162,16,160,220,169,54,32,5447

```


1830 DATA 148,59,160,4,32,131,59,32,12
8,55,41,127,201,49,48,10,2063
1840 DATA 201,58,16,6,141,177,54,76,11
1,60,41,95,162,4,100,55,2714
1850 DATA 95,56,221,104,54,240,6,202,1
6,248,76,77,55,138,10,170,6405
1860 DATA 189,109,54,133,212,232,189,1
09,54,133,213,108,212,0,162,16,8505
1870 DATA 169,7,157,66,3,169,0,168,32,
156,59,76,86,228,32,181,6256
1880 DATA 55,189,216,54,141,48,2,189,2
17,54,141,49,2,76,175,55,4923
1890 DATA 169,178,141,200,2,141,198,2,
169,12,141,197,2,169,34,141,6610
1900 DATA 47,2,96,169,0,240,248,169,13
6,133,106,32,170,73,32,181,8155
1910 DATA 55,133,82,169,64,133,16,141,
14,210,96,165,203,162,124,32,8052
1920 DATA 219,55,32,185,55,169,124,166
203,133,213,134,208,169,64,133,1836
1930 DATA 212,133,207,160,0,162,2,177,
212,145,207,200,208,249,230,213,7154
1940 DATA 230,208,202,16,242,160,192,1
98,212,198,207,177,212,145,207,136,671
0
1950 DATA 208,249,96,165,206,208,6,32,
185,55,32,226,62,32,41,57,4240
1960 DATA 165,206,208,2,230,206,32,67,
57,32,175,55,32,242,56,32,4889
1970 DATA 88,57,173,252,2,201,255,208,
34,173,31,208,201,3,240,21,9828
1980 DATA 201,5,208,3,76,47,55,201,6,2
08,3,76,7,62,173,220,5635
1990 DATA 2,201,17,208,221,32,2,57,76,
30,56,32,231,56,232,208,7623
2000 DATA 203,32,128,55,162,12,221,119
54,240,53,202,16,248,96,56,8161
2010 DATA 91,57,166,84,224,23,208,10,2
01,155,240,46,166,85,224,39,9726
2020 DATA 240,79,166,85,224,39,208,9,2
01,155,240,5,32,244,62,169,9663
2030 DATA 155,72,169,0,170,168,32,156,
59,169,11,157,66,3,169,27,4259
2040 DATA 32,86,228,104,32,164,56,76,3
0,56,166,85,240,3,76,194,6570
2050 DATA 56,169,29,76,146,56,72,169,2
55,162,2,157,178,2,202,16,7206
2060 DATA 250,169,0,168,170,32,156,59,
169,11,157,66,3,104,76,86,4087
2070 DATA 228,32,244,62,32,200,56,76,3
0,56,32,220,56,177,212,72,7062
2080 DATA 32,154,57,104,133,93,24,105,
128,160,0,145,212,96,169,64,7389
2090 DATA 133,212,165,203,133,213,160,
0,96,162,5,221,132,54,240,127,312
2100 DATA 202,16,248,96,169,8,141,31,2
08,169,255,141,252,2,169,0,9488
2110 DATA 141,220,2,96,120,32,33,60,16
9,64,133,212,133,213,32,9,6087
2120 DATA 64,165,176,141,186,54,169,58
141,200,2,141,198,2,169,0,6627
2130 DATA 141,197,2,162,2,32,144,55,32
128,55,32,29,60,162,0,1173
2140 DATA 32,144,55,169,216,141,186,54
169,0,133,19,169,7,160,233,8596
2150 DATA 162,61,32,92,228,162,3,189,1
56,54,157,197,2,202,16,247,9624
2160 DATA 162,1,181,186,149,84,202,16,
249,96,162,3,189,197,92,57,8928
2170 DATA 87,58,2,157,156,54,202,16,24
7,162,1,181,84,149,186,202,879
2180 DATA 16,249,96,138,133,212,10,24,
101,212,170,189,138,54,133,212,837
2190 DATA 232,189,138,54,133,213,160,0
232,189,138,54,48,10,177,212,9448
2200 DATA 24,105,2,145,212,76,30,56,17
7,212,56,233,2,76,141,57,6436
2210 DATA 169,0,133,84,133,85,230,85,1
69,126,76,164,56,32,173,57,6847
2220 DATA 76,93,59,160,132,169,71,162,
152,32,164,59,169,0,133,207,8470
2230 DATA 133,208,166,207,169,128,157,
69,131,224,14,240,5,169,0,157,8765
2240 DATA 70,131,32,128,55,166,207,201
155,208,7,224,3,48,42,76,6305

2250 DATA 160,58,41,127,201,27,240,33,
224,0,208,34,201,58,208,2,7557
2260 DATA 169,49,201,49,48,204,201,59,
16,200,32,139,58,230,207,169,616
2270 DATA 58,32,139,58,230,207,76,188,
57,104,104,76,47,55,201,126,7490
2280 DATA 208,29,198,207,169,32,139
58,165,208,240,2,198,208,165,2239
2290 DATA 207,201,1,208,157,198,207,16
9,32,32,139,58,76,188,57,201,9032
2300 DATA 46,208,20,166,207,224,3,48,1
37,166,208,208,133,32,139,58,9137
2310 DATA 230,207,230,208,76,188,57,20
1,42,240,25,201,63,240,21,201,1166
2320 DATA 49,48,241,201,58,16,3,76,94,
58,41,95,201,65,88,58,3795
2330 DATA 83,59,48,228,201,91,16,224,1
66,207,224,10,208,13,166,208,1894
2340 DATA 208,9,72,169,46,32,139,58,10
4,230,207,166,207,224,14,240,2699
2350 DATA 199,166,208,224,4,240,193,32
139,58,230,207,165,208,240,184,5914
2360 DATA 230,208,76,188,57,166,207,15
7,161,54,56,233,32,157,69,131,9118
2370 DATA 224,13,240,5,169,0,157,70,13
1,96,157,161,54,169,0,157,7242
2380 DATA 69,131,32,33,60,169,31,32,16
4,56,32,220,56,177,212,72,7292
2390 DATA 32,154,57,104,160,0,145,212,
32,104,60,162,32,32,123,59,4413
2400 DATA 162,32,160,160,169,54,32,148
59,164,209,32,131,59,152,48,6754
2410 DATA 108,162,32,165,210,157,66,3,
160,58,169,132,32,148,59,160,7472
2420 DATA 6,169,0,32,156,59,32,86,228,
152,48,81,162,32,165,210,8399
2430 DATA 157,66,3,201,11,240,18,160,5
185,58,132,217,226,54,208,673
2440 DATA 5,136,16,245,48,3,76,80,62,1
60,192,169,3,32,156,59,5244
2450 DATA 160,64,169,132,32,148,59,32,
86,228,152,16,3,76,67,59,3460
2460 DATA 160,156,169,54,162,32,32,148
59,160,4,169,0,32,156,59,4000
2470 DATA 32,86,228,152,48,7,165,206,2
08,2,230,206,96,32,195,59,9510
2480 DATA 169,253,32,164,56,160,157,16
9,72,162,203,32,164,59,84,59,7645
2490 DATA 79,60,32,128,55,104,104,169,
8,133,209,162,32,32,123,59,5574
2500 DATA 32,29,60,32,220,56,177,212,1
33,93,32,162,57,165,209,201,1094
2510 DATA 8,240,3,76,5,56,76,47,55,169
12,157,66,3,76,86,2197
2520 DATA 228,169,3,157,66,3,152,157,7
4,3,169,0,157,75,3,76,3298
2530 DATA 86,228,157,69,3,152,157,68,3
96,157,73,3,152,157,72,5653
2540 DATA 3,96,132,212,133,213,134,207
169,130,133,208,160,0,177,212,3423
2550 DATA 201,96,240,12,145,207,200,20
8,245,230,213,230,208,76,176,59,5702
2560 DATA 96,56,233,100,72,162,0,56,23
3,10,48,3,232,208,249,142,343
2570 DATA 170,72,169,0,24,105,10,202,2
08,250,141,171,72,104,56,237,826
2580 DATA 171,72,141,171,72,162,1,189,
170,72,24,105,144,157,170,72,8399
2590 DATA 202,16,244,169,8,133,209,96,
169,232,162,54,160,4,32,6,6284
2600 DATA 60,76,167,57,133,207,134,208
152,133,209,24,105,3,133,210,97
2610 DATA 160,11,177,207,153,186,71,13
6,16,248,96,162,0,240,2,162,9223
2620 DATA 2,142,240,2,96,162,16,32,123
59,169,2,133,82,173,223,8074
2630 DATA 54,133,106,32,170,73,169,64,
141,190,2,169,0,141,182,2,6334
2640 DATA 173,91,60,133,12,173,92,60,1
33,13,172,224,54,174,80,60,7134
2650 DATA 75,61,225,54,169,7,32,92,228
108,10,0,32,0,0,169,1947
2660 DATA 90,133,12,169,60,133,13,76,4
4,55,169,0,133,186,133,187,7650
2670 DATA 96,169,64,133,212,165,204,13

8

BIX

3,213,162,2,160,0,152,133,209,1316
 2680 DATA 145,212,200,208,251,230,213,
 202,16,246,169,128,133,213,173,177,656
 0
 2690 DATA 54,24,105,96,141,29,73,162,4
 0,169,0,168,32,141,61,162,6140
 2700 DATA 32,32,123,59,162,32,169,3,15
 7,66,3,169,6,157,74,3,3178
 2710 DATA 160,176,169,54,32,148,59,32,
 86,228,152,16,3,76,95,61,4084
 2720 DATA 162,32,169,5,157,66,3,160,0,
 169,4,32,148,59,160,20,3543
 2730 DATA 169,0,32,156,59,32,86,228,15
 2,48,74,173,0,4,201,48,5109
 2740 DATA 48,6,32,181,61,76,247,60,160
 ,124,166,209,224,2,208,3,9712
 2750 DATA 140,13,4,140,0,4,141,1,4,160
 ,13,185,0,4,201,97,3129
 2760 DATA 16,3,56,233,32,145,212,136,1
 6,241,162,13,164,209,200,192,3106
 2770 DATA 3,208,3,160,0,232,132,209,13
 8,24,101,212,133,212,144,2,9852
 2780 DATA 230,213,76,190,60,76,70,61,1
 66,209,240,27,165,212,24,105,9859
 2790 DATA 13,133,212,144,2,230,213,232
 ,224,3,208,242,165,212,24,105,3409
 2800 DATA 1,133,212,144,2,230,213,96,1
 62,32,32,123,59,162,76,61,6685
 2810 DATA 71,62,27,169,40,160,0,32,141
 ,61,169,27,141,76,61,32,3564
 2820 DATA 128,55,76,47,55,32,38,61,32,
 169,61,162,30,169,66,160,5594
 2830 DATA 5,32,141,61,169,253,32,164,5
 6,169,17,141,76,61,169,2,5947
 2840 DATA 133,209,32,38,61,165,212,56,
 233,3,133,212,176,2,198,213,1694
 2850 DATA 76,70,61,133,207,169,73,133,
 208,152,24,101,212,133,210,165,2681
 2860 DATA 213,105,0,133,211,160,0,177,
 207,145,210,200,202,208,248,165,7167
 2870 DATA 212,24,105,40,133,212,144,2,
 230,213,96,166,209,240,33,160,2946
 2880 DATA 0,169,124,145,212,160,13,145
 ,212,165,212,24,105,13,133,212,553
 2890 DATA 144,2,230,213,232,224,3,208,
 230,162,0,134,209,32,58,61,9078
 2900 DATA 160,13,185,0,4,24,105,96,153
 ,0,4,136,16,244,96,165,6065
 2910 DATA 19,201,4,208,21,169,216,205,
 186,54,240,6,141,186,54,76,9835
 2920 DATA 0,62,169,176,208,246,169,0,1
 33,19,76,98,228,32,0,64,5789
 2930 DATA 162,2,32,144,55,32,88,57,32,
 162,55,160,8,169,244,162,8462
 2940 DATA 54,32,6,60,162,5,189,226,54,
 157,58,132,202,16,247,32,8632
 2950 DATA 173,57,162,32,169,11,157,66,
 3,160,96,169,73,32,148,59,5661
 2960 DATA 160,123,169,0,32,156,59,32,8
 6,228,152,48,3,76,72,62,4115
 2970 DATA 67,63,93,59,32,67,59,76,93,5
 9,32,123,59,162,32,160,4591
 2980 DATA 160,169,54,32,148,59,160,4,3
 2,131,59,152,16,3,76,67,2547
 2990 DATA 59,169,125,32,164,56,169,1,1
 41,254,2,169,23,133,210,162,9999
 3000 DATA 32,169,5,157,66,3,160,40,169
 ,0,140,72,3,141,73,3,2882
 3010 DATA 32,156,59,160,0,169,4,140,68
 ,3,141,69,3,32,148,59,3005
 3020 DATA 32,86,228,192,137,240,8,192,
 128,48,4,32,200,62,96,169,8065
 3030 DATA 9,166,210,208,15,162,32,189,
 72,3,56,233,1,240,19,141,7620
 3040 DATA 72,3,169,11,162,0,141,66,3,3
 2,86,228,198,210,165,210,1372
 3050 DATA 16,173,160,0,140,254,2,132,8
 4,132,85,169,155,32,164,56,8044
 3060 DATA 198,84,169,64,133,94,165,206
 ,208,2,230,206,169,0,141,198,2595
 3070 DATA 2,169,4,141,200,2,169,12,141
 ,197,2,76,88,57,162,0,4822
 3080 DATA 201,128,48,3,41,127,232,201,
 96,16,12,201,32,48,5,56,3519
 3090 DATA 233,32,176,3,24,105,64,224,0

,240,3,24,105,128,133,93,6278
 3100 DATA 96,32,242,56,169,188,133,207
 ,160,14,185,41,73,24,105,128,7785
 3110 DATA 145,207,136,16,245,200,132,2
 0,165,207,141,33,63,165,208,141,1657
 3120 DATA 34,63,165,20,201,192,240,224
 ,173,252,2,201,255,240,68,63,4698
 3130 DATA 97,63,243,76,242,56,120,169,
 0,133,212,165,205,133,213,162,4080
 3140 DATA 71,169,0,168,145,212,200,208
 ,251,230,213,202,16,246,88,96,5116
 3150 DATA 226,2,227,2,72,63,0,64,49,64
 ,120,169,207,133,212,169,185
 3160 DATA 67,133,213,169,64,133,207,16
 5,204,133,208,162,2,160,0,177,1191
 3170 DATA 212,145,207,200,208,249,230,
 213,230,208,202,16,242,198,212,198,977
 1
 3180 DATA 207,160,192,177,212,145,207,
 136,208,249,88,96,196,64,210,64,3558
 3190 DATA 209,210,210,210,210,210,210,
 210,210,210,210,210,210,197,236,19
 70
 3200 DATA 64,250,64,252,221,128,168,16
 5,172,176,128,173,165,174,181,128,5469
 3210 DATA 221,252,8,65,48,65,209,210,2
 10,210,210,210,210,210,210,8992
 3220 DATA 210,210,216,210,210,210,210,
 210,210,210,210,210,210,210,210,17
 98
 3230 DATA 216,210,210,210,210,210,210,
 210,210,210,210,210,210,197,252,87,276
 3240 DATA 65,121,65,252,252,0,163,175,
 174,180,178,175,172,0,11,0,8376
 3250 DATA 220,221,222,223,0,29,0,35,11
 7,114,115,111,114,0,109,111,5607
 3260 DATA 118,101,109,101,110,116,127,
 65,128,65,252,252,167,65,32,66,9181
 3270 DATA 252,252,0,179,168,166,180,15
 ,163,180,178,172,0,11,0,220,8921
 3280 DATA 221,0,29,0,35,104,97,110,103
 ,101,0,115,99,114,101,101,5277
 3290 DATA 110,0,99,111,108,111,114,0,2
 52,252,0,179,168,166,180,15,529
 3300 DATA 163,180,178,172,0,11,0,156,1
 58,0,29,0,35,104,97,110,3226
 3310 DATA 103,101,0,98,111,114,100,101
 ,114,0,99,111,108,111,114,0,4869
 3320 DATA 252,252,0,179,168,166,180,15
 ,163,180,178,172,0,11,0,175,8251
 3330 DATA 176,0,29,0,35,104,97,110,103
 ,101,0,116,101,120,116,0,4013
 3340 DATA 108,117,109,105,110,101,110,
 14,252,252,71,66,72,66,252,252,2500
 3350 DATA 80,66,99,66,128,175,176,180,
 169,175,174,128,0,29,0,40,6252
 3360 DATA 37,44,48,0,109,101,110,117,1
 11,66,112,66,252,252,120,66,9829
 3370 DATA 139,66,128,179,165,172,165,1
 63,180,128,0,29,0,45,97,105,6070
 3380 DATA 110,0,109,101,110,117,151,66
 ,152,66,252,252,160,66,181,66,1657
 3390 DATA 128,179,180,161,178,180,128,
 128,0,29,0,51,97,118,101,0,4280
 3400 DATA 115,99,114,101,101,110,191,6
 6,192,66,252,252,231,66,49,67,1407
 3410 DATA 252,218,210,210,210,210,210,
 210,210,210,210,210,210,210,210,20
 28
 3420 DATA 210,210,210,210,210,210,210,
 210,210,210,210,210,210,210,210,19
 80
 3430 DATA 210,210,215,210,210,210,210,
 210,195,0,0,0,0,0,252,223,108
 3440 DATA 48,50,37,51,51,0,33,46,57,0,
 43,37,57,0,52,47,8460
 3450 DATA 0,35,47,46,52,41,46,53,37,22
 2,252,61,67,89,67,218,7764
 3460 DATA 210,210,210,210,210,210,210,
 210,210,210,210,210,210,210,210,20
 20
 3470 DATA 210,210,210,210,210,210,210,
 210,210,210,210,195,3,68,18,68,2019
 3480 DATA 72,128,180,169,180,172,165,1
 28,128,173,161,171,165,178,128,200,559


```

7
3490 DATA 81,68,100,68,8,99,9,17,25,24
,25,0,0,44,97,114,9747
3500 DATA 114,121,0,34,108,97,99,107,1
51,68,191,68,209,210,210,210,3786
3510 DATA 210,210,210,210,210,210,210,
210,210,196,128,173,225,233,238,128,20
9
3520 DATA 173,229,238,245,128,193,210,
210,210,210,210,210,210,210,210,17
93
3530 DATA 210,210,197,0,252,229,68,1,6
9,252,0,252,0,0,187,163,9447
3540 DATA 189,0,0,35,108,101,97,114,0,
37,100,105,116,111,114,0,4108
3550 DATA 109,101,109,111,114,121,13,6
9,35,69,252,0,252,0,0,187,6616
3560 DATA 165,189,0,0,37,100,105,116,1
11,114,0,115,99,114,101,101,6084
3570 DATA 110,53,69,85,69,252,0,252,0,
0,187,172,189,0,0,44,5488
3580 DATA 111,97,100,0,97,0,115,99,114
,101,101,110,0,102,114,111,5648
3590 DATA 109,0,100,105,115,107,93,69,
125,69,252,0,252,0,0,187,7694
3600 DATA 179,189,0,0,51,97,118,101,0,
98,105,110,97,114,121,0,4755
3610 DATA 108,111,97,100,0,115,99,114,
101,101,110,133,69,153,69,252,9757
3620 DATA 0,252,0,0,187,184,189,0,0,37
,120,105,116,0,116,111,5460
3630 DATA 0,36,47,51,173,69,196,69,252
,0,252,0,187,145,141,153,1314
3640 DATA 189,0,36,105,115,107,0,36,10
5,114,101,99,116,111,114,121,6954
3650 DATA 213,69,215,69,252,0,252,253,
69,37,70,252,0,218,210,210,4317
3660 DATA 210,210,210,210,210,210,210,
210,210,210,210,210,210,210,210,22
20
3670 DATA 210,210,210,210,210,210,210,
210,210,210,210,210,210,210,210,22
30
3680 DATA 210,210,210,195,80,70,91,70,
189,128,179,229,236,229,227,244,9018
3690 DATA 128,158,0,128,143,71,160,71,
81,82,82,82,82,82,82,5666
3700 DATA 82,82,82,82,82,82,82,82,82,6
9,183,71,200,71,124,72,7551
3710 DATA 128,179,163,178,165,165,174,
128,128,179,161,182,165,128,74,124,338
2
3720 DATA 223,71,240,71,124,36,110,26,
102,105,108,101,110,97,109,101,7310
3730 DATA 14,101,120,116,0,124,7,72,24
,72,65,82,82,82,82,82,3530
3740 DATA 82,82,82,82,82,82,82,82,82,8
2,82,68,47,72,48,72,3459
3750 DATA 124,36,64,72,64,72,124,87,72
,105,72,90,82,82,82,82,5068
3760 DATA 82,82,82,82,82,82,82,82,82,8
2,82,82,67,96,157,72,5878
3770 DATA 174,72,124,72,128,165,242,24
,2,239,242,128,141,128,131,145,144,5656
3780 DATA 144,128,74,124,197,72,215,72
,124,128,156,176,242,229,243,243,8505
3790 DATA 128,225,238,249,128,235,229,
249,158,124,96,0,73,218,73,128,2585
3800 DATA 128,128,128,128,128,128,128,
128,164,233,242,229,227,244,239,242,14
48
3810 DATA 249,128,239,230,128,164,242,
233,246,229,128,131,145,128,128,128,62
63
3820 DATA 128,128,128,128,128,128,128,
128,156,176,242,229,243,243,128,225,90
83
3830 DATA 238,249,128,235,229,249,158,
128,128,128,128,128,128,128,128,128,34
59
3840 DATA 128,128,164,233,243,235,128,
164,233,242,229,227,244,239,242,249,43
73
3850 DATA 128,161,227,227,229,243,243,
128,165,178,178,175,178,129,128,255,86

```

```

60
3860 DATA 255,102,73,212,73,32,170,73,
120,160,0,140,47,2,132,203,7944
3870 DATA 169,112,133,204,165,88,133,2
07,165,89,133,208,162,2,177,203,3789
3880 DATA 145,207,200,208,249,230,204,
230,208,202,16,242,177,203,145,207,936
6
3890 DATA 200,192,192,208,247,162,0,17
7,203,157,197,2,200,232,224,4,4365
3900 DATA 208,245,169,34,141,47,2,88,9
6,162,96,169,12,157,66,3,5906
3910 DATA 32,86,228,162,96,169,3,157,6
6,3,169,28,157,74,3,169,6862
3920 DATA 0,157,75,3,169,210,157,68,3,
169,73,157,69,3,76,86,6078
3930 DATA 228,83,58,155,226,2,227,2,10
2,73,176,123,255,123,128,128,1930
3940 DATA 128,128,128,180,169,180,172,
165,128,173,161,171,165,178,128,128,51
87
3950 DATA 128,128,128,128,226,249,128,
172,225,242,242,249,128,162,236,225,12
93
3960 DATA 227,235,128,128,128,128,128,
128,128,128,128,176,242,229,243,243,87
18
3970 DATA 128,187,175,176,180,169,175,
174,189,128,230,239,242,128,168,165,87
09
3980 DATA 172,176,128,179,227,242,229,
229,238,128,128,128,128,128,32,128,397
6
3990 DATA 63,128,112,112,112,66,64,128
,2,2,2,2,2,2,2,2,7721
4000 DATA 2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,65,5280
4010 DATA 32,128,226,2,227,2,0,55,0,0,
0,0,0,0,0,0,6571

```

LISTING 2: BASIC

```

IT 0 REM Title Maker Listing 2
HV 1 REM Save as TITLMAKR.BAS
DW 2 REM Creates binary file to retain
MS 3 REM Title screen without READY
VD 4 REM prompt and RUN "D:AUTORUN.BAS"
PV 5 REM /Append to Title screen file.
PW 10 DIM F$(14):F$="D1:AUTOBOOT.TM"
XL 20 GRAPHICS 0:POKE 710,34:POKE 709,42
OL 30 TRAP 20:IF ? "What drive to write "
: ? F$(4);" to > ";:INPUT I
TI 40 IF I<0 OR I>9 THEN 20
LS 50 TRAP 40000:F$(2,2)=CHR$(I+48):CLOSE
#1:OPEN #1,8,0,F$:RESTORE
SX 60 POKE 752,1: ? : ?
VK 70 FOR I=1 TO 120:READ BYTE: ? I;"+"
DG 80 PUT #1,BYTE:NEXT I: ? "+++ bytes"
WP 90 ? : ? "File saved as ";F$
JR 100 CLOSE #1:POKE 710,148:POKE 752,0:E
ND
QG 110 DATA 89,0,89,0,112,0,4,98,4,173,31
,208,41,4,240,10
CZ 120 DATA 169,18,141,33,3,169,4,141,34,
3,96,251,243,51,246,33
RK 130 DATA 4,163,246,51,246,60,246,76,22
8,243,0,238,33,4,172,33
DB 140 DATA 4,192,39,208,10,169,0,141,33,
3,169,228,141,34,3,185
EC 150 DATA 59,4,160,1,96,67,65,82,155,80
,79,75,46,56,57,44
HU 160 DATA 80,69,69,75,40,53,54,49,41,58
,82,85,46,34,68,58
PZ 170 DATA 65,85,84,79,82,85,78,46,66,65
,83,155,226,2,227,2
VK 180 DATA 0,4,224,2,225,2,17,4

```




```

Rem *****
Rem *
Rem *      Survey Sweepstakes      *
Rem *
Rem *      by A. Baggetta          *
Rem *      Copyright 1989          *
Rem *      by ANALOG Computing     *
Rem *
Rem *****
Dim Oldcolri(15), Oldcolrr(15), Oldcolrg(15), Oldcolrb(15)
Dim Name$(9), Nv(10), Chs$(11)
Dim Q0$(100), Q1$(100), I0$(100), I1$(100), I2$(100), I3$(100), I4$(100)
Dim I5$(100), I6$(100), I7$(100), I8$(100), I9$(100)
,
If Xbios(4)<>0
  Alert 3,"Low Rez Only",1,"Sorry",Dummy%
End
Endif
@Colr_get
Round=0
Sr=99
Turn=0
Sound 1,0
Wave 1,0
Setcolor 4,2,1,4
Setcolor 10,7,7,7
Setcolor 9,7,7,7
Print At(14,2);"Select A File"
@Lod_file
If Ed=1 Then
  Goto Finish
Endif
Hidem
Storage$=""
Your_name$="-----"
Topgame:
If Round=Sr Then
  @Winner_screen
  If Ik$="Y" Or Ik$="y" Then
    Turn=0
    Ik$=""
    Round=0
    Sr=99
    Goto Topgame
  Endif
  If Ik$="N" Or Ik$="n" Then
    Cls
    Goto Finish
  Endif
Endif
@Opening
@Place_question
Leave=0
Repeat
  If C_t_y=1 Then
    @Input_your_answers
  Endif
  ,
  , quit
  ,
  If Ik$="Y" Or Ik$="y" Then
    Turn=0
    Ik$=""
    Round=0
    Sr=99
    Goto Topgame
  Endif
  If Ik$="N" Or Ik$="n" Then
    Cls
    Goto Finish
  Endif
  ,
  If C_t_n=1 Then
    @Input_norman_answer
  Endif
Until Leave=1
If Rev$="Y" Or Rev$="y" Then
  Prbox 10,60,150,150
  Prbox 160,60,300,150
  Ancol=3
  Anrow=9
  For Blank=9 To 17
    Print At(3,Blank);"
  Next Blank
  Print At(Ancol,Anrow);"Correct Answers"

```




```

Anrow=Anrow+1
For Ani=0 To 4
  Print At(Ancol, Anrow+(Ani+1));Chs$(Ani)
Next Ani
Print At(Ancol, Anrow+(Ani+1)+2); "***Press Return**"
Repeat
  Until Inkey$<>""
Endif
Leave=0
Goto End_round
'
Face_data:
Data 0,0,11,0,11,0,17,3,17,3,25,11,25,11,29,23,29,23,31,20,31,20,33,19
Data 33,19,36,20,36,20,37,22,37,22,36,26,36,26,34,29,34,29,30,31,30,31
Data 29,39,29,39,27,47,27,47,22,52,22,52,17,54,17,54,9,55,9,55,2,55,2,55
Data -6,54,-6,54,-11,52,-11,52,-16,47,-16,47,-18,39,-18,39,-19,31,-19,31
Data -23,29,-23,29,-24,26,-24,26,-24,22,-24,22,-23,20,-23,20,-22,19,-22,19
Data -20,20,-20,20,-18,23,-18,23,-14,10,-14,10,-6,3,-6,3,0,0
' brows
Data -11,22,-7,20,-7,20,-3,20,-3,20,0,20,0,20,2,22
Data 9,22,11,21,11,21,14,20,14,20,18,20,18,20,22,22
' glasses
' bridge
' left view
Data -18,22,-10,24,-10,24,21,25,21,25,29,22
Data -10,24,-12,27,-12,27,-12,30,-12,30,-11,33,-11,33,-9,34,-9,34,-5,34
Data -5,34,0,30,0,30,1,27,1,27,1,24
' glasses
' right view
Data 10,24,10,27,10,27,11,30,11,30,16,34,16,34,20,34,20,34,22,33,22,33
Data 23,30,23,30,23,27,23,27,21,25
' nose
Data 0,30,0,33,0,33,-1,36,-1,36,-3,38,-3,38,-4,40,-4,40,-1,41,-1,41
Data 4,42,4,42,7,42,7,42,12,41,12,41,15,40,15,40,14,38,14,38,12,36,12,36
Data 11,33,11,33,11,30
' smile mouth
Data -11,40,-5,45,-5,45,2,47,2,47,9,47,9,47,16,45,16,45,22,40
' tooth
Data -2,46,-2,49,-2,49,2,49,2,49,2,49,2,47
'
Procedure Draw_faces
Color 8
Restore Face_data
Repeat
  Read C0%,R0%,C1%,R1%
  Line Cx+C0%/2,Rx+R0%/2,Cx+C1%/2,Rx+R1%/2
Until C0%=2 And R0%=49 And C1%=2 And R1%=47
Plot Cx-7/2,Rx+28/2      ! Place Knowing Norman's eyes
Plot Cx-6/2,Rx+28/2
Plot Cx+17/2,Rx+28/2
Plot Cx+18/2,Rx+28/2
Deffill 9,2,8
Fill Cx+5,32
Deffill 3,2,8
Fill Cx-5/2,Rx+29/2
Fill Cx+19/2,Rx+29/2
Deffill 12,2,8
Fill Cx+2/2,Rx+27/2
Deffill 0,2,8
Fill Cx+28/2,Rx+25/2
Return
'
'
Procedure Smile
Graphmode 2
Restore Dtl
Rx=30
Cx=225
Repeat
  Read C0%,R0%,C1%,R1%
  Line Cx+C0%/2,Rx+R0%/2,Cx+C1%/2,Rx+R1%/2
Until C0%=2 And R0%=49 And C1%=2 And R1%=47
Dtl:
Data -11,40,-5,45,-5,45,2,47,2,47,9,47,9,47,16,45,16,45,22,40
' tooth
Data -2,46,-2,49,-2,49,2,49,2,49,2,49,2,47
Return
'
Procedure Sadness
Restore Dtl
Rx=30
Cx=225
Repeat
  Read C0%,R0%,C1%,R1%
  Line Cx+C0%/2,Rx+R0%/2,Cx+C1%/2,Rx+R1%/2

```




```

Until C0%=16 And R0%=45 And C1%=22 And R1%=47
Dt1:
Data -11,47,-5,45,-5,45,16,45,16,45,22,47
Return
:
: keep the old color pallett
:
Procedure Colr_get
For I%=0 To 15
  @Vq_color(I%)
  Oldcolri(I%)=I%
  Oldcolrr(I%)=R%
  Oldcolrg(I%)=G%
  Oldcolrb(I%)=B%
Next I%
Return
:
Procedure Vq_color(I%)
Dpoke Contr1,26
Dpoke Contr1+2,0
Dpoke Contr1+4,0
Dpoke Contr1+6,2
Dpoke Contr1+8,0
Dpoke Intin,I%
Dpoke Intin+2,1
Vdisys
R%=Dpeek(Intout+2)
G%=Dpeek(Intout+4)
B%=Dpeek(Intout+6)
Return
:
: restore old colors and end
:
Finish:
For I%=0 To 15
  In%=Oldcolri(I%)
  R%=Oldcolrr(I%)
  G%=Oldcolrg(I%)
  B%=Oldcolrb(I%)
  Gosub Vr_color(In%,R%,G%,B%)
Next I%
End
:
Procedure Vr_color(In%,R%,G%,B%)
Dpoke Contr1,14
Dpoke Contr1+2,0
Dpoke Contr1+6,4
Dpoke Intin,In%
Dpoke Intin+2,R%
Dpoke Intin+4,G%
Dpoke Intin+6,B%
Vdisys
Return
:
Procedure Opening
Cls
Reveal=0
Ang=0
Mik=0
If Turn=0 Then
  Vscr=0
  Mscr=0
  Deffill 4,2,19
  Color 4
  Box 0,0,310,199
  Box 40,25,270,170
  Fill 5,5
  Print At(10,6);"Hi, I'm      Norman!"
  CX=150
  RX=30
  @Draw_faces
  Deftext 4,4,0,6
  Text 77,168,"Survey Sweepsteaks"
  Print At(10,10);"Your name, please: ";
  Print At(12,13);Your_name$
  Print At(10,13);
  Form Input 20,New_name$
  If New_name$="" Then
    New_name$=Your_name$
  Else
    Your_name$=New_name$
  Endif
  Print At(10,15);"Reveal Answers? Y/N"
  Leap:

```




```

Print At(30,15);
Form Input 1,Rev$
If Rev$<>"Y" And Rev$<>"y" And Rev$<>"N" And Rev$<>"n" Then
  Print At(30,15);" "
  Goto Leap
Endif
Asr=5
Bsr=10
Sr=Int(Rnd*(Bsr-Asr+1))+Asr
Print At(10,17);"      Rounds: "
Print At(10,18);"      In This Game ";Sr
Pause 100
Endif
Used$=""
Bonus=0
Chance%=0
Chance2%=0
Ritechk%=0
Ritechk2%=0
Col%=2
Row%=10
Coln%=22
Rown%=10
If Turn=0 Then
  Deffill 4,2,8
  Pbox 0,0,310,200
  Deftext 7,4,0,24
  Graphmode 2
  Text 19,22,"Survey Sweepstakes"
  Deftext 6,4,0,24
  Text 20,23,"Survey Sweepstakes"
  CX=225                                ! Place Norman Face
  RX=30
  @Draw_faces
  Deftext 8,0,0,4
  Text 20,55,Your_name$
  Text 170,55,"Knowing"
  Text 245,55,"Norman"
  Deffill 1,2,8                        ! Define two text windows in black
  Color 10
  Rbox 159,59,301,151                ! Outline Norman's box
  Prbox 160,60,300,150
  Color 11
  Rbox 9,59,151,151                  ! Outline your
  Prbox 10,60,150,150
  Deffill 0,2,8
  Deftext 1,0,0,4
  Color 0
  Rbox 9,154,301,186                ! Outline question box in pink
  Prbox 10,155,300,185                ! Define a white window for questions
  Get 0,0,310,199,Screen$
  Deftext 8,0,0,4
  Text 165,150,"Score:"+Str$(Nscr)
  Text 15,150,"Score:"+Str$(Yscr)
  Deftext 1,0,0,8
  Text 45,172,Chr$(158)+"rice "+Chr$(214)+"aterhose,"+Chr$(216)+"nc."
  Deftext 1,1,0,4
  Text 65,181,"Fictionalized Perlustrations"
  Deffill 0,2,8
  Pbox 100,35,190,45
  @In_song
  Deftext 1,1,0,4
  Text 115,42,"Press A Key"
  @Mug_it
  Repeat
    Until Inkey$<>" "
  Else
    Put 0,0,Screen$,3
  Endif
  @Comment
  @Mug_it
  Pause 100
  Deffill 1,2,8
  Prbox 160,60,300,150
  Turn=1
  Deffill 1,2,8
  Prbox 10,140,150,150
  Deftext 0,0,0,4
  Text 15,150,"Score: "+Str$(Yscr)
  Prbox 160,140,300,150
  Text 165,150,"Score: "+Str$(Nscr)
  Deftext 1,0,0,4                    ! Set up and identify the notice boxes
  Deffill 2,2,8
  Prbox 10,188,50,198

```




```

Text 16,195,"RIGHT"
Deffill 3,2,8
Prbox 70,188,110,198
Text 76,195,"WRONG"
Deffill 6,2,8
Prbox 130,188,210,198
Text 148,195,"ILLEGAL"
Deffill 1,2,8
Color 2
Rbox 270,188,290,198
Color 3
Rbox 271,189,289,197
Prbox 272,190,288,196
Deftext 2,0,0,4
Text 225,196,"Timer->"
Deftext 0,0,0,4
Text 277,195," "
Color 2
Rbox 95,32,200,48
Bn=Int(Rnd*5)+1
If Bn=5 Then
    Graphmode 3
    Text 20,45,"BONUS"
    @Bonus_sound
    For Rpt=1 To 6
        Text 20,45,"BONUS"
        Pause 10
    Next Rpt
    Bonus=1
Endif
Graphmode 2
Return

Procedure Place_question
Deffill 0,2,8
Round=Round+1
Pbox 108,35,190,45
Deftext 1,16,0,4
Text 125,42,"ROUND "+Str$(Round)
Deftext 1,0,0,4
Color 15
Rbox 9,154,301,186 ! Outline question box in pink
Prbox 10,155,300,185 ! Define a white window for questions
@Select_data
Text 15,170,Quest0$
Text 15,177,Quest1$
C_t_y=1
Return

Procedure Input_your_answers
C_t_n=0
C_t_y=0
Leave=0
U1:
Graphmode 2
Response$=""
Inc Chance%
Yclr%=7
Col%=2
Lk%=0
Print At(Col%-1,Row%);">"
Print At(Col%+1,Row%);""
Tmc=0
Tm=11
Deftext 0,0,0,4
Deffill 1,2,8
Repeat
    Inc Tmc
    If Tmc=1000 Then
        Sound 1,15,5,7
        Dec Tm
        Tmc=0
        Prbox 272,190,288,196
        If Tm=10 Then
            Text 274,195,Str$(Tm)
        Else
            Text 277,195,Str$(Tm)
        Endif
        If Tm=0 Then
            Graphmode 3
            Sound 1,0
            Setcolor 10,7,7,7
            Text 277,195,Str$(Tm)
            Pause 10
        
```




```

Tm=11
Tmc=0
@Bounce_out
Graphmode 2
'
' If you cannot answer subtract 10 points from your score and
' give to norman. Show capitalization on norman window.
'
Yscr=Yscr-10
Deffill 1,2,8
Prbox 10,140,150,150
Deftext 8,0,0,4
Text 15,150,"Score: "+Str$(Yscr)
Mscr=Mscr+10
Prbox 160,140,300,150
Deftext 8,0,0,4
Text 165,150,"Score: "+Str$(Mscr)+" CAPITALIZED"
@Capitalize_sound
Prbox 160,140,300,150
Deftext 8,0,0,4
Text 165,150,"Score: "+Str$(Mscr)
'
Graphmode 2
C_t_n=1
Endif
If C_t_n=1 Then
  Goto Byebye
Endif
Endif
X$=Inkey$
If Lk%=16 Then
  Goto Out
Endif
Dec Yclr%
Setcolor 10,Yclr%,Yclr%,Yclr%
If Yclr%=0 Then
  Yclr%=7
Endif
If X$<>" " And X$<>Chr$(8) Then ! IF A LEGAL KEY IS PRESSED PRINT IT
  Inc Col%
  Print At(Col%,Row%);X$
  Response$=Response$+X$
  Inc Lk%
Endif
If X$=Chr$(8) Then ! IF THE BACKSPACE KEY IS PRESSED ERASE NAMES$
  For Ers%=3 To Col%
    Print At(Ers%,Row%);" "
  Next Ers%
  Col%=2
  Response$=""
  Lk%=0
Endif
Sound 1,0
Until X$=Chr$(13) ! IF THE RETURN KEY IS PRESSED JUMP OUT
Out:
Sound 1,0
Graphmode 2
Prbox 272,190,288,196 ! FILL IN TIMER WITH BLANK
Rn=-1
Setcolor 10,7,7,7
Response$=Mid$(Response$,1,Len(Response$)-1)
If Mid$(Response$,1,1)>"\" And Mid$(Response$,1,1)<"{" Then
  Convt$=Mid$(Response$,1,1)
  Convt=Asc(Convt$)
  Convt=Convt-32
  Mid$(Response$,1,1)=Chr$(Convt)
Endif
If Mid$(Response$,1,4)="Quit" Or Mid$(Response$,1,4)="quit" Then
  @Winner_screen
  Goto Byebye
Endif
If Mid$(Response$,1,5)="Chall" Or Mid$(Response$,1,5)="chall" And Mik=0 Then
  Mik=1
  Inc Row%
  C_t_n=1
  @Chall_music
  Goto Byebye
Endif
If Mid$(Response$,1,5)="Chall" Or Mid$(Response$,1,5)="chall" And Mik=1 Then
  @Ill_sign
  Goto U1
Endif
If Mid$(Response$,1,4)=Mid$(Chs$(0),1,4) Then
  Rn=0
Endif

```




```

If Mid$(Response$, 1, 4)=Mid$(Chs$(1), 1, 4) Then
  Rn=1
Endif
If Mid$(Response$, 1, 4)=Mid$(Chs$(2), 1, 4) Then
  Rn=2
Endif
If Mid$(Response$, 1, 4)=Mid$(Chs$(3), 1, 4) Then
  Rn=3
Endif
If Mid$(Response$, 1, 4)=Mid$(Chs$(4), 1, 4) Then
  Rn=4
Endif
If Mid$(Response$, 1, 4)=Mid$(Chs$(5), 1, 4) Then
  Rn=5
Endif
If Mid$(Response$, 1, 4)=Mid$(Chs$(6), 1, 4) Then
  Rn=6
Endif
If Mid$(Response$, 1, 4)=Mid$(Chs$(7), 1, 4) Then
  Rn=7
Endif
If Mid$(Response$, 1, 4)=Mid$(Chs$(8), 1, 4) Then
  Rn=8
Endif
If Mid$(Response$, 1, 4)=Mid$(Chs$(9), 1, 4) Then
  Rn=9
Endif
If Rn<0 Or Rn>9 Then
  Goto Jumpover
Endif
'
' It it has been used catch it and offer another chance
'
@Used_check
If Cku=1 Then
  @ill_sign
  Dec Chance%
  Goto U1
Endif
Used$=Used$+Str$(Rn)
Cku=0
'
Jumpover:
@Check_answers
If Right%=1 Then
  @Correct
  Inc Ritechk%
  Inc Row%
  If Bonus=1 Then      ! If bonus round give an extra 10 points
    Vscr=Vscr+20
  Else
    Vscr=Vscr+10
  Endif
  Deffill 1,2,8
  Prbox 10,140,150,150
  Deftext 8,0,0,4
  For Rs=1 To 5
    Graphmode 3
    Text 15,150,"Score: "+Str$(Vscr)
    Pause 10
  Next Rs
  Graphmode 2
  C_t_y=1
Else
  @Incorrect
  Inc Row%
  If Bonus=1 Then
    Nscr=Nscr+20
  Else
    Nscr=Nscr+10
  Endif
  Deffill 1,2,8
  Prbox 160,140,300,150
  Deftext 8,0,0,4
  For Rs=1 To 5
    Graphmode 3
    Text 165,150,"Score: "+Str$(Nscr)
    Pause 10
  Next Rs
  Graphmode 2
  C_t_n=1
Endif
'
' Check to see if your number of chances have been allotted or
' If you have three correct answers to win

```




```

'
If Ritechk%=3 Then
  Leave=1
  Yscr=Yscr+20
  Deffill 1,2,8
  Prbox 10,140,150,150
  Deftext 8,0,0,4
  For Rs=1 To 5
    Graphmode 3
    Text 15,150,"Score: "+Str$(Yscr)
    Pause 10
  Next Rs
  Graphmode 2
  Goto Byebye
Endif
If Chance%=4 Then
  Leave=1
Endif
Byebye:
Graphmode 2
Return
'
Procedure Input_norman_answer
  C_t_n=0
  C_t_y=0
  @Mug_it
  Leave=0
  For Trpt=1 To 4
    Graphmode 3
    Deftext 0,0,0,4
    Text 230,149,"MY TURN!!"
    Pause 20
  Next Trpt
  Inc Chance2%
  Response$=""
  U2:
  Rn=0
  Coln%=22
  Rn=Int(Rnd*15)
  If Rn>9 And Ang=0 Then
    Response$="Challenge"
    Ang=1
    Coln%=21
    Ad%=0
    Print At(Coln%+1,Rown%);"
    For Prtnm=1 To Len(Response$)
      Inc Ad%
      Print At(Coln%+Ad%,Rown%);Mid$(Response$,Prtnm,1)
      Pause Int(Rnd*20)+1
    Next Prtnm
    Inc Rown%
    C_t_y=1
    @Chall_music
    Goto Byebye2
  Endif
  If Rn>9 And Ang=1 Then
    Goto U2
  Endif
  ' Do not allow any repeats of the number
  '
  @Used_check
  If Cku=1 Then
    Goto U2
  Endif
  Used$=Used$+Str$(Rn)
  Cku=0
  '
  Response$=Chr$(Rn)
  Pz=Int(Rnd*12000)+1
  Tmc=0
  Tm=11
  Graphmode 2
  Deftext 0,0,0,4
  Text 230,149,"HMMMMMM?"
  For Pzrpt=1 To Pz
    Tmc=Tmc+1
    If Tmc=1000 Then
      Sound 1,15,5,7
      Tm=Tm-1
      Tmc=0
      Prbox 272,190,288,196
      If Tm=10 Then
        Text 274,195,Str$(Tm)
      Else

```




```

Text 277,195,Str$(Tn)
Endif
If Tm=0
  Graphmode 3
  Text 230,149,"HMMMMMM?"
  Graphmode 2
  Sound 1,0
  For Rptmr=1 To 3
    Graphmode 3
    Text 277,195,Str$(Tn)
    Pause 10
  Next Rptmr
  Graphmode 2
  Tm=10
  Tmc=0
  @Bounce_out
  C_t_y=1
  If C_t_y=1 Then
    Goto 0t
  Endif
Endif
Endif
0t:
Sound 1,0
Exit If C_t_y=1
Next Pzrpt
If C_t_y=1 Then
  Goto Byebye2
Endif
Graphmode 3
Text 230,149,"HMMMMMM?"
Sound 1,0
Coln%=21
Ad%=0
Fka=65
Fkb=122
Print At(Coln%+1,Rown%);""
Stln=Len(Response$)
For Prtnm=1 To Stln
  Inc Ad%
  Print At(Coln%+Ad%,Rown%);Mid$(Response$,Prtnm,1)
  Pause Int(Rnd*40)+1
  Fk=Int(Rnd*(Fkb-Fka+1))+Fka      ! Norman makes typing errors and
  Fkk=Int(Rnd*5)+1                ! corrects them.
  If Fkk=5 And Stln<>Prtnm Then
    Print At(Coln%+Ad%+1,Rown%);Chr$(Fk)
    Pause Int(Rnd*40)+1
  Endif
Next Prtnm
Pause 80

@Check_answers
If Right%=1 Then
  @Correct
  Inc Ritechk2%
  Inc Rown%
  If Bonus=1 Then
    Nscr=Nscr+20
  Else
    Nscr=Nscr+10
  Endif
  Deffill 1,2,8
  Prbox 160,140,300,150
  Deftext 8,0,0,4
  For Rs=1 To 5
    Graphmode 3
    Text 165,150,"Score: "+Str$(Nscr)
    Pause 10
  Next Rs
  Graphmode 2
  C_t_n=1
Else
  @Incorrect
  Inc Rown%
  If Bonus=1 Then
    Yscr=Yscr+20
  Else
    Yscr=Yscr+10
  Endif
  Deffill 1,2,8
  Prbox 10,140,150,150
  Deftext 8,0,0,4
  For Rs=1 To 5
    Graphmode 3
    Text 15,150,"Score: "+Str$(Yscr)

```



```

    Pause 10
    Next Rs
    Graphmode 2
    C_t_y=1
Endif
If Chance2%=>4 Then
    Leave=1
    Goto Byebye2
Endif
If Ritechk2%=3 Then
    Leave=1
    Deffill 1,2,8
    Prbox 160,140,300,150
    Deftext 8,0,0,4
    For Rs=1 To 5
        Graphmode 3
        Text 165,150,"Score: "+Str$(Nscr)
        Pause 10
    Next Rs
    Graphmode 2
Endif
Byebye2:
Graphmode 2
Return

Procedure Correct
    Sound 1,0
    Graphmode 2
    Deftext 1,0,0,4
    Deffill 2,2,8
    For JX=1 To 5
        Sound 1,10,9,5,2
        Prbox 10,188,50,198
        Pause 5
        Text 16,195,"RIGHT"
        Sound 1,10,4,5,15
    Next JX
    Sound 1,0
    Right%=0
Return

Procedure Incorrect
    Sound 1,0
    Wave 1,0
    Graphmode 2
    Deftext 1,0,0,4
    Deffill 3,2,8
    Wave 1,1,14,35
    For JX=1 To 5
        Prbox 70,188,110,198
        Pause 5
        Text 76,195,"WRONG"
        Pause 5
    Next JX
    Wave 1,0
Return

Procedure Check_answers
    Right%=0
    Wrong%=0
    For IX=0 To 4
        If Mid$(Chs$(IX),1,4)=Mid$(Response$,1,4) Then
            Right%=1
        Endif
    Next IX
Return

End_round:
    Sound 1,0
    Wave 1,0
    Cls
    Deffill 1,2,8
    Prbox 0,40,320,170
    Deftext 2,4,0,24
    Text 50,100,"End of Round "+Str$(Round)
    Deftext 3,4,0,6
    If Yscr>Nscr Then
        Text 75,120,"The winner so far is "
        Text 80,135,Your_name$
        Text 80,150,Str$(Yscr)+" points"
    Endif
    If Nscr>Yscr Then
        Text 40,120,"Seems like Norman is winning."
        Text 40,135,Str$(Nscr)+" points"
    Endif

```

```

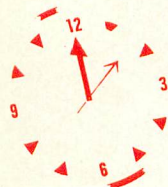
    If Nscr=Yscr Then
        Text 40,120,"Well, we have a tie so far."
        Text 40,135,Str$(Nscr)+" to "+Str$(Yscr)
    Endif
    Wave 1,1,12,300
    Scrn%=7
    For Srpt=1 To 20000
        Dec Scrn%
        If Scrn%=0 Then
            Scrn%=7
        Endif
        Setcolor 0,Scrn%,Scrn%,Scrn%
    Next Srpt
    Setcolor 0,7,7,7
    Wave 1,0
    Bonus=0
    Goto Topgame

Procedure Ill_sign
    Sound 1,0
    Wave 1,0
    Graphmode 2
    Deftext 1,0,0,4
    Wave 1,1,8,100
    For Rpt%=1 To 5
        Deffill 6,2,8
        Prbox 130,188,210,198
        Pause 10
        Text 148,195,"ILLEGAL"
        Pause 10
    Next Rpt%
    Wave 1,0
    Graphmode 1
Return

Procedure Select_data
    Ss%=Int(Rnd*(Vbls-1))
    Ss$=Str$(Ss%)
    If Len(Ss%)<2 Then
        Ss$="0"+Ss$
    Endif
    Add_data=Add_data+1
    If Add_data=Vbls-1 Then
        Storage$=""
        Add_data=0
    Endif
    For Ex%=0 To Len(Storage$)
        If Mid$(Storage$,Ex%,2)=Ss$ Then
            @Select_data
        Endif
    Next Ex%
    Storage$=Storage$+Ss$
    Quest0$=Q0$(Ss%)
    Quest1$=Q1$(Ss%)
    Chs$(0)=I0$(Ss%)
    Chs$(1)=I1$(Ss%)
    Chs$(2)=I2$(Ss%)
    Chs$(3)=I3$(Ss%)
    Chs$(4)=I4$(Ss%)
    Chs$(5)=I5$(Ss%)
    Chs$(6)=I6$(Ss%)
    Chs$(7)=I7$(Ss%)
    Chs$(8)=I8$(Ss%)
    Chs$(9)=I9$(Ss%)
Return

Procedure Lod_file
    Ed=0
    Fileselect "a:\*.swp","Quest3.swp",Aa$
    Cls
    If Exist(Aa$) Then
        Open "I",#1,Aa$
        Input #1,A$
        Vbls=Val(A$)
        Print At(10,10);Aa$
        Print At(10,12);"Loading....";Vbls;" Questions"
        Print At(10,14);"Question #";
        For IX=0 To Vbls-1
            Print At(21,14);IX+1
            Input #1,A$
            Q0$(IX)=A$
            Input #1,A$
            Q1$(IX)=A$
            Input #1,A$
            I0$(IX)=A$

```

```

Input #1,A$
I1$(I%)=A$
Input #1,A$
I2$(I%)=A$
Input #1,A$
I3$(I%)=A$
Input #1,A$
I4$(I%)=A$
Input #1,A$
I5$(I%)=A$
Input #1,A$
I6$(I%)=A$
Input #1,A$
I7$(I%)=A$
Input #1,A$
I8$(I%)=A$
Input #1,A$
I9$(I%)=A$
Next I%
Pause 100
Else
Alert 3," | Missing Question File",1,"END",Ab
Ed=1
Endif
Cls
Return
Procedure Used_check
Cku=0
For Ex2%=1 To Len(Used$)
If Mid$(Used$,Ex2%,1)=Str$(Rn) Then
Cku=1
Endif
Next Ex2%
Return
Procedure Bounce_out
Sound 1,0
Wave 1,0
For Drop=12 Downto 1
Sound 1,10,Drop,4,1
Next Drop
Sound 1,0
For Bounce=1 To 10
For Up=1 To 3
Sound 1,10,Up,4,1
Next Up
For Down=3 Downto 1
Sound 1,10,Down,4,1
Next Down
Next Bounce
Sound 1,10,1,1,10
Sound 1,0
Wave 1,0
Return
Procedure Winner_screen
Graphmode 2
Cls
Sound 1,0
Wave 1,0
Deffill 1,2,8
Prbox 0,40,320,170
Deftext 2,4,0,24
Text 100,100,"WINNER"
Deftext 3,0,0,6
If Nscr>Yscr Then
Tval=Nscr*Int(Rnd*200)+1
Tval$=Str$(Tval)
Text 70,120,"Norman"
Text 70,135,"$"+Tval$+" "+Str$(Nscr)+" points"
Nscr=0
Tval=0
Goto Cloz_up
Endif
If Yscr>Nscr Then
Tval=Yscr*Int(Rnd*200)+1
Tval$=Str$(Tval)
Text 70,120,Your_name$
Text 70,135,"$"+Tval$+" "+Str$(Yscr)+" points"
Yscr=0
Tval=0
Goto Cloz_up
Endif
If Yscr<>0 And Nscr<>0 And Yscr=Nscr Then

```



```

Text 55,120,"Tie Game...No winners here."
Endif
Cloz_up:
Wave 1,1,12,300
Scrn%=7
Text 35,150,"<Y> to play again <N> to end"
Repeat
  Dec Scrn%
  If Scrn%=0 Then
    Scrn%=7
  Endif
  Setcolor 0,Scrn%,Scrn%+2,Scrn%+4
  Ik$=Inkey$
  Pause 5
Until Ik$="Y" Or Ik$="y" Or Ik$="N" Or Ik$="n"
Setcolor 0,7,7,7
Wave 1,0
Return

Procedure Chall_music
Sound 1,0
Sound 2,0
Sound 3,0
Wave 7,0
Restore C_major
For G=1 To 5
  Read Mt1,Mt2,Mt3,Oct
  Sound 1,10,Mt1,5
  Sound 2,10,Mt2,5
  Sound 3,10,Mt3,5
  Wave 7,7,3,15000,10
Next G
C_major:
Data 1,5,8,4
C_minor:
Data 1,8,4,4
C_major_6:
Data 1,5,10,3
C_minor_7:
Data 4,1,10,3
C_7:
Data 1,5,11,4
Return

Procedure In_song
Restore Intro_song
For Sn=1 To 9
  Del=5
  If Sn=3 Or Sn=6 Or Sn=9 Then
    Del=9
  Endif
  Read Xs
  Sound 1,10,Xs,4,Del
Next Sn
Sound 1,10,12,3
Sound 2,10,3,4
Sound 3,10,8,4
Wave 7,7,3,8000,10
Intro_song:
Data 12,10,8
Data 10,8,6
Data 8,6,5
Return

Procedure Comment
Comt=Int(Rnd*10)+1
If Comt=1 Then
  Print At(22,12);" Ok, Pal... "
  Print At(22,13);"Let's gooooo!! "
Endif
If Comt=2 Then
  Print At(22,12);" Ok pushover "
  Print At(22,13);"Get cookin!!!!!"
Endif
If Comt=3 Then
  Print At(22,12);"Make out your "
  Print At(22,13);"will get???"
Endif
If Comt=4 Then
  Print At(22,12);"Might as well "
  Print At(22,13);"quit right now!"
Endif
If Comt=5 Then
  Print At(22,12);"Look out, kid.."
  Print At(22,13);"dis is war!!!"

```

```

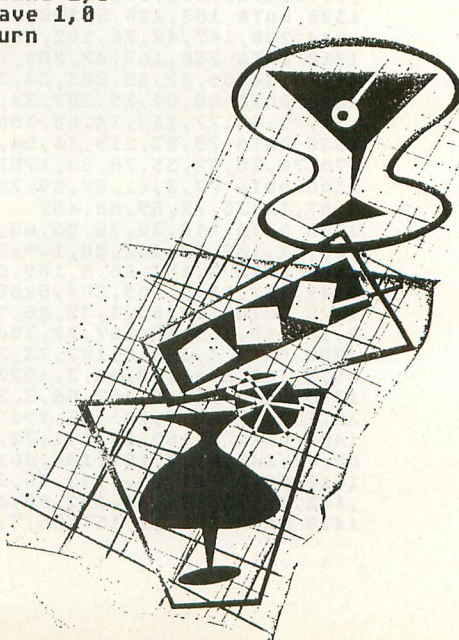
Endif
If Comt=6 Then
  Print At(22,12);"Scared, yet, "
  Print At(22,13);"Pal???"
Endif
If Comt=7 Then
  Print At(22,12);"Get your coffin"
  Print At(22,13);"ready!!"
Endif
If Comt=8 Then
  Print At(22,12);"Your feelin' da"
  Print At(22,13);"heat now...??"
Endif
If Comt=9 Then
  Print At(22,12);"Psssssssst! "
  Print At(22,13);"Booooooooooooo!!.. "
Endif
If Comt=10 Then
  Print At(22,12);"Go back to grade"
  Print At(22,13);"school!!!"
Endif
Return

Procedure Mug_it
Graphmode 2
For Facerpt=1 To 3
  Sound 1,10,12,6,1
  Color 0
  @Smile
  Color 8
  @Sadness
  Pause 2
  Sound 1,10,1,6,1
  Color 0
  @Sadness
  Color 8
  @Smile
  Pause 2
Next Facerpt
Sound 1,0
Return

Procedure Bonus_sound
For Decay=15 To 0 Step -3
  For Ups=1 To 12
    Sound 1,Decay,Ups,5,1
  Next Ups
Next Decay
Return

Procedure Capitalize_sound
For Cap=1 To 100
  For I=1 To 10
    Wave 1,1,1000,I
  Next I
  For I=10 To 1 Step -1
    Wave 1,1,1000,I
  Next I
Next Cap
Sound 1,0
Wave 1,0
Return

```



8

BIT

1000 DATA 255,255,0,32,64,42,216,173,4
8,2,141,66,42,173,49,2,1891
1010 DATA 141,67,42,165,12,141,188,32,
165,13,141,189,32,173,36,2,3393
1020 DATA 141,137,42,173,37,2,141,138,
42,169,32,133,13,169,187,133,6025
1030 DATA 12,169,0,141,136,42,141,82,4
2,141,65,42,160,0,185,26,2797
1040 DATA 3,201,69,240,8,200,200,200,1
92,33,208,242,96,200,169,168,3373
1050 DATA 153,26,3,200,169,42,153,26,3
,160,0,185,0,228,153,168,6462
1060 DATA 42,200,192,16,208,245,169,20
8,141,172,42,169,32,141,173,42,8635
1070 DATA 24,173,4,228,105,1,141,210,3
2,173,5,228,105,0,141,211,7227
1080 DATA 32,173,6,228,24,105,1,141,20
0,32,173,7,228,105,0,141,5070
1090 DATA 201,32,169,42,141,232,2,169,
248,141,231,2,169,144,141,198,1196
1100 DATA 2,162,28,189,247,41,32,193,3
2,32,178,32,32,178,32,202,5085
1110 DATA 16,241,162,255,154,76,0,160,
169,0,133,20,165,20,240,252,8901
1120 DATA 96,32,255,255,76,35,32,142,7
5,42,140,76,42,32,255,255,7461
1130 DATA 174,75,42,172,76,42,96,32,25
5,255,8,192,128,240,31,201,9770
1140 DATA 155,240,60,142,94,42,174,92,
42,224,255,208,8,162,0,142,7762
1150 DATA 136,42,142,92,42,174,94,42,2
38,136,42,76,23,33,169,0,2584
1160 DATA 141,64,42,141,82,42,141,136,
42,141,135,42,169,7,172,137,5700
1170 DATA 42,174,138,42,32,92,228,32,8
1,218,76,134,33,40,96,173,5256
1180 DATA 82,42,208,218,173,64,42,240,
21,169,5,141,134,42,169,1,4812
1190 DATA 141,135,42,169,0,133,194,141
,136,42,169,155,40,96,140,77,6246
1200 DATA 42,142,78,42,162,255,142,92,
42,160,0,140,81,42,172,81,5173
1210 DATA 42,162,0,189,128,5,217,240,3
3,208,77,200,232,189,128,5,9727
1220 DATA 217,240,33,208,67,200,232,18
9,128,5,217,240,33,208,57,200,1384
1230 DATA 185,240,33,141,132,33,200,18
5,240,33,141,133,33,172,136,42,7482
1240 DATA 169,155,153,128,5,216,32,185
,33,76,255,255,160,0,132,195,9817
1250 DATA 140,136,42,169,46,153,128,5,
200,169,155,153,128,5,174,78,7171
1260 DATA 42,172,77,42,169,155,40,96,1
73,81,42,24,105,5,141,81,2831
1270 DATA 42,201,65,208,153,172,136,42
,169,155,153,128,5,208,223,162,1030
1280 DATA 0,189,128,5,201,155,240,3,23
2,16,246,142,167,42,162,0,7548
1290 DATA 189,128,5,201,155,240,30,201
,32,208,20,138,168,185,129,5,7620
1300 DATA 153,128,5,200,204,167,42,208
,244,206,167,42,76,202,33,232,1124
1310 DATA 236,167,42,208,219,96,68,73,
82,49,34,69,82,65,203,34,3564
1320 DATA 80,82,79,207,34,85,78,80,211
,34,78,65,77,215,34,68,3981
1330 DATA 79,83,219,34,84,82,65,235,34
,79,70,70,32,35,78,85,1757
1340 DATA 77,3,41,82,69,78,197,37,68,6
9,67,11,36,72,69,88,452
1350 DATA 117,36,76,86,65,242,36,32,38
,35,32,152,34,162,80,169,3878
1360 DATA 3,157,66,3,169,6,157,74,3,16
9,42,157,69,3,169,204,5260
1370 DATA 157,68,3,32,86,228,16,2,48,6
2,162,80,169,20,157,72,3504
1380 DATA 3,169,0,157,73,3,169,42,157,
69,3,169,184,157,68,3,4073
1390 DATA 169,5,157,66,3,32,86,228,48,
30,162,0,169,9,141,66,2945
1400 DATA 3,169,20,141,72,3,142,73,3,1
69,42,141,69,3,169,184,4610
1410 DATA 141,68,3,32,86,228,16,194,32
,152,34,76,134,33,162,80,4294
1420 DATA 169,12,157,66,3,32,86,228,96
,72,32,152,34,32,38,35,761
1430 DATA 104,162,80,157,66,3,169,0,15
7,73,3,157,72,3,169,132,3942
1440 DATA 157,68,3,169,5,157,69,3,32,8
6,228,152,16,202,76,67,4620
1450 DATA 35,169,33,208,212,169,35,208
,208,169,36,208,204,169,32,208,2017
1460 DATA 200,173,188,32,133,12,173,18
9,32,133,13,32,38,35,76,116,2283
1470 DATA 228,173,65,42,208,45,238,65,
42,173,48,2,141,66,42,24,2334
1480 DATA 105,3,141,22,42,173,49,2,141
,67,42,105,0,141,23,42,361
1490 DATA 169,42,141,49,2,169,16,141,4
8,2,169,6,160,116,162,35,3703
1500 DATA 32,92,228,76,134,33,32,38,35
,76,134,33,173,67,42,141,3118
1510 DATA 49,2,173,66,42,141,48,2,169,
0,141,65,42,174,97,228,5691
1520 DATA 172,96,228,169,6,32,92,228,9
6,72,162,7,189,239,41,32,6314
1530 DATA 193,32,202,16,247,104,133,21
2,169,0,133,213,32,170,217,32,9046
1540 DATA 230,216,32,244,39,32,71,36,3
2,104,35,76,146,34,162,5,1955
1550 DATA 189,128,42,32,193,32,202,16,
247,96,165,66,240,3,76,95,6560
1560 DATA 228,173,48,2,201,16,208,7,17
3,49,2,201,42,240,33,173,6549
1570 DATA 48,2,141,66,42,24,105,3,141,
22,42,173,49,2,141,67,1301
1580 DATA 42,105,0,141,23,42,169,16,14
1,48,2,169,42,141,49,2,1160
1590 DATA 165,138,133,4,165,139,133,5,
160,0,177,4,141,72,42,200,5182
1600 DATA 177,4,141,73,42,169,0,141,68
,42,141,69,42,141,70,42,2505
1610 DATA 141,71,42,248,160,16,14,72,4
2,46,73,42,162,3,189,68,2797
1620 DATA 42,125,68,42,157,68,42,202,2
08,244,136,208,233,216,162,25,2574
1630 DATA 160,0,185,69,42,72,74,74,74,
74,9,16,157,24,42,232,2789
1640 DATA 104,41,15,9,16,157,24,42,232
,200,192,3,208,228,76,95,8225
1650 DATA 228,169,3,133,242,169,5,133,
244,169,128,133,243,32,0,216,33
1660 DATA 144,3,76,210,40,32,210,217,3
2,77,36,169,61,32,193,32,4606
1670 DATA 169,36,32,193,32,162,3,189,2
18,42,32,193,32,202,16,247,7930
1680 DATA 32,71,36,32,81,218,169,0,133
,242,76,134,33,169,155,32,6679
1690 DATA 193,32,96,160,3,162,1,142,94
,42,181,212,72,74,74,5072
1700 DATA 74,170,189,151,42,153,218,42
,136,104,41,15,170,189,151,42,6963
1710 DATA 153,218,42,136,174,94,42,202
,16,221,96,160,3,185,128,5,6272
1720 DATA 201,36,208,1,200,162,0,134,2
12,134,213,142,80,42,185,128,9411
1730 DATA 5,162,15,221,151,42,240,5,20
2,16,248,48,48,142,78,42,5451
1740 DATA 6,212,38,213,6,212,38,213,6,
212,38,213,6,212,38,213,8580
1750 DATA 216,24,165,212,109,78,42,133
,212,165,213,105,0,133,213,200,1146
1760 DATA 238,80,42,173,80,42,201,4,24
0,3,76,136,36,173,80,42,4487
1770 DATA 208,3,76,210,40,32,170,217,3
2,230,216,169,61,32,193,32,8010
1780 DATA 162,0,189,128,5,72,41,127,32
,193,32,232,104,16,243,32,5868
1790 DATA 71,36,32,81,218,76,134,33,16
5,130,133,2,165,131,133,3,5395
1800 DATA 169,127,141,93,42,165,3,197,
133,208,9,165,2,197,132,208,9263
1810 DATA 3,76,134,33,160,0,177,2,72,4
1,127,32,193,32,230,2,3832
1820 DATA 208,2,230,3,104,16,237,32,71
,36,165,136,133,0,165,137,6107
1830 DATA 133,1,169,0,141,91,42,238,93
,42,32,186,37,133,232,32,6097
1840 DATA 186,37,133,233,201,128,144,6
,32,71,36,76,255,36,32,186,5841


```

1850 DATA 37,32,186,37,32,186,37,201,2
,176,9,32,186,37,201,155,6492
1860 DATA 208,249,240,214,32,186,37,20
1,20,240,230,201,27,240,226,201,5124
1870 DATA 22,240,199,201,14,208,10,162
,6,32,186,37,202,208,250,240,2449
1880 DATA 227,201,15,208,6,32,186,37,1
70,208,238,205,93,42,208,212,2203
1890 DATA 165,232,133,212,165,233,133,
213,32,170,217,32,230,216,238,91,4423
1900 DATA 42,173,91,42,201,6,208,8,169
,0,141,91,42,32,71,36,2089
1910 DATA 32,244,39,169,32,32,193,32,2
00,192,6,208,246,76,94,37,7728
1920 DATA 160,0,177,0,230,0,208,2,230,
1,96,32,3,40,32,71,968
1930 DATA 36,162,3,142,91,42,32,14,38,
216,160,1,177,203,201,128,8390
1940 DATA 176,70,165,206,201,128,176,3
0,165,205,24,101,207,133,205,165,2099
1950 DATA 206,101,208,133,206,200,177,
203,24,101,203,133,203,165,204,105,335
1
1960 DATA 0,133,204,76,212,37,162,13,1
89,114,42,32,193,32,202,16,5592
1970 DATA 247,76,134,33,165,136,133,20
3,165,137,133,204,96,173,254,6,1441
1980 DATA 133,205,173,255,6,133,206,96
,32,14,38,32,23,38,160,0,1561
1990 DATA 177,203,133,232,200,177,203,
133,233,201,128,144,3,76,147,38,9606
2000 DATA 200,177,203,141,89,42,200,17
7,203,141,90,42,200,177,203,162,2686
2010 DATA 5,221,83,42,208,3,76,221,38,
202,16,245,201,7,208,3,7589
2020 DATA 76,159,39,201,30,208,3,76,14
3,39,173,90,42,205,89,42,5445
2030 DATA 240,9,168,177,203,141,90,42,
76,70,38,24,173,89,42,101,4158
2040 DATA 203,133,203,165,204,105,0,13
3,204,24,165,205,101,207,133,205,2329
2050 DATA 165,206,101,208,133,206,76,4
0,38,32,14,38,32,23,38,160,1655
2060 DATA 1,177,203,201,128,208,14,162
,18,189,95,42,32,193,32,202,7541
2070 DATA 16,247,76,134,33,136,173,254
,6,145,203,173,255,6,200,145,2100
2080 DATA 203,24,173,254,6,101,207,141
,254,6,173,255,6,101,208,141,1256
2090 DATA 255,6,200,177,203,24,101,203
,133,203,165,204,105,0,133,204,1269
2100 DATA 76,153,38,200,177,203,201,14
,240,3,76,90,39,140,74,42,5373
2110 DATA 152,24,105,7,168,177,203,201
,18,240,8,201,20,240,4,201,9542
2120 DATA 22,208,231,172,74,42,32,181,
39,165,136,133,228,165,137,133,783
2130 DATA 229,173,254,6,133,212,173,25
5,6,133,213,160,1,177,228,201,3453
2140 DATA 128,240,96,197,227,208,26,13
6,177,228,197,226,208,19,32,170,2399
2150 DATA 217,32,205,39,200,177,203,20
1,18,208,3,76,221,38,76,100,7625
2160 DATA 38,160,2,177,228,24,101,228,
133,228,165,229,105,0,133,229,2111
2170 DATA 24,165,212,101,207,133,212,1
65,213,101,208,133,213,76,21,39,9784
2180 DATA 169,86,32,193,32,169,82,32,1
93,32,169,45,32,193,32,165,6087
2190 DATA 205,133,212,165,206,133,213,
32,170,217,32,230,216,32,221,39,1539
2200 DATA 76,100,38,169,78,32,193,32,1
69,70,32,193,32,169,45,32,4313
2210 DATA 193,32,76,105,39,200,177,203
,201,23,208,3,76,221,38,201,9604
2220 DATA 24,240,249,208,240,200,177,2
03,201,27,208,249,200,177,203,201,8260
2230 DATA 14,208,4,136,76,221,38,136,7
6,70,38,162,0,200,177,203,8725
2240 DATA 149,212,232,224,6,208,246,32
,210,217,165,212,133,226,165,213,6856
2250 DATA 133,227,96,172,74,42,162,0,2
00,181,212,145,203,232,224,6,2594
2260 DATA 208,246,96,32,244,39,169,32,
32,193,32,174,91,42,202,16,5984

```

```

2270 DATA 5,32,71,36,162,3,142,91,42,9
6,160,255,200,177,243,72,1279
2280 DATA 41,127,32,193,32,104,16,244,
96,32,16,40,160,3,185,128,5076
2290 DATA 5,201,155,208,88,96,169,10,1
41,254,6,133,205,133,207,169,2080
2300 DATA 0,141,255,6,133,208,133,206,
96,32,16,40,32,14,38,160,3445
2310 DATA 1,177,203,16,1,96,133,1,136,
177,203,133,0,32,92,41,4165
2320 DATA 160,1,177,203,16,240,165,0,2
4,101,207,141,254,6,133,205,356
2330 DATA 165,1,105,0,141,255,6,133,20
6,96,216,173,254,6,56,229,1309
2340 DATA 207,141,254,6,173,255,6,229,
208,141,255,6,96,162,0,185,519
2350 DATA 128,5,201,44,240,16,201,155,
240,12,157,210,42,232,224,6,987
2360 DATA 240,84,200,76,105,40,157,210
,42,140,79,42,169,0,133,242,8431
2370 DATA 133,205,32,199,40,32,0,216,3
2,210,217,176,57,165,213,48,9826
2380 DATA 53,141,255,6,133,206,165,212
,141,254,6,133,205,172,79,42,657
2390 DATA 185,128,5,201,155,208,1,96,1
62,0,200,185,128,5,201,155,9555
2400 DATA 240,39,157,210,42,232,224,6,
240,12,76,180,40,169,210,133,687
2410 DATA 243,169,42,133,244,96,104,10
4,162,11,189,139,42,32,193,32,6721
2420 DATA 202,16,247,32,81,218,76,134,
33,157,210,42,169,0,133,242,9585
2430 DATA 32,199,40,32,0,216,32,210,21
7,176,219,165,212,133,207,165,4773
2440 DATA 213,133,208,48,209,32,81,218
,96,32,35,40,169,0,141,64,4668
2450 DATA 42,32,14,38,160,0,177,203,13
3,226,200,177,203,133,227,48,2868
2460 DATA 27,205,255,6,240,8,176,20,32
,92,41,76,14,41,165,226,5744
2470 DATA 205,254,6,240,4,176,5,144,23
9,76,134,33,32,84,40,169,6101
2480 DATA 7,160,132,162,41,32,92,228,3
2,81,218,169,10,141,134,42,7026
2490 DATA 169,0,133,194,141,136,42,169
,1,141,135,42,141,64,42,76,4984
2500 DATA 134,33,132,230,160,2,177,203
,216,24,101,203,133,203,165,204,3732
2510 DATA 105,0,133,204,164,230,96,32,
92,41,132,230,160,0,177,203,391
2520 DATA 133,226,200,177,203,133,227,
164,230,96,173,135,42,240,99,165,3711
2530 DATA 194,240,95,173,134,42,240,6,
206,134,42,76,236,41,169,0,7576
2540 DATA 141,135,42,173,254,6,24,101,
207,141,254,6,133,212,173,255,3562
2550 DATA 6,101,208,141,255,6,133,213,
165,213,48,26,197,227,240,8,1814
2560 DATA 144,31,32,113,41,76,178,41,1
65,212,197,226,240,8,144,17,9697
2570 DATA 32,113,41,76,178,41,238,82,4
2,169,12,141,252,2,76,236,8825
2580 DATA 41,32,170,217,32,230,216,32,
244,39,169,32,32,193,32,6310
2590 DATA 81,218,76,98,228,32,35,32,82
,79,82,82,69,155,101,103,5204
2600 DATA 97,107,99,97,80,32,115,101,1
05,116,105,108,105,116,85,32,5133
2610 DATA 67,73,83,65,66,125,112,112,6
6,24,42,1,0,0,0,0,7400
2620 DATA 0,0,0,0,0,0,0,37,56,37,35,53
,52,41,41,46,7412
2630 DATA 39,0,44,41,46,37,26,0,0,0,0,
0,0,0,0,3599
2640 DATA 0,0,0,0,0,0,0,83,42,88,42,10
,11,12,13,35,6210
2650 DATA 4,95,42,133,42,155,100,101,1
14,101,98,109,117,110,101,114,6972
2660 DATA 32,109,97,114,103,111,114,80
,155,56,54,55,50,51,32,61,2305
2670 DATA 62,32,101,110,105,76,155,89,
68,65,69,82,139,42,166,42,4879
2680 DATA 155,114,111,114,114,101,32,1
01,117,108,97,86,48,49,50,51,3168
2690 DATA 52,53,54,55,56,57,65,66,67,6

```



```
8,69,70,204,42,209,42,4764
2700 DATA 68,58,42,46,42,155,247,42,24
7,42,0,226,2,227,2,0,4988
2710 DATA 32,0,0,0,0,0,0,0,0,0,0,0
,0,0,2742
```

Listing 2: Assembly

```

0100 ;SAVE#D:BUPP1.M65
0110 ;
0120 ;-----;
0130 ; BUP.SYS part I ;
0140 ; ;
0150 ; (c) 1989 ;
0160 ; by Barry Kolbe ;
0170 ; ;
0180 ;-----;
0190 .OPT NO LIST
0200 ;
0210 ;equates
0220 ;
0230 LO = $00
0240 JO = $02
0250 EOL = $9B ;end of line
0260 LINENO = $E8 ;line number
0270 LIMPTR = $CB ;ptr to line #
0280 CLINE = $CD ;current line
0290 ILINE = $CF ;increment line
0300 FLINE = $06FE ;first line
0310 RLINE = $E2 ;referenced line
0320 SPTR = $E4 ;ptr. for search
0330 SLINE = $E6 ;search line
0340 VMPTR = $82 ;var. name table
0350 STMTAB = $88 ;BA5IC statmt.
0360 PROMPT = $C2 ;flag-BA5IC done
0370 CRITIC = $42 ;critical I/O
0380 ICCOM = $0342 ;command
0390 ICBAL = $0344 ;buff. address
0400 ICBALH = $0345
0410 ICBLL = $0348 ;buff. length
0420 ICBLH = $0349
0430 ICAUX1 = $034A ;aux. byte
0440 HATBAS = $031A ;handler table
0450 LBUF = $0580 ;input buffer
0460 MEMLO = $02E7 ;memlo
0470 D05INI = $0C ;disk init.
0480 DLISTL = $0230 ;display list
0490 DLISTH = $0231
0500 CIO = $E456 ;cio vector
0510 FASC = $D8E6 ;fp to ascii
0520 AFP = $D800 ;ascii to fp
0530 IFP = $D9AA ;integer to fp
0540 FPI = $D9D2 ;fp to integer
0550 FR0 = $D4 ;fp register 0
0560 FR1 = $D5
0570 SETLBF = $DA51 ;reset lbuf
0580 SETVBV = $E45C ;set vert. blank
0590 SYSVBV = $E45F ;system vbi
0600 XITVBV = $E462 ;exit vbi
0610 INBUF = $F3 ;input buf. ptr
0620 CIX = $F2 ;char. index
0630 WARMST = $E474 ;warnstart
0640 VVBLKD = $0224 ;deferred reg
0650 D05VEC = $0A
0660 ;
0670 * = $2000
0680 ;
0690 IN1 CLD
0700 LDA DLISTL ;save orig
0710 STA DL5 ;display list
0720 LDA DLISTH ;pointer
0730 STA DLH
0740 LDA D05INI ;save
0750 STA SYSRST+1 ;D05INI
0760 LDA D05INI+1
0770 STA SYSRST+2
0780 ;
0790 ;save immed.vbi vectors
0800 ;
0810 LDA VVBLKD
0820 STA VBLKD5
0830 LDA VVBLKD+1
0840 STA VBLKD5+1
0850 ;
0860 ;reset my SYSTEM RESET routine
0870 ;
0880 IN2 LDA # >SYSRST
0890 STA D05INI+1

```

```

0900      LDA # <SYSRST
0910      STA DOSINI
0920      ;
0930      ;reset character counter
0940      ;
0950      LDA #0
0960      STA CHRCNT
0970      STA QTFLAG
0980      STA TFLG      ;trace flag
0990      ;
1000     ;find editor vectors
1010     ;search handler for E:
1020     ;
1030     LDY #0
1040     E1  LDA HATBAS,Y
1050     CMP #'E
1060     BEQ E2      ;got.it
1070     INY
1080     INY
1090     INY
1100     CPY #33
1110     BNE E1
1120     RTS
1130     ;
1140     E2  INY      ;point to
1150     LDA # <MYEDIT ;my patch
1160     STA HATBAS,Y
1170     INY
1180     LDA # >MYEDIT
1190     STA HATBAS,Y
1200     LDY #0      ;copy system
1210     E3  LDA $E400,Y ;routines
1220     STA MYEDIT,Y ;to my
1230     INY      ;patch
1240     CPY #16
1250     BNE E3
1260     ;
1270     ;get byte vector
1280     ;
1290     LDA # <GETCOM-1 ;point it
1300     STA MYEDIT+4 ;to my
1310     LDA # >GETCOM-1 ;wedge
1320     STA MYEDIT+5
1330     CLC      ;put system
1340     LDA $E404 ;get byte
1350     ADC #1    ;at beginning
1360     STA GETCOM+1
1370     LDA $E405
1380     ADC #0
1390     STA GETCOM+2
1400     ;
1410     ;put byte vector
1420     ;
1430     LDA $E406 ;get vector
1440     CLC      ;vectors are
1450     ADC #1    ;always -1
1460     STA PUT1+1 ;add 1
1470     LDA $E407 ;install in
1480     ADC #0    ;my 'PUTC'
1490     STA PUT1+2 ;routine
1500     LDA # >ENDPRG ;set LOMEM
1510     STA MEMLO+1
1520     LDA # <ENDPRG
1530     STA MEMLO
1540     ;
1550     LDA #$90  ;chang color
1560     STA 710
1570     LDX #28   ;say we are
1580     B1  LDA BB1,X ;here!
1590     JSR PUTC
1600     JSR WAIT  ;wait a jiff
1610     JSR WAIT  ;maybe two
1620     DEX
1630     BPL B1
1640     LDX #$FFF ;clear stack
1650     TXS
1660     JMP $A000 ;jump to BASIC
1670     ;
1680     WAIT LDA #0      ;wait a jiffy
1690     STA 20
1700     WIT  LDA 20
1710     BEQ WIT
1720     RTS
1730     ;
1740     ;handle SYSTEM RESET button
1750     ;
1760     SYSRST JSR $FFFF ;does DOSINI
1770     JMP IN2      ;reset

```



```

1780 ;
1790 ;print Accumulator via put byte
1800 ;
1810 PUTC STX XSAV ;save X, Y
1820 STY PYSAV
1830 PUT1 JSR $FFFF ;put byte
1840 LDX XSAV ;restore X,Y
1850 LDY PYSAV
1860 RTS
1870 ;
1880 ;get bytes from editor
1890 ;
1900 GETCOM JSR $FFFF ;get a byte
1910 PHP ;save status
1920 CPY #580 ;break key
1930 BEQ BRK ;yes
1940 CMP #EOL ;RETURN?
1950 BEQ GOTRT1 ;yup
1960 STX X3 ;save X
1970 LDX LSTKEY ;last key
1980 CPX #5FF ;ret? =ff
1990 BNE KOK
2000 LDX #0 ;reset countr
2010 STX CHRCNT
2020 STX LSTKEY
2030 KOK LDX X3 ;ge* X
2040 INC CHRCNT
2050 JMP NOBRK ;skip over
2060 ;
2070 BRK LDA #0 ;on break turn
2080 STA ONOFF ;everything
2090 STA QTFLAG ;OFF!
2100 STA CHRCNT
2110 STA ENDLIN
2120 LDA #7 ;restore vectors
2130 LDY VBLKDS ;for VBI
2140 LDX VBLKDS+1
2150 JSR SETVBU
2160 JSR SETLBF ;floating point
2170 JMP EXIT2 ;pointers
2180 ;
2190 NOBRK PLP ;restore status
2200 RTS ;done
2210 ;
2220 ;BASIC done with line
2230 ;
2240 ;PROMPT <> 0 when BASIC
2250 ;is done parsing a line
2260 ;endlin =0 means line
2270 ;not completely entered
2280 ;from BASIC
2290 ;
2300 GOTRT1 LDA QTFLAG
2310 BNE BRK
2320 LDA ONOFF ;autonum?
2330 BEQ NOTON ;0= off
2340 LDA #5 ;VBI timer
2350 STA TIMER
2360 LDA #1
2370 STA ENDLIN
2380 LDA #0
2390 STA PROMPT ;clear for
2400 STA CHRCNT ;BASIC
2410 LDA #EOL
2420 PLP
2430 RTS
2440 ;
2450 ;autonum not on
2460 ;
2470 NOTON STY Y1 ;save regs
2480 STX X1
2490 LDX #5FF ;last key
2500 STX LSTKEY
2510 ;
2520 ;check command
2530 ;
2540 LDY #0
2550 STY CY1
2560 GTCM1 LDY CY1
2570 LDX #0
2580 LDA LBUF,X ;first char
2590 CMP COMTAB,Y ;in table?
2600 BNE NOMAT
2610 INY
2620 INX
2630 LDA LBUF,X ;try second
2640 CMP COMTAB,Y ;char
2650 BNE NOMAT
2660 INY
2670 INX
2680 LDA LBUF,X ;try third
2690 CMP COMTAB,Y
2700 BNE NOMAT
2710 ;
2720 ;got match
2730 ;
2740 INY
2750 LDA COMTAB,Y
2760 STA TAKEOF+1 ;find addr
2770 INY ;of my
2780 LDA COMTAB,Y ;routine
2790 STA TAKEOF+2
2800 LDY CHRCNT ;stick in
2810 LDA #EOL ;a RETURN
2820 STA LBUF,Y
2830 CLD
2840 JSR CRUNCH ;smash spaces
2850 TAKEOF JMP $FFFF ;go to it
2860 ;
2870 ;exit routines
2880 ;
2890 EXIT2 LDY #0 ;show BASIC
2900 STY $C3 ;no error
2910 STY CHRCNT
2920 LDA #. ;REM
2930 STA LBUF,Y
2940 INY
2950 LDA #EOL ;and EOL
2960 STA LBUF,Y
2970 EXIT1 LDX X1 ;restore X,Y
2980 LDY Y1 ;make a
2990 LDA #EOL ;normal exit
3000 PLP
3010 RTS
3020 ;
3030 ;no match
3040 ;
3050 NOMAT LDA CY1 ;try next
3060 CLC ;command
3070 ADC #5
3080 STA CY1
3090 CMP #541 ;13 commands
3100 BNE GTCM1 ;@ 5 bytes
3110 LDY CHRCNT ;each
3120 LDA #EOL ;put in RETURN
3130 STA LBUF,Y
3140 BNE EXIT1 ;and exit
3150 ;
3160 ;crunch out spaces in input
3170 ;
3180 CRUNCH LDX #0 ;find the
3190 CR1 LDA LBUF,X ;length
3200 CMP #EOL
3210 BEQ GOTEND ;found it
3220 INX
3230 BPL CR1
3240 GOTEND STX ENDPTR ;save end
3250 LDX #0 ;now collapse
3260 SQUISH LDA LBUF,X ;it and
3270 CMP #EOL ;remove spaces
3280 BEQ CR2
3290 CMP #520 ;space?
3300 BNE NOSQSH ;no squish
3310 DOSQSH TXA
3320 TAY ;copy from
3330 SQSHLP LDA LBUF+1,Y ;back
3340 STA LBUF,Y ;to front
3350 INY ;next one
3360 CPY ENDPTR ;done yet?
3370 BNE SQSHLP ;no
3380 DEC ENDPTR ;len=len-1
3390 JMP SQUISH ;more
3400 NOSQSH INX ;done yet
3410 CPY ENDPTR
3420 BNE SQUISH ;no
3430 CR2 RTS ;finished
3440 ;
3450 ;command table
3460 ;
3470 COMTAB .BYTE "DIR"
3480 .WORD DIR
3490 .BYTE "ERA"
3500 .WORD ERA
3510 .BYTE "PRO"
3520 .WORD LOCK
3530 .BYTE "UNP"
3540 .WORD UNLOCK
3550 .BYTE "NAM"
3560 .WORD RENAM
3570 .BYTE "DOS"
3580 .WORD DOS
3590 .BYTE "TRA"
3600 .WORD TRACE
3610 .BYTE "OFF"
3620 .WORD TOFF
3630 .BYTE "NUM"
3640 .WORD AUTNUM
3650 .BYTE "REN"
3660 .WORD RENUM
3670 .BYTE "DEC"
3680 .WORD DECHEX
3690 .BYTE "HEX"
3700 .WORD HEXDEC
3710 .BYTE "LVA"
3720 .WORD VARLST
3730 ;
3740 ;directory routine
3750 ;
3760 DIR JSR TURNOF ;trace off
3770 JSR CLOSE5 ;close iocb 5
3780 LDX #550
3790 LDA #3 ;open chan #5
3800 STA ICCOM,X
3810 LDA #6 ;directory
3820 STA ICAUX1,X
3830 LDA # >DIRNAM ;D1:*.
3840 STA ICBAL,X
3850 LDA # <DIRNAM
3860 STA ICBAL,X
3870 JSR CIO ;openning
3880 BPL NOERR1 ;no error
3890 BMI ENDIR ;opps
3900 ;
3910 NOERR1 LDX #550 ;chan 5
3920 LDA #20 ;20 chars
3930 STA ICBLL,X
3940 LDA #0 ;for input
3950 STA ICBLL,X
3960 LDA # >DIRBUF ;my buffer
3970 STA ICBAL,X
3980 LDA # <DIRBUF
3990 STA ICBAL,X
4000 LDA #5 ;get bytes
4010 STA ICCOM,X
4020 JSR CIO ;get 'em
4030 BMI ENDIR ;error-quit
4040 ;
4050 LDX #500
4060 LDA #9 ;put bytes
4070 STA ICCOM
4080 LDA #20 ;20 of 'em
4090 STA ICBLL
4100 STX ICBLL
4110 LDA # >DIRBUF ;my buffer
4120 STA ICBAL
4130 LDA # <DIRBUF
4140 STA ICBAL
4150 JSR CIO ;print 'em
4160 BPL NOERR1 ;ok
4170 ENDIR JSR CLOSE5 ;close #5
4180 JMP EXIT2 ;to BASIC
4190 ;
4200 ;close iocb #5
4210 ;
4220 CLOSE5 LDX #550
4230 LDA #500
4240 STA ICCOM,X
4250 JSR CIO
4260 RTS
4270 ;
4280 ;do XIOs using CIO
4290 ;
4300 DOXIO PHA ;save command
4310 JSR CLOSE5 ;close chan
4320 JSR TURNOF ;trace off
4330 PLA ;get XIO
4340 LDX #550 ;command
4350 STA ICCOM,X
4360 LDA #0 ;zero length
4370 STA ICBLL,X
4380 STA ICBLL,X
4390 LDA # <LBUF+4 ;D:filespec
4400 STA ICBAL,X
4410 LDA # >LBUF+4
4420 STA ICBAL,X
4430 JSR CIO ;do XIO
4440 TYA
4450 BPL ENDIR
4460 JMP ERROR
4470 ;
4480 ;XIO jmp table
4490 ;
4500 ERA LDA #33 ;erase file
4510 BNE DOXIO
4520 LOCK LDA #35 ;lock a file
4530 BNE DOXIO
4540 UNLOCK LDA #36 ;unlock it
4550 BNE DOXIO
4560 RENAM LDA #32 ;rename it
4570 BNE DOXIO
4580 DOS LDA SYSRST+1 ;resstore
4590 STA DOSINI ;DOSINI
4600 LDA SYSRST+2 ;vector
4610 STA DOSINI+1
4620 JSR TURNOF ;trace off
4630 JMP WARMST ;do warm start
4640 ;
4650 ;turn trace on
4660 ;special display list
4670 ;from ANALOG issue 31
4680 ;June 1985 page 74
4690 ;
4700 TRACE LDA TFLG ;check flag
4710 BNE AON ;1 = on now
4720 INC TFLG ;set to 1
4730 LDA DLISTL ;get display
4740 STA DLS ;list and
4750 CLC ;save it
4760 ADC #3
4770 STA BACK ;put it into

```



```

4780 LDA DLISTH ;my display
4790 STA DLH ;list
4800 ADC #0
4810 STA BACK+1
4820 LDA # >NDLIST ;install
4830 STA DLISTH ;the new
4840 LDA # <NDLIST ;special
4850 STA DLISTL ;display
4860 LDA #6 ;list
4870 LDY # <VBI ;turn on
4880 LDX # >VBI ;VBI
4890 JSR SETUBV
4900 AOM JMP EXIT2 ;out
4910 ;
4920 ;turn off VBI
4930 ;
4940 TOFF JSR TURNOF ;turn NUM off
4950 JMP EXIT2 ;goto BASIC
4960 ;
4970 TURNOF LDA DLH ;restore
4980 STA DLISTH ;original
4990 LDA DL5 ;display list
5000 STA DLISTL
5010 LDA #0 ;set flag
5020 STA TFLG
5030 LDX SYSUBV+2 ;reset VBI
5040 LDY SYSUBV+1 ;to system
5050 LDA #6 ;vectors
5060 JSR SETUBV
5070 RTS
5080 ;
5090 ;print error message
5100 ;
5110 ERROR PHA ;save error #
5120 LDX #7
5130 E11 LDA ERME5,X ;get byte
5140 JSR PUTC ;print it
5150 DEX
5160 BPL E11
5170 PLA ;get error
5180 STA FR0
5190 LDA #0
5200 STA FR1
5210 JSR IFP
5220 JSR FASC ;convert to
;ATA5CII
5230 JSR MPRINT
5240 JSR DOCR
5250 JSR PRRDY
5260 JMP ENDIR
5270 ;
5280 ;print "READY"
5290 ;
5300 PRRDY LDX #5
5310 RDY LDA READ,X
5320 JSR PUTC
5330 DEX
5340 BPL RDY
5350 RTS
5360 ;
5370 ;vert blank code for trace
5380 ;
5390 VBI LDA CRITIC ;critical
5400 BEQ V3 ;section? no
5410 JMP SYSUBV ;out
5420 V3 LDA DLISTL ;see if special
5430 CMP # <NDLIST ;display list
5440 BNE PUTDL ;is present
5450 LDA DLISTH ;if not - form
5460 CMP # >NDLIST ;it
5470 BEQ V9 ;yes
5480 PUTDL LDA DLISTL ; save it
5490 STA DL5
5500 CLC
5510 ADC #3 ;install in
5520 STA BACK ;new display
5530 LDA DLISTH ;list
5540 STA DLH
5550 ADC #0
5560 STA BACK+1
5570 LDA # <NDLIST
5580 STA DLISTL
5590 LDA # >NDLIST
5600 STA DLISTH
5610 V9 LDA $0A ;get executing
5620 STA $0A ;line pointer
5630 LDA $0B
5640 STA $05
5650 LDY #0 ;get executing
5660 LDA ($04),Y ;line no.
5670 STA BFR0
5680 INY
5690 LDA ($04),Y
5700 STA BFR1
5710 LDA #0 ;clear out BCD
5720 STA BCD ;work space
5730 STA BCD+1
5740 STA BCD+2
5750 STA BCD+3
5760 SED ;decimal mode
5770 LDY #510 ;16 bits
5780 V4 ASL BFR0 ;2 BCD digits
5790 ROL BFR1 ;per byte
5800 LDX #503 ;6 BCD digits
5810 V5 LDA BCD,X ;total
5820 ADC BCD,X
5830 STA BCD,X
5840 DEX
5850 BNE V5
5860 DEY
5870 BNE V4
5880 CLD ;clear decimal
5890 LDX #25 ;put digits on
5900 LDY #0 ;top line
5910 V6 LDA BCD+1,Y
5920 PHA ;high nibble
5930 LSR A
5940 LSR A
5950 LSR A
5960 LSR A
5970 ORA #510 ;internal code
5980 STA TLINE,X
5990 INX
6000 PLA ;get byte
6010 AND #50F ;low nibble
6020 ORA #510 ;internal code
6030 STA TLINE,X
6040 INX
6050 INY
6060 CPY #3 ;done?
6070 BNE V6 ;no
6080 JMP SYSUBV ;gone
6090 ;
6100 ;convert dec to hex
6110 ;
6120 DECEX LDA #3 ;point to #
6130 STA CIX ;in 'DECddd'
6140 LDA # >LBUF
6150 STA INBUF+1 ;point FP
6160 LDA # <LBUF ;to it
6170 STA INBUF
6180 JSR AFP ;to FP
6190 BCC DD2 ;ok
6200 JMP VALERR ;error
6210 DD2 JSR FPI ;to integer
6220 JSR BNT0HX ;to hex digits
6230 LDA #'=
6240 JSR PUTC ;'='
6250 LDA #'$
6260 JSR PUTC ;'$'
6270 LDX #3 ;print the
6280 DD1 LDA N2BUF,X ;hex digits
6290 JSR PUTC
6300 DEX
6310 BPL DD1
6320 JSR DOCR
6330 JSR SETLBF ;reset
6340 LDA #0 ;FP stuff
6350 STA CIX
6360 JMP EXIT2 ;to BASIC
6370 ;
6380 ;do a carriage return
6390 ;
6400 DOCR LDA #EOL
6410 JSR PUTC
6420 RTS
6430 ;
6440 ;binary to hex converter
6450 ;
6460 BNT0HX LDY #3 ;work backwards
6470 LDX #1 ;2 hex numbers
6480 HX1 STX X3
6490 LDA FR0,X ;get high
6500 PHA ;save it
6510 LSR A ;move to
6520 LSR A ;low nibble
6530 LSR A
6540 LSR A
6550 ;
6560 TAX
6570 LDA HEXTAB,X ;get hex digit
6580 STA N2BUF,Y ;store it
6590 DEY ;move pointer
6600 PLA ;down
6610 AND #50F ;now low nibble
6620 TAX
6630 LDA HEXTAB,X ;get hex
6640 STA N2BUF,Y ;save it
6650 DEY
6660 LDX X3 ;next hex numbr
6670 DEX
6680 BPL HX1
6690 RTS ;done
6700 ;
6710 ;hex to dec converter
6720 ;
6730 HEXDEC LDY #3 ;point to
6740 LDA LBUF,Y ;hex in
6750 CMP #'$
6760 BNE HD1 ;skip '$'
6770 INY
6780 HD1 LDX #0 ;clear work
6790 STX FR0 ;space
6800 STX FR0+1
6810 STX X2 ;digit counter
6820 PU1 LDA LBUF,Y ;get digit
6830 LDX #50F ;in table
6840 H2 CMP HEXTAB,X
6850 BEQ H3 ;X = 0-15
6860 DEX
6870 BPL H2
6880 BMI H6 ;oh! oh!
6890 H3 STX X1 ;= 0 (-) 15
6900 ;
6910 ;convert to binary
6920 ;
6930 ASL FR0 ;times 16
6940 ROL FR0+1
6950 ASL FR0
6960 ROL FR0+1
6970 ASL FR0
6980 ROL FR0+1
6990 ASL FR0
7000 ROL FR0+1
7010 CLD
7020 CLC
7030 LDA FR0 ;add in value
7040 ADC X1 ;0(-) 15
7050 STA FR0
7060 LDA FR0+1
7070 ADC #0
7080 STA FR0+1
7090 INY ;next dec digit
7100 INC X2 ;hex counter
7110 LDA X2
7120 CMP #4 ;got 4
7130 BEQ H6 ;yes
7140 JMP PU1 ;do more
7150 H6 LDA X2 ;# digits
7160 BNE H7
7170 JMP VALERR ;error
7180 ;
7190 ;convert to ATASCII $ print
7200 ;
7210 H7 JSR IFP
7220 JSR FASC ;to ATASCII
7230 LDA #'=
7240 JSR PUTC
7250 LDX #0
7260 H9 LDA LBUF,X
7270 PHA ;save it
7280 AND #57F ;mask inverse
7290 JSR PUTC ;table
7300 INX
7310 PLA ;get it
7320 BPL H9 ;more
7330 JSR DOCR
7340 JSR SETLBF ;reset LBUF
7350 JMP EXIT2 ;to BASIC
7360 ;
7370 ;do listing of variables
7380 ;
7390 VARLST LDA VNT ;point to
7400 STA J0 ;variable
7410 LDA VNT+1 ;table
7420 STA J0+1
7430 LDA #57F ;first var=$80
7440 STA MYTOKEN
7450 L5BEG LDA J0+1
7460 CMP $85 ;at end of
7470 BNE LPVAR ;var table?
7480 LDA J0
7490 CMP $84
7500 BNE LPVAR
7510 JMP EXIT2 ;to BASIC
7520 ;
7530 LPVAR LDY #500 ; print variable
7540 LDA (J0),Y
7550 PHA ;save it
7560 AND #57F ;mask inverse
7570 JSR PUTC ;to screen
7580 INC J0 ;move ptr up
7590 BNE LV1
7600 INC J0+1
7610 LV1 PLA ;get byte
7620 BPL LPVAR ;more if +
7630 JSR DOCR ;do CR
7640 LDA STMTAB ;get start of
7650 STA L0 ;BASIC
7660 LDA STMTAB+1 ;lines
7670 STA L0+1
7680 LDA #0 ;column cntr
7690 STA PRFLAG
7700 INC MYTOKEN ;next variable
7710 ;

```

(CONTINUED ON PAGE 117)

The Compukid Connection

ST

O T A N N E N B A U M !

The holidays are approaching, and parents will be considering gifts to buy for their children. If your children enjoy your ST, a computer-related gift may be just the thing.

If it's software you're looking for, you will be pleased to know that truly excellent programs can be had cheaply or at no cost. Last winter I discovered several arcade packages on the market. These "best hits" are groups of three or more of the older arcade games (six months or more past the release date); they are sold as one unit for the price of a single brand-new release. Some of the packages contain duds, but more typically, all of the programs in a package are high-quality arcades. Two such packages were a big hit with our three older boys last December.

At nominal cost or at no cost at all, you can obtain public-domain or freely distributed programs of high quality. [ST-LOG programs are not in the public domain; even though we allow limited distribution, we retain all rights—ed.] Some of these can be customized to create personal software gifts. For example, last December I noticed in the documentation for *Monkeys and Balloons* by Frank Cohen (from the October 1988 issue of ST-LOG) that he suggested customizing his program by altering the *DEGAS Elite* .PC1 files. I took a fresh look at the program, a version of *Breakout* in which the mouse controls two monkeys holding a trampoline. A clown bounces off the trampoline in an attempt to pop balloons overhead. I altered the *DEGAS* pictures so that the clown resembled my then-five-year-old daughter, Catherine, complete with eyeglasses and a letter "C" on her sweatshirt, and the monkeys were made to look like her brothers, Willy and Joey. The final product was copied to a green disk and a red label was attached. She loved it!

That same year I downloaded a program from GENie called *HANGOPUS* by B. Cafferky for my son, Willy, a *Bloom County* fan. This version of Hangman lets you create word-puzzle files. I made files for Willy that included vocabulary from his hobbies. The resulting program was copied to a special red disk. There are many other public-domain programs that include the possibility of customization. And there are hundreds of other challenging public-domain games for children; your online service, user's group library or public-domain dealer will have many fine offerings.

The holidays are also a good time to con-

A Holiday Tree-Trimming Kit by D.A. Brumleve



sider a new set of joysticks; certainly, ours have a life expectancy of less than a year. Another accessory a child really appreciates is a joystick extender. These are short cords with a male joystick plug at one end and a female plug at the other. They are inserted into the joystick and mouse ports of your computer; when the child must remove joysticks or the mouse, the device is unplugged from the extender and not from the computer's ports. Extenders also make changing joysticks on a 1040 much more convenient. Though not quite as exciting as a new game, mouse pads are appreciated by children also.

Tannenbaum

The Germanic tribes in ancient times had a long-standing reverence for the evergreen tree (Tannenbaum). These trees demonstrated that life could continue even in the "dead" of winter. To celebrate this sign of life, they brought the trees into their homes in the winter. When these people became Christians, they retained their pagan custom. As the evergreen ritual coincided with the celebration of Christmas, it eventually became part of that celebration: the Christmas tree.

My mother always enjoyed decorating for the holiday season, and the focal point of her endeavors was the Christmas tree. We brought home a huge tree every year and stood it up in our front hallway, where it was circled by the main stairway. We children dropped tinsel and decorations onto its top branches from the balcony.

This holiday season, you and your child can enjoy the fun of decorating the tree without a trip to the attic to gather box upon

box of bulbs and bells. *Tannenbaum*, on this month's ST-LOG disk, is a music-and-design program with a holiday theme. It is intended for use by children as young as two or three. The child (or adult!) uses the mouse to choose from a variety of animated and nonanimated decorations to decorate a Christmas tree. The animation of the decorations (twinkling bulbs and more) can then be set into motion while the computer plays the traditional German holiday song, "O Tannenbaum!"

Getting Started

To use the program, deARC the file TANNENBM.ARC and copy TANNENBM.PRg, TANNENBM.PI1, TANNENSP.MBK and TANNENMU.MBK to a freshly formatted disk. (The other file in the ARCD file, TANNENBM.BAS, is the STOS BASIC source code for the program; you will need it only if you want to examine the program's construction with the STOS interpreter.) The program will work if placed within a folder, so long as all four needed files are present in that folder. The program will run on any ST with a color monitor. To run the program from your ST's desktop, be sure that your screen is in low resolution and then double-click on TANNENBM.PRg in the directory listing. The title screen will appear, followed automatically by the main program screen.

The Main Screen

The main screen is divided into three areas: the Options menu, the Christmas tree display and the decorations boxes.

The decorations boxes contain two kinds of decorations. The bottom row of boxes

(two boxes of tinsel and a yellow globe) are framed by a gold box. These are nonanimated decorations. You may place any number of these decorations on the tree. The upper three rows of boxes hold animated decorations. Only 14 total animated decorations can be placed on the tree. These 14 may be all of one type of decoration, or a combination of many.

The Options menu includes four boxes: The EXIT option gives the user an opportunity to quit the program and return to the desktop. The PLAY option starts the movement of the animated decorations and the accompanying music. Clicking the mouse on the PLAY option will cause the computer to play the tune whether any decorations are on the tree or not. The NEW option erases any and all decorations currently placed on the tree. The UNDO option erases only the decoration that was last selected.

The Christmas tree display area, of course, contains the tree. Decorations can be placed almost anywhere within this area; it is not necessary to put them directly on the tree. Some of the larger decorations can be placed so that they nearly touch the rectangle surrounding the tree display, but others must be placed a bit farther inside the perimeter.

Using the Program

To choose a decoration, point the mouse arrow at any of the items in the decorations boxes and click either or both mouse buttons. The mouse arrow will disappear, and in its place will be a copy of the decoration you have chosen. Move the mouse to the tree display and position the decoration wherever you choose. Then press either or both mouse buttons again to put the decoration in place. If the position is not precisely what you'd like, click UNDO and repeat the procedure.

When your tree is trimmed to your satisfaction, click PLAY and watch the decorations move while the music plays. You can speed up or slow down the tempo by pressing any of the function keys on the top row of your ST's keyboard. The original tempo of the music as presented can be restored by pressing the F4 key. After the song is over, the mouse will reappear, and you can make other selections.

Special Effects

The function keys do not affect the speed of the animations. To achieve the fastest animation of a decoration, use only one single decoration; the computer will animate one decoration much faster than it will 15.

Children may enjoy exploring symmetry on this rather symmetrical representation of a tree. A pyramid of angels, a flock of birds and numerous combinations can demonstrate such balance.

Don't neglect the black areas of the tree display. Twinkling stars are just as charming when hanging from your display ceiling as they are on the tree.

Certain combinations of decorations or their unusual placement can yield surprising results and expand the ability of the program to mimic an actual Christmas tree. For example, the tinsel supplied in the decorations boxes can imitate candlesticks or true icicles, depending on where it is displayed. I chose to draw rather globby chunks of tinsel because that is how my children apply tinsel to a real tree. When combined with one of the small globes or the twinkling star, however, the tinsel becomes a convincing candlestick.

Some decorations are especially pleasing when placed so that they overlap others of the same or different type. The effect of such displays, when animated, can be quite striking. When two decorations overlap, STOS gives priority to the decoration that is lower down on the screen. When you choose a decoration and move it around in the display area, you will notice that it sometimes seems to be on top of other decorations and sometimes behind (or underneath) them. This is quite effective in duplicating the layered look of a Christmas tree. When animated, the priority of the decorations relative to one another is preserved; if a bird is behind a pinwheel when placed on the tree display, it will continue to be behind the pinwheel while both are animated.

The nonanimated decorations in the bottom row of the decorations boxes have the lowest priority. A nonanimated decoration will always be behind the animated decorations. The only exception occurs when the nonanimated decoration is the last decoration selected; in this case, it will have the same priority as the animated decorations. To see how this works, place an animated decoration on the tree. Then select a decoration from the bottom row and place it over the first one so that the nonanimated decoration is "on top." Now select another decoration of any kind. As soon as you select another decoration, the nonanimated decoration moves behind the first one.

Each type of animated decoration has its own animation sequence. When the PLAY option is selected, the first animated decoration the computer changes is the one you placed on the tree first. The next one to

change is the one you placed second, and so on. The order of animation (and thus the order in which you have placed the decorations on the tree) can be an important influence on the effectiveness of your display.

To get started in experimenting with these effects, try placing pinwheels in a row. Place the first pinwheel near the upper left-hand corner of the tree display. Then place another pinwheel just a bit to the right of it, so that it overlaps the blades on the right side of the first pinwheel. Add more pinwheels to this row and then click the mouse on PLAY. Many other types of decorations can produce such interesting effects when used alone or in combination with other types.

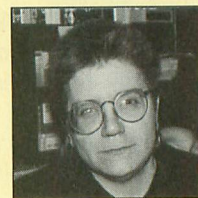
Sing Along!

Many children, at least when prodded, enjoy singing along with music programs. Below, I provide an English-language version of the song for those of you who may not be familiar with it. (Note that this version, by John W. Schaum, is not a direct translation of the German, though it does effectively convey the spirit.)

*O Christmas Tree! O Christmas Tree!
Your lovely branches thrill me.
O Christmas Tree! O Christmas Tree!
You fill my heart with glee.
Your ornaments of red and blue,
Your pretty lights of many hues,
O Christmas Tree! O Christmas Tree!
Your lovely branches thrill me.*

Before closing, I want to thank David Plotkin for his valuable assistance in the development of this program. This was my first effort in STOS, and Mr. Plotkin's contribution of time and wisdom is much appreciated.

I hope you and your children will enjoy Tannenbaum. Happy holidays!



D.A. Brumleve, M.A., is involved with children and computers in a variety of ways. The mother of five children, ages two to ten, she serves as the adult facilitator of the Children's ST Users' Group in Urbana, Illinois. An avid programmer, she has developed a beginner's course in GFA BASIC and is the author of PreSchool KidProgs (MichTron) and numerous freely distributed programs for young ST users. Her daughter, Catherine, is a student at Creative Discovery School.

DISK LISTING

8

BIT

ST

THE NOVEMBER 1989 ANALOG COMPUTING
8-BIT DISK CONTAINS 10 MAGAZINE FILES.
THEY ARE LISTED BELOW:

SIDE 1:

FILENAME.EXT	LANG.	LOAD	ARTICLE NAME
BUPSYS .OBJ	ML	(#3)	BASIC UTILITIES
BUPP1 .M65	MAC/65	LOAD	BASIC UTIL. SOURCE
BUPP2 .M65	MAC/65	LOAD	BASIC UTIL. SOURCE
MEMMATCH.BAS	BASIC	LOAD	MEMORY MATCH
TITLMAKR.OBJ	ML	(#3)	TITLE MAKER
TITLMAK2.BAS	BASIC	LOAD	TITLE MAKER, L2
TITLMAKR.M65	MAC/65	LOAD	TITLE MAKER SOURCE
ADDRESS .BAS	BASIC	LOAD	BASIC TRAINING
MLEDITOR.BAS	BASIC	LOAD	M/L EDITOR
EDITORII.LST	BASIC	ENTER	BASIC EDITOR II

TO LOAD YOUR ANALOG DISK

- 1) INSERT BASIC CARTRIDGE (NOT REQUIRED FOR XE OR XL COMPUTERS).
- 2) TURN ON DISK DRIVE AND MONITOR.
- 3) INSERT DISK IN DRIVE.
- 4) TURN ON COMPUTER. (XL AND XE OWNERS: DO NOT HOLD DOWN OPTION KEY!)

WARNING: BEFORE YOU RUN A PROGRAM, READ THE APPROPRIATE ARTICLE IN THE MAGAZINE. FAILURE TO DO SO MAY YIELD CONFUSING RESULTS.

NOTE: ONLY PROGRAMS WITH THE .BAS, .COM OR .OBJ EXTENSION MAY BE RUN FROM THE MENU. OTHER PROGRAMS SHOULD BE LOADED AS INSTRUCTED IN THE LOADING NOTES AND MAY REQUIRE ADDITIONAL SOFTWARE AS LISTED BELOW. HOWEVER, YOU SHOULD NOT ASSUME THAT EVERY FILE WITH THE PROPER FILE EXTENSION WILL RUN FROM THE MENU. YOU MAY HAVE TO MOVE CERTAIN PROGRAMS TO A DIFFERENT DISK TO OBTAIN CORRECT RESULTS.

EXT DESCRIPTION

.M65	REQUIRES THE MAC/65 ASSEMBLER
.AMA	REQUIRES THE ATARI MACRO ASSEMBLER
.ASM	REQUIRES THE ATARI ASSEMBLER/EDITOR
.ACT	REQUIRES THE ACTION! CARTRIDGE
.LGO	REQUIRES THE ATARI LOGO CARTRIDGE
.SYN	REQUIRES THE SYNAPSE SYN ASSEMBLER

LOADING NOTES

LOAD BASIC PROGRAM:	LOAD "D:FILENAME.EXT"
ENTER BASIC PROGRAM:	ENTER "D:FILENAME.EXT"
LOAD MAC/65 PROGRAM:	LOAD #D:FILENAME.EXT
ENTER ASM/ED PROGRAM:	ENTER #D:FILENAME.EXT
LOAD LOGO PROGRAM:	LOAD "D:FILENAME.EXT"
LOAD SYN/AS PROGRAM:	LOAD "D:FILENAME.EXT"

- #1: SEE ACTION! MANUAL.
- #2: SEE ATARI MACRO ASSEMBLER MANUAL.
- #3: MAY ALSO BE LOADED FROM DOS USING THE "L" OPTION OF THE DOS MENU.
- #4: THIS FILE SHOULD BE TRANSFERRED TO ANOTHER DISK AND RENAMED "AUTORUN.SYS".
- #5: READ THE APPROPRIATE ARTICLE FOR INSTRUCTIONS ON USING THIS FILE.

The NOVEMBER 1989 ANALOG COMPUTING ST disk contains 20 magazine files. They are listed below.

FILENAME.EXT	FILE TYPE	COMMENTS
CMANSHIP.ARC contains:		
C78 .C	C	C-MANSHIP
COMB078 .PRG	RUN FILE	THE PROGRAM SO FAR
MICROCHK.RSC	RESOURCE	

ASSEMBLY.ARC contains:		
TEMPLON.PRG	RUN FILE	TEMPLEMON
TEMPLON.ENG	TEXT	INSTRUCTIONS
READ .ME	TEXT	INSTRUCTIONS

COMPUKID.ARC contains:		
TANNENBM.PRG	RUN FILE	D, TANNENBAUM
TANNENBM.PII	DEGAS	PICTURE FILE
TANNENMU.MBK	DATA	
TANNENSP.MBK	DATA	
TANNENBM.BAS	BASIC	STOS SOURCE

PDPARADE.ARC contains:		
FLU .PRG	RUN FILE	FLU PROGRAM
FLU .RSC	RESOURCE	
FLU .DOC	TEXT	INSTRUCTIONS
VKILLER .PRG	RUN FILE	VIRUS KILLER
VKILLER .RSC	RESOURCE	
VKILLER .DOC	TEXT	INSTRUCTIONS

SURVEY.ARC contains:		
SWEEP .PRG	RUN FILE	SURVEY SWEEPSTAKES
QUEST .SWP	DATA	QUESTION FILE
SWEEP .LST	GFA BASIC 2.0	SOURCE CODE

ARCX .TTP	RUN FILE	UNARCHIVING PROGRAM
README .DOC	TEXT	Disk instructions
UNARCHIV.DOC	TEXT	Unarchiving instructions

Disk instructions:

Only those files with .PRG, .TOS, or .TTP extensions may be run from the GEM desktop. Other programs may require additional software as shown below. The files on this disk have been archived (compressed) into .ARC files. To restore the programs to their runnable form, follow the instructions found in the UNARCHIV.DOC file.

WARNING: Be sure to read the appropriate magazine articles before attempting to run the programs on this disk. Failure to do so may yield confusing results.

EXT DESCRIPTION

.BAS	Requires GFA BASIC
.LST	Requires GFA BASIC
.C	Requires C compiler
.S	Requires 68000 assembler
.PAS	Requires Pascal compiler

M/L EDITOR

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

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

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

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

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

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

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

M/L Editor will display, at the top of the screen, the number of the line you're currently working on. As you go through the line, you'll be prompted for each entry. Simply type the number and press Re-

turn. If you press Return without a number, the default is the last value entered.

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

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

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

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

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

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

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

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

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

LISTING 1: BASIC LISTING

```
AZ 10 DIM BF(16),NS(4),AS(1),BS(1),FS(15)
LF 11 DIM MODS(4)
BN 20 LINE=1000:RETRN=155:BACKSP=126:CHK5
UM=0:EDIT=0
GO 30 GOSUB 450:POSITION 10,6:?"Start or
@continue? ":"GOSUB 500:?" CHR$(A)
ZG 40 POSITION 10,8:?"FILENAME":INPUT F
$:POKE 752,1:?" "
FE 50 IF LEN(F$)<3 THEN POSITION 20,10:?"
":GOTO 40
NF 60 IF FS(1,2)<0:"D:" THEN F1$="D":F1$C
33=FS:GOTO 80
KL 70 F1$=FS
TN 80 IF CHR$(A)="" THEN 120
FD 90 TRAP 430:OPEN H2,4,0,F1$:TRAP 110
HQ 100 FOR X=1 TO 16:GET H2,A:NEXT X:LINE
=LINE+10:GOTO 100
HM 110 CLOSE H2:OPEN H2,9,0,F1$:GOTO 170
UT 120 TRAP 160:OPEN H2,4,0,F1$:GOSUB 440
:POSITION 10,10:?"FILE ALREADY EXISTS
!":POKE 752,0
ZU 130 POSITION 10,12:?"ERASE IT? ":"GOS
UB 500:POKE 752,1:?" CHR$(A)
UH 140 IF CHR$(A)="" OR CHR$(A)="" THEN
CLOSE H2:GOTO 30
QG 150 IF CHR$(A)<>"Y" AND CHR$(A)<>"Y" T
HEN 130
BH 160 CLOSE H2:OPEN H2,0,0,F1$
ZU 170 GOSUB 450:POSITION 10,1:?"NOW ON
[LINE]: ":"LINE=CHKSUM=0
GN 180 L1=3:FOR X=1 TO 16:POSITION 13*(X
(1)+12*(X-1)),X+2:POKE 752,0:?" BYTE M"
":X:":"GOSUB 310
KH 190 IF EDIT AND L=0 THEN BYTE=BF(X):GO
TO 210
FY 200 BYTE=VAL(NS)
OZ 201 MODS=NS
BU 210 POSITION 22,X+2:?" BYTE:" "
YZ 220 BF(X)=BYTE:CHKSUM=CHKSUM+BYTE*X:IF
CHKSUM%9999 THEN CHKSUM=CHKSUM-10000
MS 230 NEXT X:CHKSUM=CHKSUM*LINE:IF CHK5
UM%9999 THEN CHKSUM=CHKSUM-10000
ZG 240 POSITION 12,X+2:POKE 752,0:?"CHEC
KSUM: ":"L1=4:GOSUB 310
EW 250 IF EDIT AND L=0 THEN 270
QM 260 C=VAL(NS)
SY 270 POSITION 22,X+2:?" C:" "
ZL 280 IF C=CHKSUM THEN 300
DT 290 GOSUB 440:EDIT=1:CHKSUM=0:GOTO 180
LW 300 FOR X=1 TO 16:PUT H2,BF(X):NEXT X:
LINE=LINE+10:EDIT=0:GOTO 170
FV 310 L=0
KZ 320 GOSUB 500:IF (A=ASC("Q")) OR A=ASC(
"q")) AND K=1 AND NOT EDIT THEN 420
PD 330 IF A<0:RETRN AND A<0:BACKSP AND A<4
0 OR A>57 THEN 320
DK 331 IF A=RETRN AND NS="" THEN NS=MOD5
TD 335 IF A=RETRN AND L=0 AND X>1 THEN 35
0
JR 340 IF ((A=RETRN AND NOT EDIT) OR A=B
ACKSP) AND L=0 THEN 320
DH 350 IF A=RETRN THEN POKE 752,1:?" ":R
ETURN
GG 360 IF A<0:BACKSP THEN 400
SA 370 IF L>1 THEN NS=NS(1,L-1):GOTO 390
AS 380 NS=""
RE 390 ?" CHR$(BACKSP):":L=L-1:GOTO 320
BB 400 L=L+1:IF L>1 THEN A=RETRN:GOTO 35
0
MK 410 NS(L)=CHR$(A):?" CHR$(A):":GOTO 320
KN 420 GRAPHICS 0:END
VT 430 GOSUB 440:POSITION 10,10:?"NO SUC
H FILE!":FOR X=1 TO 1000:NEXT X:CLOSE
H2:GOTO 30
FD 440 POKE 710,48:SOUND 0,100,12,8:FOR X
=1 TO 50:NEXT X:SOUND 0,0,0,0:RETRN
MV 450 GRAPHICS 23:POKE 16,112:POKE 53774
,112:POKE 559,0:POKE 710,4
KR 460 DL=PEEK(560)+256*PEEK(561)+4:POKE
DL-1,70:POKE DL+2,6
HH 470 FOR X=3 TO 39 STEP 2:POKE DL+X,0
:NEXT X
ZH 480 POKE DL+41,65:POKE DL+42,PEEK(560)
:POKE DL+43,PEEK(561):POKE 87,0
AC 490 POSITION 2,0:?"Analog M1 editor":
POKE 559,34:RETRN
MZ 500 OPEN H1,4,0,"K1":GET H1,A:CLOSE H1
:RETRN
```


American Techna-Vision

For Orders Only - 1-800-551-9995

CA. Orders / Info. - 415-352-3787

1050 MECHANISM

Factory fresh TANDON mechs. make difficult repairs a snap. Units are complete with Head, Stepper, Spindle motor, belt etc. Just plug in, no difficult alignments or adjustments required.

\$47.50

POWER PACKS

Exact replacement transformer for 800/400, 1050 810, 1200XL, 850, and XF551 units. Part #CO17945.

\$14.50

XL/XE SUPPLY

Power Pak for 800XL, 600XL, 130XE, 65XE & XE Game.

\$25.00

VISICALC SPREADSHEET

Unleash the computing power of your 8 bit Atari with Visicalc. Compute everything from home finances to high powered financial projections. Hundreds of uses.

\$19.95
DISK

THE BOOKKEEPER AND CX85 KEYPAD

You get both Atari's 8 bit professional bookkeeping system and the handy CX85 numeric keypad for one low price. Packages Factory sealed.

\$19.95
4 DISK SET

ST SPECIALS

ST SOFTWARE THIS BOX ONLY!

- Ninja Mission - Arcade Game\$9.95
- 221-B Baker St.\$14.50
- 221-B Baker Case Library 1\$10.00
- 221-B Baker Case Library 2\$10.00
- Time & Magic - Adventure\$14.50
- Cosmic Relief - Arcade Game\$14.50

KEYBOARDS

New and complete subassembly. Easy internal replacement.

- 130XE/65XE \$35.00
- 800\$40.00
- 800XL\$29.50
- 400\$12.50

800 4 PIECE BOARD SET

Includes Main Board, Power Supply Assembly, CPU Module and 10K Revision B Operating System Module. All boards are new, tested and complete with all components.\$28.50

800/400 MODULES

- NEW PARTS COMPLETE WITH IC'S
- CX853 16K RAM Module \$9.50
- 800/400 CPU with GTIA . \$8.75
- 800 10K "B" OS MODULE \$8.75
- 800 POWER BOARD\$14.50
- 400 POWER BOARD\$9.50

INTEGRATED CIRCUITS

- CO14795 ..\$4.50
- CO14805 ..\$4.50
- CO12296 ..\$4.50
- CO10745 ..\$4.50
- CO14377 ..\$4.50
- CO60472 ..\$4.50
- 6532 ..\$4.50
- CO12399B \$4.50
- CO12499B \$4.50
- CO14599B \$4.50
- CO14806 ..\$4.50
- CO10750 ..\$4.50
- CO10444 ..\$8.50
- CO60302 \$9.50
- CO21697 \$15.00
- CO61991 \$15.00
- CO61598 \$20.00
- CO61618 \$24.00
- CO24947 \$15.00

1050 IC'S

- U7 - 6532 \$4.50
- U8 - 6810 \$4.50
- U9 - 6507 \$4.50
- U10 ROM \$19.50
- U13 2793 \$19.50
- U5 LM2917 \$8.50
- 5713 \$5.25

All other 1050 IC's \$3.75 each
Specify by Socket#

850 INTERFACE

12 Piece Chip set. Includes all plug in IC's except ROM. Replacement fixes vast majority of 850 failures\$19.50

REPAIR MANUALS

SAMS Service Manuals for the following units contain schematics, parts listings, labelled photographs showing the location of checkpoints and more! A special section gives oscilloscope and logic probe readings allowing you to narrow the malfunction down to a specific chip or transistor!
800, 800XL, 130XE, 400, 1025 and 1050\$19.50 each
520ST Service Manual. \$37.50

40 PIN LSI CHIP SET
Complete set of 40 Pin Large Scale Integrated Circuits for 800XL, 600XL, 1200XL. \$19.95

MISCELLANEOUS

- 13 Pin Serial I/O Cable\$4.50
- 600XL 64K Upgrade Kit\$29.95
- ACE Joystick\$7.95
- 1050 Track Zero Sensor\$8.50
- 2793 1050 Controller IC\$19.50
- U.S. Doubler\$29.95
- 400 Board Set (W/O Kybd) \$22.95
- Paddle Controllers (Pair)\$9.95
- Rev. "C" Basic ROM\$15.00
- Rambo XL w/o RAM IC's \$39.95
- 850 or PR Modem Cable ..\$14.50
- 850 or PR Printer Cable ..\$14.50
- Printer Interface\$43.95
- I/O 13 Pin PC connector ..\$4.50
- I/O 13 Pin Cable end plug ..\$4.50
- 1027 INK ROLLER\$6.50
- 1030 Modem Power Pack ..\$6.95

CARTRIDGES

For XL's/XE's 800/400

- POLE POSITION Cartridge\$15.00
- STAR WARS - Return of the Jedi\$15.00
- PAC-MAN Cartridge\$4.00
- DELUXE INVADERS Cartridge\$4.00
- JOURNEY TO THE PLANETS Cart. \$4.00
- GOLF Cart. 800/400 Only\$4.00
- DONKEY KONG Cartridge\$5.00
- STAR RAIDERS Cartridge\$5.00
- MILLIPEDE Cartridge\$10.00
- PLATTER MANIA Cart. by Epyx\$9.95
- WIZARD OF WOR Cartridge\$9.95
- CAVERNS OF MARS Cartridge\$14.50
- PENGO Cartridge\$15.00
- EASTERN FRONT Cartridge\$15.00
- ATARIWRITER Cartridge\$29.95
- TURMOIL Cartridge\$5.00
- CROSSFIRE Cartridge\$7.50
- SPRINGER Cartridge\$7.50
- ADVENTURE CREATOR Cartridge \$12.50
- ZONE RANGER Cartridge\$12.50
- SILICON WARRIOR Cartridge\$12.50
- MATH ENCOUNTER Cartridge\$12.50
- BASIC LANGUAGE Cartridge\$15.00
- PILOT LANGUAGE PACKAGE\$17.50
- SPARTANOS-X Cartridge\$59.95
- ACTION O.S.S. Cartridge\$59.95
- ACTION TOOL KIT DISK\$25.00
- MAC-65 O.S.S. Cartridge\$59.95
- MAC-65 TOOL KIT DISK\$25.00
- R-TIME 8 Cartridge\$56.95

HOT DISK TITLES

BEST SELLERS / LOW PRICES

- GHOSTBUSTERS\$9.95
- F-15 STRIKE EAGLE\$14.95
- TOP GUNNER COLLECTION\$14.95
- SILENT SERVICE\$14.95
- KENNEDY APPROACH\$14.95
- VIDEO TITLE SHOP (64K)\$12.50
- CHESSMASTER 2000\$12.50
- ALTERNATE REALITY THE CITY\$12.50
- ALTERNATE REALITY DUNGEON\$12.50
- TOMAHAWK (64K)\$12.50
- 221-B BAKER STREET (64K)\$12.50
- MUSIC STUDIO\$12.50
- INFOCOM ADVENTURES\$14.95
- WISHBRINGER\$14.95
- MOONMIST\$14.95
- SPELLBREAKER\$14.95
- SEA STALKER\$14.95
- HOLLYWOOD HINX\$14.95
- STATION FALL\$14.95
- INFIDEL\$14.95
- LURKING HORROR\$14.95
- CUT THROATS\$14.95
- BALLYHOO\$14.95
- SUSPENDED\$14.95
- WITNESS\$14.95
- HITCH HIKERS GUIDE\$14.95
- ZORK 1 & ZORK 2\$14.95 EA.

COMPUTER BOOKS

- ATARI PLAYGROUND WORKBOOK \$5.95
- HACKERBOOK TRICKS & TIPS\$5.00
- INSIDE ATARI BASIC\$5.00
- ATARI BASIC REF. MANUAL\$5.00
- HOW TO ATARI 6502 PROGRAM\$5.00
- WRITE YOUR OWN GAMES\$5.00
- PROGRAMMERS REF. GUIDE\$14.95
- ASSEMBLY LANGUAGE GUIDE\$21.95
- XE USERS HANDBOOK\$21.95
- XL USERS HANDBOOK\$21.95
- ATARI BASIC FASTER & BETTER \$22.95
- YOUR ATARI COMPUTER\$17.95

SERIAL I/O CABLE

- High Quality, 13Pin\$4.50
- Special ... 3 Cables for \$10.00

MAC-65 CARTRIDGE
6502 Machine language Macro-Assembler. First class tool for serious programmers... \$59.95

PR: CONNECTION
Serial/Parallel Interface for connecting modems and printers.\$65.00

SUPRA MODEM

Supra 300 Baud Direct Connect Modem requires no separate interface. Includes cable and Disk Software. Supports X-Modem, Download & Upload.\$29.95

Disks-XL's/XE's/800

- F-15 STRIKE EAGLE Disk\$14.95
- TOP GUNNER COLLECTION\$14.95
- SILENTSERVICE Disk\$14.95
- KENNEDY APPROACH Disk\$14.95
- NATO COMMANDER Disk\$14.95
- S.A.T. MATH\$17.95
- S.A.T. VERBAL\$17.95
- VISICALC Spreadsheet\$19.95
- BOOKKEEPER & CX85 KEYPAD\$19.95
- SPIDER MAN Disk\$4.95
- DAVID'S MIDNIGHT MAGIC Disk \$4.95
- ALIEN AMBUSH Disk\$4.95
- REAR GUARD Disk\$4.95
- TECHNA-COLOR DREAM Disk\$4.95
- FREAKY FACTORY Disk\$4.95
- LASER HAWK Disk\$4.95
- ROCKET REPAIRMAN Disk\$4.95
- HUMAN TORCH & THE THING\$5.00
- MUSICAL PILOT Ed. Disk\$5.00
- CONPUTATION Disk\$5.00
- MASTER CHESS Disk\$5.00
- SPEED KING Disk\$5.00
- LAST V-B Disk\$5.00
- PATHFINDER Disk\$5.00
- MATCH RACER Disk\$5.00
- WOMBATS I Adventure Disk\$5.00
- FAMILY FINANCE Disk by Atari ..\$6.50
- FORT APOCALYPSE Disk\$7.50
- NINJA Disk\$7.50
- ELECTRA-GLIDE Disk\$7.50
- SPORTS SPECTACULAR Disk\$7.50
- THE GAMBLER Disk\$7.50
- STRATOS Disk\$7.50
- MIND MAZES (Educational)\$7.50
- TIMEMISE\$7.50
- VIDEO VEGAS Poker & Slots\$9.95
- ACTION BIKER\$9.95
- KICK START\$9.95
- MONEY TOOLS (Financial Utility) ..\$9.95
- DIG DUG Disk\$9.95
- REPTON Disk\$9.95
- PUZZLE PANIC Disk by Epyx\$9.95
- CESTE LA VE Disk\$9.95
- WOMPER STOMPER Disk\$9.95
- OLIN EMERALD (Jr., Adv.)\$9.95
- MOVIE MAKER (Graphics)\$9.95
- FINANCIAL COOKBOOK\$9.95
- CYCLE KNIGHT\$9.95
- SONG WRITER\$9.95
- COUNTING & ADDING with Tink ..\$9.95
- CHESSMASTER 2000\$12.50
- BRIDGE 4.0\$12.50
- PEGAMMON\$12.50
- MUSIC STUDIO\$12.50
- HOTEL ALIEN\$12.50
- FOOLBILSKY Disk / Board Game\$12.50
- GHOST CHASER\$12.50
- L.A. SWAT\$12.50
- GRANDMA'S HOUSE (Educational) \$12.50
- DROPZONE Disk\$12.50
- CASTLE WOLFENSTEIN Disk\$12.50
- BEYOND CASTLE WOLFENSTEIN\$12.50
- TEMPLE OF APHAI\$12.50
- ALI-ANTS Disk\$12.50
- MOUSE QUEST Disk (64K)\$12.50
- ELECTRONIC DRUMMER Disk\$12.50
- STOCK MARKET Game\$12.50
- PIRATES OF THE BARBARY COAST\$12.50
- NICKERBOCKER Disk\$12.50
- MIDWAY BATTLES War Game\$12.50
- BOULDER DASH Construction Set ..\$14.95
- MIND SHADOW Adventure 64K\$14.95
- HACKER Adventure\$14.95
- SEA STALKER Adventure\$14.95
- CONQUEST IN VIETNAM\$14.95
- DECISION IN THE DESERT\$14.95
- CRUSADE IN EUROPE\$14.95
- SPITFIRE ACE\$14.95
- SSI 50 MISSION CRUSH\$14.95
- SSI BATTALION COMMANDER\$14.95
- SSI PANZER GRENADIER\$14.95
- SSI FIELD OF FIRE\$14.95
- SSI GEMSTONE WARRIOR\$14.95
- BANKSTYPER (Typing Tutor)\$15.00
- MASK STREET WRITER\$15.00
- CUT & PASTE W.P. Disk\$17.50
- SPARTANOS CONST. SET\$39.50
- SPARTANOS Tool-Kit\$32.95
- LIK DATA PERFECT Database\$CALL

DATASOFT DISKS

- VIDEO TITLE SHOP\$12.50
- VIDEO TITLE SHOP Companion 2\$7.50
- ALTERNATE REALITY (The City)\$12.50
- ALTERNATE REALITY The Dungeon\$12.50
- ZORRO Disk\$9.95
- SARACEN Disk\$9.95
- CROSSCHECK Disk\$9.95
- GUNSLINGER Disk (64K)\$9.95
- TOMAHAWK Helicopter Game 64K\$12.50
- THEATER EUROPE War Game\$12.50
- BRUCE LEE Disk\$12.50
- MERCENARY Disk\$12.50
- MERCENARY 2ND CITY DISK\$12.50
- THE GOONIES Disk\$12.50
- CONAN Disk\$12.50
- NEVER ENDING STORY (ENK)\$12.50
- 221-B BAKER STREET (64K)\$12.50
- 221-B BAKER ST. Case Library 1\$9.95
- 221-B BAKER ST. Case Library 2\$9.95
- Napolean in Russia (Borodino 1812) \$12.50
- BISMARCK\$12.50
- MOON SHUTTLE\$12.50

CALL TOLL FREE

1-800-551-9995

IN CALIF. OR OUTSIDE U.S.

CALL 415-352-3787

AMERICAN TECHNA-VISION

Mail Order: 15338 Inverness St., San Leandro, Ca. 94579

Sales Office: 2098 Pike Ave., San Leandro, Ca. 94577

Terms: NO MINIMUM ORDER. We accept money orders, personal checks or C.O.D.s. VISA, Master/Car okay. Credit cards restricted to orders over \$18.50. No personal checks on C.O.D. - Shipping: \$4.00 shipping and handling on orders under \$150.00. Add \$2.75 for C.O.D. orders. In Canada total \$6.00 for shipping and handling. Foreign shipping extra. Calif. residents include 7% sales tax. All products new and guaranteed. All sales final.

Prices subject to change without notice. Send SASE for free price list. Atari is a reg. trademark of Atari Corp.

BASIC TRAINING

by Clayton Walnum

As promised, this month we'll take a look at a simple but complete program in order to review, as well as to apply, much of what we've learned over the last few installments of *BASIC Training*. We'll also cover some new material; specifically, simple disk-access techniques and basic error-trapping.

Before we continue, you should type Listing 1, using BASIC Editor II to validate your work. Once you have the program typed properly, save it to disk.

Using the Program

Listing 1 is a simple address-book program that you can use to store addresses and phone numbers on disk. Run the program and press "C." You'll be asked for a filename. Type the filename, including the device (i.e., D:), and press Return. Now follow the prompts and enter as many addresses as you like. (Make sure you leave the disk in the drive.) To get back to the menu, type Return at one of the prompts.

After you've created your file, you can add more addresses by using the Load option. Press "L", then type in the filename. Enter whatever addresses you want and then press Return at any prompt to get back to the menu.

To see the addresses stored on the disk, press "V." You'll need to type the filename again, after which the addresses will be read in from the disk and displayed, four at a time, on the screen. Keep pressing Return until you've seen all the addresses.

Finally, to exit the program and return to BASIC's READY prompt, press "Q" and Return.

How it Works

Now that you're familiar with what the program does, let's see what makes it tick. (Do computers tick?)

Lines 10-90 do nothing but identify the program. All of your programs should have a series of remarks like this at the beginning.

Line 100 dimensions the string variables we'll be using in the program.

Lines 110-150 are more remarks.

A block of remarks is useful in partitioning sections of your program so you can more quickly find a particular portion of code. Remarks also help you remember what the program is doing. Every program can be broken up into a series of smaller tasks, just as we've done here. Always comment your programs; you'll be glad you did.

Line 160 clears the screen. (Never mind how it does it for the time being.)

Lines 170-250 print the menu to the screen.

See the question mark at the beginning of each line? That's Atari BASIC's abbreviation for the PRINT command. Also notice how a series of print statements lined up like this shows exactly what will appear on the screen.

Line 260 retrieves a menu selection from the user.

Lines 270-300 examine the user's input and route the program execution to the appropriate section of code.

Notice that, in this IF...THEN statement, we've used the word OR. If either of the expressions separated by the OR is true, the IF evaluates to true, and the THEN portion is executed. If both of the expressions are false, the IF evaluates to false, and program execution drops down to the next line, ignoring the THEN portion.

The opposite of OR is AND. When two expressions are separated by AND in an IF statement, both of them must be true in order for the IF statement to be true.

Confused? Let's look at this in more de-

tail. The following shows what would happen if the user typed an "L" in response to our menu:

Line 270: Is the letter that was typed a "C" or a "c"? No. Drop down to the next line.

Line 280: Is the letter an "L" or an "l"? Yes! The THEN portion of the statement is performed, causing the program to jump to line 450. (Notice that we don't need a GOTO after a THEN.)

Now let's see what happens when the user types the letter "H," a response not listed in our menu:

Line 270: Is the letter a "C" or a "c"? No. Drop down to next line.

Line 280: Is the letter an "L" or an "l"? No. Drop down to the next line.

Line 290: Is the letter a "Q" or a "q"? No. Drop down to the next line.

Line 300: Is the letter a "V" or a "v"? No. Drop down to the next line.

Line 310: None of the letters matched, so it's back up to Line 260 to get a new input.

You should have learned two things here. First, whenever you ask a user to type something in, you should check for both uppercase and lowercase responses. "C" and "c" are as different to the computer as "A" and "Z." But, because people tend to think of lowercase and uppercase as being the same, we should let them type a response either way.

You should have also noticed that if the user doesn't type one of the choices from the menu, we make him try again. "Error trapping" is an important part of writing user-friendly software. Unfortunately, due to space and time limitations, our program in Listing 1 is loaded with pitfalls for the user, places where the program may bomb due to incorrect input. (Try typing a filename without the device, for example.) But our menu is not one of them.

Line 370 gets the filename of the file the user wants to open.

Line 380 opens the requested file for output.

OPEN allows us to access files in a number of ways, and, as you can see, it has four parameters. The first parameter, preceded by the pound sign (#), is the channel we want to use as the data stream for our file. We may use any channel from one to seven. Once a file has been open, we use the channel number to refer to it, rather than the filename. The second parameter is a code that tells BASIC the type of operations we want to perform on the file. There are

five possible values:

- 4 - open for input only
- 6 - open for directory
- 8 - open for output only
- 9 - open for append
- 12 - open for input and output

When a file is opened with a 4, we may read data from the file, but we may not change the file in any way. When opened with a 6, we can read the disk's directory (a special file that contains the names of all the files on the disk). When opened with code 8, the file, if it doesn't already exist, is created on the disk, or, if it does exist, it is erased and restarted. After it is opened with code 8, a file may only have data written to it; we cannot read from the file. When opened with a 9, the file pointer, which marks where on the disk the next data will be written, is moved to the end of the file, where we may add more data. A file opened with code 12 can be used for both input and output operations.

This month we're concerned only with codes 4, 8 and 9. We'll cover the others eventually, but they're too complicated for this discussion, particularly code 12, which is used for random-access files and would require an entire article to explain thoroughly.

The fourth parameter in the OPEN command is the complete filename of the file to open. By complete, I mean that the device (D:, C:, D2:, etc.) must also be included. The filename may be stored in a string variable, as we've done in Line 380, or it may be a string literal, like this:

```
OPEN #1,4,0,"D2:MYFILE.DAT"
```

Notice that when you use a string literal for a filename, it must, just like any other string literal, be enclosed in quotes.

So, Line 380 opens a file for output. Be careful, though. This section of the program really should have some safeguards built in. If the user requests a file that already exists, it will be erased from the disk without any warning. In a complete program, we'd first check to see if the file already existed and, if it did, warn the user, giving him a chance to change his mind.

Line 310 sends program execution to the section of code that retrieves the addresses from the user.

Lines 450-460 get a filename from the

AFTER YOU'VE CREATED YOUR FILE, YOU CAN ADD MORE ADDRESSES BY USING THE LOAD OPTION.

user and open that file for append.

After the file is opened, any addresses typed will be added to the end of the file, without changing any of the addresses that were already stored there.

Lines 520-550 get an address from the user.

In order to allow the user to get back to the menu, we're using a statement like this:

```
IF NAME$="" THEN 620
```

As you can see, there's nothing between the quotes. In English, we're saying "If NAME\$ is empty, go to Line 620." A string will be empty if the user presses Return at the prompt without typing anything.

Line 560 prints to the screen the address just entered.

Line 570 asks the user if the information to a complete halt) whenever they encountered runtime errors. With a TRAP, however, we can grab an error before the user is ever aware of it and decide how to handle it ourselves.

The TRAP statement has only one parameter: the line number to which you want program execution to jump whenever an er-

ror is encountered. In Line 700, we're telling the computer to jump to Line 760.

Line 710 sets our address counter, COUNT, to zero and clears the screen.

We need to count the addresses displayed because we can fit only four addresses on the screen. After four addresses have been displayed, we want to stop and wait for the user to request more. If we didn't do this, the addresses at the top would scroll off as we added more at the bottom.

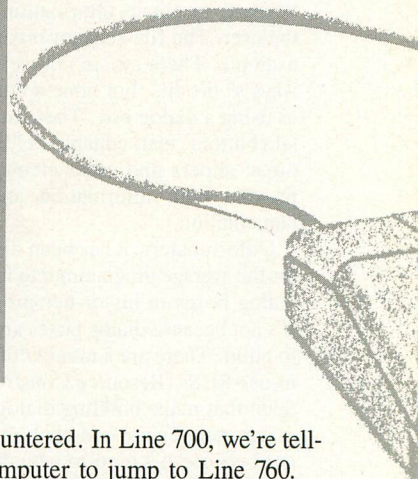
Line 720 reads an address from the disk file.

Just as with PRINT, the INPUT commands here work the same as the INPUT we've used to retrieve information from the screen (or from the keyboard, depending on how you look at it), the only difference being the pound sign and channel number. (Notice that we INPUT the addresses in exactly the same way that we PRINTed them.)

Line 730 prints to the screen the strings we retrieved from the file.

Line 740 increments our counter. If COUNT equals 4, we ask the user to press Return when he's ready to see more, after which we go back to the section of code beginning at Line 710, where we set COUNT back to 0 and get the next address.

(CONTINUED ON PAGE 127)



USING DIALOG BOXES WITH GFA BASIC 3.0

BY DAVID PLOTKIN

You've all seen dialog boxes in programs; they are GEM's "window on the world." The purpose of dialog boxes is to obtain information from the user. The file-selector box is one example. There are, of course, other ways to do this, but none so elegant as using a dialog box. They can contain buttons, text, editable fields, pictures, sliders and other elements to provide both information and ornamentation.

Unfortunately, it has been difficult for the average programmer to include dialog boxes in his or her program. It's not because dialog boxes are hard to build. There are a number of easy-to-use RCSs (Resource Construction Sets) that make building dialog boxes a matter of point-and-click. It's also not because it's hard to actually put the dialog box up on the screen. Just a few simple commands will load and draw the box. No, the problem has not been in getting a dialog box on the screen, but in getting the user-entered information back once he exits the box. How does the program get at that information?

The release of GFA BASIC 3.0 has considerably simplified the task of working with dialog boxes by providing a variety of commands that make it easier to retrieve the information. It's still not a trivial matter, but it's now much more straightforward. The purpose of this article (and the one to follow next issue) is to teach you what you need to know to build a dialog box and retrieve the information the user provides. When we get done, you should be able to build and use your own dialog boxes, making your GFA BASIC programs look more professional.

TYPES OF DIALOG BOX OBJECTS

The first thing to understand is that there are many types of objects that

can be used in dialog boxes. This should come as no surprise if you've seen some of the complex boxes that accompany commercial programs. The main types of objects that can be used in a dialog box are:

1. **G_BOX**: This is a rectangular box with adjustable parameters. Among these are its size, color, border thickness and fill pattern.

2. **G_IBOX**: This is a rectangular box identical to **G_BOX**, except that it can't have a fill pattern. The inside of the box is "I"nvisible.

3. **G_BOXCHAR**: This rectangular box is identical to **G_BOX** except that it can contain a single character. The character can have its color and size adjusted.

4. **G_BUTTON**: This is a box containing text that is always centered. The size of the box is adjustable.

5. **G_STRING**: A character string.

6. **G_TEXT**: Graphic text. The font size and text color can be varied, as can the writing mode (replace, transparent, etc.).

7. **G_BOXTEXT**: This is graphic text with a box around the text. In addition to the **G_TEXT** variations,

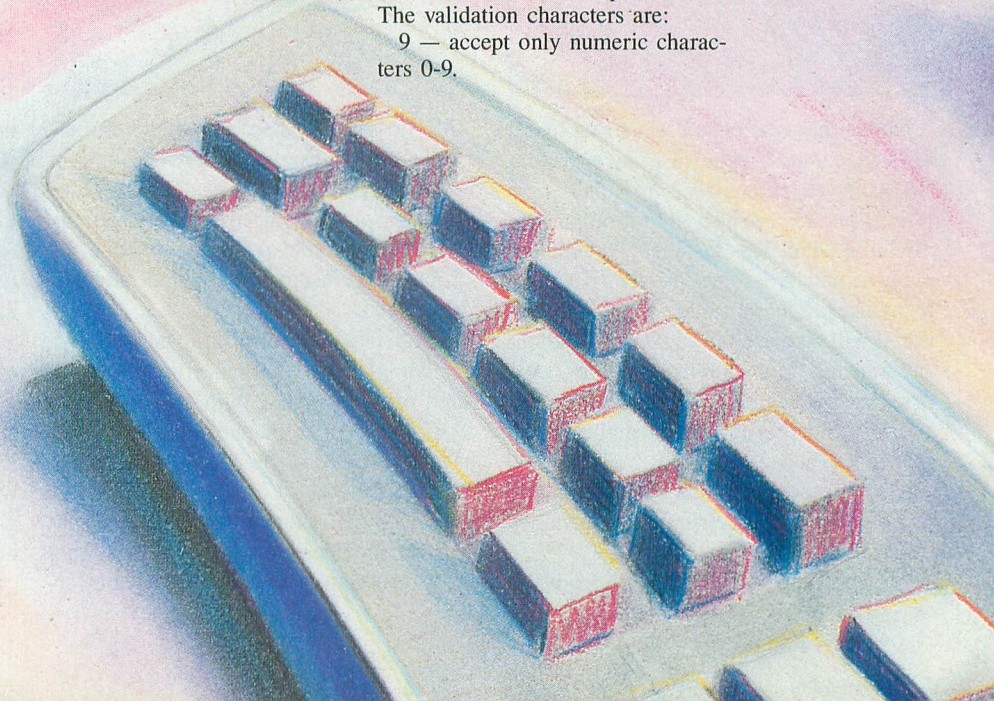
you can set the border width and color, fill pattern and color and the alignment of the text in the box.

8. **G_FTEXT**: This is the type of field used for editable text. It has three components. The first is the template field. Anything you enter here will be duplicated in the dialog box. For example, you could enter "Name: " in this field, and it will appear at the left side of this field in the dialog box. You could also enter something like "Date: ____/____/____." When the user actually goes to enter the date using the dialog box, the slashes would be automatically skipped as he or she types in the numbers.

The RCS that comes with GFA BASIC 3.0 seems to need placeholders to reserve the number of spaces you want to use for data entry. The symbol for the placeholder is the tilde (~). Everywhere you enter this symbol, an underline (__) will appear in the dialog box. Thus, the date example would actually have to be entered as "Date: ~/~/~."

The next field is the validation field. Here you specify what characters are acceptable. You must enter one character for each placeholder. The validation characters are:

9 — accept only numeric characters 0-9.





A — accept only uppercase alphabetical characters.

a — accept only upper or lowercase alphabetical characters.

N — accept uppercase alphabetical characters or numbers.

n — accept upper/lowercase alphabetical characters or numbers.

F — accept any character that can be part of a valid GEMDOS filename.

P — accept any character that can be part of a valid GEMDOS pathname.

p — same as P, except that wildcard characters (*,?) will not be accepted.

X — accept any character.

The last of the three components is the initial text field. Here you enter any string you want (usually a default) in the text field when it appears on the screen. You may enter blanks if you like, but you must enter as many characters as there are placeholders in the Template string. The color, size and writing mode of the text can be set.

9. **G_FBOXTTEXT**: This is identical to **G_FTEXT**, except it has a border around it. The thickness and color of the border can be set. A fill pattern can be selected as well (along with its color), but the fill pattern will be visible only in the portion of the box not fillable by text. That is, if the border of the box is sized to fit exactly around the maximum-size string it can contain, then the fill pattern will never be visible.

10. **G_ICON**: This is a graphic. A variety of icon editors is available for designing your own icons. If you make the icon selectable (see States) and you want it to change color when you do select it with your mouse, you must design a mask when using the icon editor. Icons can be selectable, just like buttons, and thus can serve as fancy graphic buttons.

NEW COMMANDS IN GFA BASIC

One of the things that makes working with dialog boxes easier is the addition of new commands to Version 3 of GFA BASIC. Previously, when you wanted to use the AES commands that accessed dialog box fields, you needed to construct a code that POKed in values, make a call to AES, and then jump through a series of memory addresses to get to where the data was actually stored. Commands to modify the dialog box were nearly as complicated.

With GFA BASIC 3.0, however, these complex lines of code have, in large part, been replaced with a single command. While it is outside the scope of this article to detail each command and its parameters (that's what you have the manual for, though

heaven knows its the very worst manual I've ever seen...), a brief summary of the more useful commands is in order so that you can follow along more easily:

1. **RSRC_LOAD(filename)** attempts to load your resource file (given by filename) from disk. The resource file can contain many types of resources, such as menus, alert boxes and dialog boxes. In this article, however, we are going to concentrate on dialog boxes. If the resource load is successful, this function returns a nonzero value. If it fails for any reason (i.e. the resource file is not available), it returns a zero.

2. **RSRC_GADDR(0,treenumber,adr%)** returns the address of a tree in memory. All resource structures, such as our dialog box, are identified by their tree. The address of the tree is important, as this number will be used in almost every call to a dialog-box function. The address is stored in the third parameter, *adr%*. The second parameter determines which tree to return the address of. If you have only one tree, *treenumber* would be zero. The first parameter can have other values besides zero, but again, explaining all the parameters is beyond the scope of this article.

3. **FORM_CENTER(adr%,x%,y%,w%,h%)** calculates the x, y, width and height coordinates of the dialog box with address *adr%* when it is located in the center of the screen, and places these values in the other parameters. While you don't have to put your dialog box in the center of the screen, that is where you most often want it located.

4. **FORM_DIAL(flag,minx,miny,minw,minh,maxx,maxy,maxw,maxh)** has a variety of functions, depending on the value of the parameter *flag*. However, the two most common are to create an expanding rectangle (used before drawing the dialog box, *flag* = 1) or a shrinking rectangle (used after you are through working with the dialog box and are ready to remove it from the screen, *flag* = 2). While these two effects are strictly cosmetic, they do give that "professional" look. Parameters 2-5 specify the minimum x, y, width and height of the expanding or shrinking rectangle, while the rest of the parameters specify the maximum values. Typically, the minimum value is zero, while the maximum values would correspond to the values returned by **FORM_CENTER**.

5. **OBJC_DRAW(adr%,start,end,x%,y%,w%,h%)** actually draws the dialog box given by *adr%* on the screen. *Start* specifies the starting level at which to begin drawing (0 if you are drawing the whole dialog box), while *end* specifies how many levels should be drawn (do not specify more

than seven). The concept of levels is not hard to understand graphically. The dialog box itself is at level 0. Any object contained in the dialog box is at level 1. Any object contained by a level-1 object is a level-2 object, and so on. It is important to note that this command does not activate the box for user input. This command can also be used to redraw only parts of the dialog box.

6. **FORM_DO(adr%fieldnumber)** turns control of the dialog box specified by *adr%* over to GEM so the user may interact with it: click on buttons, type in text, etc. This function returns the value of the object the user clicked on to exit the dialog box. GEM relinquishes control of the dialog box when you click on an "exit" button (see the next section). An example of an exit button is the button labeled "OK" in the file-selector box. It's important to note that, although GEM releases control of the dialog box, it does not erase it from the screen. It is up to the programmer to restore whatever was behind the dialog box. The parameter *fieldnumber* specifies which field the cursor should start in and is usually set to zero.

7. **OBJC_CHANGE(adr%,objnum,0,x%,y%,w%,h%,state,flag)** changes the state of the object given by *objnum* in the dialog box specified by *adr%*, usually an exit button. We'll talk more about "states" in the next section; however, an example of this would be when you click on the OK button, and it turns dark just prior to the dialog box disappearing from the screen. When the OK button turns dark, it is in its "selected" state. If you didn't use the **OBJC_CHANGE** command to return it to its "unselected" state, the OK button would still be dark the next time you activated the dialog box.

The parameter *state* gives the state that the object is to be changed to, and *flag* determines whether the object should be redrawn right away (1) or not (0). If the dialog box has been removed from the screen, there is no need to immediately redraw the object, because it will be redrawn the next time the dialog box is drawn on the screen with **OBJC_DRAW**. However, if you are immediately reentering **FORM_DO** to reuse the dialog box without using ***OBJC_DRAW**, then you should specify *flag* to be 1 to redraw the changed object.

8. **OB_STATE(adr%,obj)** returns the state of the object given by *obj*. States are described in the next section.

9. **OB_SPEC(adr%,obj)** returns the address of information about the object specified by *obj*. This is primarily of interest with buttons, text and editable fields, where

(CONTINUED ON PAGE 130)

(CONTINUED FROM PAGE 106)

```

7720 VU0 JSR GETTOKN ;get line
7730 STA LINENO ;number
7740 JSR GETTOKN
7750 STA LINENO+1
7760 CMP #500 ;at 32768?
7770 BCC VU1 ;no
7780 JSR DOCR ;yes try
7790 JMP LSBEG ;next variable
7800 ;
7810 VU1 JSR GETTOKN ;offset
7820 VU1 JSR GETTOKN ;offset
7830 JSR GETTOKN ;token
7840 CMP #502 ;rem /data
7850 BCS VU2
7860 VU3 JSR GETTOKN ;get byte
7870 CMP #EOL ;EOL yet
7880 BNE VU3 ;keep going
7890 BEQ VU0 ;next line
7900 ;look for end of statement
7910 ;or line
7920 VU2 JSR GETTOKN
7930 CMP #514 ;statmt end?
7940 BEQ VU1 ;next stmt
7950 CMP #51B ;THEN?
7960 BEQ VU1
7970 CMP #516 ;line end?
7980 BEQ VU0 ;next line
7990 CMP #50E ;num constant?
8000 BNE VU4 ;try string
8010 LDX #506 ;skip number
8020 VU7 JSR GETTOKN ;pull numbr
8030 DEX
8040 BNE VU7
8050 BEQ VU2
8060 VU4 CMP #50F ;a string?
8070 BNE VU6 ;no
8080 JSR GETTOKN ;length
8090 TAX ;skip it
8100 BNE VU7
8110 VU6 CMP MYTOKEN ;a variable?
8120 BNE VU2 ;nope
8130 LDA LINENO ;show line
8140 STA FR0 ;number
8150 LDA LINENO+1
8160 STA FR0+1
8170 JSR IFP ;INT to FP
8180 JSR FASC ;ATASCII
8190 INC PRFLAG ;screen
8200 LDA PRFLAG ;format
8210 CMP #6 ;5 per screen
8220 BNE VP2 ;line
8230 LDA #0
8240 STA PRFLAG ;reset
8250 JSR DOCR
8260 VP2 JSR MPRINT ;roll presses
8270 VP4 LDA #520 ;space
8280 JSR PUTC
8290 INY ;format with
8300 CPY #6 ;spaces so
8310 BNE VP4 ;in columns
8320 JMP VU2 ;try more lines
8330 ;
8340 ;get a token and increment
8350 ;the pointer in BASIC
8360 ;
8370 GETTOKN LDY #500
8380 LDA (LO),Y
8390 INC LO
8400 BNE NINC
8410 INC LO+1
8420 NINC RTS
8430 ;
8440 .INCLUDE #D:BUWP2.M65
8450 ;
8460 ;data
8470 ;
8480 ERMES .BYTE " # RORRE"
8490 BB1 .BYTE $9B,"egakcaP seiti"
8500 .BYTE "litu C15AB",$7D
8510 NDLIST .BYTE $70,$70,$42 ;new
8520 .WORD TLINE ;disply list
8530 .BYTE $01
8540 BACK .WORD $00 ;disp.1st flg
8550 TLINE .SBYTE " EXECUTI"
8560 .SBYTE "ING LINE: "
8570 .SBYTE " "
8580 ONOFF .BYTE 0 ;auto num flg
8590 TFLG #= #+1 ;trace flg
8600 DL5 #= #+1 ;save old
8610 DLH #= #+1 ;disp. list
8620 BCD #= #+4 ;bcd wrk area
8630 BFR0 #= #+1 ;psuedo FP
8640 BFR1 #= #+1 ;registers
8650 YSAV #= #+1 ;save y,x
8660 XSAV #= #+1
8670 PYSAV #= #+1 ;save y
8680 Y1 #= #+1 ;more save

```

```

8690 X1 #= #+1 ;x and y
8700 Y2 #= #+1
8710 X2 #= #+1
8720 CY1 #= #+1 ;ysave in cmd
8730 QFLAG #= #+1 ;quit num flg
8740 TOKTAB .BYTE 10,11,12,13,35,4
8750 OFFSET #= #+1 ;ren-stmt offst
8760 NXTSTM #= #+1 ;next st. offst
8770 PRFLAG #= #+1 ;print flg
8780 LSTKEY #= #+1 ;last key
8790 MYTOKEN #= #+1 ;save my token
8800 X3 #= #+1 ;save x
8810 DONMES .BYTE EOL,"derebmuner "
8820 .BYTE "margorp"
8830 OVRMES .BYTE $9B,"86723 => eniL"
8840 READ .BYTE EOL,"YDAER"
8850 TIMER #= #+1 ;vbi timer-num
8860 ENDLIN #= #+1 ;end line-num
8870 CHRCNT #= #+1 ;char. cnter
8880 VBLKDS #= #+2 ;save vtblkd
8890 FPMES .BYTE EOL,"rorre eulaV"
8900 HEXTAB .BYTE "0123456789ABCDEF"
8910 ENDPTR #= #+1 ;crunch
8920 MYEDIT #= #+16 ;ed. vector tab
8930 DIRBUF #= #+20 ;for directry
8940 DIRNAM .BYTE "D:*.*,EOL"
8950 NBUF #= #+8 ;# buffers-num
8960 M2BUF #= #+29 ;& ren
8970 .BYTE 0 ;for unify
8980 ;
8990 .OPT LIST
9000 ENDPRG #= #+1
9010 .OPT NO LIST
9020 #= $02E2
9030 .WORD IN1
9040 .END

```

Listing 3: Assembly

```

0100 ;SAVEND:BUWP2.M65
0110 ;
0120 ;-----;
0130 ;
0140 ; BUP.SYS part II ;
0150 ;
0160 ; (C) 1989 ;
0170 ; by Barry Kolbe ;
0180 ;
0190 ;-----;
0200 ;
0210 ;get line numbers
0220 ;
0230 RENUM JSR GETNBR
0240 JSR DOCR
0250 ;
0260 LDX #3 ;set prflag
0270 STX PRFLAG
0280 ;
0290 ;check if line>32768
0300 ;
0310 JSR GETPTR ;point to
0320 CLD ;first line
0330 NXTLN1 LDY #1 ;check hi byte
0340 LDA (LINPTR),Y ;of line #
0350 CMP #500 ;>32768?
0360 BCS LINOK ;yes-AOK
0370 LDA CLINE+1 ;will new line
0380 CMP #500 ;be >32768?
0390 BCS LINOVR ;yes-quit
0400 LDA CLINE ;add increment
0410 CLC ;to current
0420 ADC ILINE ;new line
0430 STA CLINE
0440 LDA CLINE+1
0450 ADC ILINE+1
0460 STA CLINE+1
0470 INY ;add offset
0480 LDA (LINPTR),Y ;to BASIC
0490 CLC ;current line
0500 ADC LINPTR
0510 STA LINPTR
0520 LDA LINPTR+1
0530 ADC #0
0540 STA LINPTR+1
0550 JMP NXTLN1 ;check more
0560 ;
0570 ; line >32768
0580 ;
0590 LINOVR LDX #13 ;show error

```

```

0600 L1 LDA OVRMES,X ;over the
0610 JSR PUTC ;max
0620 DEX
0630 BPL L1
0640 JMP EXIT2 ;to BASIC
0650 ;
0660 GETPTR LDA STMTAB ;point to
0670 STA LINPTR ;BASIC'S
0680 LDA STMTAB+1 ;first line
0690 STA LINPTR+1
0700 RTS
0710 ;
0720 FLNCLN LDA FLINE ;put new
0730 STA CLINE ;first line
0740 LDA FLINE+1 ;into
0750 STA CLINE+1 ;test line
0760 RTS
0770 ;
0780 ;renumbered lines will all
0790 ;be less than 32768
0800 ;
0810 LINOK JSR GETPTR ;reset pntrs
0820 JSR FLNCLN ;get first line
0830 CKTOK LDY #0 ;get low byte
0840 LDA (LINPTR),Y ;of line no.
0850 STA LINENO ;save it
0860 INY
0870 LDA (LINPTR),Y ;now high
0880 STA LINENO+1 ;byte
0890 CMP #500 ;done?
0900 BCC L2 ;no
0910 JMP DOREN ;do renumber!
0920 L2 INY
0930 LDA (LINPTR),Y ;offset to
0940 STA OFFSET ;next line
0950 INY ;offset to
0960 LDA (LINPTR),Y ;next stmt
0970 STA NXTSTM
0980 L4 INY
0990 LDA (LINPTR),Y ;get token
1000 LDX #5 ;is it a
1010 L3 CMP TOKTAB,X ;special?
1020 BNE L5
1030 JMP TOK1 ;yes-process
1040 L5 DEX ;try next
1050 BPL L3
1060 CMP #7 ;IF token
1070 BNE L7
1080 JMP TOK3 ;handle it
1090 L7 CMP #30 ;ON token
1100 BNE GTNXST ;do next stmt
1110 JMP TOK2 ;handle ON
1120 ;
1130 ;get next stmt or line
1140 ;
1150 GTNXST LDA NXTSTM ;is nxt stmt
1160 CMP OFFSET ;= to next line?
1170 BEQ NXTLN3 ;yes
1180 TAY ;no
1190 LDA (LINPTR),Y ;get offset
1200 STA NXTSTM
1210 JMP L4 ;look for token
1220 NXTLN3 CLC ;move to next
1230 LDA OFFSET ;line
1240 ADC LINPTR
1250 STA LINPTR
1260 LDA LINPTR+1
1270 ADC #0
1280 STA LINPTR+1
1290 CLC ;move up
1300 LDA CLINE ;new line
1310 ADC ILINE ;counter
1320 STA CLINE
1330 LDA CLINE+1
1340 ADC ILINE+1
1350 STA CLINE+1
1360 JMP CKTOK ;do next line
1370 ;
1380 ;do the renumbering
1390 ;
1400 DOREN JSR GETPTR ;pt to 1st line
1410 JSR FLNCLN ;new nmbrs
1420 DOR2 LDY #1 ;high byte of
1430 LDA (LINPTR),Y ;line no
1440 CMP #500 ;at 32768?
1450 BNE DOR1 ;no
1460 LDX #18 ;yes-say 'done'
1470 DORT LDA DONMES,X
1480 JSR PUTC
1490 DEX
1500 BPL DORT
1510 JMP EXIT2 ;to BASIC
1520 ;
1530 DOR1 DEY ;back up 1 byte
1540 LDA FLINE ;get new line
1550 STA (LINPTR),Y ;numbr
1560 LDA FLINE+1 ;and insert
1570 INY ;it into

```


by Frank Cohen



ASSEMB

"The Tao gave birth to machine language. Machine language gave birth to the assembler. The assembler gave birth to the compiler. Now there are ten thousand languages. Each language has its purpose, however humble. Each language expresses the yin and yang of software. Each language has its place within the Tao. But do not program in Cobol if you can avoid it."

The Tao Programmer

It never fails that during a typical phone conversation with a computer user, the subject of programming languages comes up.

"You programmed all that in assembly language?" asks the user in an unbelieving tone. The user assumes assembly language is a dark, oozing pond of scum and quicksand just waiting for its next victim.

The truth is that all programs are written in assembly language. High-level languages—Pascal, C, Cobol, Fortran, etc.—all convert pseudo-English programming statements into assembly code. The assembly code is then converted into machine code, which runs on your computer. For example, suppose a Pascal program is written to print the words "Hello, world" on the screen. The Pascal code might look like this:

```
begin
  WRITELN("Hello, world");
end;
```

WRITELN is Pascal's command to write a line of text to the screen. The Pascal commands are typed into an editor or word processor, then saved to a source-code file. The Pascal compiler program is opened, and it asks for the name of the source-code file. The Pascal compiler converts the Pascal source code into an assembly language program. The assembly code might look something like this:

```
BEGIN PEA myString
      MOVE #9, -(SP)
      TRAP #1
      ADD.L #6, SP
END
      RTS
myString DC.B 'Hello, world', 0
```

It probably doesn't look as pretty, but the assembly language program writes the same message to the screen. After a few moments of prestidigitation, the assembly code is turned into the machine code that will print your line of text.

So why not program everything in Pascal? Why use assembly language at all? And why do graham crackers fall apart after five minutes in milk?

During the translation process, the Pascal compiler must be able to convert the Pascal source code into assembly language regardless of what the source code is trying to do. The designers of Pascal had to consider every possible program that could be written in Pascal to make certain the compiler would work at all times.

Unfortunately, all the processing that's required to translate the source code into assembly code takes much time. When the Pascal compiler finally finishes the job, a significant portion of your life will have passed by. Also, the translation process is not perfectly accurate. The three lines of Pascal in our example above might convert into hundreds of lines of assembly code.

The insulation a high-level language offers the programmer from the computer and operating system can become the most significant problem. The simple Pascal program shown above has a 99.999% chance of running properly on your ST computer; however, what happens if it doesn't work? More importantly, suppose you have written 10,000 lines of good, clean Pascal source code. What happens when you run the program and 24 bombs appear across the screen? Can you really say you are fully in control of the computer? Is it your program

that's bombing? Or is the compiler the problem? In either case, your graham crackers get soggy while you fight to get your program to work.

Perhaps the question should be, "Why use a high-level language?"

It doesn't take long for the novice programmer to realize that the first half of every programming project is a repeat of the last project. Every program must initialize the computer, paint some pictures or text onto the screen, handle receiving input from the keyboard or mouse and be able to terminate (quit) to the program that called it. For the most part, these are basic functions of every program. Once written, they can be used over and over again. The master programmer eventually builds a library of basic functions that can be drawn from at the start of a new project. For that reason, assembly language isn't as hard to work with as you might think. Much of the code you write can be used over and over. In fact, programming in assembly language, using a complete library of subroutines, can be much like programming in a high-level language—but without the drawbacks.

"Something mysterious is formed, born in the silent void. Waiting alone and unmoving, it is at once still and yet in constant motion. It is the source of all programs. I do not know its name, so I will call it 'the Tao of Programming.'"

—The Tao Programmer

The arguments over programming in a high-level language or assembly language are moot. In today's world of high-performance personal computers, it is necessary to master all programming languages to complete a project. If the program is a large and complex database written in assembly language, a high-level language can easily create small utility programs that test the functions of the database. If a graph-

LYLINE

ics program is written in a high-level language, the functions to plot lines, circles and other patterns onto the screen will run faster if written in assembly language.

A renaissance computer-programming spirit has emerged, bringing humans out of the Dark Ages into a new day of enlightenment. It emphasizes the amazing versatility of mankind released from church doctrine and dogma. It lets us use our minds to use whatever materials we need to build a structure, develop a process and find the answers to life's perplexing questions. The personal computer frees us to use whatever language we need to get a task accomplished. More and more programmers are proving untrue the myth that high-level languages are better than assembly language.

"If the Tao is great, then the operating system is great. If the operating system is great, then the language is great. If the language is great, then the application is great. The user is pleased and there is harmony in the world." —*The Tao Programmer*

So, why should there be all this talk about Tao programming? Taoism is an ancient philosophy that teaches the natural relationship between man and the things around him, computers included. It can be thought of as the spirit of creativity, the energy that pushes an apprentice to learn new abilities.

In the time of the Asian emperors, warriors protected the land and people from invaders and enemies. Being a warrior was a noble craft, learned only after many years of training and apprenticeship. The master taught the student how to connect the energy of the soul (Tao) with the abilities of the body. Eventually, the student graduated by creating his own weapon.

Learning to program in assembly language follows a similar path. It might sound foolish to talk about ancient philosophies, but consider this: Luke Skywalker (in the

film, *Star Wars*) only fought and defeated Darth Vader after building his own weapon.

Each month we will explore the talents laying dormant in your mind. If you know how to switch the computer on and use a word processor, you know the basics of how a computer works. My task is to show you what is happening behind the windows and menus to make the computer function. We will work together to build a computer program, and eventually you will be able to do it yourself. You will build your own weapon.

The tools you gather will make the journey to assembly language programming much easier. To understand the material during your apprenticeship, you should have the following reference materials: *The 68000: Principles and Programming* by Leo J. Scanlon (Howard W. Sams & Co., Inc., 4300 W. 62nd Street, Indianapolis, IN 46268); *68000 Assembly Language Programming* by Gerry Kane, Doug Hawkins, Lance Levanthal (Osborne/McGraw-Hill, 2600 Tenth Street, Berkeley, CA 94710); *Atari ST Internals* by K. Gerits, L. Englisch, R. Bruckmann (Abacus Publishing, P.O. Box 7211, Grand Rapids, MI 49510); and *The Tao of Programming* by Geoffrey James (Info-books, P.O. Box 1018, Santa Monica, CA 90406).

These books should be kept next to your computer. They are invaluable at times when you scratch your head and wonder why your program has expired. They also present vastly different views of the same subject.

68000 Assembly Language Programming is a basic university-level course on programming, the most complete text available for the 68000 system. However, it is also very thick and sometimes bogs down with minute details. On the other hand, *68000 Principles and Programming* is a macho programmer's book. It is a bare-

bones book that assumes you have had experience with other programming languages. You will find many programming examples and instruction reference pages. *Atari ST Internals* does a good job describing the ST operating system functions available to programmers.

Kept in your computer are the tools that assemble and debug your assembly language source code. All programmers need a good editor, assembler, debugger and high-level language. For this column, I recommend the following programs: *HiSoft BASIC Professional*; *HiSoft Devpak ST* (Michtron, 576 S. Telegraph, Pontiac, MI 48053); and *Templemon* monitor (a "shareware" program available on DELPHI and most other online services; it is also on this issue's ST disk).

DevPak ST is a combination program editor and assembler. The new HiSoft products, marketed in the United States and Canada by MichTron, provide the beginning programmer with a decent editor, good assembler and great BASIC language. The examples and source code presented here will be created using *Devpak ST* and *HiSoft BASIC*.

The combination of *HiSoft BASIC* and *Devpak* was also chosen because they are the only fully integrated packages for ST programmers. Assembly language programs created in *Devpak* are easily incorporated into *HiSoft BASIC* programs. There might be a better individual assembler and high-level language, but they won't be able to use each other's resources to the extent of the HiSoft packages.

Bugs in your program have to be hunted down and smashed. *Templemon* is the best bug-swatting program for the ST. When your prized program crashes across your ST screen, *Templemon* shows you where and why the program failed. In the two years of use, *Templemon* has proven itself to be the most reliable, solid and stable debug-

ger available for the ST. And the best part is the price: *Templemon* is a shareware product. If you like it and use it, Thomas Templemann, the author, asks for a modest \$15 contribution to his bug-swatting cause.

Your final purchase of reference materials comes in the form of a subscription. DELPHI, CompuServe and GENIE are on-line services that provide an invaluable amount of utilities, source code examples and direct feedback from other programmers. If you are going to do anything with your newfound ability to program in assembly language, you will certainly need contact with other programmers.

"A well-written program is its own heaven; a poorly written program is its own hell."
The Tao Programmer

Over the next few months, we will journey where no programmer has gone before, seeking out new intelligence and interesting subroutines. At the end of our quest, the way to building your weapon will have been prepared, and you will emerge the new programming warrior.

The weapon we'll be building is a working text editor. The text editor will use GEM following the "Visual Interface Guidelines" that appeared in the February 1989 issue of ST-LOG. The editor will use the mouse to select words, manipulate windows to show text, handle drop-down menus, print the entered text onto paper, use the ST file system to store and retrieve text, and search and replace words and letters.

The text editor will be completed with instructions on how to write the *User Manual*. Many programmers gain the ability to create assembly language source code, while their brain loses the important ability of explaining the creation. The master programmer can write the documentation necessary for users to understand the program. There is an easy method of writing documentation that we will use with all of the functions, procedures and source code we develop. Compiling the *User Manual* can be done using the text editor we create.

Like a magician, once you learn the craft of assembly language programming, it is best to keep the public guessing as to how you did it. Don't reveal the hidden wires or the bunny waiting in your top hat.

When asked, "You programmed all that in assembly language?" pause a few moments while the person's mind contemplates long strings of numbers and complex mathematical formulas. Finally, reply, "Yes, but it really wasn't that hard." And bask in the glory of your new superior-being status. You will have become The Master Programmer. ■



Frank Cohen has been developing Atari programs since his first commercial product, Clowns & Balloons (written in 6502 assembly language). He later developed Regent Base, an SQL 4GL database (written in 68000 assembly language). He spends his free time meditating while watching the sun rise over Long Beach, California.

HOT WIRE

- Run any ST program simply by pressing a "hot" key...from the ST desktop!
- No more wading through folders to run programs.
- Up to 54 programs, documents, or configuration files may be installed in a single menu.
- Load and Save complete menus, with a keypress or with the GEM file selector.
- Run a program by clicking on its name in the HotWire menu, or by pressing its "hot" key from the Menu or the desktop!
- Call up the HotWire Menu instantly from the GEM desktop with the keyboard or the mouse!
- Unique "work file" features make HotWire an excellent shell for developers.
- HotWire will change the way you use your ST!

\$39.95

CODEHEAD Utilities

- A collection of very useful programs and desk accessories!
- **MultiFile:** an amazing multi-featured file maintenance utility that runs as either a desk accessory or a program!
- **AUTO Organizer:** reorganizes your AUTO folder...in a split second!
- **Art Gallery 2.0:** a multi-format graphics viewer/slideshow that runs as a desk accessory or a program!
- **Font Tricks 2.0:** Replaces the system font with a custom font, and prints it to an Epson printer! (19 fonts included.)
- **CodeCopy:** a fast disk copying program, designed for making multiple copies.
- **CodeHead Ramdisk and Print Spooler.**
- **ZeroDisk:** Erase floppy disks in a split second, without reformatting!
- And more!

\$29.95

CodeHead's MIDIMAX

- Create strings of MIDI commands triggered by any MIDI event (MIDI macros).
- Real-time multi-voice, multi-channel harmonization...single notes produce 18-note chords on any channel(s).
- Switch instantly to any one of 8 MIDI chord maps, and 8 sets of macros!
- Turn your ST into an intelligent THRU box, with unlimited keyboard splitting, filtering, and remapping.
- MIDIMAX runs as either a desk accessory or a program!

\$49.95

FatBack

- A FAST and intelligent hard disk backup system.
- Combines the speed of an "image" backup with the convenience of a "file" backup!
- Back up to floppy disks, or from one partition to another!
- Extract individual files from image backup!

\$34.95

MULTI-DESK

- Unlimited desk accessory power for the Atari ST!
- Break through the GEM limit of six desk accessories!
- Load and use up to 32 accessories or more at any time...without rebooting!
- Tested with hundreds of standard ST desk accessories!
- Flexible configuration options make it easy to customize MultiDesk.
- Run desk accessories as if they were programs!

\$29.95

G+PLUS

- The powerful and complete replacement for Atari GDOS!
- Join the ranks of users who are free of GDOS slowdown.
- Change fonts and device drivers for GDOS programs...without rebooting!
- Automatically load the correct ASSIGN files for each program you use.
- Special "solid" line mode actually speeds up all line drawing by about 25%!

\$34.95

Phone: (213) 386-5735
Visa and Mastercard accepted.
Shipping charge: US \$2,
Canada \$3, Europe \$5. CA
residents add 6.5% sales tax.

CodeHead Software
P.O. Box 74090
Los Angeles, CA 90004



CIRCLE #123 ON READER SERVICE CARD.

ProCopy ST BACKUP UTILITY

You can't backup your software because copy protection locks you out. **ProCopy** is the key!

- Protects against the accidental loss of expensive software
- Works with all Atari STs
- Copies both S/S & D/S disks
- Use with 1 or 2 disk drives
- Not copy protected
- **FREE** shipping and handling
- **TOLL-FREE** telephone number
- Updates available to registered owners
- Orders shipped same day
- Disk analyze function included

Dealer
Inquiries
Welcome



and C.O.D.
orders

Call (800) 843-1223

Send check for \$34.95 (overseas add \$2.00 for air mail) to:

PROCO PRODUCTS

P.O. BOX 665, CHEPACHET, RHODE ISLAND 02814 USA
(401) 568-8459

Available
in Europe

THE MOVING FINGER CO.
Building 2
Shamrock Quay
Southampton, SO1-1QL
England
Tel. 0703-229041

TECH-SOFT
COMPUTER WHOLESALE
324 Stirling Highway
Claremont
Western Australia 6100
Tel. 09-385-1885

CIRCLE #124 ON READER SERVICE CARD.



THE

ST:MACINTOSH

A Look at WriteNow 2.0

by Clayton Walnum

**WRITE NOW
BOASTS
MORE HELPFUL
FUNCTIONS
THAN YOU
CAN SHAKE
A STICK AT.**

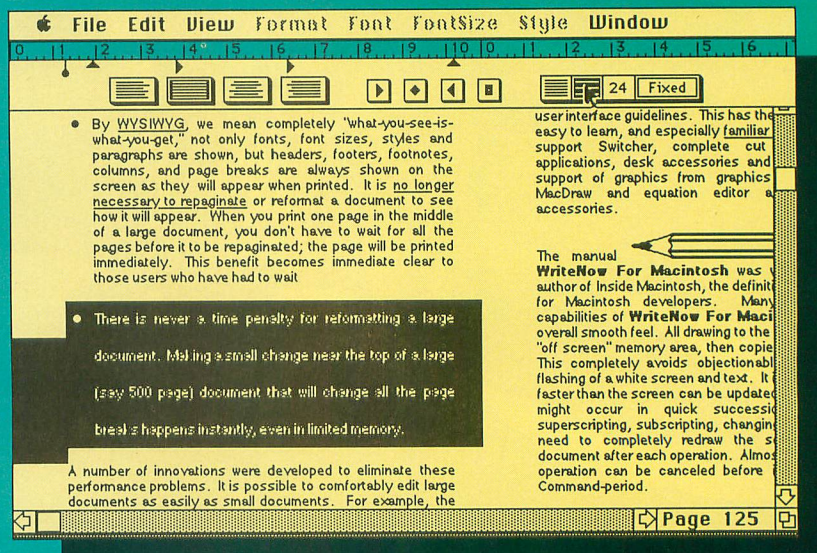
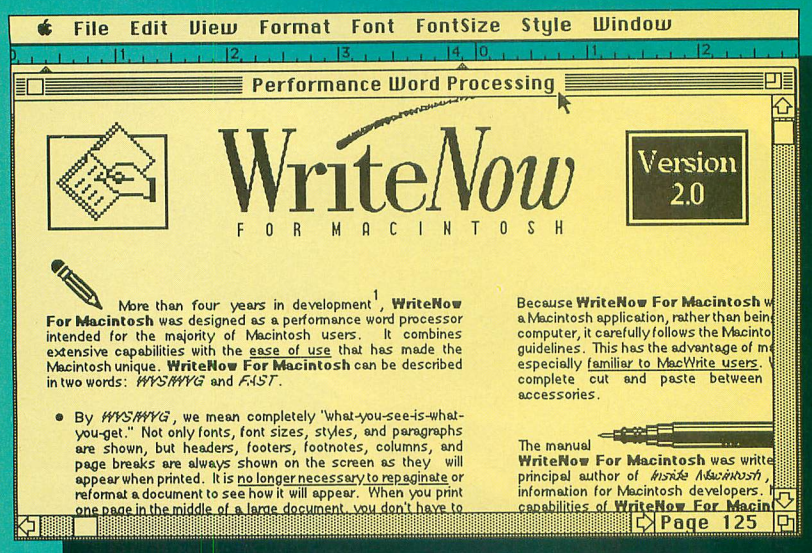
Since the release of the Magic Sac cartridge, ST owners have become more and more interested in running Macintosh software on their computers. The Spectre 128 cartridge, which was released by Gadgets by Small last year, has increased this interest enormously by allowing the ST to emulate a Mac Plus and thus run many software packages not usable on the Magic Sac.

Because Macintosh emulation on the ST has become so popular, we have decided to take a little space each month to give you a quick preview of the most interesting Macintosh software. Each piece of software we recommend here will run on an ST equipped with a Spectre 128 cartridge. We do not guarantee, however, that they will run on the Magic Sac. Some will, some won't.

Introducing *WriteNow*

No one can disagree that word processing is the number one use for a home computer. For that reason, our first ST Macintosh column will give you a preview of one of the top word processors for the Macintosh. *WriteNow 2.0* from T/Maker Company has taken the Macintosh world by storm, giving *MacWrite* serious competition for the Mac word-processing blue ribbon. Which of the two programs is better depends on whom you talk to, but every day, more and more people seem to be choosing *WriteNow*.

WriteNow is a graphics-oriented word processor, which actually comes close to a simple desktop-publishing program. It com-



combines the now-standard WYSIWYG (What You See Is What You Get) screen display with the ability to import pictures and have up to four columns of text per page. Fonts and font sizes, as well as text styles (bold, italic, underlined, shadowed, etc.) can be easily selected from the menu bar and will appear on-screen exactly as it will look in the final printout. Font sizes can be from four to 127 points. When a new font is chosen, the Font-Size menu will show, as outlined text, the point sizes that look best for that font, eliminating much trial and error.

While most word processors limit you to a certain number of headers or footers (sometimes only one), **WriteNow** can accommodate as many as you choose. For added flexibility, headers and footers can be set to appear only on odd or even pages. In addition, time and date "variable markers," which are automatically replaced with the current time and date both on the screen and at printout

time, can be placed in the headers or footers (or anywhere in a document, for that matter), assuring that your correspondence is always properly dated.

The Ruler

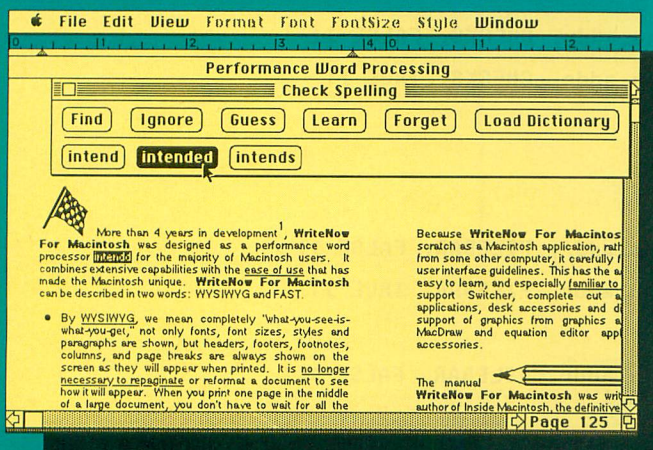
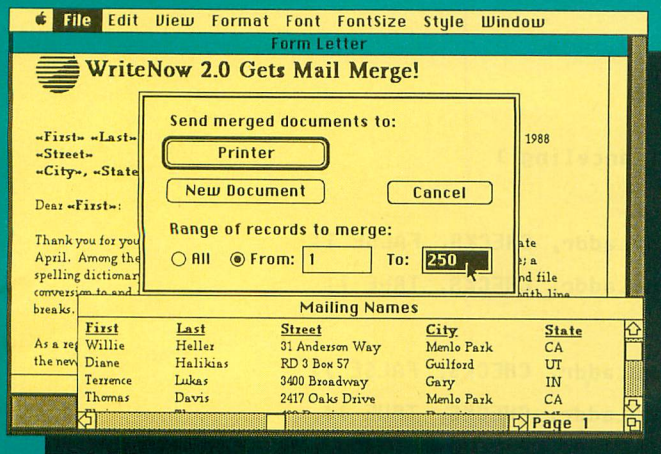
WriteNow formats each paragraph in your document according to a "ruler," which you may modify to whatever specifications you like. Each paragraph has its own ruler, and although most paragraphs will share the same style, any number of different rulers may be used within a document. Margins and tabs are set by dragging them with the mouse from the tab "toolbox," or from their current position on the ruler, to their new settings. Four types of tabs are available: left-justification tabs (the type you're accustomed to using), right-justification tabs, centering tabs and decimal tabs. Decimal tabs will align numbers at the decimal point.

Also available on the ruler are line-spacing

and text-alignment icons. A simple click of the mouse on one of these icons will cause a paragraph's text to be left-justified, both right- and left-justified, centered, or right justified. Line spacing is changed similarly.

Changing a paragraph's ruler is a simple process. Place the cursor anywhere in the paragraph you wish to modify, and click on the ruler window. The selected paragraph becomes highlighted, and its ruler appears. Any changes made to the ruler will be reflected immediately on the screen display.

Rulers may be copied and pasted, allowing you to use a specially formatted paragraph style in scattered places throughout a document without having to redo the ruler each time. This capability offers some of the advantages of "style sheets," which are usually available only in desktop-publishing programs. Clearly, **WriteNow's** ruler is one of its most powerful features and is what sets this program apart from its competition.



More Features

I mentioned previously that *WriteNow* allows the importing of graphics into your documents. In addition, any graphic element may be resized either proportionately or non-proportionately—another feature typically found in a desktop-publishing program.

Of course, no full-featured word processor would be complete without a spelling checker. *WriteNow*'s spelling checker sports both a 50,000- and 100,000-word dictionary. Which dictionary you load depends on available memory. The spelling checker is fast and will quickly find suggested spellings for unrecognized words. The program also has the ability to learn new words.

Above and beyond the major features, *WriteNow* boasts more helpful functions than you can shake a stick at. The search-and-replace function can operate on "non-printing" characters, such as spaces and carriage returns, and even supports "wild card"

searches where a question mark can represent any character. A "Select All" function allows you to mark instantly an entire document in order to perform global changes. With the "Revert to..." function, you can abandon the current document in favor of one of its previous versions up to two generations past. The intelligent Undo feature will not only undo your last action (if it's undoable), but will tell you what that last action was. You can "Show Markers" to see exactly where you've placed items, like headers and footers, "Show Space" to show non-printing characters, including spaces and returns, and "Hide Pictures" to speed up the display by not forcing the program to redraw complicated graphics. *WriteNow* also has full footnoting capabilities.

The Manual

No software package is complete without a manual, of course, and *WriteNow*'s manu-

al is terrific. The first few chapters comprise a hands-on tutorial that takes you through the program's main features step by step, with the rest of the volume giving a more detailed description of the program and its many functions. The manual is clearly written in a friendly, disarming style and includes a glossary and a complete, usable index.

Conclusion

If you've got a Spectre 128 installed on your ST, you might want to check out *WriteNow* 2.0, the winner of *MacUser*'s Editor's Choice award and the program about which Steve Jobs said, "This is the word processor that we designed and built the Macintosh for."

WRITENOW 2.0
\$195

T/Maker Company
1390 Villa Street
Mountain View, CA 94041
(415) 962-0195


```

handle_keys ()
{
    int button;

    if ( loaded && !search && !canceling )
        switch ( key ) {

            case CNTL_A:
                menu_tnormal ( menu_addr, CHECKS, FALSE );
                do_auto ();
                menu_tnormal ( menu_addr, CHECKS, TRUE );
                break;

            case CNTL_E:
                menu_tnormal ( menu_addr, CHECKS, FALSE );
                do_enter ();
                menu_tnormal ( menu_addr, CHECKS, TRUE );
                break;

            case CNTL_M:
                menu_tnormal ( menu_addr, FILEBAR, FALSE );
                do_new_mnth ();
                menu_tnormal ( menu_addr, FILEBAR, TRUE );
                break;

            case CNTL_P:
                menu_tnormal ( menu_addr, CHECKS, FALSE );
                do_check_canc ();
                menu_tnormal ( menu_addr, CHECKS, TRUE );
                break;

            case CNTL_R:
                menu_tnormal ( menu_addr, CHECKS, FALSE );
                do_reconcil ();
                menu_tnormal ( menu_addr, CHECKS, TRUE );
                break;
        }

    if ( !loaded )
        switch ( key ) {

            case CNTL_N:
                menu_tnormal ( menu_addr, FILEBAR, FALSE );
                do_newacct ();
                menu_tnormal ( menu_addr, FILEBAR, TRUE );
                break;

            case CNTL_O:
                menu_tnormal ( menu_addr, FILEBAR, FALSE );
                button = get_acct ();
                if ( button )
                    open_acct ( filename );
                menu_tnormal ( menu_addr, FILEBAR, TRUE );
                break;

            case CNTL_V:
                menu_tnormal ( menu_addr, UTILITY, FALSE );
                do_new_year ();
                menu_tnormal ( menu_addr, UTILITY, TRUE );
                break;

            case CNTL_I:
                menu_tnormal ( menu_addr, UTILITY, FALSE );
                do_import ();
                menu_tnormal ( menu_addr, UTILITY, TRUE );
                break;
        }

    switch ( key ) {

        case CNTL_Q:
            menu_tnormal ( menu_addr, FILEBAR, FALSE );
            do_quit ();
            menu_tnormal ( menu_addr, FILEBAR, TRUE );
            break;

        case CNTL_S:
            if ( loaded && !canceling ) {
                menu_tnormal ( menu_addr, CHECKS, FALSE );
                do_search ();
                menu_tnormal ( menu_addr, CHECKS, TRUE );
            }
            break;

        case CNTL_C:

```



```

        if ( loaded ) {
            menu_tnormal ( menu_addr, FILEBAR, FALSE );
            do_wind_close ();
            menu_tnormal ( menu_addr, FILEBAR, TRUE );
        }
        break;

    case CNTL_D:
        menu_tnormal ( menu_addr, UTILITY, FALSE );
        get_new_date ();
        menu_tnormal ( menu_addr, UTILITY, TRUE );
        break;

    case CNTL_W:
        if ( loaded ) {
            menu_tnormal ( menu_addr, PRINT, FALSE );
            print_wind ();
            menu_tnormal ( menu_addr, PRINT, TRUE );
        }
        break;

    case CNTL_G:
        if ( loaded ) {
            menu_tnormal ( menu_addr, PRINT, FALSE );
            print_reg ();
            menu_tnormal ( menu_addr, PRINT, TRUE );
        }
        break;
    }
}

do_newacct ()
{
    int choice, okay;
    int dial_x, dial_y, dial_w, dial_h;

    clear_newacct ();
    form_center ( newacct_addr, &dial_x, &dial_y, &dial_w, &dial_h );
    form_dial ( FMD_START, 0, 0, 10, 10, dial_x, dial_y, dial_w, dial_h );
    objc_draw ( newacct_addr, 0, 8, dial_x, dial_y, dial_w, dial_h );

    do {
        choice = form_do ( newacct_addr, NEWNAME );
        newacct_addr[choice].ob_state = SHADOWED;

        switch ( choice ) {

            case NEWOK:
                okay = check_newacct ();
                if ( !okay )
                    objc_draw ( newacct_addr, 0, 8,
                                dial_x, dial_y, dial_w, dial_h );
                else
                    newacct_file ();
                break;

            case NEWCANCL:
                clear_newacct ();
        }
    }
    while ( okay == FALSE && choice != NEWCANCL );

    form_dial ( FMD_FINISH, 0, 0, 10, 10, dial_x, dial_y, dial_w, dial_h );
}

check_newacct ()
{
    int x, okay;

    okay = TRUE;
    for ( x=NEWNAME; x<=NEWBALNC; ++x ) {
        string = get_tedinfo_str ( newacct_addr, x );
        if ( string[0] == '@' )
            okay = FALSE;
    }
    if ( !okay )
        form_alert(1, "[1][You must complete the form to start a new account!]\n[OK]");
    return ( okay );
}

```



```

newacct_file ()
{
    int choice, okay, x;
    int dial_x, dial_y, dial_w, dial_h;

    string = get_tedinfo_str ( newfile_addr, FILENAME );
    string[0] = 0;
    for ( x=0; x<64; filename[x++]=0 );
    newfile_addr[NEWOK].ob_state = SHADOWED;
    form_center ( newfile_addr, &dial_x, &dial_y, &dial_w, &dial_h );
    form_dial ( FMD_START, 0, 0, 10, 10, dial_x, dial_y, dial_w, dial_h );
    objc_draw ( newfile_addr, 0, 8, dial_x, dial_y, dial_w, dial_h );

    do {
        choice = form_do ( newfile_addr, FILENAME );
        newfile_addr[choice].ob_state = SHADOWED;

        switch ( choice ) {

            case FILEOK:
                okay = check_file ();
                if ( !okay )
                    objc_draw ( newfile_addr, 0, 8,
                                dial_x, dial_y, dial_w, dial_h );
                else {
                    string = get_tedinfo_str ( newfile_addr, FILENAME );
                    strcpy ( acct_name, string );
                    filename[0] = Dgetdrv () + 'a';
                    filename[1] = '.';
                    Dgetpath ( &filename[2], DFLT_DRU );
                    filename[strlen(filename)] = '\\';
                    strcpy ( &filename[strlen(filename)], string );
                    strcpy ( &filename[strlen(filename)], ".MCK" );
                    acctfile = fopen ( filename, "bw" );
                    if ( acctfile != 0 ) {
                        write_new_info ();
                        open_acct ( filename );
                    }
                }
                break;

            case FILECANC:
                string = get_tedinfo_str ( newfile_addr, FILENAME );
                string[0] = 0;
            }
        while ( !okay && choice != FILECANC );

        form_dial ( FMD_FINISH, 0, 0, 10, 10, dial_x, dial_y, dial_w, dial_h );
    }

clear_newacct ()
{
    int x;

    for ( x=NEWNAME; x<=NEWBALNC; ++x ) {
        string = get_tedinfo_str ( newacct_addr, x );
        string[0] = '@';
    }
    newacct_addr[NEWCANCL].ob_state = SHADOWED;
}

char *get_tedinfo_str ( tree, object )
OBJECT *tree;
int object;
{
    TEDINFO *ob_tedinfo;

    ob_tedinfo = (TEDINFO *) tree[object].ob_spec;
    return ( ob_tedinfo->te_ptext );
}

check_file ()
{
    int okay;

    okay = TRUE;
    string = get_tedinfo_str ( newfile_addr, FILENAME );
    if ( strlen ( string ) == 0 ) {
        form_alert(1, "[1][Invalid filename!][OK]");
        okay = FALSE;
    }
    return ( okay );
}

no_decimal ( s )
char *s;

```

(CONTINUED ON PAGE 35)

(CONTINUED FROM PAGE 113)

Line 750 sends the program back to the section of code beginning at Line 720 (after printing a blank line).

The only time we'll get to Line 750 is when COUNT is less than 4. Also notice that by going to Line 720, we are not setting COUNT back to 0.

Line 760 is where the program will go when it encounters an error.


What error are we expecting? The end-of-file error, of course, generated whenever we try to read more data than is contained

in a file. In this case, without the TRAP statement we'd get an "ERROR 136 AT LINE 720" on the screen, and the program would come to a halt—not an elegant way to treat our program's user. Rather than let this happen, we'll handle the error ourselves by sending the user back to the menu after informing him that we've reached the end of the file. The problem with this method is that *any* error will trigger the TRAP. We're only assuming the error encountered will be the end-of-file error. In a complete

program, that isn't good enough.

Line 770 waits for the user to press Return. Then it closes the open file and returns the user to the menu.

Conclusion

That finishes our discussion for this month. I hope that examining this program has helped you better understand what we've learned over the past few months. Next time, we'll take our simple address program and add enough error-trapping to make it foolproof. See you then. 

LISTING 1: BASIC

```

PM 10 REM *****
MI 20 REM *      BASIC TRAINING      *
KW 30 REM *      SIMPLE ADDRESS BOOK  *
PZ 40 REM *      by Clayton Walnum    *
U5 50 REM *
FG 60 REM *      Copyright 1989      *
RT 70 REM *      by ANALOG Computing  *
PT 80 REM *****
BG 90 REM
XR 100 DIM NAMES$(30),ADDRESS$(30),CITY$(30),
      PHONES$(15),A$(1),FILENAMES$(15)
QO 110 REM
YA 120 REM *****
BC 130 REM *      PRINT MENU      *
YE 140 REM *****
QW 150 REM
BE 160 ? CHR$(125)
JA 170 ? "
IE 180 ? "      CREATE FILE      "
EG 190 ? "
QC 200 ? "      LOAD FILE      "
DR 210 ? "
GE 220 ? "      VIEW FILE      "
DV 230 ? "
UP 240 ? "      QUIT      "
JJ 250 ? "
ZU 260 INPUT A$
GO 270 IF A$="C" OR A$="c" THEN 370
RE 280 IF A$="L" OR A$="l" THEN 450
TU 290 IF A$="Q" OR A$="q" THEN END
JO 300 IF A$="V" OR A$="v" THEN 680
OI 310 GOTO 260
QS 320 REM
YE 330 REM *****
NW 340 REM *      CREATE FILE      *
YI 350 REM *****
RA 360 REM
BR 370 ? "FILENAME";:INPUT FILENAMES$
AJ 380 OPEN #1,8,0,FILENAMES$
OK 390 GOTO 520
QP 400 REM
YB 410 REM *****
SI 420 REM *      LOAD FILE      *
YF 430 REM *****
QX 440 REM
BO 450 ? "FILENAME";:INPUT FILENAMES$

AT 460 OPEN #1,9,0,FILENAMES$
RD 470 REM
YP 480 REM *****
UX 490 REM *      GET ADDRESS      *
YA 500 REM *****
QS 510 REM
UR 520 ? :? "NAME:";INPUT NAMES$:IF NAMES$=""
      THEN 620
FY 530 ? :? "ADDRESS:";INPUT ADDRESS$:IF
      ADDRESS$="" THEN 620
MT 540 ? :? "CITY, STATE & ZIP:";INPUT CI
      TY$:IF CITY$="" THEN 620
ON 550 ? :? "PHONE:";INPUT PHONES$:IF PHON
      ES$="" THEN 620
WI 560 ? :? NAMES$:? ADDRESS$:? CITY$:? PH
      ONES$
EK 570 ? :? "IS THE ABOVE ENTRY OKAY (Y/N
      )";:INPUT A$
FT 580 IF A$="N" OR A$="n" THEN ? :? "PLE
      ASE REENTER THE ADDRESS.":GOTO 520
ST 590 IF A$("<"Y" AND A$("<"y" THEN ? :? "
      PLEASE ANSWER Y OR N.":GOTO 570
IG 600 ? #1;NAMES$:? #1;ADDRESS$:? #1;CITY
      $:? #1;PHONES$
NX 610 GOTO 520
BR 620 CLOSE #1;GOTO 160
QX 630 REM
YJ 640 REM *****
MO 650 REM *      VIEW FILE      *
YN 660 REM *****
RF 670 REM
BW 680 ? "FILENAME";:INPUT FILENAMES$
YO 690 OPEN #1,4,0,FILENAMES$
OS 700 TRAP 760
UQ 710 COUNT=0: ? CHR$(125)
ET 720 INPUT #1;NAMES$:INPUT #1;ADDRESS$:I
      NPUT #1;CITY$:INPUT #1;PHONES$
TR 730 ? NAMES$:? ADDRESS$:? CITY$:? PHONE
      $
GN 740 COUNT=COUNT+1:IF COUNT=4 THEN ? :?
      "PRESS [RETURN] FOR MORE":INPUT A$:GO
      TO 710
DP 750 ? :GOTO 720
WC 760 ? :? "END OF FILE.":? "PRESS [RETU
      RN] FOR MENU."
LS 770 INPUT A$:CLOSE #1;GOTO 160

```



FOOT NOTES NOTES

by
Karl E. Wiegiers

In every intellectual endeavor, certain individuals become widely recognized as gurus. These insightful chaps (and chapettes) offer unique perspectives on our thinking, making observations the rest of us immediately recognize as profound but would never have expressed on our own.

The computer industry is blessed with its share of such wise persons (I hope you think some of them write for ANALOG). In my readings of both the popular and the professional computer literature, I've come across a number of their statements that struck me as noteworthy, either for their pithiness or their humor. Some of them are most entertaining when presented out of context, and perhaps with a dash of selective editorial removal of intervening irrelevancies.

Below, I present some of the more interesting quotations I've encountered. I've added a few clarifying words in brackets where necessary. The works from which these quotes are drawn represent a broad cross-section of computer publications. I particularly recommend the books by Levy, Brooks and Peters. I've classified these tidbits of wisdom into two categories: software engineering and philosophy. Perhaps there's really no difference between the two.

Software Engineering

"Management is suspicious of all time spent prior to writing hard code. It is management's long-standing view that the project team really uses all the precoding phases to rest up before getting into the real work of building the system." (Tom DeMarco, *Structured Analysis and System Specification*, p. 33.)

"Now, after months of delay, some due to an unexpectedly long debugging period (there has never been an unexpectedly short debugging period in the history of computers)...the program was complete." (Steven Levy, *Hackers*, p. 382.)

"A dispassionate observer, a hypothetical person from Mars, might view programming as an activity dominated by testing, with occasional lapses into design and documentation." (Boris Beizer, *Software System Testing and Quality Assurance*, p. 1.)

"Programmers regard themselves as artists. As such, they consider keeping accurate records of their handiwork on a par with washing ashtrays." (Otis Port, *Business Week*, May 9, 1988, p. 144.)

"The second [information realm] is the realm of ideas and information existing in the minds of men and programmers." (James Martin, *Computer Data-Base Organization*, 2nd Ed., p. 50.)

"Because of their exaggerated claims, some of the fourth-generation languages qualify as the ultimate technical laetrile." (Tom DeMarco and Timothy Lister, *Computerworld Extra*, June 20, 1988, p. 45.)

"Reuseless code is software that is not worth reengineering for a new application... Sometimes, programmers should let old code die a natural death rather than spend any effort trying to revive it." (Will Tracz, *Computer*, December 1988, p. 75.)

"Structured programming has been credited with everything from reductions in dental caries to improvements in one's sex life." (Author unknown.)

"When a feature of any software works as advertised, regard it as a fortunate fortuity." (Larry Hiner, *Computer Technology Review*, May 1988, p. 20.)

"The next two Gemini flights would be afflicted by the same short landings... Gemini V landed 130 kilometers short after a computer programmer neglected to program a trivial point. He programmed the earth's rotation rate at 360 degrees per day, whereas the actual figure is 360.98." (Dennis Chamberland, U. S. Naval Institute *Proceedings*, January 1989, p. 40.)

Philosophy

"Is there something unique about the programming profession that attracts people who are mistake-prone by nature?... the only possible solution is the careful genetic breeding of today's best programmers." (Glenford J. Myers, *Software Reliability*, p. 15.)

"The computer, like a human employee, is subject to the Peter Principle. If it does good work at first, there is a strong tendency to promote it to more responsible tasks, until it reaches its level of incompetence." (Dr. L. J. Peters and R. Hull, *The Peter Principle*, p. 146.)

"I don't do crack. I don't do heroin. And I don't do desktop publishing." (Stephen Manes, *PC Magazine*, November 29, 1988, p. 85.)

"Long before anyone ever heard the acronym CASE [computer-aided software engineering], back in the computing Golden Age, building software systems was easy. Users petitioned the MIS [Management Information Systems] department with requests for new applications, and after an [appropriate time] MIS would pass their interpretation of the request back in the form of a finished system. For some on both sides, this was a religious experience." (Damian Rinaldi, *Software Magazine*, April 1988, p. 51.)

"Changing to any new program is, at best, a disruptive, time-consuming and inconvenient process. At worst, it's like getting a mouthful of root canal jobs in one afternoon." (Jim Seymour, *PC Magazine*, April 12, 1988, p. 78.)

"Why is programming fun? What delights may its practitioner expect as his reward? First is the sheer joy of making things... Second is the pleasure of making things that are useful to other people... Third is the fascination of fashioning complex puzzle-like objects of interlocking moving parts and watching them work in subtle cycles, playing out the consequences of principles built in from the beginning... Fourth is the joy of always learning, which springs from the nonrepeating nature of the task... Finally, there is the delight of working in such a tractable medium... Programming, then, is fun because it gratifies creative longings built deep within us and delights sensibilities we have in common with all men." (Frederick P. Brooks Jr., *The Mythical Man-Month*, p. 7.)

"I couldn't have said it better myself." (Karl Wiegiers, January 24, 1989.)



Karl Wiegiers, Ph.D., spent the '70s learning how to be an organic chemist, then spent the '80s wrestling with computers. He is now a software engineer in the Eastman Kodak Company Photographic Research Labs. He hasn't selected a career for the '90s yet.

ANALOG invites all authors to submit essays for possible use in the Footnotes column. Submissions should be no longer than 1,500 words, and may be on any aspect of Atari computing. Any style or type of essay is acceptable—opinion, humor, personal experience—but creativity is a plus. Submissions should be sent to: Footnotes, c/o ANALOG, P.O. Box 1413-M.O., Manchester, CT 06040-1413.

(CONTINUED FROM PAGE 117)

```

1580 STA (LINPTR),Y ;memory
1590 CLC
1600 LDA FLINE ;next new line
1610 ADC ILINE
1620 STA FLINE
1630 LDA FLINE+1
1640 ADC ILINE+1
1650 STA FLINE+1
1660 INY ;next BASIC
1670 LDA (LINPTR),Y ;line
1680 CLC
1690 ADC LINPTR
1700 STA LINPTR
1710 LDA LINPTR+1
1720 ADC #0
1730 STA LINPTR+1
1740 JMP DOR2 ;more!
1750 ;
1760 ;handle special tokens
1770 ;like GOTO, GO TO, GOSUB
1780 ;TRAP, RESTORE, LIST
1790 ;which precede line numbers
1800 ;
1810 TOK1 INY ;check for
1820 LDA (LINPTR),Y
1830 CMP #14 ;a number
1840 BEQ GOTN ;constant
1850 VAR JMP VARREF ;variable refer
1860 GOTN STY YSAV ;save position
1870 GOTN1 TYA ;skip 7 bytes
1880 CLC ;for constant
1890 ADC #7
1900 TAY
1910 LDA (LINPTR),Y ;get token
1920 CMP #18 ;comma?
1930 BEQ DONUM
1940 CMP #20 ;end stmt?
1950 BEQ DONUM
1960 CMP #22 ;end line?
1970 BNE VAR ;variable ref
1980 DONUM LDY YSAV
1990 JSR BCDINT ;BCD to integer
2000 ;
2010 ;now see if the new line exits
2020 ;
2030 LDA STMTAB ;point to
2040 STA SPTR ;line table
2050 LDA STMTAB+1
2060 STA SPTR+1
2070 LDA FLINE ;1st new line
2080 STA FR0
2090 LDA FLINE+1
2100 STA FR0+1
2110 SEARCH LDY #1 ;check if exits
2120 LDA (SPTR),Y ;high byte
2130 CMP #580 ;at end?
2140 BEQ NOTFND ;yes
2150 CMP RLINE+1 ;is it new line?
2160 BNE SNXTLN ;no
2170 DEY ;maybe-check
2180 LDA (SPTR),Y ;low byte
2190 CMP RLINE
2200 BNE SNXTLN ;nope
2210 ;
2220 ;got match fr0 has new line #
2230 ;
2240 JSR IFP ;INT to FP
2250 JSR INSERT ;put it in
2260 INY ;next token
2270 LDA (LINPTR),Y
2280 CMP #18 ;comma?
2290 BNE NOCOM ;no
2300 JMP TOK1 ;check for #
2310 NOCOM JMP GTNXST ;next stmt
2320 ;
2330 SNXTLN LDY #2 ;offset to
2340 LDA (SPTR),Y ;next line
2350 CLC
2360 ADC SPTR ;move up to it
2370 STA SPTR
2380 LDA SPTR+1
2390 ADC #0
2400 STA SPTR+1
2410 CLC
2420 LDA FR0 ;move up
2430 ADC ILINE ;new line
2440 STA FR0 ;to next one
2450 LDA FR0+1 ;in sequence
2460 ADC ILINE+1
2470 STA FR0+1
2480 JMP SEARCH ;keep looking
2490 ;
2500 ;show variable reference msg
2510 ;
2520 VARREF LDA #0 ;print 'VR-'
2530 JSR PUTC
2540 LDA #R
2550 JSR PUTC
2560 LDA #-
2570 JSR PUTC
2580 NF1 LDA CLINE ;at current
2590 STA FR0 ;line nmbr
2600 LDA CLINE+1
2610 STA FR0+1
2620 JSR IFP ;change to FP
2630 JSR FASC ;to ATASCII
2640 JSR PRNO ;print it
2650 JMP GTNXST ;next stmt
2660 ;
2670 ;show not found msg
2680 ;
2690 NOTFND LDA #M ;print 'NF-'
2700 JSR PUTC
2710 LDA #F
2720 JSR PUTC
2730 LDA #-
2740 JSR PUTC
2750 JMP NF1 ;print line #
2760 ;
2770 ;check for ON GOTO/GOSUB
2780 ;
2790 TOK2 INY
2800 LDA (LINPTR),Y
2810 CMP #23 ;ON GOTO
2820 BNE TK3 ;nope
2830 TK4 JMP TOK1 ;yup
2840 TK3 CMP #24 ;ON GOSUB
2850 BEQ TK4 ;yes
2860 BNE TOK2 ;try again
2870 ;
2880 ;check for IF-THEN
2890 ;
2900 TOK3 INY
2910 LDA (LINPTR),Y ;get token
2920 CMP #27 ;THEN?
2930 BNE TOK3 ;not yet
2940 INY ;get next token
2950 LDA (LINPTR),Y
2960 CMP #14 ;a constant?
2970 BNE NTNUM
2980 DEY ;yes
2990 JMP TOK1 ;process it
3000 NTNUM DEY ;back up 1 byte
3010 JMP L4 ;find if other to
3020 ;
3030 ;convert BCD to INTEGER
3040 ;
3050 BCDINT LDX #0 ;copy ATASCII
3060 BCD1 INY ;to FR0
3070 LDA (LINPTR),Y
3080 STA FR0,X
3090 INX
3100 CPX #6 ;6 bytes
3110 BNE BCD1
3120 JSR FPI ;FP to INT
3130 LDA FR0 ;save it
3140 STA RLINE
3150 LDA FR0+1
3160 STA RLINE+1
3170 RTS
3180 ;
3190 ;insert new line
3200 ;
3210 INSERT LDY YSAV ;get position
3220 LDX #0 ;move FP #
3230 IN51 INY ;to position
3240 LDA FR0,X ;in memory
3250 STA (LINPTR),Y ;of line
3260 INX
3270 CPX #6
3280 BNE IN51
3290 RTS
3300 ;
3310 ;print line number
3320 ;
3330 PRNO JSR MPRINT ;print it
3340 LDA #520
3350 JSR PUTC
3360 LDX PRFLAG
3370 DEX
3380 BPL XR2
3390 JSR DOCR
3400 LDX #3
3410 XR2 STX PRFLAG
3420 RTS
3430 ;
3440 ;print LBUF
3450 ;
3460 MPRINT LDY #5FF
3470 MPT INY
3480 LDA (INBUF),Y ;get byte
3490 PHA ;save it
3500 AND #57F ;mask inverse
3510 JSR PUTC ;put it
3520 PLA ;last byte
3530 BPL MPT ;is inverted
3540 RTS
3550 ;
3560 ;ren & num input checker
3570 ;also gets first line and
3580 ;increment from input line
3590 ;
3600 GETNBR JSR DEFAULT
3610 LDY #3
3620 LDA LBUF,Y ;no #'s?
3630 CMP #EOL
3640 BNE GETNMS
3650 RTS ;no #'s!
3660 ;
3670 ;default values are 10,10
3680 ;
3690 DEFAULT LDA #10
3700 STA FLINE ;first line
3710 STA CLINE
3720 STA ILINE ;and increment
3730 LDA #0
3740 STA FLINE+1
3750 STA ILINE+1
3760 STA CLINE+1
3770 RTS
3780 ;
3790 ;find the last program line
3800 ;
3810 GLAST JSR DEFAULT ;set defaults
3820 JSR GETPTR ;point to
3830 LDY #1 ;first line
3840 LDA (LINPTR),Y ;get high
3850 BPL GLOK ;< 32768?
3860 RTS ;no!
3870 GLOK STA 1 ;save line #
3880 DEY ;get low
3890 LDA (LINPTR),Y
3900 STA 0 ;save it
3910 JSR ADDLIN ;next line
3920 LDY #1 ;get high of
3930 LDA (LINPTR),Y ;line
3940 BPL GLOK ;< 32768
3950 LDA 0
3960 CLC
3970 ADC ILINE ;move up
3980 STA FLINE ;use last
3990 STA CLINE ;number
4000 LDA 1
4010 ADC #0 ;high byte
4020 STA FLINE+1
4030 STA CLINE+1
4040 RTS
4050 ;
4060 ;set up for first time use
4070 ;
4080 SETNUM CLD
4090 LDA FLINE ;back up
4100 SEC ;one increment
4110 SBC ILINE ;from first
4120 STA FLINE ;line
4130 LDA FLINE+1
4140 SBC ILINE+1
4150 STA FLINE+1
4160 RTS
4170 ;
4180 ;get numbers from input
4190 ;
4200 GETNMS LDX #0
4210 FNDCOM LDA LBUF,Y
4220 CMP #', ;comma?
4230 BEQ GOTFST
4240 CMP #EOL ;done yet?
4250 BEQ GOTFST
4260 STA NBUF,X ;copy number
4270 INX ;out to NBUF
4280 CPX #6
4290 BEQ FPERR ;error if
4300 INY ;too many digits
4310 JMP FNDCOM ;loop
4320 ;
4330 ;got first line #-convert it
4340 ;to integer form
4350 ;
4360 GOTFST STA NBUF,X ;, or EOL
4370 STY Y2 ;save position
4380 LDA #0 ;set up FP
4390 STA CIX ;stuff
4400 STA CLINE
4410 JSR DOINBF ;point to NBUF
4420 JSR AFP ;ATASCII to FP
4430 JSR FPI ;FP to INT
4440 BCS FPERR
4450 LDA FR0+1 ;is number
4460 BMI FPERR ;over 32768?
4470 STA FLINE+1
4480 STA CLINE+1
4490 LDA FR0
4500 STA FLINE

```

(CONTINUED ON PAGE 131)

(CONTINUED FROM PAGE 116)

this function is used to get the address of the stored text.

10. **OB_X(adr%,obj)** sets or returns the x position of the object specified by *obj* relative to its parent object. The concept of parent and child needs some explaining. Let's say you create an object that is completely contained within another object. A good example is a large box with several buttons within it. The large box is the "parent," while the buttons are the "children." Thus, this function would tell you the x coordinate of the specified button relative to the outer box.

To get the x coordinate: **X\$=OB_X(adr%,object&)** where the *adr%* is the address of the dialog box (returned by ***RSRC_GADDR**), and *object&* is the number of the object you want to know the x coordinate of (this number is set by the Resource Construction Set when you build the dialog box). To set the x coordinate: **OB_X(adr%,object&)=x&**.

11. **OB_Y(adr%,obj)** sets or returns the y position of the object specified by *obj* relative to its parent object.

12. **OB_W(adr%,obj)** returns the width of the object in pixels.

13. **OB_H(adr%,obj)** returns the height of the object in pixels.

14. **CHAR{address}** returns or sets the null-terminated string at the address specified. This function is extremely useful in finding out what text is in buttons, text fields and editable text fields, since these types of objects store their text as null-terminated strings. If we wanted to find out what text was in a **G_String-** or **G_Button-**type object, we would use the following: **X\$=CHAR{OB_SPEC(adr%,object&)}** because **OB_SPEC** returns the address of the text in a **G_String-** or **G_Button-**type object. To set the text in a **G_String-** or **G_Button-**type object, we would use **CHAR{OB_SPEC(adr%,object&)}=X\$**. Note that the length of **X\$** must not be longer than the field length set when designing the dialog box, or you may crash your machine.

Finding out what text is in text-type objects (**G_TEXT** or **G_BOXTEXT**) or formatted (editable) text-type objects (**G_FTEXT** or **G_FBOXTEXT**) is a little more complicated: **X\$=CHAR{{OB_SPEC(adr%,object&)}}**. Note the extra set of curly braces. **OB_SPEC** returns an address that points to another address, which in turn contains the address of the string. In **GFA BASIC**, the curly braces are placed around any term that will be handled as an address. Thus, this statement takes the ad-

dress returned by **OB_SPEC**, retrieves the address contained there (inner set of curly braces) and then reads the string contained at that address (**CHAR{}**).

Finally, to set the text in a **G_TEXT-**, **G_BOXTEXT-**, **G_FTEXT-** or **G_FBOXTEXT-**type object, use **CHAR{{OB_SPEC(adr%,object&)}}=X\$**. The same caveat about string length applies here as well.

STATES OF OBJECTS

The "state" of an object refers to a graphic condition of that object. Each of the following conditions corresponds to one bit in the variable that sets the state of the object:

1. **SELECTED** (Bit 0, 00 0001): This state indicates that the object was selected and is thus shown in inverse text. Typically, this state occurs for the exit button of a dialog box after the user clicks on it with the mouse. As noted above, you should use **OBJC_CHANGE** to set the object back to a state of **NONE** (0), otherwise the exit button will still be selected the next time the dialog box is drawn on the screen.

2. **CROSSED** (Bit 1, 00 0010): This state places a cross inside the object. It is only meaningful with box-type objects.

3. **CHECKED** (Bit 2, 00 0100): A check mark is drawn in the upper-left corner of the object.

4. **DISABLED** (Bit 3, 00 1000): The object is grayed out and cannot be selected. This is helpful in preventing the user from making selections in the dialog box that don't make sense at the current time (for example, a **NEXT** button when there is no next piece of data to look at).

5. **OUTLINED** (Bit 4, 01 0000): The object is drawn with a border. If it already had a border (a box-type object), then a second one is added.

6. **SHADOWED** (Bit 5, 10 0000): The object is supplied with a drop shadow.

SALUTING THE FLAGS

The attributes of the elements in a dialog box are set by their "flags." As with the state of an object, each flag detailed below corresponds to a bit set in the variable, which sets the object's attributes. There is even a command (**OB_FLAGS**) that returns the status of the flags for an object, but it is rarely used because the programmer already knows the attributes of each object; he or she designed the object with the Resource Construction Set, where these flags are set automatically. The common flags that can be set for an object are:

1. **SELECTABLE** (Bit 0, 0 0000 0001): This object may be selected by the user.

Most button objects must be selectable.

2. **DEFAULT** (Bit 1, 0 0000 0010): Setting this flag causes two things to happen. The first is that pressing the return key selects this object automatically. Second, the object is drawn with a bold border to inform the user which object is the default. A dialog box should have only a single **DEFAULT** object.

3. **EXIT** (Bit 2, 0 0000 0100): Selecting this object ends the **FORM_DO** command, returning control to the program. To select an object with an **EXIT** flag, the user must position the mouse pointer over the object, then press and release the left button (See **TOUCHEXIT**).

4. **EDITABLE** (Bit 3, 0 0000 1000): This object can be edited. It only makes sense for text fields to be **EDITABLE**.

5. **RBUTTON** (Bit 4, 0 0001 0000): This object is a radio button. It belongs to a group of radio buttons that are contained within a parent object. Radio buttons are special because only one of the group can be selected at any time. If you select one radio button of a group, the inside is darkened to show that it is selected. If you then select a different button, the new radio button is shown as selected, and the previously selected one is set back to the non-selected condition.

6. **TOUCHEXIT** (Bit 6, 0 0100 0000): Selecting this object ends the **FORM_DO** command and returns command to the program. However, unlike the **EXIT** flag, to select a **TOUCHEXIT** object the user needs merely to place the mouse pointer inside the object and press the left button; it is not necessary to release it. This object is handy with certain types of buttons, as we shall soon see.

CONCLUSION

This is all the room we have this issue. Next time, we'll pick up our discussion where we left off, as well as take a look at the source code for a dialog box written in **GFA BASIC 3.0**. See you then. ■



David Plotkin has been pounding the keys on Atari computers for almost ten years now. In that time he has written many memorable programs and articles, which have been published in a variety of magazines. He holds an M.S. in chemical engineering and is a data analyst for Chevron Corporation.

(CONTINUED ON PAGE 129)

```

4510 STA CLINE
4520 LDY Y2 ;get position
4530 LDA LBUF,Y ;check next
4540 CMP #EOL ;char. done?
4550 BNE GETSCN ; get second
4560 RTS
4570 ;
4580 ;got first line ,now get
4590 ;the increment
4600 ;
4610 GETSCN LDX #0
4620 GT1 INY
4630 LDA LBUF,Y ;get byte
4640 CMP #EOL ;done?
4650 BEQ NUMDON ;yes
4660 STA NBUF,X ;copy it
4670 INX ;too
4680 CPX #6 ;many digits?
4690 BEQ FPERR ;yes
4700 JMP GT1 ;more
4710 ;
4720 ;initialize INBUF
4730 ;
4740 DOINBF LDA # <NBUF ;point it
4750 STA INBUF ;to NBUF
4760 LDA # >NBUF ;where the
4770 STA INBUF+1 ;number is
4780 RTS
4790 ;
4800 ;show floating point error
4810 ;
4820 FPERR PLA ;remove return
4830 PLA
4840 VALERR LDX #11 ;'ERROR'
4850 FPER1 LDA FMES,X ;msg
4860 JSR PUTC
4870 DEX
4880 BPL FPER1
4890 JSR SETLBF ;reset FP
4900 JMP EXIT2 ;to BASIC
4910 ;
4920 ;got parameters so
4930 ;process second number
4940 ;
4950 NUMDON STA NBUF,X ;EOL
4960 LDA #0
4970 STA CIX ;point to NBUF
4980 JSR DOINBF
4990 JSR AFP ;to FP
5000 JSR FPI ;to INT
5010 BCS FPERR ;error!
5020 LDA FR0 ;put in
5030 STA ILINE ;increment
5040 LDA FR0+1
5050 STA ILINE+1 ;error if
5060 BMI FPERR ;>=32768
5070 JSR SETLBF ;reset FP
5080 RTS
5090 ;
5100 ;this is the entry point
5110 ;for the auto numbering
5120 ;some ideas from ANALOG
5130 ;issue 29 April 1985
5140 ;
5150 AUTNUM JSR GLAST ;get params.
5160 LDA #0
5170 STA ONOFF
5180 ;
5190 ;check for line overwrite
5200 ;
5210 JSR GETPTR ;point to
5220 NN1 LDY #0 ;first line
5230 LDA (LNPTR),Y ;get linenos
5240 STA RLINE ;low byte
5250 INY
5260 LDA (LNPTR),Y ;high byte
5270 STA RLINE+1 ;is line
5280 BMI NUMOK ;>=32768
5290 CMP FLINE+1
5300 BEQ CKLOW ;check low
5310 BCS NUMOK ;AOK
5320 NN2 JSR ADDLIN ;next line
5330 JMP NN1 ;loop
5340 ;
5350 CKLOW LDA RLINE ;check low
5360 CMP FLINE ;byte
5370 BEQ NOAUTO ;exits!
5380 BCS NUMOK ;ok
5390 BCC NN2 ;check next
5400 NOAUTO JMP EXIT2
5410 ;
5420 ;parameters ok
5430 ;
5440 NUMOK JSR SETNUM ;back up
5450 LDA #7 ;set VBI
5460 LDY # <VBNUM
5470 LDX # >VBNUM
5480 JSR SETVBV
5490 JSR SETLBF ;set FP pointers
5500 LDA #10 ;VBI timer
5510 STA TIMER
5520 LDA #0
5530 STA PROMPT ;for BASIC
5540 STA CHRCNT
5550 LDA #1
5560 STA ENDLIN
5570 STA ONOFF
5580 JMP EXIT2
5590 ;
5600 ;point to next BASIC line
5610 ;
5620 ADDLIN STY SLINE
5630 LDY #2
5640 LDA (LNPTR),Y ;get offset
5650 CLD
5660 CLC ;move pointer
5670 ADC LNPTR
5680 STA LNPTR
5690 LDA LNPTR+1
5700 ADC #0
5710 STA LNPTR+1
5720 LDY SLINE
5730 RTS
5740 ;
5750 ;get next basic line for VBI
5760 ;
5770 INCLIN JSR ADDLIN
5780 STY SLINE
5790 LDY #0
5800 LDA (LNPTR),Y ;get low
5810 STA RLINE ;store it
5820 INY
5830 LDA (LNPTR),Y ;high byte
5840 STA RLINE+1
5850 LDY SLINE
5860 RTS
5870 ;
5880 ;vertical blank interrupt
5890 ;
5900 VBNUM LDA ENDLIN ;BASIC done?
5910 BEQ XITVBI ;no
5920 LDA PROMPT
5930 BEQ XITVBI ;
5940 LDA TIMER ;time yet?
5950 BEQ PRNUM
5960 DEC TIMER ;count down
5970 JMP XITVBI
5980 PRNUM LDA #0
5990 STA ENDLIN
6000 LDA FLINE ;get line to
6010 CLC ;print
6020 ADC ILINE ;add increment
6030 STA FLINE
6040 STA FR0
6050 LDA FLINE+1
6060 ADC ILINE+1
6070 STA FLINE+1
6080 STA FR0+1
6090 OVRCHK LDA FR0+1 ;is it
6100 BMI VBQT ;over 32768?
6110 CMP RLINE+1 ;exist already?
6120 BEQ VCKLO ;maybe
6130 BCC VBOK ;ok
6140 JSR INCLIN ;look again
6150 JMP OVRCHK ;check it
6160 VCKLO LDA FR0 ;low bytes
6170 CMP RLINE ;match
6180 BEQ VBQT ;yes-exit
6190 BCC VBOK ;ok
6200 JSR INCLIN ;try next line
6210 JMP OVRCHK ;check it
6220 VBQT INC QTFLAG ;quit-exists
6230 LDA #12 ;force return
6240 STA 764
6250 JMP XITVBI ;out
6260 ;
6270 ;line # ok print it
6280 ;
6290 VBOK JSR IFP ;to ATASCII
6300 JSR FASC
6310 JSR MPRINT ;print it
6320 LDA #520 ;space
6330 JSR PUTC
6340 JSR SETLBF ;reset FP ptrs
6350 XITVBI JMP XITVBV ;leave

```


IN SIDE

this

ISSUE

**MORE
BOOT CAMP**

+

**END USER
DATABASE DELPHI**

